



Principes d'administration de Sun™ Grid Engine, Enterprise Edition 5.3

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

Référence n° 816-7480-10
Juillet 2002, révision 01

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

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

Sun Microsystems, Inc. possède les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans limitation aucune, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains répertoriés sur <http://www.sun.com/patents> et un ou plusieurs brevets supplémentaires ou demandes de brevet en cours aux États-Unis et dans d'autres pays.

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, the Sun logo, AnswerBook2, docs.sun.com et Solaris sont des marques de fabrique ou des marques déposées 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 pionniers 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

À propos des grilles de calcul et de Sun Grid Engine, Enterprise Edition 5.3	2
Rôle de la gestion de ressources distribuées	3
Politiques, hôtes et démons	4
Instructions de ligne de commande pour les tâches administratives courantes	6
▼ Ajout ou suppression de privilèges à/d'un hôte	6
▼ Ajout d'un hôte d'exécution	6
▼ Suppression d'un hôte d'exécution	7
▼ Ajout ou suppression d'un hôte de soumission	7
▼ Affichage des noms des hôtes courants	7
▼ Administration des files d'attente	8
▼ Changement de l'hôte maître	8
▼ Configuration d'un maître double	9
Utilisation de scripts et de fichiers pour les tâches administratives	10
▼ Ajout ou modification d'objets en utilisant des fichiers	10
▼ Modification de files d'attente, hôtes et environnements en utilisant des fichiers	11
▼ Modification d'une configuration globale ou du programmeur en utilisant des fichiers	15
▼ Paramétrage de l'environnement Sun Grid	17

Politiques 19

Configurations de politiques fonctionnelles 19

- ▼ Création d'une programmation fonctionnelle basées sur les utilisateurs 20
- ▼ Création d'une programmation fonctionnelle basée sur des projets 20
- ▼ Création d'une programmation fonctionnelle basée sur les services 21
- ▼ Création d'une programmation d'arborescence de partages basée sur les projets avec FCFS au sein de chaque projet 22
- ▼ Création d'une programmation d'arborescence des partages basée sur des partages avec des partages égaux pour chaque utilisateur 22
- ▼ Création d'une programmation d'arborescence des partages basée sur des projets avec des partages individuels par utilisateur au sein de chaque projet 23

Dépannage - problèmes courants 25

Diagnostic des problèmes 31

Travaux en attente non-acheminés 31

Rapports signalant un état d'erreur E au niveau d'un travail et d'une file 32

Principes d'administration de Sun Grid Engine, Enterprise Edition 5.3

Ce document est fourni à titre de référence rapide afin d'aider les administrateurs de systèmes Sun™ Grid Engine, Enterprise Edition 5.3. En plus des instructions relatives aux tâches courantes impliquées dans l'administration, vous y trouverez une brève présentation des grilles de calcul en général et, plus particulièrement, du produit Sun Grid Engine, Enterprise Edition 5.3. Ce document se termine par des conseils de réglage et de dépannage.

Utilisez ce manuel en supplément — et non pas à la place — du *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*, dans lequel vous trouverez des explications détaillées des concepts et des procédures de Sun Grid Engine, Enterprise Edition 5.3.

Ce document contient des instructions permettant d'accomplir les tâches suivantes :

- « Ajout ou suppression de privilèges à/d'un hôte », page 6 ;
- « Ajout d'un hôte d'exécution », page 6 ;
- « Suppression d'un hôte d'exécution », page 7 ;
- « Ajout ou suppression d'un hôte de soumission », page 7 ;
- « Affichage des noms des hôtes courants », page 7 ;
- « Administration des files d'attente », page 8 ;
- « Changement de l'hôte maître », page 8 ;
- « Configuration d'un maître double », page 9 ;
- « Ajout ou modification d'objets en utilisant des fichiers », page 10 ;
- « Modification de files d'attente, hôtes et environnements en utilisant des fichiers », page 11 ;
- « Modification d'une configuration globale ou du programmeur en utilisant des fichiers », page 15 ;
- « Paramétrage de l'environnement Sun Grid », page 17 ;
- « Création d'une programmation fonctionnelle basées sur les utilisateurs », page 20 ;
- « Création d'une programmation fonctionnelle basée sur des projets », page 20 ;
- « Création d'une programmation fonctionnelle basée sur les services », page 21 ;
- « Création d'une programmation d'arborescence de partages basée sur les projets avec FCFS au sein de chaque projet », page 22 ;

- « Création d'une programmation d'arborescence des partages basée sur des partages avec des partages égaux pour chaque utilisateur », page 22 ;
- « Création d'une programmation d'arborescence des partages basée sur des projets avec des partages individuels par utilisateur au sein de chaque projet », page 23 ;
- « Dépannage - problèmes courants », page 25 ;
- « Diagnostic des problèmes », page 31.

Remarque – Une bonne partie du contenu de ce document figurait à l'origine dans la section « How-To » du site Web du projet Sun Grid Engine. Ce site Web bénéficiant de mises à jour fréquentes, il est vivement recommandé aux administrateurs système Sun Grid Engine, Enterprise Edition 5.3 de le consulter régulièrement. Son URL est le suivant : <http://gridengine.sunsource.net/project/gridengine/howto/howto.html>.

À propos des grilles de calcul et de Sun Grid Engine, Enterprise Edition 5.3

Si vous découvrez l'administration d'une *grille* de calcul, vous appartenez à la grande majorité des administrateurs d'ordinateurs professionnels. Rares sont en effet les personnes qui peuvent se vanter d'être des administrateurs de grilles de calcul expérimentés car la technologie logicielle qui se cache derrière ces grilles a moins de dix ans. Selon le Professeur Ian Foster, l'un des pionniers des grilles de calcul, les concepts du « grid computing » (les grilles de calcul) n'ont été « explorés » qu'en 1995. Du coup, même les premiers adeptes de ces grilles ont moins de dix ans d'expérience.

Toutefois, quelque minime que soit votre expérience en matière de grilles de calcul, vous avez une grande expérience de leur utilisation ou, pour le moins, de celle d'un autre type de grille. À chaque fois que vous appuyez sur un interrupteur en effet, vous utilisez les ressources d'un service public, en anglais « grid » (grille). Dans la plupart des pays industrialisés, l'électricité est fournie aux particuliers et aux entreprises via un service public qui se compose de plusieurs producteurs d'électricité indépendants dont la production est rassemblée dans un groupement de ressources communes, le service public ou grille.

Le regroupement des ressources présente des avantages pour, à la fois, les consommateurs et les producteurs. Un fournisseur d'électricité est en mesure de faire tourner sa centrale à pleine capacité même si les consommateurs qui se trouvent à proximité ne requièrent pas autant d'électricité que la centrale n'est à même d'en produire. Les consommateurs qui habitent dans une zone rurale très éloignée d'une ville, par exemple, n'ont en général pas besoin de toute l'électricité pouvant être produite par une centrale électrique ordinaire. Le recours à un service public permet à la centrale de vendre sa capacité en « excès » dans ce réseau public et de desservir ainsi des consommateurs qui vivent dans des zones éloignées dont les centrales locales peuvent ne pas être en mesure d'assurer l'approvisionnement.

La capacité de mettre des ressources sous-utilisées en production et de fournir ces ressources pour répondre à la demande est la caractéristique définitionnelle du service public de fourniture d'électricité. En adoptant le mot « grid » qui est celui utilisé en anglais pour désigner le réseau de fourniture d'électricité (le « power grid »), les pionniers des grilles de calcul ont créé une analogie on ne peut plus appropriée. L'objectif fondamental d'une grille de calcul est de regrouper les ressources sous-utilisées de centaines, voire de milliers, d'ordinateurs inactifs ou pratiquement inactifs et de mettre la puissance de traitement combinée obtenue à la disposition des gros utilisateurs.

Rôle de la gestion de ressources distribuées

Cette analogie est toutefois moins évidente au niveau du détail de la fourniture. Le service public de fourniture d'électricité gère en effet les ressources du regroupement en câblant littéralement tous les membres producteurs de la grille et, en dernier, les habitations, usines, fermes et bureaux des consommateurs d'électricité. Une fois connectée et introduite dans le service, l'électricité circule en fonction de la demande. Si vous appuyez sur un interrupteur, l'électricité provenant d'une centrale électrique quelconque arrive, via le réseau, jusqu'à l'ampoule. Si l'on peut certes dire que les ressources informatiques qui composent une grille de calcul sont « connectées » les unes aux autres, la différence est qu'elles le sont au moyen de technologies d'interconnexion telles que l'Internet ou des intranets. Il ne suffit toutefois pas d'appuyer sur l'interrupteur d'un ordinateur faisant partie d'un réseau pour pouvoir utiliser les ressources d'une (ou contribuer aux ressources d'une) grille de calcul.

Vous ne pouvez accéder à une grille de calcul qu'après avoir installé un logiciel *middleware* spécial. Le « middleware », ou interlogiciel, est un logiciel qui se trouve entre le système d'exploitation de l'ordinateur, par exemple l'environnement d'exploitation Solaris™ de Sun ou le système d'exploitation Linux, et les applications, par exemple un programme de rendu graphique 3D. Le logiciel Sun Grid Engine est le middleware qui permet de participer à des grilles de calcul.

Résultat tangible d'un projet de grille de calcul « gratuite », Sun Grid Engine est un logiciel de *gestion de ressources distribuées*, qui fait office d'intermédiaire entre plusieurs utilisateurs et plusieurs ressources informatiques — souvent éloignés sur le plan géographique — pour fournir aux utilisateurs une puissance de calcul bien supérieure pour leurs travaux complexes. Grâce au logiciel Sun Grid Engine, des ressources informatiques inactives incorporées à la grille sont mises à la disposition d'utilisateurs distants, ce qui augmente considérablement la productivité de ces utilisateurs et les ressources. Par exemple, on a constaté que des ressources informatiques qui étaient utilisées en général à un quart de leur capacité réelle tournaient à pratiquement 100 pour cent en participant à une grille.

Les périphériques de calcul constituent la partie « ressources distribuées » du terme « gestion de ressources distribuées » tandis que le logiciel Sun Grid Engine correspond à la partie clé du terme, à savoir « gestion ». Sans cette gestion, ce serait le chaos. En tant que gestionnaire de la grille, le logiciel Sun Grid Engine accepte les travaux soumis par les utilisateurs et en programme l'exécution sur les systèmes appropriés de la grille en se basant sur des *politiques*, qui sont fixées par le personnel dirigeant et technique de l'organisation.

Il n'y a pas deux grilles identiques ; pas de taille unique pour tous les cas de figure. Il existe trois catégories de grilles, qui vont des systèmes individuels aux grosses fermes de calcul de type super-calculateur qui utilisent des milliers de processeurs :

- Les *grilles de grappe*, activées par le logiciel Sun Grid Engine 5.3 de base, sont les plus simples car elles sont composées d'*ordinateurs hôtes* qui travaillent de concert pour offrir un unique point d'accès aux utilisateurs dans le cadre d'un simple projet ou service.

Le logiciel Sun Grid Engine, Enterprise Edition 5.3 étend le modèle de base pour créer les deux autres catégories de grilles, plus complexes et plus puissantes.

- Les *grilles de campus* permettent de partager des ressources informatiques entre plusieurs projets ou services d'une organisation. Les organisations peuvent utiliser des grilles de campus pour traiter une foule de tâches variées, des processus commerciaux cycliques au rendu, en passant par l'exploration de données, etc.
- Les *grilles globales* sont formées par un groupe de grilles de campus qui composent de très grands systèmes virtuels qui vont au-delà des frontières de l'organisation. Les utilisateurs ont accès à une puissance de calcul qui dépasse largement les ressources disponibles au sein de leur propre organisation.

Le logiciel Sun Grid Engine, Enterprise Edition 5.3 fournit la puissance et la souplesse requises pour les grilles de campus. Ce produit est très utile pour les grilles de grappe existantes basées sur le logiciel Sun Grid Engine de base car il facilite la transition pour créer une grille de campus en rassemblant toutes les grilles de grappe Sun Grid Engine existantes sur un campus. Ce produit constitue également une excellente base pour un campus d'entreprise qui adopterait pour la première fois une grille de calcul.

Politiques, hôtes et démons

Le logiciel Sun Grid Engine, Enterprise Edition 5.3 orchestre la répartition de la puissance informatique en fonction de politiques qui régissent les ressources de l'entreprise que vous, administrateur système, administrez. Le système Sun Grid Engine, Enterprise Edition 5.3 se sert de ces politiques pour déterminer les ressources de calcul disponibles au sein de la grille de campus, rassemble ces ressources, puis les alloue et les distribue automatiquement de façon à optimiser l'utilisation à travers toute la grille de campus.

Dans le but de permettre la coopération au sein d'une grille de campus, les propriétaires de projets qui utilisent la grille doivent négocier des politiques et disposer d'une certaine souplesse autorisant les remplacements manuels dans le cadre d'un projet particulier, les politiques quant à elles doivent être surveillées et mises en vigueur automatiquement.

En tant qu'administrateur, vous pouvez définir des politiques d'utilisation de haut niveau, personnalisées en fonction d'éléments appropriés à votre suite. Il existe quatre types de politiques, qui sont décrits en détail dans le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3* :

- fonctionnelle,
- basée sur le partage,

- initiation limite,
- remplacement.

La gestion des politiques Sun Grid Engine, Enterprise Edition 5.3 contrôle automatiquement l'utilisation des ressources distribuées dans la grappe pour remplir vos objectifs. Les travaux haute priorité sont acheminés de manière préférentielle et reçoivent davantage de droits UC en cas de compétition avec d'autres travaux moins prioritaires. Le logiciel Sun Grid Engine, Enterprise Edition 5.3 surveille le déroulement de tous les travaux et en ajuste les priorités conformément aux objectifs définis dans les politiques.

Le module des politiques de Sun Grid Engine, Enterprise Edition 5.3 gère les règles d'utilisation de votre grille selon si elles sont lâches ou strictes. L'allocation de ressources basée sur des politiques octroie à chaque utilisateur, équipe et service, ainsi qu'à tous les projets un partage des ressources système pendant une période dite d'accumulation qui peut être d'une semaine, un mois ou un trimestre.

Quatre types d'hôtes sont à la base du système Sun Grid Engine, Enterprise Edition 5.3 :

- hôte maître,
- hôte d'exécution,
- hôte d'administration,
- hôte de soumission.

En sus, il est conseillé que vous-même en tant qu'administrateur, créez un *hôte maître double*. S'il n'y a qu'un hôte maître, d'autres machines de la grappe peuvent être choisies en tant qu'hôtes maîtres doubles afin d'augmenter la disponibilité. Un maître double surveille de façon continue l'hôte maître et prend automatiquement et en toute transparence le contrôle en cas de panne de ce dernier. Les travaux présents dans la grappe ne sont donc pas affectés par une panne de l'hôte maître.

Ces différents types d'hôtes sont décrits en détail dans le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*.

Quatre démons sont à la base du fonctionnement de Sun Grid Engine, Enterprise Edition 5.3 :

- `sge_qmaster` est le démon maître.
- `sge_schedd` est le démon programmeur.
- `sge_execd` est le démon d'exécution.
- `sge_commd` est le démon de communication.

Chacun de ces démons est examiné en détail dans le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*.

Le reste de ce document contient des instructions permettant d'accomplir les tâches administratives Sun Grid Engine, Enterprise Edition 5.3, les plus courantes. Pour davantage d'informations d'ordre général et les instructions permettant d'accomplir des tâches non-présentées dans ces pages, consultez le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*, le *Manuel de référence de Sun Grid Engine 5.3 et Sun Grid Engine, Enterprise Edition 5.3* ou les pages man.

Instructions de ligne de commande pour les tâches administratives courantes

Remarque – Les groupes d'instructions qui suivent sont basés sur l'utilisation de commandes au niveau de la ligne de commande. Chacune de ces commandes correspond à une fonctionnalité de l'interface homme-machine de Sun Grid Engine, Enterprise Edition 5.3, QMON. Pour savoir comment utiliser QMON pour accomplir les tâches suivantes, consultez le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*.

▼ Ajout ou suppression de privilèges à/d'un hôte

- Pour ajouter des privilèges administratifs, entrez la commande suivante.

```
qconf -ah
```

- Pour supprimer des privilèges administratifs, entrez la commande suivante.

```
qconf -dh
```

▼ Ajout d'un hôte d'exécution

1. Faites du nouvel hôte un hôte administratif en entrant la commande suivante.

```
qconf -ah
```

2. En tant que super-utilisateur sur ce nouvel hôte, exécutez le script suivant depuis \$SGE_ROOT.

```
install_execd
```

▼ Suppression d'un hôte d'exécution

- **Supprimez les files d'attente associées à l'hôte en question en entrant la commande suivante.**

```
qconf -dq
```

- **Supprimez l'hôte en entrant la commande suivante.**

```
qconf -de
```

▼ Ajout ou suppression d'un hôte de soumission

- **Pour faire d'un hôte un hôte de soumission, entrez la commande suivante sur la ligne de commande de cet hôte.**

```
qconf -as
```

- **Pour annuler la désignation d'un hôte en tant qu'hôte de soumission, entrez la commande suivante sur la ligne de commande de cet hôte.**

```
qconf -ds
```

▼ Affichage des noms des hôtes courants

- **Pour afficher les noms des hôtes administratifs, entrez la commande suivante.**

```
qconf -sh
```

- **Pour afficher les noms des hôtes de soumission, entrez la commande suivante.**

```
qconf -ss
```

- **Pour afficher les noms des hôtes d'exécution, entrez la commande suivante.**

```
qconf -sel
```

▼ Administration des files d'attente

- Pour ajouter une file d'attente, entrez la commande suivante.

```
qconf -aq
```

- Pour ajouter une file d'attente à *partir d'un fichier*, entrez la commande suivante.

```
qconf -Aq
```

- Pour supprimer une file d'attente, entrez la commande suivante.

```
qconf -dq
```

- Pour modifier une file d'attente, entrez la commande suivante.

```
qconf -mq
```

- Pour changer des attributs individuels pour plusieurs files d'attente, entrez la commande suivante.

```
qconf -mqattr
```

▼ Changement de l'hôte maître

1. Arrêtez les démons maître et programmeur sur l'hôte maître courant en entrant la commande suivante.

```
qconf -ks -km
```

2. **Modifiez le fichier** `$SGE_ROOT/default/common/act_qmaster` **en vous basant sur ce qui suit.**

a. **Dans le fichier** `act_qmaster`, **remplacez le nom de l'hôte courant par le nom du nouvel hôte maître.**

Ce nom devrait être identique à celui retourné par l'utilitaire `gethostname`.
Pour l'obtenir, entrez la commande suivante sur le nouvel hôte maître.

```
$SGE_ROOT/utilbin/$ARCH/gethostname
```

b. **Remplacez l'ancien nom qui figure dans le fichier** `act_qmaster` **par celui retourné par l'utilitaire** `gethostname`.

3. **Sur le nouvel hôte maître, exécutez le script suivant.**

```
$SGE_ROOT/default/common/sge5
```

Cette opération lancera `sge_qmaster` et `sge_schedd` sur le nouvel hôte maître.

▼ Configuration d'un maître double

1. **Créez un fichier** `shadow_masters` **en fonction des indications suivantes.**

- Créez ce fichier dans `$SGE_ROOT/default/common`.
- Mettez le nom de l'hôte maître principal sur la première ligne, puis listez les hôtes choisis pour reprendre les responsabilités du maître dans l'ordre de votre choix.
Par exemple :

```
system% cat shadow_masters
hôte1
hôte2
hôte3
```

Dans l'exemple ci-dessus, `hôte1` est l'hôte maître principal. Si `hôte1` tombait en panne, `hôte2` deviendrait le serveur maître au bout d'une période d'environ 10 minutes. Si à son tour `hôte2` tombait en panne, `hôte3` assurerait la reprise.

2. **Vérifiez si les permissions sont correctes.**

Tous les hôtes doubles maîtres doivent avoir des permissions d'accès en lecture/écriture au répertoire `pool_qmaster`.

3. Lancez les démons en double en utilisant le script de démarrage `sge5`.
Entrez la commande suivante sur chaque hôte en tant que super-utilisateur.

```
$SGE_ROOT/default/common/sge5 -shadowd
```

Une fois ces étapes effectuées, le dédoublement du maître est actif pour la grappe Sun Grid Engine, Enterprise Edition 5.3.

Utilisation de scripts et de fichiers pour les tâches administratives

Remarque – Vous pouvez utiliser l'interface homme-machine QMON pour effectuer toutes les tâches administratives Sun Grid Engine, Enterprise Edition 5.3, vous aurez ainsi l'avantage supplémentaire de découvrir l'ensemble des fonctionnalités du système. Il est également possible d'administrer une grille Sun Grid Engine, Enterprise Edition 5.3 au moyen de commandes données à l'invite de shell et appelées depuis des scripts de shell. De nombreux administrateurs expérimentés trouvent cette deuxième solution plus souple, plus rapide et plus puissante pour changer les paramètres. Les trois groupes d'instructions suivants relèvent de ces méthodes.

▼ Ajout ou modification d'objets en utilisant des fichiers

- Utilisez la commande `qconf` pour ajouter ou modifier des objets selon des spécifications établies dans un fichier.

La syntaxe de la commande est la suivante : `qconf -{A,M}<objet> <nom_fichier>`

Dans la syntaxe ci-dessus, `-A` signifie ajouter et `-M` modifier. La variable `<objet>` peut être au choix :

- `c` – Complexe
- `ckpt` – Environnement de point de reprise
- `e` – Hôte d'exécution
- `p` – Environnement parallèle
- `q` – File d'attente
- `u` – Jeu d'utilisateurs

Vous pouvez utiliser cette option en combinaison avec l'option `show` de la commande `qconf` `—qconf -s<obj>—` pour prendre un objet existant, le modifier puis mettre à jour l'objet existant ou en créer un nouveau.

Exemple

Ce qui suit est un exemple de script de shell qui modifie la *commande de migration* d'un environnement de point de reprise existant.

```
#!/bin/sh
# ckptmod.sh: modify the migration command
# of a checkpointing environment
# Usage: ckptmod.sh <nom-env-pointreprise> <chemin-complet-de-
la-commande>
TMPFILE=/tmp/ckptmod.$$

CKPT=$1
MIGMETHOD=$2

qconf -sckpt $CKPT | grep -v '^migr_command' > $TMPFILE
echo "migr_command $MIGMETHOD" >> $TMPFILE
qconf -Mckpt $TMPFILE
rm $TMPFILE
```

▼ Modification de files d'attente, hôtes et environnements en utilisant des fichiers

Vous pouvez modifier des files d'attente, des hôtes et des environnements parallèles et de point de reprise depuis la ligne de commande en utilisant la commande `qconf` de deux manières au choix et en combinaison avec d'autres commandes, selon les arguments employés. Les instructions qui suivent couvrent ces deux cas de figure.

- **Si vous avez déjà préparé un fichier, entrez la commande `qconf` et les options appropriées.**

La syntaxe de la commande est la suivante ; faites particulièrement attention à la casse dans les options car la différence entre majuscules et minuscules est importante.

```
qconf -M{q,e,p,ckpt} <nomfichier>
```

Les options sont les suivantes :

- `-M` – Modifier à partir d'un fichier existant, <nomfichier>
- `q` – File d'attente
- `e` – Hôte d'exécution
- `p` – Environnement parallèle
- `ckpt` – Environnement de point de reprise

- Si vous n'avez *pas encore préparé de fichier*, entrez la commande `qconf` et les options appropriées.

La syntaxe de la commande est la suivante ; faites particulièrement attention à la casse dans les options car la différence entre majuscules et minuscules est importante.

```
qconf -m{q,e,p,ckpt} <nomobj>
```

Les options sont les suivantes :

- -m – Modifier en ouvrant un éditeur de texte pour créer un fichier de modification
- q – File d'attente
- e – hôte d'exécution
- p – Environnement parallèle
- ckpt – Environnement de point de reprise

Discussion

La différence entre les commandes — M majuscule dans la première, m minuscule dans la seconde — détermine le résultat de la commande. Les options -M et -m signifient toutes deux *modifier*, mais le M majuscule implique que la modification se fait à partir d'un fichier *existant*, ce qui n'est pas le cas du m minuscule qui ouvre un fichier temporaire dans un éditeur. Lorsque vous enregistrez les changements apportés à ce fichier et fermez l'éditeur, le système reflète immédiatement ces changements.

Si toutefois vous voulez changer plusieurs objets d'un seul coup, ou changer de manière non-interactive la configuration des objets, utilisez l'ensemble de commandes `qconf -...attr`.

Un ensemble de commandes de ce type apporte des modifications sur la base de spécifications consignées *dans un fichier* :

```
qconf -{A,M,R,D}attr queue|exechost|pe|ckpt <nomfichier>
```

Un ensemble de commandes équivalent apporte les modifications sur la base des spécifications *sur la ligne de commande* :

```
qconf -{a,m,r,d}attr queue|exechost|pe|ckpt <attrib> <valeur> <liste_files>|<liste_hôtes>
```

Dans ces deux ensembles de commandes, les options indiquent ce qui suit :

- -A/a – Ajoute un attribut
- -M/m – Modifie un attribut
- -R/r – Remplace un attribut
- -D/d – Supprime un attribut

- *<attrib>* - Attribut d'une file d'attente ou d'un hôte à changer
- *<valeur>* - Valeur de l'attribut concerné
- *<nomfichier>* - Fichier contenant les paires attribut-valeur

Les options *a*, *m* et *d* vous permettent de travailler sur des valeurs individuelles dans une liste de valeurs, tandis que *r* remplace toute la liste des valeurs par la nouvelle spécifiée, que ce soit sur la ligne de commande ou dans le fichier.

Exemples

- **Changez le type de file d'attente de `tcf27-e019.q` en `batch-only` :**

```
% qconf -rattr queue qtype batch tcf27-e019.q
```

- **Modifiez le type de la file d'attente et le comportement de démarrage de shell de `tcf27-e019.q` sur la base du contenu du fichier `new.cfg` :**

```
% cat new.cfg
qtype batch interactive checkpointing
shell_start_mode unix_behavior
% qconf -Rattr queue new.cfg tcf27-e019.q
```

- **Attachez les complexes nommés `storage` et `license` à l'hôte `tcf27-e019` :**

```
% qconf -rattr exechost complex_list storage,license tcf27-e019
```

- **Ajoutez la ressource nommée `scratch1` avec une valeur de `1000M` et `long` avec une valeur de `2` :**

```
% qconf -rattr exechost complex_values scratch1=1000M,long=2 tcf27-e019
```

- **Attachez la ressource nommée `short` à l'hôte avec la valeur `4` :**

```
% qconf -aattr exechost complex_values short=4 tcf27-e019
```

- **Changez la valeur de `scratch1` en `500M` tout en laissant les autres valeurs inchangées :**

```
% qconf -mattr exechost complex_values scratch1=500M tcf27-e019
```

- **Supprimez la ressource** long :

```
% qconf -dattr exechost complex_values long tcf27-e019
```

- **Ajoutez tcf27-b011.q à la liste des files d'attente pour l'environnement de point de reprise sph :**

```
% qconf -aattr ckpt queue_list tcf27-b011.q sph
```

- **Remplacez le nombre d'emplacements de l'environnement parallèle, make, par 50 :**

```
% qconf -mattr pe slots 50 make
```

Ciblage des files d'attente avec la commande `qselect`

La commande `qselect` donne la liste des files d'attente. Utilisée avec des options, elle permet de n'obtenir que les files d'attente qui correspondent à des spécifications données. Cette commande est particulièrement intéressante avec l'ensemble de commandes `qconf` `-...attr`, car elle vous permet de cibler les files que vous voulez modifier.

Exemples

- **Pour lister toutes les files d'attente figurant sur des machines Linux :**

```
% qselect -l arch=glinux
```

- **Pour lister toutes les files d'attente figurant sur des machines dotées de deux UC :**

```
% qselect -l num_proc=2
```

- **Pour lister toutes les files d'attente figurant sur toutes les machines Solaris 64 bits à quatre UC :**

```
% qselect -l arch=solaris64,num_proc=4
```

- **Pour lister les files d'attente qui fournissent une licence d'application (configurée au préalable) :**

```
% qselect -l app_lic=TRUE
```

Vous pouvez combiner `qselect` avec `qconf` pour effectuer des changements de grande envergure au moyen d'une unique ligne de commande. Pour ce faire, mettez l'ensemble de la commande `qselect` entre barres obliques inverses (`/`) et utilisez cet élément à la place de la variable `<liste-files>` sur la ligne de commande `qconf`.

Exemples

- **Définissez `sol_prolog.sh` en tant que script `prolog` sur toutes les files d'attente se trouvant sur des machines Solaris :**

```
% qconf -mattr queue prolog /usr/local/scripts/sol_prolog.sh 'qselect -l arch=solaris'
```

- **Mettez l'attribut, `fluent_license`, sur deux sur toutes les files d'attente se trouvant sur des systèmes à deux processeurs :**

```
% qconf -mattr queue complex_values fluent_license=2 'qselect -l num_proc=2'
```

Utiliser la commande `qconf` conjointement avec la commande `qselect` constitue la solution la plus souple pour automatiser la configuration des files d'attente Sun Grid Engine, Enterprise Edition 5.3, car cela vous permet d'élaborer vos propres scripts d'administration personnalisés.

▼ **Modification d'une configuration globale ou du programmeur en utilisant des fichiers**

- **Pour changer une configuration globale, utilisez la commande `qconf -mconf`, selon les principes indiqués dans les sections suivantes.**
- **Pour changer le programmeur, utilisez la commande `qconf -msconf`, selon les principes indiqués dans la section suivante.**

Discussion

Ces deux commandes ouvrent un fichier temporaire dans un éditeur. Lorsque vous quittez cet éditeur, tous les changements que vous avez enregistrés dans ce fichier temporaire sont traités par le système et deviennent immédiatement effectifs. Le programme d'édition utilisé pour ouvrir le fichier temporaire est celui spécifié par la variable d'environnement `EDITOR`. Si cette variable n'est pas définie, `vi` est utilisé.

Vous pouvez vous servir de la variable d'environnement `EDITOR` pour automatiser le comportement des commandes `qconf -m. . .`. Changez la valeur de cette variable de façon à pointer sur un programme qui modifie le fichier dont le nom est donné par le premier argument. Une fois que ce programme aura modifié le fichier temporaire et se sera fermé, le système lira les modifications et procédera immédiatement aux mises à jour adéquates.

Remarque – Si l'heure de modification du fichier reste la même après l'opération d'édition, le système risque d'assumer par erreur qu'il n'a pas été modifié. Pour éviter ce problème, vous devez insérer une instruction `sleep 1` avant d'écrire le fichier, pour garantir que l'heure de modification soit différente.

Exemple

Le script de l'exemple suivant modifie l'intervalle de programmation du programmeur.

```
#!/bin/ksh
# sched_int.sh: modify the schedule interval
# usage: sched_int.sh <n>, where <n> is
# the new interval, in seconds. n < 60

TMPFILE=/tmp/sched_int.$$
if [ $MOD_SGE_SCHED_INT ]; then
    grep -v schedule_interval $1 > $TMPFILE
    echo "schedule_interval 0:0:$MOD_SGE_SCHED_INT" >> $TMPFILE
# sleep to ensure modification time changes
    sleep 1
    mv $TMPFILE $1
else
    export EDITOR=$0
    export MOD_SGE_SCHED_INT=$1
    qconf -msconf
fi
```

Le script ci-dessus modifie la variable d'environnement `EDITOR` pour qu'elle pointe sur ce même script, puis appelle la commande `qconf -msconf`. Cette seconde invocation imbriquée du script modifie le fichier temporaire spécifié par le premier argument, puis sort. Le système Sun Grid Engine, Enterprise Edition 5.3 lit ensuite automatiquement les changements et la première invocation du script se termine. La technique ci-dessus peut être utilisée en conjonction avec toute commande `qconf -m...`. Elle est toutefois spécialement utile pour l'administration du programmeur et de la configuration globale, car il n'existe pas d'autres moyens de les automatiser.

▼ Paramétrage de l'environnement Sun Grid

Sun Grid Engine, Enterprise Edition 5.3 est un outil de gestion de ressources distribuées ou DRM (*Distributed Resource Management*) complet et universel. Le composant programmeur du système prend en charge un vaste éventail de scénarios de fermes de calcul. Pour obtenir une performance maximale de votre environnement de calcul, il peut être intéressant de faire le point sur les fonctionnalités qui sont installées et celles qui sont vraiment nécessaires pour résoudre votre problème de gestion de charge. Désactiver certaines de ces fonctionnalités peut être avantageux sur le plan de la performance de débit de votre grappe.

Surveillance du programmeur

La surveillance du programmeur peut vous aider à trouver la raison du non-acheminement de certains travaux. Sachez toutefois que fournir ces informations pour tous les travaux à tout moment peut consommer beaucoup de ressources et n'est en général pas nécessaire.

- **Pour désactiver la surveillance du programmeur, mettez `schedd_job_info` sur `false` dans la configuration du programmeur, `sched_conf(5)`.**

Travaux terminés

Dans le cas de travaux de matrice, la liste des travaux terminés dans `qmaster` peut devenir particulièrement longue. La désactiver permettra d'économiser de la mémoire et d'accélérer le processus `qstat`, car le processus `qstat` extrait également la liste des travaux terminés.

- **Pour désactiver la fonction qui liste les travaux terminés, mettez `finished_jobs` sur `0` dans la configuration globale `sgc_conf(5)`.**

Vérification des travaux

Imposer la validation au moment de la soumission des travaux peut être une procédure valable pour empêcher que des travaux non-acheminables restent éternellement en attente. Sachez toutefois que valider les travaux peut aussi se révéler particulièrement long, spécialement dans les environnements hétérogènes comportant tout un éventail de noeuds d'exécution différents et de ressources consommables, et où chaque utilisateur a son propre profil de travaux. Dans les environnements homogènes ne présentant qu'une paire de travaux différents, la validation générale des travaux peut être omise.

- **Pour désactiver la vérification des travaux, ajoutez l'option `qsub(1), -wn`, aux requêtes par défaut à l'échelle de la grappe (voir `sge_request(5)`).**

Seuils de charge et seuils d'interruption

Les seuils de charge sont nécessaires si vous sursouscrivez délibérément vos machines et avez besoin d'un mécanisme qui empêche toute charge excessive du système. Les seuils d'interruption sont également utilisés à cet effet, mais sont également utiles dans le cas suivant : le noeud d'exécution reste ouvert pour une charge interactive qui n'est pas sous le contrôle du système Sun Grid Engine, Enterprise Edition 5.3 et vous voulez empêcher toute surcharge du noeud.

Si une ferme de calcul est plus monovalente — par exemple, si chaque UC de noeud de calcul est représentée par un seul emplacement de file d'attente et qu'aucune charge interactive n'est prévue sur ces noeuds — vous pouvez omettre `load_thresholds`.

- **Pour désactiver ces deux seuils, mettez `load_thresholds` sur `none` et `suspend_thresholds` sur `none` (voir `queue_conf(5)`).**

Ajustements de charge

Les ajustements de charge sont utilisés pour augmenter, virtuellement, la charge mesurée après qu'un travail a été acheminé. Ce mécanisme est utile en cas de machines sursouscrites afin d'assurer l'alignement avec les seuils de charge. Vous devez désactiver les ajustements de charge s'ils ne sont plus nécessaires car ils imposent au programmeur du travail supplémentaire lié au tri des hôtes et à la vérification des seuils de charge.

- **Pour désactiver les ajustements de charge, mettez `job_load_adjustments` sur `none` et `load_adjustment_decay_time` sur `0` dans la configuration du programmeur, `sched_conf(5)`.**

Programmation-à-la-demande

Par défaut, le système Sun Grid Engine, Enterprise Edition 5.3 lance les exécutions programmées à un intervalle de programmation fixe (voir `schedule_interval` dans `schedd_conf(5)`). L'avantage des intervalles fixes est qu'ils limitent la consommation d'UC par le `qmaster/scheduler`. Leur inconvénient est qu'ils régulent par étranglement le programmeur ce qui se traduit artificiellement par un débit limité. De nombreuses fermes de calcul ont des machines spécialement dédiées à `qmaster/scheduler` et, dans de telles configurations, il n'y a pas de raisons pour réguler le programmeur par étranglement.

- **Configurez la programmation à la demande en utilisant les paramètres `FLUSH_SUBMIT_SEC` et `FLUSH_FINISH_SEC` dans la section `schedd_params` de la configuration de grappe globale, `sge_conf(5)`.**

Si la programmation-à-la-demande est activée, le débit d'une ferme de calcul n'est limité que par la puissance de la machine qui héberge `qmaster/scheduler`.

Politiques

L'une des fonctionnalités les plus importantes du logiciel Sun Grid Engine, Enterprise Edition 5.3 est la notion de répartition de la puissance de calcul au moyen de politiques qui sont fixées par l'administrateur système. Pour des informations d'ordre général à ce sujet, consultez « Politiques, hôtes et démons », page 4.

La section suivante contient des instructions qui vous permettront de créer des politiques de base « fonctionnelles » et « basées sur le partage » pour votre site. Les instructions pour l'établissement de configurations de politiques « initiation limite » et « remplacement », ainsi que celles pour les configurations fonctionnelles et basées sur le partage, figurent dans le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*.

Configurations de politiques fonctionnelles

Les configurations de cette section utilisent toutes la politique fonctionnelle en tant que politique principale. Ce type de configuration assure qu'un partage défini est garanti à tout moment à chaque utilisateur, projet ou service. Les travaux des utilisateurs, projets ou services qui ont occupé moins de ressources que prévu sont préférés lorsque le système achemine les travaux vers des ressources inactives.

Simultanément, l'utilisation complète des ressources est garantie, car les proportions de partages inutilisées sont réparties entre les utilisateurs, projets et services qui ont besoin de ressources. La consommation passée de ressources n'est pas prise en compte.

▼ Création d'une programmation fonctionnelle basées sur les utilisateurs

L'objectif de cette configuration est de créer une certaine attribution des partages de toutes les ressources combinées dans la grappe Sun Grid Engine, Enterprise Edition 5.3 à différents utilisateurs. La programmation FCFS est utilisée avec les travaux d'un même utilisateur.

1. **Dans la section `schedd_params` de la configuration globale (`sge_conf(5)`), utilisez `SHARE_FUNCTIONAL_SHARES=1`.**
2. **Spécifiez le nombre de tickets fonctionnels (par exemple : 1000000) dans la configuration du programmeur (`sched_conf(5)`).**
3. **Ajoutez un utilisateur (`user(5)`) pour chaque utilisateur concerné par la programmation.**
4. **Attribuez des partages fonctionnels à chaque utilisateur.**

Attribuez les partages sous la forme d'un pourcentage du total, par exemple :

- utilisateurA (10)
- utilisateurB (20)
- utilisateurC (20)
- utilisateurD (50)

▼ Création d'une programmation fonctionnelle basée sur des projets

L'objectif de cette configuration est de créer une certaine attribution des partages de toutes les ressources combinées de la grappe Sun Grid Engine, Enterprise Edition 5.3 à plusieurs projets. La programmation FCFS est utilisée avec les travaux d'un même projet.

1. **Dans la section `schedd_params` de la configuration globale (`sge_conf(5)`), utilisez `SHARE_FUNCTIONAL_SHARES=1`.**
2. **Spécifiez le nombre de tickets fonctionnels (par exemple : 1000000) dans la configuration du programmeur (`sched_conf(5)`).**
3. **Ajoutez un projet (`project(5)`) pour chaque projet concerné par la programmation.**
Vous pouvez contrôler l'accès à des projets de privilèges plus élevés au moyen des fonctions `acl` et `xacl`. Pour plus d'informations sur ces fonctions, consultez le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3* ou le *Manuel de référence de Sun Grid Engine 5.3 et de Sun Grid Engine, Enterprise Edition 5.3*.
4. **Attribuez des partages fonctionnels à chaque projet.**

Attribuez les partages sous la forme d'un pourcentage du total. Par exemple :

- ProjetA (55)
- ProjetB (45)

▼ Création d'une programmation fonctionnelle basée sur les services

L'objectif de cette configuration est de créer une certaine attribution des partages de toutes les ressources combinées de la grappe Sun Grid Engine, Enterprise Edition 5.3 à plusieurs services. La programmation FCFS est utilisée avec les travaux d'un même service.

1. **Dans la section `schedd_params` de la configuration globale (`sge_conf(5)`), utilisez `SHARE_FUNCTIONAL_SHARES=1`.**
2. **Spécifiez le nombre de tickets fonctionnels (par exemple : 1000000) dans la configuration du programmeur (`sched_conf(5)`).**
3. **Ajoutez chaque service concerné par la programmation.**
4. **Attribuez des partages fonctionnels à chaque service.**

Attribuez les partages sous la forme d'un pourcentage du total. Par exemple :

- ServiceA (90)
- ServiceB (5)
- ServiceC (5)

Les configurations de cette section utilisent toutes la politique basée sur l'arborescence des partages en tant que politique principale. Ce type de configuration garantit dans le temps un partage défini aux instances configurées dans l'arborescence des partages. Les travaux associés aux « branches » de l'arborescence des partages où moins de ressources ont été consommées que prévu de par le passé (partage d'arborescence des partages) sont préférées lorsque le système achemine les travaux vers des ressources inactives. Au même moment, l'utilisation complète des ressources est garantie car des proportions de partages inutilisées sont toujours disponibles pour les travaux en attente associés à d'autres branches de l'arborescence des partages.

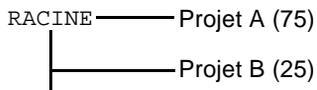
▼ Création d'une programmation d'arborescence de partages basée sur les projets avec FCFS au sein de chaque projet

L'objectif de cette configuration est de garantir dans le temps une certaine attribution des partages de toutes les ressources combinées dans la grappe Sun Grid Engine, Enterprise Edition 5.3 à différents projets. La programmation FCFS est utilisée avec les travaux d'un même projet.

Remarque – Certaines des instructions de cette section — contrairement aux instructions d'autres sections de ce document — nécessitent l'utilisation de l'interface homme-machine `qmon`. Pour des informations d'ordre général sur `qmon` et les instructions d'utilisation, consultez le *Guide d'administration et d'utilisation de Sun Grid Engine, Enterprise Edition 5.3*.

1. **Spécifiez le nombre de tickets d'arborescence des partages (par exemple : 1000000) dans la configuration du programmeur (`sched_conf(5)`).**
2. **Ajoutez un projet (`project(5)`) pour chaque projet concerné par la programmation.**
3. **Utilisez l'interface homme-machine `qmon`, configurez une arborescence de partages qui reflète la structure de tous les projets concerné par la programmation sous la forme de noeuds.**
4. **Attribuez des partages de l'arborescence des partages aux projets.**

Une structure simple ressemblerait à l'exemple suivant.



▼ Création d'une programmation d'arborescence des partages basée sur des partages avec des partages égaux pour chaque utilisateur

L'objectif de cette configuration est de garantir dans le temps une certaine attribution des partages de toutes les ressources combinées dans la grappe Sun Grid Engine, Enterprise Edition 5.3 à différents projets. Un partage égal est ciblé entre les travaux qui se disputent les ressources au sein d'un même projet.

1. **Spécifiez le nombre de tickets d'arborescence des partages (par exemple : 1000000) dans la configuration du programmeur (`sched_conf(5)`).**

2. Ajoutez un utilisateur (`user(5)`) pour chaque utilisateur concerné par la programmation.
3. Ajoutez un projet (`project(5)`) pour chaque projet concerné par la programmation.
4. En utilisant l'interface homme-machine `qmon`, configurez une arborescence de partages qui reflète la structure de tous les projets concerné par la programmation sous la forme de noeuds.
5. Attribuez des partages de l'arborescence de partages aux projets.
Un structure simple ressemblerait à l'exemple suivant.



6. Ajoutez la valeur par défaut `default` de l'utilisateur sous la forme d'un noeud terminal de l'arborescence sous chacun de ces projets.

▼ Création d'une programmation d'arborescence des partages basée sur des projets avec des partages individuels par utilisateur au sein de chaque projet

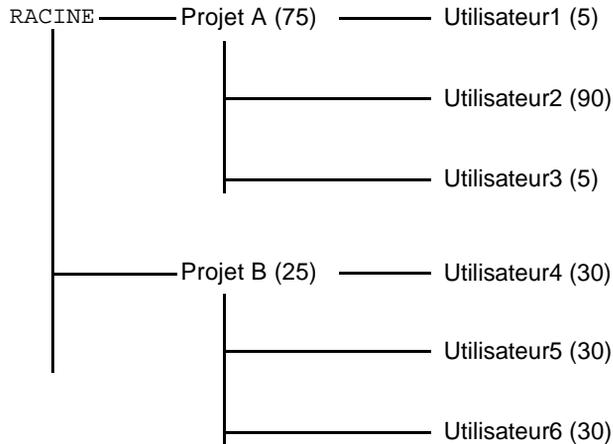
L'objectif de cette configuration est de garantir dans le temps une certaine attribution des partages de toutes les ressources combinées dans la grappe Sun Grid Engine, Enterprise Edition 5.3 à différents projets. L'attribution de partages individuels par utilisateur est nécessaire.

1. Spécifiez le nombre de tickets de l'arborescence des partages (par exemple : `1000000`) dans la configuration du programmeur (`sched_conf(5)`).
2. Ajoutez un utilisateur (`user(5)`) pour chaque utilisateur concerné par la programmation.
3. Ajoutez un projet (`project(5)`) pour chaque projet concerné par la programmation.
4. En utilisant l'interface homme-machine `qmon`, configurez une arborescence de partages qui reflète la structure de tous les projets concerné par la programmation sous la forme de noeuds.

5. Attribuez des partages de l'arborescence des partages aux projets.

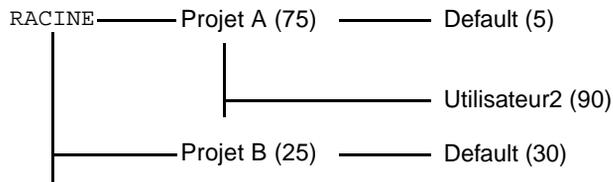
Ajoutez chaque utilisateur sous la forme d'un noeud final aux projets auxquels il ou elle a accès et attribuez des partages individuels.

Une structure simple ressemblerait à l'exemple suivant.



Discussion

Si l'attribution de partages individuels n'est requise que pour quelques utilisateurs, vous pouvez désigner la valeur par défaut `default` de l'utilisateur en combinaison avec les utilisateurs individuels sous un noeud de projet. Par exemple, l'arborescence illustrée ci-dessus peut être condensée comme suit :



Dépannage - problèmes courants

Utilisez la section suivante pour mieux diagnostiquer la cause des problèmes courants et y répondre.

- **Problème** – Le fichier de sortie relatif à votre travail indique `Warning: no access to tty; thus no job control in this shell...`
 - **Cause possible** – Un ou plusieurs de vos fichiers de connexion contiennent une commande `stty`. Ces commandes ne sont utiles que s'il y a un terminal.
 - **Solution possible** – Dans les travaux par lot de Sun Grid Engine, Enterprise Edition 5.3, aucun terminal n'est associé aux travaux. Vous devez soit supprimer toutes les commandes `stty` de vos fichiers de connexion, soit les mettre entre crochets avec une déclaration `if` qui contrôle la présence d'un terminal avant de poursuivre comme dans l'exemple ci-dessous.

```
/bin/csh:
stty -g          # checks terminal status
if ($status == 0) # succeeds if a terminal is present
<mettez toutes les commandes stty ici>
endif
```

- **Problème** – Le fichier journal d'erreurs standard indique : `'tty': Ambiguous` alors qu'il n'y a pas de référence à `tty` dans le shell de l'utilisateur, qui est appelé dans le script du travail.
 - **Cause possible** – `shell_start_mode` est, par défaut, `posix_compliant` ; par conséquent, tous les scripts de travail s'exécutent avec le shell spécifié dans la définition de la file d'attente, pas uniquement celui spécifié sur la première ligne du script des travaux.
 - **Solution possible** – Utilisez l'indicateur `-S` avec la commande `qsub` ou changez `shell_start_mode` en `unix_behavior`.
- **Problème** – Vous pouvez exécuter votre script de travail depuis la ligne de commande, mais il échoue lorsque vous l'exécutez via la commande `qsub`.
 - **Cause possible** – Il est possible que des limites de processus soient en cours de définition pour votre travail. Pour tester cela, écrivez un script de test qui effectue les fonctions `limit` et `limit -h`. Exécutez-le à la fois de façon interactive à l'invite de shell et par l'intermédiaire de la commande `qsub` pour comparer les résultats.
 - **Solution possible** – Assurez-vous d'avoir supprimé toutes les commandes des fichiers de configuration qui fixent les limites dans votre shell.

- **Problème** – Les hôtes d'exécution font état d'une charge de 99.99.
 - **Cause possible** – Les possibilités sont au nombre de trois.
 1. Le démon `execd` n'est pas exécuté sur l'hôte.
 2. Un domaine par défaut est spécifié de façon incorrecte.
 3. Le nom de l'hôte d'exécution que voit l'hôte `qmaster` est différent de celui que cet hôte d'exécution voit pour lui-même.
 - **Solution possible** – Selon la cause, une des solutions suivantes peut fonctionner (mettez les numéros qui figurent dans « Cause possible » en relation avec ceux des solutions suivantes).

1. En tant que super-utilisateur (`root`), démarrez le démon `execd` sur l'hôte d'exécution en exécutant le script `$SGE_ROOT/default/common/'rcsge'`.
2. En tant qu'administrateur Sun Grid Engine, Enterprise Edition, exécutez la commande `qconf -mconf` et changez la variable `default_domain` en `none`.
3. Si vous utilisez DNS pour résoudre les noms des hôtes de votre grappe de calcul, configurez `/etc/hosts` et NIS de sorte qu'ils retournent le nom de domaine qualifié (FQDN, *fully qualified domain name*) pour le nom de l'hôte principal (vous pouvez bien sûr toujours définir et utiliser un pseudonyme plus court ; par exemple : `168.0.0.1 monhôte.dom.com monhôte`).

Si vous n'utilisez pas DNS, assurez-vous que tous vos fichiers `/etc/hosts` et votre table NIS sont cohérents ; par exemple : `168.0.0.1 monhôte.corp monhôte` ou `168.0.0.1 monhôte`.

- **Problème** – Toutes les 30 secondes, un avertissement similaire au suivant s'imprime dans `<cellule>/spool/<hôte>/messages` :

```
Tue Jan 23 21:20:46 2001|execd|meta|W|local
configuration meta not defined - using global configuration
```

Mais il y a un fichier pour chaque hôte dans `<cellule>/common/local_conf/`, chacun avec FQDN.

- **Cause possible** – La résolution des noms d'hôtes sur votre machine, `meta`, retourne le nom abrégé ; mais au niveau de votre machine maître, `meta` avec FQDN est retourné.
- **Solution possible** – Assurez-vous que tous vos fichiers `/etc/hosts` et votre table NIS sont cohérents à ce sujet. Dans cet exemple, il pourrait y avoir par erreur une ligne telle que la suivante dans le fichier `/etc/hosts` de l'hôte `meta` :

```
168.0.0.1 meta meta.votre.domaine
```

Cette ligne *devrait* être la suivante :

```
168.0.0.1 meta.votre.domaine meta.
```

- **Problème** – Vous voyez de temps à autre des messages CHECKSUM ERROR, WRITE ERROR ou READ ERROR dans les fichiers messages des démons.
 - **Cause possible** – Aussi longtemps que ces messages n'apparaissent pas à intervalles de une seconde (il y en a en général de un à trente par jour), il est inutile de faire quoi que ce soit pour ce problème.
- **Problème** – Les travaux finiront dans une file d'attente particulière et retourneront ce qui suit dans qmaster/messages :

```
Wed Mar 28 10:57:15 2001|qmaster|masterhost|I|job 490.1
finished on host execest
```

Mais vous verrez alors les messages d'erreur suivants sur le fichier execest/messages de l'hôte d'exécution :

```
Wed Mar 28 10:57:15 2001|execd|execest|E|can't find directory
"active_jobs/490.1" for reaping job 490.1
```

```
Wed Mar 28 10:57:15 2001|execd|execest|E|can't remove
directory
"active_jobs/490.1": opendir(active_jobs/490.1) failed:
Input/output error
```

- **Cause possible** – Le répertoire \$SGE_ROOT, qui est automonté, est en train d'être démonté, ce qui fait que le démon sge_execd perd son cwd.
- **Solution possible** – Utilisez un répertoire spool local pour votre hôte execd. Définissez le paramètre, execd_spool_dir, en utilisant les commandes qmon ou qconf.
- **Problème** – Lorsque vous soumettez des travaux interactifs avec l'utilitaire qrsh, vous recevez le message d'erreur suivant :

```
% qrsh -l mem_free=1G error: error: no suitable queues
```

Des files d'attente sont tout de même disponibles pour les travaux par lots en utilisant l'utilitaire qsub, et peuvent être interrogées en utilisant qhost -l mem_free=1G et qstat -f -l mem_free=1G.

- **Cause possible** – Le message, error: no suitable queues, résulte de l'option -w e submit, qui est active par défaut pour les travaux interactifs tels que qrsh (recherchez -w e dans qrsh(1)). Cette option cause l'échec de la commande submit si qmaster n'est pas complètement sûr que le travail puisse être acheminé compte tenu de la configuration de grappe courante. L'intension de ce mécanisme est de décliner les requêtes de travail à l'avance au cas où elles pourraient ne pas être admises.

- **Solution possible** – Dans ce cas, `mem_free` est configuré pour être une ressource consommable, mais vous n'avez pas spécifié la quantité de mémoire devant être disponible sur chaque hôte. C'est délibérément que les valeurs de charge de mémoire ne sont pas prises en compte pour ce contrôle car elles varient, et ne peuvent donc pas être considérées comme faisant partie de la configuration de la grappe. Pour remédier à cela, vous pouvez effectuer au choix l'une des opérations suivantes.

Omettez ce contrôle de manière générale en court-circuitant le paramétrage par défaut de `qrsh`, `-w e`, de façon explicite en le soumettant avec `-w n`. Vous pouvez aussi utiliser `$SGE_ROOT/<cellule>/common/cod_request`.

Si vous avez l'intention de gérer `mem_free` comme une ressource consommable, spécifiez la capacité `mem_free` pour vos hôtes dans `complex_values` de `host_conf(5)` en utilisant `qconf -me <nomhôte>`.

Si vous n'avez pas l'intention de gérer `mem_free` comme une ressource consommable, faites-en de nouveau une ressource non-consommable dans la colonne `consumable` de `complex(5)` en utilisant `qconf -mc hôte`.

- **Problème** – `qrsh` n'effectuera pas l'acheminement vers le noeud sur lequel il se trouve. Depuis un shell `qsh` :

```
hôte2 [49]% qrsh -inherit hôte2 hostname
error: executing task of job 1 failed:

hôte2 [50]% qrsh -inherit hôte4 hostname
hôte4
```

- **Cause possible** – `gid_range` n'est pas suffisant. Il devrait s'agir d'une plage, pas d'un nombre isolé. Le système Sun Grid Engine, Enterprise Edition 5.3 attribue à chaque travail figurant sur un hôte un `gid` distinct.
- **Solution possible** – Ajustez `gid_range` en utilisant `qconf -mconf` ou l'interface homme-machine `qmon`. La plage préconisée est la suivante.

```
gid_range                20000-20100
```

- **Problème** – `qrsh -inherit -V` ne fonctionne pas lorsqu'il est utilisé dans un travail parallèle. Vous recevez le message suivant.

```
cannot get connection to "qlogin_starter"
```

- **Cause possible** – Ce problème survient avec les appels `qrsh` imbriqués et est dû au modificateur `-V`. Le premier appel `qrsh -inherit` définit la variable d'environnement, `TASK_ID` (l'ID de la tâche strictement intégrée au travail parallèle). Le deuxième appel `qrsh -inherit` utilise ensuite cette variable d'environnement pour l'enregistrement de sa tâche, qui échoue lorsqu'il essaie de démarrer la tâche avec le même ID que la première tâche déjà en cours d'exécution.
- **Solution possible** – Vous pouvez soit annuler le paramétrage de `TASK_ID` avant d'appeler `qrsh -inherit` soit choisir de ne pas utiliser le modificateur `-V` mais `-v`, et de n'exporter que les variables d'environnement dont vous avez vraiment besoin.
- **Problème** – `qrsh` ne semble pas fonctionner du tout. Vous recevez des messages similaires au suivant.

```

hôte2$ qrsh -verbose hostname
local configuration hôte2 not defined - using global
configuration
waiting for interactive job to be scheduled ...
Your interactive job 88 has been successfully scheduled.
Establishing /share/gridware/utilbin/solaris64/rsh session to
host exehost ...
rcmd: socket: Permission denied
/share/gridware/utilbin/solaris64/rsh exited with exit code 1
reading exit code from shepherd ...
error: error waiting on socket for client to connect:
Interrupted system call
error: error reading return code of remote command
cleaning up after abnormal exit of
/share/gridware/utilbin/solaris64/rsh
hôte2$

```

- **Cause possible** – Les permissions relatives à `qrsh` ne sont pas définies correctement.
- **Solution possible** – Contrôlez les permissions des fichiers suivants, qui se trouvent dans `$SGE_ROOT/utilbin/` (vous remarquerez que `rlogin` et `rsh` doivent être `setuid` et appartenir au `root`).

```

-r-s--x--x 1 root root 28856 Sep 18 06:00 rlogin*
-r-s--x--x 1 root root 19808 Sep 18 06:00 rsh*
-rwxr-xr-x 1 sgeadmin adm 128160 Sep 18 06:00 rshd*

```

Remarque – Le répertoire `$SGE_ROOT` a également besoin d'être monté dans un système NFS avec l'option `setuid`. S'il est monté avec `nosuid` depuis votre client de soumission, `qrsh` (et les commandes associées) ne fonctionnera pas.

- **Problème** – Lorsque vous essayez de lancer une opération `make` distribuée, `qmake` se termine avec le message d'erreur suivant.

```
qrsh_starter: executing child process qmake failed: No such
file or directory
```

- **Cause possible** – Le système Sun Grid Engine, Enterprise Edition 5.3 démarrera une instance de `qmake` sur l'hôte d'exécution. Si l'environnement Sun Grid Engine, Enterprise Edition 5.3 (en particulier la variable `PATH`) n'est pas configuré dans le fichier des ressources de shell de l'utilisateur (`.profile/.cshrc`), cet appel `qmake` échouera.
- **Solution possible** – Utilisez l'option `-v` pour exporter la variable d'environnement `PATH` vers le travail `qmake`. Voici un appel `qmake` typique.

```
qmake -v PATH -cwd -pe make 2-10 --
```

- **Problème** – Lorsque vous utilisez l'utilitaire `qmake`, vous obtenez le message d'erreur suivant.

```
waiting for interactive job to be scheduled ...timeout (4 s)
expired while waiting on socket fd 5

Your "qrsh" request could not be scheduled, try again later.
```

- **Cause possible** – La variable d'environnement `ARCH` est peut-être mal définie dans le shell depuis lequel `qmake` a été appelé.
- **Solution possible** – Définissez correctement la variable `ARCH` en lui donnant une valeur prise en charge qui corresponde à un hôte disponible dans votre grappe, ou spécifiez la valeur correcte au moment de la soumission ; par exemple : `qmake -v ARCH=solaris64 ...`

Diagnostic des problèmes

Le système Sun Grid Engine, Enterprise Edition 5.3 offre plusieurs méthodes de rapport qui peuvent vous aider à diagnostiquer les problèmes. Les sections suivantes en présentent l'utilisation.

Travaux en attente non-acheminés

Il arrive parfois qu'un travail en attente puisse être exécuté mais ne soit pas acheminé. Pour en diagnostiquer la raison, Sun Grid Engine, Enterprise Edition 5.3 propose une paire d'utilitaires et d'options, `qstat -j <idtravail>` et `qalter -w v <idtravail>`.

- `qstat -j <idtravail>`

Si activé, `qstat -j <idtravail>` fournit à l'utilisateur une liste des raisons pour lesquelles un certain travail n'a pas été acheminé à l'occasion de la dernière exécution programmée. Cette surveillance peut être activée ou désactivée, et être à l'origine d'une surcharge indésirée au niveau de la communication entre le démon `schedd` et `qmaster` (voir `schedd_job_info` dans `sched_conf(5)`). Voici un exemple de sortie pour un travail d'ID 242059.

```
% qstat -j 242059
scheduling info: queue "fangorn.q" dropped because it is temporarily not available
                 queue "lolek.q" dropped because it is temporarily not available
                 queue "balrog.q" dropped because it is temporarily not available
                 queue "saruman.q" dropped because it is full
                 cannot run in queue "bilbur.q" because it is not contained in its hard
                 queue list (-q)
                 cannot run in queue "dwain.q" because it is not contained in its hard
                 queue list (-q)
                 has no permission for host "ori"
```

Ces informations sont générées directement par le démon `schedd` et tiennent compte de l'utilisation courante de la grappe. Cela n'est pas toujours ce qui vous intéresse ; par exemple, si tous les emplacements de la file d'attente sont déjà occupés par des travaux d'autres utilisateurs, aucun message détaillé n'est généré pour le travail qui vous intéresse.

- `qalter -w v <idtravail>`

Cette commande liste les raisons pour lesquelles un travail n'est en principe pas acheminable. Dans cet objectif, une exécution de programmation d'essai est effectuée. La particularité d'une telle exécution est que toutes les ressources consommables (emplacements compris) sont considérées comme entièrement disponibles pour le travail. Dans le même ordre d'idées, toutes les valeurs de charge sont ignorées car elles varient.

Rapports signalant un état d'erreur E au niveau d'un travail et d'une file

Les erreurs de travail ou de file d'attente sont indiquées par un E majuscule dans la sortie de `qstat`. Un travail passe à l'état d'erreur lorsque le système Sun Grid Engine, Enterprise Edition 5.3 essaye d'exécuter un travail figurant dans une file d'attente mais qu'il échoue pour une raison spécifique de ce travail. Une file d'attente passe à l'état d'erreur lorsque le système Sun Grid Engine, Enterprise Edition 5.3 essaye d'exécuter un travail figurant dans cette file, mais échoue pour une raison spécifique de la file.

Le système Sun Grid Engine, Enterprise Edition 5.3 offre aux utilisateurs et aux administrateurs plusieurs options pour rassembler des informations de diagnostic en cas d'erreur d'exécution des travaux. Étant donné qu'à la fois l'état d'erreur d'une file et celui d'un travail résultent de l'exécution ratée d'un travail, les possibilités de diagnostic s'appliquent à ces deux types d'états d'erreur.

- **Courrier d'abandon utilisateur**

Si les travaux sont soumis avec l'option de `submit`, `-m a`, un courrier d'abandon est envoyé à l'adresse spécifiée avec l'option `-M utilisateur[@hôte]`. Ce courrier d'abandon contient des informations de diagnostic sur les erreurs des travaux, il s'agit de la source d'informations recommandée pour les utilisateurs.

- **Comptabilité `qacct`**

Si aucun courrier d'abandon n'est disponible, l'utilisateur peut exécuter la commande `qacct -j` pour obtenir des informations sur l'erreur du travail en provenance de la fonction de comptabilité des travaux du système Sun Grid Engine, Enterprise Edition 5.3.

- **Courrier d'abandon administrateur**

Un administrateur peut demander des courriers d'administrateur sur les problèmes d'exécution des travaux en spécifiant une adresse e-mail appropriée (voir `administrator_mail` dans `sge_conf(5)`). Ce courrier d'administrateur contient davantage d'informations de diagnostic détaillées que le courrier d'abandon utilisateur, cette méthode est celle recommandée en cas d'erreurs d'exécution de travail fréquentes.

- **Fichiers de messages**

Si aucun courrier d'administrateur n'est disponible, le fichier `qmaster messages` devrait être le premier contrôlé. Vous pouvez y trouver des messages consignés relatifs à un travail donné en recherchant l'ID de ce travail. Dans l'installation par défaut, le fichier `qmaster messages` est `$SGE_ROOT/default/spool/qmaster/messages`.

Vous pouvez également dans certains cas trouver des informations supplémentaires dans les messages du démon `execd` depuis lequel le travail a été lancé. Utilisez `qacct -j <idtravail>` pour trouver l'hôte depuis lequel le travail a été lancé, et recherchez `idtravail` dans `$SGE_ROOT/default/spool/<hôte>/messages`.