



Manuel de référence de Sun™ ONE Grid Engine 5.3 et Sun™ ONE Grid Engine, Enterprise Edition 5.3

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
+650-960-1300

Référence n° 816-7475-10
Septembre 2002, révision A

Envoyez vos commentaires sur ce document à : docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 États-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com et Solaris sont des marques de fabrique, des marques déposées ou des marques de service de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionnier de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non-exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE « EN L'ÉTAT » ET TOUTES AUTRES CONDITIONS, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.



Papier
recyclable



Adobe PostScript

Table des matières

1. Pages de manuel de Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3 1

Introduction	1
sgc_intro(1)	1
sgc_ckpt(1)	5
qacct(1)	7
qconf(1)	11
qdel(1)	38
qhold(1)	41
qhost(1)	44
qmake(1)	49
qmod(1)	53
qmon(1)	56
qrls(1)	66
qselect(1)	69
qstat(1)	72
qtcs(1)	82
submit(1)	85
access_list(5)	118
accounting(5)	119

calendar_conf(5) 121
checkpoint(5) 125
sge_request(5) 128
sge_aliases(5) 130
sge_conf(5) 132
sge_h_aliases(5) 151
sge_pe(5) 152
complex(5) 156
host_conf(5) 162
hostgroup(5) 166
project(5) 168
qtask(5) 169
queue_conf(5) 171
sched_conf(5) 182
share_tree(5) 188
user(5) 189
usermapping(5) 190
sge_commd(8) 192
sge_execd(8) 195
sge_qmaster(8) 198
sge_schedd(8) 201
sge_shadowd(8) 203
sge_sheperd(8) 205
sgecommdctl(8) 206

Tableaux

TABLEAU 1	Description des programmes de l'IHM	2
TABLEAU 2	Options de <code>qacct</code>	8
TABLEAU 3	Variables d'environnement de <code>qacct</code>	10
TABLEAU 4	Options de <code>qconf</code>	12
TABLEAU 5	Variables d'environnement de <code>qconf</code>	37
TABLEAU 6	Options de <code>qdel</code>	39
TABLEAU 7	Variables d'environnement de <code>qdel</code>	40
TABLEAU 8	Options de <code>qhold</code>	42
TABLEAU 9	Variables d'environnement de <code>qhold</code>	43
TABLEAU 10	Options de <code>qhost</code>	44
TABLEAU 11	Variables d'environnement de <code>qhost</code>	48
TABLEAU 12	Variables d'environnement de <code>qmake</code>	51
TABLEAU 13	Options de <code>qmod</code>	53
TABLEAU 14	Variables d'environnement de <code>qmod</code>	55
TABLEAU 15	Options de <code>qmon</code>	57
TABLEAU 16	Variables d'environnement de <code>qmon</code>	64
TABLEAU 17	Options de <code>qrls</code>	67
TABLEAU 18	Variables d'environnement de <code>qrls</code>	68
TABLEAU 19	Options de <code>qselect</code>	69
TABLEAU 20	Variables d'environnement de <code>qselect</code>	71

TABLEAU 21	Options de <code>qstat</code>	73
TABLEAU 22	Variables d'environnement de <code>qstat</code>	81
TABLEAU 23	Options de <code>qtcsch</code>	83
TABLEAU 24	Variables d'environnement de <code>qtsch</code>	84
TABLEAU 25	Options de la commande <code>submit</code>	89
TABLEAU 26	Variables d'environnement de <code>submit</code>	112
TABLEAU 27	Variables d'environnement supplémentaires de <code>qsub</code> , <code>qsh</code> et <code>qlogin</code>	112
TABLEAU 28	Variables définies dans l'environnement du travail par les commandes <code>submit</code>	113
TABLEAU 29	Options de <code>sge_cmmd</code>	193
TABLEAU 30	Variables d'environnement de <code>sge_cmmd</code>	194
TABLEAU 31	Options de <code>sge_execd</code>	196
TABLEAU 32	Variables d'environnement de <code>sge_execd</code>	197
TABLEAU 33	Options de <code>sge_qmaster</code>	199
TABLEAU 34	Variables d'environnement de <code>sge_qmaster</code>	200
TABLEAU 35	Option de <code>sge_schedd</code>	202
TABLEAU 36	Variables d'environnement de <code>sge_schedd</code>	202
TABLEAU 37	Variables d'environnement de <code>sge_shadowd</code>	204
TABLEAU 38	Options de <code>sgecommdcnt</code>	207
TABLEAU 39	Variables d'environnement de <code>sgecommdcnt1</code>	209

Préface

Le *Manuel de référence de Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3* est une compilation des pages de manuel (`man`) qui accompagnent ces deux produits. Ce manuel contient des informations qui sont, dans la mesure du possible, communes aux deux produits. Les informations spécifiques de l'un ou l'autre des produits sont signalées.

Structure de cet ouvrage

Outre cette préface et la table des matières, ce manuel est composé d'un unique chapitre.

- Pages de manuel de Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3

Utilisation des commandes UNIX

Ce document ne contient pas d'informations sur les commandes et les procédures de base d'UNIX®, telles que l'arrêt ou le démarrage du système, ou encore la configuration des périphériques.

Consultez les documents suivants pour plus d'informations :

- *Guide des périphériques Sun Solaris* ;
- Documentation en ligne AnswerBook2™ relative à l'environnement logiciel Solaris™ ;
- Autres documentations accompagnant les logiciels livrés avec votre système.

Conventions typographiques

Police ou symbole	Signification	Exemples
AaBbCc123	Noms de commandes, fichiers et répertoires ; messages-système	Editez votre fichier <code>.login</code> . Utilisez <code>ls -a</code> pour obtenir la liste de tous les fichiers. % Vous avez du courrier en attente
AaBbCc123	Caractères saisis par l'utilisateur, par opposition aux messages du système	% su Mot de passe :
<i>AaBbCc123</i>	Titres de publications, nouveaux termes ou mots en évidence. Remplacez les variables de ligne de commande par un nom ou une valeur.	Reportez-vous au chapitre 6 du <i>Guide de l'utilisateur</i> . On parle d'options <i>classe</i> . Vous <i>devez</i> être le superutilisateur pour effectuer cette opération. Pour supprimer un fichier, tapez <code>rm nom du fichier</code> .

Invites de shell

Shell	Invite
C	<i>nom_machine</i> %
Super-utilisateur C	<i>nom_machine</i> #
Bourne et Korn	\$
Super-utilisateur Bourne et Korn	#

Documentation connexe

Application	Titre	Référence
Administration et utilisation	<i>Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3</i>	816-7469-10
Administration et utilisation	<i>Guide d'administration et d'utilisation de Sun Grid Engine 5.3</i>	816-7462-10

Documentation Sun en ligne

Une large sélection de la documentation système Sun se trouve sur :

<http://www.sun.com/products-n-solutions/hardware/docs>

Vous trouverez l'ensemble de la documentation Solaris et de nombreux autres titres sur :

<http://docs.sun.com>

Ce site contient également des informations pour la commande de copies *papier* de ce guide.

Vos commentaires sont les bienvenus

Nous souhaitons améliorer notre documentation. Vos commentaires et suggestions sont donc les bienvenus. Vous pouvez nous les envoyer par courrier électronique à :

docfeedback@sun.com

N'oubliez pas d'indiquer le numéro de référence (816-7475-10) de votre document dans l'espace réservé à l'objet de votre courrier électronique.

Pages de manuel de Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3

Introduction

Ce document contient les pages de manuel telles qu'incluses dans le logiciel Sun™ Grid Engine distribué. Vous remarquerez que nombre des commandes listées dans ce document ne s'appliquent qu'au produit Sun Grid Engine, Enterprise Edition 5.3, comme indiqué dans la description de ces mêmes commandes.

`sge_intro(1)`

Nom

Sun Grid Engine Introduction – Utilitaire permettant l'exécution de travaux UNIX sur des machines distantes.

Description

Les logiciels Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3 — auxquels il est fait référence par Sun Grid Engine dans la plupart des descriptions de ce manuel — sont tous deux des produits permettant d'exécuter des travaux par lots UNIX (scripts shell) sur un groupe de stations de travail coopérant

entre elles. Les travaux sont mis en attente et exécutés à distance sur des stations de travail à des heures où sinon ces stations de travail seraient inactives ou peu chargées. La charge de travail est répartie entre les stations de travail de la grappe en fonction de la situation de charge de chaque machine et des ressources requises par les travaux.

Les programmes d'écriture de points de reprise de niveau utilisateur sont pris en charge et un mécanisme d'écriture de points de reprise transparent est fourni (voir `sge_ckpt(1)`). Les travaux de point de reprise migrent de station de travail en station de travail sans l'intervention de l'utilisateur au niveau de la demande de charge. En plus des travaux par lots, il est possible de soumettre à Sun Grid Engine des travaux interactifs et parallèles.

Interface utilisateur

L'interface homme-machine (IHM) de Sun Grid Engine se compose de plusieurs programmes, qui sont décrits séparément dans le TABLEAU 1.

TABLEAU 1 Description des programmes de l'IHM

Nom du programme	Description
<code>qacct(1)</code>	Extrait des informations de compatibilité arbitraires du fichier journal de la grappe.
<code>qalter(1)</code>	Change les caractéristiques des travaux déjà soumis.
<code>qconf(1)</code>	Fournit l'IHM pour la configuration, la modification, la suppression et l'interrogation des files d'attente, et la configuration de la grappe.
<code>qdel(1)</code>	Fournit à un utilisateur/opérateur/gestionnaire des moyens d'annuler des travaux.
<code>qhold(1)</code>	Empêche l'exécution des travaux soumis en les suspendant.
<code>qhost(1)</code>	Affiche des informations d'état sur les hôtes d'exécution de Sun Grid Engine.
<code>qlogin(1)</code>	Lance une session telnet ou similaire en sélectionnant automatiquement un hôte adapté et peu chargé.
<code>qmake(1)</code>	Remplace l'utilitaire <code>make</code> UNIX standard. Il a un rayon d'action plus étendu que <code>make</code> car il est en mesure de distribuer des étapes <code>make</code> indépendantes à travers une grappe de machines adaptées.
<code>qmod(1)</code>	Permet à un ou plusieurs propriétaires de files d'attente d'interrompre et d'activer toutes les files d'attente associées à leur machine (tous les processus actuellement actifs dans cette file d'attente sont également concernés) ou d'interrompre et d'activer les travaux s'exécutant dans les files possédées.

TABLEAU 1 Description des programmes de l'IHM (*suite*)

Nom du programme	Description
qmon(1)	Fournit une interface de commande Motif à toutes les fonctions de Sun Grid Engine. Le statut de toutes les files d'attente ou d'une sélection personnelle de celles-ci s'affiche en ligne en changeant les couleurs des icônes de file d'attente correspondantes.
qresub(1)	Crée de nouveaux travaux en copiant des travaux actuellement en cours ou en attente.
qrls(1)	Retire les suspensions des travaux auxquels une suspension a été attribuée au préalable au moyen de, par exemple, qhold(1) (voir ci-dessus).
qrsh(1)	Peut être utilisé à de nombreuses fins telles que l'exécution à distance d'applications interactives via Sun Grid Engine comparable à l'utilitaire UNIX standard rsh, pour autoriser la soumission de travaux par lots qui, à l'exécution, prennent en charge les E/S de terminal (sortie d'erreur/standard et entrée standard) et le contrôle de terminal, pour fournir un client de soumission de travaux par lots qui reste actif jusqu'à ce que le travail soit fini ou pour autoriser l'exécution à distance contrôlée par Sun Grid Engine des tâches de travaux parallèles.
qselect(1)	Imprime la liste des noms des files d'attente qui correspondent aux critères de sélection spécifiés. La sortie de qselect est en général utilisée dans d'autres commandes de Sun Grid Engine pour appliquer des actions à un ensemble sélectionné de files d'attente.
qsh(1)	Ouvre un interpréteur de commandes interactif (dans un xterm(1)) sur un hôte peu chargé. Il est possible d'exécuter toute sorte de travail interactif dans cet interpréteur de commande.
qstat(1)	Fournit la liste des statuts respectifs des différents travaux et files associés à la grappe.
qsub(1)	IHM pour la soumission d'un travail Sun Grid Engine.
qtcs(1)	Remplace (compatible à 100%) la très connue et répandue dérivée UNIX C-Shell (csh) tcsh. Fournit un interpréteur de commandes avec une extension permettant de distribuer de manière transparente l'exécution d'applications désignées sur des hôtes adaptés et peu chargés via Sun Grid Engine.

Voir aussi

sge_ckpt(1), qacct(1), qalter(1), qconf(1), qdel(1), qhold(1), qghost(1), qlogin(1), qmake(1), qmod(1), qmon(1), qresub(1), qrls(1), qrsh(1), qselect(1), qsh(1), qstat(1), qsub(1), qtcs(1), *Guide d'administration et d'utilisation de Sun Grid Engine 5.3* et *Guide d'administration et d'utilisation de Sun Grid Engine Enterprise Edition 5.3*.

Droits d'auteur

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 États-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com et Solaris sont des marques de fabrique, des marques déposées ou des marques de service de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionnier de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non-exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE « EN L'ÉTAT » ET TOUTES AUTRES CONDITIONS, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.

sgc_ckpt (1)

Nom

Sun Grid Engine Checkpointing – Mécanisme et prise en charge de l'écriture de points de reprise de Sun Grid Engine

Description

Sun Grid Engine prend en charge deux niveaux d'écriture de points de reprise : le niveau utilisateur et un niveau transparent fourni par le système d'exploitation. L'écriture de points de reprise de niveau utilisateur fait référence aux applications, qui mettent en oeuvre leur propre système d'écriture de points de reprise en écrivant des fichiers de redémarrage à certaines heures ou étapes algorithmiques et en traitant correctement ces fichiers de redémarrage au redémarrage.

L'écriture de points de reprise transparente doit être fournie par le système d'exploitation et est en général intégrée dans le noyau du système d'exploitation. Un exemple d'utilitaire d'écriture de points de reprise intégré dans le noyau est le module Hibernator de Softway pour les plates-formes SGI IRIX.

Les travaux dits de point de reprise doivent être identifiés par le système Sun Grid Engine en utilisant l'option `-ckpt` de la commande `qsub(1)`. L'argument de cet indicateur fait référence à un environnement dit de point de reprise, qui définit les attributs de la méthode d'écriture de points de reprise à utiliser (voir `checkpoint(5)` pour plus de détails). Les environnements de point de reprise sont configurés par les options de `qconf(1)` `-ackpt`, `-dckpt`, `-mckpt` et `-sckpt`. L'option `qsub(1) -c` peut être utilisée pour écraser l'attribut `when` pour l'environnement de points de reprise référencé.

Si une file d'attente est de type *POINT DE REPRISE*, les travaux doivent avoir l'attribut d'écriture de points de reprise marqué (voir l'option `-ckpt` de `qsub(1)`) pour pouvoir s'exécuter dans une telle file. Contrairement à ce qui se passe avec les travaux par lots normaux, les travaux de point de reprise sont abandonnés dans des conditions dans lesquelles les travaux par lots ou interactifs sont simplement interrompus, voire même non-affectés. Ces conditions sont les suivantes :

- L'interruption explicite de la file d'attente ou du travail via `qmod(1)` par l'administration de la grappe ou le propriétaire d'une file si le spécificateur d'occasion `x` (voir `qsub(1) -c` et `checkpoint(5)`) a été attribué au travail.
- Une valeur de charge moyenne dépassant les seuils de migration configurés pour les files d'attente correspondantes (voir `queue_conf(5)`).

- L'arrêt du démon d'exécution de Sun Grid Engine `sge_execd(8)` responsable du travail de point de reprise.

Après l'abandon, les travaux migreront dans d'autres files à moins qu'ils n'aient été soumis dans une file spécifique par une requête spécifique d'un utilisateur. La migration des travaux mène à un équilibrage de charge dynamique.

Remarque – Abandonner des travaux de point de reprise libérera toutes les ressources (mémoire, zone de swap) que ces travaux occupaient. C'est le contraire de ce qui se passe avec les travaux normaux interrompus qui continuent à occuper la zone de swap.

Restrictions

À l'heure actuelle, lorsqu'un travail migre dans une file située sur une autre machine aucun fichier n'est transféré automatiquement sur cette machine. Cela signifie que tous les fichiers qui sont utilisés au cours du travail, fichiers de redémarrage, exécutables et fichiers temporaires de travail compris, doivent être visibles ou transférés de manière explicite (par ex. au début du script du travail).

Il existe aussi certaines limites pratiques relatives à l'utilisation de l'espace disque pour l'écriture transparente de points de reprise pour des travaux. Les points de reprise d'une application bénéficiant de l'écriture transparente de points de reprise sont en général stockés dans un fichier ou un répertoire de point de reprise par le système d'exploitation. Ce fichier ou répertoire contient l'ensemble du texte, des données et de l'espace de pile pour le processus, ainsi que certaines informations de contrôle supplémentaires. Cela signifie que les travaux qui utilisent un espace d'adressage virtuel très important généreront des fichiers de point de reprise de très grande taille. De même, les stations de travail sur lesquelles les travaux s'exécuteront risquent d'avoir peu d'espace disque libre. Il n'est donc pas toujours possible de transférer un travail de point de reprise transparent sur une machine, même si cette dernière est inactive. Étant donné que les travaux monopolisant une mémoire virtuelle importante doivent attendre une machine inactive qui offre suffisamment d'espace disque libre, ces travaux risquent d'avoir des délais d'exécution particulièrement longs.

Voir aussi

`sge_intro(1)`, `qconf(1)`, `qmod(1)`, `qsub(1)`, `checkpoint(5)`,
Guide d'administration et d'utilisation de Sun Grid Engine 5.3 et *Guide d'administration et d'utilisation de Sun Grid Engine Enterprise Edition 5.3*.

Droits d'auteur

Voir `sg_e_intro(1)` pour l'énoncé complet des droits et autorisations.

qacct (1)

Nom

`qacct` – Rapporte et comptabilise l'utilisation de Sun Grid Engine.

Synopsis

```
qacct [ -A Compte ] [ -b HeureDébut ] [ -d Jours ]  
[ -e HeureFin ] [ -g [IdGroupe/NomGroupe] ]  
[ -h [NomHôte] ] [ -help ] [ -history CheminHisto ]  
[ -j [IdTravail/NomTravail] ] [ -l attr=val,... ] [ -nohist ]  
[ -o [Propriétaire] ] [ -pe [NomPE] ] [ -q [NomFile] ]  
[ -slots [NumEmplacement] ] [ -t liste_plage_id_tâches ]  
[ -P [Projet] ] [ -D [Service] ] [ -f NomFichierComp ]
```

Description

L'utilitaire `qacct` analyse le fichier des données de comptabilité (voir `accounting(5)`) et génère un récapitulatif des informations temps chronométré, temps UC et temps système, pour les catégories nom d'hôte, nom de file, nom de groupe, nom de propriétaire, nom de travail et ID de travail, et pour les files d'attente qui satisfont les exigences en matière de ressources spécifiées à l'aide du modificateur `-l`. Des combinaisons de différentes catégories sont autorisées. Il est également possible de lister tous les travaux ou uniquement ceux sélectionnés à l'aide du modificateur `-j`. Par exemple, les critères de recherche peuvent être définis de façon à inclure dans le récapitulatif les informations relatives à une file d'attente et un propriétaire, mais pas à deux files d'attente dans une même requête.

Options

Le TABLEAU 2 liste les options de `qacct`.

TABLEAU 2 Options de `qacct`

Option	Description
-A <i>Compte</i>	Compte des travaux à inclure dans le récapitulatif.
-b <i>HeureDébut</i>	Heure de début minimale des travaux à inclure dans le récapitulatif, au format <code>[[SS]AA]MMJJhmm[.SS]</code> . Voir aussi l'option -d.
-d <i>Jours</i>	Nombre de jours à inclure dans le récapitulatif afin d'imprimer les informations de comptabilité. Si cette option est utilisée avec l'option -b <i>HeureDébut</i> (voir ci-dessus), les travaux commencés entre <i>HeureDébut</i> et <i>HeureDébut</i> + <i>Jours</i> sont comptabilisés. Si elle est utilisée avec l'option -e <i>HeureFin</i> (voir ci-dessous), le décompte commence à <i>HeureFin</i> - <i>Jours</i> .
-e <i>HeureFin</i>	Heure de début maximale des travaux à inclure dans le récapitulatif, au format <code>[[SS]AA]MMJJhmm[.SS]</code> . Voir aussi l'option -d.
[-f <i>NomFichierComp</i>]	Fichier de comptabilité à utiliser. Si cette option est omise, le fichier de comptabilité par défaut du système est traité.
-g [<i>IdGroupe</i> / <i>NomGroupe</i>]	ID numérique de groupe système ou nom de groupe alphanumérique des propriétaires de travaux à inclure dans la comptabilité. Si <i>IdGroupe</i> / <i>NomGroupe</i> est omis, tous les groupes sont pris en compte.
-h [<i>NomHôte</i>]	Nom, sans distinction entre majuscules et minuscules, de l'hôte sur lequel les informations de comptabilité sont demandées. Si ce nom est omis, les totaux relatifs aux différents hôtes s'affichent séparément.
-help	Affiche des informations d'aide pour la commande <code>qacct</code> .
-history <i>CheminHisto</i>	Chemin du répertoire où se trouvent les données d'historique de configuration des complexes et des files d'attente, qui est utilisé pour la mise en correspondances des exigences en ressources en conjonction avec le modificateur -l. Si ce dernier n'est pas défini, cette option est ignorée.
-j [[<i>NomTravail</i> / <i>IdTravail</i>]]	Nom ou ID du travail pendant l'exécution pour lequel les informations de comptabilité sont imprimées. Si vous n'entrez ni nom ni ID, tous les travaux sont répertoriés. Cette option change le format de la sortie de <code>qacct</code> . Si elle est activée, les temps UC ne sont plus cumulés et, à la place, les informations de comptabilité « brutes » sont imprimées sous une forme formatée. Voir <code>accounting(5)</code> pour l'explication des informations affichées.

TABEAU 2 Options de `qacct` (*suite*)

Option	Description
<code>-l attr=val,...</code>	<p>Spécifications des exigences en ressources auxquelles les files d'attente dans lesquelles les travaux comptabilisés s'exécutent doivent être conformes. La correspondance est établie à l'aide des données d'historique et reflète donc la situation de la configuration des files d'attente et des complexes au moment du début du travail.</p> <p>La requête de ressources est très similaire à celle décrite dans <code>qsub(1)</code>. La principale différence est qu'il n'est pas possible de demander des informations de charge en constante évolution car elles ne sont pas contenues dans les données de configuration d'historique utilisées.</p>
<code>-nohist</code>	<p>N'est utile qu'avec l'option <code>-l</code>. Force <code>qacct</code> à ne pas utiliser les données d'historique de configuration de files d'attente et de complexes pour la mise en correspondance des ressources requises ; extrait à la place la configuration courante des files d'attente et des complexes de <code>sgc_qmaster(8)</code>.</p> <p>Vous remarquerez que cela peut déboucher sur des statistiques prêtant à confusion, car la configuration courante des files et des complexes peut être nettement différente de la situation valide pour les travaux passés. Vous remarquerez également que tous les hôtes référencés dans le fichier de comptabilité doivent être activés et en fonctionnement pour que les résultats soient obtenus.</p>
<code>-o [Propriétaire]</code>	<p>Nom du propriétaire des travaux pour lequel les statistiques de comptabilité sont rassemblées. Si l'argument optionnel <i>Propriétaire</i> est omis, la liste des statistiques de comptabilité de tous les propriétaires de travaux présents dans le fichier de comptabilité est générée.</p>
<code>-pe [NomPE]</code>	<p>Nom de l'environnement parallèle pour lequel l'utilisation est résumée. Si <i>NomPE</i> n'est pas précisé, les données de comptabilité sont listées séparément pour chaque environnement parallèle.</p>
<code>-q [NomFile]</code>	<p>Nom de la file d'attente pour laquelle l'utilisation est résumée. Si <i>NomFile</i> n'est pas précisé, les données de comptabilité sont listées séparément pour chaque file d'attente.</p>
<code>-slots [NumEmplacement]</code>	<p>Numéros des emplacements de file d'attente pour lesquels l'utilisation est résumée. Si <i>NumEmplacement</i> n'est pas précisé, les données de comptabilité sont listées séparément pour chaque numéro d'emplacement de file.</p>

TABLEAU 2 Options de `qacct` (suite)

Option	Description
<code>-t liste_plage_id_tâches</code>	Disponible uniquement avec l'option <code>-j</code> décrite ci-dessus. Le modificateur <code>-t</code> spécifie la plage des tâches de travail de matrice, pour lesquelles les informations de comptabilité doivent être imprimées. La syntaxe et la sémantique de <code>liste_plage_id_tâches</code> sont identiques à celles décrites sous l'option <code>-t</code> de <code>qsub(1)</code> . Reportez-vous aussi à cette section pour davantage d'informations sur les matrices de travaux.
<code>-P [Projet]</code>	Nom du projet pour lequel l'utilisation est résumée. Si <code>Projet</code> n'est pas indiqué, les données de comptabilité sont listées séparément pour chaque projet du propriétaire. Les projets ne sont utilisés qu'en mode Sun Grid Engine, Enterprise Edition.
<code>-D [Service]</code>	Nom du service pour lequel l'utilisation est résumée. Si <code>Service</code> n'est pas indiqué, les données de comptabilité sont listées séparément pour chaque service du propriétaire. Les services sont uniquement utilisés en mode Sun Grid Engine, Enterprise Edition.

Variables d'environnement

Le TABLEAU 3 décrit les variables d'environnement associées à `qacct`.

TABLEAU 3 Variables d'environnement de `qacct`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qacct</code> utilise (dans cet ordre) : <ul style="list-style-type: none"> Le nom de la cellule spécifiée dans la variable d'environnement, <code>SGE_CELL</code>, si elle est définie. Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sgc_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sgc_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qacct</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

- Fichier de comptabilité par défaut de Sun Grid Engine –
`<racine_sge>/<cellule>/common/accounting`
- Base de données d'historique par défaut de Sun Grid Engine –
`<racine_sge>/<cellule>/common/history`
- Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `qsub(1)`, `accounting(5)`, `sge_qmaster(8)`, `sge_commd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qconf (1)

Nom

`qconf` – Configuration des files d'attente de Sun Grid Engine

Description

`qconf` permet à l'administrateur système d'apporter des ajouts, des suppressions et des modifications à la configuration courante de Sun Grid Engine, au niveau de la gestion des files d'attente, des hôtes, des complexes et des utilisateurs. `qconf` vous permet également d'examiner la configuration courante des files d'attente existantes.

Options

Sauf spécification autre, les options répertoriées dans le TABLEAU 4 et les opérations correspondantes peuvent être employées par tous les utilisateurs qui ont un compte valide.

TABLEAU 4 Options de `qconf`

Option	Description
<code>-Attr spéc_obj nomf instance_obj...</code>	<p><i>Ajoute des attributs à l'objet</i> – Similaire à <code>-attr</code> (voir ci-dessous) mais tire les spécifications relatives aux attributs d'objet à améliorer du fichier nommé <i>nomf</i>. Contrairement à ce qui se passe avec <code>-attr</code>, plusieurs attributs peuvent être améliorés. Ils doivent être spécifiés sous la forme d'une liste dans <i>nomf</i> en respectant le format de fichier de l'objet correspondant (voir <code>queue_conf(5)</code> pour la file d'attente, par exemple). Cette option exige des privilèges de super-utilisateur/gestionnaire.</p>
<code>-Ac nom_complexe nomf</code>	<p><i>Ajoute un complexe</i> – Ajoute le complexe <i>nom_complexe</i> défini dans <i>nomf</i> à la grappe Sun Grid Engine. Le format utilisé pour la spécification d'un complexe est décrit dans <code>complex(5)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<code>-Acal nomf</code>	<p><i>Ajoute un agenda</i> – Ajoute une nouvelle définition d'agenda à l'environnement Sun Grid Engine. Les agendas sont utilisés dans Sun Grid Engine pour programmer la disponibilité et la non-disponibilité des files d'attente. Le format d'une définition d'agenda est décrit dans <code>calendar_conf(5)</code>. La définition de l'agenda est tirée du fichier <i>nomf</i>. Privilèges de super-utilisateur/gestionnaire requis.</p>
<code>-Ackpt nomf</code>	<p><i>Ajoute un environnement de point de reprise</i> – Ajoute l'environnement de point de reprise tel que défini dans <i>nomf</i> (voir <code>checkpoint(5)</code>) à la liste des environnements de point de reprise pris en charge. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-Aconf <i>liste_fichiers</i>	<i>Ajoute des configurations</i> – Ajoute les configurations de grappe (voir <code>sge_conf(5)</code>) spécifiées dans les fichiers de la liste séparée par des virgules <i>liste_fichiers</i> . Privilèges de super-utilisateur ou de gestionnaire requis.
-Ae <i>nomf</i>	<i>Ajoute un hôte d'exécution</i> – Ajoute l'hôte d'exécution défini dans <i>nomf</i> à la grappe Sun Grid Engine. Le format utilisé pour la spécification d'un hôte d'exécution est décrit dans <code>host_conf(5)</code> . Privilèges de super-utilisateur ou de gestionnaire requis.
-Ap <i>nomf</i>	<i>Ajoute une configuration PE</i> – Ajoute l'environnement parallèle (PE) défini dans <i>nomf</i> à la grappe Sun Grid Engine. Privilèges de super-utilisateur ou de gestionnaire requis.
-Aprj <i>nomf</i>	<i>Ajoute un nouveau projet</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle ajoute la description de projet définie dans <i>nomf</i> à la liste des projets enregistrés (voir <code>project(5)</code>). Privilèges de super-utilisateur ou de gestionnaire requis.
-Aq <i>nomf</i>	<i>Ajoute une nouvelle file</i> – Ajoute la file d'attente définie dans <i>nomf</i> à la grappe Sun Grid Engine. Privilèges de super-utilisateur ou de gestionnaire requis.
-Au <i>nomf</i>	<i>Ajoute une ACL</i> – Ajoute une liste de contrôle d'accès (ACL) à Sun Grid Engine. Les ACL sont utilisées dans le cadre de l'authentification pour l'utilisation des files d'attente. Privilèges de super-utilisateur/gestionnaire/opérateur requis.

TABLEAU 4 Options de `qconf` (suite)

Option	Description
-Dattr <i>spéc_obj nomf instance_obj...</i>	<p><i>Supprime des attributs d'objet</i> – Similaire à <code>-dattr</code> (voir ci-dessous) mais la définition des attributs de liste desquels les entrées vont être supprimées est contenue dans le fichier <i>nomf</i>. Contrairement à ce qui se passe avec <code>-dattr</code>, plusieurs attributs peuvent être modifiés. Ils doivent être spécifiés sous la forme d'une liste dans <i>nomf</i> en suivant le format de fichier de l'objet correspondant (voir <code>queue_conf(5)</code> pour la file d'attente, par exemple).</p> <p>Privilèges de super-utilisateur/gestionnaire requis.</p>
-Matrr <i>spéc_obj nomf instance_obj...</i>	<p><i>Modifie des attributs d'objet</i> – Similaire à <code>-matrr</code> (voir ci-dessous) mais tire les spécifications relatives aux attributs d'objet à modifier du fichier nommé <i>nomf</i>. Contrairement à ce qui se passe avec <code>-matrr</code>, plusieurs attributs peuvent être modifiés. Ils doivent être spécifiés sous la forme d'une liste dans <i>nomf</i> en suivant le format de fichier de l'objet correspondant (voir <code>queue_conf(5)</code> pour la file d'attente, par exemple).</p> <p>Privilèges de super-utilisateur/gestionnaire requis.</p>
-Mc <i>nom_complexe nomf</i>	<p><i>Modifie un complexe</i> – Surécrit le complexe spécifié avec le contenu de <i>nomf</i>. Le fichier d'argument doit respecter le format spécifié dans <code>complex(5)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-Mcal <i>nomf</i>	<p><i>Modifie un agenda</i> – Surécrit la définition d'agenda comme spécifié dans <i>nomf</i>. Le fichier d'argument doit respecter le format spécifié dans <code>calendar_conf(5)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-Mckpt <i>nomf</i>	<p><i>Modifie un environnement de point de reprise</i> – Surécrit un environnement de point de reprise existant avec les définitions contenues dans <i>nomf</i> (voir <code>checkpoint(5)</code>). L'attribut de nom dans <i>nomf</i> doit correspondre à un environnement de point de reprise existant. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-Me <i>nomf</i>	<i>Modifie un hôte d'exécution</i> – Surécrit la configuration d'hôte d'exécution de l'hôte spécifié avec le contenu de <i>nomf</i> , qui doit suivre le format défini dans <code>host_conf(5)</code> . Privilèges de super-utilisateur ou de gestionnaire requis.
-Mp <i>nomf</i>	<i>Modifie une configuration PE</i> – Identique à -mp (voir ci-dessous) mais au lieu d'appeler un éditeur pour modifier la configuration PE, le fichier <i>nomf</i> est considéré comme contenant une configuration modifiée. Reportez-vous à <code>sgc_pe(5)</code> pour tout détail sur le format de configuration PE. Privilèges de super-utilisateur ou de gestionnaire requis.
-Mprj <i>nomf</i>	<i>Modifie une configuration de projet</i> – Identique à -mprj (voir ci-dessous) mais au lieu d'appeler un éditeur pour modifier la configuration du projet, le fichier <i>nomf</i> est considéré comme contenant une configuration modifiée. Reportez-vous à <code>project(5)</code> pour tout détail sur le format de configuration des projets. Privilèges de super-utilisateur ou de gestionnaire requis.
-Mq <i>nomf</i>	<i>Modifie la configuration d'une file</i> – Identique à -mq (voir ci-dessous) mais au lieu d'appeler un éditeur pour modifier la configuration de file, le fichier <i>nomf</i> est considéré comme contenant une configuration modifiée. Reportez-vous à <code>queue_conf(5)</code> pour tout détail sur le format de configuration des files. Privilèges de super-utilisateur ou de gestionnaire requis.

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<p><code>-Mqattr nomf nom_file,...</code> Cette commande est obsolète. Utilisez <code>-Mattrib</code> à la place.</p>	<p><i>Modifie des attributs de file d'attente</i> – Permet de changer des attributs de configuration de file d'attente sélectionnés dans plusieurs files au moyen d'une unique commande. Les définitions d'attributs de file contenues dans <i>nomf</i> seront appliquées dans toutes les files d'attente contenues dans la liste séparée par des virgules de noms de files. Les attributs de file qui ne figurent pas dans <i>nomf</i> resteront inchangés. Tous les attributs de file d'attente peuvent être modifiés à l'exception de <i>nom_file</i> et <i>nomhôtef</i>. Reportez-vous à <code>queue_conf(5)</code> pour tout détail sur le format de configuration des files. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-Mu nomf</code></p>	<p><i>Modifie une ACL</i> – Prend la liste de contrôle d'accès (ACL) définie dans <i>nomf</i> pour surécrire toute ACL existante du même nom. Voir <code>access_list(5)</code> pour toute information sur le format de configuration des ACL. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-Rattrib spéc_obj nomf instance_obj,...</code></p>	<p><i>Remplace des attributs d'objets</i> – Similaire à <code>-rattrib</code> (voir ci-dessous) mais la définition des attributs de liste dont le contenu va être remplacé figure dans le fichier nommé <i>nomf</i>. Contrairement à ce qui se passe avec <code>-rattrib</code>, plusieurs attributs peuvent être modifiés. Ils doivent être spécifiés sous la forme d'une liste dans <i>nomf</i> en suivant le format de fichier de l'objet correspondant (voir <code>queue_conf(5)</code> pour la file d'attente, par exemple). Privilèges de super-utilisateur/gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-aattr <i>spéc_obj nom_attr val instance_obj...</i>	<p><i>Ajoute des attributs d'objet</i> – Permet d'ajouter des spécifications à un unique attribut de liste de configuration dans plusieurs instances d'un objet au moyen d'une unique commande. Les objets actuellement pris en charge sont les configurations de file d'attente et d'hôte spécifiées par <i>queue</i> ou <i>host</i> dans <i>spéc_obj</i>. Le paramètre de file <i>load_thresholds</i> est un exemple d'attribut de liste. L'option -aattr permet d'ajouter des entrées à de telles listes tandis que -dattr permet d'en supprimer, -mattr d'en modifier et -rattr d'en remplacer. Le nom de l'attribut de configuration à enrichir est spécifié par <i>nom_attr</i> suivi de <i>val</i> sous la forme d'une paire <i>nom=valeur</i>. La liste séparée par des virgules des instances d'objet (par ex. la liste des files d'attente) auxquelles les changements doivent être appliqués est spécifiée à la fin de la commande.</p>
-ac <i>nom_complexe</i>	<p>La restriction suivante s'applique : pour l'objet <i>host</i>, l'attribut <i>load_values</i> ne peut pas être modifié (voir <code>host_conf(5)</code>). Privilèges de super-utilisateur ou de gestionnaire requis.</p> <p><i>Ajoute un complexe</i> – Ajoute un complexe à l'environnement Sun Grid Engine. Les entrées de complexe contiennent une ou plusieurs ressources susceptibles d'être demandées par les travaux soumis au système. La page de manuel <code>complex(5)</code> contient des informations détaillées sur le format d'une définition de complexe.</p> <p>Lorsque vous utilisez l'option -ac, le nom du complexe est donné dans l'option de commande. <code>qconf</code> ouvre alors un fichier provisoire et lance l'éditeur de texte indiqué par la variable d'environnement <code>EDITOR</code> (l'éditeur par défaut est <code>vi(1)</code> si <code>EDITOR</code> n'est pas définie). Après l'entrée de la définition du complexe et la fermeture de l'éditeur, le nouveau complexe est contrôlé et enregistré auprès de <code>sge_qmaster(8)</code>. Privilèges de super-utilisateur/gestionnaire requis.</p>

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<p><code>-acal nom_agenda</code></p>	<p><i>Ajoute un agenda</i> – Ajoute une nouvelle définition d’agenda à l’environnement Sun Grid Engine. Les agendas sont utilisés dans Sun Grid Engine pour programmer la disponibilité et la non-disponibilité des files d’attente. Le format d’une définition d’agenda est décrit dans <code>calendar_conf(5)</code>.</p> <p>Avec le nom d’agenda donné dans l’argument d’option, <code>qconf</code> ouvre un fichier provisoire et lance l’éditeur de texte indiqué par la variable d’environnement <code>EDITOR</code> (l’éditeur par défaut est <code>vi(1)</code> si <code>EDITOR</code> n’est pas définie). Après l’entrée de la définition d’agenda et la fermeture de l’éditeur, le nouvel agenda est contrôlé et enregistré auprès de <code>sge_qmaster(8)</code>. Privilèges de super-utilisateur/gestionnaire requis.</p>
<p><code>-ackpt nom_ptreprise</code></p>	<p><i>Ajoute un environnement de point de reprise</i> – Ajoute un environnement de point de reprise de nom <code>nom_ptreprise</code> à la liste des environnements de point de reprise tenue par Sun Grid Engine et utilisable pour la soumission de travaux de points de reprise (voir <code>checkpoint(5)</code> pour tout détail sur le format d’une définition d’un environnement de point de reprise). <code>qconf</code> récupère une configuration d’environnement de point de reprise par défaut et exécute <code>vi(1)</code> (ou <code>\$EDITOR</code> si la variable d’environnement <code>EDITOR</code> est définie) pour vous permettre de la personnaliser. À la fermeture de l’éditeur, l’environnement de point de reprise est enregistré auprès de <code>sge_qmaster(8)</code>. Privilèges de super-utilisateur/gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-aconf <i>hôte</i> ,...	<p><i>Ajoute une configuration</i> – Ajoute ultérieurement des configurations de grappe (voir <code>sge_conf(5)</code>) pour les hôtes qui figurent dans la liste séparée par des virgules <code>liste_fichiers</code>. Pour chaque hôte, un éditeur (celui indiqué par <code>\$EDITOR</code> ou <code>vi(1)</code>) est appelé et la configuration relative à l'hôte peut être entrée. La configuration est enregistrée auprès de <code>sge_qmaster(8)</code> après la sauvegarde du fichier et la fermeture de l'éditeur. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-ae [<i>modèle_hôte</i>]	<p><i>Ajoute un hôte d'exécution</i> – Ajoute un hôte à la liste des hôtes d'exécution de Sun Grid Engine. Si une file d'attente est configurée sur un hôte, cet hôte est automatiquement ajouté à la liste des hôtes d'exécution de Sun Grid Engine. Ajouter des hôtes d'exécution offre de manière explicite l'avantage de pouvoir spécifier des paramètres tels que les valeurs de mise à l'échelle de la charge avec l'enregistrement de l'hôte d'exécution. Ces paramètres peuvent toutefois être modifiés (de leurs valeurs par défaut) à tout moment par la suite via l'option <code>-me</code> décrite plus loin.</p> <p>Si l'argument <code>modèle_hôte</code> est présent, <code>qconf</code> récupère la configuration de l'hôte d'exécution spécifié de <code>sge_qmaster(8)</code>, sinon un modèle générique est utilisé. Le modèle est ensuite stocké dans un fichier et <code>qconf</code> exécute <code>vi(1)</code> (ou l'éditeur indiqué par <code>\$EDITOR</code> si la variable d'environnement <code>EDITOR</code> est définie) pour en changer des entrées. Le format d'une spécification d'hôte d'exécution est décrit dans <code>host_conf(5)</code>. Lorsque les changements sont sauvegardés dans l'éditeur et que l'éditeur est fermé, le nouvel hôte d'exécution est enregistré auprès de <code>sge_qmaster(8)</code>. Cette option exige des privilèges de super-utilisateur ou de gestionnaire.</p>

TABLEAU 4 Options de `qconf` (suite)

Option	Description
-ah <i>nomhôte</i> ,...	<i>Ajoute un hôte administratif</i> – Ajoute les hôtes <i>nomhôte</i> à la liste des hôtes approuvés de Sun Grid Engine (un hôte doit figurer dans cette liste pour exécuter des commandes administratives de Sun Grid Engine, la seule exception à cette règle étant l'exécution de <code>qconf</code> sur le noeud <code>sge_qmaster(8)</code>). Les procédures d'installation par défaut de Sun Grid Engine ajoutent en général automatiquement tous les hôtes d'exécution désignés (voir l'option <code>-ae</code> plus haut) à la liste des hôtes approuvés de Sun Grid Engine. Privilèges de super-utilisateur ou de gestionnaire requis.
-am <i>utilisateur</i> ,...	<i>Ajoute des gestionnaires</i> – Ajoute les utilisateurs indiqués à la liste des gestionnaires de Sun Grid Engine. Privilèges de super-utilisateur ou de gestionnaire requis.
-ao <i>utilisateur</i> ,...	<i>Ajoute un opérateur</i> – Ajoute les utilisateurs indiqués à la liste des opérateurs de Sun Grid Engine. Privilèges de super-utilisateur/gestionnaire requis.
-ap <i>nom_pe</i>	<i>Ajoute un nouveau PE</i> – Ajoute une description d' <i>environnement parallèle</i> (PE) sous le nom <i>nom_pe</i> à la liste des PE tenue par Sun Grid Engine et utilisable pour soumettre des travaux parallèles (voir <code>sge_pe(5)</code> pour tout détail sur le format d'une définition de PE). <code>qconf</code> récupère une configuration de PE par défaut et exécute <code>vi(1)</code> (ou <code>\$EDITOR</code> si la variable d'environnement <code>EDITOR</code> est définie) pour vous permettre de la personnaliser. À la fermeture de l'éditeur, le PE est enregistré auprès de <code>sge_qmaster(8)</code> . Privilèges de super-utilisateur/gestionnaire requis.

TABEAU 4 Options de `qconf` (suite)

Option	Description
-aprj	<p><i>Ajoute un nouveau projet</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle ajoute une description de projet à la liste des projets enregistrés (voir <code>project(5)</code>). <code>qconf</code> récupère une configuration de projet modèle et exécute <code>vi(1)</code> (ou <code>\$EDITOR</code> si la variable d'environnement <code>EDITOR</code> est définie) pour vous permettre de la personnaliser. À la sortie de l'éditeur, le projet est enregistré auprès de <code>sge_qmaster(8)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-aq [<i>modèle_file</i>]	<p><i>Ajoute une nouvelle file d'attente</i> – <code>qconf</code> récupère soit la configuration de file par défaut (voir <code>queue_conf(5)</code>) soit la configuration de la file <i>modèle_file</i> (si l'argument optionnel est présent) et exécute <code>vi(1)</code> (ou <code>\$EDITOR</code> si la variable d'environnement <code>EDITOR</code> est définie) pour vous permettre de la personnaliser. Lorsque vous quittez l'éditeur, la file d'attente est enregistrée auprès de <code>sge_qmaster(8)</code>. Une configuration minimale ne requiert que la définition du nom de la file et celle de son hôte. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-as <i>nomhôte</i> ,...	<p><i>Ajoute des hôtes de soumission</i> – Ajoute les hôtes <i>nomhôte</i> à la liste des hôtes uniquement autorisés à soumettre des travaux Sun Grid Engine et à en contrôler le comportement. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<p><code>-astnode chemin_noeud=partages,...</code></p>	<p><i>Ajoute un noeud d'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle ajoute le ou les noeuds d'arborescence de partages spécifiés à l'arborescence des partages (voir <code>share_tree(5)</code>). Le <i>chemin_noeud</i> est un chemin hiérarchique (<code>[/]nom_noeud [/ .]nom_noeud...</code>) qui spécifie l'emplacement du nouveau noeud dans l'arborescence des partages. Le nom de base de <i>chemin_noeud</i> est le nom du nouveau noeud. Le noeud est initialisé au nombre de partages spécifié. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-astree</code></p>	<p><i>Ajoute une arborescence de partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle ajoute la définition d'une arborescence de partages au système (voir <code>share_tree(5)</code>). Une arborescence de partages modèle est récupérée et un éditeur (<code>vi(1)</code> ou celui indiqué par <code>\$EDITOR</code>) est appelé pour la modifier. À la fermeture de l'éditeur, les données modifiées sont enregistrées auprès de <code>sge_qmaster(8)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-au utilisateur,... nom_acl,...</code></p>	<p><i>Ajoute des utilisateurs aux ACL</i> – Ajoute des utilisateurs aux ACL Sun Grid Engine. Les listes d'utilisateurs sont utilisées pour l'authentification dans le cadre de l'utilisation des files d'attente. Privilèges de super-utilisateur/gestionnaire/opérateur requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-Auser <i>nomf</i>	<p><i>Ajoute un utilisateur</i> – Cette option n'est prise en charge que pour Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle ajoute l'utilisateur défini dans <i>nomf</i> à la grappe Sun Grid Engine, Enterprise Edition. Le format d'une spécification d'utilisateur est décrit dans <code>user(5)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-auser	<p><i>Ajoute un utilisateur</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle ajoute un utilisateur à la liste des utilisateurs enregistrés (voir <code>user(5)</code>). Cette commande appelle un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d'environnement <code>EDITOR</code>) pour un modèle d'utilisateur. Le nouvel utilisateur est enregistré après le changement de l'entrée et la fermeture de l'éditeur. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-clearusage	<p><i>Efface l'utilisation de l'arborescence des partages</i> – (Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.)</p> <p>Elle efface toutes les informations sur l'utilisation relatives aux utilisateurs et aux projets de l'arborescence des partages. Toute les informations d'utilisation seront remises zéro.</p>
-cq <i>nom_file,...</i>	<p><i>Nettoie la file d'attente</i> – Efface de la file d'attente les travaux qui n'ont pas été récoltés. Il s'agit principalement d'un outil de développement. Privilèges de super-utilisateur/gestionnaire/opérateur requis.</p>

TABLEAU 4 Options de `qconf` (suite)

Option	Description
-dattr <i>spéc_obj nom_attr val instance_obj...</i>	<p><i>Supprime des attributs d'objet</i> – Permet de supprimer des spécifications dans un unique attribut de liste de configurations dans plusieurs instances d'un objet avec une unique commande. Les objets actuellement pris en charge sont les configurations de file et d'hôte spécifiées par <i>queue</i> ou <i>host</i> dans <i>spéc_obj</i>. Le paramètre de file <i>load_thresholds</i> est un exemple d'attribut de liste. L'option -dattr permet de supprimer des entrées de telles listes, tandis que -aattr permet d'en ajouter, -mattr d'en modifier et -rattr d'en remplacer.</p> <p>Le nom de l'attribut de configuration à modifier est spécifié par <i>nom_attr</i> suivi de <i>val</i>, qui définit le nom de l'entrée de liste d'attributs à supprimer. La liste séparée par des virgules des instances d'objet (par ex. la liste des files d'attente) auxquelles les changements doivent être appliqués est spécifiée à la fin de la commande.</p> <p>La restriction suivante s'applique : pour l'objet <i>host</i>, l'attribut <i>load_values</i> ne peut pas être modifié (voir <code>host_conf(5)</code>). Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-dc <i>nom_complexe...</i>	<p><i>Supprime un complexe</i> – Supprime des complexes de Sun Grid Engine. Cette option exige des privilèges de super-utilisateur ou de gestionnaire.</p>
-dcal <i>nom_agenda...</i>	<p><i>Supprime un agenda</i> – Supprime la définition d'agenda spécifiée de Sun Grid Engine. Privilèges de super-utilisateur/gestionnaire requis.</p>
-dckpt <i>nom_ptreprise</i>	<p><i>Supprime un environnement de point de reprise</i> – Supprime l'environnement de point de reprise spécifié. Privilèges de super-utilisateur/gestionnaire requis.</p>
-dconf <i>hôte...</i>	<p><i>Supprime une configuration</i> – L'entrée de configuration correspondant à l'hôte spécifié est supprimée de la liste des configurations. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
<code>-de nom_hôte,...</code>	<i>Supprime un hôte d'exécution</i> – Supprime des hôtes de la liste des hôtes d'exécution de Sun Grid Engine. Privilèges de super-utilisateur/gestionnaire requis.
<code>-dh nom_hôte,...</code>	<i>Supprime un hôte administratif</i> – Supprime des hôtes de la liste des hôtes approuvés de Sun Grid Engine. L'hôte sur lequel <code>sge_qmaster(8)</code> est en cours d'exécution ne peut pas être supprimé de la liste des hôtes administratifs. Privilèges de super-utilisateur/gestionnaire requis.
<code>-dm utilisateur[,utilisateur,...]</code>	<i>Supprime des gestionnaires</i> – Supprime des gestionnaires de la liste des gestionnaires. Privilèges de super-utilisateur/gestionnaire requis.
<code>-do utilisateur[,utilisateur,...]</code>	<i>Supprime des opérateurs</i> – Supprime des opérateurs de la liste des opérateurs. Privilèges de super-utilisateur/gestionnaire requis.
<code>-dp nom_pe</code>	<i>Supprime un environnement parallèle</i> – Supprime l'environnement parallèle (PE) spécifié. Privilèges de super-utilisateur/gestionnaire requis.
<code>-dpr j projet,...</code>	<i>Supprime des projets</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle supprime le ou les projets spécifiés. Privilèges de super-utilisateur/gestionnaire requis.
<code>-dq nom_file,...</code>	<i>Supprime une file d'attente</i> – Supprime la ou les files d'attente spécifiées. Les travaux actifs seront exécutés complètement. Privilèges de super-utilisateur/gestionnaire requis.
<code>-ds nom_hôte,...</code>	<i>Supprime un hôte de soumission</i> – Supprime des hôtes de la liste des hôtes de soumission de Sun Grid Engine. Privilèges de super-utilisateur/gestionnaire requis.

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<code>-dstnode chemin_noeud,...</code>	<p><i>Supprime un noeud d'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle supprime le ou les noeuds spécifiés de l'arborescence des partages. Le <i>chemin_noeud</i> est un chemin hiérarchique (<i>/[nom_noeud[/.[nom_noeud...]</i>) qui spécifie l'emplacement du noeud à supprimer dans l'arborescence des partages. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<code>-dstree</code>	<p><i>Supprime l'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle supprime l'arborescence de partages courante. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<code>-du utilisateur,... nom_acl,...</code>	<p><i>Supprime des utilisateurs des ACL</i> – Supprime un ou plusieurs utilisateurs d'une ou plusieurs ACL de Sun Grid Engine. Privilèges de super-utilisateur/gestionnaire/opérateur requis.</p>
<code>-dul nom_acl,...</code>	<p><i>Supprime des listes d'utilisateurs</i> – Supprime une ou plusieurs listes d'utilisateurs du système. Privilèges de super-utilisateur/gestionnaire/opérateur requis.</p>
<code>-duser utilisateur,...</code>	<p><i>Supprime des utilisateurs</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle supprime le ou les utilisateurs spécifiés de la liste des utilisateurs enregistrés. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<code>-help</code>	<p>Imprime la liste de toutes les options.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-k{m s e[j] {hôte,... all}}	<p><i>Arrête le logiciel Sun Grid Engine</i> – Cette option est utilisée pour arrêter les composants (démons) de Sun Grid Engine. Sous la forme <code>-km sge_qmaster(8)</code> est obligé de s'arrêter d'une manière contrôlée. De la même façon, le modificateur <code>-ks</code> cause l'arrêt de <code>sge_schedd(8)</code>. L'arrêt des processus <code>sge_execd(8)</code> en cours couramment enregistrés est lancé par l'option <code>-ke</code>. Si l'option <code>-kej</code> est spécifiée, tous les travaux qui s'exécutent sur les hôtes d'exécution sont abandonnés avant l'arrêt du <code>sge_execd(8)</code> correspondant. La liste d'hôtes séparée par des virgules spécifie les hôtes d'exécution concernés par les options <code>-ke</code> et <code>-kej</code>. Si le mot clé <code>all</code>, est spécifié à la place d'une liste d'hôtes, tous les processus <code>sge_execd(8)</code> en cours d'exécution sont arrêtés.</p> <p>Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-kec {id,... all}	<p><i>Élimine les clients événement</i> – Cette option est utilisée pour arrêter les clients événement enregistrés sur <code>sge_qmaster(8)</code>. La liste des clients événement séparée par des virgules spécifie les clients événement concernés par l'option <code>-kec</code>. Si le mot clé <code>all</code> est spécifié à la place d'une liste de client événement, tous les clients événement en cours d'exécution à l'exception des clients spéciaux tels que <code>sge_schedd(8)</code> sont arrêtés.</p> <p>Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<p><code>-matrr spéc_obj nom_attr val instance_obj...</code></p>	<p><i>Modifie des attributs d'objet</i> – Permet de changer un attribut de configuration dans plusieurs instances d'un objet avec une unique commande.</p> <p>Les objets actuellement pris en charge sont les configurations de file et d'hôte spécifiées par <i>queue</i> ou <i>host</i> dans <i>spéc_obj</i>. Veuillez remarquer que <code>-matrr file nom_attr val nom_file...</code> est équivalent à <code>-mqattr nom_attr val nom_file...</code> (voir plus loin). Ce dernier élément est disponible pour la compatibilité à rebours.</p> <p>Le nom de l'attribut de configuration à modifier est spécifié par <i>nom_attr</i> suivi de la valeur à laquelle l'attribut va être fixé. Si l'attribut est une liste, telle que les <i>load_thresholds</i> de file, <i>val</i> peut être une paire <i>nom=valeur</i>, cas dans lequel seule une entrée correspondante est changée dans la liste. Reportez-vous aux options <code>-aattr</code>, <code>-datrr</code> et <code>-ratrr</code> pour la description des moyens supplémentaires permettant de changer de façon spécifique de tels attributs de liste.</p> <p>La liste séparée par des virgules des instances d'objet (par ex. la liste des files d'attente) auxquelles les changements doivent être appliqués est spécifiée à la fin de la commande.</p> <p>Les restrictions suivantes s'appliquent : pour l'objet file, les attributs <i>qname</i> et <i>qhostname</i> ne peuvent pas être modifiés (voir <code>queue_conf(5)</code>). Pour l'objet hôte, les attributs <i>hostname</i>, <i>load_values</i> et <i>processors</i> ne peuvent pas être modifiés (voir <code>host_conf(5)</code>).</p> <p>Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-mc nom_complexe</code></p>	<p><i>Modifie un complexe</i> – La configuration de complexe spécifiée (voir <code>complex(5)</code>) est récupérée, un éditeur est exécuté (<code>vi(1)</code> ou celui indiqué par <code>\$EDITOR</code>) et la configuration de complexe modifiée est enregistrée auprès de <code>sge_qmaster(8)</code> à la fermeture de l'éditeur. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
<code>-mcal nom_agenda</code>	<i>Modifie un agenda</i> – La définition d’agenda spécifiée (voir <code>calendar_conf(5)</code>) est récupérée, un éditeur est exécuté (<code>vi(1)</code> ou celui indiqué par <code>\$EDITOR</code>) et la définition d’agenda modifiée est enregistrée auprès de <code>sge_qmaster(8)</code> à la fermeture de l’éditeur. Privilèges de super-utilisateur ou de gestionnaire requis.
<code>-mckpt nom_ptreprise</code>	<i>Modifie un environnement de point de reprise</i> – Récupère la configuration courante pour l’environnement de point de reprise spécifié, exécute un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d’environnement <code>\$EDITOR</code>) et enregistre la nouvelle configuration auprès de <code>sge_qmaster(8)</code> . Reportez-vous à <code>checkpoint(5)</code> pour tout détail sur le format de configuration d’un environnement de point de reprise. Privilèges de super-utilisateur ou de gestionnaire requis.
<code>-mconf [hôte,... global]</code>	<i>Modifie une configuration</i> – La configuration de l’hôte spécifié est récupérée, un éditeur est exécuté (<code>vi(1)</code> ou celui indiqué par <code>\$EDITOR</code>) et la configuration modifiée est enregistrée auprès de <code>sge_qmaster(8)</code> à la fermeture de l’éditeur. Si l’argument facultatif <code>hôte</code> est omis ou si le nom d’hôte spécial <code>global</code> est spécifié, la configuration globale des cellules est modifiée. Le format d’une configuration d’hôte est décrit dans <code>sge_conf(5)</code> . Privilèges de super-utilisateur ou de gestionnaire requis.
<code>-me nomhôte</code>	<i>Modifie un hôte d’exécution</i> – Récupère la configuration courante pour l’hôte d’exécution spécifié, exécute un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d’environnement <code>\$EDITOR</code>) et enregistre la configuration modifiée auprès de <code>sge_qmaster(8)</code> à la fermeture de l’éditeur. Le format de configuration d’un hôte d’exécution est décrit dans <code>host_conf(5)</code> . Privilèges de super-utilisateur ou de gestionnaire requis.

TABEAU 4 Options de `qconf` (suite)

Option	Description
<p><code>-mp nom_pe</code></p>	<p><i>Modifie une configuration PE</i> – Récupère la configuration courante pour l'<i>environnement parallèle</i> (PE) spécifié, exécute un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d'environnement <code>EDITOR</code>) et enregistre la nouvelle configuration auprès de <code>sgc_qmaster(8)</code>. Reportez-vous à <code>sgc_pe(5)</code> pour tout détail sur le format d'une configuration de PE. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-mpr j projet</code></p>	<p><i>Modifie un projet</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Les données relatives au projet spécifié sont extraites (voir <code>project(5)</code>) et un éditeur (<code>vi(1)</code> (<code>\$EDITOR</code>)) est appelé pour modifier la définition du projet. À la fermeture de l'éditeur, les données modifiées sont enregistrées. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-mq nomfile</code></p>	<p><i>Modifie une configuration de file d'attente</i> – Récupère la configuration courante pour la file d'attente spécifiée, exécute un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d'environnement <code>EDITOR</code>) et enregistre la nouvelle configuration auprès de <code>sgc_qmaster(8)</code>. Reportez-vous à <code>queue_conf(5)</code> pour tout détail sur le format de configuration des files. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<p><code>-mqattr nom_attr val nom_file,...</code></p> <p>Cette option est obsolète. Utilisez <code>-mattr</code> à la place.</p>	<p><i>Modifie des attributs de file d'attente</i> – Permet de changer un attribut de configuration de file dans plusieurs files d'attente avec une unique commande. Dans toutes les files d'attente contenues dans la liste de noms de files séparée par des virgules la valeur de l'attribut <code>nom_attr</code> sera surécrite par <code>val</code>.</p> <p>Tous les attributs de file d'attente peuvent être modifiés à l'exception de <code>qname</code> et <code>ghostname</code>. Reportez-vous à <code>queue_conf(5)</code> pour tout détail sur le format de configuration des files. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
-msconf	<p><i>Modifie la configuration du programmeur</i> – La configuration courante du programmeur (voir <code>sched_conf(5)</code>) est récupérée, un éditeur est exécuté (<code>vi(1)</code> ou celui indiqué par <code>\$EDITOR</code>) et la configuration modifiée est enregistrée auprès de <code>sgc_qmaster(8)</code> à la fermeture de l'éditeur. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-mstnode <i>chemin_noeud=partages,...</i>	<p><i>Modifie un noeud de l'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle modifie le ou les noeuds d'arborescence de partages spécifiés dans l'arborescence des partages (voir <code>share_tree(5)</code>). Le <i>chemin_noeud</i> est un chemin hiérarchique (<code>/[nom_noeud[/[.nom_noeud...]</code>) qui spécifie l'emplacement d'un noeud existant dans l'arborescence des partages. Le noeud est fixé au nombre de <i>partages</i> spécifiés. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-mstree	<p><i>Modifie l'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle modifie la définition de l'arborescence des partages (voir <code>share_tree(5)</code>). L'arborescence des partages présente est récupérée et un éditeur (<code>vi(1)</code> ou celui indiqué par <code>\$EDITOR</code>) est appelé pour en modifier la définition. À la fermeture de l'éditeur, les données modifiées sont enregistrées auprès de <code>sgc_qmaster(8)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
-mu <i>nom_acl</i>	<p><i>Modifie les listes de contrôle d'accès</i> – Récupère la configuration courante pour l'ACL spécifiée, exécute un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d'environnement <code>EDITOR</code>) et enregistre la nouvelle configuration auprès de <code>sgc_qmaster(8)</code>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<p><code>-ratrr spéc_obj nom_attr val instance_obj...</code></p>	<p><i>Remplace des attributs d'objet</i> – Permet de remplacer un attribut de liste de configuration dans plusieurs instances d'un objet avec une unique commande. Les objets actuellement pris en charge sont les configurations de file et d'hôte spécifiées par <i>queue</i> ou <i>host</i> dans <i>spéc_obj</i>. Le paramètre de file <i>load_thresholds</i> est un exemple d'attribut de liste. L'option <code>-ratrr</code> permet de remplacer de telles listes tandis que <code>-aatrr</code> permet d'ajouter des entrées, <code>-datrr</code> d'en supprimer et <code>-matrr</code> d'en modifier.</p> <p>Le nom de l'attribut de configuration à modifier est spécifié par <i>nom_attr</i> suivi de <i>val</i> qui définit le nouveau paramétrage de l'attribut. La liste séparée par des virgules des instances d'objet (par ex. la liste des files d'attente) auxquelles les changements doivent être appliqués est spécifiée à la fin de la commande.</p>
<p><code>-Muser nomf</code></p>	<p>La restriction suivante s'applique : pour l'objet <i>host</i> l'attribut <i>load_values</i> ne peut pas être modifié (voir <i>host_conf(5)</i>).</p> <p>Privilèges de super-utilisateur ou de gestionnaire requis.</p> <p><i>Modifie un utilisateur</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle modifie l'utilisateur défini dans <i>nomf</i> dans la grappe Sun Grid Engine, Enterprise Edition. Le format d'une spécification d'utilisateur est décrit dans <i>user(5)</i>. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
<code>-muser utilisateur</code>	<p><i>Modifie un utilisateur</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Les données relatives à l'utilisateur spécifié sont extraites (voir <code>user(5)</code>) et un éditeur (<code>vi(1)</code> ou celui indiqué par la variable d'environnement EDITOR) est appelé pour modifier la définition de l'utilisateur. À la fermeture de l'éditeur, les données modifiées sont enregistrées. Privilèges de super-utilisateur ou de gestionnaire requis.</p>
<code>-sc nom_complexe,...</code>	<p><i>Affiche des complexes</i> – Affiche la configuration d'un ou plusieurs complexes.</p>
<code>-scal nom_agenda</code>	<p><i>Affiche un agenda</i> – Affiche la configuration de l'agenda spécifié.</p>
<code>-scall</code>	<p><i>Affiche la liste des agendas</i> – Affiche la liste de tous les agendas couramment définis.</p>
<code>-scl</code>	<p><i>Affiche les noms de la liste des complexes</i> – Affiche la liste de tous les complexes couramment configurés.</p>
<code>-sckpt nom_ptreprise</code>	<p><i>Affiche un environnement de point de reprise</i> – Affiche la configuration de l'environnement de point de reprise spécifié.</p>
<code>-sckptl</code>	<p><i>Affiche la liste des environnements de point de reprise</i> – Affiche la liste des noms de tous les environnements de point de reprise couramment configurés.</p>
<code>-sconf [hôte,.../global]</code>	<p><i>Affiche la configuration</i> – Imprime la configuration de grappe en vigueur au niveau global ou sur le ou les hôtes spécifiés. Si l'argument liste d'hôtes séparée par des virgules est omis ou si la chaîne spéciale <code>global</code> est donnée, la configuration de cellule globale s'affiche. Pour tout autre nom d'hôte de la liste, la fusion de la configuration globale et de celle spécifique de l'hôte s'affiche. Le format de configuration des hôtes est décrit dans <code>sg_e_conf(5)</code>.</p>

TABEAU 4 Options de `qconf` (suite)

Option	Description
<code>-sconf1</code>	<i>Affiche la liste des configurations</i> – Affiche la liste des hôtes pour lesquels des configurations sont disponibles. Le nom d'hôte spécial <code>global</code> fait référence à la configuration de cellule globale.
<code>-se nomhôte</code>	<i>Affiche l'hôte d'exécution</i> – Affiche la définition de l'hôte d'exécution spécifié.
<code>-secl</code>	<i>Affiche les clients événement</i> – Affiche la liste des clients événement de Sun Grid Engine.
<code>-sel</code>	<i>Affiche les hôtes d'exécution</i> – Affiche la liste des hôtes d'exécution de Sun Grid Engine.
<code>-sep</code>	<i>Affiche les processeurs sous licence</i> – Affiche la liste du nombre de processeurs sous licence par hôte d'exécution et au total.
<code>-sh</code>	<i>Affiche les hôtes administratifs</i> – Affiche la liste des hôtes administratifs de Sun Grid Engine.
<code>-sm</code>	<i>Affiche les gestionnaires</i> – Affiche la liste des gestionnaires.
<code>-so</code>	<i>Affiche les opérateurs</i> – Affiche la liste des opérateurs.
<code>-sp nom_pe</code>	<i>Affiche la configuration PE</i> – Affiche la définition de l'environnement parallèle (PE) spécifié par l'argument.
<code>-spl</code>	<i>Affiche la liste des PE</i> – Affiche la liste de tous les <i>environnements parallèles</i> (PE) couramment définis.
<code>-sprj projet</code>	<i>Affiche un projet</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle affiche la définition du projet spécifié (voir <code>project(5)</code>).
<code>-sprjl</code>	<i>Affiche la liste des projets</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle affiche la liste de tous les projets couramment définis.

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<code>-sq nom_file[nom_file,...]</code>	<i>Affiche les files d'attente</i> – Affiche une ou plusieurs files d'attente.
<code>-sql</code>	<i>Affiche la liste des files d'attente</i> – Affiche la liste de toutes les files d'attente couramment définies.
<code>-ss</code>	<i>Affiche les hôtes de soumission</i> – Affiche la liste des hôtes de soumission de Sun Grid Engine.
<code>-ssconf</code>	<i>Affiche la configuration du programmeur</i> – Affiche la configuration courante du programmeur dans le format expliqué dans <code>sched_conf(5)</code> .
<code>-sstnode chemin_noeud,...</code>	<i>Affiche un noeud de l'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle affiche le nom et les partages du ou des noeuds de l'arborescence des partages spécifiés (voir <code>share_tree(5)</code>). Le <i>chemin_noeud</i> est un chemin hiérarchique (<code>(/)[nom_noeud[/].[nom_noeud...]</code>) qui spécifie l'emplacement d'un noeud dans l'arborescence des partages.
<code>-sstree</code>	<i>Affiche l'arborescence des partages</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle affiche la définition de l'arborescence des partages (voir <code>share_tree(5)</code>).
<code>-sss</code>	<i>Affiche le statut du programmeur</i> – Affiche actuellement l'hôte sur lequel le programmeur de Sun Grid Engine est actif ou un message d'erreur si aucun programmeur n'est en cours d'exécution.
<code>-su nom_acl</code>	<i>Affiche une ACL</i> – Affiche une liste de contrôle d'accès (ACL) Sun Grid Engine.
<code>-sul</code>	<i>Affiche les listes d'utilisateurs</i> – Affiche la liste des noms de toutes les listes de contrôle d'accès (ACL) de Sun Grid Engine couramment définies.

TABLEAU 4 Options de `qconf` (suite)

Option	Description
<code>-suser utilisateur,...</code>	<p><i>Affiche un utilisateur</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle affiche la définition du ou des utilisateurs spécifiés (voir <code>user(5)</code>).</p>
<code>-suserl</code>	<p><i>Affiche les utilisateurs</i> – Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle affiche la liste de tous les utilisateurs couramment définis.</p>
<code>-tsm</code>	<p><i>Déclenche la surveillance du programmeur</i> – Le programmeur de Sun Grid Engine <code>sge_schedd(8)</code> est forcé par cette option à imprimer des messages de suivi pour son prochain cycle d'exécution programmé dans le fichier <code><racine_sge>/<cellule>/common/schedd_runlog</code>. Les messages indiquent les raisons pour lesquelles certains travaux ou files d'attente ne rentrent pas dans ce cycle. Privilèges de super-utilisateur ou de gestionnaire requis.</p>

Remarque – Les raisons pour lesquelles des exigences de travaux sont invalides compte tenu de la disponibilité des ressources des files d'attente s'affichent en utilisant le format décrit pour l'option `-F` de `qstat(1)` (voir la description du format complet dans la section *Formats de sortie* de la page de manuel `qstat(1)`).

Variables d'environnement

Le TABLEAU 5 décrit les variables d'environnement associées à `qconf`.

TABLEAU 5 Variables d'environnement de `qconf`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qconf</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement, <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qconf</code> réside. Par défaut, l'hôte local est utilisé.

Restrictions

Les modifications apportées à une configuration de file d'attente n'affectent pas une file d'attente active, elles ne sont appliquées qu'au prochain appel de la file d'attente (par ex. au prochain travail).

Fichiers

- Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sg_intro(1)`, `qstat(1)`, `checkpoint(5)`, `complex(5)`, `sg_conf(5)`,
`host_conf(5)`, `sg_pe(5)`, `queue_conf(5)`, `sg_execd(8)`, `sg_qmaster(8)`,
`sg_schedd(8)`

Droits d'auteur

Voir `sg_intro(1)` pour l'énoncé complet des droits et autorisations.

qdel(1)

Nom

`qdel` – Supprime des travaux Sun Grid Engine des files d'attente.

Syntaxe

```
qdel [ -f ] [ -help ] [ -verify ] [ liste_travail/ID_tâches ]
```

```
qdel [ -f ] [ -help ] [ -verify ] -u liste_utilisateurs | -uall
```

Description

`Qdel` fournit à un utilisateur, un opérateur ou un administrateur, le moyen de supprimer un ou plusieurs travaux. `Qdel` supprime les travaux dans l'ordre dans lequel leurs identificateurs de travail sont présentés.

Options

Le TABLEAU 6 liste les options de `qdel`.

TABLEAU 6 Options de `qdel`

Option	Description
<code>-f</code>	<p>Force l'action concernée pour les travaux en cours. Le ou les travaux sont supprimés de la liste des travaux enregistrés auprès de <code>sgc_qmaster(8)</code> même si le <code>sgc_execd(8)</code> qui contrôle ce ou ces travaux ne répond pas à la requête de suppression envoyée par <code>sgc_qmaster(8)</code>.</p> <p>Les utilisateurs qui ne sont ni des gestionnaires ni des opérateurs de Sun Grid Engine peuvent uniquement utiliser l'option <code>-f</code> (pour leurs propres travaux) si l'entrée de configuration de grappe <code>qmaster_params</code> contient l'indicateur <code>ENABLE_FORCED_QDEL</code> (voir <code>sgc_conf(5)</code>). Le comportement diffère toutefois pour les utilisateurs administratifs et non. Les travaux sont immédiatement supprimés de la base de données de Sun Grid Engine dans le cas des administrateurs. Dans les autres cas, une suppression normale est d'abord tentée et ce n'est qu'après l'échec de cette tentative qu'une annulation forcée est exécutée.</p>
<code>-help</code>	Imprime la liste de toutes les options.
<code>-u nomutilisateur,... -uall</code>	<p>Supprime uniquement les travaux qui ont été soumis par les utilisateurs spécifiés dans la liste de noms d'utilisateurs. Les gestionnaires ont la possibilité d'utiliser la commande <code>qdel -uall</code> pour supprimer tous les travaux de tous les utilisateurs.</p> <p>Si vous utilisez le modificateur <code>-u</code> ou <code>-uall</code>, il est possible de spécifier une <i>liste_travail/ID_tâches</i> supplémentaire.</p>
<code>-verify</code>	N'effectue aucune modification, imprime simplement ce qui serait fait si <code>-verify</code> n'était pas présent.
<i>liste_travail/ID_tâches</i>	<p>Cette option se spécifie comme suit :</p> <p><i>id_travail[.plage_tâches][,id_travail[.plage_tâches],...]</i></p> <p>Si présent, <i>plage_tâches</i> limite l'effet de l'opération à la plage de tâches de travail de matrice spécifiée par le suffixe rattaché à l'ID du travail (voir l'option <code>-t</code> de <code>qsub(1)</code> pour plus de détails sur les matrices de travail).</p> <p>Le spécificateur de plage de tâches est de la forme <i>n[-m[:s]]</i>. La plage peut être un simple chiffre, une plage simple de la forme <i>n-m</i> ou une plage avec une valeur de pas.</p> <p>À la place de <i>liste_travail/ID_tâches</i> il est possible d'utiliser le mot clé <code>all</code> pour modifier tous les travaux de l'utilisateur courant.</p>

Variables d'environnement

Le TABLEAU 7 décrit les variables d'environnement associées à `qdel`.

TABLEAU 7 Variables d'environnement de `qdel`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qdel</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qdel</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

- Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `qstat(1)`, `qsub(1)`, `sge_qmaster(8)`, `sge_execd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qhold(1)

Nom

`qhold` – Empêche l'exécution des travaux de Sun Grid Engine en les suspendant.

Syntaxe

```
qhold [ -h {u|o|s},... ] [ -help ] [ liste_travail/ID_tâches ]
```

```
qhold [ -h {u|o|s},... ] [ -help ] -u liste_utilisateurs | -uall
```

Description

`qhold` fournit à un utilisateur/opérateur/gestionnaire, le moyen de placer des *suspensions* sur un ou plusieurs travaux en attente de programmation. Tant qu'une suspension quelconque est attribuée à un travail, ce dernier ne peut pas être programmé.

Les suspensions peuvent être retirées au moyen de la commande `qrls(1)` ou de `qalter(1)`.

Il existe trois types de suspensions différents :

- **Utilisateur** – Les suspensions d'utilisateur peuvent être attribuées et retirées par les gestionnaires, les opérateurs et le propriétaire des travaux.
- **Opérateur** – Les suspensions d'opérateur peuvent être attribuées et retirées par les gestionnaires et les opérateurs.
- **Système** – Les suspensions de système ne peuvent être attribuées et retirées que par les gestionnaires.

Si aucun type de suspension n'est spécifié avec l'option `-h` (voir ci-dessous), le système assume par défaut qu'il s'agit d'une suspension d'utilisateur.

Une autre façon d'attribuer des suspensions aux travaux est la commande `qsub(1)` ou `qalter(1)` (voir l'option `-h`).

Options

Le TABLEAU 8 liste les options associées à `qhold`.

TABLEAU 8 Options de `qhold`

Option	Description
<code>-h {u o s},...</code>	Attribue une suspension <code>u</code> (utilisateur), <code>o</code> (opérateur) ou <code>s</code> (système) ou une combinaison de ces éléments à un ou plusieurs travaux.
<code>-help</code>	Imprime la liste de toutes les options.
<code>-u nomutilisateur,...</code> <code>-uall</code>	Les changements sont uniquement apportés aux travaux qui ont été soumis par les utilisateurs spécifiés dans la liste de noms d'utilisateurs. Les gestionnaires ont la possibilité d'utiliser la commande <code>qhold -uall</code> pour définir une suspension pour tous les travaux de tous les utilisateurs. Si vous utilisez le modificateur <code>-u</code> ou <code>-uall</code> , il est permis de spécifier une <i>liste_travail/ID_tâches</i> supplémentaire.
<i>liste_travail/ID_tâches</i>	Cette option se spécifie comme suit : <code>id_travail[.plage_tâches][.id_travail[.plage_tâches],...]</code> Si présent, <i>plage_tâches</i> limite l'effet de l'opération à la plage de tâches de travail de matrice spécifiée par le suffixe rattaché à l'ID du travail (voir l'option <code>-t</code> de <code>qsub(1)</code> pour plus de détails sur les matrices de travail). Le spécificateur de plage de tâches est de la forme <code>n[-m[:s]]</code> . La plage peut être un simple chiffre, une plage simple de la forme <code>n-m</code> ou une plage avec une valeur de pas. À la place de <i>liste_travail/ID_tâches</i> il est possible d'utiliser le mot clé <code>all</code> pour modifier l'état de suspension pour tous les travaux de l'utilisateur courant.

Variables d'environnement

Le TABLEAU 9 décrit les variables d'environnement associées à `qhold`.

TABLEAU 9 Variables d'environnement de `qhold`

Option	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine <code>qhold</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qhold</code> réside. Par défaut, l'hôte local est utilisé.

Voir aussi

`sge_intro(1)`, `qalter(1)`, `qrls(1)`, `qsub(1)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qhost (1)

Nom

`qhost` – Affiche le statut des hôtes, des files d’attente et des travaux de Sun Grid Engine.

Syntaxe

```
qhost [ -F [ nom_ressource,... ] [ -help ] [ -h liste_hôtes ]  
[ -j ] [ -l ressource=val,... ] [ -u utilisateur,... ]
```

Description

`qhost` affiche le statut courant des hôtes et des files d’attente disponibles de Sun Grid Engine, et des travaux associés aux files d’attente. Les options de sélection vous permettent d’obtenir des informations sur des hôtes, files d’attente, travaux ou utilisateurs spécifiques. Sans aucune option, `qhost` affichera la liste de tous les hôtes sans information aucune sur les files d’attente ou sur les travaux.

Options

Le TABLEAU 10 liste les options associées à `qhost`.

TABLEAU 10 Options de `qhost`

Option	Description
<code>-F [<i>nom_ressource</i>,...]</code>	<code>qhost</code> présentera une liste détaillée de la disponibilité courante des ressources par hôte pour toutes les ressources (si l’argument d’option est omis) ou pour celles contenues dans la liste <i>nom_ressource</i> . Reportez-vous à la description du « Format complet » dans la section, « Formats de sortie » ci-après pour plus de détails.
<code>-help</code>	Imprime la liste de toutes les options.
<code>-h <i>liste_hôtes</i></code>	Imprime la liste de tous les hôtes contenus dans <i>liste_hôtes</i> .
<code>-j</code>	Imprime tous les travaux s’exécutant sur les files d’attente hébergées par les hôtes affichés. Ce modificateur appelle <code>-q</code> de manière implicite.

TABEAU 10 Options de `qhost` (suite)

Option	Description
<code>-l ressource[=valeur],...</code>	Définit les ressources requises par les hôtes sur lesquels les informations sont requises. La correspondance est établie sur les hôtes.
<code>-q</code>	Affiche des informations sur les files d'attente hébergées par les hôtes affichés.
<code>-u utilisateur,...</code>	Affiche des informations limitées aux travaux et aux files d'attente associées avec les utilisateurs de la liste d'utilisateurs indiquée.

Formats de sortie

En fonction de si les options `-q`, `-F` et `-j` sont ou non présentes, il existe trois formats de sortie différents.

Format par défaut

Dans le format par défaut — c'est-à-dire sans les options `-q`, `-F` et `-j` —, la ligne d'en-tête s'imprime suivie d'une ligne pour chaque hôte, qui est composée des éléments suivants :

- Nom d'hôte,
- Architecture,
- Nombre de processeurs,
- Charge,
- Mémoire totale,
- Mémoire utilisée,
- Espace de swap total,
- Espace de swap utilisé.

Format avec les options

Si vous indiquez l'option `-q`, chaque ligne de statut d'hôte contient également des lignes supplémentaires pour chaque file d'attente hébergée par l'hôte, ces lignes sont composées des éléments suivants :

- Nom de la file d'attente.
- Type de la file d'attente : `B` (Lot), `I` (Interactif), `C` (Points de reprise), `P` (Parallèle), `T` (Transfert) ou une combinaison de ces types.
- Nombre d'emplacements de travail utilisés et disponibles.
- État de la file d'attente : `u` (inconnu) si le `sgc_execd(8)` correspondant ne peut pas être contacté, `a` (alarme), `A` (Alarme), `C` (interrompu agenda), `s` (interrompu), `S` (subordonné), `d` (désactivé), `D` (désactivé), `E` (erreur) ou une combinaison de ces états.

Si l'état est a(alarme) au moins un des seuils de charge définis dans la liste *load_thresholds* de la configuration de file (voir `queue_conf(5)`) est actuellement dépassé, ce qui empêche la programmation d'autres travaux sur cette file.

L'état A(larme) indique au contraire qu'au moins un des seuils de d'interruption de la file d'attente (voir `queue_conf(5)`) est actuellement dépassé. Par conséquent, les travaux s'exécutant dans cette file d'attente seront interrompus jusqu'à ce qu'aucun seuil ne soit plus violé.

Les états s(interrompue) et d(ésactivé) peuvent être attribués aux files d'attente et en être retirés via la commande `qmod(1)`. Interrompre une file d'attente entraîne l'interruption de tous les travaux s'exécutant dans cette file.

Les états D(ésactivé) et C(interrompue agenda) indiquent que la file d'attente a été désactivée ou interrompue automatiquement via l'utilitaire d'agenda de Sun Grid Engine (voir `calendar_conf(5)`), tandis que l'état S(ubordonné) indique que la file a été interrompue via la subordination à une autre file (voir `queue_conf(5)` pour plus de détails). Lorsqu'une file est interrompue (pour quelque cause que ce soit) tous les travaux s'exécutant dans cette file sont également interrompus.

Si l'état E(rreur) s'affiche pour une file, cela indique que `sge_execd(8)` a été incapable de localiser sur cet hôte l'exécutable `sge_shepherd(8)` afin de démarrer un travail. Veuillez contrôler le fichier journal des erreurs de ce `sge_execd(8)` pour des pistes sur la résolution du problème. Activez ensuite manuellement la file d'attente via l'option `-c` de la commande `qmod(1)`.

Si l'option `-F` a été utilisée, les informations sur la disponibilité des ressources sont imprimées à la suite de la ligne de statut d'hôte. Pour chaque ressource (celle sélectionnée dans un argument d'option de `-F` ou toutes les ressources si l'argument d'option a été omis), une unique ligne dont le format est le suivant s'affiche :

- Un spécificateur d'une lettre indiquant si la valeur de disponibilité de la ressource courante a été dominée par au choix :
 - `g` – un paramètre global de la grappe
 - `h` – un total de l'hôte ; ou
- Un second spécificateur d'une lettre indiquant la source de la valeur de disponibilité de ressource courante, pouvant être :
 - `l` – une valeur de charge rapportée pour la ressource,
 - `L` – une valeur de charge pour la ressource après que le facteur de mise à l'échelle de la charge définie par l'administrateur a été appliqué,
 - `c` – la disponibilité dérivée de l'utilitaire des ressources consommables (voir `complexes(5)`),
 - `v` – une valeur de configuration de complexes par défaut jamais surécrite par un rapport de charge ou une mise à jour de consommable,
 - `f` – une définition de disponibilité fixe dérivée d'un attribut de complexe non-consommable ou d'une limite de ressource fixe,

- Après un signe deux points, le nom de la ressource sur laquelle les informations s'affichent.
- Après le signe égal, la valeur de disponibilité courante de la ressource.

Les valeurs de disponibilité affichées et les sources dont elles dérivent sont toujours les valeurs minimales de toutes les combinaisons possibles. Par conséquent, par exemple, une ligne de la forme « qf:h_vmem=4G » indique qu'une file a couramment une mémoire virtuelle maximale disponible de 4 Go, que cette valeur est une valeur fixe (par exemple une limite de ressource dans la configuration de file) et qu'elle est dominée par la file d'attente, c'est-à-dire qu'au total la mémoire disponible sur l'hôte peut être supérieure mais que la file d'attente n'en autorise pas plus. Au contraire, la ligne « hl:h_vmem=4G » indiquerait une limite supérieure de 4 Go pour la mémoire virtuelle disponible, qui serait dérivée d'une valeur de charge couramment rapportée pour l'hôte. Ainsi, la file pourrait autoriser des travaux ayant des exigences de mémoire virtuelle supérieures, mais son hôte n'aurait réellement que 4 Go de disponibles.

Après la ligne de statut de la file d'attente (si -j) une ligne est imprimée pour chaque travail en cours d'exécution dans la file. Chaque ligne de statut de travail contient les éléments suivants :

- ID du travail.
- Nom du travail.
- Nom du propriétaire du travail.
- Statut du travail, selon le cas : t(ransfert en cours), r(exécution en cours), R(edémarré), s(interrompu), S(interrompu) ou T(seuil) (voir la section « Format réduit (sans -f ni -F) », page 76 pour des informations détaillées).
- Date et heure de début et fonction du travail (MASTER ou SLAVE - n'a un sens que dans le cas d'un travail parallèle).
- Priorité des travaux.

Variables d'environnement

Le TABLEAU 11 décrit les variables d'environnement associées à `qhost`.

TABLEAU 11 Variables d'environnement de `qhost`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine <code>qstat</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qhost</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

- Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `qalter(1)`, `qconf(1)`, `qhold(1)`, `qmod(1)`, `qstat(1)`,
`qsub(1)`, `queue_conf(5)`, `sge_commd(8)`, `sge_execd(8)`, `sge_qmaster(8)`,
`sge_shepherd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qmake (1)

Nom

qmake – make parallèle distribué, programmation par Sun Grid Engine

Syntaxe

qmake [*options*] -- [*gmake options*]

Description

qmake est un utilitaire `make(1)` distribué parallèle. La programmation des tâches `make` parallèles est effectuée par le logiciel Sun Grid Engine. Cet utilitaire est basé sur `gmake` (GNU `make`), version 3.78.1. Les options de ligne de commande `gmake` et Sun Grid Engine peuvent toutes être spécifiées. Elles sont séparées par les caractères `--`.

Toutes les options de Sun Grid Engine valides avec `qsub(1)` ou `qssh(1)` peuvent être spécifiées avec `qmake`. Voir `submit(1)` pour la description de toutes les options de ligne de commande de Sun Grid Engine. La page de manuel `make(1)` décrit la syntaxe de la ligne de commande `gmake`.

La syntaxe des fichiers `qmake` correspond à `gmake` et est décrite dans le *GNU Make Manual*.

Un appel `qmake` typique utilisera les options de ligne de commande de Sun Grid Engine, `-cwd`, pour qu'une opération `make` programmée soit démarrée dans le répertoire de travail courant sur l'hôte d'exécution, `-v chemin`, si l'environnement Sun Grid Engine n'est pas configuré dans le fichier de ressources d'interpréteur de commandes `.cshrc` ou `.profile` de l'utilisateur et requiert des emplacements dans un environnement parallèle (voir `sge_pe(5)`).

Si aucune requête de ressources n'est spécifiée (`QS_NAME option_ligne_commande -l`), `qmake` utilise la variable d'environnement `ARCH` pour demander la même architecture pour l'exécution des tâches que l'hôte de soumission. Si `ARCH` est définie, l'architecture spécifiée dans `ARCH` sera demandée en insérant l'option `-l arch=$ARCH` dans les options de ligne de commande. Si `ARCH` n'est pas définie, les tâches `make` peuvent être exécutées sur toute architecture disponible. Cela étant capital pour les travaux `make` typiques (compilations), un avertissement sera donné.

Exemples

L'exemple qui suit, l'EXEMPLE DE CODE 1, requerra entre un et 10 emplacements dans l'environnement `compiling` sur la même architecture que l'hôte de soumission. Les tâches `make` hériteront de l'environnement complet de l'interpréteur de commande appelant. Autant de tâches parallèles qu'il y aura d'emplacements octroyés par le logiciel Sun Grid Engine seront exécutées.

```
qmake -cwd -v PATH -pe compiling 1-10 --
```

EXEMPLE DE CODE 1

L'exemple suivant, l'EXEMPLE DE CODE 2, requerra de un à quatre emplacements dans l'environnement `make` sur la même architecture que l'hôte de soumission.

```
qmake -cwd -v PATH -- -j 4
```

EXEMPLE DE CODE 2

L'exemple suivant, l'EXEMPLE DE CODE 3, requerra que trois tâches `make` parallèles soient exécutées sur des hôtes d'architecture `solaris`. La soumission peut être effectuée sur un hôte de n'importe quelle architecture.

```
qmake -cwd -v PATH -l arch=solaris -pe make 3
```

EXEMPLE DE CODE 3

Le script shell suivant est l'EXEMPLE DE CODE 4.

```
#!/bin/sh
qmake -inherit --
```

EXEMPLE DE CODE 4

Vous pouvez soumettre le script shell de l'EXEMPLE DE CODE 4 en suivant le modèle de l'EXEMPLE DE CODE 5.

```
qsub -cwd -v PATH -pe make 1-10 [autres_options] <script>
```

EXEMPLE DE CODE 5

Dans l'EXEMPLE DE CODE 5, `qmake` héritera des ressources octroyées pour le travail soumis sous l'environnement `make`.

Variables d'environnement

Le TABLEAU 12 décrit les variables d'environnement associées à `qmake`.

TABLEAU 12 Variables d'environnement de `qmake`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine <code>qmake</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie• Le nom de la cellule par défaut, par ex. <code>default</code>
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>ARCH</code>	Architecture de l'hôte de soumission. Si cette variable est définie dans l'environnement de soumission, <code>qmake</code> demandera l'architecture indiquée pour l'exécution des travaux (voir description plus haut).
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qmake</code> réside. Par défaut, l'hôte local est utilisé.

Problèmes connus

Vous trouverez ci-après les problèmes connus associés à la commande `qmake`.

Serveur NFS lent

Une lenteur excessive du serveur peut être à l'origine de problèmes sur les fichiers en dépendant.

Par exemple : L'hôte `a` compile `a.c` en `a.o.`, l'hôte `b` compile `b.c` en `b.o.` et l'hôte `c` relie le programme `c` à partir de `a.o.` et `b.o.` Dans le cas d'une performance NFS médiocre, l'hôte `c` risque de ne pas encore voir les fichiers `a.o.` et `b.o.`

Commandes multiples dans une règle

Si plusieurs commandes sont exécutées dans une règle, le fichier makefile doit garantir qu'elles seront gérées comme *une* unique ligne de commande.

Par exemple, assumez que vous ayez entré ce qui suit.

```
libx.a:  
    cd x  
    ar ru libx.a x.o
```

Dans le cas ci-dessus, la construction de `libx.a` échouerait si les commandes étaient exécutées en parallèle (et si possible sur des hôtes différents). À la place, écrivez ce qui suit.

```
libx.a:  
    cd x ; ar ru libx.a x.o
```

Une autre solution efficace consisterait à écrire ce qui suit.

```
libx.a:  
    cd x ; \  
    ar ru libx.a x.o
```

Voir aussi

`submit(1)`, `sge_pe(5)`, ainsi que `make(1)` (page de manuel `make GNU`) et le *GNU Make Manual* dans `<racine_sge>/3rd_party/qmake`

Droits d'auteur

`qmake` contient des parties de Gnu Make (`gmake`), dont les droits d'auteur sont détenus par la Free Software Foundation, Inc., Boston, MA, États-Unis et qui est protégé par la licence Gnu General Public License.

Voir `sge_intro(1)` et les informations fournies dans `<racine_sge>/3rd_party/qmake` pour l'énoncé des autres droits et autorisations.

qmod (1)

Nom

qmod – Modifie une file d’attente Sun Grid Engine

Syntaxe

qmod [*options*] [*liste_travail/ID_tâches* | *liste_files*]

Description

qmod permet aux utilisateurs de la catégorie *propriétaires* (voir `queue_conf(5)` pour plus de détails) d’une station de travail de modifier l’état des files d’attente de Sun Grid Engine pour leur machine ainsi que l’état de leurs travaux. Un gestionnaire/opérateur ou le super-utilisateur peuvent exécuter qmod pour toute file et travail dans une grappe.

Options

Le TABLEAU 13 liste les options associées à qmod.

TABLEAU 13 Options de qmod

Option	Description
-c	Efface l’état d’erreur de la ou les files d’attente spécifiées.
-d	Désactive la ou les files d’attente, aucun travail supplémentaire ne sera acheminé vers les files d’attente désactivées tandis que les travaux s’exécutant déjà dans ces files d’attente pourront se terminer normalement. Cette option est le successeur de l’option <code>-soc</code> de Sun Grid Engine version 3.
-e	Active la ou les files d’attente. Cette option est le successeur de l’option <code>-xsoc</code> de Sun Grid Engine version 3.

TABLEAU 13 Options de `qmod` (suite)

Option	Description
<code>-f</code>	Force l'action de modification pour la file d'attente en dépit de l'état courant apparent de cette file. Par exemple, si une file semble interrompue mais que l'exécution des travaux semble continuer, le gestionnaire/opérateur peut forcer une opération d'interruption qui enverra un <code>SIGSTOP</code> aux travaux. Dans tous les cas, le statut de la file ou du travail sera défini même si le <code>sge_execd(8)</code> qui contrôle les files/travaux est inaccessible. Privilèges de gestionnaire/opérateur requis.
<code>-help</code>	Imprime la liste de toutes les options.
<code>-r</code>	Appliquée à une file d'attente, cette option reprogramme tous les travaux actuellement en cours d'exécution dans cette file. Appliquée à des travaux en cours, elle les reprogramme.
<code>-s</code>	Appliquée à des files d'attente, cette option interrompt ces files et tous les éventuels travaux actifs dans celles-ci. Appliquée à des travaux en cours, elle les interrompt. Si un travail est interrompu à la fois de manière explicite et pour cause d'interruption de sa file d'attente, la reprise successive de la file n'entraînera pas la reprise du travail.
<code>-us</code>	Appliquée à des files d'attente, cette option reprend ces files d'attente et tous les éventuels travaux actifs. Appliquée à des travaux en cours, elle en assure la reprise. Si un travail est interrompu à la fois de manière explicite et pour cause d'interruption de sa file d'attente, la reprise successive de la file n'entraînera pas la reprise du travail.
<code>-verify</code>	N'effectue aucune modification, imprime simplement ce qui serait fait si <code>-verify</code> n'était pas présent.
<code>liste_travail/ID_tâches</code> <code>liste_files</code>	<p>Travaux ou files d'attente sur lesquels <code>qmod</code> est supposé opérer. La <code>liste_travail/ID_tâches</code> est spécifiée sous l'une des formes suivantes :</p> <ul style="list-style-type: none"> • <code>id_travail[.plage_tâches][,id_travail[.plage_tâches], ...]</code> • <code>id_travail[.plage_tâches][id_travail[.plage_tâches] ...]</code> <p>Si présent, <code>plage_tâches</code> limite l'effet de l'opération à la plage de tâches de travail de matrice spécifiée par le suffixe rattaché à l'ID du travail (voir l'option <code>-t</code> de <code>qsub(1)</code> pour plus de détails sur les matrices de travail).</p> <p>Le spécificateur de plage de tâches est de la forme :</p> <p><code>n[-m[:s]][,n[-m[:s]] , ...]</code></p> <p>ou :</p> <p><code>n[-m[:s]][n[-m[:s]] ...]</code></p> <p>Par conséquent, c'est une liste séparée par des virgules ou des espaces de spécificateurs de plage <code>n[-m[:s]]</code>. Les plages sont concaténées en une plage d'ID de tâches complète. Chaque plage peut être un simple chiffre, une plage simple de la forme <code>n-m</code> ou une plage avec une valeur de pas.</p>

Variables d'environnement

Le TABLEAU 14 décrit les variables d'environnement associées à `qmod`.

TABLEAU 14 Variables d'environnement de `qmod`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qmod</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qmod</code> réside. Par défaut L'hôte local est utilisé.

Fichiers

- Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `sge_ckpt(1)`, `qstat(1)`, `queue_conf(5)`, `sge_execd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qmon (1)

Nom

qmon – Interface homme-machine OSF/Motif X-Windows pour Sun Grid Engine

Syntaxe

qmon [*options*]

Description

qmon permet aux administrateurs et aux utilisateurs de manipuler le système Sun Grid Engine depuis un environnement X-Window. qmon fournit différentes boîtes de dialogue reliées les unes aux autres de plusieurs manières. Une boîte de dialogue est affichée pour chacune des tâches que l'utilisateur veut accomplir via qmon. Il y a plusieurs façons d'arriver à la boîte de dialogue correspondant à une tâche donnée :

- La fenêtre principale de qmon qui s'ouvre au démarrage contient des icônes-boutons pour toutes les principales tâches administratives et utilisateur. Des bulles explicatives s'affichent lorsque vous pointez sur les différentes icônes.
- Le bouton de menu déroulant Tâche figure dans la barre de menu de la fenêtre principale de qmon. Cliquer dessus ouvre la liste des tâches disponibles. Sélectionner l'une de ces tâches ouvre la boîte de dialogue correspondante.
- Le menu déroulant Tâche contient également des raccourcis qui peuvent être utilisés pour appeler directement les boîtes de dialogue des tâches depuis la fenêtre principale en appuyant sur une séquence particulière de touches du clavier.
- Lorsque vous naviguez dans une boîte de dialogue et ses sous-boîtes de dialogue, des liens menant à d'autres boîtes de dialogue apparaissent à chaque fois qu'une connexion s'impose entre deux boîtes de dialogue. Appuyer sur les boutons qui représentent ces liens ouvre les autres boîtes de dialogue.

Options

Les options prises en charge sont les options X Toolkit standard décrites dans la section X(1), Options. En sus, `qmon` prend en charge les options listées dans le TABLEAU 15.

TABLEAU 15 Options de `qmon`

Option	Description
<code>-cmap</code>	Installe une carte de couleurs privée pour <code>qmon</code> . Cela est parfois utile si d'autres applications ont déjà alloué de nombreuses couleurs et si <code>qmon</code> , par conséquent, imprime les messages d'erreur correspondants. L'utilisation d'une carte de couleurs privée se traduira toutefois par des changements de carte de couleurs à chaque fois que vous entrerez dans ou quitterez les fenêtres <code>qmon</code> .
<code>-fontFamily {big medium small}</code>	Signale à <code>qmon</code> d'utiliser des familles de polices de tailles différentes pour les différentes résolutions.
<code>-help</code>	Affiche des informations sur l'utilisation.
<code>-nologo</code>	Démarre sans logo.

Boîtes de dialogue

Les sections qui suivent décrivent les boîtes de dialogue associées à la commande `qmon`.

Contrôle des travaux

La boîte de dialogue Contrôle des travaux fournit un dossier de listes présentées sous forme de tableaux des travaux encore en attente, des travaux en cours d'exécution et des travaux récemment finis. Cette boîte de dialogue permet d'afficher des informations détaillées sur les travaux, et de supprimer et d'interrompre des travaux sélectionnés. En sus, la boîte de dialogue Contrôle des travaux comporte des liens menant à la boîte de dialogue Soumission qui permet de soumettre de nouveaux travaux ou de changer les attributs des travaux en attente (bouton `ModifAttr`). Les champs qui s'affichent dans les tableaux et les travaux affichés peuvent être personnalisés en appuyant sur le bouton `Personnaliser`. La personnalisation effectuée peut ensuite être enregistrée dans le fichier `~/qmon_preferences`, elle sera utilisée aux redémarrages qui suivront pour la configuration initiale de la boîte de dialogue Contrôle des travaux.

Contrôle des files d'attente

La boîte de dialogue Contrôle des files d'attente et sa hiérarchie de sous-boîtes de dialogue permettent à l'utilisateur de contrôler le statut des files d'attente de Sun Grid Engine actuellement configurées dans le système et à l'administrateur d'ajouter de nouvelles files d'attente ou de modifier ou supprimer celles existantes. Chaque icône-bouton de la première boîte de dialogue Contrôle des files d'attente représente une file d'attente Sun Grid Engine configurée. Les symboles des icônes, la couleur et le texte des boutons vous informent sur l'architecture, le statut et certains attributs de base des files d'attente. La première boîte de dialogue vous permet également de supprimer des files d'attente sélectionnées au préalable. Les files d'attente se sélectionnent en cliquant avec le bouton gauche de la souris sur les icônes ou dans une zone rectangulaire autour des boutons.

En appuyant sur le bouton Ajouter ou Modifier ou en utilisant le menu contextuel qui s'affiche en cliquant avec le bouton droit de la souris dans la fenêtre des icônes de la première boîte de dialogue Contrôle des files d'attente, une sous-boîte de dialogue permettant de configurer les files d'attente de Sun Grid Engine s'ouvre. Une file d'attente doit être sélectionnée pour utiliser l'opération de modification. La sous-boîte de dialogue de configuration permet de définir le nom de la file d'attente et celui de l'hôte, ou affiche les noms correspondants dans le cas d'une modification. Les paramètres de configuration des files d'attente (voir `queue_conf(5)`) sont divisés en plusieurs catégories (Configuration générale, Méthode d'exécution, Points de reprise, Seuils de charge/interruption, Limites, Complexes, Accès utilisateurs, Accès projets (Sun Grid Engine, Enterprise Edition uniquement), Subordonnés, Propriétaires) que vous pouvez sélectionner dans la zone d'onglets dans le bas de la sous-boîte de dialogue de configuration des files. L'administrateur peut sélectionner les valeurs par défaut de files d'attente déjà configurées (bouton Clone). En appuyant sur le bouton Ok, les définitions sont enregistrées auprès de `sgemaster(8)`. La boîte de dialogue Contrôle des files d'attente peut être personnalisée de façon similaire à la boîte Contrôle des travaux. Les paramètres appliqués sont également sauvegardés dans `~/qmon_preferences`.

Soumission

La boîte de dialogue Soumission d'un travail permet de soumettre des travaux par lots et des travaux interactifs, c'est également celle qui est appelée lors du changement des attributs des travaux en attente depuis la boîte de dialogue Contrôle des travaux décrite ci-dessus (bouton ModifAttr). Pour basculer entre les modes lot et interactif, utilisez le bouton Lot/Interactif qui se trouve dans le haut de la colonne de boutons sur la droite de l'écran Soumission d'un travail.

Cette boîte de dialogue est un dossier contenant deux pages pour la préparation des travaux. Les paramètres les plus fréquemment utilisés dans le cadre de la soumission d'un travail sont proposés sur la page Générale. Un script de travail doit être défini, tous les autres champs sont facultatifs. Si le travail demande la spécification de prérequis avancés, l'onglet Avancée peut être utilisé pour passer à un panneau de paramètres plus complet.

Si les ressources requises sont obligatoires pour le travail, il faut utiliser le bouton icône Demander des ressources pour afficher la sous-boîte de dialogue Ressources demandées. Cette sous-boîte de dialogue permet de sélectionner les ressources requises pour le travail et de définir les quantités à fournir de ces ressources. Les ressources disponibles sont constituées par les attributs de complexe déclarés *demandables* (voir `complex(5)` pour plus de détails). Les exigences en ressources peuvent être Fixes, cas dans lequel elles doivent être satisfaites pour qu'un travail puisse démarrer dans une file d'attente, ou Souples, cas dans lequel elles sont octroyées au fur et à mesure qu'elles se libèrent.

Fermer la sous-boîte de dialogue Ressources demandées avec le bouton Terminé réserve les exigences spécifiées pour le travail. Appuyer sur le bouton Soumettre sur la première boîte de dialogue Soumission soumet le travail.

Configuration des complexes

La boîte de dialogue Configuration des complexes permet à l'administrateur d'ajouter de nouveaux complexes ou de modifier ou supprimer ceux existants (voir `complex(5)`). Cette boîte de dialogue offre une liste de sélection pour les complexes existants et affiche la configuration de celui qui est sélectionné. En appuyant sur le bouton Supprimer, le complexe sélectionné est supprimé de la configuration. En appuyant sur le bouton Ajouter/Modifier, une boîte de dialogue de configuration de complexes s'ouvre vous permettant de créer de nouveaux complexes ou vous fournissant les moyens de modifier les complexes existants. Si vous voulez créer un nouveau complexe, vous devez en définir le nom. Le nom d'un complexe à modifier s'affiche dans le même champ d'entrée de texte dans le cas d'une opération de modification. La boîte de dialogue de configuration fournit une liste prenant la forme d'un tableau des entrées de complexe, et une région d'entrée pour les entrées nouvelles ou modifiées. Le bouton Ajouter met à jour le tableau avec la nouvelle entrée ou celle modifiée et le bouton Ok enregistre le complexe supplémentaire ou modifié auprès de `sgc_qmaster(8)`.

Configuration des hôtes

La boîte de dialogue Configuration des hôtes permet d'administrer trois types de listes d'hôtes :

- hôtes d'administration,
- hôtes de soumission,
- hôtes d'exécution.

La liste des hôtes à manipuler se sélectionne en cliquant sur l'onglet correspondant. Les deux premières listes d'hôtes permettent uniquement d'ajouter et de supprimer des entrées, autorisant ainsi une permission administrative ou de soumission pour les hôtes de ces listes, ou en la refusant. Les entrées de la liste des hôtes d'exécution fournissent en sus la possibilité de définir des facteurs de mise à l'échelle pour les

capteurs de charge, les attributs de complexes consommables et les attributs d'accès (access, xaccess et projects, xprojects pour le mode Sun Grid Engine, Enterprise Edition uniquement) comme décrit dans `complex(5)`. Dans un système Sun Grid Engine, Enterprise Edition, l'utilisation des UC, de la mémoire et des E/S rapportée pour les travaux en cours d'exécution peut être mise à l'échelle et la performance relative d'un hôte rectifiée avec le facteur de capacité de ressources (voir `host_conf(5)`).

Configuration de la grappe

Cette boîte de dialogue conserve la configuration globale de la grappe ainsi que des configurations dérivées spécifiques des hôtes (voir `sge_conf(5)`). À son ouverture, elle affiche une liste de sélection de tous les hôtes auxquels une configuration est attribuée. Le nom spécial « global » fait référence à la configuration globale de la grappe. En appuyant sur le bouton Ajouter/Modifier, une sous-boîte de dialogue s'ouvre, vous permettant de modifier la configuration de la grappe. Pour les configurations spécifiques des hôtes, les champs « global » sont insensibles et seuls des paramètres autorisés peuvent être manipulés.

Configuration du programmeur

La boîte de dialogue Configuration du programmeur fournit les moyens de changer le comportement du démon programmeur de Sun Grid Engine `sge_schedd(8)`. Cette boîte de dialogue contient une représentation de tous les paramètres de configuration du programmeur (voir description dans `sched_conf(5)`). Elle est divisée en deux sections Paramètres généraux et Ajustement de charge qui peuvent être sélectionnées en utilisant les onglets correspondants. Le bouton Ok enregistre les éventuels changements auprès de `sge_qmaster(8)`.

Configuration des agendas

La boîte de dialogue Configuration des agendas permet à l'administrateur d'ajouter de nouveaux agendas ou de modifier ou supprimer ceux existants (voir `calendar_conf(5)`). Cette boîte de dialogue offre une liste de sélection des agendas existants et affiche la configuration de celui sélectionné. En appuyant sur le bouton Supprimer, l'agenda sélectionné est supprimé de la configuration. En appuyant sur le bouton Ajouter/Modifier, une boîte de dialogue de configuration d'agendas s'ouvre vous permettant de créer de nouveaux agendas ou vous fournissant les moyens de modifier ceux existants. Le bouton Ok enregistre l'agenda supplémentaire ou modifié auprès de `sge_qmaster(8)`.

Configuration des utilisateurs

Les droits d'accès des utilisateurs sont contrôlés via la boîte de dialogue Configuration des utilisateurs. Les onglets situés dans la partie gauche de cette boîte de dialogue permettent de sélectionner au choix :

- Configuration des comptes de gestionnaire,
- Configuration des comptes d'opérateur,
- Définition des jeux d'utilisateur,
- Définition des comptes d'utilisateur (mode Sun Grid Engine, Enterprise Edition uniquement).

Les comptes d'utilisateur ajoutés à la liste des comptes de gestionnaire ou d'opérateur reçoivent des droits les autorisant à agir en tant que, respectivement, gestionnaire ou opérateur lorsqu'ils accèdent à Sun Grid Engine sous leur propre compte.

Les listes de jeux d'utilisateurs sont utilisées avec les paramètres de configuration de grappe, projet, file d'attente, hôte *user_lists* et *xuser_lists* (voir *queue_conf(5)*, *project(5)* et *sge_conf(5)*) pour contrôler l'accès aux hôtes, files d'attente, projets (uniquement disponible dans un système Sun Grid Engine, Enterprise Edition) et à l'ensemble de la grappe. Un jeu d'utilisateurs est un simple ensemble de noms d'utilisateurs et de noms de groupes UNIX. Les noms de groupes sont identifiés par le préfixe « @ ». Les jeux d'utilisateurs déjà définis sont affichés dans une liste de sélection. Ces listes peuvent être modifiées et de nouvelles listes peuvent être créées en utilisant la boîte de dialogue de définition Jeu d'utilisateurs.

Dans un système Sun Grid Engine, Enterprise Edition, les jeux d'utilisateurs peuvent être utilisés en tant que listes de contrôle d'accès (utilisation équivalente à celle effectuée dans un système Sun Grid Engine) et/ou en tant que Service requis pour les politiques fonctionnelle et de remplacement (voir Configuration des tickets plus loin).

Un système Sun Grid Engine, Enterprise Edition exige également l'ajout de comptes ayant accès au système sous la forme d'entrées de la base de données d'utilisateurs de Sun Grid Engine, Enterprise Edition (voir *user(5)*). Cela peut être fait avec la sous-boîte de dialogue Utilisateur.

Le bouton Tickets dans la liste des boutons sur la droite de la boîte de dialogue ouvre la boîte de dialogue Configuration des tickets (voir ci-dessous). Cela est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition.

Configuration des environnements parallèles

Les interfaces PE (*Parallel environment*, environnement parallèle) peuvent être configurées avec cette boîte de dialogue. Les interfaces PE sont nécessaires pour décrire la façon dont les environnements de programmation parallèles tels que PVM (Parallel Virtual Machine), MPI (Message Passing Interface) ou les systèmes parallèles à mémoire partagée doivent être instanciés pour imposer des restrictions d'accès sur les PE. À l'ouverture de cette boîte de dialogue, la liste des PE déjà

configurés s'affiche avec la configuration courante (voir `pe_conf(5)`) de l'interface PE sélectionnée. Pour ajouter de nouvelles interfaces PE ou modifier celles existantes, des boutons Ajouter et Modifier sont disponibles et ouvrent une sous-boîte de dialogue de configuration PE. Après avoir appliqué les changements et fermé la sous-boîte de dialogue avec le bouton Ok, l'interface PE nouvelle ou modifiée est enregistrée auprès de `sge_qmaster(8)`.

Configuration de l'écriture de points de reprise

Cette boîte de dialogue permet de configurer les interfaces d'environnement de point de reprise. Les environnements de point de reprise sont nécessaires pour décrire les attributs des différentes méthodes d'écriture de points de reprise et de leurs dérivées sur de nombreuses plates-formes système d'exploitation prises en charge par Sun Grid Engine. À l'ouverture de la boîte de dialogue, la liste des environnements de point de reprise déjà configurés s'affiche ainsi que la configuration courante (voir `checkpoint(5)`) de l'environnement de point de reprise sélectionné. Pour ajouter un nouvel environnement de point de reprise ou modifier ceux existants, des boutons Ajouter et Modifier sont disponibles et ouvrent une sous-boîte de dialogue de configuration d'environnement de point de reprise. Après avoir appliqué les changements et fermé la sous-boîte de dialogue avec le bouton Ok, l'environnement de point de reprise nouveau ou modifié est enregistré auprès de `sge_qmaster(8)`.

Configuration des tickets

Cette boîte de dialogue offre une vue d'ensemble et un écran d'édition pour l'allocation des tickets pour les politiques de programmation basée sur le partage, fonctionnelle et de remplacement. Elle est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.

Le bouton Deadline Job (Travail avec limite) ouvre la boîte de dialogue Configuration des utilisateurs. Veuillez passer à la sous-boîte de dialogue Jeu d'utilisateurs et sélectionner les utilisateurs nommés « deadlineusers ». Seuls les utilisateurs de ce jeu d'utilisateurs peuvent soumettre des travaux d'initiation limite.

Le bouton Politique d'arborescence des partages ouvre la boîte de dialogue permettant de créer et de modifier l'arborescence des partages de Sun Grid Engine, Enterprise Edition (voir `share_tree(5)` et `schedd_conf(5)` pour la description des paramètres de configuration).

Le bouton Politique fonctionnelle ouvre la boîte de dialogue permettant de créer et de modifier l'allocation des partages fonctionnels (voir `sched_conf(5)`, `access_list(5)`, `project(5)`, `queue_conf(5)` et `user(5)` pour la description des différents types de partages fonctionnels et des paramètres de pondération configurables).

Le bouton Politique de remplacement ouvre la boîte de dialogue permettant de créer et de modifier l'allocation des tickets de remplacement (voir `access_list(5)`, `project(5)`, `queue_conf(5)` et `user(5)` pour la description des différents types de tickets de remplacement).

Configuration des projets

Ce bouton ouvre une boîte de dialogue permettant de créer des projets. Il est uniquement pris en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Il n'est pas disponible pour les systèmes Sun Grid Engine.

Cette boîte de dialogue offre une liste de sélection des projets existants et affiche la configuration de celui sélectionné. En appuyant sur le bouton Supprimer, le projet sélectionné est supprimé de la configuration. En appuyant sur le bouton Ajouter/Modifier une boîte de dialogue de configuration de projets s'ouvre vous permettant de créer de nouveaux projets ou vous fournissant les moyens de changer ceux existants. Configurer un projet signifie en fait octroyer ou refuser l'accès à un projet pour des jeux d'utilisateurs (voir Configuration des utilisateurs ci-dessus ainsi que `project(5)`). Le bouton Ok enregistre le projet supplémentaire ou modifié auprès de `sge_qmaster(8)`.

Navigateur

L'objectif de la boîte de dialogue Navigateur d'objets est multiple : en premier lieu, les messages de Sun Grid Engine et `qmon` tels que la notification d'une erreur ou de la réussite d'une action entreprise au préalable peuvent être affichés dans la fenêtre de sortie de cette boîte de dialogue. De même, la sortie standard et la sortie d'erreur standard de `qmon` peuvent être déviées sur la fenêtre de sortie du Navigateur d'objets.

En sus, le Navigateur d'objets peut être utilisé pour afficher des informations en continu sur les objets de Sun Grid Engine au fur et à mesure que le pointeur de la souris se déplace sur les icônes ou les entrées de tableau qui les représentent dans les autres boîtes de dialogue de `qmon`. À l'heure actuelle, seul l'affichage de la configuration de deux objets Sun Grid Engine dans deux boîtes de dialogue différentes est pris en charge :

- Les configurations de file s'affichent dès que le pointeur de la souris entre dans l'icône d'une file dans la première boîte de dialogue Contrôle des files d'attente (voir ci-dessus). Cet utilitaire s'active en appuyant sur le bouton File d'attente dans la boîte de dialogue Navigateur d'objets.
- Des informations détaillées sur les travaux s'impriment dès que l'utilisateur amène le pointeur de la souris sur une ligne de la boîte de dialogue Contrôle des travaux (voir ci-dessus) attribuée à un travail en attente ou en cours.
- En sus, des informations de programmation s'affichent dans le navigateur si vous appuyez sur le bouton Pourquoi ? de la boîte de dialogue Contrôle des travaux. Dans ce cas, la boîte de dialogue Navigateur s'ouvre de manière implicite et toutes les informations relatives à la programmation s'affichent.

Quitter

L'icône-bouton Quitter n'est reliée à aucune boîte de dialogue. Elle sert uniquement à fermer toutes les boîtes de dialogue `qmon` actives et à arrêter l'application.

Ressources

Les ressources disponibles, leur signification et la syntaxe à suivre pour les modifier sont décrites dans le fichier de ressources `qmon` par défaut (voir la section Fichiers ci-dessous pour l'emplacement du fichier de ressources).

Variables d'environnement

Le TABLEAU 16 décrit les variables d'environnement associées à `qmon`.

TABLEAU 16 Variables d'environnement de `qmon`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qmon</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie• Le nom de la cellule par défaut ; e.g. <code>default</code>
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qmon</code> réside. Par défaut, l'hôte local est utilisé.

Restrictions

Si la ligne qui doit être entrée dans une fenêtre d'édition est plus longue que la largeur de la fenêtre, le texte sort de la limite de la fenêtre.

Fichiers

- Fichiers de ressources d'exemple `qmon - <racine_sge>/qmon/Qmon`
- Fichiers de ressources système `qmon - /usr/lib/X11/defaults/Qmon`
- Fichiers de ressources utilisateur `qmon - $HOME/Qmon`
- Fichiers de personnalisation de travaux/files `qmon - $HOME/.qmon_preferences`

Voir aussi

`sge_intro(1)`, `sge_conf(5)`, `access_list(5)`, `sge_pe(5)`,
`calendar_conf(5)`, `complex(5)`, `project(5)`, `queue_conf(5)`,
`sched_conf(5)`, `user(5)`, `sge_qmaster(8)`

Droits d'auteur

Voir `sge_intro(1)` et les informations fournies dans `<racine_sge>/3rd_party/qmon` pour l'énoncé des autres droits et autorisations et pour les crédits devant être accordés aux développeurs de widgets gratuits et du domaine public.

qr1s(1)

Nom

`qr1s` – Retire les suspensions des travaux Sun Grid Engine suspendus.

Syntaxe

```
qr1s [ -h {u|o|s},... ] [ -help ] [ liste_travail/ID_tâches ]
```

```
qr1s [ -h {u|o|s},... ] [ -help ] -u liste_utilisateurs | -uall
```

Description

`qr1s` fournit à un utilisateur, un opérateur ou un gestionnaire le moyen de retirer les *suspensions* d'un ou plusieurs travaux en attente dont l'exécution doit être programmée. Tant qu'une suspension quelconque est attribuée à un travail, ce dernier ne peut pas être programmé.

Les suspensions peuvent être attribuées aux travaux avec les commandes `qhold(1)`, `qsub(1)` ou `qalter(1)`.

Il existe trois types de suspensions différents :

- Utilisateur – Les suspensions d'utilisateur peuvent être attribuées et retirées par les gestionnaires, les opérateurs et le propriétaire des travaux.
- Opérateur – Les suspensions d'opérateur peuvent être attribuées et retirées par les gestionnaires et les opérateurs.
- Système – Les suspensions de système ne peuvent être attribuées et retirées que par les gestionnaires.

Si aucun type de suspension n'est spécifié avec l'option `-h` (voir TABLEAU 17), le système assume par défaut qu'il s'agit d'une suspension d'utilisateur.

Une autre façon de retirer des suspensions des travaux est la commande `qalter(1)` (voir l'option `-h` dans le TABLEAU 17).

Options

Le TABLEAU 17 liste les options associées à `qrls`.

TABLEAU 17 Options de `qrls`

Option	Description
<code>-h {u o s},...</code>	Retire une suspension <code>u</code> (utilisateur), <code>o</code> (opérateur) ou <code>s</code> (système) ou une combinaison de ces éléments de un ou plusieurs travaux.
<code>-help</code>	Imprime la liste de toutes les options.
<code>-u nomutilisateur,... -uall</code>	Modifie l'état de suspension des travaux qui ont été soumis par les utilisateurs spécifiés dans la liste de noms d'utilisateurs. Les gestionnaires ont la possibilité d'utiliser la commande <code>qrls -uall</code> pour modifier l'état de suspension pour les travaux de tous les utilisateurs. Si vous utilisez le modificateur <code>-u</code> ou <code>-uall</code> , vous pouvez spécifier une <i>liste_travail/ID_tâches</i> supplémentaire.
<i>liste_travail/ID_tâches</i>	Cette option se spécifie comme suit : <i>id_travail[.plage_tâches][,id_travail[.plage_tâches],...]</i> Si présent, <i>plage_tâches</i> limite l'effet de l'opération à la plage de tâches de travail de matrice spécifiée par le suffixe rattaché à l'ID du travail (voir l'option <code>-t</code> de <code>qsub(1)</code> pour plus de détails sur les matrices de travail). Le spécificateur de plage de tâches est de la forme <i>n[-m[:s]]</i> . La plage peut être un simple chiffre, une plage simple de la forme <i>n-m</i> ou une plage avec une valeur de pas. À la place de <i>liste_travail/ID_tâches</i> il est possible d'utiliser le mot clé <code>all</code> pour modifier tous les travaux de l'utilisateur courant.

Variables d'environnement

Le TABLEAU 18 décrit les variables d'environnement associées à `qrls`.

TABLEAU 18 Variables d'environnement de `qrls`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qrls</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qrls</code> réside. Par défaut, l'hôte local est utilisé.

Voir aussi

`sge_intro(1)`, `qalter(1)`, `qhold(1)`, `qsub(1)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qselect(1)

Nom

`qselect` – Sélectionne des files d’attente.

Syntaxe

```
qselect [ -help ] [ -l ressource=val,... ] [ -pe nom_pe,... ]  
[ -q file,... ] [ -U utilisateur,... ]
```

Description

`qselect` imprime la liste des noms des files d’attente de Sun Grid Engine qui correspondent aux critères de sélection spécifiés dans les arguments de `qselect` décrits dans le TABLEAU 19. La sortie de `qselect` peut être fournie à d’autres commandes de Sun Grid Engine pour appliquer des actions à des ensembles de files d’attente sélectionnés. Par exemple, `qselect` peut être utilisé avec l’option `-mqattr` de `qconf(1)` pour modifier des attributs de file d’attente sur un ensemble de files d’attente.

Options

Le TABLEAU 19 liste les options associées à `qselect`.

TABLEAU 19 Options de `qselect`

Option	Description
<code>-help</code>	Imprime la liste de toutes les options.
<code>-l <i>ressource</i>[=<i>valeur</i>],...</code>	Définit les ressources qui seront allouées par les files d’attente qui devraient être incluses dans la sortie de liste de files d’attente.

TABEAU 19 Options de `qselect` (suite)

Option	Description
<code>-pe nom_pe,...</code>	Inclut dans la sortie les files d'attente qui sont rattachées à au moins un des environnements parallèles de la liste séparée par des virgules d'arguments.
<code>-q file,...</code>	Spécifie directement les files d'attente à inclure dans la sortie. Cette option n'a en général un sens que conjointement avec une autre option de <code>qselect</code> pour extraire un sous-ensemble de noms de files d'attente d'une liste indiquée par <code>-q</code> .
<code>-U utilisateur,...</code>	Inclut les files d'attente auxquelles les utilisateurs spécifiés ont accès dans la sortie de <code>qselect</code> .

Exemples

Les exemples suivants illustrent l'utilisation de la commande `qselect`.

```
qselect -l arch=linux
```

EXEMPLE DE CODE 6

La commande de l'EXEMPLE DE CODE 6 imprime les noms des files d'attente qui résident sur des machines Linux.

```
qselect -l arch=linux -U alain,sophie
```

EXEMPLE DE CODE 7

Dans l'EXEMPLE DE CODE 7, qui étend l'EXEMPLE DE CODE 6, la seconde commande ajoutée restreint la sortie aux files d'attente ayant des droits d'accès pour les utilisateurs `alain` et `sophie`.

```
qconf -mqattr h_vmem=1GB `qselect -l arch=linux
```

EXEMPLE DE CODE 8

L'EXEMPLE DE CODE 8 porte l'attribut de file `h_vmem` à 1 Go sur les files d'attente résidant sur des machines Linux (voir les pages de manuel de `qconf(1)` pour tout détail sur le format de l'option `-mqattr` et la page de manuel `queue_conf(5)` pour tout détail sur les entrées de configuration de file d'attente).

Variables d'environnement

Le TABLEAU 20 décrit les variables d'environnement associées à `qselect`.

TABLEAU 20 Variables d'environnement de `qselect`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qselect</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie• Le nom de la cellule par défaut, par ex. <code>default</code>
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qselect</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `qconf(1)`, `qmod(1)`, `qstat(1)`, `queue_conf(5)`,
`sge_commd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qstat(1)

Nom

qstat – Affiche le statut des travaux et des files d’attente de Sun Grid Engine.

Syntaxe

```
qstat [ -ext ] [ -f ] [ -F [ nom_ressource,... ] ] [ -g d ] [ -help ]  
[ -j [ liste_travaux ] ] [ -l ressource=val,... ] [ -ne ]  
[ -pe nom_pe,... ] [ -q file,... ] [ -r ]  
[ -s {r|p|s|z|hu|ho|hs|hj|ha|h}[+] ] [ -t ] [ -U utilisateur,... ]  
[ -u utilisateur,... ]
```

Description

qstat affiche le statut courant des files d’attente de Sun Grid Engine disponibles et des travaux y associés. Les options de sélection vous permettent d’obtenir des informations sur des travaux, des files d’attente ou des utilisateurs spécifiques. Sans aucune option, qstat n’affichera qu’une liste de travaux sans informations sur le statut des files.

Options

Le TABLEAU 21 liste les options associées à `qstat`.

TABLEAU 21 Options de `qstat`

Option	Description
<code>-alarm</code>	Affiche la ou les raisons à l'origine de l'état d'alarme d'une file d'attente, à raison d'une ligne contenant la valeur de la ressource et le seuil par raison. Pour tout détail sur la valeur des ressources, consultez la description du « Format complet » dans la section « Formats de sortie » ci-après.
<code>-ext</code>	Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine. Elle affiche des informations supplémentaires relatives à Sun Grid Engine, Enterprise Edition pour chaque travail (voir « Formats de sortie » ci-après).
<code>-f</code>	Demande le format complet pour l'affichage des informations. L'option <code>-f</code> affiche des informations récapitulatives sur toutes les files d'attente avec la liste des travaux en attente figurant dans ces files.
<code>-F [<i>nom_ressource</i>,...]</code>	Comme dans le cas de <code>-f</code> , les informations affichées sont relatives à la fois aux travaux et aux files. En sus, <code>qstat</code> présentera une liste détaillée de la disponibilité courante des ressources par file d'attente et ce pour toutes les ressources (si l'argument facultatif est omis) ou pour toutes celles contenues dans la liste <i>nom_ressource</i> . Reportez-vous à la description du « Format complet » dans la section « Formats de sortie » ci-après.

TABLEAU 21 Options de `qstat` (suite)

Option	Description
<code>-g d</code>	Affiche les matrices de travail sous forme verbeuse à raison d'une ligne par tâche de travail. Par défaut, les matrices de travail sont regroupées et toutes les tâches ayant le même statut (pour les tâches en attente uniquement) s'affichent dans une même ligne. Le champ de la plage d'ID de tâches de travail de matrice dans la sortie (voir section « Formats de sortie ») spécifie les ensembles de tâches correspondants. Le modificateur <code>-g</code> n'a actuellement que l'argument optionnel <code>d</code> . Les autres arguments optionnels sont réservés à des extensions futures.
<code>-help</code>	Imprime la liste de toutes les options.
<code>-j [liste_travaux]</code>	Imprime pour tous les travaux en attente ou pour les travaux contenus dans <i>liste_travaux</i> la raison de leur non-programmation.
<code>-l ressource[=valeur],...</code>	Définit les ressources requises par les travaux ou octroyées par les files d'attente sur lesquelles les informations sont demandées. La mise en correspondance est effectuée sur les files. Les travaux en attente sont limités aux travaux qui devraient s'exécuter dans l'une des files ci-dessus.
<code>-ne</code>	En combinaison avec l'option <code>-f</code> , supprime l'affichage des files vides. Cela signifie que les files d'attente dans lesquelles il n'y a en fait aucun travail en cours ne s'affichent pas.
<code>-pe nom_pe,...</code>	Affiche des informations de statut pour toutes les files d'attente qui sont rattachées à au moins un des environnements parallèles listés dans l'argument optionnel séparé par des virgules. Les informations de statut relatives aux travaux s'affichent soit pour les travaux qui s'exécutent dans une des files d'attente sélectionnées soit pour ceux en attente qui devraient en principe être programmés dans ces files d'attente.
<code>-q file,...</code>	Spécifie la file d'attente dans laquelle les informations sur les travaux vont être affichées.
<code>-r</code>	Imprime des informations étendues sur les ressources requises par les travaux affichés. Reportez-vous à la sous-section « Formats de sortie » de la section « Format développé » ci-après pour des informations détaillées.

TABEAU 21 Options de `qstat` (*suite*)

Option	Description
<code>-s {p r s z hu ho hs hj ha h}[+]</code>	Imprime uniquement les travaux se trouvant dans l'état spécifié — toute combinaison d'états est possible. <code>-s prs</code> correspond à la sortie <code>qstat</code> normale sans <code>-s</code> du tout. Pour afficher les travaux récemment finis, utilisez <code>-s z</code> . Pour afficher les travaux faisant l'objet d'une suspension d'utilisateur/opérateur/système, utilisez l'option <code>-s hu/ho/hs</code> . L'option <code>-s ha</code> affiche les travaux qui ont été soumis avec la commande <code>qsub -a.qstat -s hj</code> affiche tous les travaux qui ne sont pas disponibles pour l'exécution à l'exception de ceux ayant des entrées dans la liste des dépendances de travaux (voir les options <code>-a</code> et <code>-hold_jid</code> de <code>qsub(1)</code>).
<code>-t</code>	Imprime des informations étendues sur les sous-tâches contrôlées des travaux parallèles affichés. Reportez-vous à la sous-section « Formats de sortie » de la section « Format développé » ci-dessous pour des informations détaillées. Les sous-tâches de travaux parallèles ne doivent pas être confondues avec les tâches de travail de matrice (voir l'option <code>-g</code> ci-dessus et l'option <code>-t</code> de <code>qsub(1)</code>).
<code>-U utilisateur,...</code>	Affiche des informations de statut relativement aux files d'attente auxquelles les utilisateurs spécifiés ont accès. Les informations de statut relatives aux travaux s'affichent pour les travaux qui s'exécutent dans l'une des files d'attente sélectionnées ou qui sont en attente et devraient en principe être programmés sur ces files d'attente.
<code>-u utilisateur,...</code>	Affiche des informations sur les seuls travaux et files d'attente associés aux utilisateurs de la liste d'utilisateurs indiquée. Les informations de statut des files s'affichent si les options <code>-f</code> ou <code>-F</code> sont aussi spécifiées et si l'utilisateur exécute des travaux dans ces files.

Formats de sortie

Selon si les options `-alarm`, `-f` ou `-F` et `-r` et `-t` sont ou non présentes, trois formats de sortie peuvent être obtenus. Dans le cas d'un système Sun Grid Engine, Enterprise Edition, l'option `-ext` peut être utilisée pour afficher des informations supplémentaires pour chaque travail.

Format réduit (sans `-f ni -F`)

À la suite de la ligne d'en-tête, une ligne composée des éléments suivants est imprimée pour chaque travail :

- ID du travail.
- Priorité des travaux telle qu'attribuée via l'option `-p` de `qsub(1)` ou `qalter(1)` déterminant l'ordre de la liste des travaux en attente.
- Nom du travail.
- Nom d'utilisateur du propriétaire du travail.
- Statut du travail, selon le cas : `d`(supprimé), `t`(ransfert en cours), `r`(exécution en cours), `R`(edémarré), `s`(interrompu), `S`(interrompu), `T`(seuil), `w`(en attente) ou `h`(suspendu).

L'état `d`(supprimé) indique que `qdel(1)` a été utilisé pour lancer la suppression d'un travail. Les états `t`(ransfert en cours) et `r`(exécution en cours) indiquent qu'un travail est sur le point d'être exécuté ou est déjà en cours d'exécution, tandis que les états `s`(interrompu), `S`(interrompu) et `T`(seuil) indiquent qu'un travail en cours d'exécution a été interrompu. L'état `s`(interrompu) est causé par l'interruption du travail via la commande `qmod(1)`, l'état `S`(interrompu) indique que la file contenant le travail a été interrompue et que le travail l'est par conséquent et l'état `T`(seuil) indique qu'au moins un seuil d'interruption de la file correspondante a été dépassé (voir `queue_conf(5)`) et que le travail a été interrompu suite à cela. L'état `R`(edémarré) indique que le travail a été redémarré. Cela peut être dû à la migration du travail ou à l'une des raisons décrites dans la section `-r` de la commande `qsub(1)`.

Les états `w`(en attente) et `h`(suspendu) n'apparaissent que pour les travaux en attente. L'état `h`(suspendu) indique qu'un travail ne peut couramment pas être exécuté car l'état suspendu lui a été attribué via l'option `qhold(1)`, `qalter(1)` ou `qsub(1) -h` ou parce que ce travail attend la fin des travaux pour lesquels des dépendances de travaux ont été attribuées au travail via l'option `-hold_jid` de `qsub(1)` ou `qalter(1)`.

- Heure et date de soumission ou de début du travail.
- File d'attente à laquelle le travail a été attribué (pour les travaux en cours ou interrompus uniquement).
- Fonction des travaux en cours (MASTER ou SLAVE – cette dernière uniquement pour les travaux parallèles).
- ID de tâche de travail de matrice. Est vide pour les travaux non-matriciels. Voir l'option `-t` de `qsub(1)` et `-g` ci-dessus pour davantage d'informations.
Si l'option `-t` est fournie, chaque ligne de statut de travail contient également les éléments suivants.
- ID de tâche parallèle (ne confondez pas les tâches parallèles avec les tâches de travail de matrice).
- Statut de la tâche parallèle, selon le cas : `r`(exécution en cours), `R`(edémarré), `s`(interrompu), `S`(interrompu), `T`(seuil), `w`(en attente), `h`(suspendu) ou `x`(quitté).

- Utilisation UC, de la mémoire et des E/S (Sun Grid Engine, Enterprise Edition uniquement).
- Statut de sortie de la tâche parallèle.
- Code et message d'erreur de la tâche parallèle.

Format complet (avec `-f` et `-F`)

À la suite de la ligne d'en-tête, des sections séparées par des lignes horizontales s'affichent pour les différentes files. Pour chaque file, les informations imprimées sont les suivantes :

- Nom de la file.
- Type de la file, selon le cas : **B**(lot), **I**(nteractif), **C**(point de reprise), **P**(arallèle), **T**(ransfert) ou une combinaison de ces éléments.
- Nombre d'emplacements de travail utilisés et disponibles.
- Charge moyenne de l'hôte de la file.
- Architecture de l'hôte de la file.
- État de la file d'attente : **u** (inconnu) si le `sge_execd(8)` correspondant ne peut pas être contacté, **a** (alarme), **A** (Alarme), **C** (interrompu agenda), **s** (interrompu), **S** (Subordonné), **d** (désactivé), **D** (Désactivé), **E** (Erreur) ou une combinaisons de ces états.

Si l'état est **a**(larne) au moins un des seuils de charge définis dans la liste `load_thresholds` de la configuration de file (voir `queue_conf(5)`) est actuellement dépassé, ce qui empêche la programmation d'autres travaux sur cette file.

L'état **A**(larne) indique en revanche qu'au moins un des seuils d'interruption de la file d'attente (voir `queue_conf(5)`) est actuellement dépassé. Par conséquent, les travaux s'exécutant dans cette file d'attente seront interrompus jusqu'à ce qu'aucun seuil ne soit plus dépassé.

Les états **s**(interrompu) et **d**(ésactivé) peuvent être attribués aux files d'attente et retirés via la commande `qmod(1)`. Interrompre une file d'attente entraîne l'interruption de tous les travaux s'exécutant dans cette file.

Les états **D**(ésactivé) et **C**(interrompu agenda) indiquent que la file d'attente a été désactivée ou interrompue automatiquement via l'utilitaire d'agenda de Sun Grid Engine (voir `calendar_conf(5)`), tandis que l'état **S**(ubordonné) indique, que la file a été interrompue via sa subordination à une autre file (voir `queue_conf(5)` pour plus de détails). Lorsqu'une file est interrompue (quelle que soit la cause de l'interruption) tous les travaux s'exécutant dans cette file sont également interrompus.

Si un état **E**(rreur) s'affiche pour une file, `sge_execd(8)` a été incapable de localiser sur cet hôte l'exécutable `sge_shepherd(8)` afin de démarrer un travail. Veuillez contrôler le fichier journal des erreurs de ce `sge_execd(8)` pour des pistes sur la résolution du problème. Veuillez activer manuellement la file d'attente après cela via l'option `-c` de la commande `qmod(1)`.

Si l'option `-F` a été utilisée, les informations sur la disponibilité des ressources sont imprimées à la suite de la ligne de statut de l'hôte. Pour chaque ressource (pour celle sélectionnée dans un argument d'option de `-F` ou pour toutes les ressources si l'argument d'option a été omis) une unique ligne dont le format est le suivant s'affiche :

- Un spécificateur d'une lettre indiquant si la valeur de disponibilité de la ressource courante a été dominée par au choix :
 - « `g` » – un paramètre de grappe global
 - « `h` » – un total d'hôte
 - « `q` » – une consommation de ressources liée à une file
- Un second spécificateur d'une lettre indiquant la source de la valeur de disponibilité de ressource courante, pouvant être :
 - « `l` » – une valeur de charge rapportée pour la ressource,
 - « `L` » – une valeur de charge pour la ressource après que la mise à l'échelle définie par l'administrateur a été appliquée,
 - « `c` » – la disponibilité dérivée de l'utilitaire des ressources consommables (voir `complexes(5)`)
 - « `v` » – une valeur de configuration de complexes par défaut jamais surécrite par un rapport de charge ou une mise à jour de consommable
 - « `f` » – une définition de disponibilité fixe dérivée d'un attribut de complexe non-consommable ou d'une limite de ressource fixe
- Après un signe deux points, le nom de la ressource sur laquelle les informations s'affichent.
- Après le signe égal, la valeur de disponibilité courante de la ressource.

Les valeurs de disponibilité affichées et les sources dont elles dérivent sont toujours les valeurs minimales de toutes les combinaisons possibles. Par conséquent, par exemple, une ligne de la forme « `qf:h_vmem=4G` » indique qu'une file a couramment une mémoire virtuelle maximale disponible de 4 Go, que cette valeur est une valeur fixe (par exemple une limite de ressource dans la configuration de file) et qu'elle est dominée par la file d'attente, c'est-à-dire qu'au total la mémoire disponible sur l'hôte peut être supérieure mais que la file d'attente n'en autorise pas plus. Au contraire, la ligne « `hl:h_vmem=4G` » indiquerait une limite supérieure de 4 Go pour la mémoire virtuelle disponible, qui serait dérivée d'une valeur de charge couramment rapportée pour l'hôte. Ainsi, la file pourrait autoriser des travaux ayant des exigences de mémoire virtuelle supérieures, mais son hôte n'aurait réellement que 4 Go de disponibles.

Si l'option `-alarm` a été utilisée, les informations sur les ressources qui violent des seuils de charge ou d'interruption s'affichent.

Le format utilisé est le même qu'avec l'option `-F` avec les extensions suivantes.

- La ligne commence par le mot clé `alarm`.
- Le type et la valeur du seuil concerné sont rattachés à la valeur de la ressource.

Après la ligne de statut de la file d'attente (si -f) ou les informations de disponibilité des ressources (si -F), une ligne est imprimée pour chaque travail s'exécutant couramment dans la file. Chaque ligne de statut de travail contient les éléments suivants :

- ID du travail.
- Nom du travail.
- Nom du propriétaire du travail.
- Statut du travail, selon le cas : t(ransfert en cours), r(exécution en cours), R(edémarré), s(interrompu), S(interrompu) ou T(seuil) (voir la section Format réduit pour des informations détaillées).
- Date et heure de début et fonction du travail (MASTER ou SLAVE - n'a un sens que dans le cas d'un travail parallèle).
- Priorité des travaux.

Si l'option -t est fournie, chacune des lignes de statut de travail contient également les éléments suivants :

- ID de la tâche.
- Statut de la tâche, selon le cas : r(exécution en cours), R(edémarré), s(interrompu), S(interrompu), T(seuil) w(en attente), h(suspendu) ou x(quitté) (voir la section Format réduit pour des informations détaillées).
- Utilisation des UC, de la mémoire et des E/S (Sun Grid Engine, Enterprise Edition uniquement).
- Statut de sortie de la tâche.
- Code et message d'échec pour la tâche.

À la suite de la liste des sections de file, il est possible qu'une liste des *travaux en attente* soit imprimée s'il y a des travaux en attente d'être attribués à une file d'attente. Une ligne de statut similaire à celle des travaux en cours s'affiche alors pour chaque travail en attente. Les seules différences sont que le statut de ces travaux est w(en attente) ou h(suspendu), et que la date et l'heure de soumission sont indiquées à la place de la date et l'heure de début et qu'aucune fonction ne s'affiche pour les travaux.

Dans de très rares cas, par exemple si `sgc_qmaster(8)` démarre à partir d'un état inconsistant dans les fichiers de spool de file ou de travail ou si l'option de nettoyage de la file (`-cq`) de `qconf(1)` est utilisée, `qstat` ne peut pas attribuer les travaux aux sections des travaux en cours et des travaux en attente de la sortie. Dans ce cas, une inconsistance de statut (par ex. un travail a le statut en cours mais n'est pas attribué à une file) est détectée. Ces travaux s'impriment dans une section intitulée *travaux avec erreurs* à la fin de la sortie. La section TRAVAUX AVEC ERREURS devrait disparaître au redémarrage de `sgc_qmaster(8)`. Veuillez contacter votre chargé d'assistance Sun Grid Engine si vous êtes un peu perdu relativement à la cause ou aux effets de tels travaux.

Format étendu (avec `-r`)

Si l'option `-r` a été spécifiée avec `qstat`, les informations suivantes s'impriment pour chaque travail affiché (à raison d'une ligne pour chacune des caractéristiques de travail suivantes).

- Les exigences en ressources souples et fixes du travail telles que spécifiées avec l'option `qsub(1) -l`.
- L'environnement parallèle requis plage d'emplacements de file désirée comprise (voir l'option `-pe` de `qsub(1)`).
- L'environnement de point de reprise demandé du travail (voir l'option `qsub(1) -ckpt`).
- En cas de travaux en cours d'exécution, l'environnement parallèle attribué et le nombre d'emplacements de file attribués.

Sortie Sun Grid Engine, Enterprise Edition enrichie (avec `-ext`)

Les rubriques supplémentaires suivantes s'affichent pour chaque travail.

- `project` - Projet auquel le travail est attribué tel que spécifié dans l'option `qsub(1) -P`.
- `department` - Service dont fait partie l'utilisateur (utilisez les options `-sul` et `-su` de `qconf(1)` pour afficher les définitions de service courantes).
- `deadline` - Heure d'initiation limite du travail telle que spécifiée par l'option `qsub(1) -dl`.
- `cpu` - Utilisation UC cumulée du travail.
- `mem` - Utilisation de mémoire cumulée du travail.
- `io` - Utilisation E/S cumulée courante du travail.
- `tckts` - Nombre total de tickets attribués couramment au travail.
- `ovrts` - Nombre de tickets de remplacement tel qu'attribué par l'option `-ot` de `qalter(1)`.
- `otckt` - Portion des tickets de remplacement du nombre total de tickets attribués couramment au travail.
- `dtckt` - Portion des tickets d'initiation limite du nombre total de tickets attribués couramment au travail.
- `ftckt` - Portion des tickets fonctionnels du nombre total de tickets attribués couramment au travail.
- `stckt` - Portion des tickets de partage du nombre total de tickets attribués couramment au travail.
- `share` - Partage du système total auquel le travail a couramment droit.

Variables d'environnement

Le TABLEAU 22 décrit les variables d'environnement associées à `qstat`.

TABLEAU 22 Variables d'environnement de `qstat`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qstat</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement, <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qstat</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `qalter(1)`, `qconf(1)`, `qhold(1)`, `qhost(1)`, `qmod(1)`,
`qsub(1)`, `queue_conf(5)`, `sge_commd(8)`, `sge_execd(8)`, `sge_qmaster(8)`,
`sge_shepherd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

qtssh(1)

Nom

`qtssh - tcsh v6.09` avec exécution à distance transparente au moyen de `qrsh`.

Syntaxe

`qtssh [tcsh options | -ABLR]`

Description

`qtssh` est une extension de la très répandue dérivée de `ssh(1)` : `tcsh`. Elle permet les commandes d'exécution à distance transparentes entrées dans `qtssh` contrôlées via Sun Grid Engine. `qtssh` peut être utilisé en tant qu'interpréteur de commandes interactif ainsi que pour le traitement des scripts shell `tcsh`.

Lorsqu'il est appelé, `qtssh` identifie les commandes qui doivent être exécutées à distance et celles qui ne le doivent pas. Dans cet objectif, les fichiers `<racine_sge>/<cellule>/common/qtssh` et `~/ .qtssh` sont traités. Chacune des lignes de ces fichiers définit une commande devant s'exécuter à distance (voir `qtssh(5)` pour la définition du format de ce fichier). Le fichier `.qtssh` du répertoire de base de l'utilisateur contient la spécification de tâche à distance de l'utilisateur, tandis que le fichier figurant dans le répertoire commun est maintenu par l'administrateur et définit un comportement par défaut à l'échelle de la grappe. Le contenu du fichier `qtssh(5)` de l'administrateur est complètement remplacé dans le cas où il y a une entrée appropriée dans le fichier `qtssh(5)` des utilisateurs. Cela n'est pas possible si un point d'exclamation est rattaché en tant que préfixe au nom de la commande dans le fichier `qtssh` de l'administrateur.

`qtssh` essaye toujours de démarrer les tâches désignées à distance via `qrsh(1)`. Les exceptions à cette règle sont les suivantes.

- Si l'utilisateur entre ces commandes en employant un chemin relatif ou absolu au lieu du simple nom de la commande. Voir `qtssh(5)` pour plus d'informations.

- Si la variable d'environnement `JOB_ID` est définie, `qtssh` assume que l'exécution a déjà lieu à distance dans le cadre d'un travail Sun Grid Engine et exécute donc les tâches localement. Cela évite toute récursivité indésirée mais peut être annulé par l'option de ligne de commande `-R` et la commande intégrée `qrshmode -R` (voir les descriptions correspondantes ci-après).
- Si `qtssh` ne peut pas établir la connexion de Sun Grid Engine au démarrage. Cela permet d'utiliser `qtssh` en tant que shell de connexion sans risquer d'être bloqué lorsqu'aucun service Sun Grid Engine n'est disponible.

`qtssh` peut fonctionner en trois modes différents, qui déterminent si :

- Les tâches sont exécutées à distance.
- L'exécution immédiate ou par lots est demandée.
- La sortie du statut est en mode verbeux ou limitée aux cas où il y a des erreurs.

Ces modes peuvent être contrôlés par les modificateurs de ligne de commande décrits ci-après pendant une invocation de `qtssh` ou dans le cadre de l'exécution de `qtssh` via la commande intégrée, `qrshmode`, comme décrit dans la section « Commandes intégrées », page 84.

Options

Les options listées dans le TABLEAU 23 sont spécifiques de `qtssh`. Reportez-vous à la documentation de `tssh(1)` pour l'explication des autres options.

TABLEAU 23 Options de `qtssh`

Option	Description
-A	Bascule <code>qtssh</code> en mode verbeux causant une sortie de diagnostic en cas d'exécution à distance.
-B	Bascule l'exécution des tâches à distance en mode Lot. Les tâches soumises à Sun Grid Engine seront mises en attente si elles ne peuvent pas démarrer immédiatement. Par conséquent, <code>qtssh</code> risque de se bloquer jusqu'à ce que la tâche en file d'attente puisse être démarrée par Sun Grid Engine. Si ce comportement n'est pas souhaitable pendant une session interactive, il peut être très utile pour l'exécution de scripts shell par l'intermédiaire de <code>qtssh</code> car il évite l'échec des scripts pour cause d'indisponibilité temporaire des ressources pour certaines tâches.
-L	Annule le comportement par défaut d'exécution à distance des commandes. Force toutes les commandes à être exécutées localement même si elles sont contenues dans un des fichiers <code>qtask(5)</code> .
-R	Force l'exécution à distance des commandes même si <code>JOB_ID</code> est défini en tant que variable d'environnement.

Commandes intégrées

Cette section décrit uniquement les commandes intégrées à l'interpréteur de commandes supplémentaires qui ne sont pas disponibles dans la version standard de `tcsh(1)`.

`qrshmode [-ANBILR]`

Sans options, le mode opérationnel courant de `qtcsch` s'affiche. Les options ont l'effet suivant.

- -A – Bascule en mode de sortie verbeux.
- -N – Bascule en mode de sortie non-verbeux.
- -B – Bascule en mode d'exécution par lots.
- -I – Bascule en mode d'exécution immédiate.
- -L – Exécute toujours les commandes localement.
- -R – Exécute les commandes configurées à distance.

Variables d'environnement

Le TABLEAU 24 liste les variables d'environnement associées à `qtcsch`.

TABLEAU 24 Variables d'environnement de `qtcsch`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qtcsch</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement, <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qtcsch</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

- Fichier `qtask` d'utilisateur - `~/ .qtask`
- Fichier `qtask` de grappe - `<racine_sge>/<cellule>/common/qtask`

Voir aussi

`sge_intro(1)`, `qrsh(1)`, `qtask(5)`, ainsi que `tcsh(1)` dans `<racine_sge>/3rd_party/qtcs`

Droits d'auteur

`qtcs` contient des parties de `tcsh` dont les droits d'auteur sont détenus par les dirigeants de l'Université de Californie. Par conséquent, les remarques suivantes s'appliquent : Ce produit inclut des logiciels développés par l'Université de Californie, Berkeley et ses partenaires.

Voir `sge_intro(1)` et les informations fournies dans `<racine_sge>/3rd_party/qtcs` pour l'énoncé des autres droits et autorisations.

`submit(1)`

Nom

`qsub` - Soumet un travail par lots à Sun Grid Engine.

`qsh` - Soumet une session X-windows interactive à Sun Grid Engine.

`qlogin` - Soumet une session de connexion interactive à Sun Grid Engine.

`qrsh` - Soumet une session `rsh` interactive à Sun Grid Engine.

`qalter` - Modifie un travail par lots en attente de Sun Grid Engine.

`qresub` - Soumet une copie d'un travail Sun Grid Engine existant.

Syntaxe

```
qsub [ options ] [ fichierscript | -- [ args_script ] ]  
qsh [ options ] [ -- args_xterm ]  
qlogin [ options ]  
qrsh [ options ] [ commande [ args_commande ] ]  
qalter [ options ] liste_travail/ID_tâches [ -- [ args_script ] ]  
qalter [ options ] -u liste_utilisateurs | -uall [ -- [ args_script ] ]  
qresub [ options ] liste_id_travaux
```

Description

`qsub` soumet les travaux par lots au système de mise en attente Sun Grid Engine. Sun Grid Engine prend en charge les travaux à un et plusieurs noeuds. *fichierscript* contient les commandes devant être exécutées par le travail en utilisant un interpréteur de commandes (par exemple, `sh(1)` ou `csh(1)`). Les arguments du script du travail sont donnés par *args_script*. Les indicateurs de Sun Grid Engine peuvent être entrés à titre d'arguments de `qsub` ou sous la forme d'indicateurs imbriqués dans *fichierscript* si les deux premiers caractères d'une ligne de script sont « `#$` » ou correspondent à la chaîne de préfixe définie avec l'option `-C` décrite ci-après.

`qsh` soumet une session X-windows interactive à Sun Grid Engine. Un `xterm(1)` est activé depuis la machine d'exécution et l'affichage est dirigé soit sur le serveur X indiqué par la variable d'environnement `DISPLAY` soit sur l'élément spécifié avec l'option `-display qsh`. Les travaux interactifs ne sont pas mis en attente si aucune ressource n'est disponible pour leur exécution. Soit ils sont acheminés sur une machine appropriée pour une exécution immédiate, soit l'utilisateur qui soumet le travail est averti par `qsh` que les ressources appropriées pour exécuter le travail ne sont pas disponibles. Les `xterm_args` sont passés à l'exécutable `xterm(1)`.

`qlogin` est similaire à `qsh` en ce qu'il soumet un travail interactif au système de mise en file d'attente. Il n'ouvre toutefois pas de fenêtre `xterm(1)` sur l'écran X mais utilise le terminal courant pour les E/S de l'utilisateur. En général, `qlogin` établit une connexion `telnet(1)` avec l'hôte distant, en utilisant les commandes standard côté client et côté serveur. Ces commandes peuvent être configurées avec les paramètres `qlogin_daemon` (côté serveur, `telnetd` Sun Grid Engine si pas défini, sinon quelque chose comme `/usr/sbin/in.telnetd`) et `qlogin_command` (côté client, `telnet` Sun Grid Engine si pas défini, sinon quelque chose comme `/usr/bin/telnet`) dans les configurations globale et locale de `sge_conf(5)`. La commande côté client est automatiquement paramétrée avec le nom de l'hôte distant et le numéro de port à utiliser pour la connexion (cela se traduit par un appel

similaire à `/usr/bin/telnet my_exec_host 2442`). `qlogin` est appelé exactement comme `qsh` et ses travaux peuvent uniquement être exécutés sur des files INTERACTIVES. Les travaux `qlogin` peuvent uniquement être utilisés si `sgc_execd(8)` est exécuté sous le compte du super-utilisateur.

`qrsh` est similaire à `qlogin` en ce qu'il soumet un travail interactif au système de mise en file d'attente. Il utilise le terminal courant pour les E/S de l'utilisateur. En général, `qrsh` établit une connexion `rsh(1)` avec l'hôte distant. Si aucune commande n'est donnée à `qrsh`, une session `rlogin(1)` est établie. Les commandes côté serveur utilisées peuvent être configurées avec les paramètres `rsh_daemon` et `rlogin_daemon` dans les configurations globale et locale de `sgc_conf(5)`. Un `rshd` ou `rlogind` Sun Grid Engine est utilisé si les paramètres ne sont pas définis, sinon quelque chose comme `/usr/sbin/in.rshd` ou `/usr/sbin/in.rlogind` doit être configuré. Sur le côté client, les paramètres `rsh_command` et `rlogin_command` peuvent être définis dans les paramètres de configuration globale et locale de `sgc_conf(5)`. S'ils ne sont pas définis, les binaires `rsh(1)` et `rlogin(1)` fournis avec Sun Grid Engine sont utilisés. Utilisez la configuration de grappe pour intégrer des mécanismes tels que les utilitaires `ssh` ou `rsh(1)` et `rlogin(1)` fournis avec le système d'exploitation.

Les travaux `qrsh` peuvent uniquement être exécutés dans les files d'attente INTERACTIVES à moins que l'option `-now no` ne soit utilisée (voir ci-dessous). Ils peuvent uniquement être utilisés, si `sgc_execd(8)` est exécuté sous le compte du super-utilisateur.

`qrsh` fournit une fonctionnalité supplémentaire utile pour l'intégration avec des outils interactifs en fournissant un shell ou interpréteur de commandes spécifique. Si la variable d'environnement `QRSH_WRAPPER` est définie lorsque `qrsh` est appelé, l'interpréteur de commandes pointé dessus par `QRSH_WRAPPER` sera exécuté pour exécuter les commandes `qrsh` au lieu du shell de connexion de l'utilisateur ou de tout shell spécifié dans la ligne de la commande `qrsh`.

`qalter` peut être utilisé pour changer les attributs des travaux en attente. Une fois que l'exécution d'un travail est lancée, les changements ne sont plus possibles. Pour les matrices de travail, pour lesquelles une partie des tâches peuvent être en attente et une autre partie en cours d'exécution (voir l'option `-t` ci-après), les modifications avec `qalter` n'affectent que les tâches en attente. `qalter` peut changer la plupart des caractéristiques d'un travail (voir les parties correspondantes dans la section Options ci-après), y compris celles qui avaient été définies par des indicateurs imbriqués dans le fichier de script (voir ci-dessus).

`qresub` vous permet de créer des travaux en copiant des travaux existants en cours ou en attente. Les travaux copiés auront exactement les mêmes attributs que ceux dont ils auront été copiés, mais leur ID de travail sera différent. La seule modification des travaux copiés prise en charge par `qresub` consiste à attribuer un état de suspension avec l'option `-h`. Cela peut être utilisé pour copier un travail puis en changer les attributs via `qalter`.

Pour `qsub`, `qsh`, `qrsh` et `qlogin`, l'administrateur et l'utilisateur peuvent définir des fichiers de requête par défaut (voir `sge_request(5)`) qui peuvent contenir n'importe laquelle des options décrites ci-après. Si une option insérée dans un fichier de requête par défaut est comprise par `qsub` et `qlogin` mais pas par `qsh`, cette option est ignorée en silence si `qsh` est appelé. Vous pouvez donc maintenir des fichiers de requête par défaut partagés pour à la fois `qsub` et `qsh`.

Un fichier de requête par défaut de grappe peut être placé sous `$racine_sge/Scellule_sge/common/sge_request`. Les fichiers de requête par défaut d'utilisateurs sont traités sous les emplacements `$HOME/.sge_request` et `$cwd/.sge_request`. Le fichier de requête par défaut local du répertoire de travail a la priorité absolue, il est suivi du fichier se trouvant dans le répertoire de base puis du fichier global de la grappe. Les arguments d'option, les indicateurs de script imbriqués et les options figurant dans les fichiers de requête par défaut sont traités dans l'ordre suivant.

1. De gauche à droite dans la ligne de script.
2. De gauche à droite dans les fichiers de requête par défaut.
3. De haut en bas au sein du fichier de script (`qsub` uniquement).
4. De haut en bas au sein des fichiers de requête par défaut.
5. De gauche à droite sur la ligne de commande.

Autrement dit, la ligne de commande peut être utilisée pour remplacer les indicateurs imbriqués et les paramètres de requête par défaut. Les indicateurs imbriqués, quant à eux, remplaceront les paramètres par défaut.

Remarque – L'option `-clear` peut être utilisée pour éliminer des paramètres précédents à tout moment dans un fichier de requête par défaut, dans les indicateurs de script imbriqués ou dans une option de ligne de commande. Sachez toutefois qu'elle n'est pas disponible avec `qalter`.

Les options décrites dans le TABLEAU 25 peuvent être demandées, au choix, fixes ou souples. Par défaut, toutes les requêtes sont considérées comme fixes jusqu'à ce que l'option `-soft` (voir ci-dessous) soit rencontrée. Le statut fixe/souple reste en vigueur jusqu'à ce que sa contrepartie soit rencontrée de nouveau. Si toutes les requêtes fixes d'un travail ne peuvent pas être satisfaites, ce travail ne sera pas programmé. Les travaux qui ne peuvent pas être exécutés pour le moment restent en attente.

Options

Le TABLEAU 25 liste et décrit les options associées aux différentes commandes de soumission.

TABLEAU 25 Options de la commande `submit`

Option	Description
<code>-@ fichieroptions</code>	Force <code>qsub</code> , <code>qrsh</code> , <code>qsh</code> ou <code>qlogin</code> à utiliser les options contenues dans <i>fichieroptions</i> . Le fichier indiqué peut contenir toutes les options valides. Les lignes de commentaires commencent par le signe #.
<code>-a date_heure</code>	Disponible uniquement pour <code>qsub</code> , <code>qrsh</code> , <code>qsh</code> , <code>qlogin</code> et <code>qalter</code> . Définit ou redéfinit l'heure et la date auxquelles un travail est disponible pour l'exécution. <i>Date_heure</i> adopte le format <code>[[SS]]AA]MMJJhhmm.[ss]</code> , où : <i>SS</i> indique le siècle par 2 chiffres. <i>AA</i> indique l'année par 2 chiffres. <i>MM</i> indique le mois par 2 chiffres. <i>JJ</i> indique le jour par 2 chiffres. <i>hh</i> indique l'heure par 2 chiffres. <i>mm</i> indique les minutes par 2 chiffres. <i>ss</i> indique les secondes par 2 chiffres (valeur par défaut : 00). Si l'un quelconque des champs de date optionnels est omis, la valeur correspondante de la date courante est utilisée. L'utilisation de cette option peut être à l'origine de résultats imprévus si les horloges des hôtes du groupe Sun Grid Engine ne sont pas synchronisées. Par ailleurs, le comportement correct de cette option dépend considérablement de la sélection du fuseau horaire approprié, par ex. dans la variable d'environnement <code>TZ</code> (voir <code>date(1)</code> pour plus de détails), lorsque les démons Sun Grid Engine <code>sgc_qmaster(8)</code> et <code>sgc_execd(8)</code> sont appelés. <code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-ac</code> <i>variable</i>[=<i>valeur</i>],...</p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Ajoute la ou les paires <i>nom/valeur</i> indiquées au contexte du travail. La <i>valeur</i> peut être omise. Sun Grid Engine ajoute l'argument indiqué à la fin de la liste des variables de contexte du travail. Des options <code>-ac</code>, <code>-dc</code> et <code>-sc</code> multiples peuvent être indiquées. L'ordre est important. <code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>
<p><code>-A</code> <i>chaîne_compte</i></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Identifie le compte auquel la consommation de ressources du travail doit être chargée. La <i>chaîne_compte</i> peut être toute chaîne de caractères alphanumériques ASCII arbitraire à condition de ne pas contenir d'espaces ni de caractères de séparation. Le trait de soulignement « <code>_</code> » n'est pas considéré comme un séparateur. En l'absence de ce paramètre, Sun Grid Engine placera la chaîne de compte par défaut <code>sgc</code> dans l'enregistrement de comptabilité du travail.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>

TABEAU 25 Options de la commande `submit` (suite)

Option	Description
<code>-c</code> <i>spécificateur_occasion</i>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Définit ou redéfinit si l'écriture de points de reprise doit être effectuée pour le travail et si oui, dans quelles circonstances. La spécification des occasions d'écriture de points de reprise avec cette option l'emporte sur les définitions du paramètre <i>when</i> dans l'environnement de point de reprise (voir <code>checkpoint(5)</code>) référencé par le modificateur <code>qsub -ckpt</code>. Les valeurs possibles de <i>spécificateur_occasion</i> sont les suivantes :</p> <ul style="list-style-type: none"> n – Pas d'écriture de points de reprise. s – Écriture de points de reprise en cas d'arrêt du serveur. m – Écriture de points de reprise à des intervalles UC minimaux. x – Écriture de points de reprise à l'interruption du travail. <p><i><intervalle></i> – Écriture de points de reprise dans l'intervalle spécifié.</p> <p>L'intervalle UC minimal est défini dans la configuration de file (voir <code>queue_conf(5)</code> pour plus de détails). <i><intervalle></i> doit être spécifié dans le format <code>hh:mm:ss</code>. Le maximum de <i><intervalle></i> et de l'intervalle UC minimal est utilisé si <i><intervalle></i> est spécifié. Cela est fait pour éviter la surcharge d'une machine pour cause de génération trop fréquente de points de reprise.</p>
<code>-ckpt</code> <i>nom_ptreprise</i>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Sélectionne l'environnement de point de reprise (voir <code>checkpoint(5)</code>) à utiliser pour l'écriture de points de reprise du travail. Indique également que le travail est un travail de point de reprise.</p>
<code>-clear</code>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code> et <code>qlogin</code>.</p> <p>Entraîne la réinitialisation de tous les éléments du travail à leurs statuts par défaut initiaux respectifs avant d'appliquer d'éventuelles modifications (le cas échéant) apparaissant dans cette commande spécifique.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<code>-cwd</code>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Exécute le travail depuis le répertoire de travail courant. Ce modificateur activera l'utilitaire d'alias de chemins de Sun Grid Engine, si les fichiers de configuration correspondants sont présents (voir <code>sgc_aliases(5)</code>).</p> <p>En cas de <code>qalter</code>, la définition préalable du répertoire de travail courant sera surécrite, si <code>qalter</code> est exécuté depuis un autre répertoire que le <code>qsub</code> ou <code>qalter</code> précédent.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<code>-C chaîne_préfixe</code>	<p>Disponible uniquement pour <code>qsub</code>.</p> <p><i>chaîne_préfixe</i> définit le préfixe qui annonce une pseudo-instruction à <code>qsub</code> dans le fichier de script du travail. Le préfixe n'est pas un attribut du travail, mais affecte le comportement de <code>qsub</code>. Si l'option <code>-C</code> est présentée avec la valeur du préfixe de pseudo-instruction sous la forme d'une chaîne nulle, <code>qsub</code> ne balayera pas le fichier de script.</p> <p>Le préfixe de pseudo-instruction se compose de deux caractères ASCII qui, lorsqu'ils apparaissent dans les deux premiers octets d'une ligne de script, indiquent que ce qui suit est une commande Sun Grid Engine (valeur par défaut : <code>#§</code>).</p> <p>L'utilisateur doit être conscient que changer le premier de ces caractères peut avoir des effets secondaires imprévus. Si le fichier de script contient un caractère autre que <code>#</code> dans la première position d'octet de la ligne, le processeur shell pour le travail refusera la ligne et risque d'abandonner le travail.</p> <p>Si l'option <code>-C</code> est présente dans le fichier de script, elle est ignorée.</p>

TABEAU 25 Options de la commande `submit` (*suite*)

Option	Description
-dc <i>variable...</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Supprime la ou les variables indiquées du contexte du travail. Plusieurs options <code>-ac</code>, <code>-dc</code> et <code>-sc</code> peuvent être indiquées. L'ordre est important.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>
-display <i>spécificateur_affichage</i>	<p>Disponible uniquement pour <code>qsh</code>.</p> <p>Ordonne à <code>xterm(1)</code> d'utiliser <i>spécificateur_affichage</i> afin de contacter le serveur X.</p>
-dl <i>date_heure</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>. Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle spécifie l'heure d'initiation limite en utilisant le format <code>[[SS]AA]JJhmm[.ss]</code> (voir l'option <code>-a</code> plus haut). L'heure d'initiation limite est l'heure à laquelle un travail avec limite doit atteindre la priorité maximale pour pouvoir être terminé à la date limite indiquée. Avant l'heure d'initiation limite la priorité d'un travail avec limite augmentera progressivement jusqu'à atteindre le maximum configuré par l'administrateur Sun Grid Engine.</p> <p>Cette option n'est applicable que pour les utilisateurs autorisés à soumettre des travaux avec limite.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-e [nomhôte:]chemin...</code></p>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Définit ou redéfinit le chemin utilisé pour le flux d'erreur standard du travail. Si le <i>chemin</i> constitue un chemin absolu, l'attribut de chemin d'erreur du travail est mis sur cette valeur, <i>nomhôte</i> compris. Si le chemin est relatif, Sun Grid Engine prolonge <i>chemin</i> avec le chemin du répertoire de travail courant si le modificateur <code>-cwd</code> (voir ci-dessus) est également spécifié, avec le chemin du répertoire de base sinon. Si <i>nomhôte</i> est présent, le flux d'erreur standard sera placé sous l'emplacement correspondant si le travail s'exécute sur l'hôte spécifié.</p> <p>Par défaut, le nom du fichier d'erreur standard est de la forme <i>nom_travail.eid_travail</i> et <i>nom_travail.eid_travail.id_tâche</i> pour les tâches de travail de matriciel (voir l'option <code>-t</code> ci-après). Si <i>chemin</i> est un répertoire, le flux d'erreur standard du travail sera mis dans ce répertoire sous le nom de fichier par défaut. Si le chemin contient certaines pseudo-variables d'environnement, leur valeur sera développée à l'exécution du travail et utilisée pour constituer le chemin de flux d'erreur standard. Les pseudo-variables d'environnement actuellement prises en charge sont les suivantes :</p> <p><i>\$HOME</i> – Répertoire de base sur la machine d'exécution</p> <p><i>\$USER</i> – ID d'utilisateur du propriétaire du travail</p> <p><i>\$JOB_ID</i> – ID du travail courant</p> <p><i>\$JOB_NAME</i> – Nom du travail courant (voir option <code>-N</code>)</p> <p><i>\$HOSTNAME</i> – Nom de l'hôte d'exécution</p> <p><i>\$TASK_ID</i> – Indice numérique de la tâche de travail de matrice</p> <p>Il est possible d'utiliser à la place de <i>\$HOME</i> le tilde « ~ » dans <code>csh(1)</code> ou <code>ksh(1)</code>.</p> <p>Le signe « ~ » marche aussi en combinaison avec les noms d'utilisateur, « ~<utilisateur> » arrive au répertoire de base de <utilisateur>. Utiliser un autre ID d'utilisateur que celui du propriétaire du travail requiert naturellement les droits d'accès correspondants.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
-hard	<p data-bbox="815 244 1310 296">Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p data-bbox="815 305 1310 439">Cette option signifie que toutes les exigences en ressources qui suivent dans la ligne de commande sont des exigences fixes qui doivent être pleinement satisfaites pour qu'un travail puisse être programmé.</p> <p data-bbox="815 447 1310 925">Lorsque Sun Grid Engine balaye la ligne de commande et le fichier de script à la recherche des options et paramètres de Sun Grid Engine, il construit une liste des ressources requises par un travail. Toutes les requêtes de ressources de ce type sont considérées comme absolument essentielles pour que le travail commence. Si l'option <code>-soft</code> (voir ci-dessous) est rencontrée pendant le balayage alors toutes les exigences de ressources qui suivent sont des « exigences souples » ou « appréciables mais pas essentielles ». Si l'indicateur <code>-hard</code> est rencontré à un moment ultérieur du balayage, toutes les requêtes de ressources successives seront de nouveau « essentielles ». Les options <code>-hard</code> et <code>-soft</code> agissent en effet comme des « bascules » pendant le balayage.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-h</code> <code>-h {u s o n U O S}...</code></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code>, <code>qalter</code> et <code>qresub</code>.</p> <p>Liste des suspensions à mettre sur le travail.</p> <p><code>u</code> – Indique une suspension d'utilisateur. <code>s</code> – Indique une suspension de système. <code>o</code> – Indique une suspension d'opérateur. <code>n</code> – Indique l'absence de suspension.</p> <p>Tant qu'une suspension autre que « <code>n</code> » est attribuée au travail, ce dernier n'est pas disponible pour l'exécution. Les suspensions peuvent être retirées au moyen de <code>qalter</code> et <code>qrls(1)</code>. Dans le cas de <code>qalter</code>, cela est pris en charge pour les spécificateurs d'options supplémentaires suivants pour le modificateur <code>-h</code> :</p> <p><code>U</code> – Retire une suspension d'utilisateur. <code>S</code> – Retire e une suspension de système. <code>O</code> – Retire une suspension d'opérateur.</p> <p>Les gestionnaires Sun Grid Engine peuvent attribuer et retirer tous les types de suspensions, les opérateurs Sun Grid Engine peuvent attribuer et retirer les suspensions d'utilisateur et d'opérateur tandis que les utilisateurs ne peuvent qu'attribuer ou retirer les suspensions d'utilisateur.</p> <p>Dans le cas de <code>qsub</code>, seules des suspensions d'utilisateur peuvent être placées sur un travail et par conséquent seule la première forme de l'option avec le modificateur <code>-h</code> seul est admise. Au contraire, <code>qalter</code> requiert la seconde forme décrite plus haut.</p> <p>Une autre façon d'attribuer une suspension est fournie par l'utilitaire <code>qhold(1)</code>.</p> <p>Si le travail est un travail de matrice (voir l'option <code>-t</code> ci-après), toutes les tâches spécifiées via <code>-t</code> sont affectées simultanément par l'opération <code>-h</code>.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<p><code>-help</code></p>	<p>Imprime la liste de toutes les options.</p>

TABEAU 25 Options de la commande `submit` (*suite*)

Option	Description
-hold_jid <i>[id_travail/nom_travail],...</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Définit ou redéfinit la liste des dépendances de travaux du travail soumis. Une référence par nom de travail n'est acceptée que si le travail référencé appartient au même utilisateur que le travail y faisant référence. Le travail soumis n'est pas disponible pour l'exécution tant que tous les travaux figurant dans la liste de <i>id_travail</i> et/ou <i>nom_travail</i> séparée par des virgules n'ont pas été accomplis avec succès.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
-inherit	<p>Disponible uniquement pour <code>qrsh</code> et <code>qmake(1)</code>.</p> <p><code>qrsh</code> vous permet de démarrer une tâche dans un travail parallèle déjà programmé. L'option <code>-inherit</code> indique à <code>qrsh</code> de lire l'ID du travail dans la variable d'environnement <code>JOB_ID</code> et de démarrer la commande spécifiée au titre de tâche de ce travail. Veuillez noter que dans ce cas, le nom d'hôte de l'hôte où la commande devra être exécutée, doit précéder la commande à exécuter ; la syntaxe devient <code>qrsh-inherit [autres options] nomhôte commande [args_commande]</code></p> <p>Veuillez également noter qu'en combinaison avec <code>-inherit</code>, la plupart des autres options de la ligne de commande seront ignorées. Seules les options <code>-verbose</code>, <code>-v</code> et <code>-V</code> seront interprétées. En remplacement de l'option <code>-cwd</code>, utilisez <code>-v PWD</code>.</p> <p>En général, une tâche devrait avoir le même environnement (répertoire de travail courant compris) que le travail correspondant, de sorte que spécifier l'option <code>-v</code> devrait être adapté à la plupart des applications.</p> <p>Si dans votre système le port de commande n'est pas configuré en tant que service, mais via la variable d'environnement <code>COMMD_PORT</code>, assurez-vous que cette variable est définie dans l'environnement lorsque vous appelez <code>qrsh</code> ou <code>qmake</code> avec l'option <code>-inherit</code>. Si vous appelez <code>qrsh</code> ou <code>qmake</code> avec l'option <code>-inherit</code> depuis un script de travail, exportez <code>COMMD_PORT</code> avec l'option <code>submit</code> ou le commentaire spécial : <code>-v COMMD_PORT</code></p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-j y n</code></p>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Spécifie si le flux d'erreurs standard du travail est ou non fusionné au flux de sortie standard. Si les deux options <code>-j y</code> et <code>-e</code> sont présentes, Sun Grid Engine définit, mais ignore l'attribut chemin-erreur. <code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<p><code>-l ressource=valeur,...</code></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>. Lance le travail dans une file d'attente Sun Grid Engine qui satisfait la liste des requêtes de ressources. Dans le cas de <code>alter</code>, la définition précédente est remplacée par celle spécifiée. <code>complex(5)</code> décrit comment une liste de ressources disponibles et les spécificateurs de valeur associés peuvent être obtenus. Il peut y avoir plusieurs modificateurs <code>-l</code> dans une même commande. Vous pouvez demander à plusieurs options <code>-l</code> d'être fixes ou souples dans la même ligne de commande. Dans le cas d'un travail série, plusieurs modificateurs <code>-l</code> précisent la définition pour la file d'attente recherchée. <code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>

TABEAU 25 Options de la commande `submit` (suite)

Option	Description
-m b e a s n,...	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Définit ou redéfinit dans quelles circonstances un e-mail est envoyé au propriétaire du travail ou aux utilisateurs définis par l'option <code>-M</code> décrite ci-après. Les arguments d'option ont la signification suivante :</p> <p>b – Un message est envoyé au début du travail. e – Un message est envoyé à la fin du travail. a – Un message est envoyé lors de l'abandon ou de la reprogrammation du travail. s – Un message est envoyé à l'interruption du travail. n – Aucun message n'est envoyé.</p> <p>Actuellement, aucun message n'est envoyé lorsqu'un travail est interrompu.</p> <p>Pour <code>qsh</code> et <code>qlogin</code>, l'envoi d'un message au début ou à la fin du travail est supprimé lorsque cette option est rencontrée dans un fichier de requête par défaut.</p> <p><code>qalter</code> permet de changer les arguments d'option b, e et a même pendant l'exécution du travail. La modification de l'argument d'option b ne deviendra toutefois effective qu'après un redémarrage ou la migration du travail.</p>
-M <i>utilisateur[@hôte],...</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Définit ou redéfinit la liste des utilisateurs auxquels le serveur qui exécute le travail doit envoyer un message s'il est configuré pour. La valeur par défaut est le propriétaire du travail sur l'hôte dont il émane.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<code>-masterq file,...</code>	<p>Disponible uniquement pour <code>qsub</code>, <code>qsh</code>, <code>qrsh</code>, <code>qlogin</code> et <code>qalter</code>. N'a de sens que pour les travaux parallèles, c.-à-d. avec l'option <code>-pe</code>.</p> <p>Définit ou redéfinit une liste de files d'attente qui peuvent être utilisées pour devenir la soi-disant <i>file maîtresse</i> de ce travail parallèle. La <i>file maîtresse</i> est définie comme étant la file où le travail parallèle est démarré. Les autres files d'attente auxquelles le travail parallèle distribue des tâches sont les <i>files d'attente esclaves</i>. Un travail parallèle a une unique <i>file maîtresse</i>.</p> <p>Ce paramètre a toutes les propriétés d'une requête de ressources et sera fusionné avec les exigences dérivées de l'option <code>-l</code> décrite ci-dessus.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<code>-notify</code>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Cet indicateur, lorsqu'il est défini, oblige Sun Grid Engine à envoyer des signaux « d'avertissement » à un travail en cours d'exécution avant d'envoyer les signaux proprement dits. Si un signal <code>SIGSTOP</code> est en attente, le travail recevra un signal <code>SIGUSR1</code> plusieurs secondes avant <code>SIGSTOP</code>. Si un signal <code>SIGKILL</code> est en attente, le travail recevra <code>SIGUSR2</code> plusieurs secondes avant <code>SIGKILL</code>. La durée du délai est contrôlée par le paramètre <code>notify</code> dans chaque configuration de file d'attente (voir <code>queue_conf(5)</code>).</p> <p>Le système d'exploitation Linux « utilise mal » les signaux d'utilisateur <code>SIGUSR1</code> et <code>SIGUSR2</code> dans sa mise en oeuvre des threads Posix. Il vous convient de ne pas utiliser l'option <code>-notify</code> si vous exécutez des applications threadées dans vos travaux sous Linux.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>

TABEAU 25 Options de la commande `submit` (*suite*)

Option	Description
-now <i>y[es] n[0]</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code> et <code>qlogin</code>.</p> <p>-now <i>y</i> essaye de démarrer le travail immédiatement ou pas du tout. La commande retourne 0 en cas de réussite ou 1 en cas d'échec (idem si le travail n'a pas pu être immédiatement programmé). -now <i>y</i> est la valeur par défaut pour <code>qsh</code>, <code>qlogin</code> et <code>qrsh</code>. Avec l'option -now <i>n</i>, le travail sera mis dans la file en attente s'il ne peut pas être exécuté immédiatement.</p> <p>-now <i>n</i> est la valeur par défaut pour <code>qsub</code>.</p>
-N <i>nom</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Nom du travail. Ce nom peut être tout ensemble de caractères imprimables commençant par un caractère alphabétique.</p> <p>Si l'option -N n'est pas présente, Sun Grid Engine attribue le nom du script du travail au travail après avoir retiré tout chemin de répertoire du nom du script. Si le script est lu depuis l'entrée standard, le nom du travail est par défaut <code>STDIN</code>.</p> <p>Dans le cas de <code>qsh</code> ou <code>qlogin</code> — et si l'option -N est absente — la chaîne, <code>INTERACT</code>, est attribuée au travail.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<code>-noshell</code>	<p>Disponible uniquement pour <code>qrsh</code> avec une ligne de commande. Ne démarre pas la ligne de commande donnée à <code>qrsh</code> dans un shell de connexion d'utilisateur, mais l'exécute sans le shell d'ajustement.</p> <p>Cette option peut être utilisée pour accélérer l'exécution car la surcharge due par exemple au démarrage du shell et à la recherche des fichiers de ressources du shell est évitée.</p> <p>Cette option peut uniquement être utilisée si aucune analyse de ligne de commande spécifique du shell n'est requise. Si la ligne de commande contient de la syntaxe de shell comme le remplacement d'une variable d'environnement ou une référence (arrière), un shell doit être démarré. Dans ce cas, soit n'utilisez pas l'option <code>-noshell</code> soit incluez l'appel du shell dans la ligne de commande.</p> <p>Exemple :</p> <pre>qrsh echo '\$HOSTNAME' A</pre> <p>Autre solution d'appel avec l'option <code>-noshell</code> :</p> <pre>qrsh -noshell /bin/tcsh -f -c 'echo \$HOSTNAME'</pre>
<code>-nostdin</code>	<p>Disponible uniquement pour <code>qrsh</code>.</p> <p>Supprimer le flux d'entrée <code>STDIN</code> - <code>qrsh</code> transmettra l'option <code>-n</code> à la commande <code>rsh(1)</code>. Cela est particulièrement utile, si plusieurs tâches sont exécutées en parallèle en utilisant <code>qrsh</code> ; par ex., dans un processus <code>make(1)</code>. Le processus qui obtiendra l'entrée sera indéfini.</p>
<code>-o [nomhôte:]chemin,...</code>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>.</p> <p>Chemin utilisé pour le flux de sortie standard du travail. Le <i>chemin</i> est géré comme décrit dans l'option <code>-e</code> pour le flux d'erreurs standard.</p> <p>Par défaut, le nom de fichier pour la sortie standard est de la forme <i>nom_travail.oid_travail</i> et <i>nom_travail.oid_travail.id_tâche</i> pour les tâches de travail de matrice (voir l'option <code>-t</code> ci-après).</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>

TABEAU 25 Options de la commande `submit` (suite)

Option	Description
-ot <i>tickets_replacement</i>	<p>Disponible uniquement pour <code>qalter</code>. Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle change le nombre de tickets de remplacement pour le travail spécifié. Privilèges de gestionnaire/opérateur requis.</p>
-P <i>nom_projet</i>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>. Cette option est uniquement prise en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible pour les systèmes Sun Grid Engine.</p> <p>Elle spécifie le projet auquel ce travail est attribué. L'administrateur doit donner aux utilisateurs individuels la permission de soumettre des travaux à un projet spécifique (voir l'option <code>-aprj</code> de <code>qconf(1)</code>).</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-p</code> <i>priorité</i></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Définit ou redéfinit la priorité du travail par rapport à d'autres travaux. La priorité est un entier compris entre -1023 et 1024. La valeur de priorité par défaut pour les travaux est 0.</p> <p>Dans un système Sun Grid Engine, les utilisateurs peuvent uniquement diminuer la priorité de leurs travaux. Les gestionnaires et les administrateurs de Sun Grid Engine peuvent également augmenter la priorité des travaux. Si un travail en attente a une priorité supérieure, il pourra plus rapidement être acheminé par le programmeur de Sun Grid Engine. La priorité des travaux n'a pas d'effet sur les travaux en cours d'exécution dans Sun Grid Engine.</p> <p>Dans Sun Grid Engine, Enterprise Edition, la priorité des travaux influe sur la politique d'arborescence des partages et la politique fonctionnelle. Elle n'a pas d'effet sur les politiques d'initiation limite et de remplacement (voir <code>share_tree(5)</code>, <code>sched_conf(5)</code> et le <i>Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3</i> pour plus d'informations sur les politiques de gestion de ressources prises en charge par Sun Grid Engine, Enterprise Edition).</p> <p>Dans le cas de la politique d'arborescence des partages, les utilisateurs peuvent distribuer les tickets auxquels ils ont droit entre leurs travaux en utilisant différentes priorités attribuées via <code>-p</code>. Si tous les travaux ont la même valeur de priorité, les tickets sont distribués de façon équitable. Sinon, les travaux reçoivent les tickets en fonction des différentes priorités. Dans ce dernier cas, les priorités sont traitées comme un niveau supplémentaire dans l'arborescence des partages. En connexion avec la politique fonctionnelle, la priorité peut être utilisée pour pondérer les travaux au sein de la catégorie des travaux fonctionnels. Là encore les tickets sont distribués en fonction des priorités, sous la forme d'un niveau de distribution de partages virtuel sous la catégorie des travaux fonctionnels. Si les deux politiques (d'arborescence des partages et fonctionnelle) sont actives, les priorités des travaux auront un effet dans ces deux politiques et les tickets indépendamment dérivés dans chacune d'elles sont ajoutés au nombre total de tickets de chaque travail.</p>

TABEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-pe environnement_parallèle n[-[m]] / [-]m,...</code></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Environnement de programmation parallèle (PE) à instancier. Le descripteur de plage qui suit le nom du PE spécifie le nombre des processus parallèles à exécuter. Sun Grid Engine allouera les ressources appropriées en fonction de leur disponibilité. La page de manuel <code>sg_e_pe(5)</code> contient des informations sur la définition des PE et sur comment obtenir la liste des PE couramment valides.</p> <p>Vous pouvez spécifier le nom du PE avec le caractère joker, *, cas dans lequel par exemple la requête <code>pvm*</code> trouvera tous les environnements parallèles dont le nom commence par la chaîne <code>pvm</code>.</p> <p>La spécification de la plage est une liste d'expressions de plage de la forme <code>n-m</code> (n et m doivent être des entiers positifs non-nuls), où m est une abréviation de <code>m-m</code>, <code>-m</code> une forme abrégée de <code>1-m</code> et <code>n-</code> est une abréviation de <code>n-infinity</code>. La spécification de la plage est traitée comme suit : le plus grand nombre de files d'attente demandé est contrôlé en premier. Si suffisamment de files d'attente satisfaisant la liste des attributs spécifiés sont disponibles, elles sont toutes allouées. Le nombre de files d'attente immédiatement inférieur est ensuite contrôlé et ainsi de suite.</p> <p>Les options <code>-l</code> supplémentaires présentes limitent le jeu des files d'attente adaptées pour le travail parallèle.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<p><code>-q file,...</code></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Définit ou redéfinit la liste des files d'attente qui peuvent être utilisées pour exécuter ce travail. Ce paramètre a toutes les propriétés d'une requête de ressources et sera fusionné avec les exigences dérivées de l'option <code>-l</code> décrite ci-dessus.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-r y n</code></p>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Identifie la capacité d'un travail à être ou non réexécuté. Si la valeur de <code>-r</code> est <code>y</code>, réexécutez le travail s'il a été abandonné sans l'obtention d'un état de sortie adéquat (cela est en général le cas si le noeud sur lequel le travail est exécuté se plante). Si <code>-r</code> est <code>n</code>, n'exécutez en aucune circonstance de nouveau le travail. Les travaux interactifs soumis avec <code>qsh</code> ou <code>qlogin</code> ne sont pas réexécutables. <code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>
<p><code>-sc variable[=valeur],...</code></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qsh</code>, <code>qrsh</code>, <code>qlogin</code> et <code>qalter</code>. Définit les paires <i>nom/valeur</i> en tant que contexte du travail. La <i>valeur</i> peut être omise. Sun Grid Engine remplace le contexte défini au préalable du travail par celui donné en tant qu'argument. Plusieurs options <code>-ac</code>, <code>-dc</code> et <code>-sc</code> peuvent être indiquées. L'ordre est important. Les contextes constituent un moyen de joindre et supprimer des méta-informations à/d'un travail. Les variables de contexte ne sont pas transmises au contexte d'exécution du travail dans son environnement. <code>qalter</code> permet de changer cette option même pendant l'exécution du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
-soft	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Elle signifie que toutes les exigences de ressources qui suivent dans la ligne de commande sont des exigences souples et doivent être satisfaites en fonction de la disponibilité.</p> <p>Lorsque Sun Grid Engine balaye la ligne de commande et le fichier de script à la recherche des options et paramètres Sun Grid Engine, il construit une liste des ressources requises par un travail. Toutes les requêtes de ressources de ce type sont considérées comme absolument essentielles pour que le travail commence. Si l'option <code>-soft</code> (voir ci-dessous) est rencontrée pendant le balayage alors toutes les exigences de ressources qui suivent sont des « exigences souples » pour l'exécution ou « appréciables mais pas essentielles ». Si l'indicateur <code>-hard</code> est rencontré à un moment ultérieur du balayage, toutes les requêtes de ressources successives sont de nouveau « essentielles ». Les options <code>-hard</code> et <code>-soft</code> agissent en effet comme des « bascules » pendant le balayage.</p>
-S [hôte:]chemin,...	<p>Disponible uniquement pour <code>qsub</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Spécifie le shell d'interprétation pour le travail. Seul un composant <i>chemin</i> sans spécificateur <i>hôte</i> est valide et un seul chemin est autorisé pour un hôte donné. Les chemins de shell avec des attributions d'hôte définissent le shell d'interprétation pour le travail si l'hôte est l'hôte d'exécution. Le chemin de shell sans spécification d'hôte est utilisé si l'hôte d'exécution ne correspond à aucun des hôtes de la liste.</p> <p>En sus, le <i>chemin</i> peut être construit avec des pseudo-variables d'environnement comme décrit pour l'option <code>-e</code> ci-dessus.</p> <p>Dans le cas de <code>qsh</code>, le chemin de shell spécifié est utilisé pour exécuter l'interpréteur de commande correspondant dans le <code>xterm(1)</code> (via son option <code>-e</code>) démarré au nom du travail interactif.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
-t n[-m[:s]]	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Soumet un <i>travail dit de matrice</i> ; c'est-à-dire une matrice de tâches identiques qui se différencient uniquement par un indice numérique et sont traitées par Sun Grid Engine pratiquement comme une série de travaux. L'argument d'option de <code>-t</code> spécifie le nombre de tâches de travail de matrice et l'indice associé à ces tâches. Les indices numériques seront exportés vers les tâches de travail via la variable d'environnement <code>SGE_TASK_ID</code>.</p> <p>La plage d'ID de tâche spécifiée dans l'argument d'option peut être un simple chiffre, une plage simple de la forme <code>n-m</code> ou une plage avec une valeur de pas. Ainsi, la plage d'ID de tâches spécifiée par <code>2-10:2</code> correspond aux indices d'ID de tâche 2, 4, 6, 8 et 10, c'est-à-dire à un total de 5 tâches identiques avec la variable d'environnement <code>SGE_TASK_ID</code> contenant chaque fois un chiffre d'indice différent.</p> <p>Toutes les tâches de travail de matrice héritent des mêmes requêtes de ressources et définitions d'attributs que spécifié dans la ligne de commande <code>qsub</code> ou <code>qalter</code>, à l'exception de l'option <code>-t</code>. Les tâches sont programmées de façon indépendante et, à condition qu'il y ait suffisamment de ressources, simultanément très similairement à des travaux séparés. Un travail de matrice ou une sous-matrice ne peut toutefois être accédé sous la forme d'un total par des commandes comme <code>qmod(1)</code> ou <code>qdel(1)</code>. Voir les pages de manuel correspondantes pour plus de détails.</p> <p>Les matrices de travail sont souvent utilisées pour exécuter le même type d'opération sur des ensembles de données d'entrée variés corrélés avec le numéro d'indice de la tâche. Le nombre de tâches dans un travail de matrice est illimité. Les <code>STDOUT</code> et <code>STDERR</code> des tâches de travail de matrice seront écrits dans différents fichiers dans l'emplacement par défaut :</p> <pre><nomtravail>.['e' 'o']<id_travail>'.' <id_tâche></pre> <p>Pour changer cet emplacement par défaut, les options <code>-e</code> et <code>-o</code> (voir ci-dessus) peuvent être utilisées avec les pseudo-variables d'environnement <code>\$HOME</code>, <code>\$USER</code>, <code>\$JOB_ID</code>, <code>\$JOB_NAME</code>, <code>\$HOSTNAME</code> et <code>\$SGE_TASK_ID</code>.</p> <p>Vous pouvez utiliser le réacheminement de la sortie pour diriger la sortie de toutes les tâches sur le même fichier, mais le résultat de cette opération est indéfini.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<code>-u nomutilisateur,... -uall</code>	<p>Disponible uniquement pour <code>qalter</code>. Les changements sont uniquement apportés aux travaux qui ont été soumis par les utilisateurs spécifiés dans la liste de <i>nomsutilisateurs</i>. Les gestionnaires ont la possibilité d'utiliser la commande <code>qalter -uall</code> pour modifier tous les travaux de tous les utilisateurs.</p> <p>Si vous utilisez le modificateur <code>-u</code> ou <code>-uall</code>, vous ne devez <i>pas</i> spécifier de <i>liste_travail/ID_tâches</i> supplémentaire.</p>
<code>-v variable[=valeur],...</code>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code>, <code>qresub</code> et <code>qalter</code>.</p> <p>Définit ou redéfinit les variables d'environnement à exporter vers le contexte d'exécution du travail. Si l'option <code>-v</code> est présente, Sun Grid Engine ajoutera les variables d'environnement définies en tant qu'arguments au modificateur et, en option, les valeurs des variables spécifiées au contexte d'exécution du travail.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<code>-verbose</code>	<p>Disponible uniquement pour <code>qrsh</code> et <code>qmake(1)</code>. Contrairement à <code>qsh</code> et <code>qlogin</code>, <code>qrsh</code> ne donne aucun message d'information pendant l'établissement d'une session conforme aux appels système standard <code>rsh(1)</code> et <code>rlogin(1)</code>. Si l'option <code>-verbose</code> est définie, <code>qrsh</code> se comporte en mode verbeux relativement aux commandes <code>qsh</code> et <code>qlogin</code> et donne des informations sur le processus d'établissement de la session <code>rsh(1)</code> ou <code>rlogin(1)</code>.</p>
<code>-verify</code>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code>, <code>qresub</code> et <code>qalter</code>.</p> <p>Ne soumet pas un travail mais imprime des informations sur le travail tel que représenté par la ligne de commande courante et toutes les influences externes pertinentes.</p>
<code>-V</code>	<p>Disponible pour <code>qresub</code>, <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code> et <code>qalter</code>.</p> <p>Spécifie que toutes les variables d'environnement actives au sein de l'utilitaire <code>qsub</code> doivent être exportées dans le contexte du travail.</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<p><code>-w e w n v</code></p>	<p>Disponible uniquement pour <code>qsub</code>, <code>qrsh</code>, <code>qsh</code>, <code>qlogin</code>, <code>qresub</code> et <code>qalter</code>.</p> <p>Spécifie un niveau de validation appliqué au travail à soumettre (<code>qsub</code>, <code>qlogin</code> et <code>qsh</code>) ou au travail en attente spécifié (<code>qalter</code>). Les informations affichées indiquent si le travail peut éventuellement être programmé à condition que le système soit vide et sans autres travaux. Les requêtes de ressources qui dépassent les seuils maximaux configurés ou demandent des attributs de ressources non-disponibles sont parmi les causes possibles de l'échec de cette validation pour certains travaux.</p> <p>Les spécificateurs <code>e</code>, <code>w</code>, <code>n</code> et <code>v</code> définissent les modes de validation suivants :</p> <p><code>e</code> – Erreur : les travaux comportant des requêtes invalides sont refusés ; c'est le mode par défaut pour <code>qrsh</code>, <code>qsh</code> et <code>qlogin</code>.</p> <p><code>w</code> – Avertissement : seul un avertissement s'affichera pour les requêtes invalides.</p> <p><code>n</code> – Aucun : désactive la validation ; c'est le mode par défaut pour <code>qalter</code> et <code>qsub</code>.</p> <p><code>v</code> – Vérifier : ne soumet pas le travail mais imprime un rapport de validation complet.</p> <p>Les contrôles nécessaires consomment la performance et par conséquent le contrôle est désactivé par défaut.</p> <p>Les raisons de l'invalidité des exigences d'un travail par rapport à la disponibilité en ressources des files d'attente s'affichent dans le <code>-w v</code> case en utilisant le format décrit pour l'option <code>-F</code> de <code>qstat(1)</code> (voir la description du Format complet dans la section Formats de sortie de la page de manuel de <code>qstat(1)</code>).</p>

TABLEAU 25 Options de la commande `submit` (suite)

Option	Description
<i>liste_travail</i> / <i>ID_tâches</i>	<p>Cette option se spécifie comme suit : <i>id_travail</i>[.<i>plage_tâches</i>][,<i>id_travail</i>[.<i>plage_tâches</i>],...]</p> <p>Si présent, <i>plage_tâches</i> limite l'effet de l'opération à la plage de tâches de travail de matrice spécifiée par le suffixe rattaché à l'ID du travail (voir l'option <code>-t</code> de <code>qsub(1)</code> pour plus de détails sur les matrices de travail).</p> <p>Le spécificateur de plage de tâches est de la forme <i>n</i>[<i>-m</i>[:<i>s</i>]]. La plage peut être un simple chiffre, une plage simple de la forme <i>n-m</i> ou une plage avec une valeur de pas.</p> <p>À la place de <i>liste_travail</i>/<i>ID_tâches</i> il est possible d'utiliser le mot clé <code>all</code> pour modifier tous les travaux de l'utilisateur courant.</p>
<i>fichierscript</i>	<p>Disponible uniquement pour <code>qsub</code>. <i>fichierscript</i> du travail. S'il n'est pas présent ou si l'opérande est la chaîne à un caractère, <code>-</code>, <code>qsub</code> lit le script depuis l'entrée standard.</p>
<i>args_script</i>	<p>Disponible uniquement pour <code>qsub</code> et <code>qalter</code>. Arguments du travail. Pas valides si le script est entré depuis l'entrée standard.</p> <p><code>qalter</code> permet de changer cette option même pendant l'exécution du travail. Le paramètre modifié ne deviendra toutefois effectif qu'après un redémarrage ou une migration du travail.</p>
<i>args_xterm</i>	<p>Disponible uniquement pour <code>qsh</code>. Arguments de l'exécutable <code>xterm(1)</code>, comme défini dans la configuration. Pour tout détail, consultez <code>sgc_conf(5)</code>.</p>

Variables d'environnement

Le TABLEAU 26 liste les variables d'environnement associées à `submit`.

TABLEAU 26 Variables d'environnement de `submit`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>qtcsh</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement, <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port TCP sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>qsub</code> , <code>qsh</code> , <code>qlogin</code> ou <code>qalter</code> réside. Par défaut, l'hôte local est utilisé.

En plus des variables d'environnement spécifiées comme devant être exportées vers le travail via l'option `-v` ou `-V` (voir ci-dessus), `qsub`, `qsh` et `qlogin` ajoutent les variables suivantes avec les valeurs indiquées, répertoriées dans le TABLEAU 27, à la liste des variables.

TABLEAU 27 Variables d'environnement supplémentaires de `qsub`, `qsh` et `qlogin`

Nom de la variable	Description
<code>SGE_O_HOME</code>	Répertoire de base du client effectuant la soumission.
<code>SGE_O_HOST</code>	Nom de l'hôte sur lequel le client effectuant la soumission tourne.
<code>SGE_O_LOGName</code>	<i>NOMCONNEXION</i> du client effectuant la soumission.
<code>SGE_O_MAIL</code>	<i>E-MAIL</i> du client effectuant la soumission ; il s'agit du répertoire de courrier électronique du client effectuant la soumission.
<code>SGE_O_PATH</code>	Chemin de recherche de l'exécutable du client effectuant la soumission.
<code>SGE_O_SHELL</code>	<i>SHELL</i> du client effectuant la soumission.
<code>SGE_O_TZ</code>	Fuseau horaire du client effectuant la soumission.
<code>SGE_O_WORKDIR</code>	Chemin absolu du répertoire de travail courant du client effectuant la soumission.

Le logiciel Sun Grid Engine définit également des variables supplémentaires dans l'environnement du travail, qui sont indiquées dans le TABLEAU 28.

TABLEAU 28 Variables définies dans l'environnement du travail par les commandes `submit`

Nom de la variable	Description
ARC	Nom de l'architecture Sun Grid Engine du noeud sur lequel le travail est exécuté. Ce nom est inséré dans le binaire <code>sge_execd(8)</code> .
SGE_CKPT_ENV	Spécifie l'environnement de point de reprise (tel que sélectionné par l'option <code>-ckpt</code>) sous lequel un travail de point de reprise s'exécute. Cette variable n'est définie que pour les travaux de point de reprise.
SGE_CKPT_DIR	Cette variable n'est définie que pour les travaux de point de reprise. Elle contient le chemin <code>rép_ptrep</code> (voir <code>checkpoint(5)</code> de l'interface d'écriture de points de reprise.
SGE_STDERR_PATH	Chemin du fichier sur lequel le flux d'erreurs standard du travail est dévié. Cette variable est fréquemment utilisée pour enrichir la sortie avec les messages d'erreur provenant des scripts <i>prolog</i> , <i>epilog</i> , démarrage/arrêt de l'environnement parallèle et d'écriture de points de reprise.
SGE_STDOUT_PATH	Chemin du fichier sur lequel le flux de sortie standard du travail est dévié. Cette variable est fréquemment utilisée pour enrichir la sortie avec les messages d'erreur provenant des scripts <i>prolog</i> , <i>epilog</i> , démarrage/arrêt de l'environnement parallèle et d'écriture de points de reprise.
SGE_JOB_SPOOL_DIR	Répertoire utilisé par <code>sge_shepherd(8)</code> pour stocker les données relatives au travail pendant l'exécution du travail. Ce répertoire appartient au super-utilisateur ou à un compte administratif de Sun Grid Engine et n'est en général pas ouvert en lecture/écriture aux utilisateurs normaux.
SGE_TASK_ID	Indice numérique de la tâche de travail de matrice (voir l'option <code>-t</code> plus haut). Il s'agit d'un nombre unique pour chaque travail de matrice qui peut être utilisé pour référencer différents enregistrements de données d'entrée, par exemple. Cette variable est <code>undefined</code> pour les travaux non-matriciels.
ENVIRONMENT	La variable <code>ENVIRONMENT</code> est sur <code>BATCH</code> pour identifier que le travail est exécuté sous le contrôle de Sun Grid Engine.
HOME	Chemin du répertoire de base de l'utilisateur d'après le fichier <code>passwd(5)</code> .
HOSTNAME	Nom d'hôte du noeud sur lequel l'exécution du travail est en cours.
JOB_ID	Identificateur unique attribué par <code>sge_qmaster(8)</code> lorsque le travail a été soumis. L'ID du travail est un entier décimal compris entre 1 et 99 999.

TABLEAU 28 Variables définies dans l'environnement du travail par les commandes `submit` (suite)

Nom de la variable	Description
JOB_NAME	Nom du travail : au choix <code>INTERACT</code> pour les travaux interactifs ou un nom construit à partir du nom du fichier de script <code>qsub</code> , un point et les chiffres de l'ID du travail. Cette valeur par défaut peut être surécrite par l'option <code>-N</code> .
LOGNAME	Nom de connexion de l'utilisateur provenant du fichier <code>passwd(5)</code> .
NHOSTS	Nombre des hôtes utilisés par un travail parallèle.
NQUEUE	Nombre de files d'attente allouées pour le travail (toujours 1 pour les travaux série).
NSLOTS	Nombre d'emplacements de file d'attente utilisés dans un travail parallèle.
PATH	Chemin de recherche de shell par défaut : <code>/usr/local/bin:/usr/ucb:/bin:/usr/bin</code>
PE	Environnement parallèle sous lequel le travail s'exécute (travaux parallèles uniquement).
PE_HOSTFILE	Chemin d'un fichier contenant la définition de la machine parallèle virtuelle attribuée à un travail parallèle par Sun Grid Engine. Voir la description du paramètre <code>Spe_hostfile</code> dans <code>sge_pe(5)</code> pour tout détail sur le format de ce fichier. Cette variable d'environnement n'est disponible que pour les travaux parallèles.
QUEUE	Nom de la file d'attente dans laquelle le travail est exécuté.
REQUEST	Disponible uniquement pour les travaux par lots. Nom de requête d'un travail tel que spécifié par le modificateur <code>-N</code> (voir ci-dessus) ou pris comme le nom du fichier de script du travail.
RESTARTED	Cette variable est sur 1 si un travail a été redémarré à la suite d'un crash du système ou d'une migration dans le cas d'un travail de point de reprise. Sinon, cette variable a la valeur 0.
SHELL	Shell de connexion de l'utilisateur d'après le fichier <code>passwd(5)</code> . Remarque : ce n'est pas nécessairement celui utilisé pour le travail.
TMPDIR	Chemin absolu du répertoire de travail temporaire du travail.
TMP	Identique à <code>TMPDIR</code> ; fourni pour la compatibilité avec <code>NQS</code> .
TZ	Variable de fuseau horaire importée de <code>sge_execd(8)</code> si définie.
USER	Nom de connexion de l'utilisateur provenant du fichier <code>passwd(5)</code> .

Restrictions

Il n'y a pas de terminal de contrôle pour les travaux par lots sous Sun Grid Engine et tout test ou action au niveau d'un terminal de contrôle échouera. Si ces opérations sont dans votre fichier `.login` ou `.cshrc`, il est possible qu'elles entraînent l'abandon du travail.

Insérez le test suivant avant toute commande non-appropriée à vos travaux par lots dans votre fichier `.login`.

```
if ( $?JOB_NAME) then
    echo "Sun Grid Engine spooled job"
    exit 0
endif
```

N'oubliez pas de définir votre chemin de recherche de shell dans votre démarrage de shell avant ce code.

Statuts de sortie

Les valeurs de sortie suivantes sont retournées.

- 0 – L'opération a été exécutée avec succès.
- 25 – Il n'a pas été possible d'enregistrer un nouveau travail à cause de la limite `max_u_jobs` ou `max_jobs` maximale configurée. Des informations supplémentaires sont sans doute disponibles dans `sgengine_conf(5)`.
- >0 – Une erreur est survenue.

Exemples

L'exemple qui suit est la forme la plus simple d'un fichier de script Sun Grid Engine.

```
#!/bin/csh
a.out
```

L'exemple qui suit est un script Sun Grid Engine plus complexe.

```
#!/bin/csh

# Which account to be charged cpu time
#$ -A père_noël

# date-time to run, format [[CC]yy]MMDDhhmm[.SS]
#$ -a 12241200

# to run I want 6 ou more parallel processes
# under the PE pvm. the processes require
# 128M of memory
#$ -pe pvm 6- -l mem=128

# Si I run on dec_x put stderr in /tmp/foo, if I
# run on sun_y, put stderr in /usr/me/foo
#$ -e dec_x:/tmp/foo,sun_y:/usr/me/foo

# Send mail to these users
#$ -M père@ciel,noël@ciel

# Mail at beginning/end/on suspension
#$ -m bes

# Export these environmental variables
#$ -v PVM_ROOT,FOOBAR=BAR

# The job est located in the current
# working directory.
#$ -cwd

a.out
```

Fichiers

- STDOUT de l'ID de travail - `$REQUEST.oJID[.idtâche]`
- STDERR du travail - `$REQUEST.eJID[.idtâche]`
- STDOUT de l'env. par. du travail - `$REQUEST.poJID[.idtâche]`
- STDERR de l'env. par. du travail - `$REQUEST.peJID[.idtâche]`
- Fichier d'hôte de l'env. par. du travail - `$REQUEST.hostsJID[.idtâche]`
- Alias de chemin de `cwd` - `$cwd/.sge_aliases`
- Requête par défaut `cwd` - `$cwd/.sge_request`
- Alias de chemins d'utilisateur - `$HOME/.sge_aliases`
- Requête par défaut utilisateur - `$HOME/.sge_request`
- Alias de chemins de grappe - `<racine_sge>/<cellule>/common/.sge_aliases`
- Requête par défaut grappe - `<racine_sge>/<cellule>/common/.sge_request`
- Fichier hôte maître de Sun Grid Engine -
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `qconf(1)`, `qdel(1)`, `qhold(1)`, `qmod(1)`, `qrls(1)`, `qstat(1)`,
`accounting(5)`, `sge_aliases(5)`, `sge_conf(5)`, `sge_request(5)`,
`sge_pe(5)`, `complex(5)`

Droits d'auteur

Si configurées de manière correspondante, `qrsh` et `qlogin` contiennent des parties du code `rsh`, `rshd`, `telnet` et `telnetd` dont les droits d'auteur sont détenus par les dirigeants de l'Université de Californie. Par conséquent, la remarque suivante s'applique relativement à `qrsh` et `qlogin` : Ce produit inclut des logiciels développés par l'Université de Californie, Berkeley et ses partenaires.

Voir `sge_intro(1)` ainsi que les informations fournies dans `<racine_sge>/3rd_party/qrsh` et `<racine_sge>/3rd_party/qlogin` pour l'énoncé des autres droits et autorisations.

access_list(5)

Nom

`access_list` - Format des fichiers des listes d'accès de Sun Grid Engine.

Description

Les listes d'accès sont utilisées dans les produits Sun Grid Engine pour définir les droits d'accès des utilisateurs aux files d'attente (voir `queue_conf(5)`) ou aux environnements parallèles (voir `sg_pe(5)`). Une liste des listes d'accès couramment configurées peut être affichée via l'option `qconf(1) -su1`. Le contenu de chaque liste de contrôle d'accès listée peut être affiché via le modificateur `-su`. La sortie suit le format `access_list`. Il est possible de créer de nouvelles listes d'accès ou de modifier celles existantes via les options `-au` et `-du` de `qconf(1)`.

Format

Chaque utilisateur ou groupe d'utilisateurs UNIX apparaît sur une ligne. Seuls les noms symboliques sont autorisés. Les noms de groupes se différencient des noms d'utilisateurs par le préfixe «@».

Voir aussi

`sg_intro(1)`, `qconf(1)`, `sg_pe(5)`, `queue_conf(5)`

Droits d'auteur

Voir `sg_intro(1)` pour l'énoncé complet des droits et autorisations.

accounting(5)

Nom

`accounting` – Format du fichier de comptabilité de Sun Grid Engine

Description

Un enregistrement de comptabilité est écrit dans le fichier `accounting` de Sun Grid Engine pour chaque travail terminé. Le fichier `accounting` est traité par `qacct(1)` pour calculer les statistiques de comptabilité.

Format

Chaque travail est représenté par une ligne dans le fichier `accounting`. Les lignes vides et celles qui contiennent un caractère ou moins sont ignorées. Les entrées d'enregistrement de comptabilité sont séparées par des signes deux-points (:). Dans l'exemple qui suit les entrées sont indiquées dans l'ordre de leur apparition.

- `qname` – Nom de la file d'attente dans laquelle le travail a été exécuté.
- `hostname` – Nom de l'hôte d'exécution.
- `group` – ID du groupe effectif du propriétaire du travail lors de l'exécution du travail.
- `owner` – Propriétaire du travail Sun Grid Engine.
- `job_name` – Nom du travail.
- `job_number` – Identificateur ou numéro du travail.
- `account` – Chaîne de compte spécifiée par l'option `qsub(1)` ou `qalter(1)` -A.
- `priority` – Valeur de priorité attribuée au travail correspondant au paramètre `priority` dans la configuration de file (voir `queue_conf(5)`).
- `submission_time` – Heure de soumission en secondes (d'après format epoch).
- `start_time` – Heure de début en secondes (d'après format epoch).
- `end_time` – Heure de fin en secondes (d'après format epoch).
- `failed` – Indique le problème survenu dans le cas où un travail n'a pas pu être démarré sur l'hôte d'exécution (par ex. parce que le propriétaire du travail n'avait pas de compte valide sur la machine en question). Si le logiciel Sun Grid Engine essaye de démarrer plusieurs fois un travail, plusieurs entrées peuvent être consignées dans le fichier de comptabilité pour le même ID de travail.

- `exit_status` – Statut de sortie du script du travail (ou statut spécifique de Sun Grid Engine dans le cas de certaines conditions d'erreur).
- `ru_wallclock` – Différence entre `end_time` et `start_time` (voir ci-dessus).

Le reste des entrées de comptabilité de cette catégorie suivent le contenu de la structure UNIX `usage` standard comme décrit dans `getrusage(2)`.

Les entrées suivantes sont fournies :

- `ru_utime`
- `ru_stime`
- `ru_maxrss`
- `ru_ixrss`
- `ru_ismrss`
- `ru_idrss`
- `ru_isrss`
- `ru_minflt`
- `ru_majflt`
- `ru_nswap`
- `ru_inblock`
- `ru_oublock`
- `ru_msgsnd`
- `ru_msgrcv`
- `ru_nsignals`
- `ru_nvcsw`
- `ru_nivcsw`
- `project` – Projet qui a été attribué au travail. Les projets sont uniquement pris en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition.
- `department` – Service qui a été attribué au travail. Les services sont uniquement pris en charge dans le cas d'un système Sun Grid Engine, Enterprise Edition.
- `granted_pe` – Environnement parallèle ayant été sélectionné pour le travail.
- `slots` – Nombre d'emplacements qui ont été attribués au travail par le programmeur.
- `task_number` – Indice numérique de la tâche de travail de matrice.
- `cpu` – Utilisation du temps UC en secondes.
- `mem` – Utilisation complète de la mémoire en Go.
- `io` – Quantité de données transférées dans les opérations d'entrée/sortie.
- `category` – Chaîne spécifiant la catégorie du travail.
- `iow` – Temps d'attente E/S, en secondes.
- `pe_taskid` – Si cet identificateur est défini, tâche qui faisait partie d'un travail parallèle et avait été transférée au système Sun Grid Engine, Enterprise Edition via l'interface `qrsh -inherit`.
- `maxvmem` – Taille `vmem maximum`, en octets.

Voir aussi

`sgc_intro(1)`, `qacct(1)`, `qalter(1)`, `qsub(1)`, `getrusage(2)`,
`queue_conf(5)`

Droits d'auteur

Voir `sgc_intro(1)` pour l'énoncé complet des droits et autorisations.

calendar_conf (5)

Nom

`calendar_conf` - Format des fichiers de configuration d'agendas de Sun Grid Engine.

Description

`calendar_conf` reflète le format de la configuration d'agendas de Sun Grid Engine. Une définition d'agenda est utilisée pour spécifier les périodes de temps « en service » et « hors service » des files d'attente de Sun Grid Engine en fonction des heures du jour, des jours de la semaine ou des jours de l'année. Différents agendas peuvent être mis en oeuvre et la définition d'agenda appropriée à une certaine catégorie de travaux peut être jointe à une file d'attente.

Les entrées `calendar_conf` peuvent être ajoutées, modifiées et affichées avec les options `-Acal`, `-acal`, `-Mcal`, `-mcal`, `-scal` et `-scall` de `qconf(1)` ou au moyen de la boîte de dialogue de configuration d'agendas de l'IHM `qmon(1)`. Le format des entrées de configuration d'agenda est le suivant :

Format

- `calendar_name` - Nom de l'agenda à utiliser pour le joindre aux files d'attente ou pour en administrer la définition.

- *year* – Définition du statut de la file selon les jours de l'année. Ce champ spécifiera en général les jours de l'année (et en option à quelles heures de ces jours) auxquels une file d'attente, à laquelle l'agenda est joint, passera dans un état donné. La syntaxe du champ *year* est la suivante :

year :=

```
{liste_plages_jours_année[=liste_plages_heurejour][=état]
 | [liste_plages_jours_année=]liste_plages_heurejour[=état]
 | [liste_plages_jours_année=][liste_plages_heurejour=]état} ...
```

Où :

- Au moins un des éléments *liste_plages_jours_année*, *liste_plages_heurejour* et *état* doit toujours être présent.
- Tous les jours de l'année sont considérés si *liste_plages_jours_année* est omis.
- Toute la journée est considérée si *liste_plages_heurejour* est omis.
- La file est basculée sur « off » (désactivée) si *état* est omis.
- La file est supposée activée les jours qui ne sont pas référencés de façon implicite (en omettant *liste_plages_jours_année*) ou explicite.

Les composants syntaxiques sont définis comme suit :

liste_plages_jours_année := {*jourannée-jourannée* | *jourannée*},...

liste_plages_heurejour := *heures[:minutes][:secondes]-heures[:minutes][:secondes]*,...

état := {on | off | suspended}

jour_année := *mois_jour.mois.année*

jour_mois := {1 | 2 | ... | 31}

mois := {jan | feb | ... | dec | 1 | 2 | ... | 12}

année := {1970 | 1971 | ... | 2037}

- *week* – Définition du statut de la file selon les jours de la semaine. Ce champ spécifiera en général les jours de la semaine (et en option à quelles heures de ces jours) une file d'attente, à laquelle l'agenda est joint, passera dans un état donné.

La syntaxe du champ *week* est la suivante :

week :=

```
{liste_plages_jours_semaine[=liste_plages_heurejour][=état]
 | [liste_plages_jours_semaine=]liste_plages_heurejour[=état]
 | [liste_plages_jours_semaine=][liste_plages_heurejour=]état} ...
```

Où :

- Au moins un des éléments *liste_plages_jours_semaine*, *liste_plages_heurejour* et *état* doit toujours être présent.

- Tous les jours de la semaine sont considérés si *liste_plages_jours_semaine* est omis.
- La syntaxe et la sémantique de *liste_plages_heurejour* et *état* sont identiques à la définition donnée pour le champ de l'année ci-dessus.
- La file sera supposée activée les jours qui ne sont pas référencés de façon implicite (en omettant *liste_plages_jours_semaine*) ou explicite.

Et où *liste_plages_jours_semaine* est défini comme suit

liste_plages_jours_semaine := {*joursemaine-joursemaine*|*joursemaine*},...

jour_semaine := {mon|tue|wed|thu|fri|sat|sun}

Sémantique

Les entrées successives dans les champs *year* et *week* (séparées par des blancs) sont combinées conformément à la règle suivante :

- off – Les zones sont remplacées en superposant les zones on et suspended.
Ainsi, une entrée de la forme *week 12-18 tue=13-17=on* signifie que les files d'attente faisant référence à l'agenda correspondant sont désactivées pendant la semaine à l'exception du mardi de 13h00 à 17h00 où les files d'attente sont disponibles.

Exemples

(Les exemple suivants sont contenus dans le répertoire, *\$racine_sge/util/resources/calendars*).

- Agenda de nuit, weekend et jours fériés – Les jours fériés, les files d'attente « night » sont explicitement activées. Les jours ouvrés, les files d'attente sont désactivées de 6.00 à 20.00. Le samedi et le dimanche sont implicitement gérés comme des périodes d'activité.

```
calendar_namenight
year
1.1.1999,6.1.1999,28.3.1999,30.3.1999-31.3.1999,18.5.1999-19.5.1999,
3.10.1999,25.12.1999,26.12.1999=on
weekmon-fri=6-20
```

- Agenda de jour – Les jours fériés, les files d’attente « day » sont désactivées. Les jours ouvrés ces files d’attente sont fermées la nuit entre 20.00 et 6.00, c’est-à-dire qu’elles sont fermées le lundi de 0.00 à 6.00 et le vendredi de 20.00 à 24.00. Le samedi et le dimanche, les files d’attente sont désactivées.

```
calendar_nameday
year
1.1.1999,6.1.1999,28.3.1999,30.3.1999-31.3.1999,18.5.1999-19.5.1999,
3.10.1999,25.12.1999,26.12.1999
weekmon-fri=20-6 sat-sun
```

- Agenda de nuit, weekend et jours fériés avec interruption – Le scénario est pratiquement le même que dans le premier exemple mais les files d’attente sont interrompues au lieu d’être désactivées.

```
calendar_namenight_s
year
1.1.1999,6.1.1999,28.3.1999,30.3.1999-31.3.1999,18.5.1999-19.5.1999,
3.10.1999,25.12.1999,26.12.1999=on
weekmon-fri=6-20=suspended
```

- Agenda de jour avec interruption – Le scénario est pratiquement le même que dans le deuxième exemple mais les files d’attente sont interrompues au lieu d’être désactivées.

```
calendar_nameday_s
year
1.1.1999,6.1.1999,28.3.1999,30.3.1999-31.3.1999,18.5.1999-19.5.1999,
3.10.1999,25.12.1999,26.12.1999=suspended
week mon-fri=206=suspended sat-sun=suspended
```

Voir aussi

`sgc_intro(1)`, `qconf(1)`, `queue_conf(5)`

Droits d’auteur

Voir `sgc_intro(1)` pour l’énoncé complet des droits et autorisations.

checkpoint (5)

Nom

`checkpoint` – Format des fichiers de configuration d’environnement de point de reprise de Sun Grid Engine.

Description

L’écriture de points de reprise est un utilitaire qui permet de sauvegarder le statut complet d’un programme ou travail en cours d’exécution et de le restaurer et de le redémarrer à partir de ce point de reprise à un moment ultérieur si ce programme ou travail d’origine vient à être bloqué, par exemple en cas de crash du système.

Sun Grid Engine fournit différents niveaux de prise en charge de l’écriture de points de reprise (voir `sge_ckpt(1)`). L’environnement de point de reprise décrit ici est un moyen de configurer les différents types d’écriture de points de reprise utilisés dans votre grappe Sun Grid Engine ou des parties de celle-ci. Dans cet objectif, vous pouvez définir les opérations qui devront être exécutées pour lancer la génération de points de reprise, une migration d’un point de reprise sur un autre hôte ou le redémarrage d’une application avec points de reprise ainsi que la liste des files d’attente qui sont appropriées à une méthode d’écriture de points de reprise.

La prise en charge de différents systèmes d’exploitation peut facilement forcer Sun Grid Engine à introduire des dépendances à l’égard du système d’exploitation pour la configuration du fichier de configuration de points de reprise et les mises à jour des versions de système d’exploitation prises en charge peut mener à changer fréquemment les détails de la mise en oeuvre. Reportez-vous au fichier, `<racine_sge>/doc/checkpointing.asc` pour plus d’informations.

Utilisez les options `-ackpt`, `-dckpt`, `-mckpt` ou `-sckpt` de la commande `qconf(1)` pour manipuler les environnements de point de reprise depuis la ligne de commande ou utilisez la boîte de dialogue `qmon(1)` correspondante pour la configuration interactive basée sur X-Windows.

Format

Le format d’un fichier `checkpoint` est le suivant :

- `ckpt_name` – Nom de l’environnement de point de reprise. Sera utilisé dans le modificateur `qsub(1)` `-ckpt` ou pour les options `qconf(1)` mentionnées plus haut.

- `interface` – Type d'écriture de points de reprise utilisé. Les types suivants sont actuellement valides :
 - `hibernator` – Écriture de points de reprise de niveau noyau Hibernate.
 - `cpr` – Écriture de points de reprise de niveau noyau SGI.
 - `cray-ckpt` – Écriture de points de reprise de niveau noyau Cray.
 - `transparent` – Le système Sun Grid Engine assume que les travaux soumis avec une référence à cette interface d'écriture de points de reprise utilisent une bibliothèque de points de reprise telle que celle fournie par le progiciel du domaine public *Condor*.
 - `userdefined` – Le système Sun Grid Engine assume que les travaux soumis avec une référence à cette interface d'écriture de points de reprise mettent en oeuvre leur propre méthode d'écriture de points de reprise.
 - `application-level` – Utilise toutes les commandes d'interface configurées dans l'objet d'écriture de points de reprise comme dans le cas des interfaces d'écriture de points de reprise de niveau noyau (`cpr`, `cray-ckpt`, etc.) à l'exception de la commande `restart_command` (voir ci-dessous), qui n'est pas utilisée (même si elle est configurée) mais le script du travail est appelé à la place dans le cas d'un redémarrage.
- `queue_list` – Liste séparée par des virgules des files d'attente auquel les travaux parallèles appartenant à l'environnement parallèle ont accès.
- `ckpt_command` – Chaîne de commande de type ligne de commande devant être exécutée par Sun Grid Engine afin de lancer un point de reprise.
- `migr_command` – Chaîne de commande de type ligne de commande devant être exécutée par Sun Grid Engine pendant la migration d'un travail de point de reprise d'un hôte à un autre.
- `restart_command` – Chaîne de commande de type ligne de commande devant être exécutée par Sun Grid Engine lors du redémarrage d'une application pour laquelle un point de reprise a été généré.
- `clean_command` – Chaîne de commande de type ligne de commande devant être exécutée par Sun Grid Engine afin d'effectuer un nettoyage à l'arrêt d'une application de point de reprise.
- `ckpt_dir` – Emplacement du système de fichiers dans lequel les points de reprise de taille importante doivent être stockés.
- `queue_list` – Contient une liste séparée par des virgules ou des blancs des noms des files d'attente appropriées pour un travail si l'environnement de point de reprise a été spécifié à la soumission du travail.
- `ckpt_signal` – Signal UNIX qui est envoyé à un travail par Sun Grid Engine pour lancer la génération de points de reprise. La valeur de ce champ peut être un nom symbolique de la liste produite par l'option `-l` de la commande `kill(1)` ou un nombre entier qui doit être un signal valide sur les systèmes utilisés pour l'écriture de points de reprise.

- `when` – Fixe les moments auxquels les points de reprise devraient être générés. Les valeurs valides de ce paramètre sont les lettres `s`, `m`, `x` et `r` et toute combinaison de celles-ci sans séparation aucune entre les caractères. Les mêmes lettres sont autorisées pour l’option `-c` de la commande `qsub(1)` qui surécrit les définitions dans l’environnement de point de reprise utilisé. La signification de ces lettres est la suivante :
 - `s` – Avec cette lettre, un point de reprise est écrit pour un travail, puis le travail est abandonné et si possible migré si le `sgexecd(8)` correspondant est arrêté sur la machine du travail.
 - `m` – Avec cette lettre, des points de reprise sont générés périodiquement à l’intervalle `min_cpu_interval` défini par la file d’attente (voir `queue_conf(5)`) dans laquelle un travail s’exécute.
 - `x` – Avec cette lettre, un point de reprise est écrit pour un travail, puis le travail est abandonné et si possible migré dès que le travail est interrompu (manuellement ou automatiquement).
 - `r` – Avec cette lettre, un travail sera reprogrammé (aucun point de reprise ne sera écrit) lorsque l’hôte sur lequel le travail est en cours d’exécution passera à l’état inconnu et que le délai, `reschedule_unknown` (voir `sgconf(5)`), défini dans la configuration de grappe globale/locale sera dépassé.

Restrictions

Remarque – La fonctionnalité des procédures d’écriture de points de reprise, migration ou redémarrage fournies par défaut avec Sun Grid Engine, ainsi que la façon dont elles sont appelées dans les paramètres `ckpt_command`, `migr_command` ou `restart_command` de n’importe lequel des environnements de point de reprise par défaut, ne doit pas être modifiée sous peine de relever pleinement de la responsabilité de l’administrateur configurant l’environnement de point de reprise. Le logiciel Sun Grid Engine se limite à appeler ces procédures et en évaluer le statut de sortie. Le système Sun Grid Engine n’a aucun moyen de détecter si elles n’effectuent pas correctement leurs tâches ou ne sont pas appelées comme il se doit, ce qui risque de causer un comportement inattendu du mécanisme d’écriture de points de reprise.

Voir aussi

`sg_intro(1)`, `sg_ckpt(1)`, `qconf(1)`, `qmod(1)`, `qsub(1)`, `sgexecd(8)`

Droits d’auteur

Voir `sg_intro(1)` pour l’énoncé complet des droits et autorisations.

sgc_request (5)

Nom

`sgc_request` – Format des fichiers de définition des requêtes par défaut de Sun Grid Engine.

Description

`sgc_request` reflète le format des fichiers permettant de définir des profils de requête par défaut. Lorsqu'ils sont disponibles, les fichiers de requête par défaut sont lus et traités pendant la soumission d'un travail avant la prise en considération de toute option de soumission imbriquée dans le script du travail et avant les éventuelles options de la ligne de commande `qsub(1)` ou `qsh(1)`. Les options de la ligne de commande et celles imbriquées dans le script peuvent par conséquent remplacer les paramètres des fichiers de requête par défaut (voir `qsub(1)` ou `qsh(1)` pour plus de détails).

Il existe un fichier de définition de requête par défaut global pour la grappe, un fichier de définition de requête par défaut personnel de l'utilisateur et un fichier de définition de requête par défaut local du répertoire de travail. Les fichiers de requête locaux du répertoire de travail ont la précedence la plus élevée, viennent ensuite ceux personnels des utilisateurs puis les fichiers de requête par défaut globaux de la grappe.

Remarque – L'option `-clear` de `qsub(1)` ou `qsh(1)` peut être utilisée pour éliminer à tout moment tout paramètre préalable d'un fichier de requête par défaut, dans les indicateurs de script imbriqués ou dans une option de ligne de commande `qsub(1)` ou `qsh(1)`.

Le format des fichiers de définition de requête par défaut est le suivant :

- Les fichiers de requête par défaut peuvent contenir un nombre arbitraire de lignes. Les lignes vierges et celles qui commencent par le signe « # » dans la première colonne sont sautées.
- Chaque ligne non-sautée peut contenir toute option `qsub(1)` comme décrit dans le *Manuel de référence de Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3*. Il est possible d'avoir plusieurs options par ligne. Le fichier de script par lots et les arguments d'option du script par lots ne sont pas considérés comme des options de `qsub(1)` et ne sont donc pas autorisés dans un fichier de requête par défaut.

Exemples

L'exemple qui suit est un exemple simple de fichier de définition de requête par défaut :

```
# Default Requests File
# request arch to be sun4 and a CPU-time of 5hr
-l arch=sun4,s_cpu=5:0:0
# don't restart the job in case of system crashes
-r n
```

Avoir défini un fichier de définition de requête par défaut comme celui-là et soumettre un travail comme suit :

```
qsub test.sh
```

aurait exactement le même effet que si le travail avait été soumis avec :

```
sub -l arch=sun4,s_cpu=5:0:0 -r n test.sh
```

Fichiers

- Fichier de valeurs par défaut global –
`<racine_sge>/<cellule>/common/sge_request`
- Fichier de valeurs par défaut personnel de l'utilisateur – `$HOME/.sge_request`
- Fichier de valeurs par défaut du répertoire `cwd` – `$cwd/.sge_request`

Voir aussi

`sge_intro(1)`, `qsh(1)`, `qsub(1)`, *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3*.

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sge_aliases(5)

Nom

sge_aliases – Format du fichier d’alias de chemins de Sun Grid Engine

Description

L’utilitaire d’alias de chemins de Sun Grid Engine fournit aux administrateurs et aux utilisateurs les moyens de refléter des structures de fichiers compliquées et non-homogènes dans des environnements distribués (tels que des répertoires de base d’utilisateur montés sous différents chemins sur différents hôtes) et d’assurer que Sun Grid Engine soit en mesure de localiser les répertoires de travail appropriés pour exécuter les travaux par lots.

Il y a un fichier d’alias de chemins global du système et un fichier d’alias local d’utilisateur. `sge_aliases` définit le format de ces deux fichiers :

- Les lignes vierges et celles qui commencent par le signe « # » dans la première colonne sont sautées.
- Chaque ligne n’étant pas vierge et ne commençant pas par « # » doit contenir quatre chaînes séparées par un nombre quelconque de blancs ou de tabulations.
- La première chaîne spécifie un chemin-source, la seconde un hôte de soumission, la troisième un hôte d’exécution et la quatrième le chemin-source de remplacement.
- Les deux entrées hôte d’exécution et hôte de soumission peuvent consister en un unique signe « * » qui correspond à tout hôte.

Si l’indicateur `-cwd` (et seulement si `-` sinon le répertoire de base de l’utilisateur sur l’hôte d’exécution est sélectionné pour exécuter le travail) de `qsub(1)` a été spécifié, le mécanisme d’alias de chemins est activé et les fichiers sont traités comme suit :

- Après que `qsub(1)` a récupéré le chemin du répertoire de travail courant physique, le fichier d’alias de chemins global de la grappe est lu s’il est présent. Le fichier d’alias de chemins de l’utilisateur est ensuite lu s’il a été joint au fichier global.
- Les lignes qui ne doivent pas être sautées sont lues une à une en partant du haut du fichier et les translations qu’elles spécifient sont stockées si nécessaire.
- Une translation n’est stockée que si l’entrée d’hôte de soumission correspond à l’hôte sur lequel `qsub(1)` est exécuté et si le chemin-source forme la partie initiale du répertoire de travail courant ou des remplacements de chemin-source déjà stockés.

- Dès que les deux fichiers sont lus les informations d'alias de chemins stockées sont transmises avec le travail soumis.
- Sur l'hôte d'exécution, les informations d'alias seront évaluées. La partie initiale du répertoire de travail courant sera remplacée par le chemin-source de remplacement si l'entrée d'hôte d'exécution de l'alias de chemins correspond à l'hôte d'exécution.

Remarque – La chaîne du répertoire de travail courant sera changée dans ce cas et les alias de chemins successifs doivent correspondre au chemin du répertoire de travail remplacé pour être appliqués.

Exemples

L'exemple qui suit est un exemple simple de fichier d'alias de chemins qui résout les problèmes posés par les chemins non-homogènes si `automount(8)` est utilisé :

```
# Path Aliasing File
# src-pathsub-hostexec-hostreplacement
/tmp_mnt/**/
# replaces any occurrence of /tmp_mnt/ by /
# if submitting or executing on any host.
# Thus paths on nfs server and clients are the same
```

Fichiers

- Fichier d'alias global – `<racine_sge>/<cellule>/common/sge_aliases`
- Fichier d'alias local de l'utilisateur – `$HOME/.sge_aliases`

Voir aussi

`sge_intro(1)`, `qsub(1)`, *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3*

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sge_conf (5)

Nom

sge_conf – Fichiers de configuration de Sun Grid Engine.

Description

sge_conf définit les configurations globale et locale de Sun Grid Engine et peut être affiché et modifié par `qconf(1)` en utilisant les options `-sconf` et `-mconf`. Seuls le super-utilisateur ou l'administrateur de la grappe peuvent modifier `sge_conf`.

À son premier démarrage, `sge_qmaster(8)` contrôle s'il y a une configuration Sun Grid Engine valide dans un emplacement connu de la hiérarchie des répertoires interne de Sun Grid Engine. Si c'est le cas, il charge ces informations de configuration et les traite. Si ce n'est pas le cas, `sge_qmaster(8)` écrit une configuration générique contenant les valeurs par défaut dans ce même emplacement. Au démarrage, les démons d'exécution de Sun Grid Engine `sge_execd(8)` tirent leur configuration de `sge_qmaster(8)`.

La configuration actuelle pour à la fois `sge_qmaster(8)` et `sge_execd(8)` est une superposition de la configuration dite *globale* et d'une configuration *locale* pertinente pour l'hôte sur lequel un démon maître ou d'exécution réside. Si une configuration locale est disponible, ses entrées surécrivent les entrées correspondantes de la configuration globale.

Remarque – Il n'est pas nécessaire que la configuration locale contienne toutes les entrées de configuration valides, il suffit qu'elle présente celles à modifier par rapport aux entrées globales.

Format

Les paragraphes qui suivent fournissent de brèves descriptions des différents paramètres qui composent les configurations globale et locale pour une grappe Sun Grid Engine.

- **qmaster_spool_dir** – Emplacement où le répertoire spool maître réside. Seuls `sge_qmaster(8)` et `sge_shadowd(8)` ont besoin d’avoir accès à ce répertoire. Le super-utilisateur doit cependant pouvoir y accéder en lecture/écriture. Le répertoire spool maître — en particulier le répertoire des travaux et le fichier journal de messages — peut devenir relativement grand en fonction de la taille de la grappe et du nombre des travaux. Pensez à allouer suffisamment d’espace disque et à nettoyer régulièrement les fichiers journaux, par exemple en recourant à un travail `cron(8)`.

Étant donné qu’il s’agit d’un paramètre défini au moment de l’installation, changer **qmaster_spool_dir** dans un système en fonctionnement n’est pas pris en charge.

L’emplacement par défaut du répertoire spool maître est :

`<racine_sge>/<cellule>/spool/qmaster`

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l’hôte d’exécution.

- **execd_spool_dir** (Rép. spool `execd`) – Chemin du répertoire de spool du démon d’exécution. Là encore, un répertoire de spool doit présenter un droit d’accès en lecture/écriture pour le super-utilisateur. L’entrée relative à ce paramètre dans la configuration globale peut être surécrite par les configurations locales des hôtes d’exécution, c’est-à-dire que chaque `sge_execd(8)` peut avoir un répertoire spool privé avec un chemin différent, cas dans lequel des droits d’accès en lecture/écriture doivent être fournis au seul compte de super-utilisateur de l’hôte d’exécution correspondant.

Sous **execd_spool_dir**, un répertoire nommé d’après le nom d’hôte de l’hôte d’exécution est ouvert et contient toutes les informations mises en attente sur disque. Par conséquent, il est possible que les **rep_spool_execd** de tous les hôtes d’exécution fassent physiquement référence au même chemin de répertoire (les droits d’accès de super-utilisateur mentionnés plus haut doivent cependant être fournis).

Étant donné qu’il s’agit d’un paramètre défini au moment de l’installation, changer **execd_spool_dir** dans un système en fonctionnement n’est pas pris en charge.

L’emplacement par défaut du répertoire de spool du démon d’exécution est `<racine_sge>/<cellule>/spool`.

L’entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l’hôte d’exécution.

- **binary_path** (chemin d’accès binaire) – Chemin du répertoire où résident les binaires de Sun Grid Engine. Il est utilisé au sein des composants de Sun Grid Engine pour localiser et démarrer d’autres programmes Sun Grid Engine.

Le chemin donné ici est exploré à la recherche des binaires ainsi que tout répertoire situé sous un nom de répertoire égal à l’architecture système d’exploitation courante. Par conséquent, `/usr/Sun Grid Engine/bin` fonctionnera pour toutes les architectures, si les binaires correspondants sont situés dans les sous-répertoires nommés `aix43`, `cray`, `glinux`, `hp10`, `irix6`, `osf4`, `solaris`, etc.

Chaque `sge_execd(8)` peut avoir son chemin de binaires propre. Changer le chemin des binaires a un effet immédiat pour `sge_execd(8)`.

L'emplacement par défaut pour le chemin des binaires est : `<racine_sge>/bin`

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- **mailer** (protocole e-mail) – Chemin absolu de l'agent de protocole e-mail sur votre système. Il doit accepter la syntaxe suivante :

```
mailer -s <objet-du-message-électronique> <destinataire>
```

Chaque `sge_execd(8)` peut utiliser un agent de messagerie propre. Changer *mailer* a un effet immédiat.

La valeur par défaut de *mailer* dépend du système d'exploitation de l'hôte sur lequel l'installation du maître Sun Grid Engine a été exécutée. Les valeurs les plus courantes sont `/bin/mail` ou `/usr/bin/Mail`.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- **xterm** – Chemin absolu de l'émulateur de terminal X Window System, `xterm(1)`.

Chaque `sge_execd(8)` peut utiliser un émulateur de terminal propre. Changer *xterm* a un effet immédiat.

La valeur par défaut de *xterm* est : `/usr/bin/X11/xterm`

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- **load_sensor** (capteur de charge) – Liste séparée par des virgules des chemins de script shell des exécutables ou des programmes devant être démarrés par `sge_execd(8)` et utilisés afin de récupérer les informations de charge configurables sur le site (par ex. l'espace libre sur une partition de disque donnée).

Chaque `sge_execd(8)` peut utiliser un ensemble propre de programmes ou de scripts *load_sensor*. Un changement de *load_sensor* deviendra effectif après deux intervalles de rapport de charge (voir *load_report_time*). Un capteur de charge sera automatiquement redémarré si l'heure de modification du fichier de l'exécutable du capteur de charge change.

L'entrée de configuration globale pour cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

En sus des capteurs de charge configurés via *load_sensor*, `sge_execd(8)` recherche un fichier exécutable nommé `qloadsensor` dans le chemin du répertoire des binaires Sun Grid Engine de l'hôte d'exécution. Si ce fichier est trouvé, il est traité comme les capteurs de charge configurables définis dans *load_sensor*. Cet utilitaire a été conçu pour pré-installer un capteur de charge par défaut.

- **prolog** (prologue) – Chemin de l'exécutable d'un script shell qui est démarré avant l'exécution des travaux Sun Grid Engine avec les mêmes paramètres d'environnement que pour les travaux Sun Grid Engine qui seront commencés après. Un préfixe « utilisateur@ » optionnel spécifie l'utilisateur sous lequel cette procédure va être démarrée. Cette procédure est conçue comme un moyen à la

disposition de l'administrateur Sun Grid Engine d'automatiser l'exécution de tâches générales spécifiques d'un site telles que la préparation de systèmes de fichiers temporaires ayant besoin des mêmes informations de contexte que le travail. Chaque `sge_execd(8)` peut utiliser un script *prolog* propre. De la même façon, la configuration locale de l'hôte d'exécution peut être surécrite par la configuration de file (voir `queue_conf(5)`). Changer *prolog* a un effet immédiat.

Remarque – *prolog* est exécuté *exactement* comme le script du travail. Par conséquent, toutes les implications décrites sous les paramètres *shell_start_mode* et *login_shells* ci-après s'appliquent.

La valeur par défaut de *prolog* est la valeur spéciale `NONE`, qui empêche l'exécution d'un script de prologue.

Les variables spéciales suivantes qui sont développées au moment de l'exécution peuvent être utilisées (en plus de toutes autres chaînes devant être interprétées par la procédure) pour constituer une ligne de commande :

- *\$host* – Nom de l'hôte sur lequel les procédures *prolog* et *epilog* sont démarrées.
- *\$job_owner* – Nom d'utilisateur du propriétaire du travail.
- *\$job_id* – Numéro d'identification de travail unique du système Sun Grid Engine.
- *\$job_name* – Nom du travail.
- *\$processors* – La chaîne *processors* telle que contenue dans la configuration de file (voir `queue_conf(5)`) de la file maîtresse (file d'attente dans laquelle les procédures *prolog* et *epilog* sont démarrées).
- *\$queue* – La file maîtresse, c'est-à-dire la file dans laquelle les procédures *prolog* et *epilog* sont démarrées. L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.
- *epilog* (épilogue) – Chemin de l'exécutable d'un script shell qui est démarré après l'exécution des travaux Sun Grid Engine avec les mêmes paramètres d'environnement que pour les travaux Sun Grid Engine qui viennent d'être terminés. Un préfixe utilisateur@ optionnel spécifie l'utilisateur sous lequel cette procédure va être démarrée. Cette procédure a été conçue pour permettre aux administrateurs Sun Grid Engine d'automatiser l'exécution de tâches générales spécifiques d'un site telles que le nettoyage des systèmes de fichiers temporaires, ayant besoin des mêmes informations de contexte que le travail. Chaque `sge_execd(8)` peut utiliser un script *epilog* propre. De la même façon, la configuration locale de l'hôte d'exécution peut être surécrite par la configuration de file (voir `queue_conf(5)`). Changer *epilog* a un effet immédiat.

Remarque – *epilog* est exécuté *exactement* comme le script du travail. Par conséquent, toutes les implications décrites sous les paramètres *shell_start_mode* et *login_shells* ci-après s'appliquent.

La valeur par défaut de *epilog* est la valeur spéciale `NONE`, qui empêche l'exécution d'un script d'épilogue. Les mêmes variables spéciales que pour *prolog* peuvent être utilisées pour constituer une ligne de commande.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *shell_start_mode* (mode de démarrage de shell) – Ce paramètre définit les mécanismes qui sont utilisés pour appeler les scripts de travaux sur les hôtes d'exécution. Les valeurs suivantes sont reconnues :
 - *unix_behavior* – Si un utilisateur démarre un script shell de travail de manière interactive sous UNIX en l'appelant juste avec le nom du script, le chargeur d'exécutables du système d'exploitation utilise les informations fournies dans un commentaire tel que « `#!/bin/csh` » dans la première ligne du script pour détecter quel interpréteur de commandes démarrer pour interpréter le script. Ce mécanisme est utilisé par Sun Grid Engine au démarrage des travaux si *unix_behavior* est défini comme *shell_start_mode*.
 - *posix_compliant* – POSIX ne prend pas en considération les commentaires de la première ligne tels que « `#!/bin/csh` ». Le standard POSIX pour les systèmes de files d'attente par lots (P1003.2d) exige donc qu'un système de mise en file d'attente conforme ignore ces lignes et utilise à la place des interpréteurs de commandes par défaut configurés ou spécifiés par l'utilisateur. Par conséquent, si *shell_start_mode* est sur *posix_compliant* Sun Grid Engine utilisera l'interpréteur de commandes indiqué par l'option `-S` de la commande `qsub(1)` ou le paramètre *shell* de la file d'attente à utiliser (voir `queue_conf(5)` pour plus de détails).
 - *script_from_stdin* – Régler le paramètre *shell_start_mode* sur *posix_compliant* ou *unix_behavior* exige que vous définissiez le *umask* utilisé pour `sge_execd(8)` de sorte que chaque utilisateur ait accès en lecture au répertoire des travaux actifs dans le répertoire de spool du démon d'exécution correspondant. Dans le cas où vous auriez des scripts *prolog* et *epilog* configurés, ils doivent aussi pouvoir être lus par tout utilisateur susceptible d'exécuter des travaux.

Si cela viole les politiques de sécurité de votre site, vous pouvez décider de mettre *shell_start_mode* sur *script_from_stdin*. Cela forcera Sun Grid Engine à ouvrir le script du travail ainsi que les scripts *epilog* et *prolog* pour lire dans STDIN en tant que super-utilisateur (si `sge_execd(8)` a été démarré en tant que super-utilisateur) avant de passer au compte d'utilisateur du propriétaire du travail. Le script est ensuite introduit dans le flux STDIN de l'interpréteur de commandes indiqué par l'option `-S` de la commande `qsub(1)` ou le paramètre *shell* de la file d'attente à utiliser (voir `queue_conf(5)` pour plus de détails).

Par conséquent, mettre *shell_start_mode* sur *script_from_stdin* implique aussi un comportement *posix_compliant*.

Remarque – Introduire des scripts dans le flux `STDIN` d'un interpréteur de commandes peut poser des problèmes si les commandes telles que `rsh(1)` sont appelées dans un script de travail car elles traitent également le flux `STDIN` de l'interpréteur de commandes. Ces problèmes peuvent en général être éliminés en redirigeant le canal `STDIN` de ces commandes pour qu'il provienne de `/dev/null` (par ex. `rsh host date < /dev/null`).

Remarque – Toutes les options de ligne de commande associées au travail sont transférées au shell d'exécution. Ce shell ne les transmettra au travail que si elles ne sont pas reconnues comme des options de shell valides.

Changer `shell_start_mode` a un effet immédiat. La valeur par défaut de `shell_start_mode` est `posix_compliant`.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- `login_shells` (shells de connexion) – Les interpréteurs de commandes UNIX tel que Bourne-Shell (voir `sh(1)`) ou C-Shell (voir `csh(1)`) peuvent être utilisés par Sun Grid Engine pour démarrer les scripts de travail. Les interpréteurs de commandes peuvent soit être démarrés comme des shells de connexion (c'est-à-dire que tous les fichiers de ressources par défaut du système et de l'utilisateur tels que `.login` ou `.profile` seront exécutés au démarrage de l'interpréteur de commandes et l'environnement pour le travail sera configuré comme si l'utilisateur venait de se connecter) ou juste pour l'exécution de la commande (c'est-à-dire que seuls des fichiers de ressources spécifiques de shell tels que `.cshrc` seront exécutés et qu'un environnement par défaut minimal sera configuré par Sun Grid Engine – voir `qsub(1)`). Le paramètre `login_shells` contient la liste séparée par des virgules des noms d'exécutables des interpréteurs de commandes à démarrer en tant que shells de connexion.

Changer `login_shells` a un effet immédiat. La valeur par défaut de `login_shells` est `sh, csh, tcsh, ksh`.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- `min_uid` (UID min.) – Ce paramètre définit une limite inférieure pour les ID d'utilisateur susceptibles d'utiliser la grappe. Les utilisateurs dont l'ID d'utilisateur (tel que retourné par `getpwnam(3)`) est inférieur à `min_uid` ne seront pas autorisés à exécuter des travaux sur la grappe.

Changer `min_uid` a un effet immédiat. La valeur par défaut de `min_uid` est 0.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *min_gid* (GID min.) – Ce paramètre définit une limite inférieure pour les ID de groupe susceptibles d'utiliser la grappe. Les utilisateurs dont l'ID de groupe par défaut (tel que retourné par `getpwnam(3)`) est inférieur à *min_gid* ne seront pas autorisés à exécuter des travaux sur la grappe.

Changer *min_gid* a un effet immédiat. La valeur par défaut de *min_gid* est 0.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *user_lists* (listes d'utilisateurs) – Ce paramètre contient une liste séparée par des virgules d'ACL (voir description dans `access_list(5)`). Tout utilisateur figurant dans au moins une des ACL listées a accès à la grappe. Si le paramètre *user_lists* est sur NONE (la valeur par défaut) tout utilisateur qui n'est pas explicitement exclu au moyen du paramètre *xuser_lists* décrit ci-après, a accès à la grappe. Si un utilisateur figure à la fois dans une ACL listée dans *xuser_lists* et dans une ACL listée dans *user_lists*, il se voit refuser l'accès à la grappe.

Changer *user_lists* a un effet immédiat

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *xuser_lists* (listes d'utilisateurs x) – Ce paramètre contient une liste séparée par des virgules d'ACL (voir description dans `access_list(5)`). Tout utilisateur figurant dans au moins une des ACL listées se voit refuser l'accès à la grappe. Si le paramètre *xuser_lists* est sur NONE (la valeur par défaut), tout utilisateur a accès. Si un utilisateur figure à la fois dans une ACL listée dans *xuser_lists* et dans une ACL listée dans *user_lists* (voir ci-dessus), il se voit refuser l'accès à la grappe.

Changer *xuser_lists* a un effet immédiat

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *administrator_mail* (mail administrateur) – Ce paramètre spécifie la liste séparée par des virgules des adresses de messagerie électronique du ou des administrateurs de la grappe auxquels les rapports relatifs aux problèmes générés en interne sont envoyés. Le format des messages dépend de votre système de messagerie et de sa configuration ; consultez le guide de configuration de votre système pour plus de détails.

Changer *administrator_mail* a un effet immédiat. La valeur par défaut de *administrator_mail* est une liste de messagerie vide.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *projects* (projets) – Ce paramètre n'est disponible que pour les systèmes Sun Grid Engine, Enterprise Edition. Il n'est pas présent dans Sun Grid Engine.

La liste *projects* contient tous les projets qui bénéficient de l'accès au système Sun Grid Engine, Enterprise Edition. Les utilisateurs qui n'appartiennent à aucun de ces projets ne peuvent pas utiliser le système Sun Grid Engine, Enterprise Edition. Si des utilisateurs relèvent de projets figurant à la fois dans la liste *projects* et dans la liste *xprojects* (voir ci-dessous), ils ne peuvent pas utiliser le système.

Changer *projects* a un effet immédiat. La valeur par défaut de *projects* est aucun. Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *xprojects* (projets x) – Ce paramètre n'est disponible que pour les systèmes Sun Grid Engine, Enterprise Edition. Il n'est pas présent dans Sun Grid Engine.

La liste *xprojects* contient tous les projets qui ne bénéficient pas de l'accès au système Sun Grid Engine, Enterprise Edition. Les utilisateurs qui appartiennent à ces projets ne peuvent pas utiliser le système Sun Grid Engine, Enterprise Edition. Si des utilisateurs relèvent de projets figurant à la fois dans la liste *projects* et la liste *xprojects* (voir ci-dessous), ils ne peuvent pas utiliser le système.

Changer *xprojects* a un effet immédiat. La valeur par défaut de *xprojects* est aucun. Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *load_report_time* (heure du rapport de charge) – La charge du système est rapportée périodiquement par les démons d'exécution à *sge_qmaster(8)*. Le paramètre *load_report_time* définit l'intervalle de temps entre deux rapports de charge.

Chaque *sge_execd(8)* peut utiliser une heure de rapport de charge différente. Changer *load_report_time* a un effet immédiat.

Remarque – Faites attention lorsque vous modifiez *load_report_time*. Des rapports de charge trop fréquents peuvent bloquer *sge_qmaster(8)*, en particulier si le nombre d'hôtes d'exécution est important. En sus, étant donné que la charge du système augmente et diminue en général doucement, les rapports de charge fréquents ne présentent guère d'avantages.

La valeur par défaut de *load_report_time* est 40 secondes.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *reschedule_unknown* (reprogrammation inconnue) – Détermine si les travaux sur les hôtes à l'état *unknown* sont reprogrammés et, par conséquent, envoyés à d'autres hôtes. Les hôtes sont enregistrés comme *unknown* si *sge_master(8)* ne peut pas établir de contact avec *sge_execd(8)* sur ces hôtes (voir *max_unheard*). Cela peut être dû à une panne de l'hôte ou de la connexion réseau, ou au fait que *sge_execd(8)* ne soit pas exécuté sur ces hôtes.

Dans tous les cas, Sun Grid Engine peut reprogrammer les travaux en cours sur ces hôtes sur un autre système. *reschedule_unknown* contrôle le délai respecté par Sun Grid Engine avant de reprogrammer les travaux après le passage d'un hôte à l'état *unknown*. Le format utilisé est *hh:mm:ss*. Si la valeur spéciale *00:00:00* est définie, les travaux ne sont pas reprogrammés.

La reprogrammation n'est lancée que pour les travaux dont l'indicateur de ré-exécution est activé (voir l'option `-r y` de `qsub(1)` et l'option `rerun` de `queue_conf(5)`). Les travaux parallèles ne sont reprogrammés que si l'hôte sur lequel leur tâche maître s'exécute est à l'état `unknown`. Les travaux de point de reprise ne seront reprogrammés que lorsque l'option de l'environnement de point de reprise correspondant contient l'indicateur approprié (voir `checkpoint(5)`). Les travaux interactifs (voir `qsh(1)`, `qrsh(1)`, `qtchsh(1)`) ne sont pas reprogrammés.

La valeur par défaut de `reschedule_unknown` est : `00:00:00`

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- ***stat_log_time*** (heure de consignation de stat) – Le logiciel Sun Grid Engine enregistre périodiquement un cliché du statut des files d'attente couramment configurées dans la grappe sur le disque. L'intervalle en secondes entre deux clichés consécutifs est défini par *stat_log_time*.

Changer *stat_log_time* a un effet immédiat. La valeur par défaut de *stat_log_time* est 2 heures 30 minutes.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- ***max_unheard*** (délai HC maximum) – Si `sge_qmaster(8)` n'a pas pu contacter ou n'a pas été contacté par le démon d'exécution d'un hôte pendant le délai *max_unheard* secondes, toutes les files d'attente résidant sur cet hôte prennent le statut inconnu. `sge_qmaster(8)`, au moins, doit être contacté par les démons d'exécution pour obtenir les rapports de charge. Par conséquent, *max_unheard* doit être supérieur à *load_report_time* (voir ci-dessus).

Changer *max_unheard* a un effet immédiat. La valeur par défaut de *max_unheard* est 2 minutes 30 secondes.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- ***loglevel*** (niveau de consignation) – Ce paramètre spécifie le niveau de détail que les composants de Sun Grid Engine tels que `sge_qmaster(8)` ou `sge_execd(8)` utilisent pour générer les messages d'information, avertissement ou erreur, qui sont consignés dans les fichiers `messages` dans les répertoires de spool des démons d'exécution et maître (voir description des paramètres *qmaster_spool_dir* et *execd_spool_dir* plus haut). Les niveaux de message disponibles sont les suivants :
 - `log_err` – Tous les événements de type erreur reconnus sont consignés.
 - `log_warning` – Tous les événements de type erreur reconnus et tous les signes de comportements erronés potentiels détectés sont consignés.
 - `log_info` – Tous les événements de type erreur reconnus, tous les signes de comportements erronés potentiels détectés et une variété de messages d'information sont consignés.

Changer *loglevel* a un effet immédiat.

La valeur par défaut de *loglevel* est `log_info`.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- ***max_aj_instances*** (instances de matrice de travail maximum) – Ce paramètre définit la quantité maximale de tâches de matrice pouvant être programmées pour s'exécuter simultanément par travail de matrice. Une instance d'une tâche de matrice sera créée au sein du démon maître lorsqu'il recevra un ordre de démarrage du programmeur. Cette instance sera détruite à la fin de la tâche de matrice. Par conséquent, ce paramètre fournit principalement un contrôle sur la consommation de mémoire des travaux de matrice dans le maître et le démon programmeur. Elle est surtout utile pour les grappes de très grande taille et les travaux de matrice très importants. La valeur par défaut de ce paramètre est 2000. La valeur 0 désactivera cette limite et permettra au programmeur de démarrer autant de tâches de travail de matrice qu'il y a de ressources adaptées disponibles dans la grappe.

Changer *max_aj_instances* a un effet immédiat. Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- ***max_aj_tasks*** (tâches de matrice de travail maximum) – Ce paramètre définit le nombre maximal de tâches de travail de matrice au sein d'un travail de matrice. `sge_qmaster(8)` refusera les soumissions de travaux de matrice qui requièrent plus de *max_aj_tasks* tâches de travail de matrice. La valeur par défaut de ce paramètre est 75 000. La valeur 0 désactivera cette limite.

Changer *max_aj_tasks* a un effet immédiat.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- ***max_u_jobs*** – Contrôle le nombre de travaux actifs (pas terminés) que chaque utilisateur Sun Grid Engine peut avoir dans le système simultanément. Une valeur supérieure à 0 définit la limite. La valeur par défaut 0 signifie « unlimited ». Si la limite *max_u_jobs* est dépassée par la soumission d'un travail, la commande de soumission se termine avec le statut 25 et un message d'erreur approprié.

Changer *max_u_jobs* a un effet immédiat.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- ***max_jobs*** (travaux max.) – Contrôle le nombre de travaux actifs (pas terminés) autorisés simultanément dans un système Sun Grid Engine. Une valeur supérieure à 0 définit la limite. La valeur par défaut 0 signifie « illimité ». Si la limite *max_u_jobs* est dépassée par la soumission d'un travail, la commande de soumission se termine avec le statut 25 et un message d'erreur approprié.

Changer *max_jobs* a un effet immédiat.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *enforce_project* (forcer projet) – Ce paramètre n'est disponible que pour les systèmes Sun Grid Engine, Enterprise Edition. Il n'est pas présent dans Sun Grid Engine.

S'il est sur *true*, il est demandé aux utilisateurs de demander un projet à chaque fois qu'ils soumettent un travail. Voir l'option *-P* de *qsub(1)* pour plus de détails.

Changer *enforce_project* a un effet immédiat. La valeur par défaut de *enforce_project* est *false*.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *enforce_user* (forcer utilisateur) – Ce paramètre n'est disponible que pour les systèmes Sun Grid Engine, Enterprise Edition. Il n'est pas présent dans Sun Grid Engine.

S'il est sur *true*, un utilisateur *user(5)* Sun Grid Engine, Enterprise Edition doit exister pour permettre la soumission de travaux. Les travaux sont refusés s'il n'existe pas d'utilisateur correspondant.

Changer *enforce_user* a un effet immédiat. La valeur par défaut de *enforce_user* est *false*.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *set_token_cmd* (commande *set_token*) – Ce paramètre n'est présent que si la licence de votre système Sun Grid Engine inclut la prise en charge d'AFS.

Set_token_cmd pointe sur une commande qui définit et étend les jetons AFS pour les travaux Sun Grid Engine. Dans la distribution AFS standard de Sun Grid Engine, il est fourni sous la forme d'un script qui attend deux paramètres de ligne de commande. Il lit le jeton dans *STDIN*, étend l'heure d'échéance du jeton et définit le jeton :

```
<cde_définition_jeton> <utilisateur> <jeton_étendu_après_secondes>
```

À l'instar d'un script shell, cette commande appellera les programmes :

- *SetToken*
- *forge*

qui sont fournis par votre distributeur sous forme de code source. Le script ressemble à ce qui suit :

```
#!/bin/sh
# set_token_cmd
forge -u $1 -t $2 | SetToken
```

Étant donné qu'il est nécessaire que `forge` lise la clé secrète du serveur AFS, un site peut vouloir remplacer le script `set_token_cmd` par une commande, qui assure la connexion avec un démon auto-écrit sur le serveur AFS. Le jeton doit être forgé sur le serveur AFS et retourné à la machine locale où `SetToken` est exécuté.

Changer `set_token_cmd` a un effet immédiat. La valeur par défaut de `set_token_cmd` est aucun.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- `pag_cmd` (commande pag) – Ce paramètre n'est présent que si la licence de votre système Sun Grid Engine inclut la prise en charge d'AFS.

Le chemin de votre `pagsh` est spécifié via ce paramètre. Le processus `sge_shepherd(8)` et le travail s'exécutent dans un `pagsh`. Pour plus de détails, consultez votre administrateur AFS.

Changer `pag_cmd` a un effet immédiat. La valeur par défaut de `pag_cmd` est aucun.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- `token_extend_time` (temps étendu jeton) – Ce paramètre n'est présent que si la licence de votre système Sun Grid Engine inclut la prise en charge d'AFS.

Ce paramètre définit le délai pendant lequel les jetons AFS sont périodiquement étendus. Sun Grid Engine appellera l'extension des jetons 30 minutes avant l'arrivée à échéance des jetons jusqu'à ce que les travaux soient terminés et que les jetons correspondants ne soient plus requis.

Changer `token_extend_time` a un effet immédiat. La valeur par défaut de `token_extend_time` est 24:0:0, soit 24 heures.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- `gid_range` (plage GID) – Ce paramètre n'est disponible que pour les systèmes Sun Grid Engine, Enterprise Edition. Il n'est pas présent dans Sun Grid Engine.

`gid_range` est une liste séparée par des virgules d'expressions de plage de la forme `n-m` (`n` et `m` sont des entiers positifs non-nuls), `m` étant une abréviation de `m-m`. Ces nombres sont utilisés dans `sge_execd(8)` pour identifier les processus appartenant au même travail.

Chaque `sge_execd(8)` peut utiliser un ensemble séparé d'ID de groupes dans cet objectif. Tous les nombres rentrant dans la plage d'ID de groupes doivent être des ID de groupe supplémentaires inutilisés sur le système où `sge_execd(8)` est lancé.

Changer `gid_range` a un effet immédiat. Il n'y a pas de valeur par défaut pour `gid_range`. L'administrateur devra attribuer une valeur pour `gid_range` pendant l'installation de Sun Grid Engine, Enterprise Edition.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *qmaster_params* (paramètres du maître) – Une liste de paramètres supplémentaires peut être transmise au `qmaster` de Sun Grid Engine. Les valeurs suivantes sont reconnues :
 - `ENABLE_FORCED_QDEL` – Si ce paramètre est défini, les utilisateurs non-administratifs peuvent forcer la suppression de leurs propres travaux via l'option `-f` de `qdel(1)`. Sans ce paramètre, la suppression forcée des travaux n'est autorisée que par le gestionnaire ou l'opérateur de Sun Grid Engine.

Remarque – La suppression forcée des travaux est exécutée différemment selon si les utilisateurs sont des administrateurs de Sun Grid Engine ou non. Dans le cas d'utilisateurs administratifs, les travaux sont supprimés immédiatement de la base de données interne de Sun Grid Engine. Pour les utilisateurs normaux, l'équivalent d'une `qdel(1)` normale est d'abord exécuté et la suppression n'est forcée que si l'annulation normale échoue.

Changer *qmaster_params* a un effet immédiat. La valeur par défaut de *qmaster_params* est `none`.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- `FORBID_RESCHEDULE` – Si ce paramètre est défini, la remise en file d'attente des travaux ne peut pas être commencée par le script de travail qui est sous le contrôle de l'utilisateur. Sans ce paramètre les travaux qui retournent la valeur 99 sont reprogrammés. Cela peut être utilisé pour obliger le travail à être redémarré sur une autre machine, par exemple s'il n'y a pas suffisamment de ressources sur la machine courante.
- `DISABLE_AUTO_RESCHEDULING` – S'il est mis `true` ou `1`, le paramètre *reschedule_unknown* n'est pas pris en compte.

Changer *qmaster_params* a un effet immédiat. La valeur par défaut de *qmaster_params* est `none`.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *schedd_params* (paramètres Schedd) – Ce paramètre est prévu pour transférer des paramètres supplémentaires au programmeur de Sun Grid Engine. Les valeurs suivantes sont actuellement reconnues :
 - `FLUSH_SUBMIT_SEC`, `FLUSH_FINISH_SEC` – Ces paramètres sont fournis pour régler le comportement de programmation du système. Par défaut, un cycle du programmeur est déclenché dans l'intervalle de programmeur qui est défini dans la configuration du programmeur `sched_conf(5)`, paramètre *schedule_interval*.

Les paramètres `FLUSH_SUBMIT_SEC/FLUSH_FINISH_SEC` définissent les intervalles de temps entre le déclenchement d'un cycle du programmeur et la soumission/fin d'un travail.

La réaction la plus immédiate du programmeur peut être activée en mettant les deux valeurs sur 0. Le comportement de programmation par défaut est imposé soit en retirant les paramètres soit en les mettant sur la valeur -1.

- `CLASSIC_SGEEE_SCHEDULING` – Si ce paramètre est sur `true` ou 1, l'algorithme de programmation des travaux en attente original de Sun Grid Engine, Enterprise Edition est utilisé. L'algorithme de programmation des travaux en attente original de Sun Grid Engine, Enterprise Edition calcule les tickets pour les travaux en attente en divisant les tickets de chaque politique de programmation entre les travaux actifs et en attente. La liste des travaux en attente est ensuite classée en fonction des tickets attribués aux travaux en attente. La valeur par défaut est `false`. Ce paramètre n'est valide que dans un système Sun Grid Engine, Enterprise Edition.
- `POLICY_HIERARCHY` – Ce paramètre définit une chaîne de dépendances des politiques. Chaque politique figurant dans cette chaîne de dépendances est influencée par les politiques précédentes et influence les suivantes. Un scénario type consiste à attribuer la précedence à la politique de remplacement sur la politique basée sur le partage. La politique de remplacement détermine dans ce cas comment les tickets basés sur le partage sont attribués entre les travaux d'un même utilisateur ou projet. Vous remarquerez que toutes les politiques contribuent au total des tickets attribués à un travail donné quelle que soit la hiérarchie de politiques définie. Les tickets calculés dans chaque politique peuvent être différents selon `POLICY_HIERARCHY`.

Le paramètre `POLICY_HIERARCHY` peut être une combinaison de jusqu'à 4 des premières lettres des noms anglais des politiques `S`(basée sur le partage), `F`(fonctionnelle), `D`(initiation limite) et `O`(remplacement). Ainsi `OFSD` signifie que la politique de remplacement a la précedence sur la politique fonctionnelle, qui influence la politique basée sur le partage, qui précède la politique d'initiation limite. S'il y a moins de quatre lettres, cela signifie que certaines des politiques n'influencent pas les autres et ne sont pas influencées par celles-ci. Ainsi, `FS` signifie que la politique fonctionnelle influence la politique basée sur le partage et qu'il n'y a pas d'autres interférences avec d'autres politiques.

La valeur spéciale `NONE` désactive les hiérarchies de politiques.

- `SHARE_OVERRIDE_TICKETS` – S'il est sur `true` ou 1, les tickets de remplacement de toute instance d'objet de remplacement sont partagés de façon équitable entre les travaux associés à l'objet. S'il est sur `false` ou 0, chaque travail reçoit la valeur complète des tickets de remplacement associés à l'objet. La valeur par défaut est `true`. Ce paramètre n'est valide que dans un système Sun Grid Engine, Enterprise Edition.
- `SHARE_DEADLINE_TICKETS` – S'il est sur `true` ou 1, le total des tickets deadline est partagé entre tous les travaux deadline. S'il est sur `false` ou 0, chaque travail avec limite recevra le total des tickets d'initiation limite une fois que l'initiation limite du travail atteinte. La valeur par défaut est `true`. Ce paramètre n'est valide que dans un système Sun Grid Engine, Enterprise Edition.

- `MAX_FUNCTIONAL_JOBS_TO_SCHEDULE` – Il s’agit du nombre maximal de travaux en attente à programmer dans la politique fonctionnelle. La valeur par défaut est 200. Ce paramètre n’est valide que dans un système Sun Grid Engine, Enterprise Edition.
- `MAX_PENDING_TASKS_PER_JOB` – Le nombre maximal de sous-tâches par travail de matrice en attente à programmer. Ce paramètre existe pour réduire la surcharge due à la programmation. La valeur par défaut est 50. Ce paramètre n’est valide que dans un système Sun Grid Engine, Enterprise Edition.
- `PROFILE` – Si défini, le programmeur enregistre des informations dans des profils qui résument chaque cycle de programmation.

Changer `schedd_params` a un effet immédiat. La valeur par défaut de `schedd_params` est none (aucune).

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l’hôte d’exécution.

- `execd_params` (paramètres `execd`) – Ce paramètre a été prévu pour transmettre des paramètres supplémentaires au démon d’exécution de Sun Grid Engine. Les valeurs suivantes sont reconnues :
 - `ACCT_RESERVED_USAGE` – Si ce paramètre est sur `true`, l’utilisation réservée est utilisée pour les entrées de comptabilité `cpu`, `mem` et `io` au lieu de l’utilisation mesurée.
 - `KEEP_ACTIVE` – Cette valeur ne devrait être définie que pour le débogage. Si elle est sur `true`, le démon d’exécution ne supprimera pas le répertoire de spool tenu par `sge_shepherd(8)` pour un travail.
 - `NO_REPRIORITIZATION` – Si ce paramètre est sur `true` ou 1, les priorités des travaux ne sont pas automatiquement recalculées. Sans paramétrer cet attribut, le paramètre de priorité figurant dans `sched_conf(5)` n’a pas d’effet durable dans un système Sun Grid Engine, Enterprise Edition à cause du recalcul dynamique des priorités effectué par le module PTF au sein de `sge_execd(8)`.
 - `PTF_MIN_PRIORITY`, `PTF_MAX_PRIORITY` – Ces paramètres sont uniquement disponibles dans un système Sun Grid Engine, Enterprise Edition.

Priorité maximale/minimale que Sun Grid Engine, Enterprise Edition attribuera à un travail. Il s’agit typiquement d’une valeur négative/positive comprise dans une plage de -20 (maximum) à 19 (minimum) pour les systèmes qui autorisent la fixation de priorités avec l’appel système `nice(2)`. D’autres systèmes peuvent fournir d’autres plages.

La plage de priorités par défaut (qui varie d’un système à l’autre) est installée soit en retirant les paramètres soit en les mettant sur -999.

Consultez le fichier `messages` du démon d’exécution pour la valeur par défaut prédéfinie sur vos hôtes. Ces valeurs sont consignées pendant le démarrage du démon d’exécution.

- NOTIFY_KILL – Ce paramètre vous permet de changer le signal d’avertissement pour le signal SIGKILL (voir l’option `-notify` de `qsub(1)`). Il accepte selon le cas les noms de signaux (utilisez l’option `-l` de `kill(1)`) ou la valeur spéciale `none`. S’il est sur `none`, aucun signal d’avertissement ne sera envoyé. S’il est sur `TERM`, par exemple, ou un autre nom de signal, ce signal sera envoyé à titre de signal d’avertissement.
- NOTIFY_SUSP – Avec ce paramètre il est possible de modifier le signal d’avertissement pour le signal SIGSTOP (voir le paramètre `-notify` de `qsub(1)`). Il accepte selon le cas les noms de signaux (utilisez l’option `-l` de `kill(1)`) ou la valeur spéciale `none`. S’il est sur `none`, aucun signal d’avertissement ne sera envoyé. S’il est sur `TSTP`, par exemple, ou un autre nom de signal, ce signal sera envoyé à titre de signal d’avertissement.
- SET_SGE_ENV, SET_COD_ENV, SET_GRD_ENV – Dans `qsub(1)` vous pouvez trouver la liste des variables d’environnement qui sont exportées dans l’environnement d’exécution de chaque travail Sun Grid Engine, Enterprise Edition. Certains de ces noms de variables commencent par le préfixe `SGE_`. Dans les produits qui ont précédé Sun Grid Engine, Enterprise Edition (Codine, GRD), les préfixes `COD_` et `GRD_` étaient utilisés pour ces variables. Cela rend nécessaire la mise à jour de tous les scripts de travail se basant sur l’une de ces variables d’environnement. Les paramètres, `SET_SGE_ENV`, `SET_COD_ENV` et `SET_GRD_ENV` peuvent être utilisés pour faciliter la transition de Codine/GRD à Sun Grid Engine, Enterprise Edition.

Chacun de ces paramètres assure qu’un ensemble de variables d’environnement mentionné dans `qsub(1)` et commençant par le préfixe correspondant sera exporté dans l’environnement du travail. Chaque paramètre peut être fixé sur `true`, `1` ou `false`, `0`. Les valeurs par défaut sont `SET_SGE_ENV=1`, `SET_COD_ENV=0` et `SET_GRD_ENV=0`.

Si ces trois paramètres sont sur `false` ou `0`, les travaux risquent d’échouer vu qu’aucune des variables d’environnement correspondantes ne sera définie. Par conséquent, dans ce cas, les variables d’environnement commençant par le préfixe `SGE_` sont définies.

- SHARETREE_RESERVED_USAGE – Si ce paramètre est sur `true`, l’utilisation réservée est prise pour la consommation de l’arborescence des partages de Sun Grid Engine, Enterprise Edition au lieu de l’utilisation mesurée.
- USE_QSUB_GID – Si ce paramètre est sur `true`, l’ID de groupe principal actif lorsqu’un travail a été soumis sera défini pour devenir l’ID de groupe principal pour l’exécution des travaux. Si ce paramètre n’est pas défini, l’ID de groupe principal défini pour le propriétaire du travail dans le fichier `passwd(5)` de l’hôte d’exécution est utilisé.

Cette fonctionnalité est uniquement disponible pour les travaux soumis via `qsub(1)`, `qrsh(1)`, `qmake(1)` et `qtcsh(1)`. En sus, elle fonctionne uniquement pour les travaux `qrsh(1)` (et par voie de conséquence aussi pour `qtcsh(1)` et `qmake(1)`) si les composants `rsh` et `rshd` fournis avec Sun Grid Engine sont utilisés (c’est-à-dire que les paramètres `rsh_daemon` et `rsh_command` doivent obligatoirement conserver leurs valeurs par défaut).

Changer *execd_params* a un effet immédiat. La valeur par défaut de *execd_params* est *none*.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *admin_user* (utilisateur admin) – Ce paramètre est le compte d'utilisateur administratif utilisé par Sun Grid Engine pour toute la gestion en interne des fichiers (mise en attente de statut, consignation des messages, etc.). Il peut être utilisé dans les cas où le compte de super-utilisateur ne dispose pas des droits d'accès aux fichiers correspondants (par ex. sur un système de fichiers sans accès lecture/écriture de super-utilisateur global).

Étant donné qu'il s'agit d'un paramètre défini au moment de l'installation, changer le paramètre *admin_user* sur un système en fonctionnement n'est pas pris en charge. Le changer sur une grappe arrêtée est possible, mais si l'accès à la zone de spool de Sun Grid Engine est interrompu, cela se traduira par un comportement imprévisible.

Le paramètre *admin_user* n'a pas de valeur par défaut, mais est défini pendant la procédure d'installation du maître.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *finished_jobs* (travaux terminés) – Le logiciel Sun Grid Engine stocke un certain nombre de travaux *juste terminés* pour fournir des informations de statut post mortem. Le paramètre *finished_jobs* définit le nombre de travaux terminés stockés. Si ce nombre maximal est atteint, le travail fini le plus ancien est éliminé à chaque fois qu'un travail venant d'être fini est ajouté à la liste des travaux terminés.

Tout changement apporté à *finished_jobs* est immédiatement appliqué. La valeur par défaut de *finished_jobs* est 0.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *qlogin_daemon* (démon qlogin) – Ce paramètre spécifie l'exécutable qui va être démarré sur le côté serveur d'une requête *qlogin(1)*. Il s'agit en général du chemin complet du démon *telnet* du système. Si aucune valeur n'est donnée, un composant spécialisé de Sun Grid Engine est utilisé.

Changer *qlogin_daemon* a un effet immédiat. La valeur par défaut de *qlogin_daemon* est *none*.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *qlogin_command* (commande qlogin) – Commande qui va être exécutée sur le côté client d'une requête *qlogin(1)*. Il s'agit en général du chemin complet du programme client *telnet* du système. Si aucune valeur n'est donnée, un composant spécialisé de Sun Grid Engine est utilisé. Elle est automatiquement démarrée avec l'hôte et le numéro de port cibles en tant que paramètres.

Changer *qlogin_command* a un effet immédiat. La valeur par défaut de *qlogin_command* est *none*.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *rlogin_daemon* (démon rlogin) – Ce paramètre spécifie l'exécutable qui va être démarré sur le côté serveur d'une requête `qrsh(1)` sans argument de commande pour être exécuté à distance. Il s'agit en général du chemin complet du démon `rlogin` du système. Si aucune valeur n'est donnée, un composant spécialisé de Sun Grid Engine est utilisé.

Changer *rlogin_daemon* a un effet immédiat. La valeur par défaut de *rlogin_daemon* est `none`.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *rlogin_command* (commande rlogin) – Commande qui va être exécutée sur le côté client d'une requête `qrsh(1)` sans argument de commande pour être exécutée à distance. Il s'agit en général du chemin complet du programme client `rlogin` du système. Si aucune valeur n'est donnée, un composant spécialisé de Sun Grid Engine est utilisé. La commande est automatiquement démarrée avec l'hôte et le numéro de port cibles en tant que paramètres comme requis pour `telnet(1)`. Le client `rlogin` de Sun Grid Engine a été étendu pour accepter et utiliser l'argument numéro de port. Vous pouvez uniquement utiliser des clients, tels que `ssh`, qui comprennent également cette syntaxe.

Changer *rlogin_command* a un effet immédiat. La valeur par défaut de *rlogin_command* est `none`.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *rsh_daemon* (démon rsh) – Ce paramètre spécifie l'exécutable qui va être démarré sur le côté serveur d'une requête `qrsh(1)` avec un argument de commande pour être exécuté à distance. Il s'agit en général du chemin complet du démon `rsh` du système. Si aucune valeur n'est donnée, un composant spécialisé de Sun Grid Engine est utilisé.

Changer *rsh_daemon* a un effet immédiat. La valeur par défaut de *rsh_daemon* est `none`.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *rsh_command* (commande rsh) – Commande à exécuter sur le côté client d'un `qrsh(1)` avec un argument de commande pour être exécuté à distance. Il s'agit en général du chemin complet du programme client `rsh` du système. Si aucune valeur n'est donnée, un composant spécialisé de Sun Grid Engine est utilisé. La commande est automatiquement démarrée avec l'hôte et le numéro de port cibles en tant que paramètres comme requis pour `telnet(1)` plus la commande avec ses arguments pour être exécutée à distance. Le client `rsh` de Sun Grid Engine a été étendu pour accepter et utiliser l'argument numéro de port. Vous pouvez uniquement utiliser des clients, tels que `ssh`, qui comprennent également cette syntaxe.

Changer *rsh_command* a un effet immédiat. La valeur par défaut de *rsh_command* est *none*.

L'entrée de configuration globale relative à cette valeur peut être surécrite par la configuration locale de l'hôte d'exécution.

- *ignore_fqdn* (ignorer les noms de domaine complet) – Ce paramètre fait que le système ignore le composant nom de domaine complet des noms d'hôte. Ce paramètre doit être défini si tous les hôtes appartenant à une grappe Sun Grid Engine font partie d'un unique domaine DNS. Il est activé s'il est sur *true* ou *1*. L'activer peut résoudre des problèmes survenant au niveau des rapports de charge à cause de résolutions de noms d'hôtes différentes au sein de la grappe.

Étant donné qu'il s'agit d'un paramètre défini au moment de l'installation, changer le paramètre *ignore_fqdn* sur un système en fonctionnement n'est pas pris en charge. La valeur par défaut de *ignore_fqdn* est *true*.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

- *default_domain* (domaine par défaut) – Ce paramètre n'est nécessaire que si votre grappe Sun Grid Engine comprend des hôtes appartenant à plusieurs domaines DNS. Dans ce cas, il peut être utilisé si votre résolution de noms d'hôtes donne à la fois des noms d'hôte complets et non pour les hôtes dans un des domaines DNS. La valeur de *default_domain* est rattachée à un nom d'hôte incomplet pour constituer un nom d'hôte complet. Le paramètre *default_domain* n'aura aucun effet si *ignore_fqdn* est sur *true*.

Étant donné qu'il s'agit d'un paramètre défini au moment de l'installation, changer le paramètre *default_domain* sur un système en fonctionnement n'est pas pris en charge. La valeur par défaut de *default_domain* est *none*, cas dans lequel il ne sera pas utilisé.

Cette valeur est uniquement un paramètre de configuration globale. Elle ne peut pas être surécrite par la configuration locale de l'hôte d'exécution.

Voir aussi

sg_intro(1), *csh(1)*, *qconf(1)*, *qsub(1)*, *rsh(1)*, *sh(1)*, *getpwnam(3)*, *queue_conf(5)*, *sched_conf(5)*, *sg_execd(8)*, *sg_qmaster(8)*, *sg_shepherd(8)*, *cron(8)*, *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3*.

Droits d'auteur

Voir *sg_intro(1)* pour l'énoncé complet des droits et autorisations.

sg_e_h_aliases(5)

Nom

`sg_e_h_aliases` - Format des fichiers d'alias d'hôtes de Sun Grid Engine.

Description

Tous les composants de Sun Grid Engine utilisent un service de résolution de noms fourni par `sg_e_commd(8)` pour identifier les hôtes au moyen d'un nom d'hôte. `sg_e_commd(8)` lui-même référence les services d'annuaire UNIX standards tels que DNS, NIS et `/etc/hôtes` pour résoudre les noms d'hôtes. Dans de rares cas, ces services standard ne peuvent pas être configurés correctement et les démons de communication de Sun Grid Engine s'exécutant sur des hôtes différents sont dans l'impossibilité de déterminer automatiquement un nom d'hôte unique pour un ou tous les hôtes, qui puisse être utilisé sur tous les hôtes. Dans de tels cas, un fichier d'alias de noms d'hôtes Sun Grid Engine peut être utilisé pour fournir aux démons de communication une base de données de résolution de noms d'hôtes privée et consistante.

L'emplacement par défaut du fichier d'alias de noms d'hôtes est :

`<racine_sg_e>/<cellule>/common/host_aliases`

Pour utiliser un autre fichier d'alias de noms d'hôtes, vous devez le spécifier de façon explicite à `sg_e_commd(8)` via l'option de ligne de commande `-a`.

Format

Pour chaque hôte, une unique ligne qui précise les alias de noms d'hôtes sous la forme d'une liste séparée par des blancs, des virgules ou des points-virgules est requise. Le premier alias est défini comme étant le nom d'hôte unique qui sera utilisé par tous les composants de Sun Grid Engine utilisant le service d'alias de noms d'hôtes de `sg_e_commd(8)`.

Voir aussi

`sg_e_intro(1)`, `sg_e_commd(8)`

Droits d'auteur

Voir `sg_e_intro(1)` pour l'énoncé complet des droits et autorisations.

sgc_pe (5)

Nom

sgc_pe – Format des fichiers de configuration d'environnement parallèle de Sun Grid Engine.

Description

Les environnements parallèles sont des environnements d'exécution et de programmation parallèles qui autorisent l'exécution d'applications parallélisées à mémoire partagée ou mémoire distribuée. Les environnements parallèles requièrent en général un minimum de configuration pour être opérationnels avant de démarrer des applications parallèles. Des exemples d'environnements parallèles courants sont les systèmes d'exploitation parallèles à mémoire partagée et les environnements à mémoire distribuée Parallel Virtual Machine (PVM) et Message Passing Interface (MPI).

sgc_pe permet de définir des interfaces vers des environnements parallèles arbitraires. Une fois qu'un environnement parallèle est défini ou modifié avec les options `-ap` ou `-mp` de `qconf(1)`, cet environnement peut être demandé pour un travail via le modificateur `-pe` de `qsub(1)` avec une requête de plage pour le nombre de processus parallèles devant être alloués par le travail. Des options `-l` supplémentaires peuvent être utilisées pour préciser encore les exigences du travail.

Format

Le format d'un fichier *sgc_pe* est le suivant :

- *pe_name* (nom du PE) – Nom de l'environnement parallèle. Sera utilisé dans le modificateur `qsub(1) -pe`.
- *queue_list* (liste des files d'attente) – Liste séparée par des virgules des files d'attente auxquelles les travaux parallèles appartenant à cet environnement parallèle ont accès.
- *slots* (emplacements) – Nombre de processus parallèles autorisés au total à s'exécuter simultanément dans l'environnement parallèle.
- *user_lists* (listes d'utilisateurs) – Liste séparée par des virgules des noms des ACL (voir description dans `access_list(5)`). Tout utilisateur figurant dans au moins une des ACL listées a accès à l'environnement parallèle. Si le paramètre *user_lists* est sur NONE (la valeur par défaut) tout utilisateur qui n'est pas explicitement

exclu via le paramètre *xuser_lists* décrit ci-après, a accès. Si un utilisateur figure à la fois dans une ACL listée dans *xuser_lists* et dans une ACL listée dans *user_lists*, il se voit refuser l'accès à l'environnement parallèle.

- *xuser_lists* (listes d'utilisateurs X) – Le paramètre *xuser_lists* contient une liste séparée par des virgules des ACL (voir description dans `access_list(5)`). Tout utilisateur figurant dans au moins une des ACL listées n'est pas autorisé à accéder à l'environnement parallèle. Si le paramètre *xuser_lists* est sur NONE (la valeur par défaut) tout utilisateur a accès. Si un utilisateur figure à la fois dans une ACL listée dans *xuser_lists* et dans une ACL listée dans *user_lists*, il se voit refuser l'accès à l'environnement parallèle.
- *start_proc_args* (arg. procédure de démarrage)– Ligne de commande d'appel d'une procédure de démarrage pour l'environnement parallèle. La procédure de démarrage est appelée par `sge_shepherd(8)` avant l'exécution du script du travail. Son objectif est de configurer l'environnement parallèle en fonction de ses besoins. Un préfixe « utilisateur@ » optionnel spécifie l'utilisateur sous lequel cette procédure va être démarrée. La sortie standard de la procédure de démarrage est réacheminée sur le fichier *NomREQ.poIDT* dans le répertoire de travail du travail (voir `qsub(1)`), *NomREQ* étant le nom du travail tel qu'affiché par `qstat(1)` et *IDT* le numéro d'identification du travail. Similairement, la sortie d'erreur standard est réacheminée sur *NomREQ.peIDT*.

Les variables spéciales suivantes qui sont développées au moment de l'exécution peuvent être utilisées (en plus de toutes autres chaînes devant être interprétées par les procédures de démarrage et d'arrêt) pour constituer une ligne de commande :

- *\$pe_hostfile* – Chemin d'un fichier contenant une description détaillée de l'agencement de l'environnement à configurer pour la procédure de démarrage. Chaque ligne du fichier fait référence à un hôte sur lequel des processus parallèles vont être exécutés. La première entrée de chaque ligne indique le nom d'hôte, la seconde le nombre de processus parallèles à exécuter sur l'hôte et la troisième une plage de processeurs à utiliser en cas de machines multiprocesseurs.
- *\$shost* – Nom de l'hôte sur lequel les procédures de démarrage ou d'arrêt sont démarrées.
- *\$job_owner* – Nom d'utilisateur du propriétaire du travail.
- *\$job_id* – Numéro d'identification unique du travail de Sun Grid Engine.
- *\$job_name* – Nom du travail.
- *\$pe* – Nom de l'environnement parallèle utilisé.
- *\$pe_slots* – Nombre d'emplacements octroyés pour le travail.
- *\$processors* – Chaîne *processors* telle que contenue dans la configuration de file (voir `queue_conf(5)`) de la file maîtresse (la file dans laquelle les procédures de démarrage et d'arrêt sont démarrées).
- *\$queue* – La File maîtresse, c'est à dire celle dans laquelle les procédures de démarrage et d'arrêt sont démarrées.

- *stop_proc_args* (arg. procédure d'arrêt) – Ligne de commande d'appel d'une procédure d'arrêt pour l'environnement parallèle. La procédure d'arrêt est appelée par *sge_shepherd(8)* après la fin du travail. Son objectif est d'arrêter l'EP et de le retirer de tous les systèmes participant. Un préfixe « utilisateur@ » optionnel spécifie l'utilisateur sous lequel cette procédure va être démarrée. La sortie standard de la procédure d'arrêt est aussi réacheminée sur le fichier *NomREQ.poIDT* dans le répertoire de travail du travail (voir *qsub(1)*), *NomREQ* étant le nom du travail tel qu'affiché par *qstat(1)* et *IDT* le numéro d'identification du travail. Similairement, la sortie d'erreur standard est réacheminée sur *NomREQ.peIDT*.

Les mêmes variables spéciales que pour *start_proc_args* peuvent être utilisées pour constituer une ligne de commande.

- *signal_proc_args* – Ligne de commande d'appel d'une procédure de signalisation pour l'environnement parallèle. La procédure de signalisation est appelée par *sge_shepherd(8)* à chaque fois qu'un signal est envoyé au travail parallèle via *qmod(1)*, *qdel(1)* ou dans le cas d'une requête de migration. Son objectif est d'avertir les composants de l'environnement parallèle et leurs processus d'applications associés de façon adéquate. La sortie standard de la procédure de signalisation est réacheminée sur le fichier *NomREQ.poIDT* dans le répertoire de travail du travail (voir *qsub(1)*), *NomREQ* étant le nom du travail tel qu'affiché par *qstat(1)* et *IDT* le numéro d'identification du travail. Similairement, la sortie d'erreur standard est réacheminée sur *NomREQ.peIDT*.

Les mêmes variables spéciales que pour *start_proc_args* peuvent être utilisées pour constituer une ligne de commande.

- *allocation_rule* (règle d'allocation) – La règle d'allocation est interprétée par *sge_schedd(8)* et aide le programmeur à décider comment distribuer les processus parallèles entre les machines disponibles. Si, par exemple, un environnement parallèle est construit uniquement pour des applications à mémoire partagée, tous les processus parallèles doivent être attribués à une unique machine, quelque soit le nombre de machines adaptées disponibles. Si en revanche, l'environnement parallèle suit le paradigme de la mémoire distribuée, une distribution équitable des processus entre les machines peut être avantageuse.

La version courante du programmeur comprend uniquement les règles d'allocation suivantes :

- *<int>* – Nombre entier qui fixe le nombre de processus par hôte. Si ce nombre est 1, tous les processus doivent résider sur des hôtes différents. Si le dénominateur spécial *\$pe_slots* est utilisé, la plage complète de processus telle que spécifiée avec le modificateur *qsub(1) -pe* doit être allouée sur un même hôte (quelle que soit la valeur appartenant à la plage choisie pour le travail pour être allouée).
- *\$fill_up* – Tous les emplacements disponibles sont alloués en commençant par l'hôte/la file le/la plus adapté(e). Les hôtes et files d'attente supplémentaires sont « remplis » au fur et à mesure si le travail nécessite davantage d'emplacements pour les tâches parallèles.

- `$round_robin`– Depuis tous les hôtes appropriés, un emplacement est alloué jusqu'à ce que toutes les tâches requises par le travail parallèle soient réparties. Si davantage de tâches qu'il n'y a d'hôtes sont demandées, l'allocation recommence à partir du premier hôte, toujours par ordre d'appropriation décroissante.
- `control_slaves` (esclaves de contrôle) – Ce paramètre peut être fixé sur TRUE ou FALSE (la valeur par défaut). Il indique si Sun Grid Engine est le créateur des tâches esclaves d'une application parallèle via `sge_execd(8)` et `sge_shepherd(8)` et, par conséquent, a le contrôle complet de tous les processus dans une application parallèle, qui active certaines fonctionnalités telles que la limitation des ressources et une comptabilité correcte. Toutefois, pour prendre le contrôle des tâches esclaves d'une application parallèle, une interface PE sophistiquée, qui fonctionne en étroite collaboration avec les utilitaires de Sun Grid Engine, est requise. Des interfaces PE de ce type sont disponibles auprès de votre bureau de support Sun Grid Engine local.

Veillez mettre le paramètre `control_slaves` sur `false` pour toutes les autres interfaces PE.

- `job_is_first_task` (travail est première tâche) – Ce paramètre n'est contrôlé que si `control_slaves` (voir ci-dessus) est sur TRUE et que, par conséquent, Sun Grid Engine est le créateur des tâches esclaves d'une application parallèle via `sge_execd(8)` et `sge_shepherd(8)`. Dans ce cas, une interface PE sophistiquée, qui mette en étroite collaboration l'environnement parallèle et Sun Grid Engine, est requise. La documentation qui accompagne ces interfaces PE préconisera le paramétrage de `job_is_first_task`.

Le paramètre `job_is_first_task` peut être fixé sur TRUE ou FALSE. La valeur TRUE indique que le script du travail Sun Grid Engine contient déjà une des tâches de l'application parallèle, tandis que la valeur FALSE indique que le script du travail (et ses processus fils) ne fait pas partie du programme parallèle.

Restrictions

Remarque – La fonctionnalité des procédures de démarrage, d'arrêt et de signalisation reste sous la pleine responsabilité de l'administrateur qui configure l'environnement parallèle. Sun Grid Engine se limitera à appeler ces procédures et à en évaluer le statut de sortie. Sun Grid Engine n'a aucun moyen de détecter le fait que ces procédures n'effectuent pas leurs tâches correctement ou que l'environnement parallèle ou l'application parallèle se comportent de façon inattendue.

Voir aussi

`sge_intro(1)`, `qconf(1)`, `qdel(1)`, `qmod(1)`, `qsub(1)`, `access_list(5)`, `sge_qmaster(8)`, `sge_schedd(8)`, `sge_shepherd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

complex(5)

Nom

`complex` - Format des fichiers de configuration de complexes de Sun Grid Engine.

Description

Complex reflète le format de la configuration des complexes de Sun Grid Engine. La définition de complexes fournit toutes les informations adéquates sur les attributs de ressource qu'un utilisateur peut demander pour un travail Sun Grid Engine via l'option `qsub(1) -l` et pour l'interprétation de ces paramètres au sein du système Sun Grid Engine.

Il ne faut pas accéder directement aux fichiers de configuration de complexes. Pour ajouter ou modifier des complexes, vous devez utiliser les options `qconf(1) -Ac, -ac, -Mc` et `-mc`. Tandis que les options `-Ac` et `-Mc` prennent un fichier de configuration de *complexes* en tant qu'argument, les options `-ac` et `-mc` activent un éditeur qui s'ouvre avec une configuration de *complexe* modèle ou la configuration d'un complexe existant.

Les objets complexes de Sun Grid Engine intègrent quatre types de complexes :

- **Le complexe file d'attente** - Il est référencé par le nom spécial *queue*.

Dans sa forme par défaut, il contient une sélection de paramètres de la configuration de file telle que définie dans *queue_conf(5)*. Les paramètres de configuration de file pouvant être requis pour un travail par l'utilisateur sont en principe les suivants :

- *qname*
- *hostname*
- *notify*
- *calendar*
- *min_cpu_interval*
- *tmpdir*
- *seq_no*
- *s_rt*

- *h_rt*
- *s_cpu*
- *h_cpu*
- *s_data*
- *h_data*
- *s_stack*
- *h_stack*
- *s_core*
- *h_core*
- *s_rss*
- *h_rss*

Le complexe de file peut être étendu s'il est prévu que d'autres attributs soient disponibles pour chaque file. Le complexe de file définit les caractéristiques (telles que le type des données) des attributs qu'il contient. Un paramétrage de la valeur pour les attributs de complexe de file d'attente est défini par la configuration de file de chaque file dans le cas des paramètres standard listés ci-dessus, ou par l'entrée *complex_values* dans la configuration de file (voir `queue_conf(5)` pour plus de détails) pour tout paramètre ayant été ajouté au complexe de file par défaut. Si aucune définition de valeur n'est donnée dans ce dernier cas dans l'entrée *complex_values* de la configuration de file, la valeur est fixée à la valeur définie par le champ de valeur décrit ci-après.

- **Le complexe hôte** – Référencé par le nom spécial *host*, ce complexe contient la définition des caractéristiques de tous les attributs qu'il est prévu de gérer par hôte. L'ensemble standard d'attributs relatifs aux hôtes se compose de deux catégories, mais peut être amélioré similairement au complexe de file comme décrit ci-dessus. La première catégorie est constituée de plusieurs attributs de configuration de file qui sont particulièrement adaptés à une gestion par hôte. Ces attributs sont les suivants :
 - *slots*
 - *s_vmem*
 - *h_vmem*
 - *s_fsize*
 - *h_fsize*

(Reportez-vous à `queue_conf(5)` pour plus de détails).

Remarque – Définir ces attributs dans le complexe hôte n'entre pas en contradiction avec le fait qu'ils figurent également dans la configuration de file. Cela permet de conserver les ressources correspondantes au niveau des hôtes et, simultanément, au niveau des files. La mémoire libre virtuelle totale (*h_vmem*) peut être gérée pour un hôte, par exemple, tandis qu'un sous-ensemble de cette quantité peut être associé à une file de l'hôte concerné.

La seconde catégorie d'attributs d'un complexe hôte standard est celle des valeurs de charge par défaut. Chaque `sge_execd(8)` effectue périodiquement des rapports de charge à `sge_qmaster(8)`. Les valeurs de charge rapportées sont, selon le cas, les valeurs de charge Sun Grid Engine standard telles que la moyenne de charge des UC (voir `uptime(1)`) ou des valeurs de charge définies par l'administration de Sun Grid Engine (voir le paramètre `load_sensor` dans configuration de la grappe `sge_conf(5)`, le *Guide d'installation et d'administration de Sun Grid Engine* et le *Guide d'administration et d'utilisation de Sun Grid Engine 5.3* pour plus de détails). La définition des caractéristiques pour les valeurs de charge standard fait partie du complexe hôte par défaut, tandis que les valeurs définies par l'administrateur requièrent l'extension du complexe hôte. Reportez-vous au fichier, `<racine_sge>/doc/load_parameters.asc` pour des informations détaillées sur l'ensemble des valeurs de charge standard.

En général, le complexe hôte n'est pas seulement étendu pour inclure les paramètres de charge non-standard, il l'est aussi pour gérer les ressources liées à l'hôte telles que le nombre de licences de logiciels attribuées à un hôte ou l'espace disque disponible sur un système de fichiers local de l'hôte.

La valeur concrète d'un attribut de complexe hôte donné est déterminée par une configuration de file associée dans le cas d'attributs dérivés de la configuration de file, une valeur de charge rapportée ou la définition explicite d'une valeur dans l'entrée `complex_values` de la configuration d'hôte correspondante (voir `host_conf(5)`). Si aucun de ces éléments n'est disponible (par ex. si la valeur est supposée être un paramètre de charge, mais `sge_execd(8)` ne rapporte pas de valeur de charge pour elle), le champ de valeur décrit ci-après est utilisé.

- **Le complexe global** – Cet élément est référencé par le nom spécial *global*.

Les entrées configurées dans le complexe global font références aux attributs de ressource de la grappe, tels que le nombre de licences « flottantes » disponibles d'un logiciel donné ou l'espace disque libre sur un système de fichiers disponible dans tout le réseau. Les attributs de ressource globaux peuvent aussi être associés à des rapports de charge, si le rapport de charge correspondant contient l'identificateur `GLOBAL` (voir la section correspondante du *Guide d'installation et d'administration de Sun Grid Engine* ou du *Guide d'administration et d'utilisation de Sun Grid Engine 5.3* pour plus de détails). Les valeurs de charge globales peuvent être rapportées depuis tout hôte de la grappe. Par défaut, aucune valeur de charge globale n'est rapportée par Sun Grid Engine et, par conséquent, il n'y a pas de configuration de complexe globale par défaut.

Les valeurs concrètes des attributs de complexe global sont déterminées par les rapports de charge globaux ou par une définition explicite dans l'entrée `complex_values` de la configuration d'hôte « global » (voir `host_conf(5)`). Si aucun de ces deux éléments n'est disponible (par ex. si aucune valeur de charge n'a encore été rapportée), le champ de valeur décrit ci-après est utilisé.

- **Complexes définis par l'utilisateur** – En paramétrant les complexes définis par l'utilisateur, l'administrateur Sun Grid Engine a la possibilité d'étendre l'ensemble d'attributs géré par Sun Grid Engine tout en limitant l'influence de ces attributs à des files d'attente et/ou des hôtes particuliers. Un complexe d'utilisateur est tout

simplement une collection nommée d'attributs et la définition correspondante de comment ces attributs doivent être gérés par le logiciel Sun Grid Engine. Un ou plusieurs de ces complexes définis par l'utilisateur peuvent être rattachés à une file d'attente et/ou un hôte via les paramètres de configuration de file d'attente et d'hôte `complex_list` (voir `queue_conf(5)` et `host_conf(5)`). Les attributs définis dans tous les complexes attribués deviennent disponibles pour, dans l'ordre, la file d'attente et l'hôte, en sus des attributs de complexe par défaut.

Les valeurs concrètes des attributs des complexes définis par l'utilisateur doivent être fixées par le paramètre `complex_values` dans la configuration de file et d'hôte, sinon le champ `value` décrit ci-après est utilisé.

Format

Le format principal d'une configuration de *complexe* est une liste présentée sous la forme d'un tableau. Chaque ligne commençant par le caractère « # » est une ligne de commentaire. Chaque ligne, lignes de commentaires exceptées, définit un élément du complexe. Une ligne de définition d'élément se compose des six entrées de colonne (par ordre d'apparition) suivantes :

- *name* (nom) – Nom de l'élément de complexe à utiliser pour demander cet attribut pour un travail dans l'option `qsub(1) -l`. Un *name* d'attribut ne peut apparaître qu'une fois dans l'ensemble des complexes, ce qui revient à dire que la définition d'un attribut de complexe est unique.
- *shortcut* (raccourci) – Raccourci de *name* pouvant aussi être utilisé pour demander cet attribut pour un travail dans l'option `qsub(1) -l`. Un *shortcut* d'attribut ne peut apparaître qu'une fois dans l'ensemble des complexes, afin d'éviter tous risques de références d'attributs de complexe ambiguës.
- *type* (type) – Ce paramètre détermine comment les valeurs correspondantes doivent être traitées en interne par le logiciel Sun Grid Engine dans le cadre de comparaisons ou dans le cas de mise à l'échelle pour les entrées de complexe de charge :
 - Avec `INT` (`ENT`) seuls les entiers bruts sont autorisés.
 - Avec `DOUBLE` (`DOUBLE`) des nombres à virgule flottante en double précision (notation décimale et scientifique) peuvent être spécifiés.
 - Avec `TIME` (`HEURE`) des spécificateurs temporels sont autorisés. Reportez-vous à `queue_conf(5)` pour la description du format.
 - Avec `MEMORY` (`MÉMOIRE`) des spécificateurs de taille de mémoire sont autorisés. Reportez-vous à `queue_conf(5)` pour la description du format.
 - Avec `BOOL` (`BOOL`) les chaînes `TRUE` et `FALSE` sont autorisées. Si ce type est utilisé dans une formule de charge (voir `sched_conf(5)`) `TRUE` et `FALSE` sont mappés à « 1 » et « 0 ».
 - Avec `STRING` (`CHAÎNE`) toutes les chaînes sont autorisées et `strcmp(3)` est utilisé pour les comparaisons.

- CSTRING (CHAÎNEC) est identique à STRING à l'exception du fait que les comparaisons ne font pas la distinction Majuscules/minuscules.
- HOST (HÔTE) est identique à CSTRING mais la chaîne doit être un nom d'hôte valide.
- *value* (valeur) – Le champ *value* est une valeur prédéfinie pour un attribut, qui n'a d'effet que si elle n'est pas surécrite pendant une tentative visant à déterminer une valeur concrète pour l'attribut en relation avec une file, un hôte ou la grappe Sun Grid Engine. Le champ *value* peut être surécrit par ce qui suit.
 - Les valeurs de la configuration de file d'une file d'attente référencée.
 - Les valeurs de charge relatives à la grappe et spécifiques de l'hôte.
 - La spécification explicite d'une valeur via le paramètre *complex_values* dans la configuration de file ou d'hôte (voir `queue_conf(5)` et `host_conf(5)` pour plus de détails.

Si aucun des éléments ci-dessus ne s'applique, la valeur est définie pour l'attribut.

- *relop* (relation) – *relation operator* ou opérateur de relation. Cet élément est utilisé lorsque la valeur requise par l'utilisateur pour ce paramètre est comparée à la valeur correspondante configurée pour les files d'attente prises en compte. Si le résultat de la comparaison est *false*, le travail ne peut pas s'exécuter dans cette file. Les opérateurs de relation possibles sont `==`, `<`, `>`, `<=` et `>=`. Le seul opérateur admis pour les attributs de type chaîne est `==`.
- *requestable* (demandable) – L'entrée peut être utilisée dans une requête de ressources `qsub(1)` si ce champ est sur *y* ou *yes*. S'il est sur *n* ou *no*, cette entrée ne peut pas être utilisée par un utilisateur afin de demander une file d'attente ou une classe de files. Si l'entrée est fixée à *forced* ou *f*, l'attribut doit être demandé par un travail ou est rejeté.
- *consumable* (consommable) – Le paramètre *consumable* peut être fixé sur *yes* (*y* abrégé) ou *no* (*n*). Il ne peut être fixé sur *yes* que pour les attributs numériques (`INT`, `MEMORY`, `TIME`— voir *type* plus haut). S'il est sur *yes*, la consommation de la ressource correspondante peut être gérée par la comptabilité interne de Sun Grid Engine. Dans ce cas, Sun Grid Engine comptabilise la consommation de cette ressource pour tous les travaux en cours d'exécution et assure que les travaux ne sont acheminés que si la comptabilité interne de Sun Grid Engine indique qu'il y a suffisamment de ressources consommables disponibles. Les consommables constituent un moyen efficace de gérer des ressources limitées telles que la mémoire disponible, l'espace libre sur un système de fichiers, la bande passante réseau ou les licences de logiciels flottantes.

Les consommables peuvent être combinés avec les paramètres de charge par défaut ou définis par l'utilisateur (voir `sgc_conf(5)` et `host_conf(5)`); par ex., des valeurs de charge peuvent être rapportées pour les attributs consommables ou l'indicateur consommable peut être défini pour les attributs de charge. Dans ce cas, la gestion des ressources consommables de Sun Grid Engine prend en compte à la fois la charge (en mesurant la disponibilité de la ressource) et la comptabilité interne et assure qu'aucun de ces deux éléments ne dépasse une limite donnée.

Pour permettre la gestion des ressources consommables, la disponibilité de base d'une ressource doit être définie. Cela peut être fait globalement pour une grappe, par hôte ou par file d'attente, ces catégories pouvant se superposer les unes aux autres dans l'ordre indiqué (par ex., un hôte peut limiter la disponibilité d'une ressource d'une grappe et une file d'attente limiter les ressources de l'hôte et de la grappe). La définition de la disponibilité des ressources s'effectue avec l'entrée *complex_values* dans *host_conf(5)* et *queue_conf(5)*. La définition de *complex_values* de l'hôte « global » spécifie les paramètres globaux des consommables de la grappe. À chaque attribut de complexe consommable d'une liste *complex_values*, une valeur est attribuée qui indique la quantité maximale disponible de cette ressource. La comptabilité interne soustraira à ce total la consommation de ressources supposée de tous les travaux en cours d'exécution telles qu'exprimée au travers des requêtes de ressources de ces travaux.

Remarque – Les travaux peuvent être forcés à demander une ressource et, par conséquent, à spécifier leur consommation supposée via la valeur « force » (forcé) du paramètre demandable (voir ci-dessus).

Remarque – Une valeur de consommation de ressources par défaut peut être pré-définie par l'administrateur pour les attributs de consommables pas explicitement demandés par le travail (voir le paramètre *default* ci-après). Cela n'est intéressant que si la requête de l'attribut n'est pas forcée comme expliqué ci-dessus.

Voir le *Guide d'administration et d'utilisation de Sun Grid Engine 5.3* ou le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3* pour des exemples de l'utilisation de l'utilitaire des ressources consommables.

- *default* (valeur par défaut) – Ce paramètre n'est intéressant que pour les attributs de complexe consommables (voir le paramètre *consumable* plus haut). Sun Grid Engine assume de façon implicite que la quantité de la ressource indiquée va être consommée par les travaux acheminés vers un hôte ou une file gérant l'attribut consommable. Les travaux qui requièrent de façon explicite l'attribut via l'option `-l` de *qsub(1)* ignorent cette valeur par défaut.

Voir aussi

sge_intro(1), *qconf(1)*, *qsub(1)*, *uptime(1)*, *host_conf(5)*, *queue_conf(5)*, *sge_execd(8)*, *sge_qmaster(8)*, *sge_schedd(8)*, *Guide d'installation et d'administration de Sun Grid Engine*, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3*.

Droits d'auteur

Voir *sge_intro(1)* pour l'énoncé complet des droits et autorisations.

host_conf (5)

Nom

`host_conf` – Format des fichiers de configuration des hôtes d'exécution de Sun Grid Engine.

Description

`Host_conf` reflète le format du fichier modèle pour la configuration des hôtes d'exécution. Via les options `-ae` et `-me` de la commande `qconf(1)`, vous pouvez ajouter des hôtes d'exécution et modifier la configuration de tout hôte d'exécution de la grappe. Les entrées d'hôte d'exécution par défaut sont ajoutées automatiquement dès que `sge_execd(8)` s'enregistre auprès de `sge_qmaster(8)` pour la toute première fois depuis un certain hôte. Le modificateur `qconf(1) -sel` peut être utilisé pour afficher la liste des hôtes d'exécution couramment configurés dans votre système Sun Grid Engine. L'option `-se` vous permet d'imprimer la configuration d'hôte d'exécution d'un hôte spécifié.

Le nom d'hôte spécial, `global`, peut être utilisé pour définir les caractéristiques globales de la grappe.

Format

Le format d'un fichier `host_conf` est le suivant :

- *hostname* (nom d'hôte) – Nom de l'hôte d'exécution.
- *load_scaling* (mise à l'échelle de la charge) – Liste séparée par des virgules des valeurs de mise à l'échelle à appliquer à chacune ou à une partie des valeurs de charge faisant l'objet de rapports par `sge_execd(8)` sur l'hôte et définies dans le complexe « hôte » global de la grappe (voir `complex(5)`). Les facteurs de mise à l'échelle de la charge ont pour rôle de niveler les différences liées au matériel ou au système d'exploitation entre les hôtes d'exécution. Si, par exemple, la valeur de charge moyenne (`load_avg` dans le complexe « hôte » ; voir aussi `uptime(1)`) d'une machine multiprocesseur doit être comparée à celle d'une machine monoprocesseur, la charge telle que rapportée par l'hôte mono-UC doit être pondérée par rapport à la charge multiprocesseur (matériel UC identique) pour être comparable. Les facteurs de mise à l'échelle de charge sont des entiers qui sont multipliés par les quantités de charge rapportées pour constituer les valeurs

de charge pondérées. Par conséquent, en suivant l'exemple donné plus haut, la valeur de charge d'une machine monoprocesseur doit être multipliée par le nombre de processeurs de la machine multiprocesseur pour être comparable.

La syntaxe de la spécification d'un facteur de mise à l'échelle est la suivante : le nom de la valeur de charge (tel que défini dans le complexe « hôte ») est donné puis, séparée par un signe égal, la valeur de mise à l'échelle de la charge est donnée. Aucun blanc n'est autorisé dans la chaîne de valeur `load_scaling`.

Le paramètre `load_scaling` n'a pas de sens pour la définition de l'hôte « global ».

- **`complex_list`** (liste_complexes) – Liste séparée par des virgules des complexes définis par l'administrateur (voir `complex(5)` pour plus de détails) à associer à l'hôte. Seuls les attributs de complexe contenus dans les complexes listés et ceux des complexes « global » et « hôte », qui sont rattachés de manière implicite à chaque hôte, peuvent être utilisés dans la liste `complex_values` ci-dessous. Dans le cas de l'hôte « global », le complexe « hôte » n'est pas rattaché et seuls les attributs du complexe « global » sont autorisés par défaut dans la liste `complex_values` de l'hôte « global ».

La valeur par défaut de ce paramètre est NONE, c'est-à-dire qu'aucun complexe défini par l'administrateur n'est associé à l'hôte.

- **`complex_values`** (valeurs_complexes) – Les `complex_values` définissent des quotas pour les attributs de ressource gérés via cet hôte. Chaque attribut de complexe est suivi d'un signe « = » et de la spécification de valeur conforme au type d'attribut de complexe (voir `complex(5)`). Les spécifications de quotas sont séparées par des virgules. Seuls les attributs tels que définis dans `complex_list` (voir ci-dessus) peuvent être utilisés.

Les quotas sont liés à la consommation de ressources de tous les travaux sur un hôte dans le cas des ressources consommables (voir `complex(5)` pour plus de détails sur les ressources consommables) ou sont interprétés sur la base des emplacements de travail dans le cas de ressources non-consommables. Les attributs de ressources consommables sont fréquemment utilisés pour gérer la mémoire libre, l'espace disque libre ou les licences de logiciels flottantes tandis que les attributs non-consommables définissent en général des caractéristiques distinctives telles que le type de matériel installé.

Pour les attributs de ressources consommables, la quantité de ressources disponibles se détermine en soustrayant la consommation courante de ressources de tous les travaux en cours d'exécution sur l'hôte au quota de la liste `complex_values`. Les travaux peuvent uniquement être acheminés vers un hôte si aucune requête de ressource ne dépasse la disponibilité correspondante de la ressource obtenue par ce système. La définition du quota dans la liste `complex_values` est automatiquement remplacée par la valeur de charge courante rapportée pour cet attribut, si la charge est surveillée pour cette ressource et si la valeur de charge rapportée est plus contraignante que le quota. Ce mécanisme évite de manière efficace la sur-souscription des ressources.

Remarque – Les valeurs de charge qui remplacent les spécifications de quotas peuvent être devenues plus contraignantes suite à leur mise à l'échelle (voir `load_scaling` plus haut) et/ou à des ajustements de charge (voir `sched_conf(5)`). L'option `-F` de `qstat(1)` et l'affichage de la charge dans la boîte de dialogue de contrôle des files de `qmon(1)` (qui s'ouvre en cliquant sur l'icône d'une file tout en appuyant sur la touche « Maj ») fournit des informations détaillées sur la disponibilité courante des ressources consommables et sur l'origine des valeurs couramment prises en compte.

Remarque – La consommation de ressources des travaux en cours d'exécution (utilisée pour le calcul de la disponibilité) ainsi que les requêtes des travaux en attente d'être acheminés peuvent être dérivées des requêtes explicites des utilisateurs pendant la soumission des travaux (voir l'option `-l` de `qsub(1)`) ou d'une valeur « par défaut » configurée pour un attribut par l'administrateur (voir `complex(5)`). L'option `-r` de `qstat(1)` peut être utilisée pour récupérer des détails complets sur les requêtes de ressources courantes de tous les travaux au sein du système.

Pour les ressources non-consommables, Sun Grid Engine compare tout simplement les requêtes des attributs du travail à la spécification correspondante dans `complex_values` en tenant compte de l'opérateur de relation de la définition de l'attribut de complexe (voir `complex(5)`). Si le résultat de la comparaison est « true », l'hôte est adapté au travail en ce qui concerne l'attribut considéré. Pour les travaux parallèles, chaque emplacement de travail devant être occupé par une tâche parallèle est sensé fournir la même valeur d'attribut de ressource.

Remarque – Seuls les attributs de complexe numériques peuvent être définis en tant que ressources consommables et, par conséquent, les attributs non-numériques sont toujours gérés par emplacement travail.

La valeur par défaut de ce paramètre est NONE, aucun quota d'attribut de ressource défini par l'administrateur n'est associé à l'hôte.

- *load_values* (valeurs_charge) – Cette entrée ne peut pas être configurée, elle est uniquement affichée dans le cas d'une commande `qconf(1) -se`. Toutes les valeurs de charge s'affichent telles que rapportées par `sgc_execd(8)` sur l'hôte. Les valeurs de charge sont listées dans une liste séparée par des virgules. Chaque valeur de charge commence par son nom, suivi du signe égal et de la valeur rapportée.
- *processors* (processeurs) – Cette entrée ne peut pas être configurée, elle est uniquement affichée dans le cas d'une commande `qconf(1) -se`. Sa valeur est le nombre de processeurs qui a été détecté par `sgc_execd(8)` sur l'hôte correspondant.

- *usage_scaling* (mise à l'échelle de l'utilisation) – Cette entrée n'est présente que dans un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible dans Sun Grid Engine.

Son format est similaire à celui de *load_scaling* (voir plus haut), les seuls attributs valides à mettre à l'échelle toutefois sont `cpu` pour la consommation de temps UC, `mem` pour la consommation de mémoire cumulée pendant la durée de vie des travaux et `io` pour les données transférées via tous les périphériques E/S. La valeur par défaut NONE signifie « pas de mise à l'échelle », soit des facteurs de mise à l'échelle de 1.

- *resource_capability_factor* (facteur de capacité de ressources) – Cette entrée n'est présente que dans un système Sun Grid Engine, Enterprise Edition. Elle n'est pas disponible dans Sun Grid Engine.

Il s'agit du facteur de capacité de ressources qui est utilisé par Sun Grid Engine, Enterprise Edition lors de l'attribution des travaux aux hôtes d'exécution. Ce facteur indique à Sun Grid Engine, Enterprise Edition les ressources (UC, mémoire, E/S, etc.) d'un hôte d'exécution par rapport à celles d'autres hôtes d'exécution. Cela permet d'assurer qu'un travail requérant un pourcentage important de ressources (par ex. beaucoup de tickets) soit placé sur un hôte d'exécution contenant un pourcentage important de ressources disponibles. La situation de charge sur les hôtes d'exécution est en sus prise en compte, pour garantir que l'hôte sélectionné soit à la fois suffisamment puissant et peu chargé.

Par exemple, vous pouvez envisager de définir vos facteurs de capacité de ressources pour chaque hôte d'exécution sur la base du nombre d'UC, de la vitesse des UC et de la mémoire principale installée :

$\text{nbre_UC} * (\text{MHz}/200) + \text{Go_de_mémoire}$

Cela donnerait à un hôte d'exécution ayant 32 UC à 200 MHz et 10 Go de mémoire un facteur de capacité de ressources de 42, tandis qu'un hôte d'exécution ayant 24 UC à 200 MHz et 40 Go de mémoire obtiendrait un facteur de capacité de ressources de 64, ce qui indique que la mémoire a un impact considérable dans cet exemple.

Vous pouvez aussi envisager de prendre en compte les facteurs suivant lors du paramétrage du facteur de capacité de ressources :

- mélange de travaux – travaux liés à l'UC ou à la mémoire.
- comparatifs UC – comparaison par constructeur d'UC
- Mégaflops de mémoire (MFLOPS) – pour les calculs
- capacités E/S – vitesse disque/réseau
- espace disque disponible – sur l'hôte d'exécution

Le facteur de capacité de ressources est stocké sous la forme d'une valeur à virgule flottante double. La plage de valeurs utilisée n'a pas d'importance. Sun Grid Engine, Enterprise Edition ne regarde que la relation entre les valeurs des différents hôtes.

Voir aussi

`sge_intro(1)`, `qconf(1)`, `uptime(1)`, `complex(5)`, `sge_execd(8)`,
`sge_qmater(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

hostgroup(5)

Nom

`hostgroup` - Format des fichiers d'entrée de groupe d'hôtes.

Description

Une entrée `hostgroup` est utilisée pour fusionner des noms d'hôtes en un groupe. Chaque fichier d'entrée `hostgroup` définit un groupe. Le fait que le signe @ soit le premier caractère d'un nom indique qu'il s'agit d'un groupe. À ce stade de la mise en oeuvre, vous pouvez utiliser `hostgroups` dans la configuration `usermapping(5)`. Dans un fichier de définition de groupe, vous pouvez aussi référencer d'autres groupes. Ces groupes sont appelés *sous-groupes*.

Chaque ligne d'un fichier d'entrée `hostgroup` spécifie un nom d'hôte ou un groupe appartenant à ce groupe.

Vous pouvez afficher la liste des entrées `hostgroup` couramment configurées via l'option `qconf(1) -shgrp1`. Vous pouvez afficher le contenu de chaque entrée `hostgroup` listée via le modificateur `-shgrp`. La sortie suit la description du format de `hostgroup`. Vous pouvez créer et modifier des entrées `hostgroup` via les options `-ahgrp`, `-mhgrp` et `-dhgrp` de `qconf(1)`.

Format

Une entrée `hostgroup` contient au moins deux paramètres.

- le mot clé *group_name* (nom du groupe) – Le mot clé *group_name* définit le nom du `hostgroup`. Le texte qui suit le mot clé *group_name* sera considéré comme une valeur de nom `hostgroup`.
- *hostname* (nom d'hôte) – Nom de l'hôte qui est maintenant membre du groupe spécifié par *group_name*. Si le premier caractère de *hostname* est le signe @, ce nom est utilisé pour référencer un `hostgroup(5)`, qui est considéré comme un sous-groupe de ce groupe.

Exemple

L'exemple qui suit est une entrée `hostgroup` typique.

```
group_name grossesMachines
@calcul
hôterapide
```

Cette entrée définira un nouvel `hostgroup` appelé `grossesMachines`. De ce `hostgroup` font partie l'hôte `hôterapide` et tous les membres de l'`hostgroup`, `calcul`.

Voir aussi

`qconf(1)`, `usermapping(5)`

Droits d'auteur

Voir `sgc_intro(1)` pour l'énoncé complet des droits et autorisations.

project(5)

Nom

`project` – Format des fichiers d'entrée de projet de Sun Grid Engine, Enterprise Edition.

Description

L'objet `project` n'est disponible que dans le cas d'un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine n'a pas d'objet `project`.

Dans Sun Grid Engine, Enterprise Edition, les travaux peuvent être soumis à des projets et un projet peut se voir attribuer un certain niveau d'importance via la politique fonctionnelle ou de remplacement. Ce niveau d'importance est ensuite hérité par les travaux qui s'exécutent sous ce projet.

La liste des projets couramment configurés peut être affichée via l'option `qconf(1) -sprjl`. Le contenu de chaque définition de projet listée peut être affiché via le modificateur `-sprj`. La sortie adopte la description du format `project`. De nouveaux projets peuvent être créés et ceux existants peuvent être modifiés via les options `-aprj`, `-mprj` et `-dprj` de `qconf(1)`.

Format

Une définition de projet contient les paramètres suivants :

- `name` – Nom du projet.
- `oticket` – Quantité de tickets de remplacement couramment attribués au projet.
- `fshare` – Partage fonctionnel courant du projet.
- `facl` – Liste de contrôle d'accès (ACL - voir `access_list(5)`) précisant les utilisateurs autorisés à soumettre des travaux au projet.
- `fxacl` – Liste de contrôle d'accès (ACL - voir `access_list(5)`) précisant les utilisateurs qui ne sont pas autorisés à soumettre des travaux au projet.

Voir aussi

`sgc_intro(1)`, `qconf(1)`, `access_list(5)`

Droits d'auteur

Voir `sgc_intro(1)` pour l'énoncé complet des droits et autorisations.

`qtask(5)`

Nom

`qtask` - Format du fichier `qtask`.

Description

Un fichier `qtask` définit les commandes qui sont soumises à Sun Grid Engine pour une exécution à distance par `qtcsh(1)`. Le fichier `qtask` peut en option contenir des paramètres de ligne de commande `qrsh(1)`. Ces paramètres sont transmis à la commande `qrsh(1)` utilisée par `qtcsh` pour soumettre les commandes.

Un fichier `qtask` global de grappe définissant les valeurs par défaut à l'échelle de la grappe et un fichier `qtask` spécifique de l'utilisateur qui éventuellement remplacerait ou améliorerait ces définitions sont pris en charge. Le fichier global de grappe réside en `<racine_sge>/<cellule>/common/qtask`, tandis que le fichier spécifique de l'utilisateur se trouve dans `~/ .qtask`. L'administrateur peut mettre un point d'exclamation devant les définitions de commande dans le fichier global de la grappe pour refuser le remplacement de ces commandes par les utilisateurs.

Format

Le format du fichier `qtask` est celui d'une liste présentée sous la forme d'un tableau. Toute ligne commençant par le caractère « # » est une ligne de commentaire. Chaque ligne, lignes de commentaires exceptées, définit une commande à démarrer à distance.

La définition commence par le nom de la commande qui doit correspondre exactement au nom tapé dans une ligne de commande `qtcsh(1)`. Les chemins ne sont pas autorisés dans les fichiers `qtask`. Par conséquent, les chemins absolus ou relatifs indiqués dans les lignes de commande `qtcsh(1)` portent toujours à l'exécution locale si les commandes elles-mêmes sont identiques à celles définies dans les fichiers `qtask`.

Le nom de la commande peut être suivi d'un nombre arbitraire d'arguments d'option de `qrsh(1)` qui sont transmis à `qrsh(1)` par `qtcsh(1)`. La présence d'un point d'exclamation devant la commande dans le fichier `qtask` global de grappe en interdit le remplacement par le fichier `qtask` fourni par l'utilisateur.

Exemple

```
netscape -l a=solaris64 -v DISPLAY=myhost:0
grep -l h=filesurfer
verilog -l veri_lic=1
```

Le fichier `qtask` ci-dessus désigne les applications `netscape`, `grep` et `verilog` pour l'exécution à distance interactive via le logiciel Sun Grid Engine. Il est demandé à Netscape de ne s'exécuter que sur les architectures `Solaris64` dont la variable d'environnement `DISPLAY` est sur `myhost:0`, à `grep` de ne s'exécuter que sur le nom d'hôte `filesurfer` et à `verilog` de demander la disponibilité d'une licence `verilog` pour pouvoir être exécuté à distance.

Voir aussi

`sge_intro(1)`, `qtcsh(1)`, `qrsh(1)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

queue_conf (5)

Nom

`queue_conf` – Format des fichiers de configuration de file d'attente de Sun Grid Engine.

Description

`queue_conf` reflète le format du fichier modèle pour la configuration d'une file d'attente. Via les options `-aq` et `-mq` de la commande `qconf(1)`, vous pouvez ajouter des files d'attente et modifier la configuration de toute file d'attente dans la grappe.

Les paramètres `queue_conf` acceptent comme valeurs des chaînes, des nombres entiers décimaux ou booléens, des spécificateurs temporels et de mémoire ainsi que des listes séparées par des virgules. Un spécificateur temporel consiste en une constante décimale positive, hexadécimale ou entière octale, cas dans lequel la valeur est interprétée en secondes, ou est constitué par trois nombres entiers décimaux séparés par le signe deux-points, le premier nombre indiquant les heures, le deuxième les minutes et le troisième les secondes. Si un nombre est égal à zéro il est possible de le sauter à condition toutefois de laisser le signe deux-points de séparation (par ex. `1:0:1 = 1::1` signifie 1 heure et 1 seconde).

Les spécificateurs de mémoire sont des constantes décimales positives, hexadécimales ou entières octales, qui peuvent être suivies par une lettre multiplicatrice. Les lettres multiplicatrices admises sont `k`, `K`, `m` et `M`, `k` indiquant de multiplier la valeur par 1000, `K` par 1024, `m` par 1000*1000 et `M` par 1024*1024. Si aucun multiplicateur n'est présent, la valeur est simplement comptabilisée en octets.

Format

La liste suivante de paramètres de `queue_conf` spécifie le contenu de `queue_conf` :

- *qname* – Nom de la file d'attente sur le noeud (type chaîne ; modèle par défaut : `template`).
- *hostname* – Nom d'hôte complet du noeud (type chaîne ; modèle par défaut : `hôte.dom.dom.dom`).

- *seq_no* – En conjonction avec la situation de charge de l'hôte, à un moment donné ce paramètre spécifie la position de cette file d'attente dans l'ordre de programmation au sein des files d'attente adaptées pour qu'un travail soit acheminé en prenant en compte la *queue_sort_method* (voir *sched_conf(5)*).

Quel que soit le paramétrage de *queue_sort_method*, *qstat(1)* rapporte les informations sur les files dans l'ordre défini par la valeur de *seq_no*. Mettez ce paramètre sur une séquence croissante monotone. Le type est numérique et la valeur par défaut est 0.

- *load_thresholds* (seuils de charge) – Liste des seuils de charge. Si l'un de ces seuils est dépassé aucun travail supplémentaire ne sera programmé sur les files d'attente du noeud concerné et *qmon(1)* signalera une condition de surcharge pour ce noeud. Les valeurs de charge arbitraires définies dans les complexes « hôte » et « global » (voir *complex(5)* pour plus de détails) peuvent être utilisées.

Il s'agit d'une liste séparée par des virgules dont chaque élément est constitué par le nom d'une valeur de charge suivi du signe égal et de la valeur seuil sensée déclencher la situation de surcharge (par ex. : *load.avg=175,users_logged_in=5*).

Remarque – Les valeurs de charge ainsi que les ressources consommables peuvent être mises à l'échelle de façon différente pour différents hôtes si spécifié dans les définitions d'hôte d'exécution correspondantes (consultez *host_conf(5)* pour plus d'informations). Les seuils de charge sont comparés aux valeurs de charge mises à l'échelle et consommables.

- *suspend_thresholds* (seuils d'interruption) – Liste de seuils de charge employant la même sémantique que celle du paramètre *load_thresholds* (voir ci-dessus), la différence étant que le dépassement de l'un des seuils indiqués lance l'interruption de l'un des travaux de la file. Voir le paramètre *nsuspend* ci-après pour plus de détails sur le nombre de travaux interrompus.
- *nsuspend* – Nombre de travaux interrompus/activés par intervalle de temps si au moins un des seuils de charge figurant dans la liste *suspend_thresholds* est dépassé ou si aucun *suspend_threshold* n'est plus violé. Les travaux *nsuspend* sont interrompus pendant l'intervalle de temps jusqu'à ce qu'aucun *suspend_thresholds* ne soit plus dépassé ou que tous les travaux de la file soient interrompus. Inversement, les travaux sont activés en suivant la même logique si les *suspend_thresholds* ne sont plus dépassés. L'intervalle de temps durant lequel les interruptions de travaux surviennent est défini dans *suspend_interval* ci-après.
- *suspend_interval* (intervalle d'interruption) – Intervalle de temps durant lequel d'autres travaux *nsuspend* sont interrompus si un des *suspend_thresholds* (voir ci-dessus pour les deux) est dépassé par la charge courante sur l'hôte sur lequel se trouve la file d'attente. Cet intervalle de temps est également utilisé lors de l'activation des travaux.

- *priority* (priorité) – Le paramètre *priority* spécifie la valeur *nice*(2) à laquelle les travaux de cette file d'attente seront exécutés. Le type est *numérique* et la valeur par défaut est zéro (ce qui signifie qu'aucune valeur *nice* n'est définie de façon explicite). Les valeurs négatives (jusqu'à -20) correspondent à une priorité de programmation supérieure. Les valeurs positives (jusqu'à +20) correspondent à une valeur de priorité inférieure. Dans un système Sun Grid Engine, Enterprise Edition, la valeur de *priority* est sans effet car le système Sun Grid Engine, Enterprise Edition ajuste les priorités de façon dynamique pour mettre en oeuvre les objectifs des politiques de droits de Sun Grid Engine, Enterprise Edition. L'ajustement dynamique des priorités peut toutefois être désactivé en mettant *execd_param NO_REPRIORITIZATION* sur *true* dans la configuration globale ou la configuration de grappe locale du démon d'exécution (voir *sge_conf*(5)). Dans ce cas, le paramétrage de *priority* aura aussi un effet dans le système Sun Grid Engine, Enterprise Edition.
- *min_cpu_interval* (intervalle UC min.) – Temps entre deux points de reprise automatiques en cas de travaux à écriture de points de reprise transparente. Le maximum du temps demandé par l'utilisateur via *qsub*(1) et du temps défini par la configuration de file est utilisé en tant qu'intervalle entre deux points de reprise. Les fichiers de point de reprise pouvant avoir une taille considérable et leur écriture dans le système pouvant par conséquent se révéler coûteuse, il est conseillé aux utilisateurs et aux administrateurs de choisir des intervalles de temps suffisamment longs. *min_cpu_interval* est de type temporel et sa valeur par défaut est 5 minutes (cette valeur, en général, ne convient que pour les tests).
- *processors* (processeurs) – Ensemble des processeurs dans le cas d'un hôte d'exécution multiprocesseur auquel les travaux s'exécutant dans la file sont liés. La valeur type de ce paramètre est une plage telle que celle de l'option *-pe* de *qsub*(1) (par ex. 1-4,8,10) qui indique les numéros des processeurs pour le groupe de processeurs à utiliser. Il est évident que l'interprétation de ces valeurs dépend du système d'exploitation et elle est par conséquent effectuée au sein du *sge_execd*(8) s'exécutant sur l'hôte de la file. Par conséquent, l'analyse du paramètre doit être fournie par le démon d'exécution et le paramètre n'est transmis au travers de *sge_qmaster*(8) que sous forme de chaîne.
 À l'heure actuelle, la prise en charge n'est fournie que pour les machines multiprocesseur SGI exécutant IRIX 6.2 et les machines multiprocesseur UNIX de Digital. Dans le cas de ces dernières, seul un travail par ensemble de processeurs est autorisé à s'exécuter simultanément ; c'est-à-dire que *slots* (voir ci-dessus) doit être mis sur 1 pour cette file d'attente.
- *qtype* – Type de la file d'attente. Il s'agit actuellement de *batch*, *interactive*, *parallel* ou *checkpointing* ou de toute combinaison de ces éléments dans une liste séparée par des virgules. Le type est *chaîne* ; la valeur par défaut *batch interactive parallel*.
- *rerun* (réexécuter les travaux) – Définit un comportement par défaut pour les travaux qui sont abandonnés à la suite d'un crash du système ou de l'arrêt manuel « violent » (via *kill*(1)) de l'ensemble du système Sun Grid Engine (y compris du *sge_shepherd*(8) des travaux et leur hiérarchie de processus) sur l'hôte de la file.

Dès que `sge_execd(8)` est redémarré et détecte un job ayant été abandonné pour ces raisons, il peut le redémarrer s'il s'agit d'un travail redémarrable. Un travail peut ne pas être redémarrable si, par exemple, il met à jour les bases de données (lit puis écrit dans un enregistrement d'une base de données/fichier) car l'abandon du travail peut avoir laissé la base de données dans un état inconsistant. Si le propriétaire d'un travail veut annuler le comportement par défaut pour les travaux dans la file, l'option `-r` de `qsub(1)` peut être utilisée à cet effet.

Ce paramètre étant de type booléen, TRUE ou FALSE peut être spécifié. La valeur par défaut est FALSE, qui correspond à ne pas redémarrer automatiquement les travaux.

- *slots* (emplacements) – Il s'agit du nombre maximal de travaux autorisés à s'exécuter simultanément dans la file. Le type est numérique.
- *tmpdir* (répertoire tmp) – Le paramètre *tmpdir* spécifie le chemin absolu de la base du système de fichiers du répertoire temporaire. Lorsque `sge_execd(8)` lance un travail, il crée un répertoire de nom unique dans ce système de fichiers dans l'objectif de conserver les fichiers temporaires de travail pendant l'exécution du travail. Une fois le travail terminé, ce répertoire et son contenu sont automatiquement supprimés. Les variables d'environnement `TMPDIR` et `TMP` sont réglées sur le chemin de chaque répertoire de travail temporaire (type chaîne ; valeur par défaut : `/tmp`).
- *shell* – Si *posix_compliant* ou *script_from_stdin* est spécifié par le paramètre *shell_start_mode* dans `sge_conf(5)`, le paramètre *shell* spécifie le chemin de l'exécutable de l'interpréteur de commandes (par ex. `sh(1)` ou `csh(1)`) à utiliser pour traiter les scripts de travaux exécutés dans la file. La définition de *shell* peut être annulée par le propriétaire du travail via l'option `qsub(1) -S`.

Le type du paramètre est chaîne. La valeur par défaut est `/bin/csh`.

- *shell_start_mode* (mode de démarrage de shell) – Ce paramètre définit les mécanismes qui sont utilisés pour appeler les scripts des travaux sur les hôtes d'exécution. Les valeurs suivantes sont reconnues :
 - *unix_behavior* (comportement UNIX) – Si un utilisateur démarre un script shell de travail de manière interactive sous UNIX en l'appelant juste après le nom du script le chargeur d'exécutables du système d'exploitation utilise les informations fournies dans un commentaire tel que « `#!/bin/csh` » dans la première ligne du script pour détecter quel interpréteur de commandes démarrer pour interpréter le script. Ce mécanisme est utilisé par Sun Grid Engine au démarrage des travaux si *unix_behavior* est défini en tant que *shell_start_mode*.
 - *posix_compliant* (conforme POSIX) – POSIX ne prend pas en considération les commentaires de la première ligne tels que « `#!/bin/csh` ». Le standard POSIX pour les systèmes de files d'attente par lots (P1003.2d) exige donc qu'un système de mise en file d'attente conforme ignore ces lignes et utilise à la place des interpréteurs de commandes par défaut configurés ou spécifiés par l'utilisateur. Par conséquent, si *shell_start_mode* est sur *posix_compliant* Sun Grid Engine utilisera l'interpréteur de commandes indiqué par l'option `-S` de la commande `qsub(1)` ou le paramètre *shell* de la file d'attente à utiliser (voir ci-dessus).

- *script_from_stdin* (script de stdin) – Régler le paramètre *shell_start_mode* sur *posix_compliant* ou *unix_behavior* exige que vous définissiez le umask utilisé pour *sgexecd(8)* de sorte que chaque utilisateur ait accès en lecture au répertoire des travaux actifs dans le répertoire de spool du démon d'exécution correspondant. Dans le cas où vous auriez des scripts *prolog* et *epilog* configurés, ils devront aussi pouvoir être lus par tout utilisateur susceptible d'exécuter des travaux.

Si cela viole les politiques de sécurité de votre site, vous pouvez décider de mettre *shell_start_mode* sur *script_from_stdin*. Cela forcera Sun Grid Engine à ouvrir le script du travail ainsi que les scripts *epilog* et *prolog* pour lire dans STDIN à titre de super-utilisateur (si *sgexecd(8)* a été démarré en tant que super-utilisateur) avant de passer au compte d'utilisateur du propriétaire du travail. Le script est ensuite introduit dans le flux STDIN de l'interpréteur de commandes indiqué par l'option *-S* de la commande *qsub(1)* ou le paramètre *shell* de la file d'attente à utiliser (voir ci-dessus).

Par conséquent, mettre *shell_start_mode* sur *script_from_stdin* implique aussi un comportement *posix_compliant*.

Remarque – Introduire des scripts dans le flux STDIN d'un interpréteur de commandes peut poser des problèmes si des commandes telles que *rsh(1)* sont appelées dans un script de travail car elles traitent également le flux STDIN de l'interpréteur de commandes. Ces problèmes peuvent en général être éliminés en redirigeant le canal STDIN de ces commandes pour qu'il provienne de */dev/null* (par ex. *rsh host date < /dev/null*).

Remarque – Toutes les options de ligne de commande associées au travail sont transférées au shell d'exécution. Ce shell ne les transmettra au travail que si elles ne sont pas reconnues comme des options de shell valides.

La valeur par défaut de *shell_start_mode* est *posix_compliant*.

- *prolog* (prologue) – Chemin de l'exécutable d'un script shell qui est démarré avant l'exécution des travaux Sun Grid Engine avec les mêmes paramètres d'environnement que pour les travaux Sun Grid Engine qui seront commencés après. Un préfixe « utilisateur@ » optionnel spécifie l'utilisateur sous lequel cette procédure va être démarrée. Cette procédure est conçue comme un moyen à la disposition de l'administrateur Sun Grid Engine d'automatiser l'exécution de tâches générales spécifiques d'un site telles que la préparation de systèmes de fichiers temporaires ayant besoin des mêmes informations de contexte que le travail. Cette entrée de configuration surécrit les définitions globale de grappe et spécifique de l'hôte d'exécution (voir *sgconf(5)*).

Remarque – *prolog* est exécuté *exactement* comme le script du travail. Par conséquent, toutes les implications décrites sous les paramètres *shell_start_mode* et *login_shells* ci-après s'appliquent.

La valeur par défaut de *prolog* est la valeur spéciale `NONE`, qui empêche l'exécution d'un script *prolog*. Les variables spéciales utilisées pour constituer une ligne de commande sont les mêmes que dans les définitions *prolog* de la configuration de grappe (voir `sge_conf(5)`).

- *epilog* (épilogue) – Chemin de l'exécutable d'un script shell qui est démarré après l'exécution des travaux Sun Grid Engine avec les mêmes paramètres d'environnement que pour les travaux Sun Grid Engine qui viennent d'être terminés. Un préfixe `utilisateur@` optionnel spécifie l'utilisateur sous lequel cette procédure va être démarrée. Cette procédure a été conçue pour permettre aux administrateurs Sun Grid Engine d'automatiser l'exécution de tâches générales spécifiques d'un site telles que le nettoyage des systèmes de fichiers temporaires ayant besoin des mêmes informations de contexte que le travail. Cette entrée de configuration surécrit les définitions globale de grappe et spécifique de l'hôte d'exécution (voir `sge_conf(5)`).

Remarque – *epilog* est exécuté *exactement* comme le script du travail. Par conséquent, toutes les implications décrites sous les paramètres *shell_start_mode* et *login_shells* ci-après s'appliquent.

La valeur par défaut de *epilog* est la valeur spéciale `NONE`, qui empêche un script *epilog* de s'exécuter. Les variables spéciales pour constituer une ligne de commande sont les mêmes que dans les définitions de *prolog* de la configuration de la grappe (voir `sge_conf(5)`).

- *starter_method* (méthode de démarrage) – Le chemin d'exécutable spécifié sera utilisé en tant qu'utilitaire démarreur de travaux pour démarrer des travaux par lots. L'exécutable désigné sera exécuté à la place du shell configuré pour démarrer le programme. Les arguments du travail seront passés sous la forme d'arguments au démarreur de travaux. Les variables d'environnement suivantes sont utilisées pour passer les informations au démarreur des travaux concernant l'environnement de shell qui a été configuré ou demandé pour démarrer le travail.
 - *SGE_STARTER_SHELL_PATH* – Nom du shell demandé pour démarrer le travail.
 - *SGE_STARTER_SHELL_START_MODE* – *mode_démarrage_shell* configuré.
 - *SGE_STARTER_USE_LOGIN_SHELL* – Sur `true` si le shell est supposé être utilisé en tant que shell de connexion (voir *login_shells* dans `sge_conf(5)`).

La *starter_method* ne sera pas appelée pour `qsh`, `qlogin`, ou `qcrsh` se comportant comme `rlogin`.

- *suspend_method*, *resume_method*, *terminate_method* (méthode d'interruption, méthode de reprise, méthode de conclusion) – Ces paramètres peuvent être utilisés pour surécrire la méthode par défaut utilisée par Sun Grid Engine pour l'interruption, la reprise et la conclusion d'un travail. Par défaut, les signaux `SIGSTOP`, `SIGCONT` et `SIGKILL` sont fournis au travail pour effectuer ces actions. Cela n'est toutefois pas approprié pour certaines applications.

Si aucun chemin d'exécutable n'est donné, Sun Grid Engine prend les entrées de paramètre spécifiées comme le signal à fournir au lieu du signal par défaut. Un signal doit être un nombre positif ou un nom de signal doté du préfixe SIG et le nom du signal tel qu'imprimé par `kill -l` (par ex. SIGTERM).

Si un chemin d'exécutable est indiqué (il doit s'agir d'un *chemin absolu* commençant par le signe « / »), cette commande est démarrée avec ses arguments par Sun Grid Engine pour effectuer l'action appropriée. Les variables spéciales suivantes qui sont étendues au moment de l'exécution peuvent être utilisées (en plus de toutes autres chaînes devant être interprétées par la procédure) pour constituer une ligne de commande :

- *\$host* – Nom de l'hôte sur lequel la procédure est démarrée.
- *\$job_owner* – Nom d'utilisateur du propriétaire du travail.
- *\$job_id* – Numéro d'identification unique du travail du système Sun Grid Engine.
- *\$job_name* – Nom du travail.
- *\$queue* – Nom de la file d'attente.
- *\$job_pid* – PID du travail.
- *notify* (avertir) – Délai respecté entre la fourniture des signaux d'avertissement SIGUSR1/SIGUSR2 et des signaux suspend/kill si le travail a été soumis avec l'option `qsub(1) -notify`.
- *owner_list* (liste des propriétaires) – La *owner_list* liste les noms de connexion (sous la forme d'une liste séparée par des virgules) des utilisateurs qui sont autorisés à interrompre cette file d'attente (les administrateurs et les opérateurs Sun Grid Engine peuvent interrompre des files par défaut). Il est obligatoire de définir ce champ pour les files d'attente sur des stations de travail interactives où les ressources de calcul sont partagées entre les sessions interactives et les travaux Sun Grid Engine, permettant au propriétaire de la station de travail d'avoir un accès prioritaire (type *chaîne* ; valeur par défaut : NONE).
- *user_lists* (listes d'utilisateurs) – Le paramètre *user_lists* contient une liste séparée par des virgules d'ACL (voir description dans `access_list(5)`). Tout utilisateur figurant dans au moins une des ACL listées a accès à la file d'attente. Si le paramètre *user_lists* est sur NONE (la valeur par défaut), tout utilisateur qui n'est pas explicitement exclu via le paramètre *xuser_lists* a accès. Si un utilisateur figure à la fois dans une ACL listée dans *xuser_lists* et dans une ACL listée dans *user_lists*, il se voit refuser l'accès à la file d'attente.
- *xuser_lists* (listes d'utilisateurs X) – Le paramètre *xuser_lists* contient une liste séparée par des virgules d'ACL (voir description dans `access_list(5)`). Tout utilisateur figurant dans au moins une des ACL listées n'est pas autorisé à accéder à la file d'attente. Si le paramètre *xuser_lists* est sur NONE (la valeur par défaut) tout utilisateur a accès. Si un utilisateur figure à la fois dans une ACL listée dans *xuser_lists* et une ACL listée dans *user_lists*, il se voit refuser l'accès à la file d'attente.
- *projects* (projets) – Le paramètre *projects* contient une liste séparée par des virgules des projets qui ont accès à la file d'attente. Tout projet ne figurant pas dans cette liste se voit refuser l'accès à la file d'attente. S'il est sur NONE (la valeur par

défaut), tout projet qui n'est pas spécifiquement exclu via le paramètre *xprojects* décrit ci-après a accès. Si un projet figure dans les deux paramètres *projects* et *xprojects*, il se voit refuser l'accès à la file d'attente. Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition.

- *xprojects* (projets X) – Le paramètre *xprojects* contient une liste séparée par des virgules des projets auxquels l'accès à la file d'attente est refusé. S'il est sur *NONE* (la valeur par défaut), l'accès n'est refusé à aucun projet à l'exception de ceux auxquels il est refusé d'après le paramètre *projects* ci-dessus. Si un projet figure dans les deux paramètres *projects* et *xprojects*, il se voit refuser l'accès à la file d'attente. Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition.
- *subordinate_list* (subordonnés) – Liste des files de Sun Grid Engine, résidant sur le même hôte que la file configurée, à interrompre lorsqu'un nombre spécifié de travaux est en cours d'exécution dans cette file. La spécification de liste est identique à celle du paramètre *load_thresholds* plus haut ; c'est-à-dire *low_pri_q=5,small_q*. Les nombres indiquent les emplacements de travail de la file qui doivent être remplis pour déclencher l'interruption de la file subordonnée. Si aucune valeur n'est attribuée une interruption est déclenchée lorsque tous les emplacements de la file sont remplis.

Sur les noeuds qui hébergent plusieurs files d'attente, il se peut que vous vouliez accorder un meilleur service à certaines catégories de travaux (par ex. les files d'attente dédiées au traitement parallèle peuvent requérir la priorité sur les files de production peu prioritaires ; la valeur par défaut est *NONE*).

- *complex_list* (liste des complexes) – Liste séparée par des virgules de complexes définis par l'administrateur (voir *complex(5)* pour plus de détails) devant être associée à la file d'attente. Seuls les attributs de complexe contenus dans les complexes listés et ceux des complexes *global*, *hôte* et *file*, qui sont rattachés de façon implicite à chaque file, peuvent être utilisés dans la liste *complex_values* ci-dessous.

La valeur par défaut de ce paramètre est *NONE* ; c'est-à-dire qu'aucun complexe défini par l'administrateur n'est associé à la file.

complex_values (valeurs de complexes) – Les *complex_values* définissent les quotas des attributs de ressources gérés via cette file. Les attributs de complexe autorisés à apparaître dans *complex_values* sont définis par la *complex_list* (voir ci-dessus). La syntaxe est la même que pour *load_thresholds* (voir ci-dessus). Les quotas sont liés à la consommation de ressources de tous les travaux d'une file d'attente dans le cas des ressources consommables (voir *complex(5)* pour plus de détails sur les ressources consommables) ou sont interprétés par emplacement. La définition du quota dans la liste *complex_values* est automatiquement remplacée par la valeur de charge courante rapportée pour cet attribut, si la charge est surveillée pour cette ressource et si la valeur de charge rapportée est plus contraignante que le quota. Cela évite de manière efficace la sur-souscription des ressources.

Remarque – Les valeurs de charge qui remplacent les spécifications de quotas peuvent être devenues plus contraignantes suite à leur mise à l'échelle (voir `host_conf(5)`) et/ou à des ajustements de charge (voir `sched_conf(5)`). L'option `-F` de `qstat(1)` et l'affichage de la charge dans la boîte de dialogue de contrôle des files d'attente de `qmon(1)` (qui s'ouvre en cliquant sur l'icône d'une file tout en appuyant sur la touche « Maj ») fournissent des informations détaillées sur la disponibilité courante des ressources consommables et sur l'origine des valeurs couramment prises en compte.

Remarque – La consommation de ressources des travaux en cours d'exécution (utilisée pour le calcul de la disponibilité) ainsi que les requêtes des travaux en attente d'être acheminés peuvent être dérivées des requêtes explicites des utilisateurs pendant la soumission des travaux (voir l'option `-l` de `qsub(1)`) ou de la valeur « par défaut » configurée pour un attribut par l'administrateur (voir `complex(5)`). L'option `-r` de `qstat(1)` peut être utilisée pour récupérer des détails complets sur les requêtes de ressources courantes de tous les travaux au sein du système.

Pour les ressources non-consommables, Sun Grid Engine compare tout simplement les requêtes des attributs du travail à la spécification correspondante dans `complex_values` en tenant compte de l'opérateur de relation de la définition de l'attribut de complexe (voir `complex(5)`). Si le résultat de la comparaison est « true », l'hôte est adapté au travail en ce qui concerne l'attribut considéré. Pour les travaux parallèles, chaque emplacement de travail devant être occupé par une tâche parallèle est sensé fournir la même valeur d'attribut de ressource.

Remarque – Seuls les attributs de complexe numériques peuvent être définis en tant que ressources consommables et, par conséquent, les attributs non-numériques sont toujours gérés par emplacement de file.

La valeur par défaut de ce paramètre est NONE, aucun quota d'attribut de ressource défini par l'administrateur n'est associé à la file.

- *calendar* (agenda) – Spécifie le *calendar* valide pour cette file d'attente ou contient NONE (la valeur par défaut). Un agenda définit la disponibilité d'une file en fonction de l'heure du jour, de la semaine et de l'année. Reportez-vous à `calendar_conf(5)` pour plus de détails sur l'utilitaire d'agendas de Sun Grid Engine.

Remarque – Les travaux peuvent demander des files d'attente avec un certain modèle d'agenda via une option `-l c=<nom_agenda>` de `qsub(1)`.

- *initial_state* (état initial) – Définit l'état initial de la file d'attente soit au moment où la file est ajoutée pour la première fois au système soit au démarrage de `sgexecd(8)` sur l'hôte sur lequel la file réside. Les valeurs possibles sont les suivantes :

- *Default* – La file est activée au moment de l’ajout de la file ou est réinitialisée à son statut préalable lorsque `sgexecd(8)` est activé (cela correspond au comportement en vigueur dans les versions antérieures de Sun Grid Engine qui ne prenaient pas en charge `initial_state`).
- *Enabled* – La file est activée dans les deux cas. Est équivalent à une commande `qmod -e` manuelle et explicite (voir `qmod(1)`).
- *Disabled* – La file est désactivée dans les deux cas. Est équivalent à une commande `qmod -d` manuelle et explicite (voir `qmod(1)`).
- *fshare* – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.
Il s’agit des partages fonctionnels de la file (par ex. la catégorie de travaux). Les travaux s’exécutant dans cette file d’attente peuvent recevoir des tickets fonctionnels dérivés de l’importance relative de la file si la politique fonctionnelle est activée.
- *oticket* – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.
Il s’agit des tickets de remplacement de la file (par ex. les classes de travaux). Sun Grid Engine, Enterprise Edition distribue la quantité configurée de tickets de remplacement entre les travaux s’exécutant dans cette file d’attente.

Limites de ressources

Les deux premiers paramètres de limite des ressources, `s_rt` et `h_rt`, sont mis en oeuvre par Sun Grid Engine. Ils définissent le « temps réel », aussi dit « écoulé » ou « chronométré », écoulé depuis le début du travail. Si `h_rt` est dépassé par un travail en cours d’exécution dans la file d’attente, ce travail est abandonné via le signal `SIGKILL` (voir `kill(1)`). Si `s_rt` est dépassé, le travail est d’abord « averti » via le signal `SIGUSR1` (qui peut être saisi par le travail) puis abandonné au bout d’un délai d’avertissement défini dans le paramètre *notify* (voir ci-dessus) du fichier de configuration.

Les paramètres de limite des ressources `s_cpu` et `h_cpu` sont mis en oeuvre par Sun Grid Engine en tant que limite de travail. Ils imposent une limite sur la quantité de temps UC consommée par tous les processus d’un travail. Si `h_cpu` est dépassé par un travail en exécution dans la file d’attente, ce travail est abandonné via un signal `SIGKILL` (voir `kill(1)`). Si `s_cpu` est dépassé, un signal `SIGXCPU` est envoyé au travail qui peut le saisir. Si vous voulez autoriser un travail à être « averti » de façon à pouvoir s’arrêter normalement avant d’être éliminé, vous devez mettre la limite `s_cpu` sur une valeur inférieure à `h_cpu`. Pour les processus parallèles, la limite est appliquée par emplacement, ce qui signifie qu’elle est multipliée par le nombre d’emplacements utilisés par le travail avant d’être appliquée.

Les paramètres de limite de ressources `s_vmem` et `h_vmem` sont mis en oeuvre par Sun Grid Engine en tant que limite de travail. Ils imposent une limite au total de mémoire de virtuelle combiné consommé par tous les processus du travail. Si `h_vmem` est dépassé par un travail en cours d'exécution dans la file d'attente, ce travail est abandonné avec un signal `SIGKILL` (voir `kill(1)`). Si `s_vmem` est dépassé, un signal `SIGXCPU` est envoyé au travail qui peut le saisir. Si vous voulez autoriser un travail à être « averti » de façon à pouvoir s'arrêter normalement avant d'être éliminé, vous devez mettre la limite `s_vmem` sur une valeur inférieure à `h_vme`. Pour les processus parallèles, la limite est appliquée par emplacement, ce qui signifie qu'elle est multipliée par le nombre d'emplacements utilisés par le travail avant d'être appliquée.

Les paramètres restants du modèle de configuration de file spécifient les limites fixes et souples par travail telles que mises en oeuvre par l'appel système `setrlimit(2)`. Consultez cette page de manuel sur votre système pour plus d'informations. Par défaut, chaque champ de limite est sur l'infini (qui signifie `RLIM_INFINITY` comme décrit dans la page de manuel `setrlimit(2)`). Le type de valeur des limites temps-UC est `s_cpu` et `h_cpu` est temporel. Le type de valeur des autres limites est mémoire.

Remarque – Tous les systèmes ne prennent pas en charge `setrlimit(2)`.

Remarque – `s_vmem` et `h_vmem` (mémoire virtuelle) ne sont disponibles que sur les systèmes prenant en charge `RLIMIT_VMEM` (voir `setrlimit(2)` sur votre système d'exploitation).

Le système d'exploitation UNICOS fourni par SGI/Cray ne prend pas en charge l'appel système `setrlimit(2)`, car il utilise son propre appel système de fixation de limite. Pour les systèmes UNICOS uniquement, les éléments suivants s'appliquent.

- `s_cpu` – Temps UC limite par processus en secondes.
- `s_core` – Taille de fichier de base maximale par processus en octets.
- `s_data` – Mémoire maximale limite par processus en octets.
- `s_vmem` – Identique à `s_data` (si tous deux sont définis, le plus bas est utilisé).
- `h_cpu` – Temps UC limite par travail en secondes.
- `h_data` – Mémoire maximale limite par travail en octets.
- `h_vmem` – Identique à `h_data` (si tous deux sont définis, le plus bas est utilisé).
- `h_fsize` – Nombre total de blocs de disque que ce travail peut créer.

Voir aussi

`sge_intro(1)`, `csh(1)`, `qconf(1)`, `qmon(1)`, `qrestart(1)`, `qstat(1)`,
`qsub(1)`, `sh(1)`, `nice(2)`, `setrlimit(2)`, `access_list(5)`,
`calendar_conf(5)`, `sge_conf(5)`, `complex(5)`, `host_conf(5)`,
`sched_conf(5)`, `sge_execd(8)`, `sge_qmaster(8)`, `sge_shepherd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sched_conf (5)

Nom

`sched_conf` – Fichier de configuration du programmeur par défaut de Sun Grid Engine.

Description

`sched_conf` définit le format du fichier de configuration pour le programmeur par défaut du système de Sun Grid Engine fourni par `sge_schedd(8)`. Pour modifier la configuration, utilisez l'IHM `qmon(1)` ou l'option `-msconf` de la commande `qconf(1)`. Une configuration par défaut est fournie avec le logiciel Sun Grid Engine.

Format

Les paramètres suivants sont reconnus par le programmeur de Sun Grid Engine si présents dans `sched_conf`:

- *algorithm* (algorithme) – Permet de sélectionner d'autres algorithmes de programmation. Actuellement, `default` est le seul paramétrage autorisé.
- *load_formula* (formule de charge) – Expression algébrique simple utilisée pour dériver une unique valeur de charge pondérée de tout ou partie des paramètres de charge rapportés par `sge_execd(8)` pour chaque hôte et depuis tout ou partie des ressources consommables (voir `complex(5)`) maintenues pour chaque hôte. La syntaxe de la formule de charge est celle d'une somme de valeurs de charge pondérées, c'est-à-dire :

```
val_charge1[*w1][{+|-}val_charge2[*w2][{+|-}...]]
```

Remarque – Les blancs ne sont pas autorisés dans les formules de charge.

Les valeurs de charge et les ressources consommables (`val_charge1, ...`) sont spécifiées par le nom défini dans le complexe (voir `complex(5)`).

Remarque – Les valeurs de charge définies par l'administrateur (voir le paramètre `load_sensor` dans `sgc_conf(5)` pour plus de détails) et les ressources consommables pour tous les hôtes (voir `complex(5)`) peuvent être tout aussi bien utilisées que les paramètres de charge par défaut de Sun Grid Engine.

Les facteurs de pondération (`w1, ...`) sont des entiers positifs. Une fois que l'expression a été évaluée pour chaque hôte, les résultats sont attribués aux hôtes puis utilisés pour trier les hôtes correspondant à la charge pondérée. La liste des hôtes triées est ensuite utilisée pour trier les files d'attente.

La formule de charge par défaut est `load_avg`.

- *job_load_adjustments* (ajustements de charge) – La charge imposée par les travaux Sun Grid Engine en cours d'exécution sur un système varie dans le temps et, souvent, par ex. pour la charge UC, requiert qu'une certaine quantité de temps soit reportée dans la quantité appropriée dans le système d'exploitation. Par conséquent, si un travail a été démarré très récemment, la charge rapportée peut ne pas fournir une représentation suffisante de la charge qui est déjà imposée sur cet hôte par le travail. La charge rapportée s'adaptera dans le temps à la charge réelle, mais pendant la période de temps pendant laquelle la charge rapportée est trop basse pourra conduire à la sur-souscription de l'hôte. Pour compenser ce problème, Sun Grid Engine permet à l'administrateur de spécifier des *job_load_adjustments* qui sont utilisés dans le programmeur de Sun Grid Engine.

Les *job_load_adjustments* sont spécifiés par une liste séparée par des virgules de paramètres de charge arbitraires ou de ressources consommables suivis (avec un signe égal de séparation) des valeurs de correction de charge *y* associées. À chaque fois qu'un travail est acheminé sur un hôte par `sgc_schedd(8)`, l'ensemble des paramètres de charge et des valeurs de consommables de cet hôte est augmenté avec les valeurs fournies dans la liste *job_load_adjustments*. Ces valeurs de correction deviennent obsolètes de façon linéaire dans le temps, et du début à l'échéance du *load_adjustment_decay_time* les corrections diminuent progressivement jusqu'à atteindre la valeur 0. Si la liste *job_load_adjustments* se voit attribuer le dénominateur spécial NONE, aucune correction de charge n'est effectuée.

La charge ajustée et les valeurs consommables sont utilisées pour calculer la charge combinée et pondérée des hôtes avec la *load_formula* (voir ci-dessus) et pour comparer les valeurs de charge et consommables aux listes de seuils de charge définies dans la configuration de file (voir `queue_conf(5)`). Si votre *load_formula* se compose simplement du paramètre de charge UC moyenne *load_avg* et si vos travaux requièrent une puissance de calcul élevée, vous pouvez mettre la liste *job_load_adjustments* sur *load_avg=100*, ce qui signifie que chaque nouveau travail acheminé sur un hôte requerra 100% du temps UC et, par conséquent, la charge de la machine sera instantanément augmentée à 100.

- *load_adjustment_decay_time* (temps de déclin d'ajustement de charge) – Les corrections de charge dans la liste « *job_load_adjustments* » ci-dessus deviennent obsolètes de façon progressive entre le moment où démarre le travail, où le paramètre de charge ou de consommable correspondant est augmenté à la valeur de correction, et l'arrivée à échéance du délai « *load_adjustment_decay_time* » où la correction devient nulle (0). Les valeurs appropriées de « *load_adjustment_decay_time* » dépendent grandement des paramètres de charge ou de consommables utilisés et du ou des systèmes d'exploitation spécifiques. Elles ne peuvent donc qu'être déterminées sur le terrain et par l'expérience. Pour le paramètre de charge *load_avg* par défaut, il a été démontré qu'un « *load_adjustment_decay_time* » de 7 minutes donnait des résultats satisfaisants.
- *maxujobs* – Nombre maximal de travaux qu'un utilisateur quelconque peut avoir simultanément en exécution dans une grappe Sun Grid Engine. Si ce paramètre est sur 0 (valeur par défaut), les utilisateurs peuvent exécuter un nombre arbitraire de travaux.
- *schedule_interval* (intervalle de programmation) – Temps que *sge_schedd(8)* enregistre au départ auprès de *sge_qmaster(8)*. *schedule_interval* est utilisé pour fixer l'intervalle de temps pendant lequel *sge_qmaster(8)* envoi des mises à jour d'événements de programmation à *sge_schedd(8)*. Un événement de programmation est un changement de statut qui est survenu au sein de *sge_qmaster(8)* pouvant déclencher ou affecter les décisions du programmeur (par ex., un travail s'est terminé et les ressources allouées sont donc de nouveau disponibles).

Dans le programmeur par défaut de Sun Grid Engine, l'arrivée d'un rapport d'événement de programmation déclenche un cycle du programmeur. Le programmeur attend sinon les rapports d'événements.

Schedule_interval est une valeur temporelle (voir *queue_conf(5)* pour la définition de la syntaxe des valeurs temporelles).

- *user_sort* (trié/e(s) par l'utilisateur) – Le système Sun Grid Engine programme en général les travaux des utilisateurs en fonction d'une politique de type premier-arrivé-premier servi. Dans le cas où un utilisateur soumettrait une grande quantité de travaux dans un très court laps de temps, cela pourrait mener à une situation plutôt injuste, puisque tous les utilisateurs qui soumettraient des travaux après ce premier utilisateur seraient bloqués jusqu'à ce que la plupart des travaux du premier soient terminés. Sun Grid Engine vous permet, par conséquent, de changer cette politique en un tri dit à parts égales : dès que l'un des travaux d'un utilisateur est en cours le reste de ses travaux est renvoyé à la fin de la liste des travaux en attente. Par conséquent, les premiers travaux de tous les autres utilisateurs ont les mêmes chances de trouver une file d'attente.

Remarque – Le tri à parts égales ne s'applique qu'au sein d'une catégorie de travaux de priorité donnée (voir l'option *mp* des commandes *qsub(1)* et *qalter(1)* pour plus d'informations).

La valeur par défaut de *user_sort* est *FALSE*.

- *queue_sort_method* (méthode de tri de files) – Ce paramètre détermine l'ordre dans lequel les différents critères sont pris en compte pour constituer une liste de files triée. Deux paramétrages sont admis : *seq_no* et *load*. Dans ces deux cas, Sun Grid Engine essaye de maximiser le nombre de demandes souples (voir l'option *qsub(1) -s*) déposées par les files d'attente pour un travail particulier comme premier critère.
 - Si le paramètre *queue_sort_method* est sur *seq_no*, le logiciel Sun Grid Engine utilisera le paramètre *seq_no* tel que configuré dans les configurations des files courantes (voir *queue_conf(5)*) comme premier critère pour trier la liste des files. La *load_formula* (voir plus haut) n'a de sens que si deux files ont le même numéro de séquence.
 - Si le paramètre *queue_sort_method* est sur *load*, la charge qu'accorde la *load_formula* est le premier critère après la maximisation des requêtes souples d'un travail. Avec ce paramétrage, le numéro de séquence ne sera utilisé que si deux hôtes ont la même charge.

Le tri par numéro de séquence convient plus particulièrement si vous voulez définir un ordre fixe pour le remplissage des files d'attente (par ex. : les ressources les moins chères en premier). La valeur par défaut de ce paramètre est *load*.

- *sgeee_schedule_interval* (intervalle de programmation sgeee) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Il s'agit de la période de temps entre les ajustements de priorité des travaux effectués par le programmeur dynamique de Sun Grid Engine, Enterprise Edition. Pour être valides, les valeurs spécifiées doivent être de type temporel comme spécifié dans *queue_conf(5)*.

- *halftime* (demi-vie) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Dans le cadre d'une exécution soumise à une politique basée sur le partage Sun Grid Engine, Enterprise Edition « date » (c'est-à-dire diminue) l'utilisation pour mettre en oeuvre une plage glissante pour obtenir les droits aux partages tels que défini par l'arborescence des partages. Le paramètre *halftime* définit l'intervalle de temps pendant lequel l'utilisation accumulée aura diminué de la moitié de sa valeur d'origine. Pour être valides, les valeurs spécifiées doivent être de type temporel comme spécifié dans *queue_conf(5)*.

- *usage_weight_list* (liste des pondérations) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Sun Grid Engine, Enterprise Edition tient compte de la consommation des ressources temps-UC, mémoire et E/S pour déterminer l'utilisation qui est imposée sur un système par un travail. Une unique valeur d'utilisation est calculée à partir de ces trois paramètres d'entrée en multipliant les valeurs individuelles par des indices de pondération et en les additionnant. Les indices de pondération sont définis dans la *usage_weight_list*. Le format de cette liste est le suivant :

cpu=puc,mem=pmém,io=pes

Où *puc*, *pmém* et *pes* sont des indices de pondération configurables. Les indices de pondération sont des nombres réels. La somme des trois indices de pondération devrait être égale à 1.

- *compensation_factor* (facteur de compensation) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Détermine la rapidité avec laquelle Sun Grid Engine, Enterprise Edition doit compenser toute utilisation passée au-dessous ou au-dessus du droit aux partages défini dans l'arborescence des partages. Les valeurs recommandées sont situées entre 2 et 10, 10 correspondant à la compensation la plus rapide.

- *weight_user* (pondération utilisateur) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Importance relative des partages d'utilisateur dans la politique fonctionnelle. Les valeurs sont de type réel.

- *weight_project* (pondération projet) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Importance relative des partages de projets dans la politique fonctionnelle. Les valeurs sont de type réel.

- *weight_jobclass* (pondération catégorie travaux) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Importance relative des partages de catégories de travaux (par ex. file) dans la politique fonctionnelle. Les valeurs sont de type réel.

- *weight_department* (pondération service) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Importance relative des partages de service dans la politique fonctionnelle. Les valeurs sont de type réel.

- *weight_job* (pondération travail) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Importance relative des partages de travaux dans la politique fonctionnelle. Importance relative des partages.

- *weight_tickets_functional* (pondération tickets fonctionnels) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Nombre maximal de tickets fonctionnels disponibles pouvant être distribués par Sun Grid Engine, Enterprise Edition. Détermine l'importance relative de la politique fonctionnelle.

- *weight_tickets_share* (pondération tickets partage) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Nombre maximal de tickets basés sur le partage disponibles pouvant être distribués par Sun Grid Engine, Enterprise Edition. Détermine l'importance relative de la politique d'arborescence des partages.

- *weight_deadline* (pondération initiation limite) – Ce paramètre est uniquement disponible dans un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine ne le prend pas en charge.

Nombre maximal de tickets d'initiation limite pouvant être distribués par Sun Grid Engine, Enterprise Edition. Détermine l'importance relative de la politique d'initiation limite.

- *schedd_job_info* (informations de programmation des travaux) – Le programmeur par défaut peut garder la trace des raisons pour lesquelles des travaux n'ont pas pu être programmés pendant le dernier cycle du programmeur. Ce paramètre active ou désactive l'observation. La valeur `true` active la surveillance, `false` la désactive.

Il est également possible d'activer l'observation uniquement pour certains travaux. Cela sera fait si le paramètre est mis sur *job_list* suivi par une liste séparée par des virgules d'ID de travail.

L'utilisateur peut obtenir les informations collectées avec la commande `qstat -j`.

Fichiers

- Configuration de `sge_schedd` –
`<racine_sge>/<cellule>/common/sched_configuration`

Voir aussi

`sge_intro(1)`, `qalter(1)`, `qconf(1)`, `qstat(1)`, `qsub(1)`, `complex(5)`, `queue_conf(5)`, `sge_execd(8)`, `sge_qmaster(8)`, `sge_schedd(8)`, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3* et *Guide d'administration et d'utilisation de Sun Grid Engine Enterprise Edition 5.3*.

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

share_tree(5)

Nom

`share_tree` – Format des fichiers d'arborescence de partages de Sun Grid Engine, Enterprise Edition.

Description

L'objet arborescence des partages n'est disponible que dans le cas d'un système Sun Grid Engine, Enterprise Edition. Sun Grid Engine n'a pas d'objet d'arborescence des partages.

L'arborescence des partages définit les droits à long terme aux ressources des utilisateurs/projets et d'une hiérarchie de groupes arbitraires de ces éléments.

L'arborescence de partages courante peut être affichée via l'option `qconf(1) -sstree`. La sortie suit la description du format `share_tree`. Il est possible de créer une arborescence de partages ou d'en modifier une existante via les options `-astree` et `-mstree` de `qconf`. Des noeuds d'arborescence des partages individuels peuvent être créés, modifiés ou supprimés via les options `-astnode`, `-dstnode`, `-mstnode` et `-sstnode` de `qconf(1)`.

Format

Le format d'un fichier d'arborescence des partages est le suivant :

- Un nouveau noeud commence par l'attribut `id`, suivi du signe égal et du numéro d'identification du noeud. Les autres attributs du noeud suivent jusqu'à ce qu'un autre mot-clé `id` soit rencontré.
- L'attribut `childnodes` contient une liste séparée par des virgules des noeuds fils de ce noeud.
- Le paramètre `name` fait référence à un nom arbitraire pour le noeud ou à un utilisateur (voir `user(5)`) ou un projet (voir `project(5)`) correspondant si le noeud est un noeud final de l'arborescence des partages. Le nom du noeud racine de l'arborescence est par convention « Root » (racine).
- Le paramètre `shares` définit le partage du noeud entre les noeuds ayant le même noeud père.

- Un noeud final d'utilisateur appelé *default* peut être défini en tant que descendant d'un noeud `project(5)` dans l'arborescence des partages. Ce noeud par défaut définit le nombre de partages pour les utilisateurs qui participent au projet, mais n'ont pas de noeud d'utilisateur défini sous le projet. Le noeud d'utilisateur par défaut est une manière pratique de spécifier un noeud unique pour tous les utilisateurs devant recevoir une part égale des ressources d'un projet. Le noeud *default* peut être spécifié en tant que tel ou avec d'autres noeuds `user(5)` du même niveau sous un projet. Tous les utilisateurs, qu'ils soient explicitement spécifiés par un noeud d'utilisateur ou soient mappés au noeud d'utilisateur *default*, doivent avoir un objet `user(5)` correspondant défini afin d'obtenir les partages. Ne configurez pas un objet `user(5)` qui soit explicitement appelé *default*.

Voir aussi

`sge_intro(1)`, `qconf(1)`, `project(5)`, `user(5)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

`user(5)`

Nom

`user` – Format des fichiers d'entrée d'utilisateur de Sun Grid Engine, Enterprise Edition.

Description

L'objet utilisateur est uniquement disponible dans le cas d'un système Sun Grid Engine, Enterprise Edition. Le système Sun Grid Engine n'a pas d'objet utilisateur.

Une entrée d'utilisateur est utilisée dans Sun Grid Engine, Enterprise Edition pour stocker les informations relatives aux tickets et à l'utilisation par utilisateur. Maintenir des entrées d'utilisateur pour tous les utilisateurs participant à un système Sun Grid Engine, Enterprise Edition est requis si Sun Grid Engine, Enterprise Edition est utilisé sous une politique d'arborescence de partages d'utilisateur.

La liste des entrées d'utilisateurs couramment configurées peut être affichée via l'option `qconf(1) -suser1`. Le contenu de chaque entrée d'utilisateur de la liste peut être affiché via le modificateur `-suser`. La sortie suit la description du format d'utilisateur. De nouvelles entrées d'utilisateur peuvent être créées tandis que celles existantes peuvent être modifiées via les options `-auser`, `-muser` et `-duser` de `qconf(1)`.

Format

Une entrée d'utilisateur contient quatre paramètres :

- *name* - Nom de l'utilisateur.
- *oticket* - Quantité de tickets de remplacement actuellement attribués à l'utilisateur.
- *fshare* - Partage fonctionnel courant de l'utilisateur.
- *default_project* - Projet par défaut de l'utilisateur.

Voir aussi

`sge_intro(1)`, `qconf(1)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

usermapping(5)

Nom

`usermapping` - Format des fichiers d'entrée de mappage d'utilisateurs.

Description

Une entrée `usermapping` est utilisée pour définir les alias pour un utilisateur de grappe. Le nom de l'utilisateur — tel que le connaît le système de programmation — est appelé *cluster user*. Si le nom de grappe ne correspond pas au nom du compte d'utilisateur sur un hôte d'exécution, la fonctionnalité de mappage `usermapping` peut résoudre le problème.

Chacune des lignes du fichier d'entrées `usermapping` spécifie un nom d'utilisateur et le ou les hôtes sur lesquels il ou elle a un compte.

Vous pouvez afficher la liste des entrées `usermapping` couramment configurées via l'option `-sumapl` de `qconf(1)`. Le contenu de chaque entrée `usermapping` listée peut être affiché via le modificateur `-sumap`. La sortie suit la description du format `usermapping`. Vous pouvez créer ou modifier les entrées d'utilisateur via les options `-aumap`, `-mumap` et `-dumap` de `qconf(1)`.

Format

Une entrée `usermapping` contient au moins trois paramètres.

- le mot clé `cluster_user` (utilisateur grappe) – Le mot clé `cluster_user` définit le nom d'utilisateur de grappe. La partie de la ligne de texte qui suit le mot clé `cluster_user` sera prise en tant que valeur d'utilisateur de grappe.
- `name` (nom) – Nom de l'utilisateur sur un hôte d'exécution.
- `hostname` (nom d'hôte) – Nom de l'hôte où l'utilisateur spécifié dans `name` a un compte. Si le premier caractère du nom d'hôte est le signe `@`, le nom est utilisé pour faire référence à un `hostgroup(5)`.

Exemple

L'exemple qui suit est une entrée `usermapping` typique pour un mappage d'utilisateur de grappe.

```
cluster_user bureau
alice merveilles, monderéel, @bureau2
```

Cette entrée mapperà l'utilisateur, `bureau`, qui est défini dans le système de grappe à l'utilisateur, `alice`, sur les hôtes `merveilles` et `monderéel`. Le signe `@` est utilisé pour faire référence à un groupe d'hôtes. Ainsi l'utilisateur, `bureau`, est mappé à `alice` sur tous les hôtes spécifiés dans le groupe d'hôtes, `bureau2`. Pour plus d'informations sur ce point, voir `hostgroup(5)`.

Voir aussi

`qconf(1)`, `hostgroup(5)`

Copyright

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sge_commd(8)

Nom

`sge_commd` – Agent de communication de Sun Grid Engine

Synopsis

```
sge_commd [ -S ] [ -a fichieralias ] [ -dhr ] [ -help ]  
[ -ll niveaucons ] [ -ml nomf ] [ -nd ] [ -p port ]  
[ -s service ]
```

Description

Au sein d'une grappe Sun Grid Engine, toutes les communications réseau s'effectuent via les démons de communication `sge_commd`. Les programmes clients tels que `qsub(1)` ou `qstat(1)` ainsi que les démons de Sun Grid Engine tels que `sge_qmaster(8)` ou `sge_execd(8)` utilisent le service fourni par `sge_commd` pour envoyer/recevoir des messages aux/des composants de Sun Grid Engine.

`sge_commd` gère un nombre arbitraire de communications synchrones ou asynchrones simultanées. En général, un `sge_commd` est démarré automatiquement sur chaque hôte sur lequel `sge_qmaster(8)`, `sge_execd(8)` ou/et `sge_schedd(8)` sont appelés. Il est toutefois possible de connecter plusieurs hôtes via un `sge_commd` ou d'utiliser un `sge_commd` sur un hôte de soumission ou administratif Sun Grid Engine (sans exécuter un des autres démons de Sun Grid Engine) en tant qu'agent de communication pour les programmes clients Sun Grid Engine appelés depuis cet hôte.

Options

Le TABLEAU 29 liste et décrit les options associées à `sgc_commd`.

TABLEAU 29 Options de `sgc_commd`

Option	Description
-S	Impose l'utilisation de ports sécurisés pour la communication entre <code>sgc_commd</code> et entre d'autres composants Sun Grid Engine et les <code>sgc_commds</code> . Pour ce, tous les démons de Sun Grid Engine doivent être démarrés avec une permission de super-utilisateur et les programmes clients configurés avec <code>set-uid root</code> . En contrepartie, cette option assure l'interdiction de toute communication non-autorisée pour les comptes autres que celui du super-utilisateur.
-a <i>fichieralias</i>	Fichier contenant les alias des hôtes de Sun Grid Engine utilisés par <code>sgc_commd</code> pour résoudre les noms d'hôtes Sun Grid Engine uniques pour tous les hôtes de la grappe. Remplace l'utilisation du fichier <code>host_aliases</code> par défaut sous <code><racine_sgc>/<cellule>/common/host_aliases</code> . Le service de résolution des noms d'hôtes de <code>sgc_commd</code> est également utilisé par tous les autres composants de Sun Grid Engine. Le format de ce fichier et les conséquences de son utilisation sont décrits dans <code>sgc_h_aliases(5)</code> .
-dhr	Les fonctions de résolution des noms d'hôtes C-library (par ex. : <code>gethostent(3)</code> , <code>gethostbyname(3)</code> et <code>gethostbyaddr(3)</code>) effectuent une sorte de mise en cache sur certaines architectures SE. Les bases de données de noms d'hôtes de réseau distribuées par des services tels que DNS (Domain Name Service) et NIS sont mises à jour avec un délai de plusieurs minutes. Cela affecte uniquement les applications qui résolvent de façon répétée des noms d'hôtes (par ex. : <code>sgc_commd</code>). Au démarrage d'un programme, les informations les plus récentes sont accédées ; par conséquent des commandes telles que <code>telnet(1)</code> ou <code>nslookup(1)</code> ne sont pas affectées. Cependant, pour <code>sgc_commd</code> résoudre les noms d'hôtes à tout moment n'a pas de sens (les informations retournées risquent d'être quand même obsolètes) et la résolution peut se révéler coûteuse si le réseau est surchargé et/ou les serveurs NIS ou DNS sont très occupés. Par conséquent, <code>sgc_commd</code> résout les informations de noms d'hôtes seulement de temps en temps. Enfin, si la résolution des noms d'hôtes continue à poser des problèmes à cause de, par exemple, la charge du réseau, elle peut être désactivée avec le commutateur <code>-dhr</code> . L'administrateur doit être conscient, que si la résolution des noms d'hôtes est désactivée, <code>sgc_commd</code> doit être redémarré dès que la base de données des noms d'hôtes change sensiblement.
-help	Imprime la liste de toutes les options.
-ll <i>niveaucons</i>	Définit un niveau de consignation pour le suivi des erreurs. Les informations de suivi sont écrites dans le fichier <code>/tmp/commd/err.<pid></code> . Le répertoire <code>/tmp/commd</code> doit être présent, sinon la sortie de suivi sera éliminée. Actuellement, 255 est le seul niveau de consignation valide.

TABLEAU 29 Options de `sge_cmmd` (suite)

Option	Description
<code>-ml nomf</code>	Active la consignation des messages dans le fichier.
<code>-nd</code>	Ne pas démoniser. Si démarré avec <code>-nd</code> , <code>sge_cmmd</code> maintient sa connexion avec le terminal de contrôle et fait sortir les informations de suivi directement sur le terminal depuis lequel <code>sge_cmmd</code> a été appelé. Les informations de suivi sont les mêmes que celles accessibles via l'option <code>-ll</code> (voir ci-dessus).
<code>-p numéro_port</code>	Utilise ce port TCP pour la communication avec d'autres commandes.
<code>-s nom_service</code>	Utilise ce nom de service et le port TCP y associé pour la communication avec d'autres <code>cmds</code> .

Variables d'environnement

Le TABLEAU 30 liste les variables d'environnement associées à `sge_cmmd`.

TABLEAU 30 Variables d'environnement de `sge_cmmd`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_cmmd</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	(n'affecte pas le comportement de <code>sge_cmmd</code> mais celui des autres composants de Sun Grid Engine qui contactent <code>sge_cmmd</code> .) Si elle est définie, spécifie l'hôte sur lequel le <code>sge_cmmd</code> particulier à utiliser pour la communication Sun Grid Engine de programmes clients ou démons arbitraires de Sun Grid Engine réside. Par défaut, l'hôte local est utilisé.

Restrictions

`sge_cmmd` est en général appelé par un `sge_qmaster(8)` et un `sge_execd(8)` en cours de démarrage et s'exécute donc sous des droits de super-utilisateur. S'il est démarré par un utilisateur normal, le modificateur `-S` risque de ne pas être autorisé car le mode sécurisé requiert des droits d'accès de super-utilisateur (voir ci-dessous).

Voir aussi

`sgc_intro(1)`, `sgc_h_aliases(5)`, `sgc_execd(8)`, `sgc_qmaster(8)`,
`commdctl(8)`

Droits d'auteur

Voir `sgc_intro(1)` pour l'énoncé complet des droits et autorisations.

sgc_execd(8)

Nom

`sgc_execd` - Agent d'exécution de travaux de Sun Grid Engine

Synopsis

```
sgc_execd [ -help ] [ -lj fichier_journal ] [ -nostart-commande ]
```

Description

`sgc_execd` contrôle les files d'attente de Sun Grid Engine locales de la machine sur laquelle `sgc_execd` est en cours d'exécution et exécute/contrôle les travaux envoyés par `sgc_qmaster(8)` devant s'exécuter sur ces files.

`sgc_commd(8)` est automatiquement activé avec `sgc_execd` sur la même machine (s'il n'est pas déjà présent).

Options

Le TABLEAU 31 liste les options associées à `sge_execd`.

TABLEAU 31 Options de `sge_execd`

Option	Description
<code>-help</code>	Imprime la liste de toutes les options.
<code>-lj fichier_journal</code>	Active la consignation des travaux. Toutes les actions entreprises par <code>sge_execd</code> de la réception du travail à son retour à <code>sge_qmaster(8)</code> sont consignés dans le <code>fichier_journal</code> . Cette fonctionnalité est également disponible avec le démon <code>sge_qmaster(8)</code> .
<code>-nostart-commande</code>	Ne démarre pas automatiquement <code>sge_commd(8)</code> avec <code>sge_execd</code> et évalue la variable d'environnement <code>COMMD_HOST</code> pour trouver le <code>sge_commd(8)</code> correspondant.

Capteurs de charge

Si un capteur de charge est configuré pour `sge_execd` via la configuration de grappe globale ou celle spécifique de l'hôte d'exécution (voir `sge_conf(5)`), le chemin de l'exécutable du capteur de charge est appelé régulièrement par `sge_execd` et délivre un ou plusieurs chiffres de charge pour l'hôte d'exécution (par ex. les utilisateurs actuellement connectés) ou sur l'ensemble de la grappe (par ex. l'espace disque libre sur un système de fichiers de travail temporaires à l'échelle du réseau). Le capteur de charge peut être un script ou un exécutable binaire. Dans les deux cas, sa gestion des flux `STDIN` et `STDOUT` et son flux de contrôle doivent être conformes aux règles qui suivent.

Le capteur de charge doit être écrit sous la forme d'une boucle infinie attendant à un certain point une entrée de `STDIN`. Si la chaîne `quit` est lue de `STDIN`, le capteur de charge est supposé se terminer. Dès qu'une fin de ligne est lue dans `STDIN` un cycle de récupération des données de charge est supposé démarrer. Le capteur de charge effectue ensuite les opérations nécessaires pour calculer les chiffres de charge désirés. À la fin du cycle, le capteur de charge écrit le résultat dans `stdout`. Le format est le suivant.

- Un rapport de valeur de charge commence par une ligne contenant uniquement le mot `start`.
- Les valeurs de charge individuelles sont séparées par de nouvelles lignes.
- Chaque rapport de charge se compose de trois parties séparées par des signes deux-points (`:`) et ne contenant pas de blancs.
- La première partie d'une information de valeur de charge est soit le nom de l'hôte pour lequel la charge est rapportée, soit le nom spécial, `global`.

- La seconde partie est le nom symbolique de la valeur de charge tel que défini dans la liste des complexes hôte ou global (voir `complex(5)` pour plus de détails). S'il n'existe pas d'entrée dans la liste de complexes hôte ou global correspondant à une valeur de charge rapportée, la valeur de charge rapportée n'est pas utilisée.
- La troisième partie est la valeur de charge mesurée.
- Un rapport de valeur de charge se termine par une ligne comportant le mot `end`.

Variables d'environnement

Le TABLEAU 32 liste et décrit les variables d'environnement associées à `sge_execd`.

TABLEAU 32 Variables d'environnement de `sge_execd`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>sge_execd</code> utilise (dans cet ordre) : <ul style="list-style-type: none"> • Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie. • Le nom de la cellule par défaut ; par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>sge_execd</code> réside. Elle n'est évaluée que si l'option <code>-nostart-commande</code> a été spécifiée sur la ligne de commande <code>sge_execd</code> . Par défaut, l'hôte local est utilisé.

Restrictions

`sge_execd` est en général démarré depuis la racine sur chacune des machines du groupe Sun Grid Engine. S'il est démarré par un utilisateur normal, un répertoire de spool auquel l'utilisateur a accès en lecture/écriture doit être utilisé. Dans ce cas, seuls les travaux soumis par ce même utilisateur sont traités correctement par le système.

Fichiers

- Configuration globale de Sun Grid Engine –
`<racine_sge>/<cellule>/common/configuration`
- Configuration spécifique de l'hôte Sun Grid Engine –
`<racine_sge>/<cellule>/common/local_conf/<hôte>`
- Répertoire de spool des hôtes d'exécution par défaut –
`<racine_sge>/<cellule>/spool/<hôte>`
- Fichier hôte maître de Sun Grid Engine –
`<racine_sge>/<cellule>/common/act_qmaster`

Voir aussi

`sge_intro(1)`, `sge_conf(5)`, `complex(5)`, `sge_commd(8)`, `sge_qmaster(8)`.

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

`sge_qmaster(8)`

Nom

`sge_qmaster` – Démon de contrôle maître de Sun Grid Engine.

Synopsis

```
sge_qmaster [ -help ] [ -lj fichier_journal ] [ -nohist ]  
[ -nostart-commande ] [ -s ]  
  
sge_qmaster -show-license
```

Description

`sge_qmaster` contrôle le comportement général de Sun Grid Engine dans une grappe. Dans le cadre de la programmation des travaux `sge_qmaster` coopère avec `sge_schedd(8)`. Au démarrage de `sge_qmaster` `sge_commd(8)` est en général automatiquement activé sur la même machine (s'il n'est pas déjà présent).

Options

Le TABLEAU 33 liste et décrit les options associées à `sge_qmaster`.

TABLEAU 33 Options de `sge_qmaster`

Option	Description
<code>-help</code>	Imprime la liste de toutes les options.
<code>-lj <i>fichier_journal</i></code>	Active la consignation des travaux. Toutes les actions entreprises par <code>sge_qmaster</code> de la soumission à la sortie du travail sont consignées dans le <i>fichier_journal</i> . Cette fonctionnalité est également disponible avec le démon <code>sge_execd(8)</code> .
<code>-nohist</code>	Pendant le fonctionnement normal, <code>sge_qmaster</code> vide un historique des changements de configuration des files d'attente, complexes et hôtes dans une base de données d'historique. Cette base de données est principalement utilisée avec la commande <code>qacct(1)</code> pour autoriser des requêtes de ressources <code>-l</code> de type <code>qsub(1)</code> dans la ligne de commande <code>qacct(1)</code> . Ce modificateur supprime l'écriture dans cette base de données.
<code>-nostart-<i>commande</i></code>	Ne démarre pas <code>sge_commd(8)</code> automatiquement avec <code>sge_qmaster</code> .
<code>-s</code>	Active le mode silencieux de <code>sge_qmaster</code> . En général, <code>sge_qmaster</code> affiche des informations de licence au démarrage et attend un retour pour continuer. Avec le modificateur <code>-s</code> , <code>sge_qmaster</code> démarre en silence.

Variables d'environnement

Le TABLEAU 34 liste et décrit les variables d'environnement associées à `sge_qmaster`.

TABLEAU 34 Variables d'environnement de `sge_qmaster`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>sge_qmaster</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.

Restrictions

`sge_qmaster` est en général démarré depuis la racine sur les machines maître ou maîtres de réserve de la grappe (consultez le *Guide d'installation et d'administration de Sun Grid Engine* pour plus d'informations sur la configuration des hôtes maîtres de réserve). S'il est démarré par un utilisateur normal, un répertoire spool maître auquel l'utilisateur peut accéder en lecture/écriture doit être utilisé. Dans ce cas, seuls les travaux soumis par cet utilisateur sont traités correctement par le système.

Fichiers

- Configuration globale de Sun Grid Engine - `<racine_sge>/<cellule>/common/configuration`
- Configuration spécifique de l'hôte Sun Grid Engine - `<racine_sge>/<cellule>/common/local_conf/<hôte>`
- Base de données d'historique - `<racine_sge>/<cellule>/common/history`
- Fichier d'arguments `sge_qmaster` - `<racine_sge>/<cellule>/common/qmaster_args`
- Répertoire spool maître par défaut - `<racine_sge>/<cellule>/spool`

Voir aussi

`sge_intro(1)`, `sge_conf(5)`, `sge_commd(8)`, `sge_execd(8)`, `sge_schedd(8)`, `sge_shadowd(8)`, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3* et *Guide d'administration et d'utilisation de Sun Grid Engine Enterprise Edition 5.3*

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sge_schedd(8)

Nom

`sge_schedd` – Agent de programmation de travaux de Sun Grid Engine

Synopsis

```
sge_schedd [ -help ]
```

Description

`sge_schedd` calcule la décision de programmation dans une grappe Sun Grid Engine. Les informations nécessaires pour la décision sont extraites de `sge_qmaster(8)` via une interface d'événements. Après l'application de l'algorithme de programmation, `sge_schedd` communique la décision de programmation à `sge_qmaster(8)` toujours via le GDI de Sun Grid Engine. Pour déclencher un cycle `sge_schedd`, `sge_qmaster(8)` échantillonne les changements de statut dans la grappe et notifie `sge_schedd` à intervalles réguliers.

Avec `sge_schedd`, un `sge_commd(8)` est automatiquement activé sur la même machine (s'il n'est pas déjà présent).

En utilisant l'option `-tsm` de la commande `qconf(1)`, `sge_schedd` peut être forcé à imprimer des messages de suivi lors du prochain cycle d'exécution programmé dans le fichier `<racine_sge>/<cellule>/common/schedd_runlog`. Ces messages indiqueront les raisons pour lesquelles certains travaux ou files d'attente ne rentrent pas dans cette exécution.

Option

Le TABLEAU 35 indique l'option associée à `sge_schedd`.

TABLEAU 35 Option de `sge_schedd`

Option	Description
<code>-help</code>	Imprime le numéro de version du programmeur.

Variables d'environnement

Le TABLEAU 36 liste les variables d'environnement associées à `sge_schedd`.

TABLEAU 36 Variables d'environnement de `sge_schedd`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>sge_schedd</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.

Fichiers

- Répertoire de spool `sge_schedd` -
`<racine_sge>/<cellule>/spool/qmaster/schedd`
- Informations de suivi `sge_schedd` -
`<racine_sge>/<cellule>/common/sched_runlog`
- Configuration `sge_schedd` -
`<racine_sge>/<cellule>/common/sched_configuration`
- Voir `sched_conf(5)` pour tout détail sur le fichier de configuration du programmeur.

Voir aussi

`sge_intro(1)`, `sge_gdi(3)`, `sched_conf(5)`, `sge_commd(8)`, `sge_qmaster(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sge_shadowd(8)

Nom

`sge_shadowd` – Démon maître double de Sun Grid Engine.

Synopsis

`sge_shadowd`

Description

`sge_shadowd` est un processus « lightweight » (léger) qui peut être exécuté sur les hôtes maîtres dits en double dans une grappe Sun Grid Engine pour détecter toute défaillance du démon maître Sun Grid Engine courant `sge_qmaster(8)` et démarrer un nouveau `sge_qmaster(8)` sur l'hôte sur lequel le `sge_shadowd` tourne. Si plusieurs démons en double sont actifs dans une grappe, ils exécutent un protocole qui assure que seul l'un d'entre eux démarre un nouveau démon maître.

Les hôtes adaptés à une utilisation en tant qu'hôte maître en double doivent avoir un accès en lecture écriture de super-utilisateur partagé au répertoire `<racine_sge>/<cellule>/common` ainsi qu'au répertoire de spool du démon maître (par défaut, `<racine_sge>/<cellule>/spool/qmaster`). Les hôtes maîtres en double doivent être contenus dans le fichier `<racine_sge>/<cellule>/common/shadow_masters`.

Restrictions

`sge_shadowd` peut uniquement être démarré depuis la racine.

Variables d'environnement

Le TABLEAU 37 liste et décrit les variables d'environnement associées à `sge_shadowd`.

TABLEAU 37 Variables d'environnement de `sge_shadowd`

Nom de la variable	Description
<code>SGE_ROOT</code>	Spécifie l'emplacement des fichiers de configuration standard de Sun Grid Engine.
<code>SGE_CELL</code>	Si elle est définie, spécifie la cellule par défaut de Sun Grid Engine. Pour indiquer une cellule de Sun Grid Engine, <code>sge_shadowd</code> utilise (dans cet ordre) : <ul style="list-style-type: none">• Le nom de la cellule spécifiée dans la variable d'environnement <code>SGE_CELL</code>, si elle est définie.• Le nom de la cellule par défaut, par ex. <code>default</code>.
<code>SGE_DEBUG_LEVEL</code>	Si elle est définie, spécifie que les informations de débogage doivent être écrites dans <code>stderr</code> . En sus, cette variable définit le niveau de détail des informations de débogage générées.
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sge_commd(8)</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd(8)</code> particulier à utiliser pour la communication Sun Grid Engine du client <code>sge_shadowd</code> réside. Par défaut, l'hôte local est utilisé.

Fichiers

- Répertoire de configuration par défaut – `<racine_sge>/<cellule>/common`
- Fichier des noms des hôtes maîtres en double – `<racine_sge>/<cellule>/common/shadow_masters`
- Répertoire de spool du démon maître par défaut – `<racine_sge>/<cellule>/spool/qmaster`

Voir aussi

`sge_intro(1)`, `sge_conf(5)`, `sge_commd(8)`, `sge_qmaster(8)`,
Guide d'installation et d'administration de Sun Grid Engine, *Guide d'administration et d'utilisation de Sun Grid Engine 5.3*.

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

sgc_sheperd(8)

Nom

`sgc_shepherd` – Agent de contrôle mono-travail de Sun Grid Engine.

Synopsis

`sgc_shepherd`

Description

`sgc_shepherd` fournit la fonctionnalité de processus père pour un unique travail Sun Grid Engine. Cette fonctionnalité est nécessaire sur les systèmes UNIX pour récupérer les informations sur l'utilisation des ressources (voir `getrusage(2)`) à la fin d'un travail. En sus, `sgc_shepherd` transmet au travail les signaux tels que les signaux d'interruption, activation et conclusion ou encore le signal de point de reprise de Sun Grid Engine (voir `sgc_ckpt(1)` pour plus de détails).

`sgc_shepherd` reçoit les informations sur le travail à commencer de `sgc_execd(8)`. Pendant l'exécution du travail, il démarre en fait jusqu'à cinq processus fils. Premièrement, un script *prolog* si cette fonctionnalité est activée par le paramètre *prolog* dans la configuration de grappe (voir `sgc_conf(5)`). Puis un script de démarrage d'environnement parallèle si le travail est un travail parallèle (voir `sgc_pe(5)` pour plus d'informations). Ensuite le travail proprement dit, suivi d'une procédure d'arrêt pour les travaux parallèles et, enfin, un script *epilog* si demandé par le paramètre *epilog* dans la configuration de la grappe. Les scripts *prolog* et *epilog*, ainsi que les procédures de démarrage et d'arrêt de l'environnement parallèle, doivent être fournies par l'administrateur Sun Grid Engine et sont relatifs à des actions spécifiques du site à entreprendre avant et après l'exécution du travail d'utilisateur proprement dit.

Une fois le travail terminé et le script *epilog* traité, `sgc_shepherd` extrait les statistiques d'utilisation des ressources sur le travail, les met dans un sous-répertoire spécifique du travail du répertoire spool de `sgc_execd(8)` pour les rapports par établis via `sgc_execd(8)` et se termine.

Restrictions

`sgc_shepherd` ne doit pas être appelé manuellement mais uniquement par `sgc_execd(8)`.

Fichiers

Répertoire spécifique du travail – `<spool_execd>/rép_travail/<id_travail>`

Voir aussi

`sgc_intro(1)`, `sgc_conf(5)`, `sgc_execd(8)`

Droits d’auteur

Voir `sgc_intro(1)` pour l’énoncé complet des droits et autorisations.

sgccommdctl(8)

Nom

`sgccommdctl` – Commande de contrôle de l’agent de communication de Sun Grid Engine.

Synopsis

```
sgccommdctl [ -d | -k | -t niveau ]  
[ -gid nomprocomm ] [ -h[elp] ] [ -p port ] [ -S ]  
[ -unreg nomprocomm id ]
```

Description

`sgccommdctl` peut être utilisé pour contrôler le comportement de `sgc_commd(8)` ou récupérer des informations d’un `sgc_commd(8)` en cours d’exécution.

Options

Le TABLEAU 38 liste et décrit les options associées à `sgecommdctl`.

TABLEAU 38 Options de `sgecommdctl`

Option	Description
-d	Vide les structures internes du processus <code>sge_commd(8)</code> en cours d'exécution dans <code>/tmp/commd/commd.dump</code> . Le répertoire <code>/tmp/commd</code> doit exister et <code>sge_commd(8)</code> doit y avoir accès en écriture. Sinon, la requête est ignorée. L'emploi de cette option est en général réservé au débogage. La fonctionnalité du processus <code>sge_commd(8)</code> concerné n'est pas affectée.
-k	Élimine le <code>sge_commd(8)</code> concerné. Les communications en cours au moment d'une requête d'élimination seront immédiatement supprimées, l'arrêt d'un <code>sge_commd(8)</code> ne laissera pas les processus connectés au processus en cours d'abandon dans un état incohérent.
-t <i>niveau</i>	<code>sgecommdctl</code> établit une connexion c <code>sge_commd(8)</code> et affiche une sortie de suivi continue correspondant au niveau de suivi spécifié par <i>niveau</i> . La sortie est un sous-ensemble de la sortie de suivi affichée si <code>sge_commd(8)</code> est appelé avec l'option <code>-ll</code> . Actuellement, le seul niveau de suivi pris en charge est 255.
-gid <i>nomprocomm</i>	Récupère le numéro d'identification de processus de communication de <i>nomprocomm</i> . Les composants de Sun Grid Engine qui s'en remettent à <code>sge_commd(8)</code> pour pouvoir communiquer avec d'autres processus de Sun Grid Engine sont enregistrés par <code>sge_commd(8)</code> avec un identificateur unique composé d'un nom et d'un numéro d'identification. Le nom d'identification est identique au nom du composant de Sun Grid Engine (par ex. <code>sge_qmaster</code> pour <code>sge_qmaster(8)</code>). Le numéro d'identification peut être récupéré par l'option <code>-gid</code> .
-help	Imprime la liste de toutes les options.

TABLEAU 38 Options de `sgecommd(8)` (suite)

Option	Description
-p <i>portcde</i>	Numéro de port à utiliser pour adresser <code>sge_commd(8)</code> .
-S	Force l'utilisation de ports sécurisés pour la communication entre <code>sge_commds</code> et entre d'autres composants Sun Grid Engine et les commandes <code>sge_commds</code> . Pour ce, tous les démons de Sun Grid Engine doivent être démarrés avec une permission de super-utilisateur et les programmes clients configurés avec <code>set-uid root</code> . En contrepartie, assure l'interdiction de toute communication non-autorisée pour les comptes autres que celui du super-utilisateur.
-unreg <i>nomprocomm id</i>	<p>Désenregistre le composant de Sun Grid Engine <i>nomprocomm</i> enregistré avec l'ID <i>id</i> sur <code>sge_commd(8)</code> (voir l'option <code>-gid</code> ci-dessus pour la description de <i>nomprocomm</i> et <i>id</i>).</p> <p>Désenregistrer un composant de Sun Grid Engine de <code>sge_commd(8)</code> peut devenir nécessaire si un démon de Sun Grid Engine tel que <code>sge_qmaster(8)</code>, <code>sge_execd(8)</code> ou <code>sge_schedd(8)</code> est abandonné d'une façon non-contrôlée (par ex. en envoyant le signal <code>SIGKILL</code> via <code>kill(1)</code>) et que <code>sge_commd(8)</code> refuse de redémarrer le composant avec le message <code>error enrolling to commd: COMMPROC ALREADY REGISTERED</code>.</p> <p>L'utilitaire d'enregistrement de <code>sge_commd(8)</code> est utilisé pour éviter que des démons Sun Grid Engine redondants ne s'exécutent sur le même hôte. Si un composant de Sun Grid Engine est interrompu mais qu'il est impossible de le désenregistrer de <code>sge_commd(8)</code> l'enregistrement reste valide jusqu'à l'arrivée à échéance d'un délai de plusieurs minutes ou jusqu'à ce que le processus de communication soit désenregistré manuellement.</p>

Variables d'environnement

Le TABLEAU 39 liste et décrit les variables d'environnement associées à `sgecommdcnt1`.

TABLEAU 39 Variables d'environnement de `sgecommdcnt1`

Nom de la variable	Description
<code>COMMD_PORT</code>	Si elle est définie, spécifie le port tcp sur lequel <code>sgecommdcnt1</code> devrait écouter les requêtes de communication. La plupart des installations utiliseront une entrée de table de mappage de services au lieu de définir ce port.
<code>COMMD_HOST</code>	Si elle est définie, spécifie l'hôte sur lequel le <code>sge_commd</code> particulier à utiliser pour la communication Sun Grid Engine de <code>sgecommdcnt1</code> réside. Par défaut, l'hôte local est utilisé.

Voir aussi

`sge_intro(1)`, `sge_commd(8)`, `sge_execd(8)`, `sge_qmaster(8)`,
`sge_schedd(8)`

Droits d'auteur

Voir `sge_intro(1)` pour l'énoncé complet des droits et autorisations.

