Sun™ ONE Grid Engine, Enterprise Edition 5.3 管理およびユーザーマニュアル

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No. 816-7470-10
2002 年 9 月, Revision A

コメントの宛先: docfeedback@sun.com

原典: *Sun Grid Engine, Enterprise Edition 5.3 Administration and User's Guide*
Part No: 816-4739-11
Revision A

目次

はじめに

このマニュアルは、Sun Grid Engine, Enterprise Edition 5.3 製品に関する予備知識的な情報とインストール方法、その全機能の使用方法をまとめた包括的な使用手引き書です。

内容の紹介

このマニュアルは、Sun Grid Engine, Enterprise Edition 5.3 のユーザーと、必ずしもユーザーと同じ作業を行うわけでないシステム管理者の両方を対象にしています。このため、このマニュアルは、ユーザーまたは管理者のどちらに特に関係しているかに従って 4 部に分かれています。

各部の内容と対象読者は以下のとおりです。

- PART I - 背景と定義

  PART I では、ユーザーと管理者の両方を対象に、製品の用途とコンポーネント、用語などを詳しく説明しています。

- PART II - 最初に行う作業

  PART II では、製品をインストールするユーザー (一般には管理者) を対象に、新規インストール方法とアップグレードインストール方法を詳しく説明しています。

- PART III - Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアの使用方法

  PART IIIでは、ユーザーと管理者の両方を対象にしています。Sun Grid Engine, Enterprise Edition を使用するにあたっての予備知識的な情報を提供するとともに、実際の使用方法をまとめています。

- PART IV - 管理

  PART IV では、経験の豊富なシステム管理者向けに予備知識的な情報と管理作業の実施方法をまとめています。

UNIX コマンド

このマニュアルには、UNIX® の基本的なコマンド、およびシステムの停止、システムの起動、デバイスの構成などの基本的な手順の説明は記載されていません。

基本的なコマンドや手順についての説明は、次のマニュアルを参照してください。

- 『Sun 周辺機器 使用の手引き』
- Solaris™ オペレーティング環境についてのオンライン AnswerBook2™
- 本システムに付属している他のソフトウェアマニュアル

書体と記号について

書体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例。	.login ファイルを編集します。 ls -a を実行します。 % You have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表します。	**マシン名% su** Password:
AaBbCc123 または**ゴシック**	コマンド行の可変部分。実際の名前や値と置き換えてください。	rm *filename* と入力します。 rm **ファイル名** と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 この操作ができるのは「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅をこえる場合に、継続を示します。	**% grep '^#define \ XV_VERSION_STRING'**

シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	マシン名%
UNIX の Bourne シェルと Korn シェル	$
スーパーユーザー (シェルの種類を問わない)	#

関連マニュアル

用途	タイトル	Part No.
リファレンス	『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』	816-7476-10

Sun のオンラインマニュアル

サンの各種システムマニュアルは下記 URL より参照できます。

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

Solaris およびその他のマニュアルは下記 URL より参照できます。

`http://docs.sun.com`

コメントをお寄せください

弊社では、マニュアルの改善に努力しており、お客様からのコメントおよびご忠告をお受けしております。コメントは下記宛に電子メールでお送りください。

`docfeedback@sun.com`

電子メールの表題にはマニュアルの Part No. (8xx-xxxx-xx) を記載してください。

なお、現在日本語によるコメントには対応できませんので、英語で記述してください。

PART I 背景と定義

このマニュアルの PART I は 1 つの章で構成されています。

- 第 1 章 - 1 ページの「Sun™ Grid Engine, Enterprise Edition 5.3 入門」

この章は簡潔ですが、重要でないというわけではありません。ユーザーおよび管理者のどちらも、この章の内容を理解しておくことが大切です。この章の内容は次のとおりです。

- 複雑なコンピューティング環境における Product Name ソフトウェアの第一の役割の説明
- Product Name 製品の主要コンポーネントとそれらの働きの一覧
- Product Name 環境で理解しておくべき重要な用語集

第1章

Sun™ Grid Engine, Enterprise Edition 5.3 入門

この章では、Sun™ Grid Engine, Enterprise Edition 5.3 システムに関する、ユーザーおよび管理者のどちらにも有用な予備知識的な情報を提供します。Sun Grid Engine, Enterprise Edition の役割は、それがなければ無秩序になってしまう可能性があるクラスタ化されたコンピュータの世界を管理することです。この章は以下のような内容になっています。

- グリッドコンピューティングの概要
- Sun Grid Engine, Enterprise Edition 5.3 のグラフィカルユーザーインタフェース、QMON の概要
- Sun Grid Engine, Enterprise Edition の各主要コンポーネントの説明
- ユーザーおよび管理者が使用できるクライアントコマンドの説明付き一覧
- Sun Grid Engine, Enterprise Edition 5.3 の全用語集

グリッドコンピューティングとは

概念的には、グリッドはかなり単純です。グリッドとは、仕事をするコンピューティング資源の集まりです。最も簡単な形態では、ユーザーからはグリッドは、分散された強力な資源への単一アクセスポイントを提供する大きなシステムに見えます。この節で後ほど説明するように、もっと複雑な形態では、グリッドは多数のアクセスポイントをユーザーに提供します。しかし、どのような場合も、ユーザーはグリッドを単一の計算資源とみなすことができます。Sun Grid Engine, Enterprise Edition などの資源管理ソフトウェアはユーザーからジョブを受け付け、それらのジョブが適切なシステム上で実行されるように、資源管理ポリシーに基づいて実行予定を立てます。ユーザーは、実行場所を気にすることなく、文字通り一度に百万単位のジョブの実行を要求できます。

2 つとして同じグリッドはありません。あらゆる状況にマッチするグリッドの規模もありません。大きく分けてグリッドには 3 つのクラスがあり、小は単独のシステムから、大は数千のプロセッサを利用するスーパーコンピュータ級の規模までをカバーします。

- クラスタグリッド - 連携するコンピュータホストで構成され、単一のプロジェクトまたは部署内のユーザーに単一アクセスポイントを提供する最も単純なグリッドです。
- 構内グリッド - 組織内の複数のプロジェクトまたは部署でコンピューティング資源を共有することを可能にするグリッドです。構内グリッドを導入することで、周期的な業務プロセスからデータのレンダリング、マイニングなどの広範囲のさまざまな業務を処理できるようになります。
- グローバルグリッド - 組織の垣根を越えて非常に大規模な仮想システムを構築する構内グリッドの集まりです。ユーザーは自分の組織内で利用できる資源をはるかにしのぐ計算パワーを利用できます。

図 1-1 は、これら 3 つのグリッドクラスの概念図です。クラスタグリッドでは、ユーザーの各ジョブは、クラスタを構成するシステムの 1 つで処理されます。ただし、ユーザーのクラスタグリッドがもっと複雑な構内グリッドの構成要素で、その構内グリッドが最大規模のグローバルグリッドの構成要素である場合、そのユーザーのジョブは、世界中に分散している任意のメンバー (実行ホスト) によって処理されることがあります。

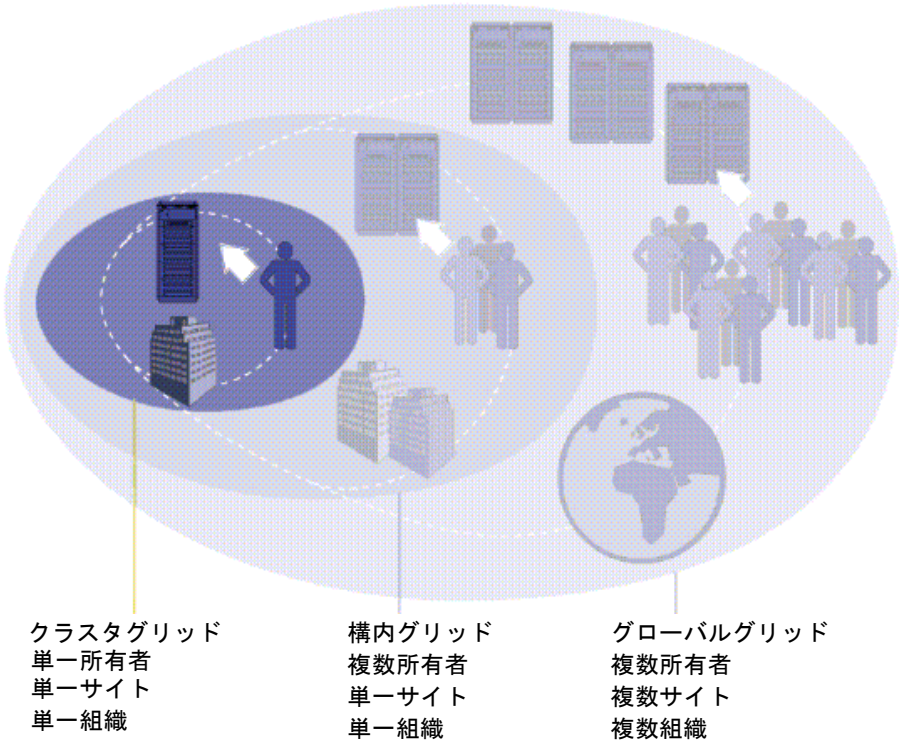


図 1-1　3 つのグリッドクラス

Sun の資源管理用ソフトウェアの最新版、Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアは構内グリッドに求められるパワーと柔軟性を提供します。Sun Grid Engine, Enterprise Edition は、その姉妹品の Sun Grid Engine によって実現されている既存のクラスタグリッドに特に有用であり、構内にある既存のすべて Sun Grid Engine クラスタグリッドを統合することによって構内グリッドへのスムーズな移行を可能にします。Sun Grid Engine, Enterprise Edition はまた、企業が初めてグリッドコンピューティングモデルに移行するのに適した製品でもあります。

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアは、組織の技術および管理スタッフによって決められた社内資源ポリシーに基づいて計算パワーの供給を調整をします。すなわち、それらのポリシーに基づいて、構内グリッド全体で最適な資源利用が図れるよう、グリッドで使用可能な計算資源を調べて、その情報を収集し、それらの資源を自動的に割り当てます。

構内グリッド内で協調が実現されるようにするには、グリッドを使用するプロジェクト所有者がポリシーを取り決める必要があります。特殊なプロジェクト要件で手動の優先指定が行えるようポリシーに柔軟性をもたせ、自動的にポリシーが監視、実施されるようにします。

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアは、計算資源を求める多数の部署やプロジェクト間の資源利用を調停することができます。

資源およびポリシー管理に基づく作業負荷の管理

Sun Grid Engine, Enterprise Edition システムは、異機種からなる分散コンピューティング環境用の高度な資源管理ツールです。作業負荷の管理では、資源の管理およびポリシーの運用を通じて共有資源の利用を管理することによって、企業の目標 (生産性、タイミング、サービスレベルなど) が最高度に達成されるようにします。現場では、タイミング (ジョブの締め切り) や重要性 (ジョブ優先順位と事前ユーザー配分) の変更に対応しながら、計算資源を最大限に利用し、最高のスループットを上げられるようシステムを構成します。

Sun Grid Engine, Enterprise Edition ソフトウェアは、複数の共有資源から構成される UNIX 環境において高度な資源管理とポリシー運用を可能にします。このシステムが標準的な負荷管理ツールに比べて優れているのは、主なものでも以下に挙げる機能があるためです。

- 画期的な動的スケジューリングと資源管理。個々の現場に固有の管理ポリシーを実施できます。
- 動的なパフォーマンスデータの収集。ジョブレベルの最新の資源消費およびシステム負荷情報をスケジューラに提供します。
- CSP (Certificate Security Protocol) を使用した暗号化に基づくセキュリティの機能強化。セキュリティ強化したシステムでは、メッセージのテキストがそのまま転送されるのではなく、秘密鍵を使用して暗号化されます。
- 高度なポリシー運用。生産性、タイミング、サービスレベルなどの企業の目標を定義・実現することができます。

Sun Grid Engine, Enterprise Edition ソフトウェアは、計算面で要求の厳しい仕事の実行をシステムに要求し、それに伴う作業負荷を透過的に分散できるようにする手段を提供します。ユーザーは Sun Grid Engine, Enterprise Edition システムにバッチジョブや対話形式のジョブ、並列ジョブの実行を要求することができます。

Sun Grid Engine, Enterprise Edition ソフトウェアはチェックポイントプログラムにも対応しています。チェックポイントジョブは、負荷に応じてユーザーの介入なしでワークステーション間を移動します。

Sun Grid Engine, Enterprise Edition ソフトウェアによって、管理者は、Sun Grid Engine, Enterprise Edition のジョブを監視、制御するための包括的なツールを手にすることができます。

システム運用の仕組み

Sun Grid Engine, Enterprise Edition システムは外部世界からジョブ (ユーザーからのコンピュータ資源の要求) を受け付けて、実行可能な状態になるまでそのジョブを保留しておき、実行可能な状態になると、実行デバイスに送信します。実行中はジョブを管理し、実行が完了するとその記録をログに書き込みます。

一例として、世界の主要都市にある大規模な「マネーセンター」銀行を想像してください。

資源と要求の引き合わせ

銀行のロビーには、それぞれに用件が異なる利用者が何十人も並び、サービスを受ける順番を待っています。ある利用者は、単に自分の口座から少額の現金を引き出したいだけです。その利用者のすぐ後に訪れた利用者は銀行の投資専門家と会う約束をしていて、複雑な事業に着手する前に助言を得たいと考えています。長い列のその 2 人の利用者の前には、多額の借り入れを受けるためにやってきた別の利用者もいます。その利用者の前にはさらに 8 人の利用者が並んでいます。

銀行の利用者や利用目的によって、必要とされる銀行の資源の種類やレベルは異なります。その日の銀行には、1 人の利用者の単純な口座からの現金引き出しに対処する時間が十分にある行員が多数いました。しかし、借り入れの申し込みをする利用者を助ける貸し付け担当者は 1 人か 2 人いるだけです。別の日には、この状況は逆転しているかもしれません。

当然のことながら、結果的に利用者はサービスを待つ必要があります。用件がただちに確認されて、使用可能な資源に引き合わせられさえすれば、その多くがただちにサービスを受けられるとしても、です。

Sun Grid Engine, Enterprise Edition システムが銀行の責任者であった場合、サービスの提供方法は異なったものになります。

- 銀行のロビーに入ると、利用者は名前と関係先 (会社の代理など)、用件を告げるよう求められます。
- 利用者の来店時間が記録されます。
- ロビーで利用者が告げた情報に基づいて、適切でただちに使用可能な資源に合致する用件の利用者、優先順位が最も高い用件の利用者、ロビーで最も長時間待っている利用者がサービスを受けます。
- 当然、「Sun Grid Engine, Enterprise Edition 銀行」の行員は、同時に複数の利用者にサービスを提供できます。Sun Grid Engine, Enterprise Edition システムは、最も負荷が小さく、最も適切な行員に新しい利用者を割り当てようとします。

- 銀行の責任者としての Sun Grid Engine, Enterprise Edition システムでは、サービスポリシーを定義することができます。代表的なサービスポリシーは、「利幅が大きい商用の利用者には優先的なサービスする」「これまでのサービスがよくなかった特定の利用者には手厚いサービスをする」「約束がある利用者にはタイムリーな応対をする」「銀行役員の直接の要求があったときは特定の利用者を優先する」などといったものです。
- Sun Grid Engine, Enterprise Edition 責任者は、こうしたポリシーを自動的に実施、監視、調整します。優先順位の高い利用者はすぐにサービスを受け、他の利用者も担当する必要がある行員からより多くの配慮が払われます。Sun Grid Engine, Enterprise Edition 責任者は、期待したペースで利用者に対するサービスが進行しているかどうかを調べ、そうでない場合は、銀行のサービスポリシーを守れるよう、ただちにサービスレベルを調整して、問題に対処します。

ジョブとキュー: Sun Grid Engine の世界

Sun Grid Engine, Enterprise Edition システムでは、ジョブは銀行の利用者に相当し、銀行のロビーではなくコンピュータの保留域で待機します。コンピュータサーバー上にあるキューは銀行行員の働きをして、ジョブにサービスを提供します。銀行の利用者同様、ジョブの要求内容 (一般には使用可能なメモリー、実行速度、使用可能なソフトウェアライセンスなどの要求で構成される) はそれぞれ非常に異なり、一部のキューしかその要求に対応するサービスを提供できないことがあります。

Sun Grid Engine, Enterprise Edition ソフトウェアは以下のようにして使用可能な資源とジョブの要求を調整します。

- Sun Grid Engine, Enterprise Edition システムを使ってジョブの実行依頼をするユーザーは、そのジョブの要求プロファイルを宣言します。また、システムは、ユーザーの識別情報とそのプロジェクトまたはユーザーグループとの関係情報を取り込みます。ユーザーがジョブの実行依頼をした時間も記録されます。
- 文字通り、新しいジョブの実行に対するキューの使用予定が立てられたその瞬間、Sun Grid Engine, Enterprise Edition システムはそのキューに適したジョブを特定し、最も優先順位が高いか、最も待ち時間が長いジョブをただちにディスパッチします。
- Sun Grid Engine, Enterprise Edition のキューは、多数のジョブを並行して実行することを可能にします。Sun Grid Engine, Enterprise Edition システムは、最も負荷が小さく、最も適切なキューで新しいジョブを開始しようとします。

資源利用ポリシーの多様性

Sun Grid Engine, Enterprise Edition クラスタの管理者は、そのサイトに適切な条件に従って高度な資源利用ポリシーを定義することができます。そうしたポリシーには以下の 4 つがあります。

- **業務優先 (Functional)** - このポリシーでは、管理者は、ユーザーまたはジョブが持つ、特定のユーザーグループあるいはプロジェクトなどとの関係に従って特別な対処をすることができます。
- **基本割当 (Share-based)** - このポリシーにおけるサービスレベルは、割り当てられた資源利用資格、他のユーザーおよびユーザーグループの対応する利用資格、全ユーザーの過去の資源利用、システム内の現在のユーザーの有無に依存します。
- **締め切り優先 (Deadline)** - このポリシーは、ジョブが特定の時点までに、または特定の時点で完了する必要がある場合に必ず呼び出されます。このため、このポリシーを達成するには、特別な対処が必要になることがあります。
- **一時優先 (Override)** - このポリシーには、Sun Grid Engine, Enterprise Edition のクラスタ管理者による手動の介入が必要で、自動的なポリシー実施を変更します。

Sun Grid Engine, Enterprise Edition のポリシー管理は自動的にクラスタ内の共有資源の使用を制御して、最高度にその運用目標を達成できるようにします。複数のジョブが同じ資源を得ようとする場合は、優先順位の高いジョブが優先的にディスパッチされて、高い CPU 使用資格を受けます。Sun Grid Engine, Enterprise Edition ソフトウェアはすべてのジョブの進行状況を監視し、その状況に応じて、またポリシーに定義されている目標に従ってその相対的な優先順位を調整します。

チケットパラダイムによるポリシー運用

ポリシーはすべて、Sun Grid Engine, Enterprise Edition の「チケット」という概念を使って定義されます。チケットは、株式を公開している会社の株式に例えることができます。株を多く所有している株主ほど、会社に対するその株主の重要性が増します。株主 A が株主 B の 2 倍の株を保有している場合、A は B の 2 倍の投票権を持ち、会社に対する重要性は 2 倍です。これと同じで、チケットを多く持っている Sun Grid Engine, Enterprise Edition ジョブほど重要性が増します。ジョブ A がジョブ B の 2 倍のチケットを持っている場合、ジョブ A にはジョブ B の 2 倍の資源利用資格が与えられます。

Sun Grid Engine, Enterprise Edition のジョブは 4 通りあるポリシーのすべてからチケットを取り出すことができます。そして、その総数ばかりでなく、各ポリシーから取り出されたチケット数も、しばしば時間の経過とともに変化します。

Sun Grid Engine, Enterprise Edition クラスタの管理者は、各ポリシーに割り当てるチケット数を全体として制御します。ジョブに対するのと同様、ポリシーとポリシーの間の相対的な重要性は、このチケットの割り当て量によって決まります。特定のポリシーに割り当てられているチケットプールを使用し、管理者は基本割当モードでの

み Sun Grid Engine, Enterprise Edition システムを実行することも、90% 基本割当、10% 業務優先というように混在モードでシステムを実行することもできます。図 1-2 は、ポリシーとチケットのこの相互関係を表しています。

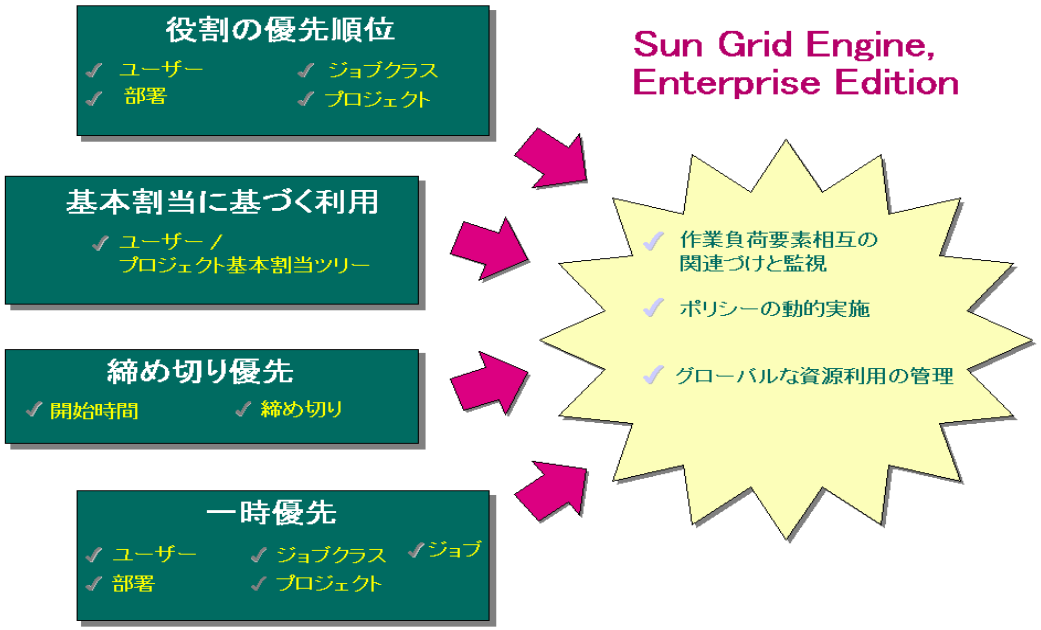


図 1-2　Sun Grid Engine, Enterprise Edition 5.3 システムにおけるポリシーとチケットの相互関係

Sun Grid Engine, Enterprise Edition 5.3 のコンポーネント

図 1-3 は、Sun Grid Engine, Enterprise Edition で特に重要なコンポーネントと、システムにおけるそれらコンポーネントの相互関係を示しています。以下では、これらのコンポーネントの働きを説明します。

ホスト

Sun Grid Engine, Enterprise Edition 5.3 システムでは、次の 4 種類のホストが不可欠です。

- マスター
- 実行
- 管理
- 実行依頼

マスターホスト

マスターホストは、クラスタ活動全体の中心です。マスターホストはマスターデーモン (sge_qmaster) とスケジューラデーモン (sge_schedd) を実行します。この 2 つのデーモンはともに、キューやジョブなどの Sun Grid Engine, Enterprise Edition 5.3 コンポーネントのすべてを制御し、コンポーネントの状態やユーザーのアクセス権限などの表を管理します。

デフォルトでは、マスターホストは管理および実行依頼ホストでもあります。これらのホストに関連する節を参照してください。

実行ホスト

実行ホストは、Sun Grid Engine, Enterprise Edition ジョブを実行する権限を持つノードです。このため、実行ホストは Sun Grid Engine, Enterprise Edition のキューのホストの役割を果たし、Sun Grid Engine, Enterprise Edition 実行デーモン (sge_execd) を実行します。

管理ホスト

Sun Grid Engine, Enterprise Edition システムに対するあらゆる種類の管理運用業務を行う権限をホストに付与することができます。

実行依頼ホスト

実行依頼ホストは、バッチジョブのみの実行依頼と制御を行うためのホストです。具体的には、実行依頼ホストにログインしているユーザーは、qsub を使ってジョブの実行を依頼したり、qstate を使ってジョブの状態を制御したりすることができます。また、Sun Grid Engine, Enterprise Edition の OSF/1 Motif グラフィカルユーザーインタフェース (QMON) を使用することもできます (QMON については、13 ページの「Sun Grid Engine,Enterprise Edition のグラフィカルユーザーインタフェース (QMON)」の節で説明)。

注 – 1 つのホストが、上記の複数のクラスに属することができます。

デーモン

Sun Grid Engine, Enterprise Edition 5.3 システムの機能は、4 つのデーモンによって実現されます。

sge_qmaster - マスターデーモン

クラスタの管理とスケジューリングの中心として sge_qmaster マスターデーモンはホストやキュー、ジョブ、システム負荷、ユーザーのアクセス権に関する表を管理します。このデーモンは sge_schedd からスケジューリング情報を受け取り、適切な実行ホスト上の sge_execd に処理要求をします。

sge_schedd - スケジューラデーモン

sge_schedd スケジューリングデーモンは、sge_qmaster の助けを借りて、常に最新のクラスタ状態を表示できるようにします。また、スケジューリングに関して次の決定をします。

- どのジョブをどのキューにディスパッチするかの決定
- 配分、優先順位、締め切りを維持するためのジョブの順序および優先順位の変更方法に関する決定

sge_schedd はこれらの情報を sge_qmaster に転送し、それを受けた sge_qmaster が要求された処理を開始します。

sge_execd - 実行デーモン

sge_schedd 実行デーモンは、それが動作しているホスト上のキューとキュー内のジョブの実行を担当します。このデーモンは、そのホスト上のジョブの状態や負荷などの情報を定期的に sge_qmaster に転送します。

sge_commd - 通信デーモン

sge_commd 通信デーモンは、既知の TCP ポートを使って通信します。このデーモンは、Sun Grid Engine, Enterprise Edition コンポーネント間のあらゆる通信に使用されます。

キュー

Sun Grid Engine, Enterprise Edition の各キューは、特定のホスト上で並行して実行することが可能な、1 つのクラスのジョブ用のコンテナです。移動の可不可などのジョブのいくつかの属性は、キューによって決まります。存在している間ずっと、ジョブは特定のキューに関連付けられています。ジョブに対して可能なことの一部は、この関連付けの影響を受けます。たとえば、キューが一時停止された場合は、そのキューに関連付けられているすべてのジョブも一時停止されます。

Sun Grid Engine, Enterprise Edition システムでは、ジョブを直接キューに実行要求する必要はありません。ジョブの要求プロファイル (メモリー、オペレーティングシステム、使用可能なソフトウェアなど) を指定しさえすれば、Sun Grid Engine, Enterprise Edition ソフトウェアが、負荷の小さいホスト上の適切なキューに自動的にジョブをディスパッチします。キューにジョブを直接実行依頼すると、ジョブがそのキューとホストに結び付けられ、Sun Grid Engine, Enterprise Edition が、負荷が小さいデバイス、あるいはより適したデバイスを選択できなくなります。

クライアントコマンド

Sun Grid Engine, Enterprise Edition のコマンド行ユーザーインタフェースは、キューの管理やジョブの実行依頼・削除、ジョブの状態調査、さらにはキューやジョブを一時停止したり、使用可能にしたりするための、一群の補助的なプログラム (コマンド) で構成されています。Sun Grid Engine, Enterprise Edition システムでは、次の補助的なプログラムを使用します。

- `qacct` - クラスタログファイルから任意のアカウンティング情報を抽出します。
- `qalter` - 保留中のジョブの属性を変更します。
- `qconf` - クラスタとキュー構成用のユーザーインタフェースを提供します。
- `qdel` - ユーザーやオペレータ、マネージャーにジョブまたはそのサブセットにシグナルを送信する手段を提供します。
- `qhold` - 実行依頼されたジョブの実行を保留します。
- `qhost` - Sun Grid Engine, Enterprise Edition 実行ホストの状態情報を表示します。
- `qlogin` - 自動的に選択された、負荷の小さい適切なホストとの `telnet` または同様のログインセッションを開始します。
- `qmake` - UNIX 標準の `make` 機能の代わりに使用できるコマンドです。機能的には `make` を拡張して、適切なマシンのクラスタに個々の `make` ステップを分散できるようになっています。
- `qmod` - キューを一時停止または使用可能にすることができます (所有者のみ)。そのキューに関連付けられていて、現在アクティブなすべてのプロセスにも、シグナルが送信されます。
- `qmon` - X-windows の Motif コマンドインタフェースと監視機能を提供します。

- qresub - 実行または保留中のジョブをコピーすることによってジョブを新規作成します。
- qrls - qhold などを使って割り当てられていたホールドからジョブを解放します (上記の qhold を参照)。
- qrsh - このコマンドは、以下のようなさまざまな目的に使用することができます。
  - Sun Grid Engine, Enterprise Edition システムを使用して対話型のアプリケーションを遠隔実行する (UNIX 標準の rsh 機能に相当)
  - 実行後すぐに端末入出力 (標準 / エラー出力と標準入力) と端末制御が可能なバッチジョブの実行依頼を可能にする
  - ジョブが完了するまでアクティブな状態を継続するバッチジョブ実行依頼クライアントを実現する
  - Sun Grid Engine, Enterprise Edition ソフトウェアの制御下での並列ジョブのタスクの遠隔実行を可能にする
- qselect - 指定された選択条件に一致するキュー名を一覧表示します。通常、このコマンドの出力は、選択されたキューにアクションを適用する目的で他の Sun Grid Engine, Enterprise Edition コマンドに供給されます。
- qsh - 負荷の小さいホスト上で xterm 内に対話形式のシェルを開きます。このシェルであらゆる種類の対話形式のジョブを実行することができます。
- qstat - クラスタに関連付けられているすべてのジョブとキューの状態を一覧表示します。
- qsub - Sun Grid Engine, Enterprise Edition システムにバッチジョブを実行依頼するためのユーザーインタフェースです。
- qtcsh - 広く知られ、かつ使用されている UNIX の C-Shell (csh) の高機能版、tcsh と完全互換で、その代わりに使用できるコマンドです。コマンドシェルの機能が拡張され、Sun Grid Engine, Enterprise Edition ソフトウェアを使用し、特定のアプリケーションの実行を負荷の小さい適切なホストに透過的に分散できるようになります。

これらのプログラムはすべて sge_commd を介して sge_qmaster と通信します。図 1-3 の Sun Grid Engine, Enterprise Edition のコンポーネントの相互関係図には、このことが示されています。

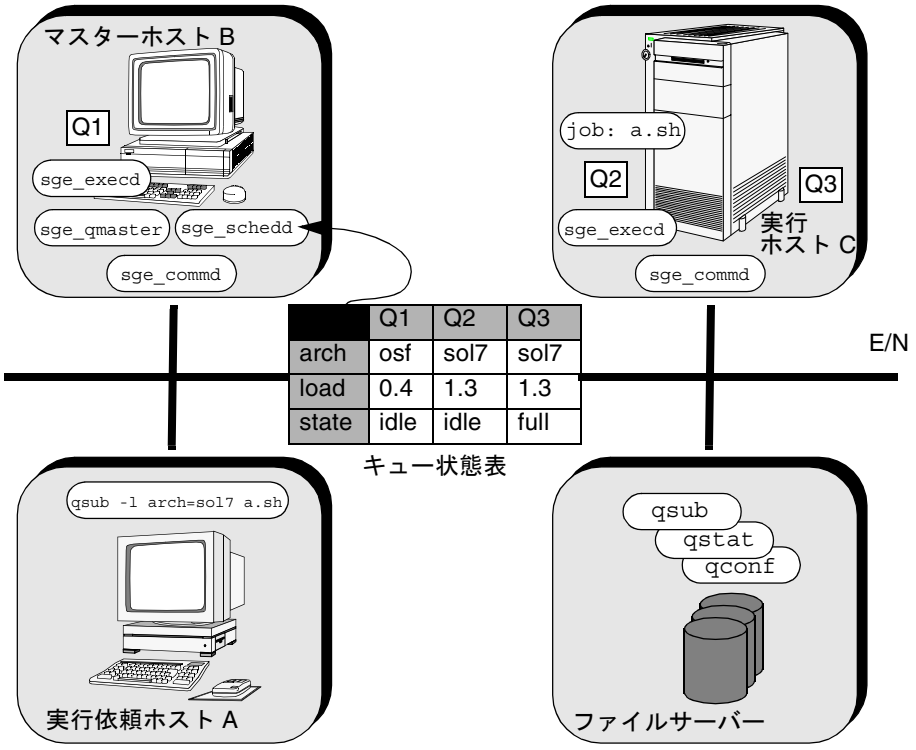


図 1-3　Sun Grid Engine, Enterprise Edition システムのコンポーネントの相互関係

Sun Grid Engine,Enterprise Edition のグラフィカルユーザーインタフェース (QMON)

すべてではありませんが、Sun Grid Engine, Enterprise Edition 5.3 システムにおける大部分の作業は、グラフィカルユーザーインタフェース (GUI) ツールの QMON を使用して行うことができます。図 1-4 は QMON のメインメニューを示しています。このメニューは、しばしばユーザーおよび管理者両方の機能の使用開始場所になります。各アイコンは GUI ボタンで、ボタンをクリックすると、さまざまな作業を開始することができます。各ボタンの名前はそのボタンの機能の説明にもなっていて、ボタンの上にマウスポインタを置くと、名前が表示されます。

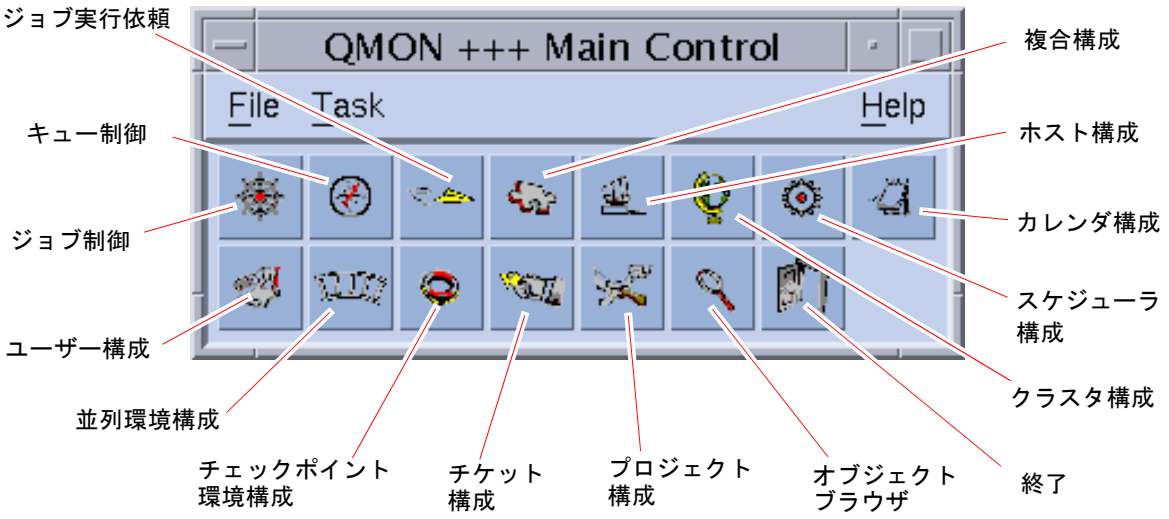


図 1-4　　QMON のメインメニュー、定義

QMON のカスタマイズ

QMON のルック & フィールは、大体は専用のリソースファイルで定義されます。QMON にはデフォルト値として標準的な値がすでに設定されていますが、*<sge_root>*/qmon/Qmon にサンプルのリソースファイルを参考にカスタマイズすることもできます。

クラスタ管理では、QMON 専用のリソース定義を標準の `.Xdefaults` または `.Xresources` に取り込むか、XAPPLRESDIR などの標準の検索パスで参照される場所にサイト専用の Qmon ファイルを置くことによって、`/usr/lib/X11/app-defaults/Qmon` などの標準の場所にサイト専用のデフォルト値をインストールすることができます。上記のどれが自分のケースに当てはまるかについては、管理者にお尋ねください。

また、ユーザーは自分のホームディレクトリ (または個人用の XAPPLRESDIR 検索パスが指し示す別の場所) に Qmon ファイルをコピーして変更するか、専用の `.Xdefaults` または `.Xresources` ファイルに必要なリソース定義を含めることによって、自分の好みに合った設定を行うことができます。個人用の Qmon リソースファイルは、X11 環境の運用中または起動時に `.xinitrc` ファイルなどで `xrdb` コマンドを使用して組み込むこともできます。

可能なカスタマイズについての詳細は、サンプルの Qmon ファイルのコメント行を参照してください。

図 5-3 および図 5-13 に示すジョブ制御およびキュー制御の「カスタマイズ」ダイアログボックスに、QMON をカスタマイズするもう 1 つの方法の説明があります。このどちらのダイアログボックスでも、「保存」ボタンを使用して、ユーザー個人のホームディレクトリの .qmon_preferences ファイルにフィルタおよび表示の定義を保存することができます。再起動すると、QMON はこのファイルを読み取り、定義された動作を有効にします。

Sun Grid Engine 用語集

ここでは、Sun Grid Engine, Enterprise Edition の世界と資源管理全般でよく使われる用語を簡単にまとめています。これまでのところ、それらの用語の多くはまだ現れていませんが、これからこのマニュアルの他の部分で現れます。

アクセスリスト ユーザーおよび UNIX グループのリストで、このリストに登録されたユーザーおよび UNIX グループはキューまたは特定のホストなどの資源へのアクセスが許可または拒否されます。ユーザーおよびグループは複数のアクセスリストに登録することができ、同じアクセスリストをさまざまなコンテキストで利用することができます。

一時停止 実行中のジョブを保留状態にして、実行マシン上に残しておくことです。ジョブが異常終了するチェックポイントとは異なります。一時停止されたジョブは、スワップメモリーやファイル領域などの資源を消費し続けます。

一時優先ポリシー Sun Grid Engine, Enterprise Edition のポリシーの 1 つで、業務優先、基本割当、締め切り優先ポリシーの自動的な資源管理を取り消す目的でよく使用されるポリシーです。Sun Grid Engine, Enterprise Edition では、ジョブ、ユーザー、ユーザーグループ、ジョブクラス、プロジェクトに優先指定を行うことができます。

移動 ジョブの実行が再開される前にチェックポイントを別のホストに移動することです。

オペレータ 構成や設定を変更できないことを除けば、マネージャーと同じコマンドを実行できる、Sun Grid Engine, Enterprise Edition の運用を担当するユーザーです。

基本割当ツリー Sun Grid Engine, Enterprise Edition の基本割当ポリシーを階層形式で定義したものです。

基本割当ポリシー Sun Grid Engine, Enterprise Edition のポリシーの 1 つで、ユーザーやプロジェクト、グループの資源利用資格を階層形式で定義することができます。たとえば企業は、事業部や部署、部署で活動中のプロジェクト、それらプロジェクトで仕事をするユーザーグループ、それらユーザーグループのユーザーに分割す

ることができます。資源配分に基づく階層は基本割当ツリーと呼ばれ、このツリーを定義すると、Sun Grid Engine, Enterprise Edition によって利用資格の配分が自動的に行われます。

キュー 実行ホスト上で並行して実行することが可能な特定のクラスのジョブ用のコンテナです。

業務優先ポリシー Sun Grid Engine, Enterprise Edition のポリシーの 1 つで、ジョブやユーザー、ユーザーグループ、プロジェクト、ジョブクラスに特定のレベルの重要性を割り当てるポリシーです。たとえば高い優先順位のプロジェクト (そのすべてのジョブも含む) は、優先順位が低いプロジェクトよりも多くの資源配分を受けることができます。

クラスタ Sun Grid Engine, Enterprise Edition の機能が動作する、ホストと呼ばれるマシンの集まりです。

グループ UNIX グループのことです。

資源 ジョブを実行することで消費または占有される計算デバイスです。メモリーや CPU、入出力帯域幅、ファイル領域、ソフトウェアなどがこれにあたります。

資源利用 「資源消費」のもう 1 つの言い方です。Sun Grid Engine, Enterprise Edition システムでは、資源利用は、CPU 消費やメモリー占有の経過時間、実行された入出力量の合計 (管理者が設定可能な重み付き) によって決まります。

締め切り優先ポリシー Sun Grid Engine, Enterprise Edition のポリシーの 1 つで、特定の期限までに、または特定の期限で完了する必要があるポリシーです。管理者は、締め切りのあるジョブの重要性のレベルの上限値や、そうしたジョブの実行依頼を許可するユーザーを指定することができます。

ジョブ バッチジョブは、ユーザーの介入なしに実行することが可能で、端末にアクセスする必要のない UNIX シェルスクリプトです。

対話形式のジョブは、ユーザーとの対話のために *xterm* ウィンドウを開くか、遠隔ログインセッションに相当するものを提供する Sun Grid Engine, Enterprise Edition コマンド (`qrsh`、`qsh`、`qlogin`) を使って開始されたセッションです。

ジョブクラス ある意味で同等で、同様に扱われる一群のジョブを意味します。Sun Grid Engine, Enterprise Edition では、ジョブクラスは、資源の要求内容が同じで、適切なキューが同じ一群のジョブと定義とされます。

所有者 キューを一時停止 / 停止解除、使用可能 / 使用不可にすることができるユーザーです。一般にユーザーは、そのワークステーションにあるキューの所有者になります。

セル 独立した構成とマスターマシンを持つ独立した Sun Grid Engine, Enterprise Edition クラスタです。セルを使用して、独立した管理ユニットを疎結合することができます。

ソフト資源 必要ではあるが、ジョブを開始するために必ずしも割り当てておく必要のない資源です。こうした資源は、使用可能になった時点でジョブに割り当てられます。この逆は「ハード資源」です。

チェックポイント環境 Sun Grid Engine, Enterprise Edition の構成の 1 つで、特定のチェックポイントの実行方法に関係するイベントやインタフェース、アクションを定義します。

チェックポイント機能 いわゆるチェックポイントにジョブの実行状態を保存することをいいます。実行状態を保存することによって、ジョブの実行が異常終了しても、それまでの情報やすでに完了している作業を失うことなく、再開することができます。実行を再開する前に別のホストにチェックポイントを移動することを「移動」といいます。

チケット Sun Grid Engine, Enterprise Edition で資源配分の定義に使用される一般的な単位です。チケットを多く持っている Sun Grid Engine, Enterprise Edition のジョブ、ユーザー、プロジェクトほど、その重要性が増します。あるジョブが別のジョブより 2 倍多くのチケットを持っている場合、そのジョブには 2 倍の資源消費資格が与えられます。

ハード資源 ジョブを開始するために必ず割り当てておく必要がある資源です。この逆は「ソフト資源」です。

配分 (Sun Grid Engine, Enterprise Edition のみ) 利用資格 (下記を参照) と同じです。特定のジョブやユーザー、またはユーザーグループ、プロジェクトによる消費が計画されている資源量を意味します。

配列ジョブ 同じタスクだが、それぞれに独立したタスクの一群からなるジョブを意味します。タスクは独立したジョブに非常によく似ています。配列ジョブのタスクがジョブと異なるのは、一意のタスク識別子 (整数 1 つ) が割り当てられることだけです。

複合 キューやホスト、あるいはクラスタ全体に関連付けることができる一群の属性です。

部署 Sun Grid Engine, Enterprise Edition の業務優先および一時優先スケジューリングポリシーで同様に扱われるユーザーおよびグループのリストです。ユーザーおよびグループは、1 つの部署にしか所属できません。

プロジェクト Sun Grid Engine, Enterprise Edition のプロジェクトを単にプロジェクトといいます。

並列環境 Sun Grid Engine, Enterprise Edition の構成の 1 つで、Sun Grid Engine, Enterprise Edition が並列ジョブを正しく処理するために必要なインタフェースを定義します。

並列ジョブ 相互に密接に関連する複数のタスクで構成されるジョブです。タスクは複数のホストに分散することができます。通常、並列ジョブは共有メモリーやメッセージ受け渡し (MPI、PVM) などの通信ツールを使用して、タスクを同期、連携させます。

ホスト Sun Grid Engine, Enterprise Edition の機能が動作するマシンです。

ポリシー Sun Grid Engine, Enterprise Edition 管理者がその動作の定義に使用できる一群の規則や構成をポリシーといいます。ポリシーは、Sun Grid Engine, Enterprise Edition によって自動的に実施されます。

マネージャー Sun Grid Engine, Enterprise Edition のすべてを操作することが可能なユーザーのことです。マスターホストおよび管理ホストとして宣言された他のすべてのマシンのスーパーユーザーは、マネージャー特権を持ちます。マネージャー特権は、スーパーユーザー以外のユーザーアカウントにも割り当てることができます。

ユーザー 少なくとも 1 つの実行依頼ホストまたは実行ホストに正当なログインアカウントを持つユーザーは、Sun Grid Engine, Enterprise Edition にジョブの実行を依頼をして実行できます。

ユーザーセット 上記のアクセスリストか部署のいずれかを意味します。

優先順位 Sun Grid Engine, Enterprise Edition のジョブ相互の相対的な重要性のレベルを意味します。

利用資格 (Sun Grid Engine, Enterprise Edition のみ) 配分 (上記を参照) と同じです。特定のジョブやユーザー、またはユーザーグループ、プロジェクトによる消費が計画されている資源量を意味します。

PART II 最初に行う作業

このマニュアルの PART II は 1 つの章で構成されています。

- 第 2 章 - 21 ページの「インストール」

この章では、Sun Grid Engine, Enterprise Edition 5.3 製品を初めてインストールするための手順ばかりでなく、以前のバージョンを新しいリリースにアップグレードするための手順についても説明しています。

第2章

インストール

この章では、次の 3 つのインストール作業の手順を詳細に説明します。

- Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアの完全な新規インストール
- 特殊な暗号化機能を利用した保護インストール
- インストールの検証

注 – この章の手順では、Solaris™ オペレーティング環境が動作するコンピュータにインストールするものと仮定しています。Sun Grid Engine, Enterprise Edition がインストールされる他のオペレーティングシステムアーキテクチャとの機能上の相違点は、*<sge_root>*/doc ディレクトリにある arc_depend_ から始まる名前のファイルに記載されています。このファイル名の残りの部分は、そのファイルのコメントが該当するオペレーティングシステムのアーキテクチャを示しています。

基本インストールの概要

注 – 以降の説明は、新規の基本 Sun Grid Engine, Enterprise Edition 5.3 インストールにのみ該当します。セキュリティ保護機能を追加した新規システムのインストール方法については、36 ページの「CSP 保護されたシステムをインストールして設定する」を参照してください。以前のバージョンの Sun Grid Engine 製品をアップグレードインストールする方法については、『Sun Grid Engine, Enterprise Edition 5.3 ご使用にあたって』をご覧ください。

完全インストールは、以下の広範な作業で構成されます。

- Sun Grid Engine, Enterprise Edition の構成および環境の計画
- 外部媒体からワークステーションへの Sun Grid Engine, Enterprise Edition 配布ファイルの読み込み

- Sun Grid Engine, Enterprise Edition システムを構成するマスターホストとすべての実行ホスト上でのインストールスクリプトの実行
- 管理および実行依頼ホスト情報の登録
- インストールの検証

インストール作業は、Solaris オペレーティング環境に精通しているスタッフが行ってください。プロセス全体は次の 3 つの段階に分けて行います。

フェーズ 1 - 計画作成

インストールの計画作成段階では、以下の作業を行います。

- Sun Grid Engine, Enterprise Edition 環境を単一クラスタ、またはセルと呼ばれるサブクラスタの集合のどちらの環境にするかの決定
- Sun Grid Engine, Enterprise Edition のホストにするマシンの選定。各マシンをどのタイプのホスト (マスターホスト、シャドウマスターホスト、管理ホスト、実行依頼ホスト、実行ホスト、またはその組み合わせ) にするかを決定します。
- 各 Sun Grid Engine, Enterprise Edition ユーザーのユーザー名が、あらゆる実行依頼および実行ホストで共通であることの確認
- Sun Grid Engine, Enterprise Edition のディレクトリ構成の決定。たとえばすべてのワークステーションで完全な 1 つのツリーとしてディレクトリを構成することも、ディレクトリをクロスマウントすることも、あるいは一部ワークステーションは部分ディレクトリツリーの構成にすることもできます。また、Sun Grid Engine, Enterprise Edition の各ルートディレクトリの作成場所を決定する必要もあります。
- サイトのキューの構成の決定
- ネットワークサービスを NIS ファイルとして定義するか、`/etc/services` において各ワークステーションにローカルにするかの決定
- 以降のインストール手順で使用するインストールワークシートの完成 (30 ページの「インストール計画を作成する」を参照)。

フェーズ 2 - ソフトウェアのインストール

インストール段階では、以下の作業を行います。

- インストールディレクトリの作成とそのディレクトリへの配布ファイルの読み込み
- マスターホストのインストール
- すべての実行ホストのインストール
- すべての管理ホストの登録
- すべての実行依頼ホストの登録

フェーズ 3 - インストールの検証

検証段階では、以下の作業を行います。

- マスターホスト上でデーモンが動作していることの確認
- 各実行ホスト上でデーモンが動作していることの確認
- Sun Grid Engine, Enterprise Edition が簡単なコマンドを実行することの確認
- テストジョブの実行依頼

インストール計画の作成

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアをインストールする前に、実際の環境に完全に合った結果を得るための方法を綿密に計画する必要があります。この節では、以降の作業に影響する重要な決定に役立つ情報を提供します。

前提となる作業

ここでは、本番用の Sun Grid Engine, Enterprise Edition システムをインストールするために必要な情報を提供します。

インストールディレクトリ<*sge_root*>

Sun Grid Engine, Enterprise Edition 配布媒体の内容の読み込み先となるディレクトリを準備します。このディレクトリは Sun Grid Engine, Enterprise Edition のルートディレクトリと呼ばれ、以降の Sun Grid Engine, Enterprise Edition システムの運用中、現在のクラスタ構成や、ディスクへのスプールに必要なすべてのデータの保存に使用されます。

どのホストでも、適切に参照されるディレクトリパス名を使用してください。たとえばオートマウンタを使用してファイルシステムをマウントする場合、<*sge_root*> は `/tmp_mnt/usr/SGE` ではなく、`/usr/SGE` に設定してください。このマニュアルでは、このインストールディレクトリを参照する際、<*sge_root*> 環境変数を使用します。

<*sge_root*> は、Sun Grid Engine, Enterprise Edition ディレクトリツリーの最上位のディレクトリです。セルを構成するすべての Sun Grid Engine, Enterprise Edition コンポーネントは、起動時に <*sge_root*>/<*cell*>/`common` を読み取れる必要があります(28 ページの「セル」の節を参照)。必要なアクセス権限については、26 ページの「ファイルアクセス権限」の節を参照してください。

インストールと管理が簡単に行えるよう、このディレクトリは、Sun Grid Engine, Enterprise Edition のインストールを行うどのホストでも読み取り可能である必要があります。このためには、たとえば、NFS などのネットワークファイルシステムから利用できるディレクトリを使用することができます。ホストにローカルのファイルシステムを使用するようにした場合は、インストールを開始する前にホストごとにインストールディレクトリをコピーする必要があります。

ルートディレクトリ内のスプールディレクトリ

- Sun Grid Engine, Enterprise Edition マスターホストの場合、スプールディレクトリは *<sge_root>*/*<cell>*/spool/qmaster と *<sge_root>*/*<cell>*/spool/schedd の下に作成されます。
- 実行ホストの場合は、*<sge_root>*/*<cell>*/spool/<exec_host> というスプールディレクトリが作成されます。

これらのディレクトリを他のマシンにエクスポートする必要はありません。ただし、マスターおよびすべての実行ホストで*<sge_root>* ツリー全体をエクスポートして、書き込みアクセス可能にすると、管理が容易になります。

ディレクトリ構成

Sun Grid Engine, Enterprise Edition のディレクトリ構成 (たとえばすべてのワークステーションで完全なツリーにするか、ディレクトリをクロスマウントするか、一部ワークステーションは部分ディレクトリツリーにするなど) とそのルートディレクトリの作成場所を決定します。

注 – 前回のインストールの重要な情報はすべて残すことができるものの、基本的に、インストールディレクトリかスプールディレクトリ、あるいはその両方を変更するには、システムをインストールし直す必要があります。このため、綿密な検討を行って適切なインストールディレクトリを選択するようにしてください。

Sun Grid Engine, Enterprise Edition のインストールのデフォルトでは、インストールディレクトリ下のディレクトリ階層に Sun Grid Engine, Enterprise Edition のシステムやマニュアル、スプール領域、構成ファイルがインストールされます (25 ページの図 2-1「ディレクトリ階層の例」を参照)。このデフォルトのインストールでよければ、26 ページの「ファイルアクセス権限」で説明しているアクセス権限を許可するディレクトリを選択してください。

スプール領域は、基本インストール中に別の場所に配置するように選択することができます (第 6 章、145 ページの「ホストおよびクラスタ構成」を参照)。

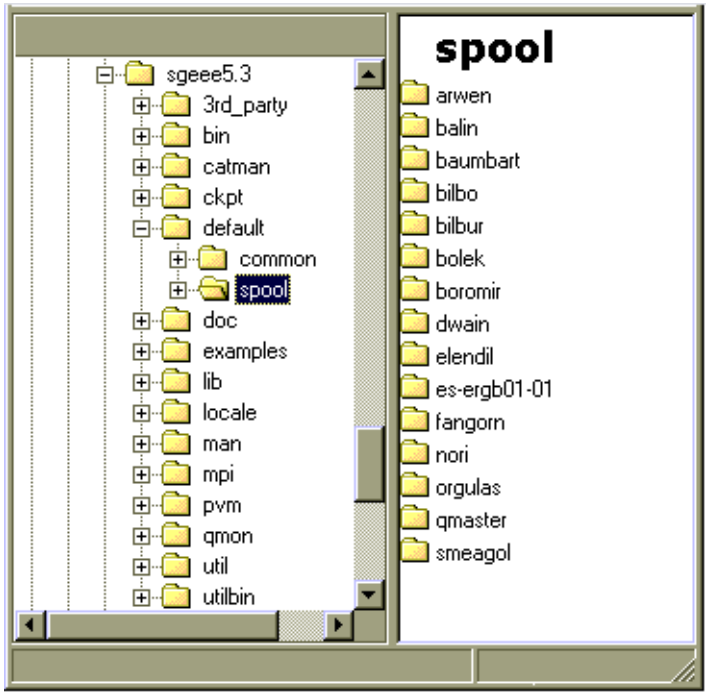


図 2-1　　ディレクトリ階層の例

必要な空きディスク容量

Sun Grid Engine, Enterprise Edition ディレクトリツリーには、以下の一定の空きディスク容量が必要です。

- バイナリを含まないインストールキット (マニュアル類を含む) 用に 40M バイト
- バイナリ 1 セットあたり 10 ～ 15M バイト (Cray アーキテクチャの場合は全バイナリで約 35M バイトを消費)

Sun Grid Engine, Enterprise Edition ログファイル用の理想的な空きディスク容量は以下のとおりです。

- マスターホスト: スプールディレクトリ用に 30 ～ 200M バイト (クラスタサイズに依存)
- 実行ホスト: 10 ～ 20M バイト

注 – マスターホストと実行ホストのスプールディレクトリはユーザー設定可能で、必ずしもデフォルトの *<sge_root>* の下に置く必要はありません。スプールディレクトリの場所の変更は、基本インストールの終了後に行ってください (第 6 章、145 ページの「ホストおよびクラスタ構成」を参照)。

インストールアカウント

Sun Grid Engine, Enterprise Edition は、root アカウントでインストールすることも、特権のないアカウント (たとえば自分のアカウント) でインストールすることもできます。特権のないアカウントでインストールした場合、Sun Grid Engine, Enterprise Edition のジョブを実行できるのは、そのアカウントを所有する特定の 1 人のユーザーだけになり、他のすべてのアカウントに対してアクセスは拒否されます。root アカウントでインストールすると、この制限は解消されますが、完全なインストールを行うには root 権限が必要になります。

ファイルアクセス権限

root でインストールした場合は、共有ファイルシステムでのすべてのホストに対する root の読み取り・書き込みアクセス権の設定で問題が起き、ネットワーク全体のファイルシステムに *<sge_root>* を作成できないことがあります。root 以外の管理ユーザーアカウント (たとえば `sgeadmin` という) を使用して、すべての Sun Grid Engine, Enterprise Edition コンポーネント全体のファイル処理を強制的に Sun Grid Engine, Enterprise Edition ソフトウェアに行わせることができます。その場合、必要になるのは、その特定のユーザーについて共有 foot ファイルシステムに対する読み取り・書き込みアクセス権が必要なだけです。Sun Grid Engine, Enterprise Edition のインストールでは、管理ユーザーアカウントでファイルを処理するかどうかを問い合わせます。これに対して Yes と応答して、正当なユーザー名を指定すると、そのユーザー名を使用したファイルが処理されます。これ以外の場合は、インストールを実行しているユーザー名が使用されます。

どの場合も、あらゆるホスト上でファイルの処理に使用するアカウントは、Sun Grid Engine, Enterprise Edition ルートディレクトリに読み取り・書き込みアクセスできるようにする必要があります。また Sun Grid Engine, Enterprise Edition のインストールでは、その配布媒体の読み込み元のホストがこのディレクトリにアクセスできることが前提になります。

ネットワークサービス

ネットワークサービスを NIS ファイルとして定義するか、`/etc/services` において各ワークステーションにローカルにするかを決定します。NIS を使用する場合は、NIS services マップにエントリを追加できるよう NIS サーバーホストを特定します。

Sun Grid Engine, Enterprise Edition サービスは sge_commd です。NIS マップにこのサービスを追加するには、予約済みの未使用ポート番号 (1024 より小さい番号) を選択してください。以下は、sge_commd エントリの例です。

```
sge_commd 536/tcp
```

マスターホスト

Sun Grid Engine, Enterprise Edition はマスターホストから制御します。マスターホストはマスターデーモンの sge_qmaster を実行します。マスターホストは Sun Grid Engine, Enterprise Edition の運用の中心であり、このため次の条件を満たす必要があります。

- 安定したプラットフォームであること。
- 他の処理で過度にビジーにならないこと。
- Sun Grid Engine, Enterprise Edition のデーモンの実行用として、少なくとも 20M バイトの未使用主メモリーがあること。非常に大規模なクラスタの場合 (数百、数千のホストで構成されていて、一度に数万のジョブが発生するようなシステム) は、1G バイト以上の未使用主メモリーが必要になることがあります。また、CPU も 2 つあると良いかもしれません。
- (省略可能) Sun Grid Engine, Enterprise Edition ディレクトリの *<sge_root>* がローカルに存在すること (ネットワークトラフィックの削減に役立つ)。

シャドウマスターホスト

シャドウマスターホストは、マスターホストまたはマスターデーモンで問題が発生した場合に、sge_qmaster の機能をバックアップします。シャドウマスターホストになるには、マシン次の条件を満たしている必要があります。

- sge_shadowd を実行していること。
- ディスクに記録される sge_qmaster のステータス、ジョブ、キュー構成情報を共有していること。具体的には、シャドウマスターホストには、sge_qmaster のスプールディレクトリと *<sge_root>*/*<cell>*/common ディレクトリに対する読み取り・書き込み root または admin ユーザーアクセス権が必要です。
- *<sge_root>*/*<cell>*/common/shadow_masters ファイルに、シャドウマスターホストであることを定義する行が含まれていること。

ホストのシャドウマスターホストの機能は、上記の条件が満たされるとただちに有効になります。このため、ホストをシャドウホストにするために、Sun Grid Engine, Enterprise Edition のデーモンを再起動する必要はありません。

実行ホスト

実行ホストは、Sun Grid Engine, Enterprise Edition に実行依頼されたジョブを実行します。実行ホストごとにインストールスクリプトを実行します。

管理ホスト

Sun Grid Engine, Enterprise Edition のオペレータおよびマネージャーは、管理ホストから、キューの再構成や Sun Grid Engine, Enterprise Edition ユーザーの追加などの管理業務を行います。 マスターホストのインストールスクリプトは、マスターホストを自動的に管理ホストにします。

実行依頼ホスト

Sun Grid Engine, Enterprise Edition のジョブは、実行依頼ホストから実行依頼し、制御することができます。マスターホストのインストールスクリプトは、マスターホストを自動的に実行依頼ホストにします。

セル

Sun Grid Engine, Enterprise Edition は単一のクラスタとして構成することも、セルと呼ばれる疎結合されたクラスタの集まりとして構成することもできます。`SGE_CELL` 環境変数は参照先のクラスタを示します。Sun Grid Engine, Enterprise Edition を単一クラスタとしてインストールすると、`SGE_CELL` が設定されずに、セル値は `default` とみなされます。

ユーザー名

ジョブの実行依頼をしようとしているユーザーが実行依頼して、その実行に必要な実行ホストの使用権限を持っていることを Sun Grid Engine, Enterprise Edition が確認するには、関係する実行依頼ホストと実行ホストでそのユーザー名が同じである必要があります。この条件があるため、一部マシンでユーザー名の変更が必要になることがあります。

注 – マスターホスト上のユーザー名は権限検査に関係なく、一致する必要も、存在する必要さえありません。

キュー

サイトのニーズに合ったキュー構成を考えてください。このことは、どのキューのどの実行ホストに配置するか、順次、対話形式、並列などの種類のジョブ用のキューが必要かどうか、各キューに必要なジョブスロット数などのキュー構成を決定することを意味します。

また、Sun Grid Engine, Enterprise Edition の管理者は、インストールでデフォルトのキュー構成を作成させることもできます。このデフォルトのキュー構成は、システムを理解したり、最初のキュー構成として利用して、後で調整したりするの適しています。

注 – Sun Grid Engine, Enterprise Edition ソフトウェアはディレクトリにインストールされますが、そのときに作成される大部分の設定は、システムの運用中に自由に変更することができます。

すでに Sun Grid Engine, Enterprise Edition について十分な知識があるか、以前にクラスタに適用するキュー構成の決定をしている場合、インストールで自動的にデフォルトのキュー構成を作成する必要はありません。その場合は、インストールの完了後に独自のキュー構成を定義した文書を作成し、第 7 章、169 ページの「キュー構成とキューカレンダの構成」に進んでください。

▼ インストール計画を作成する

1. インストールを開始する前に、以下に示すような表の形式でインストール計画をまとめます。.

パラメータ	値
<*sge_root*>	
admin ユーザー	
admin グループ	
sge_commd ポート番号	
マスターホスト	
シャドウマスターホスト	
実行ホスト	
管理ホスト	
実行依頼ホスト	

図 2-2　　インストール計画を記入するための書式

2. 上記の定義のようにアクセス権を設定することによって、Sun Grid Engine, Enterprise Edition の配布内容とスプールおよび構成ファイルを入れるファイルシステムおよびディレクトリが正しく作成されるようにします。

▼ 配布媒体を読み込む

Sun Grid Engine, Enterprise Edition は CD-ROM で提供されます。この CD-ROM の入手方法については、システム管理者に尋ねるか、お持ちのシステムのマニュアルを参照してください。CD-ROM には、`Sun_Grid_Engine_Enterprise_5.3` という

ディレクトリが含まれています。そして、このディレクトリに tar とサンの pkagadd の両方の形式で製品が含まれています。推奨する形式は pkgadd 形式です。

1. admin ユーザーアカウントを作成します (26 ページの「ファイルアクセス権限」を参照)。
2. 配布媒体にアクセスできるようにして、システムにログインします。ファイルサーバーが直接接続しているシステムにログインすることを推奨します。
3. 23 ページの「インストールディレクトリ<sge_root>」の説明に従って、Sun Grid Engine, Enterprise Edition のインストールキットを読み込み先となるインストールディレクトリを作成します。インストールディレクトリに対するアクセス権限が正しく設定されていることを確認してください。

   以下の説明では、インストールディレクトリを *<install_dir>* と記します。
4. Sun Grid Engine, Enterprise Edition クラスを構成する qmaster、実行、実行依頼ホストが使用する、すべてのバイナリアーキテクチャ用のバイナリをインストールします。

   使用するインストール方法に従って、以下のいずれかを行います。

pkgadd を使用する場合

以下のコマンドを入力したら、ベースディレクトリ (デフォルトは /gridware/sge) と admin ユーザー (デフォルトは sgeadmin)、admin ユーザーグループ (デフォルトは adm) に関する、スクリプトからの質問に答える必要があります。それらの質問に対して、インストールの計画作成段階で行った選択内容を入力します (30 ページの「インストール計画を作成する」の節を参照)。

a. コマンドプロンプトで次のコマンドを入力し、表示されるスクリプトからの質問に答えます。

```
# cd <CDx-ROM マウントポイント>/Sun_Grid_Engine_Enterprise_5.3/Packages
# pkgadd -d . SDRMEcomm
# pkgadd -d . SDRMEdoc
# pkgadd -d . SDRMEsp32 (省略可能。ただし、少なくとも 1 つのバイナリセットが必要)。
# pkgadd -d . SDRMEsp64 (省略可能。ただし、少なくとも 1 つのバイナリセットが必要)。
```

これらのコマンドによって、次のパッケージがインストールされます。

- SDRMEcomm - アーキテクチャ独立ファイル
- SDRMEdoc - マニュアル類
- SDRMEsp32 - Solaris 2.6/7/8/9 オペレーティング環境用の Solaris (SPARC® プラットフォーム) 32 ビットバイナリ
- SDRMEsp64 - Solaris 7/8/9 オペレーティング環境用の Solaris (SPARC プラットフォーム) 64 ビットバイナリ

tar を使用する場合

b. コマンドプロンプトで次のコマンドを入力します (この例の *<tardir>* は *<CD-ROM* マウントポイント*>*/Sun_Grid_Engine_Enterprise_5.3/tar ディレクトリの略記)。

```
# cd <sge_root>
# gzip -dc <tar_dir>/sgeee-5_3-common.tar.gz | tar xvpf -
# gzip -dc <tardir>/sgeee-5_3-doc | tar xvpf -
# gzip -dc <tardir>/sgeee-5_3-bin-solsparc32.tar.gz | tar xvpf -
# gzip -dc <tardir>/sgeee-5_3-bin-solsparc64.tar.gz | tar xvpf -
# util/setfileperm.sh <admin ユーザー> <admin グループ> <sge_root>
```

- solsparc32 tar ファイルには、Solaris 2.6/7/8/9 オペレーティング環境用の Solaris (SPARC® プラットフォーム) 32 ビットバイナリが含まれています。
- solsparc64 tar ファイルには、Solaris 7/8/9 オペレーティング環境用の Solaris (SPARC プラットフォーム) 64 ビットバイナリが含まれています。

5. コマンドプロンプトから以下を実行します。

```
% cd <install_dir>
% tar -xvpf distribution_source
```

<install_dir> はインストールディレクトリのパス名、*distribution_source* は CD-ROM 上のテープアーカイブファイル名です。これで、Sun Grid Engine, Enterprise Edition インストールキットが読み込まれます。

基本インストールの手順

ここでは、Sun Grid Engine, Enterprise Edition 5.3 システムのマスター、実行、管理、実行依頼ホストなどのすべてのコンポーネントをインストールする方法を説明します。

注 – セキュリティを強化したシステムをインストールする場合は、インストールに進む前に35 ページの「セキュリティを強化するインストールの手順」を参照してください。

▼ マスターホストをインストールする

注 – Sun Grid Engine, Enterprise Edition のインストールでは、そのインストールが実行されているシステムに合わせてデフォルトの構成が作成されます。インストールのホストになっているオペレーティングシステムが調べられ、その情報に基づいて意味のある設定が行われます。

---

1. `root` でマスターホストにログインします。
2. インストールキットが存在するディレクトリがマスターホストから見えるかどうかに従って、以下のいずれかを行います。
   a. インストールキットが存在するディレクトリがマスターホストから見える場合は、インストールディレクトリに移動 (`cd`) して、手順 3 に進みます。
   b. ディレクトリが見えず、見えるようにすることもできない場合は、以下の操作を行います。
      i. マスターホスト上にローカルのインストールディレクトリを作成します。
      ii. `ftp` または `rcp` などの適切なツールを使用してネットワークからローカルのインストールディレクトリにインストールキットをコピーします。
      iii. ローカルのインストールディレクトリに移動 (`cd`) します。
3. 以下の命令を実行します。

注 – CSP (Certificate Security Protocol) を使用した方法でインストールを行う場合は、次のコマンドに `-csp` フラグを追加する必要があります (36 ページの「CSP 保護されたシステムをインストールして設定する」を参照)。

```
% ./install_qmaster
```

これで、マスターのインストールが開始されます。いくつかの質問があり、管理操作の実行が求められることがあります。これらの質問と要求操作の内容は、メッセージを読めばわかるようになっています。

注 – 管理操作を実行するにあたっては、もう 1 つの端末セッションを開いていた方が便利です。

マスターのインストールでは、sge_qmaster と sge_schedd が必要とする適切なディレクトリ階層が作成されます。マスターホスト上で Sun Grid Engine, Enterprise Edition コンポーネントの sge_commd と sge_qmaster and sge_schedd が起動されます。また、マスターホストは、管理および実行依頼権限を持つホストとして登録されます。

何か問題があると思われる場合は、いつでもインストールを中止して、やり直すことができます。

▼ 実行ホストをインストールする

1. root で実行ホストにログインします。
2. マスターのインストール同様、ローカルのインストールディレクトリにインストールキットをコピーするか、ネットワーク上のインストールディレクトリを使用します。
3. インストールディレクトリに移動 (cd) して、次のコマンドを実行します。

注 – CSP (Certificate Security Protocol) を使用した方法でインストールを行う場合は、次のコマンドに -csp フラグを追加する必要があります (36 ページの「CSP 保護されたシステムをインストールして設定する」を参照)。

```
% ./install_execd
```

これで、実行ホストのインストールが開始されます。実行ホストのインストールの動作と処理は、マスターホストのときと非常によく似ています。

4. インストールスクリプトからのプロンプトに応答します。

注 – マスターホストはジョブの実行にも使用できます。このため、マスターマシンに実行ホストのインストールを行えばよいだけです。マスターホストとして非常に低速のマシンを使用するか、クラスタがかなり大規模な場合は、マスターマシンをマスター専用にすることを推奨します。

実行ホストのインストールでは、sge_execd が必要とする適切なディレクトリ階層が作成されます。実行ホスト上で Sun Grid Engine, Enterprise Edition コンポーネントの sge_commd と sge_execd が起動されます。

▼ 管理ホストと実行依頼ホストをインストールする

マスターホストには、管理業務の実施とジョブの実行依頼・監視・削除権限が暗黙で付与されます。このため、管理または実行依頼ホストとしての追加インストールを行う必要はありません。これに対し、純粋な管理ホストあるいは実行依頼ホストは登録が必要になります。

- **管理ホスト (たとえばマスターホスト) から管理アカウント (たとえばスーパーユーザーアカウント) を使用して、次のコマンドを入力します。**

```
% qconf -ah admin_host_name[,...]
% qconf -as submit_host_name[,...]
```

admin_host_name は管理ホストの名前です。

各種のホストの構成についての詳細と意味は、147 ページの「デーモンとホスト」の節を参照してください。

セキュリティを強化するインストールの手順

ここでは、インストールするシステムのセキュリティを強化する方法を説明します。この方法を使用することによって、CSP (Certificate Security Protocol) に基づく暗号化機能をシステムに持たせることができます。

このセキュリティ強化機能は、Sun Grid Engine 5.3 および Sun Grid Engine, Enterprise Edition 5.3 製品のどちらにも使用することができ、ここで紹介する方法は両方の製品に当てはまります。説明を簡潔にするため、説明では Sun Grid Engine 製品だけ取り上げます。

機能強化されたシステムでは、メッセージのテキストがそのまま転送されるのではなく、秘密鍵を使用して暗号化されます。秘密鍵は、公開 / 非公開鍵プロトコルを使用してやりとりされます。ユーザーは、Sun Grid Engine システムを通じて自分の身元証明書を提示し、Sun Grid Engine システムから証明書を受け取って、自身が適切なシステムと交信していることを確認します。この初期告知段階を通過すると、暗号化された形式で通信が透過的に続行されます。このセッションは一定期間有効で、その期間が終了すると、セッションを再告知する必要があります。

必要な追加設定

CSP (Certificate Security Protocol) 強化版の Sun Grid Engine システムを構築するための手順は、標準の設定手順に非常によく似ています。一般には、30 ページの「インストール計画を作成する」、30 ページの「配布媒体を読み込む」、33 ページの「マスターホストをインストールする」、34 ページの「実行ホストをインストールする」、35 ページの「管理ホストと実行依頼ホストをインストールする」の手順に従ってください。

それらの作業のほかに、CSP 強化版を構築するために以下の追加作業が必要になります。

- マスターホスト上での認証局 (CA) システムキーと証明書の生成。

  この生成は、`-csp` フラグを指定してインストールスクリプトを呼び出すことによって行われます。

- 実行および実行依頼ホストへのシステムキーと証明書の配付。

  この配付を安全な方法で行うのはシステム管理者の仕事です。すなわち、`ssh` などを使用した保護された方法で実行ホストと実行依頼ホストにキーを送信する必要があります。

- ユーザーキーと証明書の生成。

  これは、マスターインストールの完了後にシステム管理者が自動的に行うことができます。

- システム管理者による新規ユーザーの許可

▼ CSP 保護されたシステムをインストールして設定する

1. **21 ページの「基本インストールの概要」、45 ページの「インストール計画の作成」、32 ページの「基本インストールの手順」の節の説明に従って Sun Grid Engine システムをインストールします。ただし、インストールスクリプトを起動する際は、追加フラグの `-csp` を使用します。**

   たとえば `./install_qmaster` を入力することによってマスターホストの基本インストールするときに、インストール命令に `-csp` フラグを追加します。つまり、CSP 保護されたシステムをインストールするには、マスターホストをインストールするためのコマンドを以下のように変更して入力します。

```
% ./install_qmaster -csp
```

2. インストールスクリプトからのプロンプトに応答します。

CSP 証明書とキーを生成するには、次の情報が必要です。

- 英字 2 字からなる国別コード (たとえば米国ならば US)
- 州
- 所在地 (都市など)
- 組織
- 組織単位
- CA 電子メールアドレス

インストールを行うと、認証局が作成されます。また、マスターホストに Sun Grid Engine 専用の CA が作成されます。セキュリティ関連情報を含むディレクトリは以下のとおりです。

- `$SGE_ROOT/`*{default | $SGE_CELL}*`/common/sgeCA` - 一般にアクセス可能な CA および デーモン証明書が含まれます。
- `/var/sgeCA/`*{sge_service | port$COMM_PORT}/{default | $SGE_CELL}*`/private` - 対応する非公開鍵が含まれます。
- `/var/sgeCA/`*{sge_service | port$COMM_PORT}/{default | $SGE_CELL}*`/userkeys/$USER` - ユーザーキーと証明書がが含まれます。

ディレクトリの作成中、スクリプトからの出力はコード例 2-1 に示すようになります。

コード例 2-1 CSP インストールスクリプト - ディレクトリの作成

```
Initializing Certificate Authority (CA) for OpenSSL security framework
------------------------------------------------------------------------

Creating /scratch2/eddy/sge_sec/default/common/sgeCA
Creating /var/sgeCA/port6789/default
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/certs
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/crl
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/newcerts
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/serial
Creating /scratch2/eddy/sge_sec/default/common/sgeCA/index.txt
Creating /var/sgeCA/port6789/default/userkeys
Creating /var/sgeCA/port6789/default/private
Hit Return to continue >>
```

ディレクトリが作成されると、続いて CA 専用の証明書と非公開鍵が生成されます。Sun Grid Engine システムは、特殊なファイルに含まれている疑似ランダムデータ、または `/dev/random` (存在する場合) を使用して、疑似乱数ジェネレータ (PRNG) をシードします。(乱数についての詳細は、`http://www.openssl.org/support/faq.html` および `http://www.cosy.sbg.ac.at/~andi` を参照してください。)

CA インフラストラクチャがインストールされると、CA によって admin ユーザーと疑似デーモンユーザー、ユーザー root 用にアプリケーション証明書とユーザー証明書、非公開鍵が作成、署名されます。このときスクリプトからは、まずサイト情報の質問があり、コード例 2-2 に示すよう出力が表示されます。

コード例 2-2 CSP インストールスクリプト - 情報収集

```
Creating CA certificate and private key
----------------------------------------

Please give some basic parameters to create the distinguished name (DN)
for the certificates.

We will ask for

     - the two letter country code
     - the state
     - the location, e.g city or your buildingcode
     - the organization (e.g. your company name)
     - the organizational unit, e.g. your department
     - the email address of the CA administrator (you!)

Hit Return to continue >>

Please enter your two letter country code, e.g. >US< >> DE
Please enter your state >> Bavaria
Please enter your location, e.g city or buildingcode >> Regensburg
Please enter the name of your organization >> Myorg
Please enter your organizational unit, e.g. your department >> Mydept
Please enter the email address of the CA administrator >> admin@my.org


You selected the following basic data for the distinguished name of
your certificates:

Country code:         C=DE
State:                ST=Bavaria
Location:             L=Regensburg
Organization:         O=Myorg
Organizational unit:  OU=Mydept
CA email address:     emailAddress=admin@my.org


Do you want to use these data (y/n) [y] >>
```

入力した情報に間違いがないことを確認すると、CA インフラストラクチャの作成が始まり、CA 証明書と非公開鍵が生成されます。このときのスクリプトからの出力は、コード例 2-3 のようになります。

.

コード例 2-3　CSP インストールスクリプト - CA インフラストラクチャの作成

```
Creating RANDFILE from >/kernel/genunix< in
>/var/sgeCA/port6789/default/private/rand.seed<

1513428 semi-random bytes loaded
Creating CA certificate and private key

Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.................++++++
writing new private key to '/var/sgeCA/port6789/default/private/cakey.pem'
-----
Hit Return to continue >>
```

CA インフラストラクチャがインストールされると、CA によって 疑似デーモンユーザーと root ユーザー用のアプリケーション証明書とユーザー証明書、非公開鍵が作成され、署名されます。このときのスクリプトからの出力は、コード例 2-4(複数ページにまたがる) のようになります。この例では、1 行に収まるよう一部の行を短縮しています。短縮している箇所は省略符号 (...) で示しています。

コード例 2-4　CSP インストールスクリプト - 証明書と非公開鍵の作成

```
Creating Daemon certificate and key
-----------------------------------

Creating RANDFILE from >/kernel/genunix< in >/var/sgeCA/(...)/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
..............++++++
.................++++++
writing new private key to '/var/sgeCA/port6789/default/private/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
```

コード例 2-4　　CSP インストールスクリプト - 証明書と非公開鍵の作成 (続き)

```
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'Bavaria'
localityName          :PRINTABLE:'Regensburg'
organizationName      :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'root'
commonName            :PRINTABLE:'SGE Daemon'
emailAddress          :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:57 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for SGE daemons
Creating RANDFILE from >/kernel/genunix< in>/var/(...)/userkeys/root/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
............++++++
.................++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/root/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'Bavaria'
localityName          :PRINTABLE:'Regensburg'
organizationName      :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'root'
commonName            :PRINTABLE:'SGE install user'
emailAddress          :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:59 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >root< in >/var/(...)/userkeys/root<
Creating RANDFILE from >/kernel/genunix< in >/(...)/userkeys/eddy/rand.seed<
1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
............++++++
.....................................................++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/eddy/key.pem'
-----
```

コード例 2-4　　CSP インストールスクリプト - 証明書と非公開鍵の作成 (続き)

```
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'Bavaria'
localityName          :PRINTABLE:'Regensburg'
organizationName      :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'root'
commonName            :PRINTABLE:'SGE install user'
emailAddress          :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:59 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >root< in >/var/(...)/userkeys/root<
Creating RANDFILE from >/kernel/genunix< in >/var/(...)/userkeys/eddy/rand.seed<

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.............++++++
.....................................................++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/eddy/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'Bavaria'
localityName          :PRINTABLE:'Regensburg'
organizationName      :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'eddy'
commonName            :PRINTABLE:'SGE admin user'
emailAddress          :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:51:02 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >root< in >/var/(...)/userkeys/root<
Creating RANDFILE from >/kernel/genunix< in >/var/(...)/userkeys/eddy/rand.seed<
```

コード例 2-4　　CSP インストールスクリプト - 証明書と非公開鍵の作成 (続き)

```
1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.............++++++
.....................................................++++++
writing new private key to '/var/sgeCA/port6789/default/userkeys/eddy/key.pem'
-----
Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'Bavaria'
localityName          :PRINTABLE:'Regensburg'
organizationName      :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier      :PRINTABLE:'eddy'
commonName            :PRINTABLE:'SGE admin user'
emailAddress          :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:51:02 2003 GMT (365 days

Write out database with 1 new entries
Data Base Updated
created and signed certificate for user >eddy< in >/var/(...)/userkeys/eddy<
Hit Return to continue >>
```

マスターホスト sge_qmaster のセキュリティ関連の設定が完了すると、スクリプトからインストールの続行を促す、コード例 2-5 に示すようなメッセージが表示されます。

コード例 2-5　　CSP インストールスクリプト - インストールの続行

```
SGEEE startup script
--------------------


Your system wide SGEEE startup script is installed as:

     "/scratch2/eddy/sge_sec/default/common/rcsge"

Hit Return to continue >>
```

3. 以下のいずれかを行います。

   a. 実行デーモンがアクセス可能で、CSP セキュリティ情報を保存する場所として共有ファイルシステムが安全ではないと思われる場合は、手順 4 に進みます。

   b. 共有ファイルシステムが安全であると思われる場合は、34 ページの「実行ホストをインストールする」の節の説明に従って基本インストールを続けます。

      実行ホストのインストールで「`./install_execd`」スクリプトを呼び出すときに `-csp` フラグを付けることを忘れないでください。

      残りのインストール手順がすべて完了したら、67 ページの「ユーザー用の証明書と非公開鍵を生成する」の節に進んでください。

4. (省略可能) 実行デーモンがアクセス可能で、CSP セキュリティ情報を保存する場所として共有ファイルシステムが安全ではない場合は、デーモンの非公開鍵とランダムファイルを含むディレクトリを実行ホストに転送する必要があります。

   a. マスターホストで `root` になり、次のコマンドを入力することによって、実行ホストとして設定するマシンに非公開鍵をコピーする準備をします。

```
# umask 077
# cd /
# tar cvpf /var/sgeCA/port6789.tar /var/sgeCA/port6789/default
```

   b. 実行ホストで `root` になり、次のコマンドを入力することによってファイルをコピーします。すべての実行ホストで、この操作を繰り返してください。

```
# umask 077
# cd /
# scp <マスターホスト>:/var/sgeCA/port6789.tar .
# umask 022
# tar xvpf /port6789.tar
# rm /port6789.tar
```

   c. 次のコマンドを入力することによってファイル権限を確認します。

```
# ls -lR /var/sgeCA/port6789/
```

このときの出力は、コード例 2-6 のようになります。

コード例 2-6　　ファイル権限の確認

```
/var/sgeCA/port6789/:
total 2
drwxr-xr-x   4 eddy     other        512 Mar  6 10:52 default
/var/sgeCA/port6789/default:
total 4
drwx------   2 eddy     staff        512 Mar  6 10:53 private
drwxr-xr-x   4 eddy     staff        512 Mar  6 10:54 userkeys
/var/sgeCA/port6789/default/private:
total 8
-rw-------   1 eddy     staff        887 Mar  6 10:53 cakey.pem
-rw-------   1 eddy     staff        887 Mar  6 10:53 key.pem
-rw-------   1 eddy     staff       1024 Mar  6 10:54 rand.seed
-rw-------   1 eddy     staff        761 Mar  6 10:53 req.pem
/var/sgeCA/port6789/default/userkeys:
total 4
dr-x------   2 eddy     staff        512 Mar  6 10:54 eddy
dr-x------   2 root     staff        512 Mar  6 10:54 root
/var/sgeCA/port6789/default/userkeys/eddy:
total 16
-r--------   1 eddy     staff       3811 Mar  6 10:54 cert.pem
-r--------   1 eddy     staff        887 Mar  6 10:54 key.pem
-r--------   1 eddy     staff       2048 Mar  6 10:54 rand.seed
-r--------   1 eddy     staff        769 Mar  6 10:54 req.pem
/var/sgeCA/port6789/default/userkeys/root:
total 16
-r--------   1 root     staff       3805 Mar  6 10:54 cert.pem
-r--------   1 root     staff        887 Mar  6 10:54 key.pem
-r--------   1 root     staff       2048 Mar  6 10:53 rand.seed
-r--------   1 root     staff        769 Mar  6 10:54 req.pem
```

d. 次のコマンドを入力することによって、Sun Grid Engine のインストールを続行します。

```
# cd $SGE_ROOT
# ./install_execd -csp
```

e. 34 ページの「実行ホストをインストールする」の節の手順 4 からのインストール手順に従って操作を進めます。

残りのインストール手順がすべて完了したら、45 ページの「ユーザー用の証明書と非公開鍵を生成する」の節に進んでください。

▼ ユーザー用の証明書と非公開鍵を生成する

ユーザーが CSP 保護されたシステムを使用するには、それぞれのユーザー専用の証明書と非公開鍵にアクセスできる必要があります。このための最も便利な方法は、ユーザーの識別情報を含むテキストファイルを作成することです。

1. **ユーザーの識別情報を含むテキストファイルを作成して、保存します。**

   次の例 (`myusers.txt`) に示す形式でファイルを作成してください。(ファイルのフィールドは、「*UNIX_username:Gecos_field:email_address*」の形式です。

```
eddy:Eddy Smith:eddy@my.org
sarah:Sarah Miller:sarah@my.org
leo:Leo Lion:leo@my.org
```

2. **マスターホストで root になり、次のコマンドを入力します。**

```
# $SGE_ROOT/util/sgeCA/sge_ca -usercert myusers.txt
```

3. **次のコマンドを入力することによって確認します。**

```
# ls -l /var/sgeCA/port6789/default/userkeys
```

以下の例に示すようなディレクトリリストが表示されます。

```
dr-x------   2 eddy  staff        512 Mar  5 16:13 eddy
dr-x------   2 sarah staff        512 Mar  5 16:13 sarah
dr-x------   2 leo   staff        512 Mar  5 16:13 leo
```

4. ファイル (この例では myusers.txt) に登録した各ユーザーに、次のコマンドを入力することによって各自の $HOME/.sge ディレクトリにセキュリティ関連のファイルをインストールするよう指示します。

```
% source $SGE_ROOT/default/common/settings.csh
% $SGE_ROOT/util/sgeCA/sge_ca -copy
```

各ユーザーに次のような情報が返されます (この例ではユーザーは eddy)。

```
Certificate and private key for user eddy have been installed
```

Sun Grid Engine がインストールされたあらゆる場所で、対応する COMMD_PORT 番号用のサブディレクトリがインストールされます。以下は、myusers.txt ファイルの場合のディレクトリリストの出力例です。

```
% ls -lR $HOME/.sge
/home/eddy/.sge:
total 2
drwxr-xr-x   3 eddy staff        512 Mar  5 16:20 port6789

/home/eddy/.sge/port6789:
total 2
drwxr-xr-x   4 eddy staff        512 Mar  5 16:20 default

/home/eddy/.sge/port6789/default:
total 4
drwxr-xr-x   2 eddy staff        512 Mar  5 16:20 certs
drwx------   2 eddy staff        512 Mar  5 16:20 private

/home/eddy/.sge/port6789/default/certs:
total 8
-r--r--r--   1 eddy staff       3859 Mar  5 16:20 cert.pem

/home/eddy/.sge/port6789/default/private:
total 6
-r--------   1 eddy staff        887 Mar  5 16:20 key.pem
-r--------   1 eddy staff       2048 Mar  5 16:20 rand.seed
```

▼ 証明書を確認する

- 何を確認するかによって、使用するコマンドは異なります。

証明書の表示

次のコマンドを 1 行で入力します (このマニュアルでは 1 行に収まらないため 2 行に分けていますが、実際には 1 行です。-in と ~/.sge の間には空白文字を 1 つ挿入します)。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -in
~/.sge/port6789/default/certs/cert.pem -text
```

発行者の確認

次のコマンドを 1 行で入力します (このマニュアルでは 1 行に収まらないため 2 行に分けていますが、実際には 1 行です。-in と ~/.sge の間には空白文字を 1 つ挿入します)。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -issuer -in
~/.sge/port6789/default/certs/cert.pem -noout
```

サブジェクトの確認

次のコマンドを 1 行で入力します (このマニュアルでは 1 行に収まらないため 2 行に分けていますが、実際には 1 行です。-in と ~/.sge の間には空白文字を 1 つ挿入します)。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -subject -in
~/.sge/port6789/default/certs/cert.pem -noout
```

証明書の電子メールの確認

次のコマンドを 1 行で入力します (このマニュアルでは 1 行に収まらないため 2 行に分けていますが、実際には 1 行です。-in と ~/.sge の間には空白文字を 1 つ挿入します)。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -email -in
~/.sge/default/port6789/certs/cert.pem -noout
```

有効期間の確認

次のコマンドを 1 行で入力します (このマニュアルでは 1 行に収まらないため 2 行に分けていますが、実際には 1 行です。-in と ~/.sge の間には空白文字を 1 つ挿入します)。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -dates -in
~/.sge/default/port6789/certs/cert.pem -noout
```

フィンガープリントの確認

次のコマンドを 1 行で入力します (このマニュアルでは 1 行に収まらないため 2 行に分けていますが、実際には 1 行です。-in と ~/.sge の間には空白文字を 1 つ挿入します)。

```
% $SGE_ROOT/utilbin/$ARCH/openssl x509 -fingerprint -in
~/.sge/port6789/default/certs/cert.pem -noout
```

インストールの検証

Sun Grid Engine, Enterprise Edition デーモンが動作していることを確認するには、マスターホスト上、続いて実行ストに sge_qmaster、sge_schedd、sge_commd デーモンが存在することを確認する必要があります。この確認を行ったら、Sun Grid Engine, Enterprise Edition 5.3 のコマンドを使用し、最後にジョブの実行依頼の準備をします。

▼ インストールが正しく行われたことを確認する

マスターホストでの確認

1. マスターホストにログインします。
2. 使用しているオペレーティングシステムに従って、以下のいずれか適切なコマンドを実行します。

   a. BSD 版 UNIX システムの場合は、次のコマンドを入力します。

   ```
   % ps -ax
   ```

   b. Solaris オペレーティング環境などの UNIX System 5 版オペレーティングシステムの場合は、次のコマンドを入力します。

   ```
   % ps -ef
   ```

3. 次の例に示すような sge 文字列が出力に含まれているかどうかを確認します。

   BSD 版 UNIX システムの場合は、以下のような文字列です。

   ```
   14673 p1 S <  2:12 /gridware/sge/bin/solaris/sge_commd
   14676 p1 S <  4:47 /gridware/sge/bin/solaris/sge_qmaster
   14678 p1 S <  9:22 /gridware/sge/bin/solaris/sge_schedd
   ```

UNIX System 5 版のシステムの場合は、以下のような文字列です。

```
root 439 1 0 Jun 2 ? 3:37 /gridware/sge/bin/solaris/sge_commd
root 439 1 0 Jun 2 ? 3:37 /gridware/sge/bin/solaris/sge_qmaster
root 446 1 0 Jun 2 ? 3:37 /gridware/sge/bin/solaris/sge_schedd
```

こうした文字列が見つからない場合は、マスターホストで必要な Sun Grid Engine, Enterprise Edition デーモンがそのマシンで動作していないことが考えられます (本当にマスターホストにいるかどうかは、*<sge_root>*/*<cell>*/common/act_qmaster の内容を見ると判ります)。次の手順に進みます。

4. **(省略可能) 手動でデーモンを再起動します。**

   次に行う作業については、147 ページの「デーモンとホスト」の節を参照してください。

実行ホストでの確認

1. **Sun Grid Engine, Enterprise Edition の実行ホストのインストールを行った実行ホストにログインします。**
2. **マスターホストのときと同様、使用しているシステムに従って適切な `ps` コマンドを入力します。**
3. **出力に `sge` 文字列が含まれているかどうかを確認します。**

   BSD 版 UNIX システムの場合は、以下のような文字列です。

```
14685 p1 S <    1:13 /gridware/sge/bin//sge_commd
14688 p1 S <    4:27 /gridware/sge/bin/solaris/sge_execd
```

   Solaris オペレーティング環境などの UNIX System 5 版のシステムの場合は、以下のような文字列です。

```
root 169 1 0 Jun 22 ? 2:04 /gridware/sge/bin/solaris/sge_commd
root 171 1 0 Jun 22 ? 7:11 /gridware/sge/bin/solaris/sge_execd
```

   こうした `sge` 文字列が見つからない場合は、実行ホストに必要なデーモンが動作していないことが考えられます。次の手順に進みます。

4. **(省略可能) 手動でデーモンを再起動します。**

   次に行う作業については、147 ページの「デーモンとホスト」の節を参照してください。

コマンドの実行テスト

必要なデーモンがマスターホストと実行ホストで動作していれば、Sun Grid Engine, Enterprise Edition は運用可能な状態になっています。試験的なコマンドを発行することによって、このことを確認してください。

1. **マスターホストまたは他の管理ホストのいずれかにログインします。**

   Sun Grid Engine, Enterprise Edition のバイナリをインストールしたパスを標準の検索パスに含めることを忘れないでください。

2. **コマンド行から次のコマンドを入力します。**

```
% qconf -sconf
```

   この `qconf` コマンドは、現在のグローバルクラス構成を表示します (184 ページの「基本クラスタ構成」を参照)。このコマンドで問題が発生した場合、たいていその原因は、`SGE_ROOT` 環境変数が正しく設定されていないか、`qconf` が `sge_qmaster` に関連付けられている `sge_commd` と通信できなかったことにあります。次の手順に進んでください。

3. **スクリプトファイル** *<sge_root>*`/`*<cell>*`/common/settings.csh` **または** *<sge_root>*`/`*<cell>*`/common/settings.sh` **に環境変数** `COMMD_PORT` **が設定されているかどうかを確認します。**

   設定されている場合、上記のコマンドを再度試してみる前に、`COMMD_PORT` 環境変数に適切な値が設定されていることを確認します。settings ファイルで `COMMD_PORT` 変数が使用されていない場合は、コマンドを実行したマシンの services データベース (`/etc/services` または NIS services マップ) が `sge_commd` エントリを供給する必要があります。そうなっていない場合は、マシンの services データベースにそのようなエントリを追加して、Sun Grid Engine, Enterprise Edition マスターホストに設定されているのと同じ値を設定し、次の手順に進みます。

4. `qconf` **コマンドを再実行します。**

ジョブの実行依頼の準備

Sun Grid Engine, Enterprise Edition システムにバッチスクリプトを実行依頼する前に、サイトの標準および個人シェルリソースファイル (`.cshrc`、`.profile`、`.kshrc` のいずれか) に `stty` などのコマンドが含まれているかどうかを調べます。デフォルトでは、バッチジョブには端末接続はありません。このため、`stty` を呼び出そうとするとエラーになります。

1. マスターホストにログインします。
2. 次のコマンドを入力します。

```
% rsh an_exec_host date
```

an_exec_host は、使用するインストール済みの実行ホストです。すべての実行ホストで、ログインまたはホームディレクトリがホストごとに異なることを確認してください。rsh コマンドは、マスターホストでローカルに実行した date コマンドに非常によく似た出力を生成します。通常の行のほかにエラーメッセージを含む行が返された場合、バッチジョブを実行する前に、そのエラーの原因を取り除いておく必要があります。

どのコマンドインタプリタでも、stty などのコマンドを実行する前に、実際の端末接続を調べることができます。以下は、Bourne/Korn シェル用のスクリプト例です。

```
tty -s
if [ $? = 0 ]; then
   stty erase ^H
fi
```

C シェルの構文も非常によく似ています。

```
tty -s
if ( $status = 0 ) then
   stty erase ^H
endif
```

3. <*sge_root*>/examples/jobs ディレクトリに含まれているサンプルスクリプトをどれか実行依頼します。

このためには、次のコマンドを入力します。

```
% qsub script_path
```

4. Sun Grid Engine, Enterprise Edition の qstat コマンドを使用して、ジョブの動作を監視します。

バッチジョブの実行依頼と監視についての詳細は、97 ページの「バッチジョブの実行依頼」を参照してください。

5. ジョブの実行が完了したら、自分のホームディレクトリ内にリダイレクトされた stdout/stderr ファイルの <スクリプト名>.e<*job_id*> および <スクリプト名>.o<*job_id*> がないか調べます。<*job_id*> は、各ジョブに割り当てられた連続する固有の整数番号です。

   問題が発生した場合、第 11 章、303 ページの「エラーの通知と障害追跡」を参照してください。

PART III Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアの使用方法

この PART III は、主としてユーザー、すなわち、システム管理者の仕事 (143 ページの「管理」を参照) を行うことのないユーザーを対象に、以下の 3 つの章で構成されています。

- 第 3 章 - 57 ページの「Sun Grid Engine, Enterprise Edition の概要」

  この章では、Sun Grid Engine, Enterprise Edition の基礎とともに、さまざまな資源の一覧表示方法を説明します。

- 第 4 章 - 71 ページの「ジョブの実行依頼」

  この章では、Sun Grid Engine, Enterprise Edition システムを使用してジョブの実行依頼を行う方法を詳細に説明します。最初に「練習」ジョブの実行依頼をして、手順を習得してください。

- 第 5 章 - 115 ページの「チェックポイントジョブとジョブの監視、制御」

  この章では、ジョブ制御の概念とともに、さまざまジョブ制御を行う方法を説明します。

PART III の各章には、Sun Grid Engine, Enterprise Edition システムを使用して多数の作業を行う方法に関する予備知識的な情報とともに、その詳細な説明も含まれています。

第3章

Sun Grid Engine, Enterprise Edition の概要

この章では、Sun Grid Engine, Enterprise Edition 5.3 を使い始めるにあたって理解しておくと役立つと思われる基礎的な概念と用語を説明します。総合的な用語集をはじめとする、この製品に関する予備的な情報については、第 1 章、1 ページの「Sun™ Grid Engine, Enterprise Edition 5.3 入門」をお読みください。

この章ではまた、以下の作業を行う方法も説明します。

Sun Grid Engine, Enterprise Edition ユーザーの種類

Sun Grid Engine, Enterprise Edition のユーザーは次の 4 つのカテゴリに分類されます。

- **マネージャー** - Sun Grid Engine, Enterprise Edition の運用に関する全権を持つユーザーです。デフォルトでは、各管理ホストのスーパーユーザーはマネージャー特権を持ちます。
- **オペレーター**は、キューの追加・削除・変更などの構成変更以外の、マネージャーが実行できるコマンドの多くを実行できるユーザーです。

- **所有者** - キュー所有者は、所有しているキューやそのキュー内のジョブを一時停止したり、使用可能にしたりできます。それ以上の管理権限はありません。
- **ユーザー** - ユーザーは 68 ページの「ユーザーのアクセス権」で説明しているようないくつかのアクセス権を持ちますが、クラスタやキューの管理を行うことはできません。

表 3-1 は、これらのカテゴリのユーザーが使用できる Sun Grid Engine, Enterprise Edition コマンド機能をまとめています。

表 3-1 ユーザーカテゴリと使用できるコマンド機能

コマンド	マネージャー	オペレータ	所有者	ユーザー
qacct	全機能	全機能	所有ジョブのみ	所有ジョブのみ
qalter	全機能	全機能	所有ジョブのみ	所有ジョブのみ
qconf	全機能	システム構成の変更不可	構成とアクセス権の表示のみ	構成とアクセス権の表示のみ
qdel	全機能	全機能	所有ジョブのみ	所有ジョブのみ
qhold	全機能	全機能	所有ジョブのみ	所有ジョブのみ
qhost	全機能	全機能	全機能	全機能
qlogin	全機能	全機能	全機能	全機能
qmod	全機能	全機能	所有ジョブと所有キューのみ	所有ジョブのみ
qmon	全機能	システム構成の変更不可	構成の変更不可	構成の変更不可
qrexec	全機能	全機能	全機能	全機能
qselect	全機能	全機能	全機能	全機能
qsh	全機能	全機能	全機能	全機能
qstat	全機能	全機能	全機能	全機能
qsub	全機能	全機能	全機能	全機能

キューとキュープロパティ

Sun Grid Engine, Enterprise Edition システムを最大限に活用するには、キューの構成とシステムに設定されているキューのプロパティを理解しておく必要があります。

QMON ブラウザ

Sun Grid Engine, Enterprise Edition には、グラフィカルユーザーインタフェース (GUI) コマンドツールとして QMON ブラウザが用意されています。QMON ブラウザは、ジョブの実行依頼、ジョブ制御、重要情報収集などの、さまざまな Sun Grid Engine, Enterprise Edition の機能を提供します。

▼ QMON ブラウザを起動する

- **コマンド行から次のコマンドを入力します。**

```
% qmon
```

メッセージウィンドウが現れた後、次に示すような QMON メインコントロールパネルが表示されます (各アイコンの名前については、図 1-4 を参照)。

図 3-1 QMON メインコントロールメニュー

このマニュアルで説明する多くの手順で、QMON ブラウザを使用する必要があります。アイコンボタンの上にマウスポインタを置くと、その名前が表示されます。それぞれのボタンには、その働きを類推できる名前が付けられています。

QMON ブラウザはカスタマイズすることができます。カスタマイズ方法については、14 ページの「QMON のカスタマイズ」を参照してください。

「QMON キュー制御」ダイアログボックス

「QMON キュー制御」ダイアログボックス (137 ページの「QMON からキューを制御する」の節に表示例と説明があります) では、インストール済みのキューとその現在のステータスを簡単に確認することができます。

▼ キューのリストを表示する

- 次のコマンドを入力します。

```
% qconf -sql
```

▼ キューのプロパティを表示する

キューのプロパティは、QMON またはコマンド行のどちらからでも表示することができます。

QMON ブラウザを使用する場合

1. QMON メインメニューから「ブラウザ」のアイコンをクリックします。
2. 「キュー」ボタンをクリックします。
3. 「キュー制御」ダイアログボックスで適切なキューのアイコンの上にマウスポインタを置きます。

   図 3-2 は、この操作で表示されるキュープロパティ情報の画面例です。

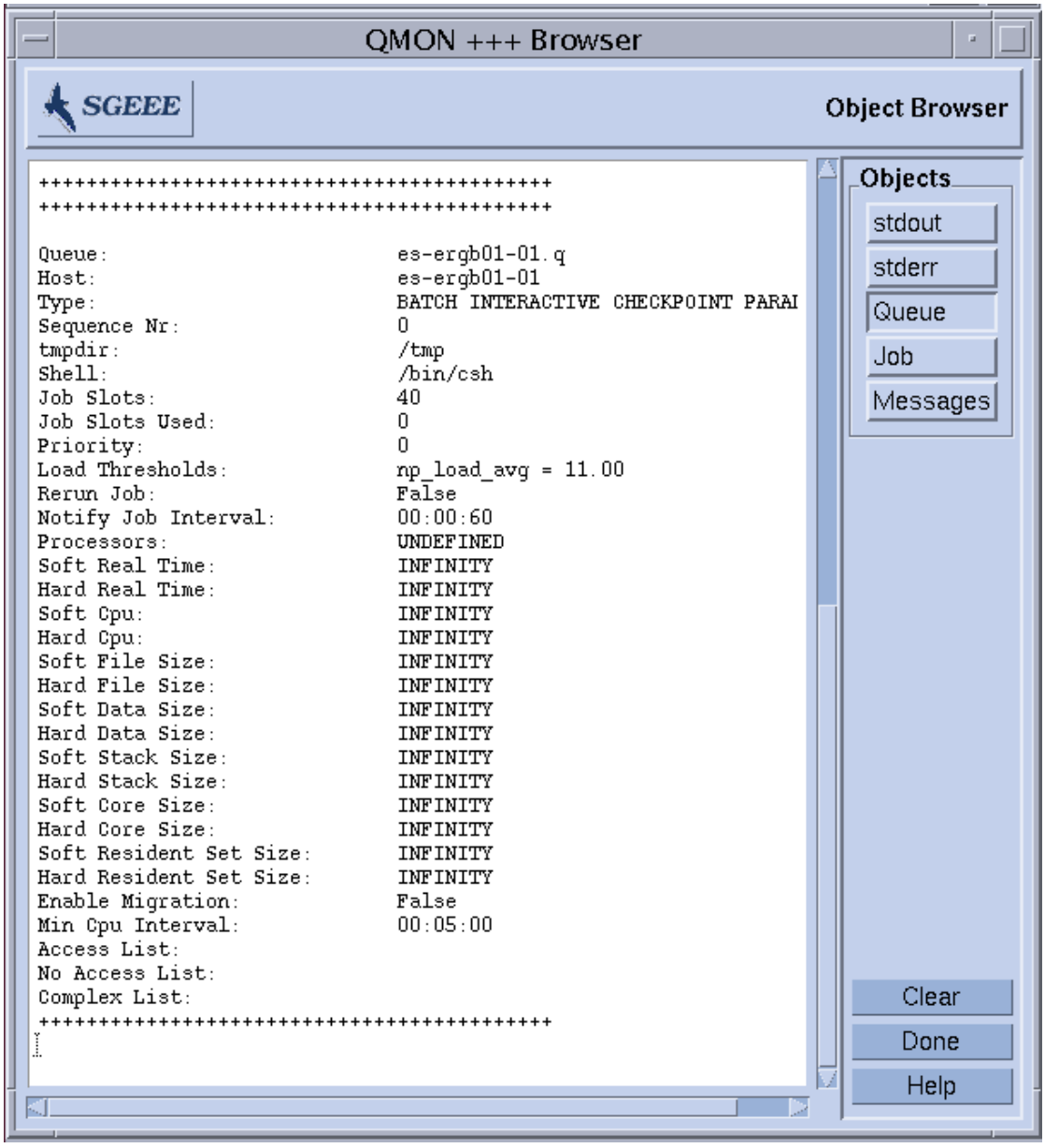


図 3-2　　QMON ブラウザのキュープロパティの表示例

コマンド行を使用する場合

- **次のコマンドを入力します。**

```
% qconf -sq queue_name
```

queue_name は、キューの名前です。

図 3-2 に示すような情報が表示されます。

キュープロパティの意味

キューの各種プロパティについての詳細は、queue_conf マニュアルページと『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の queue_conf の節の各プロパティの説明を参照してください。

ここでは、特に重要なパラメータをまとめています。

- qname - 要求されたキューの名前
- hostname - キューのホスト名
- processors - キューがアクセス可能な、マルチプロセッサシステムのプロセッサ。
- qtype - このキューで実行可能なジョブの種類。現在は、バッチ、対話形式、チェックポイント、並列のいずれかその任意の組み合わせ、または代替転送です。
- slots - キューで並行して実行可能なジョブ数
- owner_list - キューの所有者 (70 ページの「マネージャーとオペレータ、所有者」の節を参照)
- user_lists - ユーザーアクセスリストに登録されていて、ここに列挙されたユーザー / グループ識別名は、このキューを利用できます (68 ページの「ユーザーのアクセス権」を参照)。
- xuser_lists - ユーザーアクセスリストに登録されていて、ここに列挙されたユーザー / グループ識別名は、このキューを利用できません (68 ページの「ユーザーのアクセス権」を参照)。
- project_lists - ここに列挙されたプロジェクト識別名で実行依頼されたジョブはこのキューを利用できます (233 ページの「プロジェクト」を参照)。
- xproject_lists - ここに列挙されたプロジェクト識別名で実行依頼されたジョブはこのキューを利用できません (233 ページの「プロジェクト」を参照)。
- complex_list - このパラメータに列挙されている複合はこのキューに関連付けられていて、その複合に含まれる属性は、このキューに対する要求可能属性セットに影響します (64 ページの「要求可能属性」を参照)。

- complex_values - 複合属性に対してこのキューに提供されている能力値を割り当てます (64 ページの「要求可能属性」を参照)。

ホスト機能

QMON メインメニューの「ホスト構成」ボタンをクリックすると、Sun Grid Engine, Enterprise Edition クラスタ内のホストに関連付けられている機能の概要が表示されます。ただし、表示された構成に変更を加えられるのは、Sun Grid Engine, Enterprise Edition のマネージャー特権を持っているユーザーだけです。

ホスト構成用のダイアログについては、147 ページの「デーモンとホスト」で説明します。ここでは、コマンド行からこの種の情報を取り出すためのコマンドについて説明します。

▼ マスターホスト名を確認する

マスターホストは現在のマスターホストとシャドウマスターホストとの間で自由に切り替わることができるため、マスターホストの場所はユーザーには透過的です。

- **テキストエディタを使用して、<*sge_root*>/<*cell*>/common/act_qmaster ファイルを開きます。**

  現在のマスターホスト名は、このファイルに記録されています。

▼ 実行ホストのリストを表示する

クラスタ内で実行ホストとして構成されているホストのリストを表示するには、次のコマンドを使用します。

```
% qconf -sel
% qconf -se hostname
% qhost
```

hostname は、ホスト名です。

最初のコマンドは、現在実行ホストとして構成されているすべてのホストの名前を一覧表示します。2 つ目のコマンドは、指定された実行ホストに関する詳細情報を表示します。3 つ目のコマンドは、実行ホストに関するステータスおよび負荷情報を表示

します。qonf で表示される情報についての詳細は、host_conf のマニュアルページ、その出力と他のオプションについての詳細は、 qhost のマニュアルページを参照してください。

▼ 管理ホストのリストを表示する

管理権限を持つホストのリストは、次のコマンドで表示することができます。

```
% qconf -sh
```

▼ 実行依頼ホストのリストを表示する

実行依頼ホストのリストは、次のコマンドで表示することができます。

```
% qconf -ss
```

要求可能属性

Sun Grid Engine, Enterprise Edition ジョブの実行依頼では、ジョブの要求プロファイルを指定することができます。要求プロファイルに指定できるのは、ジョブが正しく動作するために必要なホストあるいはキューの属性すなわち特性です。Sun Grid Engine, Enterprise Edition はジョブ要求を Sun Grid Engine, Enterprise Edition クラスのホストおよびキュー構成と引き合わせ、ジョブに適したホストを見つけます。

ジョブの要求に指定できる属性は、Sun Grid Engine, Enterprise Edition クラスタ関連 (ネットワーク共有ディスクに必要な空き領域など)、ホスト関連 (オペレーティングシステムのアーキテクチャなど)、キュー関連 (使用できる CPU 時間など) に分けられ、その他一部のホストにしかインストールされていないソフトウェアなど、サイトのポリシーから得られる属性もあります。

このように、使用可能な属性には、キュープロパティリスト (58 ページの「キューとキュープロパティ」を参照)、グローバルおよびホスト関連の属性リスト (193 ページの「複合の種類」を参照) 、さらには、管理者定義の属性などがあります。しかし、便宜上、Sun Grid Engine, Enterprise Edition 管理者は一般に使用可能な属性のうちの一部だけを要求可能属性として定義します。

現在要求可能な属性は、「QMON 実行依頼」ダイアログボックスの「要求資源」サブダイアログボックス (図 3-3 を参照) に表示されます (ジョブの実行依頼方法についての詳細は、77 ページの「バッチジョブの実行依頼」の節を参照)。要求可能属性は、そのダイアログボックスの「使用可能な資源」選択リストに列挙されます。

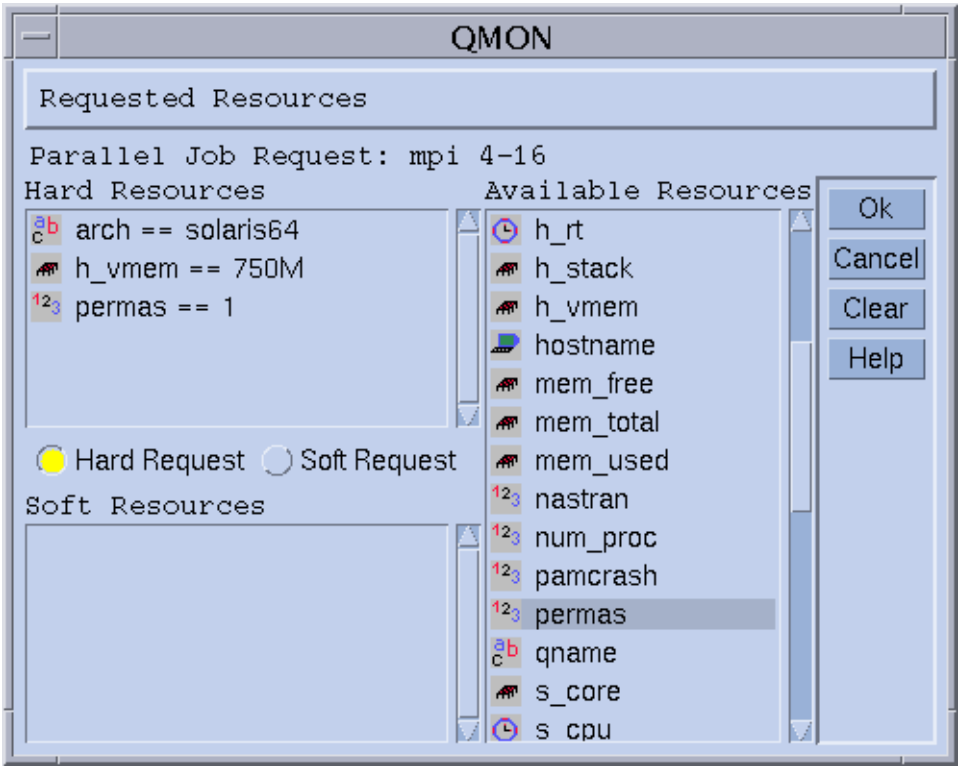


図 3-3　「要求資源」ダイアログボックス

▼ 要求可能属性のリストを表示する

1. **コマンド行から次のコマンドを入力することによって、構成済み複合リストを表示します。**

```
% qconf -scl
```

複合には、一群の属性の定義が含まれています。以下は、3 つの標準の複合です。

- `global` - クラスタ全体のグローバル属性 (省略可能)

- host - ホスト固有の属性
- queue - キュープロパティ属性

上記のコマンドでこれ以外の複合名が表示された場合は、管理者定義の複合であることを意味します (複合についての詳細は、このマニュアルの第 8 章、 191 ページの「複合の概念」、または『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の複合の形式の説明を参照)。

2. 特定の複合の属性を表示するには、次のコマンドを使用します。

```
% qconf -sc complex_name[,...]
```

表 3-2 は、queue 複合の出力例です。

表 3-2　queue 複合の属性表示の例

#Name	Shortcut	Type	Value	Relop	Requestable	Consumable	Default
qname	q	STRING	NONE	==	YES	NO	NONE
hostname	h	HOST	unknown	==	YES	NO	NONE
tmpdir	tmp	STRING	NONE	==	NO	NO	NONE
calendar	c	STRING	NONE	==	YES	NO	NONE
priority	pr	INT	0	>=	NO	NO	0
seq_no	seq	INT	0	==	NO	NO	0
rerun	re	INT	0	==	NO	NO	0
s_rt	s_rt	TIME	0:0:0	<=	NO	NO	0:0:0
h_rt	h_rt	TIME	0:0:0	<=	YES	NO	0:0:0
s_cpu	s_cpu	TIME	0:0:0	<=	NO	NO	0:0:0
h_cpu	h_cpu	TIME	0:0:0	<=	YES	NO	0:0:0
s_data	s_data	MEMORY	0	<=	NO	NO	0
h_data	h_data	MEMORY	0	<=	YES	NO	0
s_stack	s_stack	MEMORY	0	<=	NO	NO	0
h_stack	h_stack	MEMORY	0	<=	NO	NO	0
s_core	s_core	MEMORY	0	<=	NO	NO	0
h_core	h_core	MEMORY	0	<=	NO	NO	0
s_rss	s_rss	MEMORY	0	<=	NO	NO	0
h_rss	h_rss	MEMORY	0	<=	YES	NO	0

表 3-2　queue 複合の属性表示の例 (続き)

#Name	Shortcut	Type	Value	Relop	Requestable	Consumable	Default
min_cpu_interval	mci	TIME	0:0:0	<=	NO	NO	0:0:0
max_migr_time	mmt	TIME	0:0:0	<=	NO	NO	0:0:0
max_no_migr	mnm	TIME	0:0:0	<=	NO	NO	0:0:0

基本的に「name」列は、qconf -sq コマンドの出力の最初の列と同じです。キュー属性は、Sun Grid Engine, Enterprise Edition のキュープロパティの大部分をカバーしています。「shortcut」列には、最初の列の完全名の省略名で、管理者はこの省略名を定義することができます。ユーザーは、qsub コマンドの要求オプションで、完全名または省略名のどちらでも使用できます。

「requestable」列は、そのエントリを qsub で使用できるかどうかを示します。たとえば管理者は、エントリ qname か qhostname エントリ、またはその両方を要求不可に設定することによって、クラスタのユーザーがそのジョブでマシン / キューを直接要求するのを禁止することができます。このようにすることは、一般に、ユーザー要求を満たすことが可能なキューが複数あることを意味し、Sun Grid Engine, Enterprise Edition の負荷均衡機能が適用されます。

「relop」列は、キューがユーザー要求を満たすかどうかを計算で求める際に使用する関係演算を定義します。行われる比較は次のようなものです。

- User_Request　relop　Queue/Host/...-Property

比較結果が偽の場合、検討対象のそのキューではユーザーのジョブは実行できません。たとえば、キュー *q1* にソフト CPU 時間制限として 100 秒が設定されいるのに対し、キュー *q2* には同じソフト CPU 時間制限として 1000 秒が設定されていることがあります (ユーザープロセス制限については、「queue_conf と setrlimit のマニュアルページ参照)。

「consumables」列と「default」列は、管理者がいわゆる消費可能資源を定義する際に意味を持ちます (201 ページの「消費可能資源」の節を参照)。ユーザーは、他の属性と同様に消費可能資源を要求します。ただし、Sun Grid Engine, Enterprise Edition 内部の資源ブックキーピングはこれと異なります。

ユーザーから次の要求があったと仮定しましょう。

```
% qsub -l s_cpu=0:5:0 nastran.sh
```

この s_cpu=*0:5:0* 要求 (この構文についての詳細は qsub のマニュアルページを参照) が求めているのは、少なくとも 5 分のソフト CPU 時間を付与するキューです。このため、ジョブの実行に適切なのは、少なくとも 5 分のソフト CPU 時間を提供するキューだけということになります。

注 – 複数のキューでジョブを実行できる場合、Sun Grid Engine, Enterprise Edition はスケジューリングプロセスで作業負荷情報だけを検討します。

ユーザーのアクセス権

Sun Grid Engine, Enterprise Edition 管理者は、キューおよびその他の Sun Grid Engine, Enterprise Edition 機能 (289 ページの「並列環境」で説明している並列環境インタフェースなど) に対する、特定のユーザーまたはユーザーグループのアクセスを制限することができます。

注 – Sun Grid Engine, Enterprise Edition は、クラスタ管理で構成されているアクセス制限を自動的に考慮します。ここでは、自分の個人的なアクセス権を調べる場合にのみ有用な情報を提供します。

アクセス権を制限するために、管理者はいわゆるアクセス制御リスト (*ACL*) を作成して、管理します。こうした ACL には、任意のユーザーおよび UNIX グループ名が含まれます。ACL を作成したら、それをキューまたは並列環境インタフェースの構成でアクセス許可 (*access-allowed*) またはアクセス拒否 (*access-denied*) リストに追加します (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `queue_conf` および `sge_pe` を参照)。

ACL に登録されていて、アクセス許可リストに登録されているユーザーは、キューまたは並列環境インタフェースに対するアクセス権を持ちます。これに対し ACL に登録されていて、アクセス拒否リストに登録されているユーザーはアクセスできません。

ACL はまた、Sun Grid Engine, Enterprise Edition プロジェクトの定義にも使用されます。登録されているユーザーはプロジェクトにアクセスし、プロジェクトの配下に自分のジョブを置くことができます。管理者は、プロジェクト単位でもクラスタ資源へのアクセスを制限することができます。

`QMON` メインメニューで「ユーザー構成」アイコンボタンをクリックしたときに表示される「ユーザーセット構成」ダイアログボックスを使用し、そのダイアログボックスからアクセスできる ACL を調べることができます。詳細は、第 9 章、219 ページの「ユーザーアクセスとポリシーの管理」を参照してください。

Sun Grid Engine, Enterprise Edition プロジェクトへのアクセス権情報は、QMON メインメニューの「プロジェクト構成」アイコンをクリックすることによって表示することができます。詳細は、233 ページの「プロジェクト」の節を参照してください。

構成済みの ACL のリストをコマンド行から表示するには、次のコマンドを使用します。

```
% qconf -sul
```

次のコマンドは、指定された名前の ACL の内容を表示します。

```
% qconf -su acl_name[,...]
```

acl_name は、ACLの名前です。

ACL はユーザーアカウント名と UNIX グループ名で構成され、このうち UNIX グループ名は先頭の @ 記号で識別されます。このようにして、自分のアカウントが登録されている ACL を調べることができます。

注 – `newgrp` コマンドを使用して一次 UNIX グループを切り替える権限を持っている場合は、自分のアクセス権を変更できます。

これで、自分がアクセス可能な、またはアクセス拒否されるキューまたは並列環境インタフェースを確認することができます。58 ページの「キューとキュープロパティ」、290 ページの「QMON から並列環境を構成する」の説明に従ってキューまたは並列環境インタフェース構成を調べてください。アクセス許可リストの名前は `user_lists`、アクセス拒否リストの名前は `xuser_lists` です。自分のユーザーアカウントまたは一次 UNIX グループがアクセス許可リストに関連付けられている場合は、その資源へのアクセスが許可されます。アクセス拒否リストに関連付けられている場合は、アクセスできません。両方のリストとも空の場合、正当なアカウントを持つあらゆるユーザーが資源にアクセスできます。

Sun Grid Engine, Enterprise Edition のプロジェクト構成をコマンド行から制御するには、次のコマンドを使用します。

```
% qconf -sprjl
% qconf -sprj <プロジェクト名>
```

これらのコマンドは、定義済みのプロジェクトのリストと特定のプロジェクトの構成をそれぞれ表示します。プロジェクトは ACL を使用して定義するため、上記のようにして ACL 構成を調べる必要があります。

プロジェクトにアクセスできる場合は、そのプロジェクトの配下にあるジョブを実行依頼することができます。コマンド行からは、この操作は次のコマンドを使用して行うことができます。

```
% qsub -p <プロジェクト名> <その他のオプション>
```

クラスタやホスト、キュー構成では、project_lists および xproject_lists パラメータを使用して、ACL に対するのと同じ方法でプロジェクトへのアクセス権を定義します。

マネージャーとオペレータ、所有者

Sun Grid Engine, Enterprise Edition マネージャーのリストは、次のコマンドで得ることができます。

```
% qconf -sm
```

オペレータのリストは次のコマンドで得ることができます。

```
% qconf -so
```

注 – Sun Grid Engine, Enterprise Edition 管理ホストのスーパーユーザーは、デフォルトでマネージャーとみなされます。

---

58 ページの「キューとキュープロパティ」の節で説明しているように、特定のキューの所有者であるユーザーは、キュー構成データベースに記録されます。このデータベースは、次のコマンドを実行することによって読み出すことができます。

```
% qconf -sq queue_name
```

queue_name は、キューの名前です。

キュー構成のそうしたエントリは owners となっています。

第4章

ジョブの実行依頼

この章では、Sun Grid Engine, Enterprise Edition 5.3 を使用したジョブの実行依頼に関する予備知識的な情報とその実施方法を説明します。最初に練習で簡単なジョブを実行し、より複雑なジョブの実行方法へと進みます。

具体的には、この章では以下の作業を行う方法を説明します。

簡単なジョブの実行

この節では、Sun Grid Engine, Enterprise Edition 5.3 ジョブの実行依頼をする基本的な手順を習得します。

注 – 特権のないアカウントで Sun Grid Engine, Enterprise Edition プログラムをインストールした場合、ジョブを実行するには、そのアカウントのユーザーとしてログインする必要があります (詳細は、23 ページの「前提となる作業」を参照)。

▼ コマンド行から簡単なジョブを実行する

Sun Grid Engine, Enterprise Edition のコマンドを実行するには、実行可能ファイルの検索パスとその他の環境条件を正しく設定しておく必要があります。

1. **使用しているコマンドインタプリタに従って、以下のコマンドのいずれか適切な方を入力します。**

   a. **コマンドインタプリタとして** `csh` **または** `tcsh` **を使用している場合**

   ```
   % source sge_root_dir/default/common/settings.csh
   ```

   sge_root_dir は、インストール手順の最初に選択した Sun Grid Engine, Enterprise Edition のルートディレクトリの場所です。

   b. **コマンドインタプリタとして** `sh` **か** `ksh`、`bash` **のいずれか使用している場合**

   ```
   # . sge_root_dir/default/common/settings.sh
   ```

   注 – `.login`、`.cshrc`、`.profile` のいずれか適切なファイルに上記のコマンドを追加しておくことによって、後で取り組むどの対話セッションでも、適切な Sun Grid Engine, Enterprise Edition 設定が行われるようにすることができます。

   ---

2. **Sun Grid Engine, Enterprise Edition クラスタに次の簡単なジョブスクリプトの実行依頼をします。**

   次のジョブは、Sun Grid Engine, Enterprise Edition のルートディレクトリの `examples/jobs/simple.sh` に含まれています。

   ```
   #!/bin/sh
   #This is a simple example of a Sun Grid Engine batch script
   #
   # Print date and time
   date
   # Sleep for 20 seconds
   sleep 20
   # Print date and time again
   date
   # End of script file
   ```

このジョブの実行依頼をするには、次のコマンドを入力します。ここでは、simple.sh が上記のジョブが含まれているスクリプトファイルで、そのファイルが現在の作業ディレクトリにあるものと仮定しています。

```
% qsub simple.sh
```

qsub コマンドによって、ジョブの実行依頼が正しく行われたことが確認されます。

```
your job 1 ("simple.sh") has been submitted
```

3. **次のコマンドを入力することによって、ジョブのステータス情報を読み出します。**

```
% qstat
```

現在 Sun Grid Engine, Enterprise Edition システムが認識しているすべてのジョブに関する情報からなるステータスレポートが表示されます。このレポートには、ジョブごと、いわゆるジョブ *ID* (実行依頼の確認に含まれていた一意の番号) とジョブスクリプト名、ジョブの所有者、状態情報 (r は実行中を意味する)、実行依頼または開始時間、ジョブが実行されるキューの名前が含まれます。

qstat コマンドからの出力がない場合は、システムが認識しているジョブは存在しないことになります。たとえば、ジョブはすでに完了している可能性があります。完了したジョブの出力は、その stdout および stderr リダイレクトファイルを調べることによって制御することができます。デフォルトでは、これらのファイルは、ジョブを実行したホストのジョブの所有者のホームディレクトリに作成されます。また、これらのファイルの名前は、ジョブスクリプトファイル名とピリオド (.)、stdout または stderr ファイルのどちらであるかを示す英字 1 字 (o または e)、それに一意のジョブ ID で構成されます。たとえば、ジョブが新規インストールされた Sun Grid Engine, Enterprise Edition システムで初めて実行されたジョブである場合、ジョブの stdout および stderr ファイル名はそれぞれ simple.sh.o1 と simple.sh.e1 になります。

▼ GUI の QMON からジョブの実行依頼をする

QMON グラフィカルユーザーインタフェースを使用すると、もっと簡単に Sun Grid Engine, Enterprise Edition ジョブの実行依頼や制御、Sun Grid Engine, Enterprise Edition システムの概要情報の表示を行うことができます。QMON には、ジョブの実行依頼と監視を行うためのジョブの実行依頼メニューと「ジョブ制御」ダイアログボックスが用意されています。

コマンド行プロンプトから次のコマンドを入力してください。

```
% qmon
```

起動中にメッセージウィンドウが現れ、その後で QMON メインメニューが表示されます。

4. 「ジョブ制御」ボタンをクリックして、「実行依頼」ボタンをクリックします。

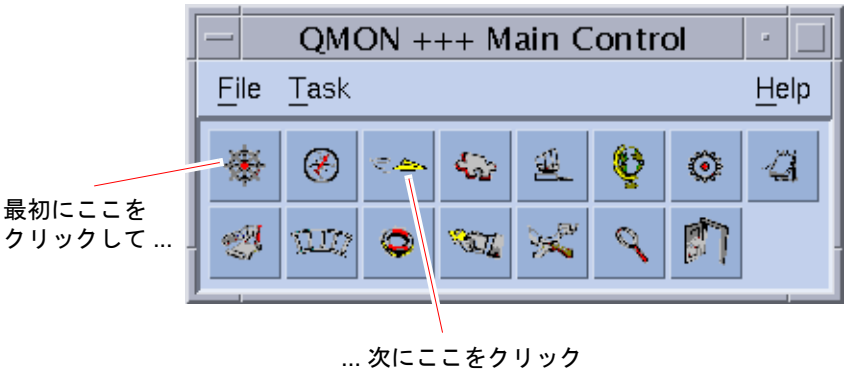


図 4-1　QMON メインメニュー

「ジョブの実行依頼」ダイアログボックスと「ジョブ制御」ダイアログボックスが表示されます (図 4-2 と 図 4-3 を参照)。ボタンの上にマウスポインタを置くと、そのボタン名 (ジョブ制御など) が表示されます。

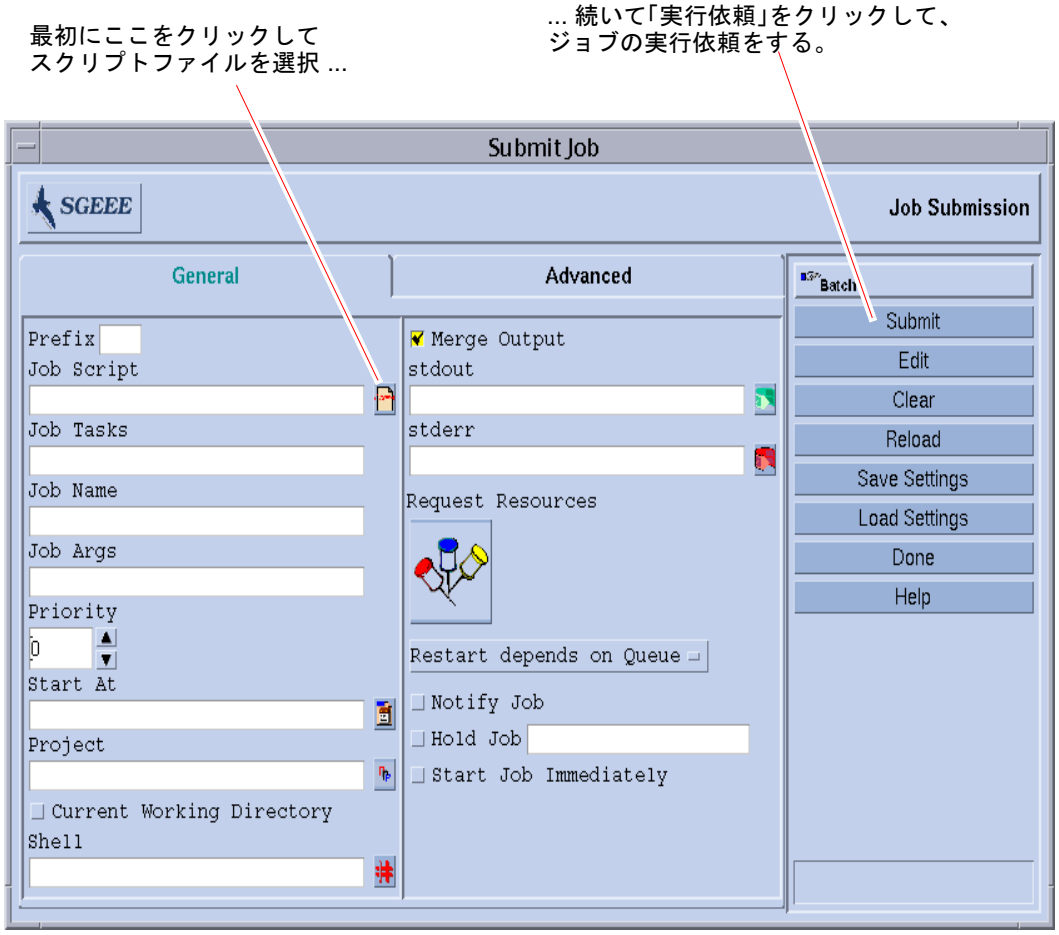


図 4-2　　QMON の「ジョブの実行依頼」ダイアログボックス

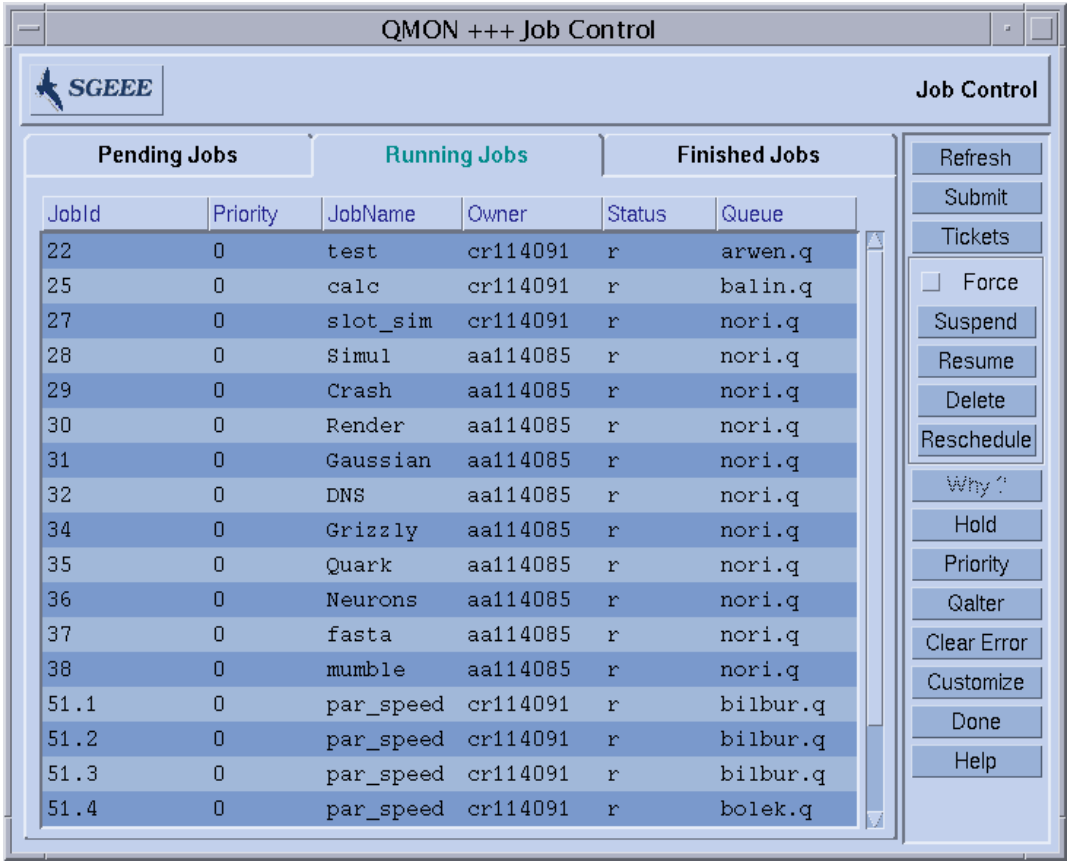


図 4-3　　QMON の「ジョブ制御」ダイアログボックス

5. **「ジョブの実行依頼」メニューで「ジョブスクリプト」ファイル選択アイコンをクリックしてファイル選択用のダイアログボックスを開きます。**

   ジョブスクリプト選択用のダイアログボックスが表示されます。

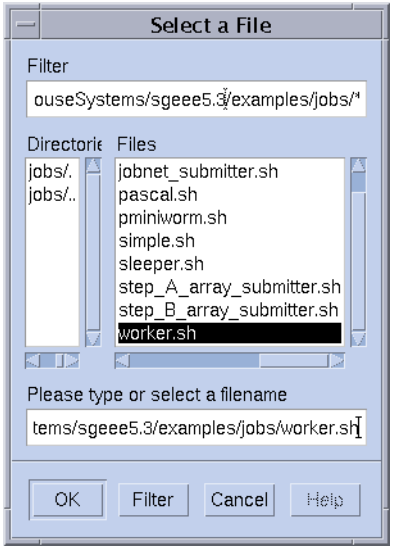


図 4-4　　ジョブスクリプト選択用のダイアログボックス

6. **適切なファイル名をクリックして、スクリプトファイルを選択します (たとえば上記のコマンド行の例の *simple.sh* のようなファイル)。**

7. **「ジョブの実行依頼」メニューの下部にある「実行依頼」ボタンをクリックします。**

   数秒経過すると、「ジョブ制御」パネルでジョブを監視することができます。実行依頼したジョブは、最初に「保留中のジョブ」欄に現れ、実行が開始されると、すぐに「実行中のジョブ」欄に移動します。

バッチジョブの実行依頼

この節では、Sun Grid Engine, Enterprise Edition 5.3 システムを使用してもっと複雑なジョブの実行依頼をする方法を説明します。

シェルスクリプト

バッチジョブとも呼ばれるシェルスクリプトは、基本的に、ファイルに組み込まれた一連のコマンド行命令です。スクリプトファイルは、chmod コマンドで実行可能にします。スクリプトを起動すると、適切なコマンドインタプリタ (csh、tcsh、sh、ksh など) が起動され、個々の命令が手動で入力されかのように解釈されていきます。シェルスクリプトからは、任意のコマンド、アプリケーション、他のシェルスクリプトを起動することができます。

適切なコマンドインタプリタが login-shell として起動されるかどうかは、その名前がジョブを実行する特定のホストおよびキューに対して有効な Sun Grid Engine, Enterprise Edition 構成の login_shells エントリの値リストに含まれているどうかに依存します。

注 – Sun Grid Engine, Enterprise Edition の構成は、クラスタに構成されているホストやキューによって異なることがあります。有効な構成は、qconf コマンドの -sconf オプションと -sq オプションを使用して表示することができます (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル 』を参照)。

コマンドインタプリタが login-shell として起動された場合、ジョブの環境は、ログインしてスクリプトを実行したのと完全に同じ環境になります。たとえば csh が login-shell として起動されなかった場合は .cshrc だけが実行されるのに対し、login-shell として起動された場合は、.login と .cshrc に加えて、システムのデフォルト起動リソースファイル (たとえば /etc/login のようなもの) も実行されます。login-shell として起動された場合とそうでない場合の相違点については、ご使用のコマンドインタプリタのマニュアルページを参照してください。

スクリプトファイルの例

コード例 4-1 は簡単なシェルスクリプトの例です。このスクリプトは、Fortran77 ソースの flow をコンパイルすることによってそのアプリケーションを生成し、実行します。

```
#!/bin/csh
# This is a sample script file for compiling and
# running a sample FORTRAN program under Sun Grid Engine, #
Enterprise Edition.
cd TEST
# Now we need to compile the program 'flow.f' and
# name the executable 'flow'.
f77 flow.f -o flow
```

コード例 4-1　　簡単なシェルスクリプト例

ご使用のシステムのユーザーマニュアルには、シェルスクリプトの作成とカスタマイズに関する詳細な記述があります (sh、ksh、csh、tcsh のマニュアルページも参照するとよい)。以降の節では、Sun Grid Engine, Enterprise Edition 用にバッチスクリプトを作成する際に特に注意すべき事項を重点的に説明します。

一般に、端末接続を必要とせず (自動的にリダイレクトされる標準エラーおよび標準出力デバイスは除く)、対話形式のユーザー介入を必要としない限り、手動でコマンドプロンプトから実行可能なシェルスクリプトは、すべて Sun Grid Engine, Enterprise Edition に実行依頼することができます。このため、コード例 4-1 は、そのまま Sun Grid Engine, Enterprise Edition に実行依頼すれば、目的の処理が行われます。

QMON におけるジョブの実行依頼の拡張設定と高度設定

ここでは、もっと複雑な形態のジョブの実行依頼に進む前に、ジョブの実行依頼プロセスに関する予備知識的な重要情報を提供します。

拡張設定

標準の形式の「ジョブの実行依頼」ダイアログボックス (図 4-2 を参照) では、以下のパラメータを設定することができます。

- 接頭辞文字列 - Sun Grid Engine, Enterprise Edition のスクリプト埋め込み実行依頼オプションに使用する文字列です (詳細は、93 ページの「アクティブな Sun Grid Engine, Enterprise Edition コメント」の節を参照)。
- 使用するジョブスクリプト

右横のファイルアイコンのボタンをクリックすると、ファイル選択用のダイアログボックスが開きます (図 4-4 を参照)。

- タスク ID 範囲 - 配列ジョブの実行依頼で使用されます (98 ページの「配列ジョブ」を参照)。
- ジョブ名 - ジョブスクリプトを選択するとデフォルト名が設定されます。
- ジョブスクリプトに渡す引数
- ジョブの初期優先順位 - カウントボックスを使用して設定できます。

  Sun Grid Engine, Enterprise Edition では、この優先順位で単一ユーザーの複数のジョブをランク付けします。Sun Grid Engine, Enterprise Edition スケジューラは、この値によって、単一ユーザーのジョブがシステムに複数存在する場合にその選択方法を決定します。

注 – 管理者は業務優先ポリシーにチケット、業務優先ジョブカテゴリに配分を割り当てて、ユーザーは自分のジョブに重みを付けられるようにする必要があります。

- ジョブを実行対象とみなす時間

  右横のファイルアイコンのボタンをクリックすると、正しい書式で時間を入力するためのダイアログボックスが表示されます (図 4-5 を参照)

- ジョブを配下に置く Sun Grid Engine, Enterprise Edition プロジェクト

  入力フィールド横のボタンを使用して、使用可能なプロジェクトを選択することができます (図 4-6 を参照)。

- 現在の作業ディレクトリでジョブを実行するかどうかを示すフラグ (実行依頼ホストと実行ホスト候補間のディレクトリ階層が同じ場合のみ)
- ジョブスクリプトの実行に使用するコマンドインタプリタ (92 ページの「コマンドインタプリタの選択方法」)

  横のボタンをクリックすると、ジョブに使用するコマンドインタプリタを指定するためのダイアログボックスが表示されます (図 4-7 を参照)

- ジョブの標準出力と標準エラー出力を標準出力ストリームに結合するかどうかを示すフラグ
- 使用する標準出力のリダイレクト先 (93 ページの「出力のリダイレクト」を参照)。

  何も指定されなかった場合にデフォルトが使用されます。横のファイルアイコンのボタンをクリックすると、出力のリダイレクト先を指定するためのヘルパーダイアログボックスが表示されます (93 ページの「出力のリダイレクト」を参照)。

- 使用する標準エラー出力のリダイレクト先 - 標準出力のリダイレクト先によく似ています。
- ジョブの資源要求

  ジョブの資源要求を定義するには、対応するアイコンのボタンをクリックします。ジョブの資源要求を定義すると、アイコンのボタンの色が変わります。

- システムクラッシュまたは類似イベントでジョブの実行が中止された後にジョブを再開可能にするかどうか、あるいは再開時の動作をキューに依存させるか、あるいはジョブに要求させるかどうかの指定 - 選択用のボタンを使用して指定します。
- ジョブが一時停止または取り消されようとしている場合に、それぞれ `SIGUSR1` または `SIGUSR2` シグナルでそのことをジョブに通知するかどうかを示すフラグ。
- ジョブにユーザーホールドまたはジョブ依存関係を割り当てることを示すフラグ。

  ホールドの種類に関係なく、ホールドが割り当てられている限り、ジョブが実行対象になることはありません (ホールドについての詳細は、121 ページの「Sun Grid Engine, Enterprise Edition ジョブの監視と制御」の節を参照)。「ホールド」フラグの入力フィールドを使用して、配列ジョブの特定の範囲のタスクだけホールドすることができます (98 ページの「配列ジョブ」を参照)。
- ジョブを強制的にただちに開始させるか (可能な場合)、拒否させるフラグ。

  このフラグが選択された場合、ジョブはキューに入れられません。

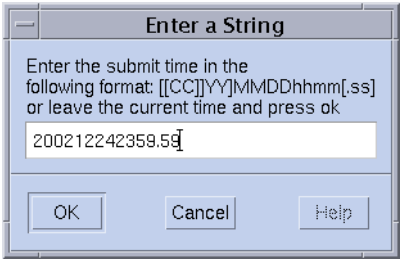


図 4-5　時間入力用のダイアログボックス

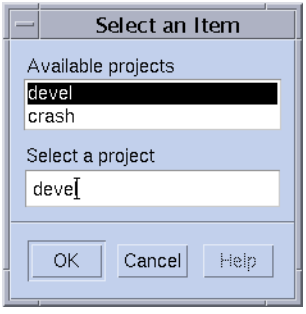


図 4-6　プロジェクト選択用のダイアログボックス

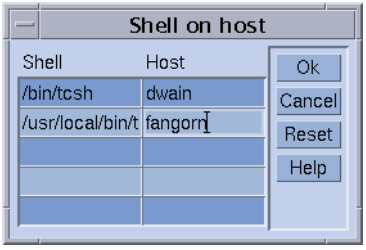


図 4-7　　シェル選択用のダイアログボックス

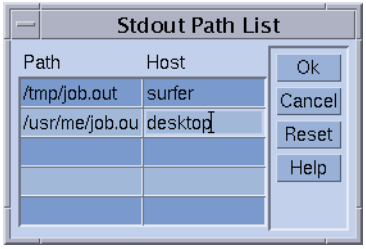


図 4-8　　出力のリダイレクト用のダイアログボックス

「ジョブの実行依頼」ダイアログボックスの右側にあるボタンを使用して、いろいろな操作を開始することができます。

- **実行依頼** - ダイアログボックスの指定内容に従ってジョブの実行依頼をします。
- **編集** - `vi` または `$EDITOR` 環境変数に設定されたエディタを使用して、X 端末で選択されているスクリプトファイルを編集します。
- **クリア** - 資源要求の指定をはじめとする、「ジョブの実行依頼」ダイアログボックスのすべての設定をクリアします。
- **再読み込み** - 指定されたスクリプトファイルを再度読み込んで、すべてのスクリプト埋め込みオプションとデフォルトの設定を構文解析し、それらの設定に対する未確定の手動変更を廃棄します (93 ページの「アクティブな Sun Grid Engine, Enterprise Edition コメント」と 97 ページの「デフォルトの要求」を参照)。この操作は、以前のスクリプトファイルで指定を行ってからクリア操作を行うのと同じことです。このオプションは、スクリプトファイルがすでに選択されている場合にのみ有効になります。
- **設定を保存** - 現在の設定をファイルに保存します。ファイル選択用のダイアログボックスが開いて、ファイルを選択することができます。保存したファイルは後で明示的に読み込んだり、デフォルトの要求として使用したりできます (97 ページの「デフォルトの要求」の節を参照)。
- **設定を読み込み** - 以前に「設定を保存」ボタンを使用して保存された設定を読み込みます。読み込まれた設定によって、現在の設定は書き換えられます。

- **完了** - 「ジョブの実行依頼」ダイアログボックスを閉じます。
- **ヘルプ** - このダイアログボックス専用のヘルプを表示します。

図 4-9 は、大部分のパラメータを設定した「ジョブの実行依頼」ダイアログボックスを示しています。

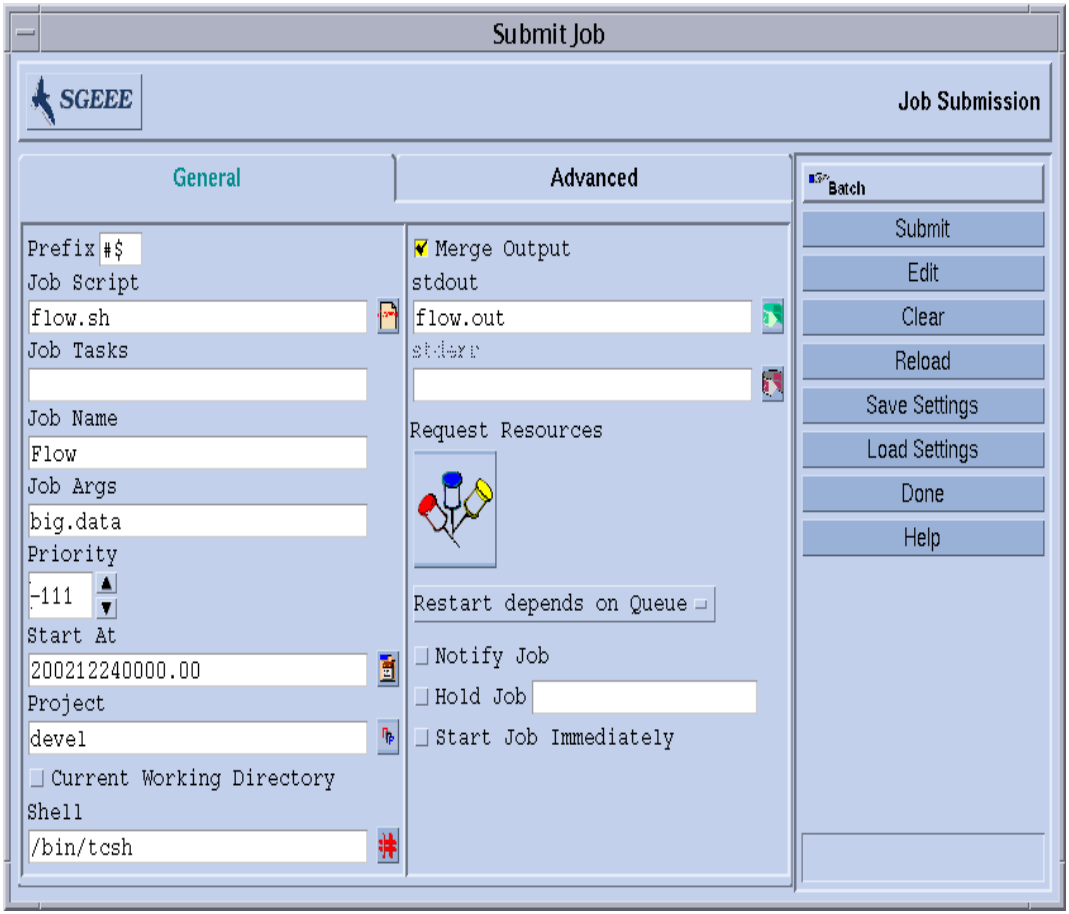


図 4-9 ジョブの実行依頼の拡張設定例

この例で設定したジョブのスクリプトファイル名は `flow.sh` で、QMON の作業ディレクトリに存在している必要があります。ジョブ名は `Flow` で、スクリプトファイルは `big.data` という引数を 1 つ受け取ります。ジョブは優先順位 `-111` で開始され、2002 年 12 月 24 日の真夜中より前に実行対象になることはありません。Sun Grid Engine, Enterprise Edition 固有のプロジェクト定義では、ジョブはプロジェクト `devel` の配下に置かれます。実行依頼と同じ作業ディレクトリで実行され、`tcsh` コマンドインタプリタを使用します。 最後に、標準出力と標準エラー出力はファイル `flow.out` に結合され、このファイルも現在の作業ディレクトリに作成されます。

高度な設定

高度な実行依頼画面では、次の追加のパラメータを定義することができます。

- 使用する並列環境インタフェース
- 実行前にジョブに設定する一群の環境変数。

  右横のアイコンのボタンをクリックすると、エクスポートする環境変数を定義するためのヘルパーダイアログボックスが表示されます (図 4-10 を参照)。QMON の実行時環境から環境変数を取り込んだり、任意の環境変数を定義したりできます。

- コンテキストと呼ばれる名前と値の対のリスト (図 4-11 を参照) - このリストを使用して、Sun Grid Engine, Enterprise Edition 内のあらゆる場所からアクセス可能なジョブ関連情報を保存したり、やりとりしたりすることができます。

  コンテキスト変数は、コマンド行から `qsub` や `qrsh`、`qsh`、`qlogin`、`qalter` の `-ac/-dc/-sc` オプションを使用して変更し、`qstat -j` で読み出すことができます。

- 使用するチェックポイント環境 - ジョブに対するチェックポイント機能の使用が望ましくかつ適切な場合に指定します (115 ページの「チェックポイントジョブ」の節を参照)。
- ジョブに関連付けるアカウント文字列

  アカウント文字列は、このジョブの記録であるアカウンティングレコードに追加され、アカウンティング分析に利用できます。

- 検査フラグ - ジョブの整合性検査モードを制御します。

  ジョブ要求の整合性の検査では、Sun Grid Engine, Enterprise Edition はクラスが空で無負荷状態であるとみなし、ジョブの実行が可能なキューを少なくとも 1 つ見つけようとします。指定可能な検査モードは次のいずれかです。

  - **スキップ** - 整合性検査を行いません。
  - **警告** - 整合性に関する問題点を報告しますが、ジョブは受け付けられます (ジョブの実行依頼後にクラスタ構成が変更される場合がある)。
  - **エラー** - 整合性に関する問題点が報告され、ジョブが拒否されます。
  - **検査のみ** - ジョブは実行依頼されませんが、クラスタ内の各ホストおよびキューに対するジョブの適切さに関する広範囲のレポートが生成されます。

- 電子メールでユーザーに通知するイベント

  現在、定義されているジョブのイベントは開始、終了、中止、一時停止です。

- 通知メールの送信先の一群の電子メールアドレスのリスト

  右横のボタンをクリックすると、メーリングリストを定義するためのヘルパーダイアログボックスが表示されます (図 4-12 を参照)。

- ジョブの実行に必須のキューとして要求するキューの名前のリスト

  「ハードキューリスト」と「ソフトキューリスト」は、80 ページの「ジョブの資源要求」の箇条書きの項目で説明している資源要求と同じものとみなされます。

- 並列ジョブ用のマスターキュー候補にするキューの名前のリスト。

  並列ジョブは、マスターキューで開始されます。ジョブの並列タスク生成先の他のすべてのキューは、スレーブキューと呼ばれます。

- 実行依頼するジョブを開始する前に正常に終了している必要があるジョブの ID リスト。

  新しく生成されたジョブが開始されるかどうかは、それらのジョブの正常終了に依存します。

- 締め切り優先ジョブの締め切り優先開始時間

  締め切り優先開始は、締め切り優先ジョブが指定された締め切り前に完了するために最高の優先順位に達している必要がある時間点を定義します。締め切り優先開始時間を決めるには、締め切り優先ジョブの締め切り期限から、最高の優先順位での実行時間の控え目な見積値を差し引きます。「締め切り」入力フィールド横のボタンをクリックすると、図 4-13 に示すようなヘルパーダイアログボックスが開きます。

注 – 必ずしもすべての Sun Grid Engine, Enterprise Edition ユーザーは締め切り優先ジョブの実行依頼を行えるわけではありません。締め切り優先の実行依頼が許可されているかどうかについては、システム管理者にお尋ねください。また、締め切り優先ジョブに割り当てられる最高優先順位については、クラスタ管理者にお尋ねください。

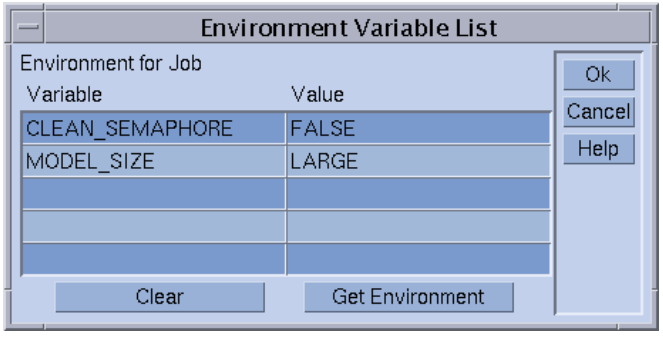


図 4-10 ジョブ環境の定義

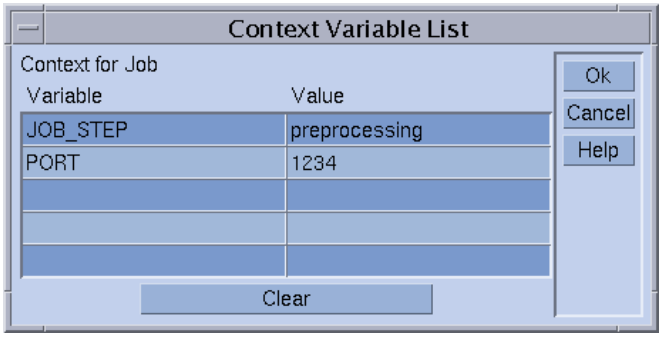


図 4-11　ジョブのコンテキストの定義

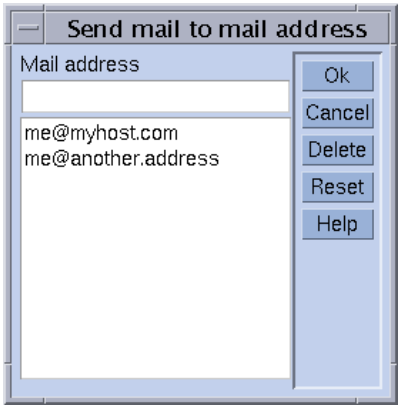


図 4-12　メールアドレスの指定

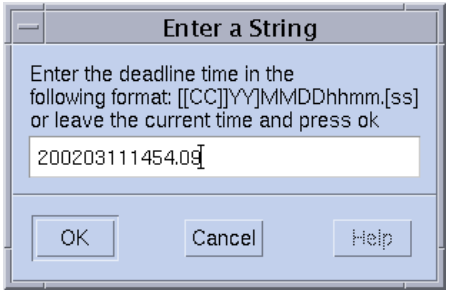


図 4-13　締め切り優先開始時間入力用のダイアログボックス

図 4-14 で定義しているジョブには、79 ページの「拡張設定」の節のジョブの定義に加えて、さらに次の特性が定義されています。

- 並列環境の `mpi` を使用する必要があります。少なくとも 4 つの並列プロセスを作成する必要があり、プロセスが利用可能な場合は、最高 16 個のプロセスを利用することができます。
- 2 つの環境変数が設定されていて、エクスポートされます。
- 2 つのコンテキスト変数が設定されています。
- ジョブアカウンティングレコードにアカウント文字列として `FLOW` が追加されます。
- システムクラッシュが原因で問題が発生した場合は、再開されます。
- ジョブ要求とクラスタ構成の間で整合性の問題が検出された場合は、警告が発行されます。
- ジョブが開始・完了したら、ただちに 2 つの電子メールアドレスにメールを送信する必要があります。
- 可能な場合は、`big_q` でジョブを実行するようにします。

図 4-14 は、ジョブの実行依頼の高度な設定例を示しています。

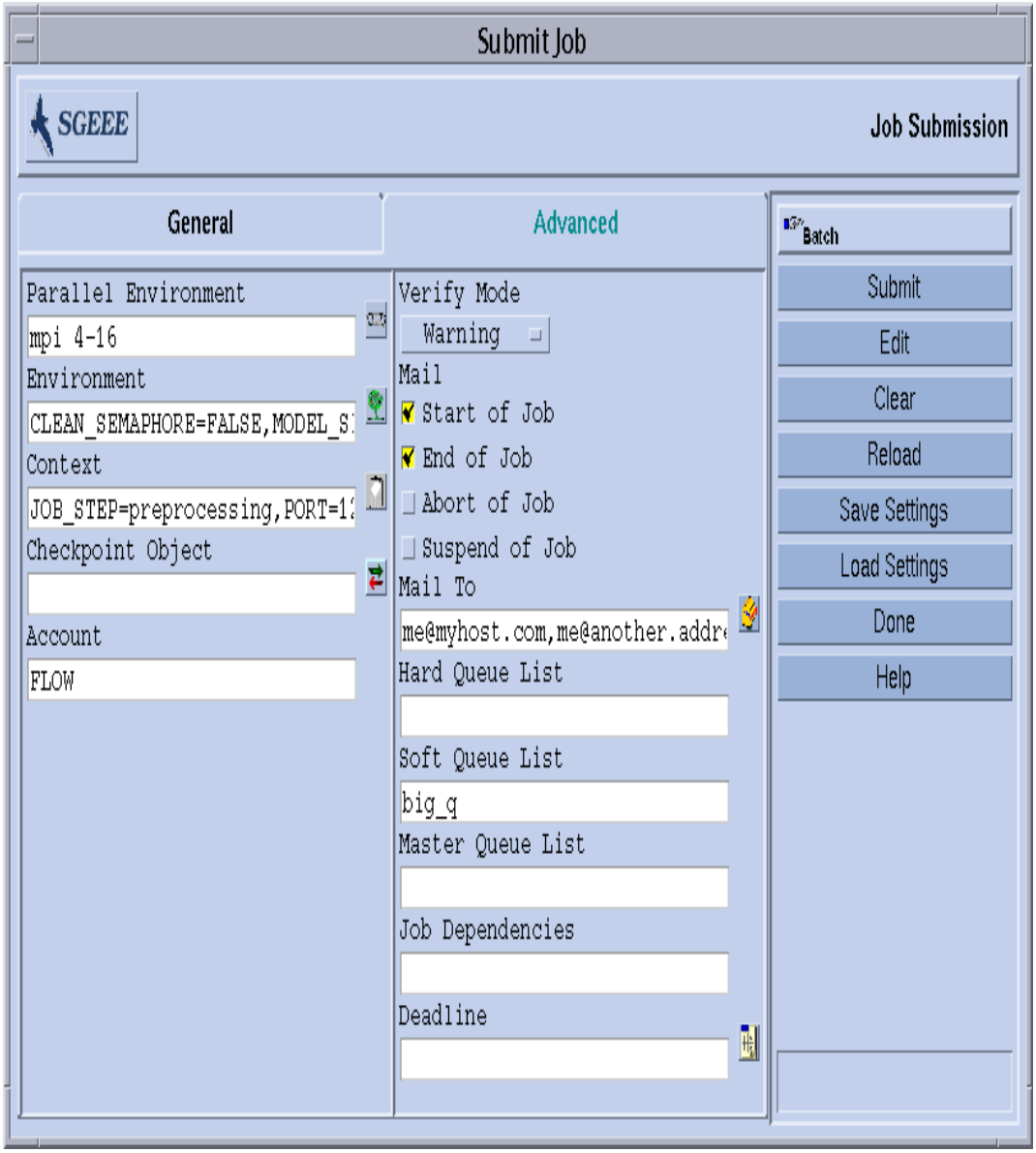


図 4-14　ジョブの実行依頼の高度な設定例

資源要求の定義

これまでの例では、ジョブを実行するホストに対する要求を表している実行依頼オプションはありません。Sun Grid Engine, Enterprise Edition は、そのようなジョブは任意のホストで実行可能であるとみなします。しかし、実際には大部分のジョブは、正常に終了するために実行ホストで満たされる必要がある何らかの前提条件があります。たとえば、使用可能なメモリーが十分にあること、必要なソフトウェアがインストールされていること、特定のオペレーティングシステムアーキテクチャであることなどの条件です。また、通常、クラスタ管理は、クラスタを構成するマシンの使用に対して制限を課します。たとえば、ジョブが消費できる CPU 時間はしばしば制限されます。

Sun Grid Engine, Enterprise Edition には、クラスタの装備やその利用ポリシーに関する明確な知識がなくても、ユーザーがジョブに適したホストを見つけられるようにするための手段が用意されています。ユーザーが行う必要があることは、ジョブの要求内容を指定して、Sun Grid Engine, Enterprise Edition に適切で負荷が軽いホストを見つける作業を管理させることだけです。

資源要求は、64 ページの「要求可能属性」の節で説明している要求可能属性を使用して指定します。`QMON` には、こうしたジョブの要求を指定する非常に便利な手段が用意されています。「ジョブの実行依頼」ダイアログボックス (図 4-15 を参照) で「要求資源」ボタンをクリックすると開く「要求資源」ダイアログボックスの「使用可能な資源」選択リストには、現在使用可能な属性だけが表示されます。属性をダブルクリックすると、その属性がジョブのハードまたはソフト資源リストに追加され (単に `True` に設定される `BOOLEAN` 型の属性は除く)、ヘルパーダイアログボックスが開いて、その属性に対する値指定の操作案内をします。

図 4-15 の「要求資源」ダイアログボックス例では、ジョブに対する要求資源プロファイルとして、`permas` ライセンスが使用可能で、少なくとも 750M バイトのメモリーがある `solaris64` のホストを要求しています。この要求を満たすキューが複数見つかった場合は、すでに定義されているソフト資源要求が考慮されます (この例ではなし)。ハードとソフト両方の要求を満たすキューが見つからなかった場合は、ハード要求を満たすキューが適切とみなされます。

注 – ジョブに適するキューが複数ある場合にのみ、負荷条件によって、ジョブの開始場所が決定されます。

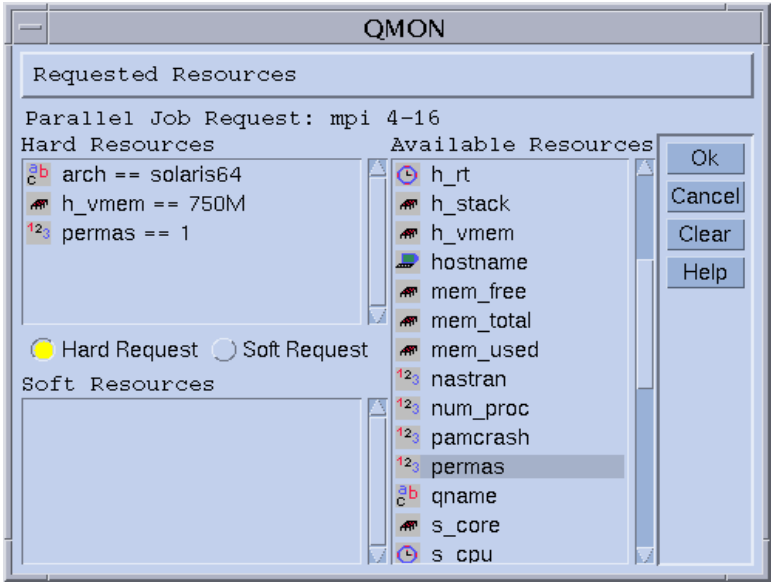


図 4-15　「要求資源」ダイアログボックス

注 – INTEGER 属性の `permas` は、管理者が「global」複合を拡張することによって導入された属性です。STRING 属性の `arch` は「host」複合から、また MEMORY 属性の `h_vmem` は「queue」複合からインポートされています。

同等の資源要求プロファイルは、以下のコマンドを使用してコマンド行から指定することもできます。

```
% qsub -l arch=solaris64,h_vmem=750M,permas=1 \
    permas.sh
```

注 – 最初の `-l` オプションの前には、暗黙の `-hard` スイッチが省略されています。

Sun Grid Engine, Enterprise Edition では、数量をたとえば *750M* (750M バイトを表す) というような構文で表現します。メモリーを要求する属性の場合は、整数型 10 進数、浮動小数点型 10 進数、整数型 8 進数、整数型 16 進数に、いわゆる乗数子を付けた値を指定することができます。

- 小文字の k - 数値の 1000 倍
- 大文字の K - 数値の 1024 倍
- 小文字の m - 数値の 1000 の 1000 乗倍

■ 大文字の M - 数値の 1024 の 1024 乗倍

8 進定数は、先行ゼロ (0) で 0 から 7 の範囲の数字を指定します。16 進定数の指定では、0x から始めて 0 ～ 9、a ～ f、 A ～ F の範囲の数値を使用する必要があります。乗数子が付けられていない場合、その値はバイト数とみなされます。浮動小数点型の 10 進数を使用した場合は、整数値に切り詰められます。

時間制限を課す属性の場合は、時、分、秒単位、またはその任意の組み合わせで時間値を指定することができます。時、分、秒は、コロンで区切った 10 進数で指定します。`3:5:11` という時間は 11111 秒に変換されます。時、分、秒の部分が 0 の場合は、コロンを残すことによって、その部分の指定を省略することができます。たとえば :5: という値は 5 分とみなされます。「要求資源」ダイアログボックスで使用されている上記の形式は拡張形式であり、`QMON` でしか使用できません。

Sun Grid Engine, Enterprise Edition の資源割り当て方法

前節で説明したように、Sun Grid Engine, Enterprise Edition ソフトウェアの資源要求の処理方法と資源割り当て方法を理解しておくことは重要です。以下に、Sun Grid Engine, Enterprise Edition ソフトウェアの資源割り当てアルゴリズムの仕組みをまとめておきます。

1. デフォルトの要求ファイルをすべて読み込んで構文解析します (97 ページの「デフォルトの要求」の節参照)。
2. スクリプトファイルの埋め込みオプションの処理をします (93 ページの「アクティブな Sun Grid Engine, Enterprise Edition コメント」の節を参照)。
3. スクリプト内の位置に関係なく、ジョブを実行依頼するときにすべてのスクリプト埋め込みオプションを読み取ります。
4. コマンド行からすべての要求を読み取って構文解析します。

すべての `qsub` 要求を収集すると、ハードおよびソフト要求が別々に処理されます (ハードが先)。これらの要求は、次の優先順で評価されます。

1. スクリプト / デフォルト要求ファイルの左から右
2. スクリプト / デフォルト要求ファイルの上から下
3. コマンド行の左から右

言い替えれば、コマンド行は埋め込まれたフラグに優先する指定を行う目的に使用することができます。

ハード資源として要求された資源が割り当てられます。要求が不正な場合、実行依頼は拒否されます。要求されたキューが使用中などの理由で実行依頼時に少なくとも 1 つの要求を満たすことができない場合、ジョブはスプールされ、後でスケジューリングし直されます。すべてのハード要求を満たすことができる場合は、それらハード資源が割り当てられ、ジョブの実行が可能になります。

ソフト資源として要求された資源が調べられます。ソフト要求の一部またはすべてを満たすことができなくても、ジョブは実行することができます。ハード要求をすでに満たしている複数のキューがソフト資源リストに含まれている場合 (重複しているか、部分的に異なる)、Sun Grid Engine, Enterprise Edition ソフトウェアは最も多くのソフト要求を満たすキューを選択します。

ジョブは開始され、割り当てられた資源をカバーします。

`hostname` または `date` などの UNIX コマンドを入れた小さなテストスクリプトファイルをテスト実行することによって、引数リストオプションと埋め込みオプション、あるいはハードおよびソフト要求が互いにどのように影響し合うのかを経験的に理解するようにしてください。

通常のシェルスクリプトの拡張

Sun Grid Engine, Enterprise Edition の管理下で実行した場合にスクリプトの動作に影響する、通常のシェルスクリプトにはない拡張機能があります。以下では、そうした拡張機能について説明します。

コマンドインタプリタの選択方法

ジョブスクリプトファイルの処理に使用するコマンドインタプリタは、実行依頼時に指定することができます (たとえば図 4-9 を参照)。ただし、何も指定されなかった場合は、構成変数の `shell_start_mode` によってコマンドインタプリタの選択方法が決まります。

- `shell_start_mode` が `unix_behavior` に設定されている場合は、スクリプトファイルの先頭行 (#! 文字列から始まる行) が評価されて、使用するコマンドインタプリタが決定されます。先頭行に `#!` がない場合は、デフォルトで Bourne シェルの `sh` が使用されます。
- `shell_start_mode` が上記以外に設定されている場合は、ジョブが開始されるキューに対する `shell` パラメータで設定されているデフォルトのコマンドインタプリタが使用されます (58 ページの「キューとキュープロパティ」の節と `queue_conf` のマニュアルページを参照)。

出力のリダイレクト

バッチジョブは端末接続がないため、その標準出力と標準エラー出力はファイルにリダイレクトする必要があります。Sun Grid Engine, Enterprise Edition では、出力のリダイレクト先のファイルの場所を定義することができます (何も指定されていない場合は、デフォルトが使用される)。

これらのファイルの標準の場所は、ジョブが実行されている現在の作業ディレクトリです。そして、デフォルトの標準出力ファイル名は <**ジョブ名**>.o<*Job_id*>、デフォルトの標準エラー出力は <**ジョブ名**>.e<*Job_id*> にリダイレクトされます。<**ジョブ名**> はスクリプトファイル名から作成されるか、ユーザー自身が定義することができます (`qsub` のマニュアルページの `-N` オプションの例を参照)。<*job_id*> は、Sun Grid Engine, Enterprise Edition によってジョブに割り当てられた一意の識別子です。

配列ジョブのタスクの場合は (98 ページの「配列ジョブ」の節を参照)、ピリオドで区切ったタスク識別子がファイル名に追加されます。つまり、標準のリダイレクトパスはそれぞれ <**ジョブ名**>.o<*Job_id*>.<*Task_id*> と <**ジョブ名**>.e<*Job_id*>.<*Task_id*> になります。

標準の場所が適切でない場合は、`QMON` または `qsub` の `-e` および `-o` オプションを使用して出力のリダイレクト先を指定することができます (図 4-14 および 図 4-8 を参照)。標準出力と標準エラー出力は 1 つのファイルに結合することができ、またリダイレクト先は、実行ホストごとに指定することができます。このため、ジョブが実行されるホストによって、出力のリダイレクト先ファイルは異なることになります。`qsub` の `-e` および `-o` オプションと疑似環境変数を組み合わせて、独自で一意のリダイレクト先ファイルパスを作成することができます。この目的で使用可能な変数は以下のとおりです。

- `$HOME` - 実行マシンのホームディレクトリ
- `$USER` - ジョブ所有者のユーザー ID
- `$JOB_ID` - 現在のジョブ ID
- `$JOB_NAME` - 現在のジョブ名 (`-N` オプションを参照)
- `$HOSTNAME` - 実行ホスト名
- `$TASK_ID` - 配列ジョブのタスクの添字番号

これらの変数は、ジョブの実行中に実際の値に展開され、その値でリダイレクト先のパスが作成されます。

詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `qsub` の項を参照してください。

アクティブな Sun Grid Engine, Enterprise Edition コメント

シェルスクリプト内の # 記号から始まる行は、コメントとして扱われます。ただし、Sun Grid Engine, Enterprise Edition は特殊なコメント行を認識し、そうした行を特別な使用の仕方をします。そうしたスクリプト行の残りの部分は、Sun Grid Engine, Enterprise Edition の実行依頼コマンド `qsub` のコマンド行引数リストの一部であるかのように扱われます。それらコメント行内に指定されている

qsub オプションも「QMON ジョブの実行依頼」ダイアログボックスで解釈され、スクリプトファイルが選択されると、対応するパラメータがプリセットされます。

デフォルトでは特殊なコメント行は、#$ 接頭辞文字列で識別されます。この接頭辞文字列は、qsub の -C オプションを使用して変更することができます。

こうした仕組みを、実行依頼引数のスクリプト埋め込みといいます。以下は、スクリプト埋め込みコマンド行オプションを利用したスクリプトファイルの例です。

```
#!/bin/csh
#Force csh if not Sun Grid Engine, Enterprise Edition default
#shell
#$ -S /bin/csh
# This is a sample script file for compiling and
# running a sample FORTRAN program under Sun Grid Engine,
# Enterprise Edition.
# We want Sun Grid Engine, Enterprise Edition to send mail
# when the job begins
# and when it ends.
#$ -M EmailAddress
#$ -m b,e
# We want to name the file for the standard output
# and standard error.
#$ -o flow.out -j y
# Change to the directory where the files are located.
cd TEST
# Now we need to compile the program 'flow.f' and
# name the executable 'flow'.
f77 flow.f -o flow
# Once it is compiled, we can run the program.
flow
```

コード例 4-2　　スクリプト埋め込みコマンド行オプションの使用例

環境変数

Sun Grid Engine, Enterprise Edition ジョブの実行では、以下に示す多数の変数がジョブの環境にプリセットされます。

- ARC - ジョブが実行されているノードの Sun Grid Engine, Enterprise Edition アーキテクチャ名。この名前は、コンパイルで sge_execd バイナリに組み込まれます。
- COMMD_PORT - sge_commd(8) が通信要求を待機することになっている TCP ポート

- SGE_ROOT - 起動前に sge_execd に設定された Sun Grid Engine, Enterprise Edition のルートディレクトリか、デフォルトの /usr/SGE
- SGE_CELL - ジョブが実行される Sun Grid Engine, Enterprise Edition のセル
- SGE_JOB_SPOOL_DIR - ジョブの実行中に sge_shepherd(8) がジョブ関連のデータの格納に使用するディレクトリ
- SGE_O_HOME - ジョブの実行依頼元のホスト上のジョブ所有者のホームディレクトリのパス
- SGE_O_HOST - ジョブの実行依頼元のホスト
- SGE_O_LOGNAME - ジョブの実行依頼元のホスト上のジョブ所有者のログイン名
- SGE_O_MAIL - ジョブ実行依頼コマンドのコンテキスト内の MAIL 環境変数の内容
- SGE_O_PATH - ジョブ実行依頼コマンドのコンテキスト内の PATH 環境変数の内容
- SGE_O_SHELL - ジョブ実行依頼コマンドのコンテキスト内の SHELL 環境変数の内容
- SGE_O_TZ - ジョブ実行依頼コマンドのコンテキスト内の TZ 環境変数の内容
- SGE_O_WORKDIR - ジョブ実行依頼コマンドの作業ディレクトリ
- SGE_CKPT_ENV - チェックポイントジョブが実行されるチェックポイント環境 (qsub の -ckpt オプションで選択)
- SGE_CKPT_DIR - チェックポイントインタフェースのパス ckpt_dir (checkpoint のマニュアルページを参照)。チェックポイントジョブの場合にのみ設定されます。
- SGE_STDERR_PATH - ジョブの標準エラーストリームのリダイレクト先のファイルのパス名。一般には、プロローグ、エピローグ、並列環境の開始 / 停止、チェックポイントスクリプトからのエラーメッセージで出力を拡張する目的に使用されます。
- SGE_STDOUT_PATH - ジョブの標準出力ストリームのリダイレクト先のファイルのパス名。一般には、プロローグ、エピローグ、並列環境の開始 / 停止、チェックポイントスクリプトからのエラーメッセージを含む出力の機能強化に使用されます。
- SGE_TASK_ID - 配列ジョブ内のこのタスクが表すタスク ID
- ENVIRONMENT - つねに BATCH。スクリプトがバッチモードで実行されることを示します。
- HOME - passwd ファイルから読み取られたユーザーのホームディレクトリパス
- HOSTNAME - ジョブが実行されているノードのホスト名
- JOB_ID - ジョブを実行依頼したときに、sge_qmaster によってジョブに割り当てられた一意の識別子。99999 までの範囲の 10 進整数です。
- JOB_NAME - qsub スクリプトファイル名から作成されたジョブ名とピリオド 1 つ、ジョブ ID の数字からなるジョブ名。このデフォルト名は、qsub の -N で変更することができます。
- LOGNAME - passwd ファイルから読み取られたユーザーのログイン名

- NHOSTS - 並列ジョブが使用するスロット数
- NQUEUES - ジョブに割り当てられたキュー数 (シリアルジョブの場合はつねに 1)
- NSLOTS - 並列ジョブが使用するキューホスト数
- PATH - デフォルトのシェル検索パス :/usr/local/bin:/usr/ucb:/bin:/usr/bin
- PE - ジョブが実行される並列環境 (並列ジョブのみ)
- PE_HOSTFILE - Sun Grid Engine, Enterprise Edition が並列ジョブに割り当てる仮想並列マシンの定義を含むファイルのパス

  このファイルの形式についての詳細は、sge_pe の $pe_hostfile パラメータの説明を参照してください。この環境変数は、並列ジョブに対してのみ使用できます。
- QUEUE - ジョブが実行されているキューの名前
- REQUEST - ジョブスクリプト名か、qsub -N オプションを使用してジョブに明示的に割り当てられたジョブの要求名
- RESTARTED - チェックポイントジョブが再開されたかどうかを示します。ジョブが少なくとも 1 回中断されて、その後再会された場合に、値 1 が設定されます。
- SHELL - passwd ファイルから読み取られたユーザーのログインシェル

注 – これは、必ずしもジョブが使用しているシェルではありません。

- TMPDIR - ジョブの一時作業ディレクトリへの絶対パス
- TMP -TMPDIR 同じ。NQS との互換性を維持するために提供されています。
- TZ - sge_execd からインポートされた時間帯変数 (設定されている場合)
- USER - passwd ファイルから読み取られたユーザーのログイン名

▼ コマンド行からジョブの実行依頼をする

● **適切な引数を付けて qsub コマンドを入力します。**

たとえば 72 ページの「コマンド行から簡単なジョブを実行する」の節で説明したスクリプトファイル名 flow.sh を使用した簡単なジョブは、次のコマンドで実行依頼することができます。

```
% qsub flow.sh
```

ただし、図 4-9 に示すような QMON の拡張実行依頼と同等の結果を得るには、次のようなコマンドを使用します。

```
% qsub -N Flow -p -111 -P devel -a 200012240000.00 -cwd \
-S /bin/tcsh -o flow.out -j y flow.sh big.data
```

さらにコマンド行オプションを追加して、もっと複雑な要求を作成することもできます。たとえば 図 4-14 示す高度なジョブ実行依頼では、コマンドは以下のようになります。

```
% qsub -N Flow -p -111 -P devel -a 200012240000.00 -cwd \
    -S /bin/tcsh -o flow.out -j y -pe mpi 4-16 \
    -v SHARED_MEM=TRUE,MODEL_SIZE=LARGE \
    -ac JOB_STEP=preprocessing,PORT=1234 \
    -A FLOW -w w -r y -m s,e -q big_q\
    -M me@myhost.com,me@other.address \
    flow.shbig.data
```

デフォルトの要求

前節の最後の例を見ると、高度なジョブ実行要求はかなり複雑で手軽には使用できないことが分かります。同様の要求を頻繁に行う必要がある場合は、特に大変です。こうしたコマンド行の入力という面倒で誤りやすい作業を行うのを回避するには、スクリプトファイルに qsub オプションを埋め込むか (93 ページの「アクティブな Sun Grid Engine, Enterprise Edition コメント」を参照)、いわゆるデフォルトの要求を使用します。

クラスタ管理者は、すべての Sun Grid Engine, Enterprise Edition ユーザー用のデフォルト要求ファイルを作成することができます。これに対しユーザーもまた個人用のデフォルト要求ファイルばかりでなく、アプリケーション別のデフォルト要求ファイルを作成することができます (これらのファイルは、それぞれ自分のホームディレクトリと作業ディレクトリに置く)。

デフォルトの要求ファイルは、Sun Grid Engine, Enterprise Edition のジョブにデフォルトで適用する qsub オプションを指定した行で構成されるだけです。クラスタ全体のグローバルなデフォルト要求ファイルは、
<sge_root>/*<cell>*/common/sge_request に置きます。また、一般的な個人用デフォルト要求ファイルは $HOME/.sge_request、アプリケーション別のデフォルト要求ファイルは $cwd/.sge_request に置きます。

こうしたデフォルト要求ファイルが複数ある場合は、次の優先順で 1 つのデフォルト要求に結合されます

1. グローバルなデフォルト要求ファイル

2. 一般的な個人用デフォルト要求ファイル

3. アプリケーション別のデフォルト要求ファイル

注 – デフォルト要求ファイルよりも、スクリプト埋め込みおよび qsub コマンド行の方が優先順位が高くなります。つまり、デフォルト要求ファイルの設定はスクリプト埋め込みによって書き換えられ、qsub コマンド行オプションによっても書き換えられることがあります。

注 – デフォルト要求ファイルや埋め込みスクリプトコマンド、qsub コマンド行で qsub の -clear オプションを使用することによって、いつでも以前の設定を廃棄することができます。

以下は、個人用のデフォルト要求ファイルの内容例です。

```
-A myproject -cwd -M me@myhost.com -m b,e
-r y -j y -S /bin/ksh
```

このユーザーのすべてのジョブは、書き換えられない限り、アカウント文字列が *myproject* で、現在の作業ディレクトリで実行され、ジョブの開始と終了時にメール通知が *me@myhost.com* に送信されます。また、システムのクラッシュ後は再開され、標準出力と標準エラー出力は結合され、コマンドインタプリタとして ksh が使用されます。

配列ジョブ

Sun Grid Engine, Enterprise Edition の配列ジョブは、ジョブスクリプト内で同じ一群の操作の実行をパラメータ化して、繰り返す用途に最適です。そうした用途の代表例としては、デジタルコンテンツ制作業界のレンダリングなどの作業に見られます。アニメーションの計算をフレームに分割し、フレームごとに独立して同じレンダリング計算を行うことができます。

配列ジョブ機能は、そうした用途のアプリケーションを実行依頼して監視、制御する便利な手段です。他方、Sun Grid Engine, Enterprise Edition は、配列ジョブを効率的に実装することによって、単一のジョブに結合された多数の独立したタスクとして計算を処理します。配列ジョブを構成するタスクはすべて、配列の添字番号を使用して参照します。そして、それらの添字は、1 つの qsub コマンドによる配列ジョブの実行依頼中に定義された、そのジョブ全体の添字範囲にまたがります。

配列ジョブは、その全体を監視・制御 (一時停止、再開、取り消しなど) することも、個別タスク、または一部タスクを監視・制御することもできます。後者の場合、タスクを参照するには、ジョブ ID の末尾に対応する添字番号を追加します。タスク

が実行されると (通常のジョブの実行に非常によく似ている)、それらのタスクは環境変数 $SGE_TASK_ID を使用して自身のタスク添字番号を読み出したり、そのタスク ID 向けの入力データセットにアクセスしたりできます。

▼ コマンド行から配列ジョブの実行依頼をする

- **適切な引数を付けて qsub コマンドを入力します。**

  以下は、配列ジョブの実行依頼です。

```
% qsub -l h_cpu=0:45:0 -t 2-10:2 render.sh data.in
```

  -t オプションはタスクの添字範囲を定義します。この例の *2-10:2* は、*2* が最小、*10* が最大添字番号で、1 つおきに添字を使用 (*:2* の部分) することを指定しています。つまり、この配列ジョブは、添字 2、4、6、8、10 の 5 つのタスクから構成されます。タスクはそれぞれ *45* 分のハード CPU 時間制限を要求し (-l オプション)、Sun Grid Engine, Enterprise Edition によってディスパッチされ、開始されると、ジョブスクリプト *render.sh* を実行します。また、タスクは $SGE_TASK_ID を使用してタスク 2、4、6、8、10 のどれであるかを調べ、その添字番号を使用して、データファイル *data.in* 内の自分の入力データレコードを探すことができます。

▼ QMON から配列ジョブの実行依頼をする

- **追加の注意事項として以下のことを考慮しながら、73 ページの「GUI の QMON からジョブの実行依頼をする」の手順に従って操作します。**

注 – QMON から配列ジョブの実行依頼方法は、73 ページの「GUI の QMON からジョブの実行依頼をする」で説明している方法とほぼ同じです。唯一の違いは、図 4-9 に示すダイアログボックスの「ジョブのタスク」入力フィールドに、qsub の -t オプションに対するのと同じ構文でタスク範囲を指定する必要があることだけです。配列の添字構文についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qsub の項を参照してください。

このマニュアルの 121 ページの「Sun Grid Engine, Enterprise Edition ジョブの監視と制御」、134 ページの「コマンド行からの Sun Grid Engine, Enterprise Edition ジョブの制御」、さらには、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の QSTAT、QHOLD、QRLS、QMOD、QDEL の節に、Sun Grid Engine, Enterprise Edition ジョブの制御全般および配列ジョブの制御に関する関連情報が含まれています。

注 – 配列ジョブでは、通常のジョブに対する Sun Grid Engine, Enterprise Edition 機能のすべてを完全に利用することができます。たとえば配列ジョブを同時に並列ジョブにしたり、他のジョブと相互依存させたりすることができます。

対話形式のジョブの実行依頼

ジョブの実行で直接の入力を必要とするジョブの場合は、バッチジョブではなく、対話形式のジョブを実行依頼した方が便利です。そうしたジョブとしては、たとえば X-windows アプリケーション (定義では対話形式のアプリケーション) や、次の計算を制御するには中間結果の解釈が必要になる作業などがあります。

Sun Grid Engine, Enterprise Edition システムには、対話形式のジョブを作成する 3 通りの方法が存在します。

- `qlogin` - Sun Grid Engine, Enterprise Edition ソフトウェアによって選択されたホスト上で telnet に似たセッションを開始します。
- `qrsh` - UNIX 標準の `rsh` と同等の機能です。Sun Grid Engine, Enterprise Edition システムによって選択されたホストでコマンドを遠隔実行するか、コマンドが実行指定されなかった場合は、遠隔ホストと遠隔ログイン (`rlogin`) セッションを開始します。
- `qsh` - 指定に応じた表示セットまたは `DISPLAY` 環境変数の設定でジョブを実行するマシンから `xterm` を起動します。`DISPLAY` 変数の設定がなく、かつ表示先が明示的に定義されていない場合、Sun Grid Engine, Enterprise Edition は、その対話形式の実行依頼元のホスト上の X サーバーの 0.0 画面に `xterm` の出力を送信します。

注 – これらの機能が正しく機能するには、Sun Grid Engine, Enterprise Edition クラスタパラメータが正しく設定されている必要があります。qsh には適切な `xterm` 実行パスを定義し、この種のジョブで対話形式のキューが使用できるようになっている必要があります。クラスタが対話形式のジョブを実行できるように構成されているかどうかについては、システム管理者にお尋ねください。

対話形式のジョブのデフォルトの扱いは、実行依頼の時点で実行できない場合はキューに入れられないという点でバッチジョブの扱いと異なります。このことは、適切な資源が十分に使用できないために、実行依頼の直後に対話形式のジョブをディスパッチできないことを意味します。そのような場合、ユーザーは、Sun Grid Engine, Enterprise Edition クラスタが非常にビジーであることの通知を受けます。

このデフォルトの動作は、qsh、qlogin、qrsh に -now no オプションを付けることによって変更することができます。このオプションを指定すると、対話形式のジョブはバッチジョブと同様キューに入れられます。qsub で -now yes を使用すると、バッチジョブを対話形式のジョブのように扱うことができ、その場合はただちにディスパッチされて実行されるか、拒否されます。

注 – 対話形式のジョブは、INTERACTIVE タイプのキューでのみ実行することができます (詳細は、169 ページの「キューの構成」を参照)。

以降の節では、qlogin および qsh 機能の使用方法を簡単に説明します。qrsh コマンドについては、105 ページの「透過的な遠隔実行」の節でもっと広い文脈で説明します。

QMON からの対話形式のジョブの実行依頼

QMON から実行依頼できる対話形式のジョブは、Sun Grid Engine, Enterprise Edition によって選択されたホスト上で xterm を起動するタイプのジョブだけです。

▼ QMON から対話形式のジョブの実行依頼をする

- **「対話形式」のアイコンが表示されるまで、「ジョブの実行依頼」ダイアログボックスの右側のボタン欄の上にあるアイコンをクリックします。**

  このアイコンが表示されると、「ジョブの実行依頼」ダイアログボックスから対話形式のジョブの実行依頼をすることができます (図 4-16 と図 4-17 を参照)。

  ダイアログボックスの選択オプションの意味と使用方法は、77 ページの「バッチジョブの実行依頼」の節でバッチジョブに関して説明している意味および使用方法と同じです。基本的な違いは、対話形式のジョブに適用されない入力フィールドがいくつか入力不可表示になっていることです。

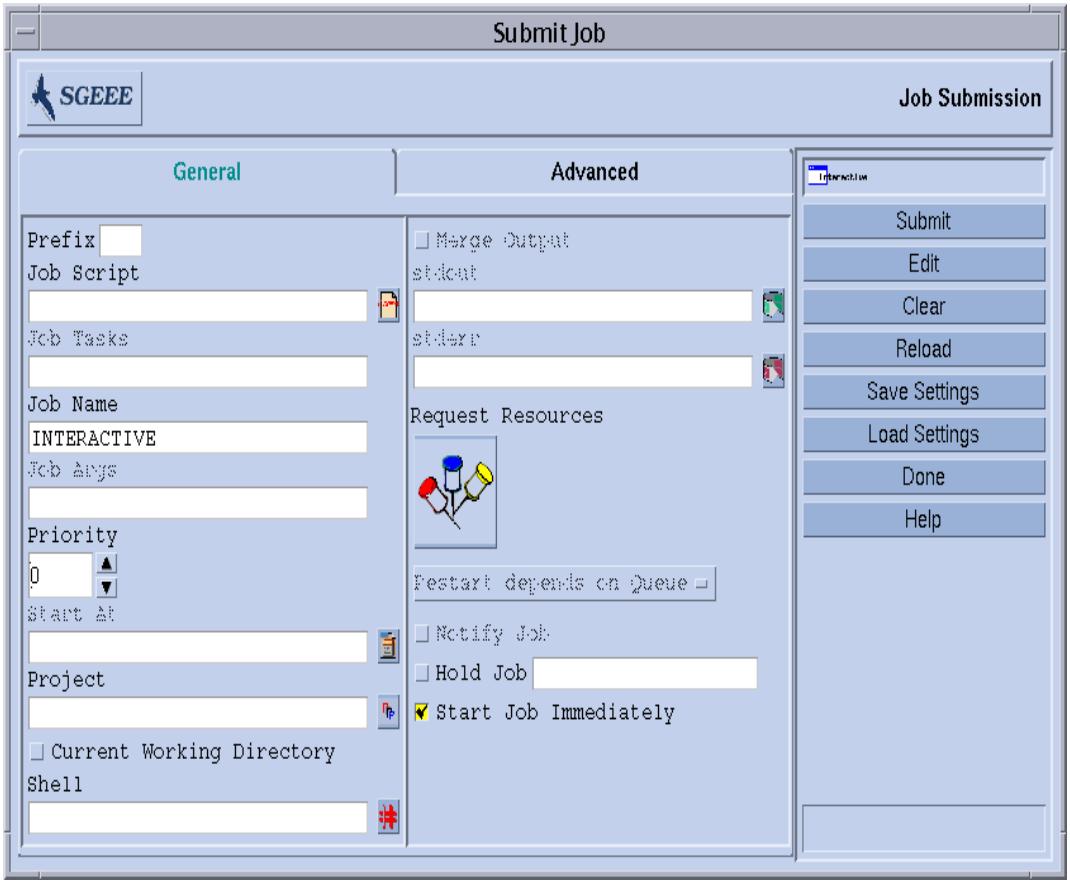


図 4-16　対話形式のジョブの実行依頼ダイアログボックス - 一般

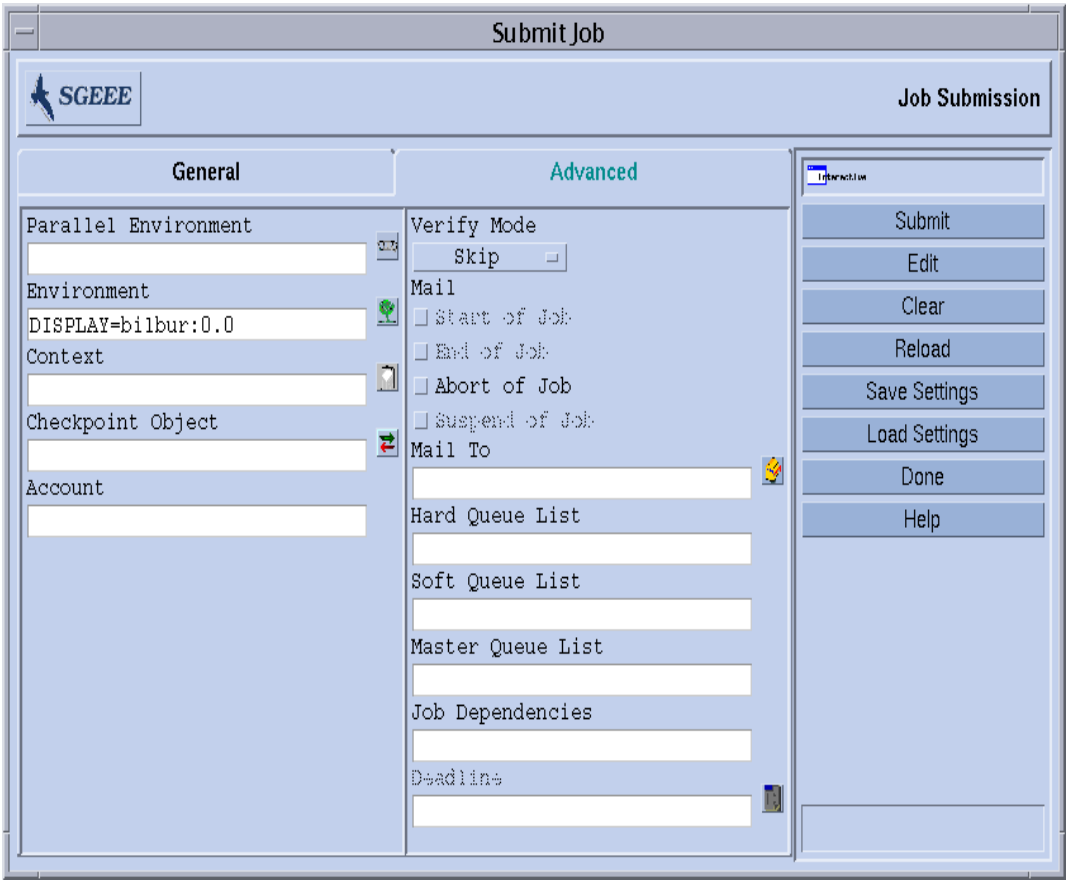


図 4-17　対話形式のジョブの実行依頼ダイアログボックス - 高度

qsh を使用した対話形式のジョブの実行依頼

qsh は qsub に非常によく似ており、qsub のオプションをいくつかサポートしているほか、起動する xterm の表示出力を送信する追加のスイッチ -display もサポー

トしています (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qsh の項を参照)。

▼ qsh を使用して対話形式のジョブの実行依頼をする

- **次のコマンドを入力すると、Sun の 64 ビット Solaris オペーレティング環境が動作する任意の使用可能ホストで xterm が起動されます。**

```
% qsh -l arch=solaris64
```

qlogin を使用した対話形式のジョブの実行依頼

任意の端末または端末エミュレータから qlogin コマンドを使用し、Sun Grid Engine, Enterprise Edition の管理下で対話形式のセッションを開始することができます。

▼ qlogin を使用して対話形式のジョブの実行依頼をする

- **次のコマンドを入力すると、Star-CD ライセンスを使用可能で、最低でも 6 時間のハード CPU 時間制限を提供するキューが少なくとも 1 つある、負荷の軽いホストが検索されます。**

```
% qlogin -l star-cd=1,h_cpu=6:0:0
```

注 – Sun Grid Engine, Enterprise Edition システムが使用するように設定されている遠隔ログイン機能によっては、ログインプロンプトが表示されたときにユーザー名かパスワード、またはその両方を入力する必要があります。

透過的な遠隔実行

Sun Grid Engine, Enterprise Edition には、いくとおりかの計算業務の透過的な遠隔実行をサポートする、密接に関連する一群の機能があります。この機能の中心は、105 ページの「qrsh を使用した遠隔実行」の節で取り上げている qrsh コマンドです。この qrsh の上位の 2 つの機能、qtcsh と qmake は、Sun Grid Engine, Enterprise Edition を使用して暗黙の計算業務を透過的に分散させることを可能にすることによって、make や csh などの標準の UNIX 機能を強化します。qtcch については、106 ページの「qtcsh を使用した透過的なジョブ分散」、qmake については、109 ページの「qmake を使用した並列メークファイル処理」を参照してください。

qrsh を使用した遠隔実行

qrsh は標準の rsh 機能を包む形で作成されており (rsh の関わりについての詳細は、<*sge_root*>/3rd_party で提供している情報を参照)、さまざまな目的に利用することができます。

- Sun Grid Engine, Enterprise Edition を使用して対話型のアプリケーションを遠隔実行する。この機能は、UNIX 標準の rsh 機能 (HP-UX では remsh ともいう) に相当します。
- Sun Grid Engine, Enterprise Edition を使用して対話形式のログインセッションを行う。この機能は、UNIX 標準の rlogin 機能に似ています (ただし、UNIX の telnet 機能を Sun Grid Engine, Enterprise Edition で実現するものとして、qlogin も必要です)。
- 実行後すぐに端末入出力 (標準 / エラー出力と標準入力) と端末制御が可能なバッチジョブの実行依頼を可能にする。
- シェルスクリプトに埋め込まれていないスタンドアロンプログラムの実行依頼をするための手段を提供する。
- バッチジョブの保留または実行中はアクティブで、完了するか、取り消された場合にのみ終了するバッチジョブ実行依頼クライアントを提供する。
- 並列ジョブによって割り当てられた分散資源の枠組み内でジョブのタスクの遠隔実行を Sun Grid Engine, Enterprise Edition システムから制御することを可能にする (300 ページの「並列環境と Sun Grid Engine, Enterprise Edition ソフトウェアの密統合」を参照)。

これらのすべての機能のおかげで、qrsh は、qtcsh や qmake 機能を実現するばかりでなく、MPI や PVM などの並列環境と Sun Grid Engine, Enterprise Edition とを密に統合することを可能にする重要な基盤になります。

▼ qrsh を使用して透過的に遠隔実行する

- **以下の説明に従い適切なオプションと引数を追加して、qrsh コマンドを入力します。**

```
% qrsh [options] program|shell-script [arguments] \
    [> stdout_file] [>&2 stderr_file] [< stdin_file]
```

qrsh は qsub のほぼすべてのオプションを認識し、qsub にはないオプションもいくつか提供します。

- - now yes|no - 対話形式のジョブをただちにスケジューリングして、適切な資源が使用できない場合は拒否するようにするかどうか、言い替えれば、実行依頼時に開始できない場合にバッチジョブのようにキューに入れるかどうかを制御します。通常、対話形式のジョブに適したオプションで、デフォルトでは yes です。
- - inherit - qrsh はジョブのタスクを開始する際 Sun Grid Engine, Enterprise Edition のスケジューリングプロセスを経由しませんが、指定された遠隔実行ホストで適切な資源をすでに確保している並列ジョブのコンテキスト内にスケジューリング情報が埋め込まれていると想定します。この形の qrsh は一般に、qmake、さらには並列環境と密に統合されたシステム内で使用されます。デフォルトでは、外部ジョブ資源は継承されません。
- - noshell - qrsh に与えられたコマンド行の実行をユーザーのログインシェル内で開始せず、ラッピングシェルなしでコマンド行を実行します。シェルの起動、シェルリソースファイルの供給などのオーバーヘッドが回避されるため、このオプションを使用して実行速度を上げることできます。
- - nostdin - 入力ストリーム STDIN を抑止します。この場合、qrsh は rsh (1) コマンドに -n オプションを渡します。これは、qrsh を使用し、たとえば make(1) プロセスで複数のタスクを並列実行する場合に特に有用です。どのプロセスが入力を得るのかは、定義されません。
- - verbose - このオプションは、スケジューリングプロセスで出力を行うことを表します。主としてデバッグの用途に使用されるため、デフォルトでは無効になっています。

qtcsh を使用した透過的なジョブ分散

qtcsh は広く知られ、かつ使用されている UNIX の C シェル (csh) の高機能版の tcsh を基にした、tcsh と完全互換のコマンドです (tsch の関わりについての詳細は、<*SGE_ROOT*>/3rd_party で提供している情報を参照)。qtcsh はコマンドシェルの機能を拡張し、Sun Grid Engine, Enterprise Edition を使用して、負荷の小さい

適切なホストに指定されたアプリケーションの実行を透過的に分散できるようにします。遠隔実行するアプリケーションおよび実行ホストの選択条件は、.qtask という構成ファイルで定義します。

Sun Grid Engine, Enterprise Edition へのアプリケーションの実行依頼は、 qrsh 機能を使用してユーザーに透過的に行われます。qrsh には、標準出力、標準エラー出力、標準入力処理ばかりでなく、遠隔実行対象のアプリケーションとの端末制御接続も用意されているため、そうしたアプリケーションをシェルと同じホスト上でローカル実行することと、遠隔実行することとの間に、目立った相違点は 3 つしかありません。

- アプリケーションをまったく実行でないわけでないにしても、ローカルホストよりリモートホストの方が、能力、負荷、必要なハードウェア / ソフトウェアの有無の点でずっと適している可能性がある。とうぜん、これは望ましい相違点です。
- ただし、ジョブの遠隔起動と Sun Grid Engine, Enterprise Edition による処理のために、わずかな遅延がある。
- 管理者が、対話形式のジョブ (qrsh)、つまり qtcsh による資源の利用を制限できる。qrsh 機能を使用して起動するアプリケーション用の適切な資源が十分にないか、適切なシステムがすべて過負荷状態の場合は、暗黙の qrsh の実行依頼ができず、対応するエラーメッセージが返されます (「Not enough resources ... try later (十分な資源がありません ... 後でやり直してください)」)。

こうした標準的な用途のほかに、qtcsh はサン以外のコードおよびツールとの統合にも適したプラットフォームです。統合環境において単一アプリケーション実行形式の qtcsh -c *appl_name* で qtcsh を使用することによって、ほぼ変更する必要のない統一的なインタフェースを実現できます。.qtask ファイルで適切に定義することによって、必要なアプリケーション、ツール、統合、サイト、ユーザー固有の構成までのすべての情報を含めることができます。このインタフェースをあらゆる種類のスクリプト、C プログラム、さらには Java アプリケーションからも使用できるという、また別の利点もあります。

qtcsh の使用法

qtcsh の起動は tcsh の起動とまったく同じです。qtcsh は .qtask ファイルをサポートし、一群の特殊なシェル組み込みモードを提供することによって tcsh の機能を拡張します。

.qtask ファイルは以下のように定義します。各行の形式は次のとおりです。

```
% [!]appl_name qrsh_options
```

先行感嘆符 (!) は省略可能で、クラスタ全体のグローバル .qtask ファイルと qtcsh ユーザーの個人用 .qtask ファイルとの間に矛盾する定義がある場合の優先順位を定義します。グローバルファイルに感嘆符がない場合は、ユーザーファイル内の最終的に矛盾する定義が優先し、グローバルファイルに感嘆符がある場合、その定義は書き換えられません。

行の以降の部分には、アプリケーション名 (qtcsh のコマンド行にこのアプリケーション名を入力すると、Sun Grid Engine, Enterprise Edition にそのアプリケーションの遠隔実行が依頼される) と qrsh 機能のオプション (アプリケーションに使用し、その資源要求を定義するオプション) を指定します。

注 – コマンド行に入力するアプリケーション名は、.qtask ファイルに定義した名前と完全に同じである必要があります。名前の前に絶対または相対ディレクトリ指定がある場合は、ローカルのバイナリで、遠隔実行の依頼ではないとみなされます。

注 – csh の別名は、アプリケーション名と比較する前に展開されます。遠隔実行するアプリケーション名は、qtcsh のコマンド行の任意の位置、具体的には、標準入出力のリダイレクトの前後のどちらにも置くことができます。

このため、次の例は正当で意味のある構文です。

```
# .qtask file
netscape -v DISPLAY=myhost:0
grep -l h=filesurfer
```

.qtask がこのようになっている場合に、次の qtcsh コマンド行を入力すると、

```
netscape
~/mybin/netscape
cat very_big_file | grep pattern | sort | uniq
```

暗黙で次のようになります。

```
qrsh -v DISPLAY=myhost:0 netscape
~/mybin/netscape
cat very_big_file | qrsh -l h=filesurfer grep pattern | sort | uniq
```

qtcsh は、オンまたはオフに設定可能なスイッチに応じてさまざまなモードで動作することができます。

- コマンドのローカルまたは遠隔実行 (デフォルトは遠隔)
- 即時またはバッチ遠隔実行 (デフォルトは即時)

- 詳細または簡易出力 (デフォルトは簡易)

これらのモードの設定は、起動時に qtcsh のオプション引数を使用するか、実行時にシェル組み込みコマンドの qrshmode を使用して変更することができます。詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qtcsh の項を参照してください。

qmake を使用した並列メークファイル処理

qmake は UNIX 標準の make 機能の代わりに使用できるコマンドです。機能的には make を拡張して、一群の適切なマシンに個々の make ステップを分散できるようにします。qmake は、一般的な GNU のメーク機能、qmake を基にしています。qmake の関わりについての詳細は、<*sge_root*>/3rd_party で提供している情報を参照してください。

複雑な分散 make プロセスが最後まで実行されるよう、qmake はまず並列ジョブのような形態で必要な資源を確保します。そして、Sun Grid Engine, Enterprise Edition スケジューリング機能と対話することなく、それらの資源を管理します。qmake は、資源が使用可能になるか、-inherit オプションを有効にした qrsh 機能を使用して使用可能にされると、make のステップを分散します。

qrsh には、標準出力、標準エラー出力、標準入力処理ばかりでなく、遠隔実行対象のアプリケーションとの端末制御接続も用意されているため、make プロシージャをローカル実行することと、qmake を使用することとの間に、目立った相違点は 3 つしかありません。

- 個々の make ステップが一定時間持続し、十分な数のステップがある場合は、make プロセスの並列化の処理が大幅に高速化される。当然、これは望ましい相違点です。
- 遠隔実行する make ステップでは、qrsh および遠隔実行を原因とする暗黙の小さなオーバーヘッドが伴う。
- qmake の make ステップ分散を活用するには、最低条件の 1 つとして並列化の度合い、すなわち、並行実行可能な make ステップ数を指定する必要がある。同時に、使用可能なソフトウェアライセンス、マシンのアーキテクチャ、必要なメモリーまたは CPU 時間などの、make ステップが必要とする資源特性を指定することもできます。

一般に、make の最も一般的な用途が複雑なソフトウェアパッケージのコンパイルであることは確実です。しかし、これは qmake の第一の用途ではないかもしれません。プログラムファイルはしばしばかなり小さく (優れたプログラミング慣行の問題)、このため、1 つのプログラムファイルのコンパイルが単一の make ステップで構成され、数秒で終えてしまうことがよくあります。また、通常、コンパイルは多くのファイルアクセス (入れ子のインクルードファイル) を意味し、すべてのファイルアクセスを効果的にシリアル化するときにファイルサーバーがネックになることがある

ため、並列で複数の make ステップを実行した場合は、高速化されないことがあります。このため、コンパイルプロセスで満足のいく高速化を期待できないことがあります。

qmake には、これ以外にもっと適切な用途が考えられます。たとえばメークファイルを使用した複雑な分析業務の相互依存性とワークフローの制御です。これは EDA などの一部分野で一般的であり、一般にそうした環境では、個々の make ステップは、無視できないほどの資源および計算時間を必要とするシミュレーションあるいはデータ分析操作になります。そうした場合は、かなりの高速化を達成することができます。

qmake の使用法

qmake のコマンド行構文は、qrsh と非常によく似ています。

```
% qmake [-pe pe_name pe_range][further options] \
         -- [gnu-make-options][target]
```

注 – この節で後で説明するように、qmake では -inherit オプションも使用できます。

---

qmake の -pe オプションの使用法と、その -j オプションとの関係には特別な注意を払う必要があります。これらのオプションはともに、実現する並列化の量の指定に使用することができます。違いは、-j オプションでは、使用する並列環境などの情報を指定することができないことです。このため、-j オプションでは、qmake は並列メーク用のデフォルトの環境 (make という) が構成されているものと仮定します。また、qmake の -j では、範囲の指定もできず、指定できるのは数字 1 つだけです。qmake は、-j に指定された数字を 1 からその数字までの範囲とみなします。これに対し、-pe では、これらのパラメータすべてを詳細に指定することができます。したがって、次のコマンド行の例は同じことです。

```
% qmake -- -j 10
% qmake -pe make 1-10 --
```

-j オプションを使用して、次のコマンド行と同じものを表すことはできません。

```
% qmake -pe make 5-10,16 --
% qmake -pe mpi 1-99999 --
```

構文とは別に、qmake は、コマンド行からの対話モード (-inherit なし) とバッチジョブ内 (-inherit あり) の 2 通りの起動モードをサポートしています。モードによって、開始される処理シーケンスは異なります。

- **対話** - コマンド行から qmake を起動すると、qmake コマンド行に指定された資源要求を考慮しながら、qrsh によって暗黙で make プロセスが Sun Grid Engine, Enterprise Edition に実行依頼されます。Sun Grid Engine, Enterprise Edition は、並列 make ジョブに関連付けられている並列ジョブ実行用のマスターマシンを選択し、そこで make プロシージャを開始します。この選択が必要なのは、make プロセスがアーキテクチャに依存している可能性があり、必要なアーキテクチャが qmake コマンド行に指定されているためです。マスターマシン上の qmake プロセスは、個々の make ステップの実行を他のホストに分散します。それらのホストは、Sun Grid Engine, Enterprise Edition によってそのジョブ用に事前に割り当てられ、並列環境 hosts ファイルを使用してその情報が qmake に渡されます。
- **バッチ** - この場合、qmake は、バッチスクリプト内に -inherit オプションとともに現れます (-inherit オプションが存在しない場合は、上記の最初の例で説明しているように新しいジョブが生成されます)。こうした qmake は、それが埋め込まれているジョブにすでに割り当てられている資源を利用し、qrsh -inherit を直接使用して、make ステップを開始します。バッチモードでの qmake の呼び出しで資源要求や -pe、-j オプションを指定しても、無視されます。

注 – 単一 CPU のジョブも、並列環境を要求する必要があります (qmake -pe make 1 --)。並列実行が必要ない場合は、Sun Grid Engine, Enterprise Edition オプションと "-" のない gmake コマンド行構文で qmake を呼び出してください。この場合、qmake は gmake のように動作します。

---

qmake についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qmake の項を参照してください。

Sun Grid Engine, Enterprise Edition のジョブスケジューリング方法

Sun Grid Engine, Enterprise Edition のポリシー管理は自動的にクラスタ内の共有資源の使用を制御して、最高度にその運用目標を達成できるようにします。優先順位の高いジョブは優先的にディスパッチされて、より資源を利用できるようになります。Sun Grid Engine, Enterprise Edition クラスタの管理では、高度な資源利用ポリシーを定義することができます。定義可能なポリシーは以下のとおりです。

- **業務優先 (Functional)** - 特定のユーザーグループ、プロジェクトなどとの関係で特別な扱いを受けられます。
- **基本割当 (Share-based)** - このポリシーにおけるサービスレベルは、割り当てられた資源利用資格、他のユーザーおよびユーザーグループの対応する利用資格、全ユーザーの過去の資源利用、システム内の現在のユーザーの有無に依存します。

- **締め切り優先 (Deadline)** - 特定の時点またはその前に完了する必要があるジョブは、その条件を達するために特別な扱いが必要になることがあります。
- **一時優先 (Override)** - このポリシーでは、Sun Grid Engine, Enterprise Edition クラスタの管理者による手動の介入で、自動的なポリシー実施を変更します。

Sun Grid Engine, Enterprise Edition システムが日常的に基本割当ポリシーか業務優先ポリシー、またはその両方を使用するように構成することができます。これらのポリシーは、0 から 1 の範囲で重み与えたり、2 つ目だけ使用して両方に同じ重みを与えたりなどの任意の比率で組み合わせることができます。

これらの定期ポリシーのほかに、締め切り優先でジョブの実行を依頼することもできます (84 ページの「高度な設定」の締め切り優先実行依頼パラメータの説明を参照)。締め切り優先ジョブは、定期スケジューリングに影響します。管理者はまた、一時的にあるいは基本割当や業務優先、締め切り優先スケジューリングを無効にすることもできます。一時優先は、特定の 1 つジョブに対して適用することも、特定のユーザー、部署、プロジェクト、ジョブクラス (すなわち、キュー) に関連付けられているすべてのジョブに適用することもできます。

ジョブの優先順位

Sun Grid Engine, Enterprise Edition には、すべてのジョブ間の調停をするためのこれら 4 つのポリシーのほかに、ユーザーが自分のジョブに優先順位を設定する機能もあります。たとえばユーザーが 3 つのジョブを実行依頼しようとしていて、ジョブ 3 が最重要で、ジョブ 1 と 2 の重要性は同じであると仮定します。

注 – これが可能なのは、Sun Grid Engine, Enterprise Edition のポリシーの組み合わせに、業務優先カテゴリの「ジョブ」に配分が割り当てられた業務優先ポリシーが含まれている場合だけです。

ジョブの優先順位は、`QMON` の一般ジョブ実行依頼画面のパラメータの優先順位 (図 4-10 を参照) か、 `qsub` の `-p` オプションを使用して設定します。設定できる優先順位は、-1024 (最低) から 1023 (最高) の範囲です。この優先順位は 1 人のユーザーのジョブの間でジョブをランク付けします。Sun Grid Engine, Enterprise Edition スケジューラは、この値によって、単一ユーザーのジョブがシステムに複数存在する場合にその選択方法を決定します。特定のジョブに割り当てられる相対的な重要性は、そのユーザーのジョブに割り当てされた最高および最低の優先順位と、そのジョブの優先順位値に依存します。

チケット

スケジューリングポリシーは、チケットを使用して実現されます。各ポリシーにはチケットプールがあり、そこから、複数マシンからなる Sun Grid Engine, Enterprise Edition システムに入るジョブにチケットが割り当てられます。定期ポリシーを有効

にすると、新規ジョブの 1 つ 1 つにチケットが割り当てられ、スケジューリングのたびに実行中のジョブへのチケットの再割り当てが試みられます。以下では、各ポリシーがチケットを割り当てるときに使用する基準を説明します。

チケットは 4 つのポリシーに重みを付けます。たとえば業務優先ポリシーにチケットが割り当てられていない場合、業務優先ポリシーは使用されません。業務優先と基本割当チケットプールに同数のチケットが割り当てられている場合、両方のポリシーはジョブの重要性の決定に際して同等の重みを持ちます。

システム構成時、Sun Grid Engine, Enterprise Edition のマネージャーはこれらの定期ポリシーにチケットを割り当てます。その後、マネージャーおよびオペレータはいつでもチケット割当量を変更して、すぐに有効にすることができます。締め切り優先または一時優先を指示するには、システムに一時的に追加チケットを注入します。ポリシーはチケットの割り当てによって組み合わせられます。複数のポリシーにチケットが割り当てられている場合、ジョブは有効な各ポリシーにおけるその重要性に応じて割当分のチケットを受け取ります。

Sun Grid Engine, Enterprise Edition は、システムに入るジョブにチケットを付与することによって、有効な各ポリシーにおけるその重要性を指示します。実行中のジョブは、スケジューリングのたびにチケットが増えることもあれば (たとえば優先指定が行われたり、締め切りが迫っていたりするなどの理由)、減ることもあり (たとえば、正当な量を超える資源配分を受けているなどの理由)、同じチケット数が維持されることもあります。ジョブが保持するチケット数は、Sun Grid Engine, Enterprise Edition がスケジューリング中にそのジョブに付与しようとする資源配分量を表します。

そして、このチケット数は `QMON` (121 ページの「`QMON` からジョブを監視、制御する」を参照) または `qstat -ext` を使用して表示することができます。この `qstat` コマンドは、たとえば `-qsub -p` でジョブに割り当てられた優先順位値も表示します (`qstat` についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。

キューの選択

ジョブに特定のキューの要求がなく、ただちに開始できない場合、Sun Grid Engine, Enterprise Edition はそのジョブをディスパッチしません。そうしたジョブには、`sge_qmaster` でスプール済みのマークが付けられ、`sge_qmaster` はその時々に再スケジューリングを試みます。つまり、そうしたジョブは、次に適切なキューが使用可能になったときに、そのキューにディスパッチされます。

これに対し、名前で特定のキューを要求しているジョブは、開始可能かどうか、あるいはスプールする必要があるかどうかに関係なく、直接そのキューに移動します。このため、Sun Grid Engine, Enterprise Edition のキューを情報科学として見ると、バッチキューは名前で要求するジョブに対してのみ意味を持ちます。具体的な要求を付けずに実行依頼されたジョブは、`sge_qmaster` のスプール機能を使用して待機状態になります (より抽象的で柔軟なキュー概念の採用)。

ジョブがスケジューリングされて、複数の空きキューがその資源要求を満たしている場合、通常、ジョブは適切なもののうち、負荷が最も軽いホストに属するキューにディスパッチされます。Sun Grid Engine, Enterprise Edition のクラスタ管理で、その構成パラメータの `queue_sort_method` を `seq_no` に設定することによって、この負荷依存方式は固定順序アルゴリズムに変更することができます。これに対し、キュー構成パラメータの `seq_no` は、最小の連続番号のキューに最高の優先順位を割り当てる、キューの間の優先順位を定義する目的に使用されます。

第5章

チェックポイントジョブとジョブの監視、制御

Sun Grid Engine, Enterprise Edition 5.3 システムを使用してジョブの実行依頼をしたら、それらのジョブを監視、制御します。この章では、そうした作業に関する予備知識的な情報と実施方法を説明します。

具体的には、この章では以下の作業を行う方法を説明しています。

チェックポイントジョブ

この節では、ジョブのチェックポイントを生成する次の 2 つのレベルの機能について説明します。

- ユーザーレベル
- カーネルレベル

ユーザーレベルのチェックポイント機能

多くのアプリケーションプログラム、特に、通常かなりの CPU 時間を消費するプログラムには、障害に対する強度を高めるチェックポイント生成機能とチェックポイント再開機能が実装されています。このアルゴリズムでは、いくつかの段階でステータス情報と処理データの重要部分が繰り返しファイルに書き出されます。そうしたファイル (チェックポイントまたは再開ファイルという) は、アプリケーションが異常終了して後で再起動された場合に処理され、チェックポイント生成直前の状況に相当する、安定した状態に戻ることを可能にします。たいていの場合、チェックポイントファイルを処理するには、それらのファイルを適切な場所に移動する必要があるため、この種のチェックポイント機能はユーザーレベルのチェックポイントと呼ばれます。

アプリケーションプログラムにユーザーレベルのチェックポイント機能が組み込まれていない場合、その代替機能として、パブリックドメイン (たとえばウィスコンシン大学の Condor プロジェクトを参照) あるいは一部ハードウェアベンダーから入手可能な、いわゆるチェックポイントライブラリを利用することができます。そうしたライブラリにアプリケーションを再リンクすると、ソースコードを変更しなくても、アプリケーションにチェックポイント機能が組み込まれます。

カーネルレベルのチェックポイント機能

一部のオペレーティングシステムには、そのカーネル内にチェックポイントのサポート機能が用意されています。この場合、アプリケーションプログラムでの準備やアプリケーションの再リンクは必要ありません。カーネルレベルのチェックポイント機能は、通常、個別プロセス単位ばかりでなく、プロセス階層全体にも使用することができます。すなわち、相互に依存するプロセスの階層構造全体のチェックポイントを生成して、後で再開することができます。通常、カーネルレベルのチェックポイント機能には、チェックポイントを開始する機能としてユーザーコマンドと C ライブラリインタフェースの両方が用意されています。

オペレーティングシステムがチェックポイント機能を提供している場合、Sun Grid Engine, Enterprise Edition はそのチェックポイント機能をサポートします。現在サポートしているチェックポイント機能については、『Sun Grid Engine, Enterprise Edition 5.3 ご使用にあたって』を参照してください。

チェックポイントジョブの移動

再開時に繰り返す作業を少なくするため、チェックポイントジョブはいつでも実行を中断できるようになっています。Sun Grid Engine, Enterprise Edition の移動および動的負荷均衡は、この機能に基づいています。要求があると、チェックポイントジョ

ブは中止され、Sun Grid Engine, Enterprise Edition プールの他のマシンに移動されることによってクラスタ内の負荷は動的に平均化されます。チェックポイントジョブが中止、移動される理由としては、以下があります。

- 実行キューまたはジョブが、qmod または qmon コマンドによって明示的に一時停止された。
- ジョブのチェックポイント生成タイミングの指定に一時停止が指定されていて、実行キューが一時停止しきい値を超えたため、キューまたはジョブが自動的に一時停止された (175 ページの「負荷および一時停止しきい値を設定する」の節と、118 ページの「コマンド行からチェックポイントジョブを実行依頼、監視、削除する」の節を参照)。

移動しているジョブは sge_qmaster に戻り、別の適切なキューが使用可能になると、そのキューにディスパッチされます。その場合、qstat の出力にはステータスとして R が表示されます。

チェックポイントジョブスクリプトの作成

カーネルレベルのチェックポイント生成用シェルスクリプトと、通常のシェルスクリプトとの間に違いはありません。

ユーザーレベルのチェックポイント生成用シェルスクリプトは、再開された場合にその処理を正しく行う機能があるという点で通常の Sun Grid Engine, Enterprise Edition バッチスクリプトと異なります。チェックポイントジョブが再開されると、RESTARTED 環境変数が設定されます。この情報に基づいて、最初の起動中にだけ実行されるジョブスクリプトの部分を省略することができます。

このため、透過的なチェックポイントジョブスクリプトは、コード例 5-1 のようになります。

```
#!/bin/sh
#Force /bin/sh in Sun Grid Engine, Enterprise Edition
#$ -S /bin/sh
# Test if restarted/migrated
if [ $RESTARTED = 0 ]; then
    # 0 = not restarted
    # Parts to be executed only during the first
    # start go in here
    set_up_grid
fi
# Start the checkpointing executable
fem
#End of scriptfile
```

コード例 5-1　　チェックポイントジョブスクリプトの例

ユーザーレベルのチェックポイントジョブが移動された場合は、ジョブスクリプトの先頭から再開されることに注意してください。シェルスクリプトの実行の流れをジョブの実行中断場所に移し、複数回の実行に不可欠な、スクリプト内の行をスキップするのは、ユーザーの責任です。

注 – カーネルレベルのチェックポイントジョブはいつでも中断可能で、最後のチェックポイント発生位置から外側のシェルスクリプトを再開します。このため、カーネルレベルのチェックポイントジョブには、RESTARTED 環境変数は関係ありません。

▼ コマンド行からチェックポイントジョブを実行依頼、監視、削除する

適切なスイッチを付けて次のコマンドを入力します。

```
#qsub options arguments
```

チェックポイントジョブの実行依頼は、qsub の -ckpt および -c スイッチを除けば、通常のバッチスクリプトの実行依頼と同じです。これらのオプションはチェックポイント機能を要求し、ジョブに対してチェックポイントを生成するタイミングを定義します。-ckpt オプションは、使用するチェックポイント環境名を示す引数を 1 つとります (281 ページの「チェックポイント機能のサポート」を参照)。-c オプションは必須ではなく、やはり引数を 1 つとります。このオプションを使用して、チェックポイント環境構成内の when パラメータの定義を書き換えることができます (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の checkpoint の項を参照)。

-c オプションに対する引数には、次の英字 1 字 (またはその任意の組み合わせ) か、時間値を指定することができます。

- n - チェックポイント機能を実行しない。最高の優先順位になります。
- s - ジョブのホスト上の sge_execd が停止した場合にのみチェックポイントを生成する。
- m - 対応するキュー構成に定義されている最小 CPU 間隔でチェックポイントを生成する (queue_conf のマニュアルページの min_cpu_interval パラメータを参照)。
- x - ジョブが一時停止された場合にチェックポイントを生成する。

- `interval` - 指定された間隔 (ただし、上記の `min_cpu_interval` に定義された回数を超えない間隔)でチェックポイントを生成する。時間値は hh:mm:ss (それぞれ 2 桁の時、分、秒値をコロンで区切る) の形式で指定します。

チェックポイントジョブの監視が通常のジョブの監視と異なるのは、ジョブがときどき移動することがあり、1 つのキューに拘束されないということだけです。ただし、ジョブ名とともに一意のジョブ ID 番号は維持されます。

チェックポイントジョブは、134 ページの「コマンド行からの Sun Grid Engine, Enterprise Edition ジョブの制御」の節で説明しているのと同じ方法で削除できます。

▼ QMON からチェックポイントジョブの実行依頼をする

- **以下の注意点を考慮しながら、84 ページの「高度な設定」の手順に従って操作を行います。**

QMON からのチェックポイントジョブの実行依頼は、適切なチェックポイント環境を指定することを除けば、通常のバッチジョブの実行依頼と同じです。84 ページの「高度な設定」の節で説明したように、「ジョブの実行依頼」ダイアログボックスには、ジョブに関連付けるチェックポイント環境を指定するための入力フィールドがあります。この入力フィールドの横にアイコンボタンがあり、このボタンをクリックすると、図 5-1 に示すような選択用のダイアログボックスが開きます。このダイアログボックスの使用可能なチェックポイント環境のリストから適切な環境を選択することができます。実際に組み込まれているチェックポイント環境のプロパティについては、システム管理者にお尋ねください。また、281 ページの「チェックポイント機能のサポート」を参照してください。

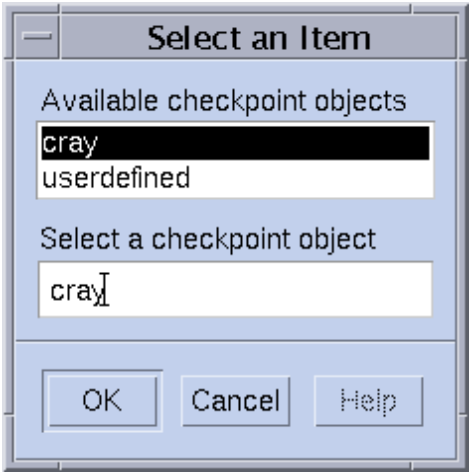


図 5-1　　チェックポイントオブジェクトの選択

ファイルシステム要件

チェックポイントライブラリを使用したユーザーまたはカーネルレベルのチェックポイント機能の実行では、チェックポイント生成対象のプロセスまたはジョブがカバーする仮想メモリーの完全なイメージをダンプする必要があります。このため、十分なディスク領域が必要です。チェックポイント環境構成パラメータの `ckpt_dir` を設定している場合、チェックポイントデータは、`ckpt_dir` の下のジョブ専用の場所にダンプされます。`ckpt_dir` が `NONE` に設定されている場合は、チェックポイントジョブが開始されたディレクトリが使用されます。チェックポイント環境の構成についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `checkpoint` の項を参照してください。

注 – `ckpt_dir` を `NONE` に設定している場合は、`qsub` の `-cwd` オプションを使用してチェックポイントジョブを開始してください。

ファイルシステムについては、満たす必要がある条件がもう 1 つあります。すなわち、ジョブが正しく移動、再開されるには、すべてのマシンからチェックポイント関係のすべてのファイルが見えるようになっている必要があり、このためには、NFS か類似のファイルシステムが必要になります。この条件が満たされているかどうかについては、クラスタ管理者にお尋ねください。

NFS が使用されていないか、何らかの理由でその使用が望ましくない場合は、シェルスクリプトの開始時に `rcp` または `ftp` などを使用して明示的にチェックポイントファイルを転送する必要があります (ユーザーレベルのチェックポイントジョブの場合)。

Sun Grid Engine, Enterprise Edition ジョブの監視と制御

基本的に、実行依頼したジョブを監視する方法は以下の 3 通りあります。

- Sun Grid Engine, Enterprise Edition グラフィカルユーザーインタフェースの `QMON` を使用する。
- コマンド行から `qstat` コマンドを使用する。
- 電子メールを使用する。

▼ QMON からジョブを監視、制御する

Sun Grid Engine, Enterprise Edition のグラフィカルユーザーインタフェースの `QMON` には、ジョブ制御専用のダイアログボックスがあります。

- **`QMON` メインメニューで「ジョブ制御」ボタンをクリックし、この後の説明に従って操作を進めます。**

「ジョブ制御」ダイアログボックスの一般的な目的は、システムが認識している実行中か保留中、または完了ジョブのすべてまたは一部を監視する手段を提供することにあります。このダイアログボックスはまた、ジョブの操作、すなわち、その優先順位の変更や一時停止、再開、取り消しにも使用することができます。「ジョブ制御」ダイアログボックスには、実行中ジョブと、保留中ジョブ (適切な資源へのディスパッチ待ち)、最近完了したジョブ用の 3 つのリストがあります。これら 3 つのリストの表示は、ダイアログボックス上部のタブをクリックすることによって切り替えることができます。

図 5-2 に示すデフォルトの形式では、実行中または保留中のジョブのすべてに、「ジョブ ID」と「優先順位」「ジョブ名」「キュー」が表示されます。

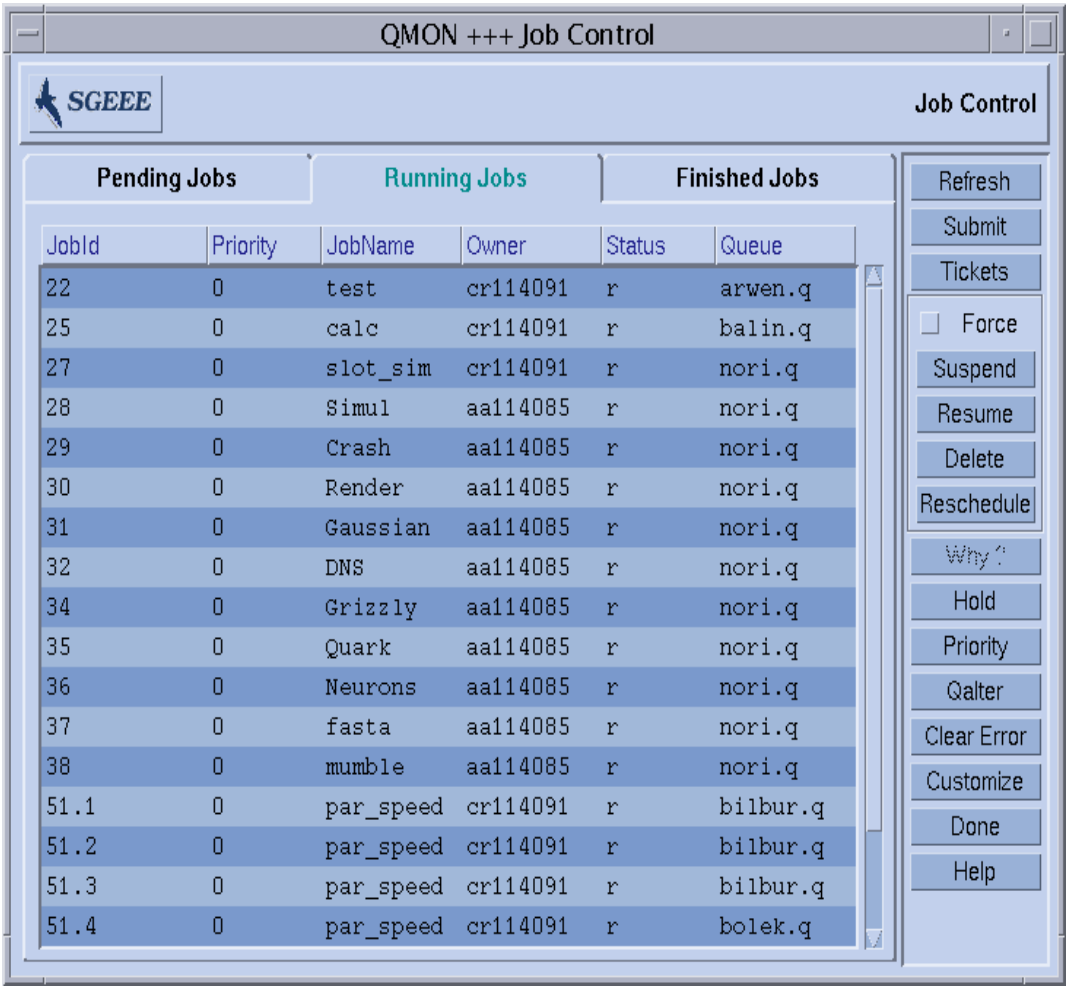


図 5-2　「ジョブ制御」ダイアログボックス - 標準形式

リストに表示するフィールド (列) は、「ジョブ制御」ダイアログボックスで「カスタマイズ」ボタンをクリックすると開く「カスタマイズ」ダイアログボックスが使用して設定することができます。

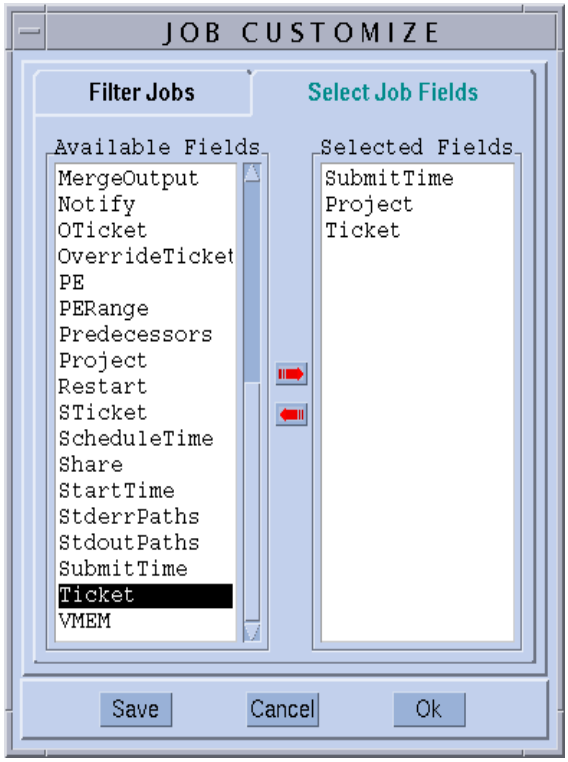


図 5-3　　ジョブ制御の「カスタマイズ」ダイアログボックス

「カスタマイズ」ダイアログボックスでは、表示する Sun Grid Engine, Enterprise Edition ジョブオブジェクトのエントリを選択したり、表示するジョブをフィルタで選択したりすることができます。図 5-3 の例では、追加フィールドとして「プロジェクト」「チケット」「実行依頼時間」が選択されています。

図 5-4 は、「完了ジョブ」リストにカスタマイズ内容を適用した後の「ジョブ制御」ダイアログボックスを示しています。

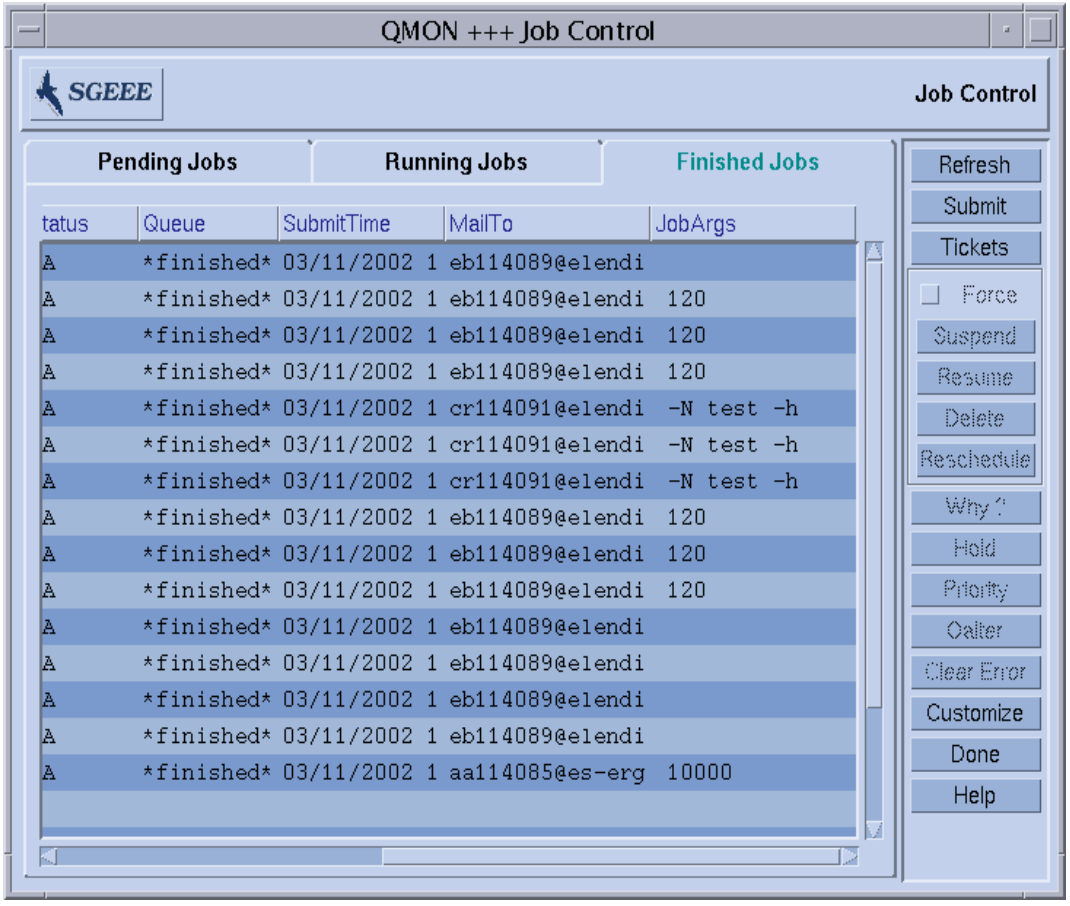


図 5-4　「ジョブ制御」ダイアログボックスの完了ジョブの表示 - 拡張後

図 5-5 のフィルタ機能の例では、`chaubal` が所有するジョブで、アーキテクチャ solaris で実行されるか、`solaris` に適したジョブだけを表示するように選択しています。

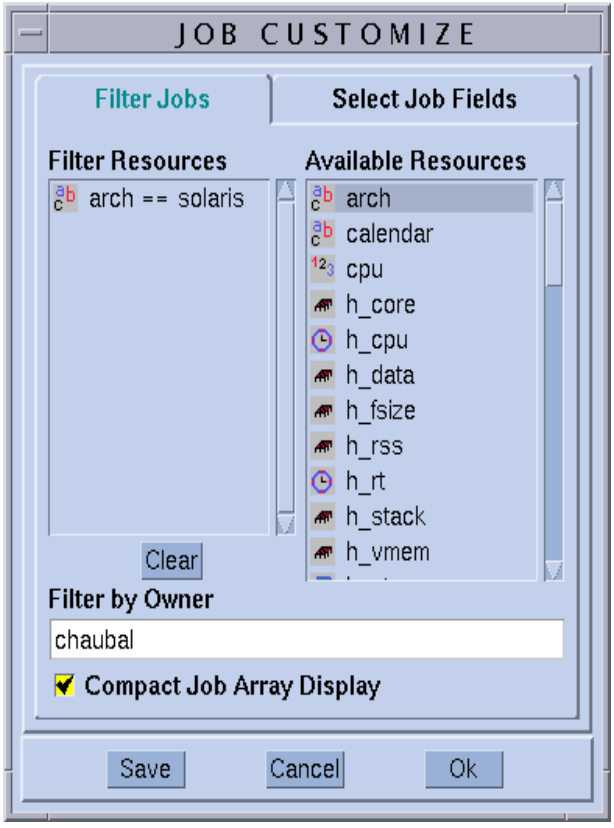


図 5-5　　ジョブ制御のフィルタ機能

図 5-6 は、実行中のジョブに上記のフィルタを適用したときの「ジョブ制御」ダイアログボックスを示しています。

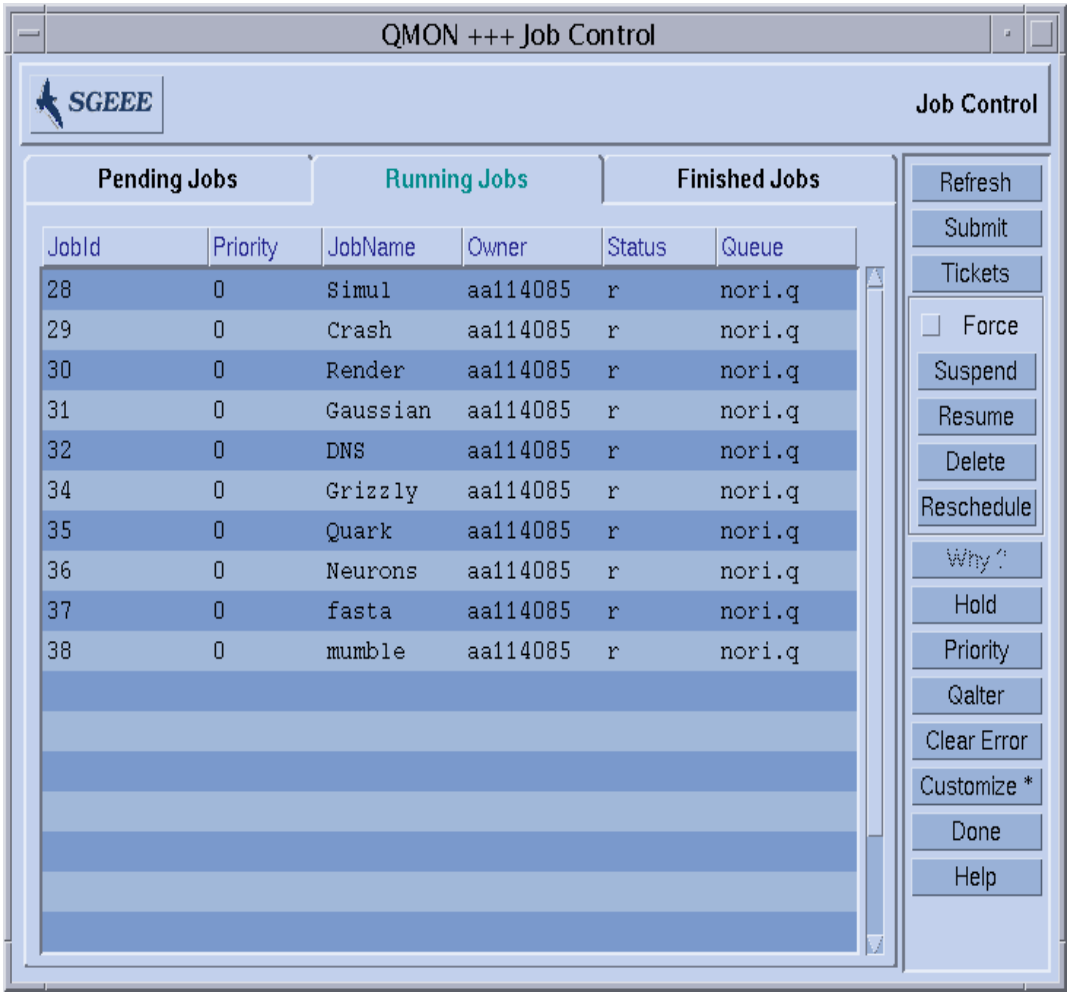


図 5-6　「ジョブ制御」ダイアログボックス - フィルタの適用後

注 – 図 5-3 の「カスタマイズ」ダイアログボックスにある「**保存**」ボタンをクリックすると、そのユーザーのホームディレクトリの `.qmon_preferences` ファイルにカスタマイズ内容が保存され、「ジョブ制御」ダイアログボックスのデフォルトの外観が再定義されます。

図 5-6 の「ジョブ制御」ダイアログボックスは、QMON で配列ジョブを表示している例でもあります。

操作対象にするジョブは、次のマウスとキーの組み合わせ操作で選択することができます。

- Control キーを押しながら、マウスの左ボタンでジョブを選択すると、複数のジョブの選択開始になります。
- Shift キーを押しながら、マウスの左ボタンで別のジョブを選択すると、選択を開始したジョブから現在のジョブまでのすべてのジョブが選択されます。
- Control キーを押しながら、マウスの左ボタンでジョブを選択すると、そのジョブの選択状態が切り替わります。

選択したジョブは、画面右側の適切なボタンを使用して、一時停止、再開 (停止解除)、削除、ホールド (およびホールド解除)、優先順位変更、変更 (`Qalter`) することができます。

ジョブの一時停止、停止解除、削除、ホールド、優先順位変更、変更操作を行えるのは、そのジョブの所有者か、Sun Grid Engine, Enterprise Edition のマネージャー・オペレータだけです (70 ページの「マネージャーとオペレータ、所有者」を参照)。このうち、一時停止と停止解除は実行中のジョブ、ホールドと変更 (優先順位などの変更) は保留中のジョブにのみ行うことができます。

ジョブを一時停止するということは、UNIX の `kill` コマンドを使用してジョブのプロセスグループに `SIGSTOP` シグナルを送信するとのと同じことを意味し、ジョブは停止して、それ以上 CPU 時間を消費しなくなります。ジョブを停止解除すると、SIGCONT シグナルが送信され、ジョブの実行が再開されます (プロセスへのシグナル送信についての詳細は、`kill` のマニュアルページを参照)。

注 – ジョブの一時停止、停止解除、削除は、たとえばネットワーク上の問題のためにそのジョブを制御している `sge_execd` にアクセスできない場合に、その `sge_execd` に連絡することなく、`sge_qmaster` に登録することによって強制的に行うことができます。このためには、`Force` フラグを使用します。

選択した保留中のジョブに「ホールド」ボタンを使用した場合は、「ホールド設定」サブダイアログボックスが開きます (図 5-7 を参照)。

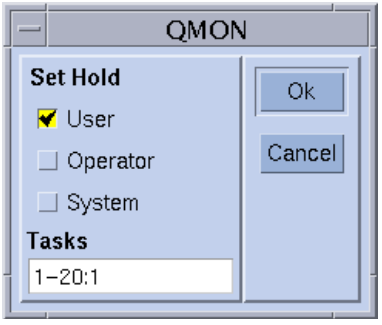


図 5-7 ジョブ制御のホールド

「ホールド設定」サブダイアログボックスでは、ユーザー、システム、オペレータホールドの設定とリセットを行うことができます。ユーザーホールドは、ジョブの所有者ばかりでなく、Sun Grid Engine, Enterprise Edition のオペレータおよびマネージャーも設定またはリセットすることができます。これに対しオペレータホールドはマネージャーとオペレータ、システムホールドはマネージャーだけが設定またはリセットすることができます。ジョブにホールドが設定されている限り、そのジョブが実行対象になることはありません。ホールドを設定またはリセットする方法としては、その他にも `qalter`、`qhold`、`qrls` コマンドを使用する方法があります (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の対応する項を参照)。

「ホールド設定」ボタンの「タスク」フィールドは配列ジョブに適用されます。このボタンを使用して、配列ジョブの特定のタスクをホールドすることができます。図 5-7 の「タスク」フィールドのテキストの書式に注意してください。このフィールドに、タスク ID 範囲として、単一の番号、*n-m* の形式の簡単な範囲、ステップ付きの範囲を指定することができます。たとえば、タスク ID 範囲として `2-10:2` を指定すると、タスク ID が添字 2、4、6、8、10 の合計で 5 つのタスク (それぞれ、この 5 つの添字番号の 1 つを含む *SGE_TASK_ID* 環境変数を持つ) を指定したことになります。ジョブホールドについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qsub の項か `qsub(1)` のマニュアルページを参照してください。

「優先順位」ボタンをクリックすると、別のサブダイアログボックスが開き (図 5-8 を参照)、このダイアログボックスから、選択した保留中または実行中のジョブの新しい優先順位値を入力することができます。Sun Grid Engine, Enterprise Edition では、この優先順位で単一ユーザーの複数のジョブをランク付けします。Sun Grid Engine, Enterprise Edition スケジューラは、この値によって、単一ユーザーのジョブがシステムに複数存在する場合にその選択方法を決定します。

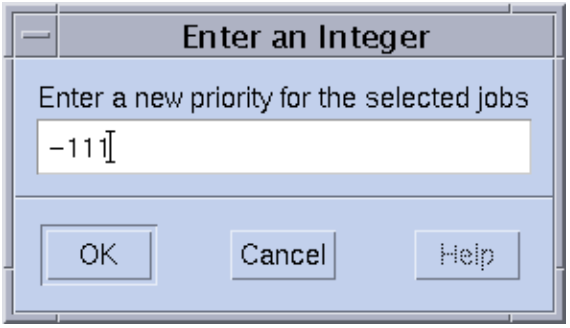


図 5-8 ジョブ制御における優先順位設定

保留中のジョブに対して「Qalter」ボタンをクリックすると、実行依頼で定義されたジョブの属性に応じてダイアログボックスのすべてのエントリが設定された状態で、73 ページの「GUI の `QMON` からジョブの実行依頼をする」で説明している「ジョブの実行依頼」ダイアログボックスが表示されます。それらのエントリのうち、変更ができないエントリは、変更不可表示になっています。それ以外のエントリは変更可能

で、「Qalter」ボタン (「ジョブの実行依頼」ダイアログボックスの「実行依頼」ボタンの働きをする) をクリックすると、変更内容が Sun Grid Engine, Enterprise Edition に登録されます。

「ジョブの実行依頼」ダイアログボックスの「検査」フラグは、Qalter モードのとき特別な意味を持ちます。保留中のジョブの整合性を調べ、スケジューリングされない理由を調べることができます。このためには、「検査」フラグで目的の整合性検査モードを選択し、「Qalter」ボタンをクリックすればよいだけです。選択した検査モードによっては、整合性に問題があることを示す警告が表示されます。詳細は、84 ページの「高度な設定」の節と `qalter` のマニュアルページを参照してください。

ジョブが保留中のままになっている理由を調べるもう 1 つの方法は、「ジョブ制御」ダイアログボックスでジョブを選択して、「**調査**」ボタンをクリックする方法です。「オブジェクトブラウザ」ダイアログボックスが開き、Sun Grid Engine, Enterprise Edition のスケジューラによってその最後のパスでジョブがディスパッチされない理由が一覧表示されます。図 5-9 は、そうしたメッセージを表示しているブラウザ画面の例です。

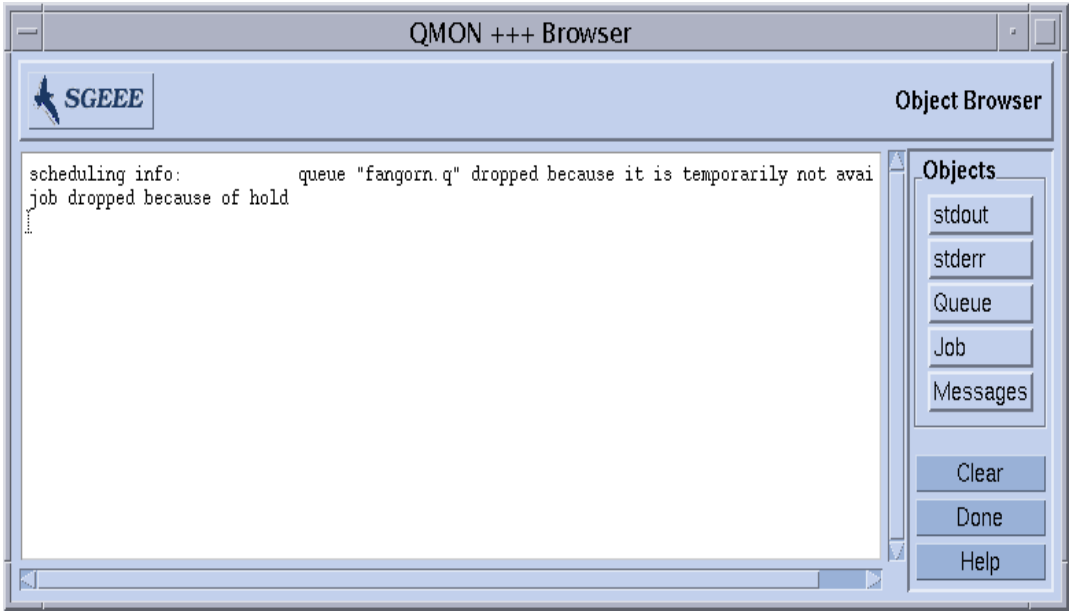


図 5-9 ブラウザに表示されたスケジューリング情報

注 – 「**調査**」ボタンは、スケジューラ構成パラメータの `schedd_job_info` が `true` に設定されている場合にのみ意味のある情報を出力します。表示されるスケジューラ情報は、最新のスケジューリングに関する情報です。ジョブがスケジューリングされない理由を調べるときには、正確でなくなっている可能性があります。

「エラーをクリア」ボタンを使用して、選択されている保留中のジョブのエラー状態を解除することができます。このエラー状態は、以前に開始されたが、ジョブに依存する問題 (たとえば、指定されたジョブ出力ファイルに対する書き込み権限が不足しているなど) が原因で最後まで実行されなかったことを示します。

注 – 保留中のジョブリスト中、エラー状態は赤いフォントで表示されます。エラー状態を解決した後でのみ、`qalter` などを使用して解除するようにしてください。中止された場合に電子メールを送信するようにというジョブの要求がある場合は (たとえば `qsub` の `-m a` オプションを使用)、電子メールでそうしたエラー状態が自動的に報告されます。

つねに最新の情報が表示されるよう、QMON ではポーリング方式を採用して、`sge_qmaster` からジョブの状態を読み出します。「再表示」ボタンをクリックすることによって強制的に更新することもできます。

このボタンは、QMON の「ジョブの実行依頼」ダイアログボックスへのリンクになっています (たとえば図 5-10 を参照)。

QMON のオブジェクトブラウザを使用した追加情報の表示

QMON オブジェクトブラウザを使用して、「ジョブ制御」ダイアログボックスをカスタマイズしなくても (121 ページの「QMON からジョブを監視、制御する」の節で説明)、Sun Grid Engine, Enterprise Edition ジョブに関する追加情報を即座に読み出すことができます。

オブジェクトブラウザは、QMON メインメニューで「ブラウザ」アイコンボタンをクリックすると開きます。ブラウザで「ジョブ」ボタンを選択し、「ジョブ制御」ダイアログボックスでマウスポインタをジョブの行に置くと、ブラウザ画面にそのジョブに関する情報が表示されます (たとえば図 5-2 を参照)。

図 5-10 は、そうした状況でのブラウザ画面の表示例です。

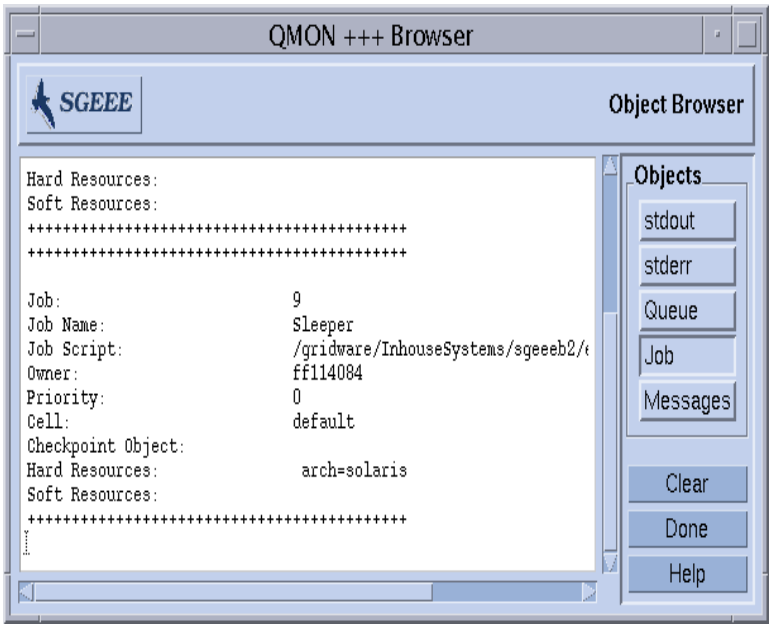


図 5-10 オブジェクトブラウザ - ジョブが選択されている場合

▼ qstat を使用してジョブを監視する

- **この後の説明に従い、コマンド行から次のうちの適切なコマンドを使用します。**

```
% qstat
% qstat -f
% qstat -ext
```

最初の形式は、実行依頼されたジョブだけの概要を提供します (表 5-1 を参照)。2 つ目の形式では、さらに現在構成済みのキューに関する情報が含まれます (表 5-2 を参照)。3 つ目の形式は、最新のジョブ使用状況やジョブに割り当てられているチケットなどの詳細情報を提供します。

最初の形式の出力ヘッダー行は、各列の見出しを示します。これらの列の大部分の意味は、見出しをみるとすぐに解るはずです。「state」列には、英字 1 文字が含まれ、実行中の場合 r、一時停止中の場合 s、キューの場合 q、待機中の場合 w になります (qstate の出力形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qstat の項を参照)。

2 つ目の形式の出力は 2 つのセクションに分かれ、最初のセクションには、使用可能なすべてのキューの状態、- PENDING JOBS - ... のタイトルの付いた 2 つ目のセクションには sge_qmaster ジョブスプール領域の状態が表示されます。キューのセクションの先頭行は、列挙されているキューに対する列の見出しです。キューは横罫線で区切られます。ジョブがキューで実行されている場合は、最初の形式の qstat コマンドと同じ形式で、そのジョブの情報がキューの下に表示されます。また最初の形式の qstat 同様、この 2 つ目のセクションには保留中のジョブの情報も表示されます。

キューの情報の次の列については、少し説明が必要です。

- qtype - キューの種類で、B (バッチ) か I (対話形式)、P (並列)、C (チェックポイント)、またはその組み合わせになります。
- used/free - キューの使用/空きジョブスロット数です。
- states - キューの状態で、u (不明) か a (アラーム)、s (一時停止)、d (使用不可)、E (エラー)、またはその組み合わせになります。

qstat のマニュアルページに、その出力形式についての詳細な説明があります。

Sun Grid Engine, Enterprise Edition 独特の 3 つ目の形式には、次の列で利用状況とジョブに割り当てられているチケット値に関する情報が含まれます。

- cpu/mem/io - 現在の CPU、メモリー、入出力の累積使用量です。
- tckts/ovrts/otckt/dtckt/ftckt/stckt - これらの値は、qalter -ot を使用してジョブに割り当てられているチケットの合計数とポリシー (一時優先、締め切り優先、業務優先、基本割当) ごとの割り当て数を示します。

締め切り優先開始時間が設定されている場合は「deadline」列にその時間が示され、「share」列には、クラスタ内のすべてのジョブの資源利用量に対する各ジョブの現在の資源配分量が表示されます。詳細は、qstat のマニュアルページを参照してください。

qstat のどちらのバージョンにも、このほかにさまざまなオプションがあり、機能を拡張します。-r オプションは、実行依頼されたジョブの資源要求内容を表示します。また、特定のユーザー、特定のキューのみに出力を制限したり、-l オプションを使用して、資源要求を指定したりすることもできます (qsub コマンドに関する 89 ページの「資源要求の定義」の節を参照)。qstat コマンド行で資源要求が指定された場合は、その指定に一致するキュー (およびそのキューで実行されているジョブ) だけが表示されます。

表 5-1 と表 5-2 は、qstat コマンドと qstat -f コマンドの出力例です。

表 5-1　qstat の出力例

job-ID	prior	name	user	state	submit/start at	queue	function
231	0	hydra	craig	r	07/13/96 20:27:15	durin.q	MASTER
232	0	compile	penny	r	07/13/96 20:30:40	durin.q	MASTER
230	0	blackhole	don	r	07/13/96 20:26:10	dwain.q	MASTER
233	0	mac	elaine	r	07/13/96 20:30:40	dwain.q	MASTER
234	0	golf	shannon	r	07/13/96 20:31:44	dwain.q	MASTER
236	5	word	elaine	qw	07/13/96 20:32:07		
235	0	andrun	penny	qw	07/13/96 20:31:43		

表 5-2　qstat -f の出力例

queuename			qtype	used/free	load_avg	arch	states	
dq			BIP	0/1	99.99	sun4	au	
durin.q			BIP	2/2	0.36	sun4		
231	0	hydra		craig	r	07/13/96	20:27:15	MASTER
232	0	compile		penny	r	07/13/96	20:30:40	MASTER
dwain.q			BIP	3/3	0.36	sun4		
230	0	blackhole		don	r	07/13/96	20:26:10	MASTER
233	0	mac		elaine	r	07/13/96	20:30:40	MASTER
234	0	golf		shannon	r	07/13/96	20:31:44	MASTER
fq			BIP	0/3	0.36	sun4		
###								
- PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS -								
###								
236	5	word		elaine	qw	07/13/96	20:32:07	
235	0	andrun		penny	qw	07/13/96	20:31:43	

▼ 電子メールでジョブを監視する

- **この後の説明に従い、コマンド行から適切な引数を付けて次のコマンドを入力します。**

```
#qsub arguments
```

qsub -m スイッチは、特定のイベントが発生した場合に、ジョブの実行依頼をしたユーザーまたは -M フラグで指定された電子メールアドレスに電子メールを送信するよう要求します (フラグについては、qsub のマニュアルページを参照)。-m オプションには、イベントを示す引数を指定します。指定できる引数は次のとおりです。

- b - ジョブの開始でメールを送信
- e - ジョブの終了でメールを送信
- a - qdel コマンドなどによるジョブの中止でメールを送信
- s - ジョブの一時停止でメールを送信
- n - メールの送信なし (デフォルト)

1 つの -m オプションに、コンマで区切ってこれらの引数を複数指定することができます。

QMON の「ジョブの実行依頼」ダイアログボックスで、同じメールイベントを設定することができます。84 ページの「高度な設定」の節を参照してください。

コマンド行からの Sun Grid Engine, Enterprise Edition ジョブの制御

121 ページの「QMON からジョブを監視、制御する」の節では、Sun Grid Engine, Enterprise Edition のグラフィカルユーザーインタフェースの QMON を使用してジョブを削除、一時停止、再開する方法を説明しました。

以下で説明するように、同等の機能をコマンド行から使用することもできます。

▼ コマンド行からジョブを制御する

- **この後の説明に従い、コマンド行から適切な引数を付けて次のいずれか適切なコマンドを入力します。**

```
% qdel arguments
% qmod arguments
```

実行中かどうか、あるいはスプールされているかどうかに関係なく、Sun Grid Engine, Enterprise Edition ジョブを取り消すには、qdel コマンドを使用します。qmod コマンドは、すでに実行中のジョブを一時停止または停止解除 (再開) します。

両方のコマンドとも、実行するには、正常に実行された qsub コマンドから返されるジョブ ID 番号を知っている必要があります。番号を思い出せない場合は、qstat を使用して確認することができます (131 ページの「qstat を使用してジョブを監視する」の節を参照)。

以下は、両方のコマンドの入力例です。

```
% qdel job_id
% qdel -f job_id1, job_id2
% qmod -s job_id
% qmod -us -f job_id1, job_id2
% qmod -s job_id.task_id_range
```

ジョブを削除、一時停止、停止解除するには、そのジョブの所有者であるか、Sun Grid Engine, Enterprise Edition のマネージャーまたはオペレータである必要があります (70 ページの「マネージャーとオペレータ、所有者」を参照)。

両方のコマンドとも、-f (強制) オプションを使用して、ネットワーク上の問題などで sge_execd にアクセスできない場合に、sge_execd に連絡することなく、sge_qmaster にジョブの状態変更を登録することができます。これは、管理者用として用意されているオプションです。ただし、クラスタ構成の qmaster_params エントリが設定されている場合は、ユーザーが qdel を使用して自分のジョブを強制的に削除することができます (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の sge_conf のマニュアルページを参照)。

ジョブの依存関係

しばしば、複雑なタスクを構築する最も簡単な方法は、そのタスクをサブタスクに分割することです。そうした場合、サブタスクの開始は、他のサブタスクが正常に完了したかどうかに依存します。たとえば先行タスクが出力ファイルを生成し、後続タスクがそのファイルを読み取り、処理する必要がある場合などです。

Sun Grid Engine, Enterprise Edition では、そのジョブ依存関係機能を使用して相互に依存するタスクに対応しています。1 つまたは複数の他のタスクの正常終了に依存するようにジョブを構成することができます。この機能は、qsub -hold_jid オプションによって実現されます。このオプションを使用して、実行依頼するジョブが依

存するジョブのリストを指定することができます。このジョブのリストには、配列ジョブのサブセットを含むこともできます。依存関係リストのすべてのジョブが正常終了しない限り、実行依頼するジョブが実行対象になることはありません。

キューの制御

58 ページの「キューとキュープロパティ」の節で説明しているように、キューの所有者は自分のキューを一時停止/停止解除、あるいは使用可能/使用不可にすることができます。これは、そうしたユーザーが大切な仕事に特定のマシンを使用する必要があり、バックグラウンドで動作している Sun Grid Engine, Enterprise Edition ジョブの影響をかなり受ける場合に役立ちます。

キューを一時停止または使用可能にする方法は 2 通りあります。

- 「QMON キュー制御」ダイアログボックスを使用する
- qmod コマンドを使用する

▼ QMON からキューを制御する

- QMON のメインメニューで「キュー制御」ボタンをクリックします。

  図 5-11 に示すような「キュー制御」ダイアログボックスが表示されます。

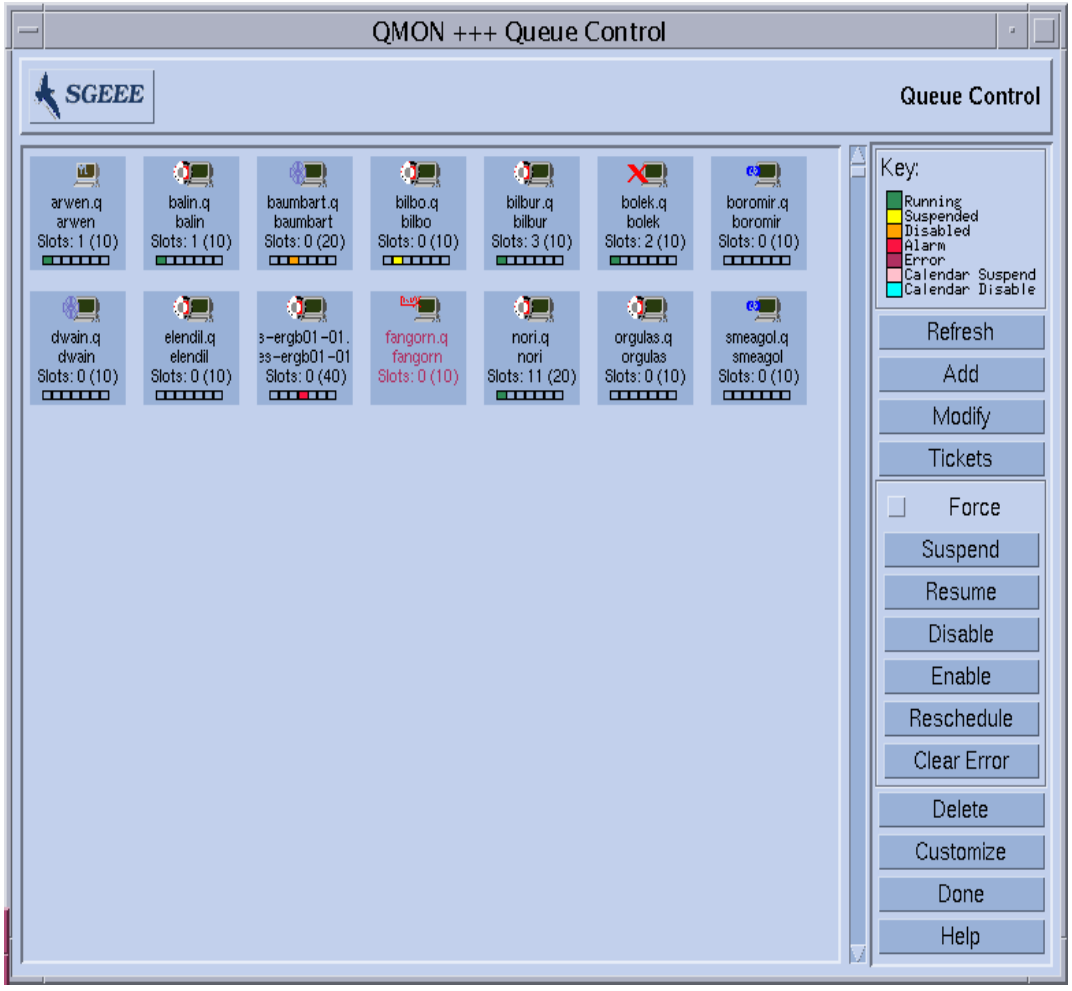


図 5-11　「キュー制御」ダイアログボックス

「キュー制御」ダイアログボックスの目的は、クラスタ内の使用可能な資源と活動の概要を素早く確認できるようにすることにあります。また、このダイアログボックスには、キューの一時停止/停止解除や使用不可/使用可能ばかりでなく、キューの構成を行う手段も用意されています。表示される各アイコンはキューを表します。主表示区画が空の場合は、構成されているキューがないことを意味します。各キューのアイコンには、そのラベルとして、キュー名とそのキューが存在するホスト名、占有さ

れているジョブスロット数が表示されます。キューのホストで sge_execd が動作中で、sge_qmaster に登録されている場合は、そのキューのアイコンに絵柄でキューのホストのオペレーティングシステムアーキテクチャ、最下部のカラーバーでキューの状態が示されます。「キュー制御」ダイアログボックスの右側の説明は、色の意味を示します。

そうしたキューについては、現在の属性や負荷、資源消費情報を確認することができます。また、キーボードの Shift キーを押しながらマウスの左ボタンでキューのアイコンをクリックすることによってキューのホストになっているマシンについても、同等の情報を得ることができます。その場合は、図 5-12 に示すような画面がポップアップ表示されます。

キューは、マウスの左ボタンでキューのアイコンボタンをクリックするか、長方形でボタンを囲むことによって選択します。「削除」「一時停止 / 停止解除」「使用不可 / 使用可能」ボタンを使用すると、選択したキューに対してそれぞれの対応する操作を行うことができます。一時停止/停止解除および使用不可/使用可能操作は、対応する sge_execd にそのことを通知する必要があります。ホストが停止しているなどの理由でこの通知を行えない場合は、「強制」トグルボタンをオンにすることによって、sge_qmaster の内部ステータスを強制的に変更することができます。

一時停止された場合、キューは閉じて、それ以上ジョブを受け付けなくなり、121 ページの「QMON からジョブを監視、制御する」の節で説明しているように、そのキューで実行中のジョブも一時停止されます。キューおよびそのジョブは、停止解除されるとただちに再開されます。

注 – 一時停止されたキュー内のジョブがさらに明示的に一時停止されている場合は、キューが停止解除しても、そのジョブは再開されません。再開するには、そのジョブを明示的に停止解除する必要があります。

これに対し、キューを使用不可にすると、キューは閉じられますが、そのキューのジョブはそのまま実行を継続することができます。一般にキューの使用不可操作は、キューを「空にする」目的で使用します。使用可能にすると、キューは再びジョブの実行が可能な状態になります。このとき、実行中のジョブには何の処理も行われません。

一時停止/停止解除、あるいは使用不可/使用可能操作を行うには、キューの所有者か、Sun Grid Engine, Enterprise Edition のマネージャーまたはオペレータ権限が必要です (70 ページの「マネージャーとオペレータ、所有者」を参照)。

「キュー制御」ダイアログボックスの情報は定期的に更新されます。「再表示」ボタンをクリックすることによって強制的に更新することもできます。「完了」ボタンは、ダイアログボックスを閉じます。

「カスタマイズ」ボタンを使用すると、フィルタ機能を使用して表示するキューを選択することができます。図 5-13 は、osf4 アーキテクチャ (バージョン 4 の Compaq UNIX) に属するホスト上にあるキューだけを選択しています。「カスタマイズ」ダ

イアログボックスの「保存」ボタンを使用すると、自分のホームディレクトリの .qmon_preferences ファイルに設定を保存し、以降 QMON を起動したときの標準として使用することができます。

「キュー制御」画面の右側の「追加」または「変更」ボタンをクリックすると、キュー構成用のサブダイアログボックスが開きます (詳細は、170 ページの「QMON からキューを構成する」を参照)。

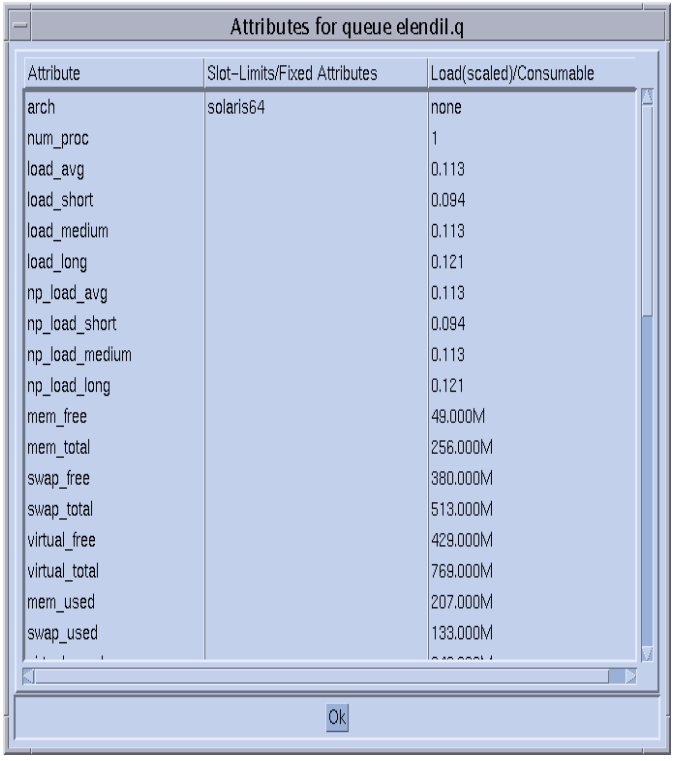


図 5-12　キュー属性の表示

ホストまたはクラスタから継承したものも含めて、キューに割り当てられている属性はすべて、「属性」欄に表示されます。「スロット制限/固定属性」欄は、キュー別のスロット制限または固定複合属性として定義されている属性の値を示します。「負荷 (調整済み)/消費可能」欄は、報告された負荷パラメータ (調整済みに設定されている場合、214 ページの「負荷パラメータ」の節を参照) と、Sun Grid Engine, Enterprise Edition 消費可能資源機能に基づいて使用可能な資源の能力に関する情報を提供します (201 ページの「消費可能資源」の節を参照)。

注 – 負荷属性が消費可能資源として設定されている場合、負荷レポートと消費可能な資源能力は互いに書き換えられることがあります。ジョブのディスパッチアルゴリズムで使用されているいずれか小さい方の値が表示されます。

注 – 現在のところ、28 ページの「実行ホスト」の節で説明しているような負荷調整は、表示される負荷および消費可能値に反映されません。

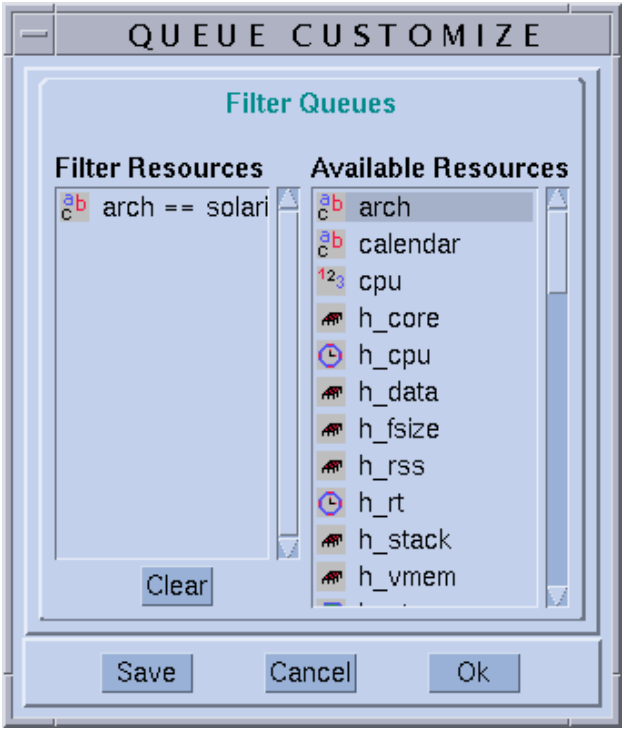


図 5-13 キュー制御のカスタマイズ

▼ qmod を使用してキューを制御する

134 ページの「コマンド行からジョブを制御する」の節では、Sun Grid Engine, Enterprise Edition の `qmod` コマンドを使用してジョブを一時停止/停止解除する方法を説明しました。この `qmod` コマンドには、キューを一時停止/停止解除、あるいは使用不可/使用可能にする機能もあります。

- この後の説明に従い、コマンド行から適切な引数を付けて次のコマンドを入力します。

```
% qmod arguments
```

以下は、キューに対する qmod の使用例です。

```
% qmod -s q_name
% qmod -us -f  q_name1, q_name2
% qmod -d q_name
% qmod -e  q_name1, q_name2, q_name3
```

最初の 2 つのコマンドはそれぞれキューを一時停止、停止解除し、3 つ目と 4 つ目のコマンドはそれぞれキューを使用不可、使用可能にします。2 つ目のコマンドでは、さらに -f オプションを使用して、ネットワーク上の問題などのために対応する sge_execd にアクセスできない場合に、sge_qmaster に強制的にステータス変更の登録を行うようにしています。

注 – 一時停止/停止解除、あるいは使用不可/使用可能操作を行うには、キューの所有者か、Sun Grid Engine, Enterprise Edition のマネージャーまたはオペレータ権限が必要です (70 ページの「マネージャーとオペレータ、所有者」を参照)。

注 – crontab または at ジョブで qmod コマンドを使用することができます。

QMON のカスタマイズ

QMON のルック&フィールは、大体は専用のリソースファイルで定義されています。QMON にはデフォルト値として標準的な値がすでに設定されていますが、*<sge_root>*/qmon/Qmon にサンプルのリソースファイルを参考にカスタマイズすることもできます。

クラスタの管理者は、QMON 専用のリソース定義を標準の .Xdefaults または .Xresources に取り込むか、XAPPLRESDIR などの標準の検索パスで参照される場所にサイト専用の Qmon ファイルを置くことによって、/usr/lib/X11/app-defaults/Qmon などの標準の場所にサイト専用のデフォルト値をインストールすることができます。上記のどのケースが自分に当てはまるかについては、管理者にお尋ねください。

また、ユーザーは自分のホームディレクトリ (または個人用の XAPPLRESDIR 検索パスが指し示す別の場所) に Qmon ファイルをコピーして変更するか、専用の .Xdefaults または .Xresources ファイルに必要なリソース定義を含めることによって、自分の好みに合った設定を行うことができます。個人用の Qmon リソースファイルは、X11 環境の運用中または起動時に .xinitrc ファイルなどでxrdb コマンドを使用して組み込むこともできます。

可能なカスタマイズについての詳細は、サンプルの Qmon ファイルのコメント行を参照してください。

図 5-2 および図 5-13 に示すジョブ制御およびキュー制御の「カスタマイズ」ダイアログボックスに、QMON をカスタマイズするもう 1 つの方法の説明があります。このどちらのダイアログボックスでも、「保存」ボタンを使用して、ユーザー個人のホームディレクトリの .qmon_preferences ファイルにフィルタおよび表示の定義を保存することができます。再起動すると、QMON はこのファイルを読み取り、定義された動作を有効にします。

PART IV 管理

PART IV は、管理者向けの次の 6 つの章で構成されています。

- 第 6 章 - 145 ページの「ホストおよびクラスタ構成」

  Product Name ホストおよびクラスの構成に関する予備知識的な情報を提供し、その構成方法を詳しく説明します。

- 第 7 章 - 169 ページの「キュー構成とキューカレンダの構成」

  この章では、さまざまな種類の Product Name ジョブの「コンテナ」として働くキューの重要な概念について説明します。また、キューの構成方法についても詳しく説明します。

- 第 8 章 - 191 ページの「複合の概念」

  この章では、ジョブでユーザーが要求可能な資源属性に関するあらゆる関連情報を定義するにあたり、Product Name システムでどのように複合が用いられているのかについて説明します。管理者は、環境の条件に応じてさまざま複合を構成します。この章では、その方法についても詳しく説明します。

- 第 9 章 - 219 ページの「ユーザーアクセスとポリシーの管理」

  この章では、Product Name システムで使用可能なユーザーポリシーに関する予備的な情報を提供し、実際のコンピューティング環境に応じてそれらのポリシーを構成する方法を説明します。

- 第 10 章 - 289 ページの「並列環境の管理」

  この章では、Product Name システムの、並列環境への適応と、並列環境に対応するための構成方法を詳しく説明します。

- 第 11 章 - 303 ページの「エラーの通知と障害追跡」

  この章では、Product Name でエラーメッセージを調査する方法とデバッグモードでシステムを実行する方法を説明します。

第6章

ホストおよびクラスタ構成

この章では、Sun Grid Engine, Enterprise Edition 5.3 システムのさまざまな部分の構成に関する予備知識的な情報とその方法を説明します。具体的には、この章で以下の作業を行う方法を説明します。

マスターおよびシャドウマスターの構成

シャドウマスターホスト名ファイルの *<sge_root>*/*<cell>*/common/shadow_masters には、主マスターホスト (Sun Grid Engine, Enterprise Edition マスターデーモンの sge_qmaster が当初動作しているマシン) の名前とシャドウマスターホストの名前が含まれています。このマスターホスト名ファイルの形式は以下のとおりです。

- ファイルの先頭行には、主マスターホストを指定します。
- 2 行目以降には、1 行に 1 つシャドウマスターホストを指定します。

シャドウマスターホストが現れる順番は重要です。主マスターホスト (ファイルの先頭行に指定されたホスト) で問題が発生すると、2 行目に指定されたシャドウマスターがマスターホストの役割を引き継ぎます。このシャドウマスターでも問題が発生すると、3 行目のシャドウマスターがマスターホストの役割を引き継ぎます。

ホストを Sun Grid Engine, Enterprise Edition のシャドウマスターにするには、次の条件が満たされる必要があります。

- sge_shadowd を実行していること。
- ディスクに記録された sge_qmaster のステータス情報とジョブ、キュー構成を共有していること。具体的には、シャドウマスターホストには、sge_qmaster のスプールディレクトリと *<sge_root>*/*<cell>*/common ディレクトリに対する読み取り・書き込み root アクセス権が必要です。
- シャドウマスターホスト名ファイルに、シャドウマスターホストとして定義されていること。

これらの条件が満たされると、そのホストに対してシャドウマスターホスト機能が有効になります。この機能を有効にするために、Sun Grid Engine, Enterprise Edition デーモンを再起動する必要はありません。

シャドウマスターホスト上で sge_qmaster が自動的にフェイルオーバーを開始するまでには、多少時間 (1 分ほど) がかかります。その間、Sun Grid Engine, Enterprise Edition コマンドを実行しようとすると、必ずエラーメッセージが返されます。

注 – *<sge_root>*/*<cell>*/common/act_qmaster ファイルには、実際に sge_qmaster デーモンを実行するホスト名が含まれます。

シャドウの sge_qmaster を起動できるようにするには、Sun Grid Engine, Enterprise Edition は、古いシャドウマスターを前もって終了させるか、起動されたシャドウの sge_qmaster に干渉する処理を行うことなく、古い sge_qmaster が終了するようにする必要があります。非常にまれにですが、このようにすることが不可能なことがあります。そのような場合は、すべてのシャドウマスターホストの sge_shadowd のメッセージログファイルにエラーメッセージが記録され (第 11 章、303 ページの「エラーの通知と障害追跡」を参照)、sge_qmaster デーモンとの tcp

接続を開こうとするあらゆる試みは失敗します。その場合は、マスターデーモンが動作していないことを確認し、手動で任意のシャドウマスターマシンの sge_qmaster を起動してください (161 ページの「コマンド行からデーモンを終了する」の節を参照)。

デーモンとホスト

Sun Grid Engine, Enterprise Edition ホストは、そのシステムで動作しているデーモンと sge_qmaster へのホストの登録方法に従って 4 つのグループに分類されます。

- **マスターホスト** - マスターホストは、クラスタ活動全体の中心です。マスターホストはマスターデーモンの sge_qmaster を実行します。sge_qmaster は、キューやジョブなどの Sun Grid Engine, Enterprise Edition コンポーネントのすべてを制御し、コンポーネントの状態やユーザーのアクセス権限などの表を管理します。マスターホストの初期設定方法については、33 ページの「マスターホストをインストールする」の節、動的なマスターホストの変更については、146 ページの「マスターおよびシャドウマスターの構成」の節でそれぞれ説明しています。通常、マスターホストは、Sun Grid Engine, Enterprise Edition スケジューラの sge_schedd も実行します。マスターホストには、インストールで行う以外の構成作業は必要ありません。
- **実行ホスト** - 実行ホストは、Sun Grid Engine, Enterprise Edition ジョブを実行する権限を持つノードです。このため、実行ホストは Sun Grid Engine, Enterprise Edition のキューのホストの役割を果たし、Sun Grid Engine, Enterprise Edition 実行デーモンの sge_execd を実行します。34 ページの「実行ホストをインストールする」の節で説明しているように、当初、実行ホストは実行ホストのインストールで設定します。
- **管理ホスト** - Sun Grid Engine, Enterprise Edition システムに対するあらゆる種類の管理運用業務を行う権限をマスターホスト以外のホストに付与することができます。管理ホストは、次のコマンドで設定します。

  ```
  qconf -ah hostname
  ```

  詳細は、qconf のマニュアルページを参照してください。
- **実行依頼ホスト** - 実行依頼ホストは、バッチジョブのみの実行依頼と制御を行うためのホストです。具体的には、実行依頼ホストにログインしているユーザーは、qsub を使ってジョブの実行を依頼したり、qstate を使ってジョブの状態を制御したりすることができます。また、Sun Grid Engine, Enterprise Edition の OSF/1 Motif グラフィカルユーザーインタフェースの QMON を実行することもできます。実行依頼ホストは、次のコマンドで設定します。

  ```
  qconf -as hostname
  ```

  詳細は、qconf のマニュアルページを参照してください。

注 – 1 つのホストが、上記の複数のクラスに属することができます。デフォルトでは、マスターホストは管理ホストでもあり、実行依頼ホストでもあります。

ホストの構成

Sun Grid Engine, Enterprise Edition は、マスターホスト以外のあらゆる種類のホストに関するオブジェクトリストを管理します。管理および実行依頼ホストの場合、それらのリストは単に、管理または実行依頼権限がホストにあるかどうかに関する情報を提供するだけです。実行ホストオブジェクトの場合は、さらに、そのホストで動作する `sge_execd` が報告する負荷情報や、Sun Grid Engine, Enterprise Edition 管理者が提供する負荷パラメータのスケーリング率などのパラメータがリストに記録されます。

以下の節では、Sun Grid Engine, Enterprise Edition グラフィカルユーザーインタフェースの `QMON` とコマンド行を使用して、さまざまなホストオブジェクトを構成する方法を説明します。

GUI の管理機能は、一群のホスト構成ダイアログボックスによって提供され、それらのダイアログボックスは、`QMON` メインメニューの「ホスト構成」アイコンボタンをクリックすることによって開きます。具体的には、用意されているダイアログボックスは 「管理ホスト構成」(図 6-1 を参照)、「実行依頼ホスト構成」(図 6-2 を参照)、「実行ホスト構成」(図 6-3 を参照) です。これらのダイアログボックスは、画面上部の選択リストボタンを使用して切り替えることができます。

`qconf` コマンドは、ホストオブジェクト管理用のコマンド行インタフェースを提供します。

不正なホスト名

以下は、不正または予約されているために使用できないホスト名です。

- `global`
- `template`
- `all`
- `default`
- `unknown`
- `none`

▼ QMON から管理ホストを構成する

1. QMON メインメニュー上部の「管理ホスト」タブをクリックします。

   次の図に示すような「管理ホスト構成」ダイアログボックスが開きます。

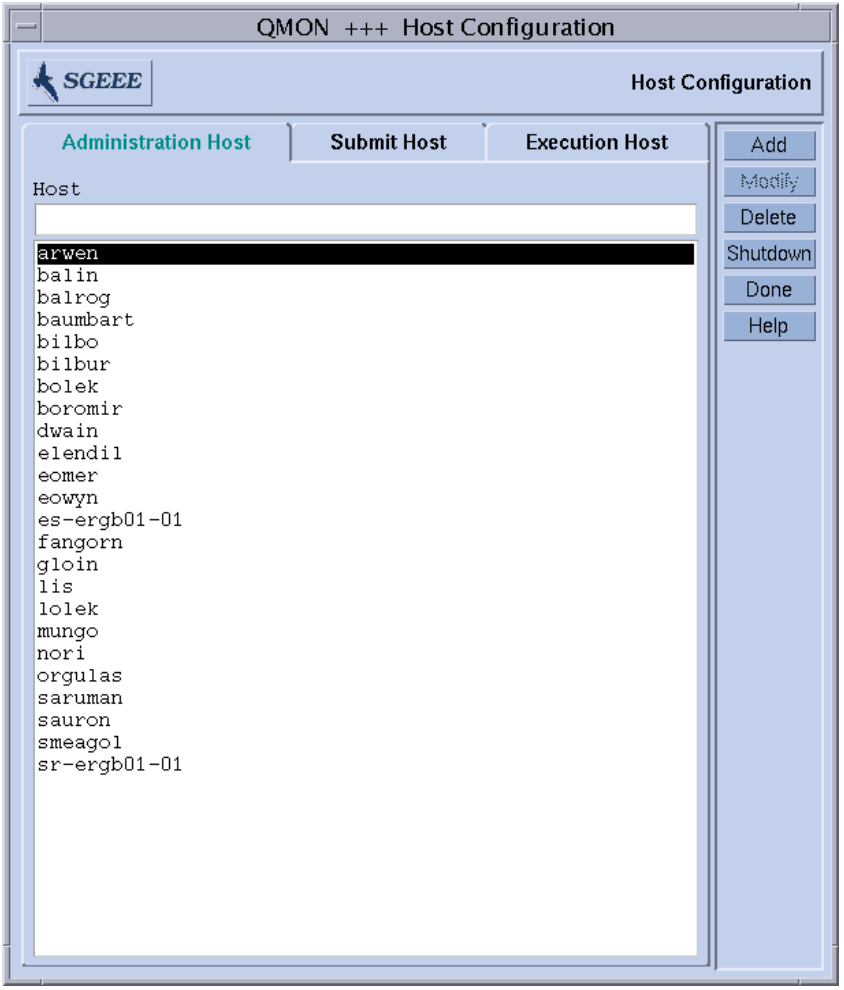


図 6-1　「管理ホスト構成」ダイアログボックス

注 – デフォルトでは、「ホスト構成」ボタンを初めてクリックすると、「管理ホスト構成」ダイアログボックスが開きます。

---

2. 以下の説明に従い、ホストの構成作業を行います。

「管理ホスト構成」ダイアログボックスでは、Sun Grid Engine, Enterprise Edition 管理コマンドの使用を許可するホストを構成することができます。画面の下半分は選択リストで、すでに管理権限を持つことが定義されているホストが表示されます。

▼ 管理ホストを削除する

- 選択リストから既存のホストを削除するには、マウスの左ボタンでその名前をクリックし、ダイアログボックス下部にある「削除」ボタンをクリックします。

▼ 管理ホストを追加する

- ホストを新規追加するには、「ホスト名」入力フィールドにその名前を入力し、「追加」ボタンをクリックするか、Return キーを押します。

▼ コマンド行から管理ホストを構成する

- 目的のホスト構成作業に応じて適切な引数を付けて次のコマンドを入力します。

```
% qconf arguments
```

qconf コマンドの引数とその働きは以下のとおりです。

- qconf -ah *hostname*

  管理ホストの追加 - 指定されたホストを管理ホストリストに追加します。

- qconf -dh *hostname*

  管理ホストの削除 - 管理ホストリストから指定されたホストを削除します。

- qconf -sh

  管理ホストの表示 - 構成されているすべての管理ホストのリストを表示します。

▼ QMON から実行依頼ホストを構成する

1. QMON メインメニュー上部の「実行依頼ホスト」タブをクリックします。

   次の図に示すような「実行依頼ホスト構成」ダイアログボックスが開きます。

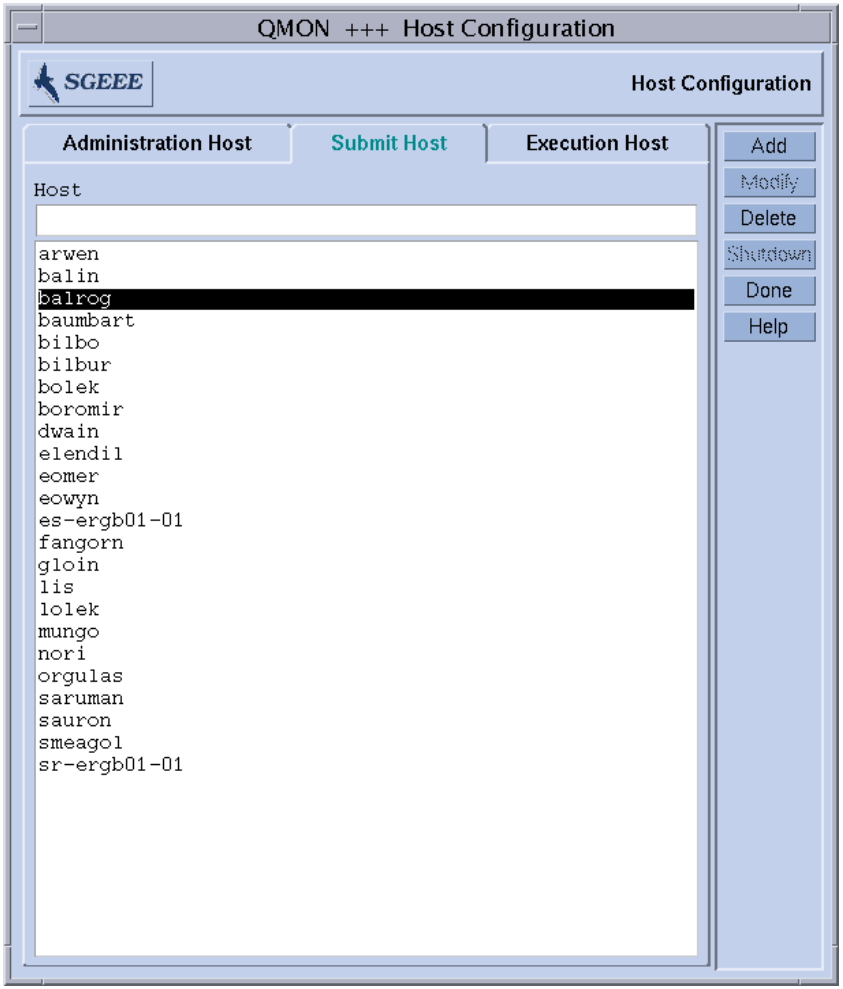


図 6-2　　実行依頼ホスト構成

2. 以下の説明に従い、ホストの構成作業を行います。

   「実行依頼ホスト構成」ダイアログボックスでは、ジョブの実行依頼、監視、制御が可能なホストを定義することができます。管理ホストとしても定義しない限り、実行依頼ホストから Sun Grid Engine, Enterprise Edition の管理コマンドを使用すること

はできません (149 ページの「QMON から管理ホストを構成する」を参照)。画面の中央は選択リストで、すでに実行依頼権限を持つことが定義されているホストが表示されます。

▼ 実行依頼ホストを削除する

- 選択リストから既存のホストを削除するには、マウスの左ボタンでその名前をクリックし、ダイアログボックス下部にある「削除」ボタンをクリックします。

▼ 実行依頼ホストを追加する

- ホストを新規追加するには、「ホスト名」入力フィールドにその名前を入力し、「追加」ボタンをクリックするか、Return キーを押します。

▼ コマンド行から実行依頼ホストを構成する

- 目的のホスト構成作業に応じて適切な引数を付けて次のコマンドを入力します。

```
% qconf arguments
```

qconf コマンドの引数とその働きは以下のとおりです。

- qconf -as *hostname*

  実行依頼ホストの追加 - 指定されたホストを実行依頼ホストリストに追加します。

- qconf -ds *hostname*

  実行依頼ホストの削除 - 実行依頼ホストリストから指定されたホストを削除します。

- qconf -ss

  実行依頼ホスト - 実行依頼権限を持つことが定義されているすべてのホスト名のリストを表示します。

▼ QMON から実行ホストを構成する

1. QMON メインメニュー上部の「実行ホスト」タブをクリックします。

   次の図に示すような「実行ホスト構成」ダイアログボックスが開きます。

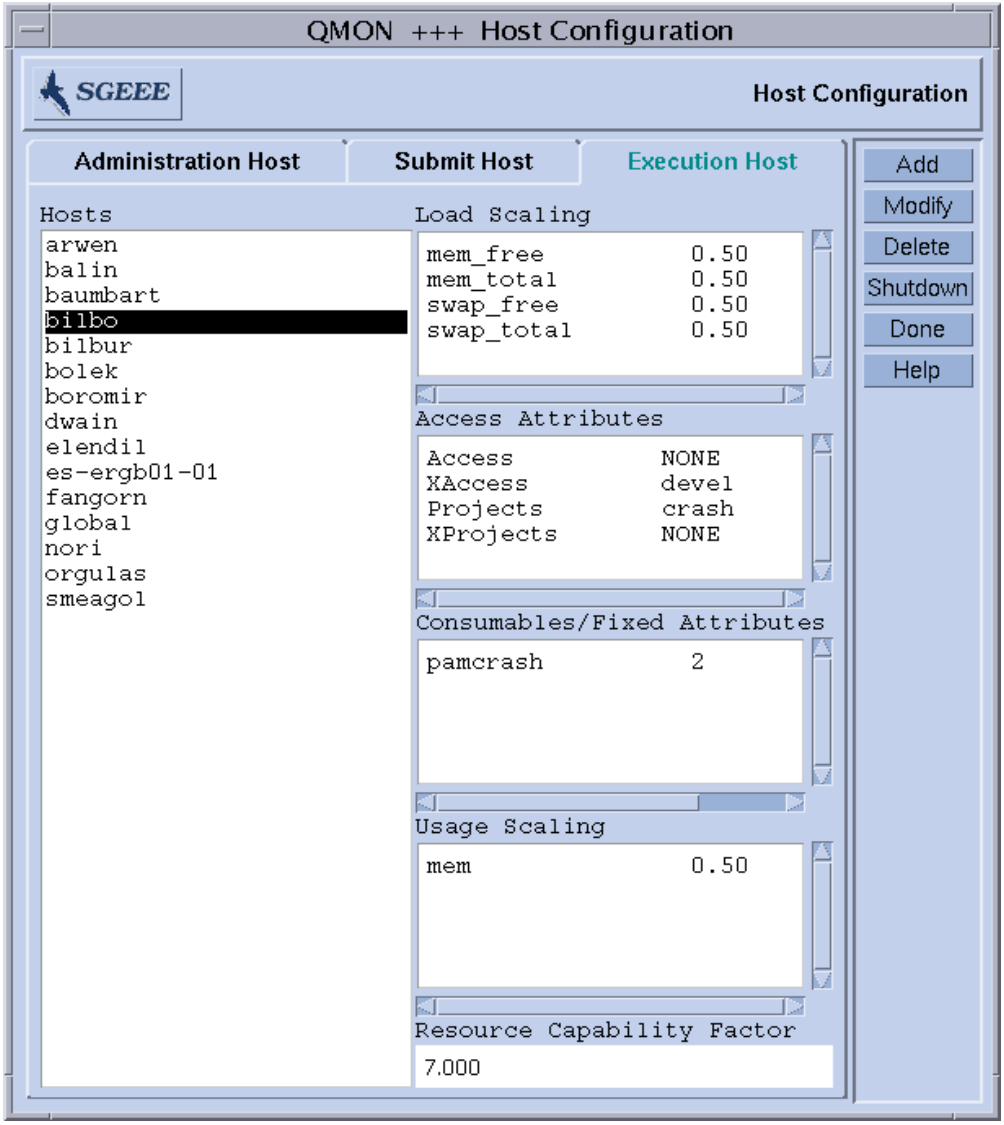


図 6-3　　実行ホスト構成

2. **以下の説明に従い、ホストの構成作業を行います。**

   Sun Grid Engine, Enterprise Edition の実行ホストは、「実行ホスト構成」ダイアログボックスから構成することができます。管理または実行ホストとしても定義しない限り、実行ホストから管理コマンドや実行依頼コマンドを使用することはできません (149 ページの「QMON から管理ホストを構成する」と 151 ページの「QMON から実行依頼ホストを構成する」を参照)。

   「ホスト」選択リストには、すでに定義されている実行ホストが表示されます。実行ホストを選択すると、そのホストに現在設定されている負荷スケーリング係数、アクセス権限、関連付けられている消費可能および固定複合属性の資源可用性が、それぞれ「負荷スケーリング」、「アクセス属性」、「消費可能 / 固定属性」欄に表示されます。複合属性、ユーザーのアクセス権、負荷パラメータについての詳細は、それぞれ 191 ページの「複合」、68 ページの「ユーザーのアクセス権」、214 ページの「負荷パラメータ」を参照してください。

   Sun Grid Engine, Enterprise Edition の場合は、この他にも「資源利用スケーリング」欄があり、さまざまなマシンの CPU、メモリー、入出力の利用メトリック (計測データ) に対する現在のスケーリング係数が表示されます。資源利用状況は、現在実行中のジョブごとに `sge_execd` よって定期的に報告されます。スケーリング係数は、ジョブを実行しているユーザーまたはプロジェクトの特定のマシン上での相対的な資源利用コストを示します。この係数は、たとえば 400MHz のプロセッサと 600MHz CPU との間の CPU 時間 1 秒のコストの比較などに使用することができます。「資源利用スケーリング」欄に表示されないメトリックのスケーリング係数は 1 です。

   「資源能力係数」フィールドも Sun Grid Engine, Enterprise Edition にしかないフィールドで、ジョブの振り分け時にスケジューラによって使用されます。この係数は、ホストの総合的で相対パワーを示す数字です。資源能力係数に関係する要素としては、CPU の個数や、CPU のクロック速度、CPU のタイプ、使用可能なメモリー容量、接続されているデバイスの速度などがあります。

▼ 実行ホストを削除する

- **「実行ホスト」ダイアログボックスで削除する実行ホストの名前をクリックし、右側のボタン欄にある「削除」ボタンをクリックします。**

▼ 実行ホストデーモンを停止する

- **「実行ホスト」ダイアログボックスでホストを選択して、「停止」ボタンをクリックします。**

▼ 実行ホストを追加または変更する

1. 「実行ホスト」ダイアログボックスのボタン欄にある「追加」または「変更」ボタンをクリックします。

   図 6-4 に示すようなダイアログボックスが表示されます。

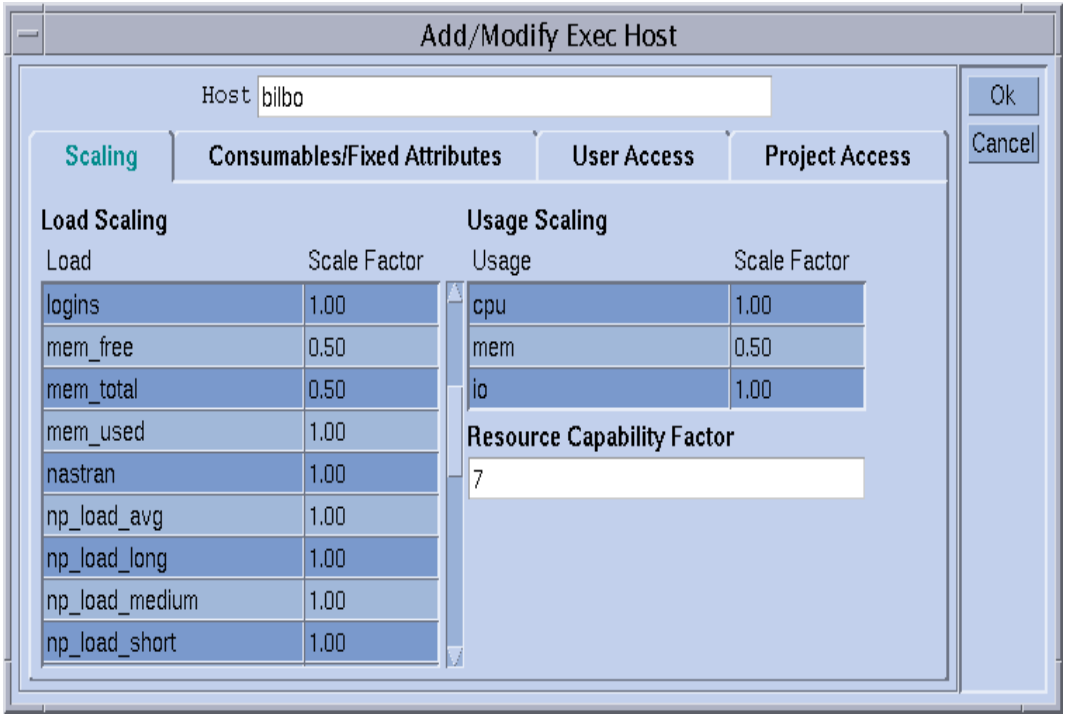


図 6-4　　負荷スケーリングの変更

2. **以下の説明に従い、ホストの変更作業を行います。**

   実行ホストを新規追加または既存の実行ホストの構成を変更するためのダイアログボックスでは、ホストに関係するすべての属性を変更することができます。「ホスト」入力フィールドには、実行ホスト名が表示されるか、ホスト名を追加することができます。ダイアログボックスの「スケーリング」タブを選択することによって、スケーリング係数を定義することができます (図 6-4 を参照)。

   「負荷スケーリング」表の「負荷」列には、使用可能なすべての負荷パラメータ、「スケール率」列には、それぞれの負荷に対応するスケーリング値がそれぞれ表示されます。「スケール率」列は編集することができます。有効なスケーリング係数は、固定小数点または科学的記数法形式の正の浮動小数点数です。

Sun Grid Engine, Enterprise Edition の場合は、この他にも、CPU、メモリー、入出力の利用メトリックに対する現在のスケーリング係数が「資源利用スケーリング」の「資源利用」列、対応するスケーリングの定義が「スケール率」列に表示されます。「スケール率」列は編集することができます。有効なスケーリング係数は、固定小数点または科学的記数法形式の正の浮動小数点数です。

また、同じく Sun Grid Engine, Enterprise Edition の場合は、「資源能力係数」入力フィールドに資源能力係数を設定することもできます。有効なスケーリング係数は、固定小数点または科学的記数法形式の正の浮動小数点数です。

タブウィジェットで「消費可能 / 固定属性」を選択すると、ホストに関連付けられている複合属性を定義することができます (図 6-5 を参照)。ホストに複合を関連付けるということは (191 ページの「複合」の節を参照)、ダイアログボックスの左下の複合選択リストを使用してホストに「グローバル複合」や「ホスト複合」、あるいは「管理者定義の複合」を関連付けることです。選択可能な管理者定義の複合は左側に表示され、赤い矢印を使用して関連付けたり、関連付けを解除したりできます。現在の複合構成の他の情報が必要な場合、または複合構成を変更する場合は、「複合構成」アイコンボタンをクリックして、最上位の「複合構成」ダイアログボックスを開きます。

ダイアログボックスの右下の「消費可能 / 固定属性」表は、現在、値が定義されているすべての複合属性のリストです。このリストは、上部の「負荷」または「値」ボタンをクリックすることによって拡張することができます。ボタンをクリックすると、ホストに関連付けられているすべての属性 (すなわち、グローバル、ホスト、管理者定義の複合に設定されているすべての属性をまとめたもの) の入った選択リストが開きます。図 6-6 は、「属性の選択」ダイアログボックスを示しています。どれか属性を選択し、「了解」ボタンをクリックして選択を確定すると、その属性が「消費可能 / 固定属性」表の「名前」列に追加され、対応する「値」フィールドにポインタが移動します。値の変更は、マウスの左ボタンで「値」フィールドをダブルクリックすることによって行うことができます。属性を削除する場合は、マウスの左ボタンで対応する行を選択して、Ctrl-D を押すか、マウスの右ボタンをクリックして削除ボックスを開き、削除を確定します。

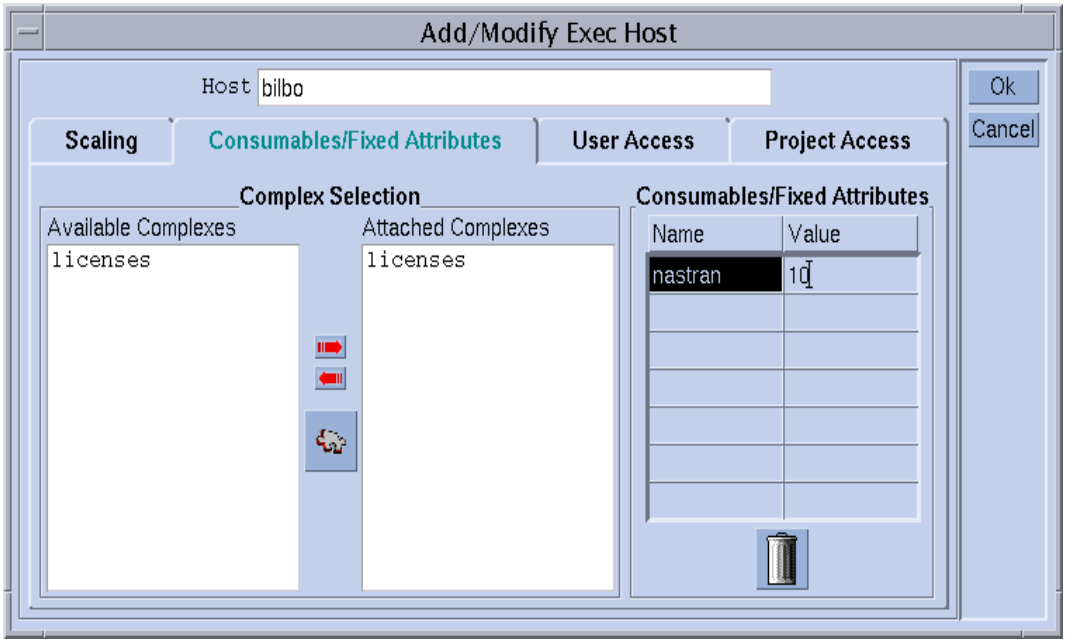


図 6-5　　消費可能 / 固定属性の変更

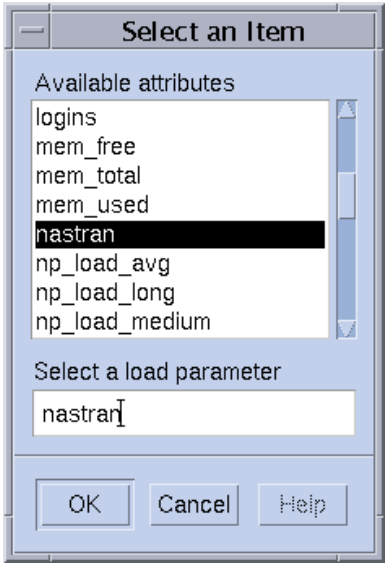


図 6-6　　使用可能な複合属性

「ユーザーアクセス」タブ (図 6-7) を選択すると、以前に構成したユーザーアクセスリストに基づいて実行ホストに対するアクセス権を定義することができます。

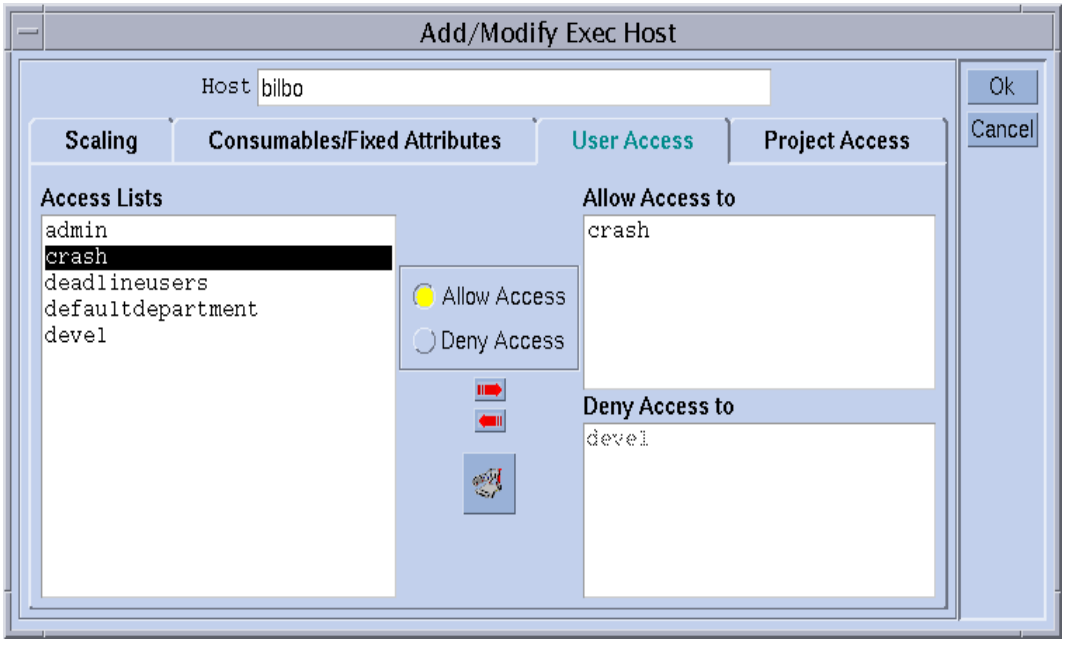


図 6-7　　ユーザーアクセス権の変更

「プロジェクトアクセス」タブ (図 6-8) を選択すると、以前に構成したプロジェクトに基づいて実行ホストに対するアクセス権を定義することができます。

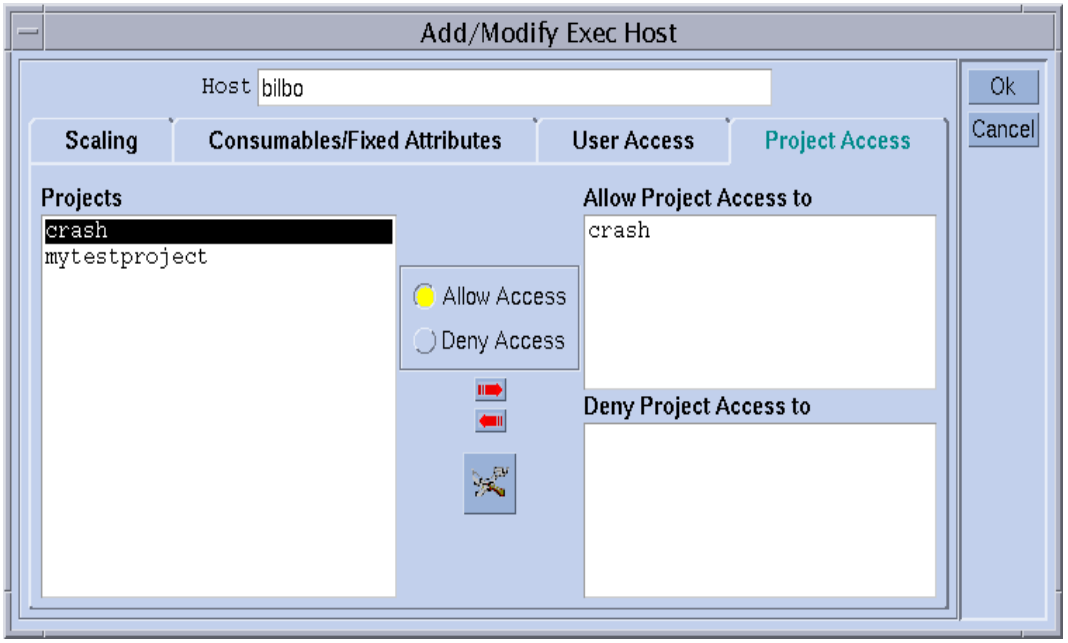


図 6-8　　プロジェクトアクセス権の変更

▼ コマンド行から実行ホストを構成する

- 目的ホストの構成作業に応じて適切な引数を付けて次のコマンドを入力します。

```
% qconf arguments
```

実行ホストのリストを管理するためのコマンド行インタフェースは、qconf コマンドの次のオプションでアクセスできます。

- qconf -ae *[exec_host_template]*

実行ホストの追加 - このコマンドは、エディタ (デフォルトの vi か、$EDITOR 環境変数に指定されたエディタ) を使用して、実行ホスト構成用のテンプレートを開きます。省略可能なパラメータの *exec_host_template* (すでに構成済みの実行ホスト名) を指定すると、その実行ホストの構成がテンプレートとして使用されます。テンプレートの内容を変更し、ディスクに保存することによって、実行ホストを構成してください。変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の host_conf の項を参照してください。

- `qconf -de` *hostname*

実行ホストの削除 - 実行ホストリストから指定されたホストを削除します。実行ホストの構成のすべてのエントリが失われます。

- `qconf -me` *hostname*

実行ホストの変更 - このコマンドは、エディタ (デフォルトの `vi` か、`$EDITOR` 環境変数に指定されたエディタ) を使用して、指定された実行ホストの構成をテンプレートとして開きます。テンプレートの内容を変更し、ディスクに保存することによって、実行ホストの構成を変更してください。変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `host_conf` のマニュアルページを参照してください。

- `qconf -Me` *filename*

実行ホストの変更 - 指定されたファイルを実行ホスト構成用のテンプレートして使用します。指定されたファイルの構成は既存の実行ホストを参照している必要があります。この実行ホストの構成が、指定されたファイルの内容で置き換えられます。この `qconf` オプションは、手動の介入を必要としないため、`cron` ジョブなど、オフラインで実行ホストの構成を変更する場合に便利です。

- `qconf -se` *hostname*

実行ホストの表示 - `host_conf` に定義されている指定されたされた実行ホストの構成を表示します。

- `qconf -sel`

実行ホストのリストの表示 - 実行ホストとして設定されているホスト名のリストを表示します。

▼ qhost を使用して実行ホストを監視する

`qhost` コマンドは、実行ホストのステータスの概要を素早く確認するための便利な手段です。

- **次のコマンドを入力します。**

```
% qhost
```

以下のような出力が生成されます。

表 6-1　qhost の出力例

```
HOSTNAME            ARCH      NPROC  LOAD  MEMTOT  MEMUSE   SWAPTO   SWAPUS
-------------------------------------------------------------------------
global              -         -      -     -       -        -        -
BALROG.genias.de    solaris6  2      0.38  1.0G    994.0M   900.0M   891.0M
BILBUR.genias.de    solaris   1      0.18  96.0M   70.0M    164.0M   9.0M
DWAIN.genias.de     irix6     1      1.13  149.0M  55.8M    40.0M    0.0
GLOIN.genias.de     osf4      2      0.05  768.0M  701.0M   1.9G     13.5M
SPEEDY.genias.de    alinux    1      0.08  248.8M  60.6M    125.7M   232.0K
SARUMAN.genias.de   solaris   1      0.11  96.0M   77.0M    192.0M   9.0M
FANGORN.genias.de   linux     1      2.01  124.8M  49.9M    127.7M   4.3M
```

qhost のオプションについては、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qhost の項を参照してください。

▼ コマンド行からデーモンを終了する

- **以下のいずれかのコマンドを使用します。これらの操作には、Sun Grid Engine, Enterprise Edition マネージャーまたはオペレータ特権が必要であることに注意してください (第 9 章、219 ページの「ユーザーアクセスとポリシーの管理」を参照)。**

```
% qconf -kej
% qconf -ks
% qconf -km
```

  - 最初のコマンドは、現在アクティブなジョブをすべて終了し、すべての Sun Grid Engine, Enterprise Edition 実行デーモンを停止します。

注 – qconf -ke コマンドを使用すると、すべての Sun Grid Engine, Enterprise Edition 実行デーモンが停止するだけで、アクティブなジョブは取り消されません。sge_execd が動作していない間に完了したジョブは、そのシステムで sge_execd が再起動されるまで sge_qmaster に報告されません。ただし、ジョブレポートが失われることはありません。

  - 2 つ目のコマンドは、Sun Grid Engine, Enterprise Edition スケジューラの sge_schedd を停止します。
  - 3 つ目のコマンドは sge_qmaster プロセスを強制的に終了させます。

実行中のジョブがあり、現在のアクティブなジョブがすべて終了するまで Sun Grid Engine, Enterprise Edition の停止を遅らせてもよい場合は、上記の qconf シーケンスを実行する前にキューごとに下記のコマンドを使用します。

```
% qmod -d queue_name
```

queue_name は、キューの名前です。

この qmod コマンドは、使用不可にされたキューに新しいジョブがスケジューリングされないようにします。このようにして、キューで実行されているジョブがなくなってから、デーモンを終了することができます。

▼ コマンド行からデーモンを再起動する

1. Sun Grid Engine, Enterprise Edition のデーモンを再起動するマシンに root でログインします。
2. 次のスクリプトを実行します。

```
% <sge_root>/<cell>/common/rcsge
```

このスクリプトは、このホストで正常に動作しているデーモンを探し、対応するデーモンを起動します。

基本クラスタ構成

Sun Grid Engine, Enterprise Edition 基本クラスタ構成とは、mail や xterm などのプログラムに対するパスなどのサイト依存関係を反映し、Sun Grid Engine, Enterprise Edition の動作を制御する構成情報の集まりです。Sun Grid Engine, Enterprise Edition のグローバル構成が 1 つあり、その情報は、そのマスターホストばかりでなく、プール内のあらゆるホストによって供給されます。また、グローバル構成内の特定のエントリに優先する、各ホストにローカルの構成を使用するように Sun Grid Engine, Enterprise Edition システムを構成することもできます。

構成エントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の sge_conf の項を参照してください。Sun Grid Engine, Enterprise Edition クラスタの管理者は、インストールの終了後ただちにサイトのニーズに合わせてグローバルおよびローカル構成を変更し、その後は、それらの構成を最新に保つ必要があります。

▼ コマンド行から基本クラスタ構成を表示する

Sun Grid Engine, Enterprise Edition の現在の構成を表示するには、qconf コマンドで構成表示オプションを使用します。以下は、その例です (詳細は『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。

- **以下のいずれかのコマンドを使用します。**

```
% qconf -sconf
% qconf -sconf global
% qconf -sconf <host>
```

最初の 2 つのコマンドは同等で、グローバル構成を表示します。3 つ目のコマンドは、指定されたホストのローカル構成を表示します。

▼ コマンド行から基本クラスタ構成を変更する

注 – クラスタ構成を変更する Sun Grid Engine, Enterprise Edition コマンドの qconf を使用できるのは、Sun Grid Engine, Enterprise Edition の管理者だけです。

- **以下のいずれかのコマンドを使用します。**

```
% qconf -mconf global
% qconf -mconf <ホスト>
```

  - 最初のコマンドは、グローバル構成を変更します。
  - 2 つ目の21コマンドは、指定された実行またはマスターホストのローカル構成を操作します。

使用可能な qconf コマンドは数多くあります。上記のコマンドはそのうちの 2 例です。その他の qconf コマンドについては、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照してください。

▼ QMON からクラスタ構成を表示する

1. QMON のメインメニューで「クラスタ構成」ボタンをクリックします。
   図 6-9 に示すような「クラスタ構成」ダイアログボックスが表示されます。

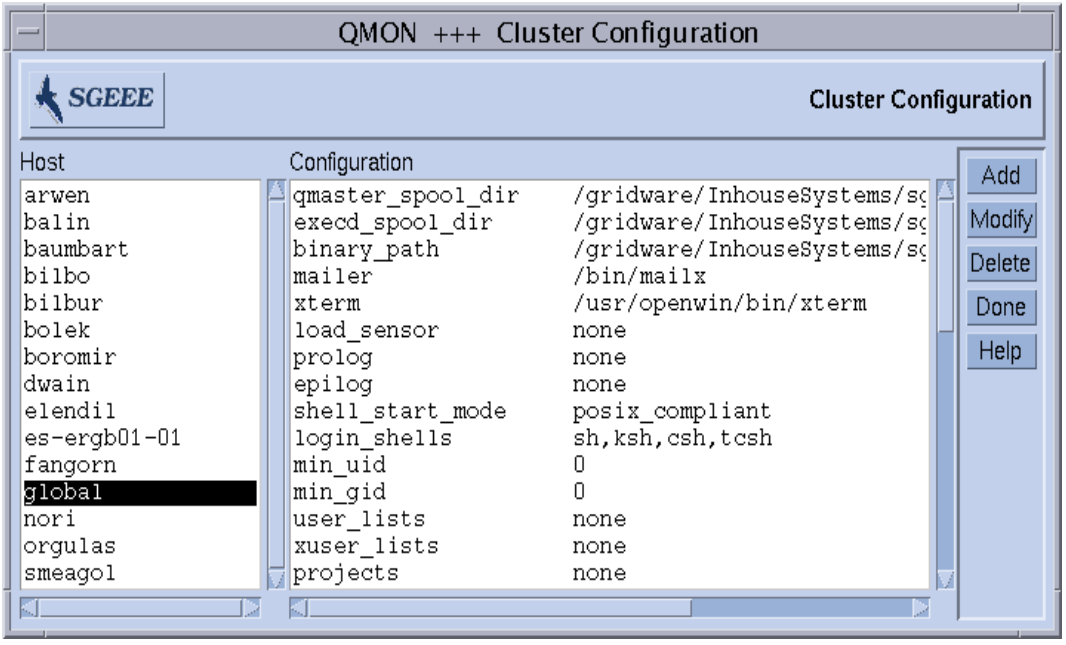


図 6-9　「クラスタ構成」ダイアログボックス

2. ダイアログボックス左側の「ホスト」選択リストでその現在の構成を表示するホスト名をクリックします。

▼ QMON からクラスタ構成を削除する

1. QMON のメインメニューで「クラスタ構成」ボタンをクリックします。
2. ダイアログボックス左側の「ホスト」選択リストで構成を削除するホスト名をクリックします。
3. 「削除」ボタンをクリックします。

▼ QMON からグローバルクラスタ構成を表示する

- **「ホスト」選択リストから名前、`global` を選択します。**

  `sge_conf` のマニュアルページに説明している形式で構成が表示されます。選択したグローバル構成またはホストにローカルの構成 (ローカル構成) を変更するには、「変更」ボタンを使用します。特定のホストに新しい構成を追加するには、「追加」ボタンを使用します。

▼ QMON からグローバルまたはホスト構成を変更する

1. **「クラスタ構成」ダイアログボックス (164 ページの「QMON からクラスタ構成を表示する」を参照) で「追加」または「変更」ボタンをクリックします。**

   図 6-10 に示すような「クラスタ設定」ダイアログボックスが表示されます。

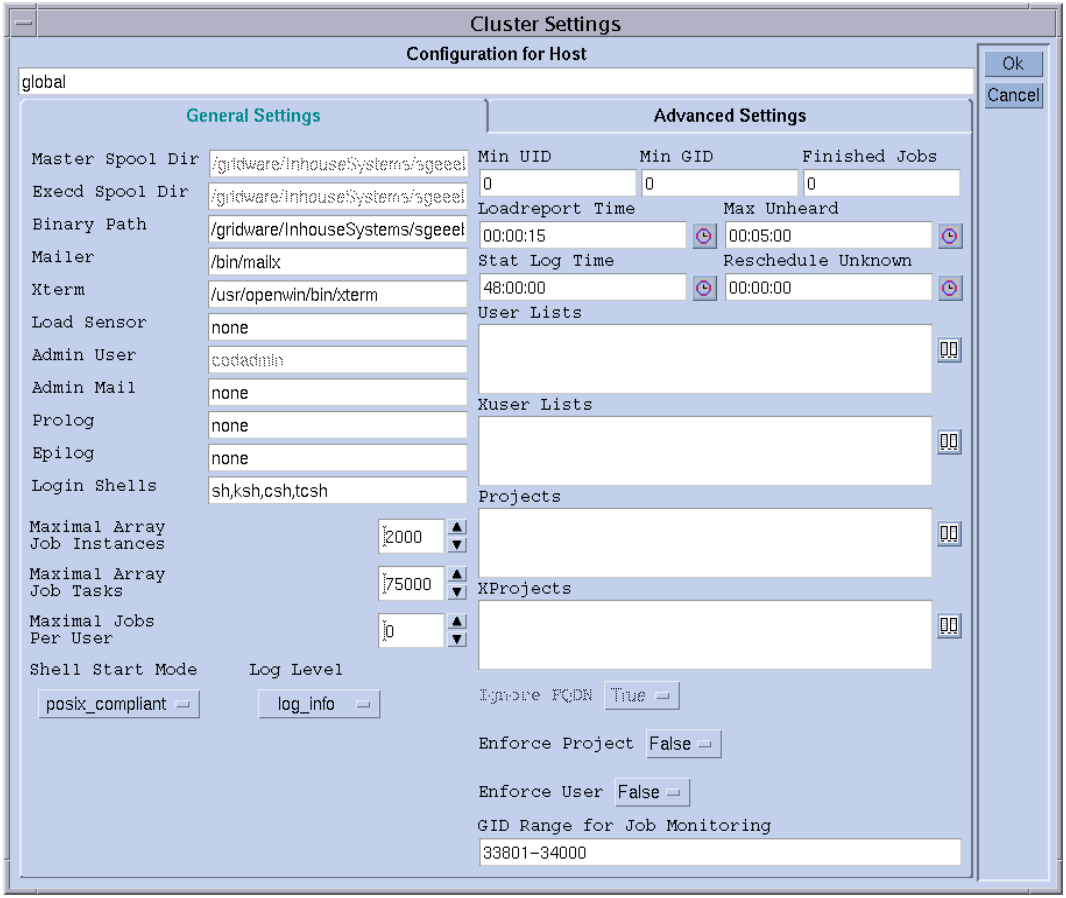


図 6-10　「クラスタ設定」ダイアログボックス - 一般設定

2. 以下の説明に従って必要な変更を行います。

「クラスタ設定」ダイアログボックスでは、グローバルまたはローカル構成のあらゆるパラメータを変更できます。すべての入力フィールドにアクセスできるるようになるのは、グローバル構成を変更する、すなわち、ホストとして global を選択して「変更」をクリックした場合だけです。通常のホストの変更の場合は、その実際の構成がダイアログボックスに反映され、ローカルの変更に当てはまるパラメータだけ変更することができます。新しいローカル構成の追加では、ダイアログボックスのフィールドは空の状態で表示されます。

「高度設定」タブの表示 (図 6-11) は、グローバルまたはローカル構成の変更、あるいは新規構成の追加のどの操作であるかによって異なります。このダイアログボックスでは、あまり使用されることのないクラスタ構成パラメータにアクセスできます。

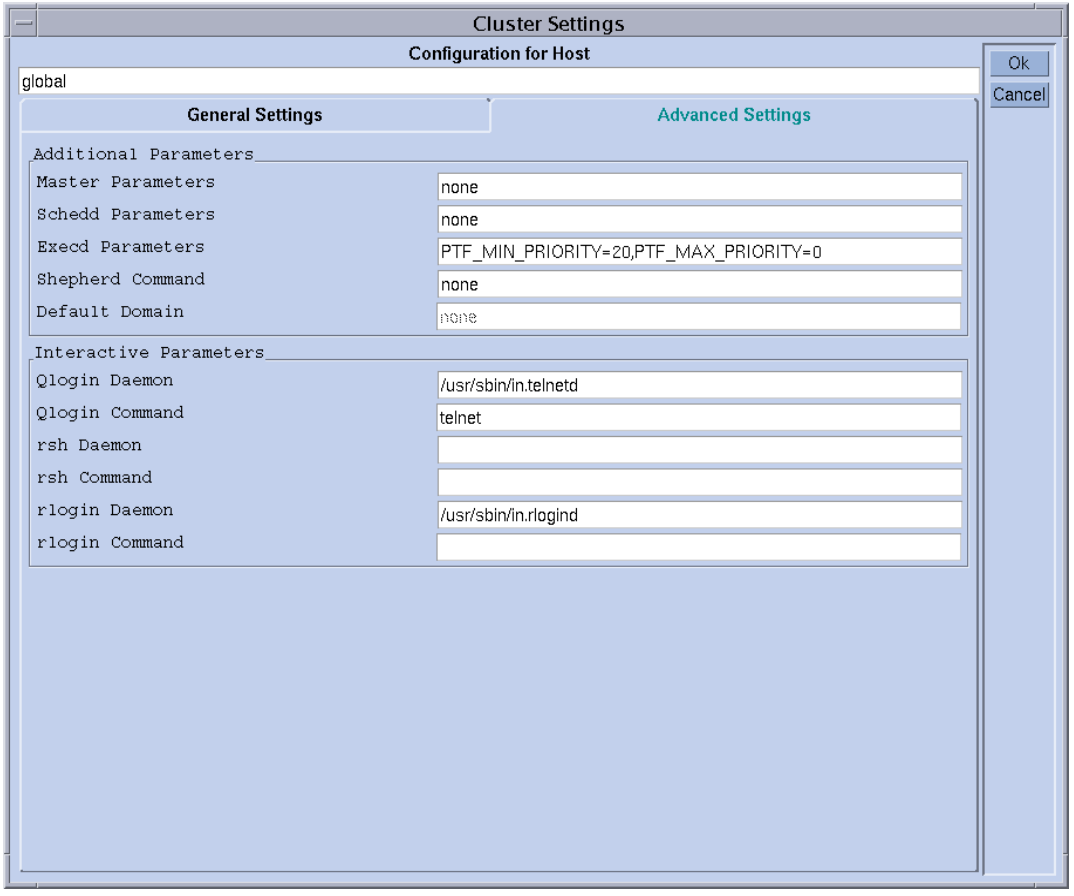


図 6-11　「クラスタ設定」ダイアログボックス - 高度設定

変更を終えたら、右上隅の「了解」ボタンをクリックして、変更した構成を登録します。「キャンセル」をクリックすると、変更は廃棄されます。どちらの場合も、ダイアログボックスは閉じます。

クラスタ構成パラメータについては、`sge_conf` のマニュアルページを参照してください。

第7章

キュー構成とキューカレンダの構成

この章では、Sun Grid Engine, Enterprise Edition 5.3 のキューおよびキューカレンダの構成に関する予備知識的な情報を提供し、その構成方法を説明します。

具体的には、この章では以下の作業を行う方法を説明します。

キューの構成

Sun Grid Engine, Enterprise Edition のキューはさまざまなカテゴリのジョブのコンテナであり、同じカテゴリに属する複数のジョブの並行実行に必要な資源を提供します。ジョブが Sun Grid Engine, Enterprise Edition のキューで待機することはなく、ディスパッチされるとただちに実行が開始されます。Sun Grid Engine, Enterprise Edition ジョブの唯一の待機場所は、Sun Grid Engine, Enterprise Edition スケジューラのジョブ保留リストです。

Sun Grid Engine, Enterprise Edition のキューを構成すると、そのキュー属性が sge_qmaster に登録されます。構成されたキューはすぐにクラスタ全体、また Sun Grid Engine, Enterprise Edition プール内のあらゆるホストのどの Sun Grid Engine, Enterprise Edition ユーザーからも見えるようになります。

▼ QMON からキューを構成する

1. **QMON のメインメニューで「キュー制御」ボタンをクリックします。**
2. **「キュー制御」ダイアログボックスで「追加」または「変更」ボタンをクリックします。**

   「キュー構成」ダイアログボックスが開きます。「キュー制御」ダイアログボックスとキューのステータスを監視、操作する機能については、137 ページの「QMON からキューを制御する」の節を参照してください。「キュー制御」ダイアログボックスを初めて開いた場合は、「一般パラメータ」フォームが表示されます (171 ページの「一般的なパラメータを設定する」を参照)。

3. **以下の説明に従って構成の変更を行います。**

   このダイアログボックスの上部にある「キュー」および「ホスト名」フィールドは、操作対象となるキューを示します。キューを変更する場合は、「キュー構成」ダイアログボックスを開く前に「キュー制御」ダイアログボックスでその既存のキューを選択する必要があります。キューを追加する場合は、そのキュー名とキューが存在するホストを指定する必要があります。

   「キュー構成」ダイアログボックスの「ホスト名」フィールドのすぐ下に、便利な 3 つのボタンが用意されています。「クローン作成」ボタンと「リセット」ボタン、「再表示」ボタンです。「クローン作成」ボタンでは、キュー選択リストを使用して既存のキューのすべてのパラメータをインポートすることができます。「リセット」ボタンは、テンプレートキューの構成を読み込みます。「再表示」ボタンは、「キュー構成」ダイアログボックスが開いている間に変更された他のオブジェクトの構成を読み込みます。「再表示」ボタンについての詳細は、178 ページの「ユーザー複合を設定する」と 180 ページの「ユーザーアクセスの設定をする」を参照してください。

   ダイアログボックスの右上隅の「了解」ボタンは変更を sge_qmaster に登録し、その下の「キャンセル」ボタンはすべての変更を廃棄します。両方のボタンともダイアログボックスを閉じます。

   キューの定義に使用可能なパラメータは 10 組あります。

   - 一般 - 171 ページの「一般的なパラメータを設定する」
   - 実行方法 - 173 ページの「実行方法関係のパラメータを設定する」
   - チェックポイント - 174 ページの「チェックポイント関係のパラメータを設定する」

- 負荷 / 一時停止しきい値 - 175 ページの「負荷および一時停止しきい値を設定する」
- 制限 - 176 ページの「制限を設定する」
- 複合 - 178 ページの「ユーザー複合を設定する」
- 従属 - 179 ページの「従属キューを設定する」
- ユーザーアクセス - 180 ページの「ユーザーアクセスの設定をする」
- プロジェクトアクセス - 182 ページの「プロジェクトアクセスの設定をする」
- 所有者 - 183 ページの「所有者を設定する」

「キューパラメータ」タブを使用して設定するパラメータセットを選択してください。

▼ 一般的なパラメータを設定する

- **「一般」パラメータセットを選択します。**
  図 7-1 に示すような画面が表示されます。

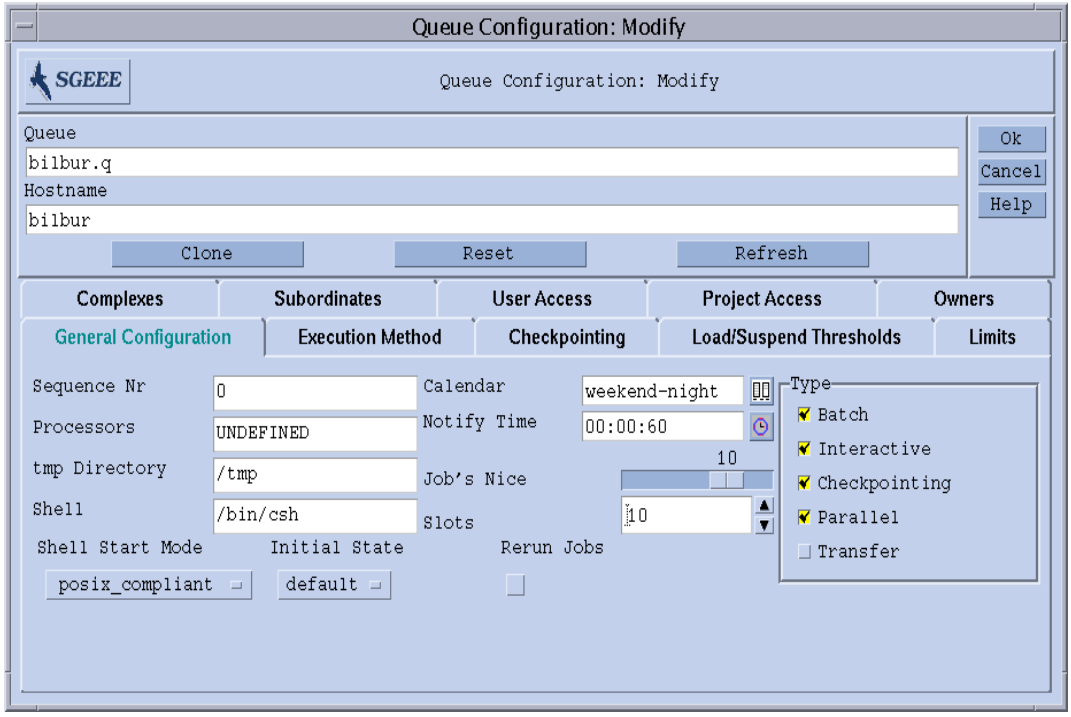


図 7-1 キュー構成 - 一般パラメータ

提供されているフィールドで、、以下のパラメータを設定できます。

- キューの連続番号
- プロセッサ - キューで実行するジョブが使用するプロセッサセットの指定。オペレーティングシステムアーキテクチャによっては、1-4,8,10 というように範囲指定するものもあれば、単にプロセッサセットの整数識別子を入力するものあります。詳細は、Sun Grid Engine, Enterprise Edition ディストリビューションの `doc` ディレクトリにある `arc_depend_*.asc` ファイルを参照してください。
- 一時ディレクトリパス
- ジョブスクリプトの実行に使用するデフォルトのコマンドインタプリタ (**シェル**)
- キューに関連付けるカレンダ - キューの当番時間と非番時間を定義します。
- `SIGUSR1/SIGUSR2` 通知シグナルを送信してから一時停止 / 終了シグナルを送信するまでの待ち時間 (**通知**)
- キュー内のすべてジョブの開始に使用する `nice` 値 - `0` は、システムデフォルトを使用することを意味します。
- キューで並行実行可能なジョブ数 (ジョブスロット)
- キューおよびキューで実行可能なジョブの種類 - 複数の種類を選択できます。
- **シェル起動モード** - ジョブスクリプトの実行を開始するモードです。
- **初期状態** - キューが新規追加されたとき、あるいはキューホストで動作する `sge_execd` が再起動された場合にキューが復元されたときの状態です。
- システムクラッシュなどの原因で中止されたジョブに適用するキューのデフォルトの**再実行**ポリシー - このポリシーは、`qsub -r` オプションまたは「ジョブの実行依頼」ダイアログボックス (図 4-9 を参照) を使用して書き換えることができます。

これらのパラメータについての詳細は、`queue_conf` のマニュアルページを参照してください。

▼ 実行方法関係のパラメータを設定する

- 「実行方法」パラメータセットを選択します。
  図 7-2 に示すような画面が表示されます。

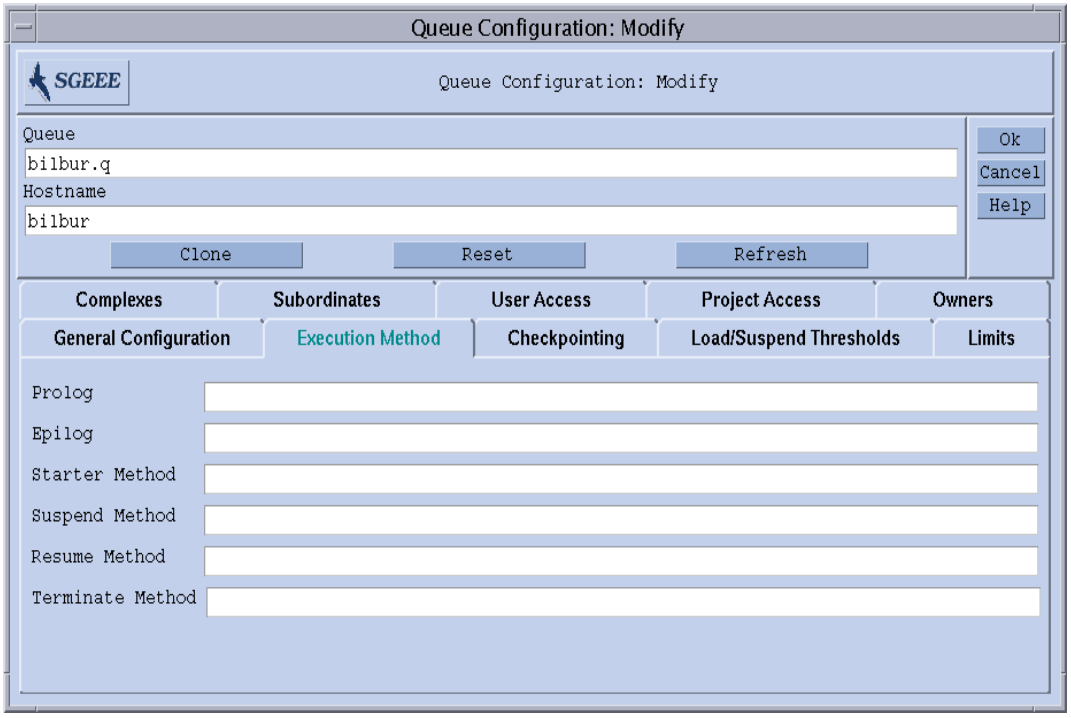


図 7-2　　キュー構成 - 実行方法パラメータ

提供されているフィールドで、以下のパラメータを設定できます。

- ジョブスクリプトを開始する前およびジョブが完了した後、ジョブと同じ環境を使用して実行するキュー固有のプロローグとエピローグスクリプト
- Sun Grid Engine, Enterprise Edition のデフォルトの開始 / 一時停止 / 再開 / 終了方法を書き換えるキュー固有の方法 - 指定された方法で対応する処理がジョブに適用されます。

これらのパラメータについての詳細は、`queue_conf` のマニュアルページを参照してください。

▼ チェックポイント関係のパラメータを設定する

- 「チェックポイント」パラメータセットを選択します。
  図 7-3 に示すような画面が表示されます。

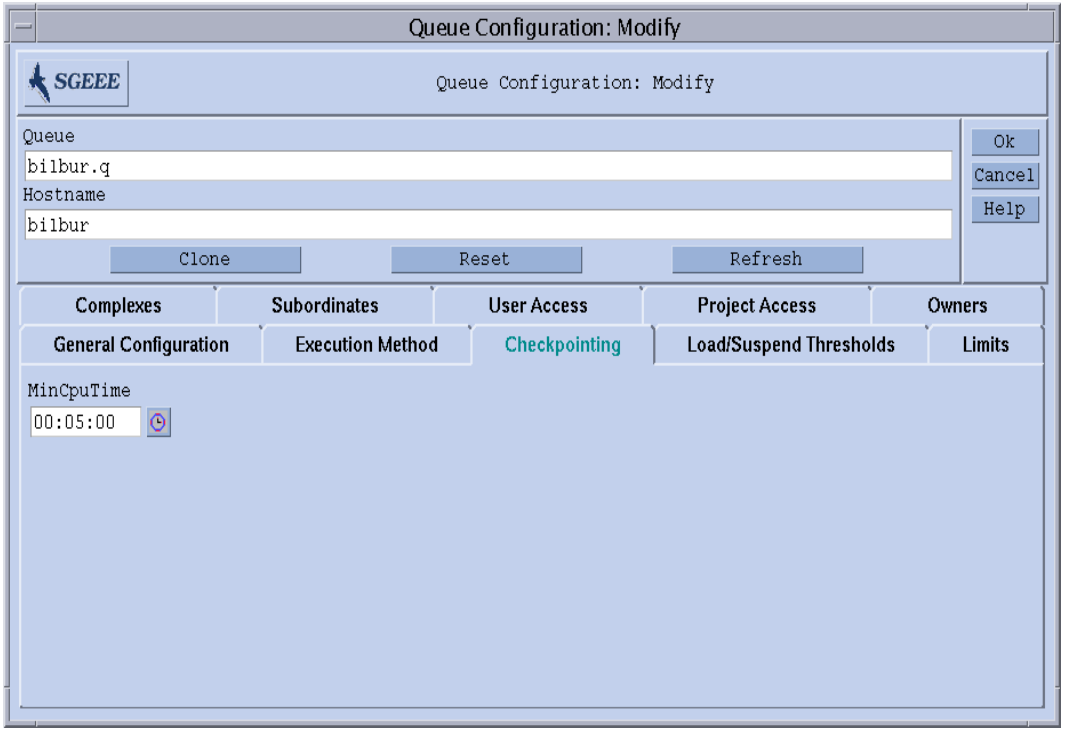


図 7-3 キュー構成 - チェックポイントパラメータ

提供されているフィールドで、、以下のパラメータを設定できます。

- 定期的なチェックポイント間隔 (**最小 CPU 時間**)

このパラメータについての詳細は、`queue_conf` のマニュアルページを参照してください。

▼ 負荷および一時停止しきい値を設定する

- 「負荷 / 一時停止しきい値」パラメータセットを選択します。
  図 7-4 に示すような画面が表示されます。

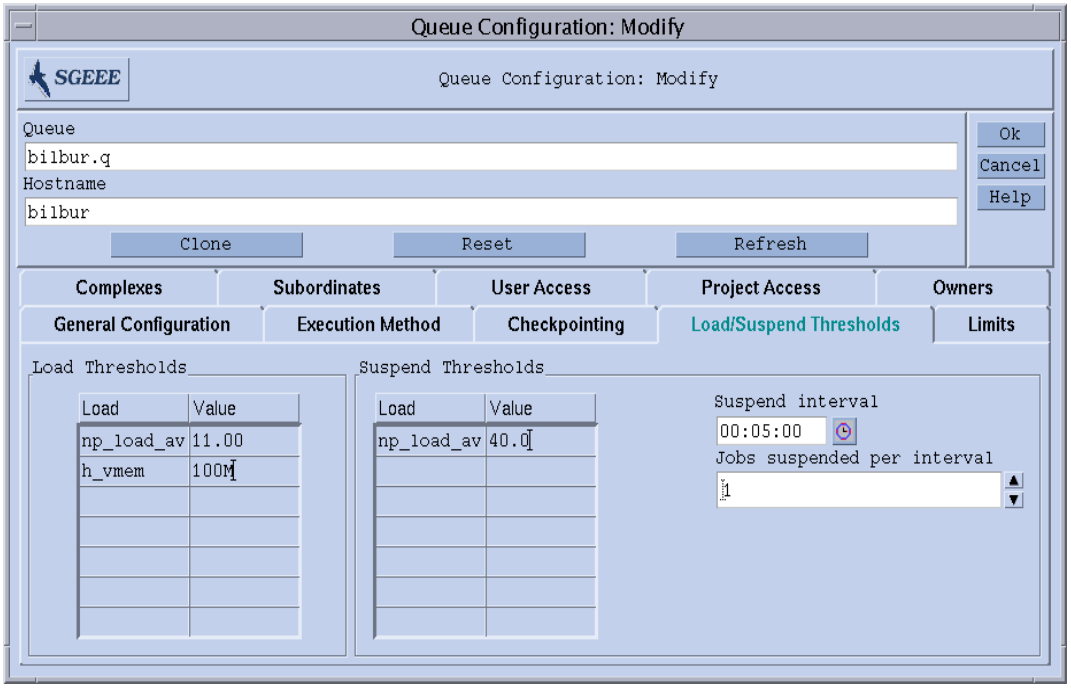


図 7-4 キュー構成 - 負荷 / 一時停止しきい値

提供されているフィールドで、以下のパラメータを設定できます。

- 「負荷しきい値」表と「一時停止しきい値」表 - 負荷パラメータと消費可能複合属性に対する過負荷しきい値を定義します (191 ページの「複合」参照)。

  負荷しきい値を超える過負荷になると、キューは Sun Grid Engine, Enterprise Edition からそれ以上ジョブを受け付けなくなります。一時停止しきい値を超えると、キュー内のジョブが一時停止されるか、負荷が軽減されます。表には、現在設定されているしきい値が表示されます。既存のしきい値は、マウスの左ボタンで「値」フィールドをダブルクリックすることによって選択、変更することができます。新しいしきい値を変更するには、表の最上部の「名前」または「値」ボタンをクリックします。この操作によって、キューに関連付けられている有効な属性のすべてを列挙した選択リストが開きます。図 7-6 は、その「属性の選択」ダイアログボックスを示しています。どれか属性を選択し、「了解」ボタンをクリックして選択を確定すると、その属性が対応するしきい値表の「名前」列に追加され、その「値」フィールドにポインタが移動します。選択したエントリを削除するには、Ctrl-D を押すか、マウスの右ボタンをクリックして削除ボックスを開き、削除を確定します。

- キューのホストになっているシステムの負荷を軽減するために指定時間の間一時停止するジョブ数
- 一時停止しきい値を下回らない場合にさらにジョブを一時停止する時間

これらのパラメータについての詳細は、queue_conf のマニュアルページを参照してください。

▼ 制限を設定する

- **「制限」パラメータセットを選択します。**
  図 7-5 に示すような画面が表示されます。

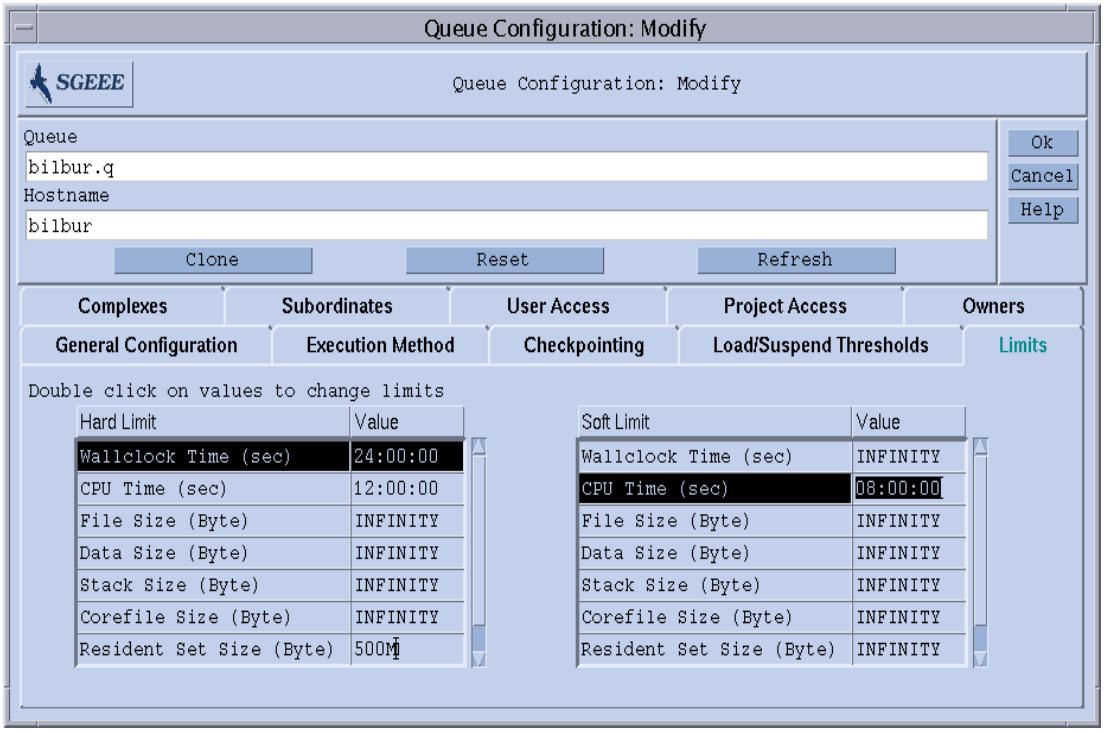


図 7-5 キュー構成 - 制限

提供されているフィールドで、、以下のパラメータを設定できます。

- キューで実行するジョブに課すハードおよびソフト制限。

制限値を変更するには、その制限エントリの「値」フィールドをダブルクリックします。このとき「値」フィールドを 2 回ダブルクリックすると、メモリーまたは時間制限値用の入力ダイアログボックスが開きます (図 7-6 と 図 7-7 を参照)。

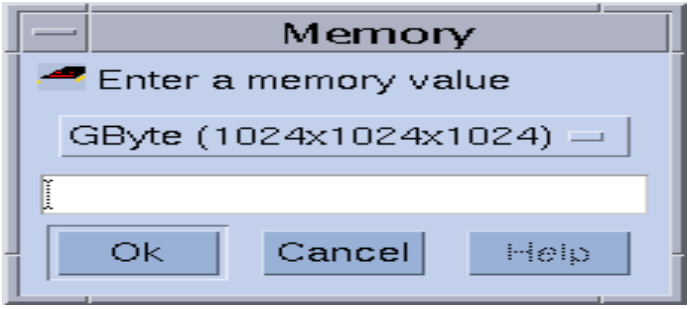


図 7-6　「メモリー」入力ダイアログボックス

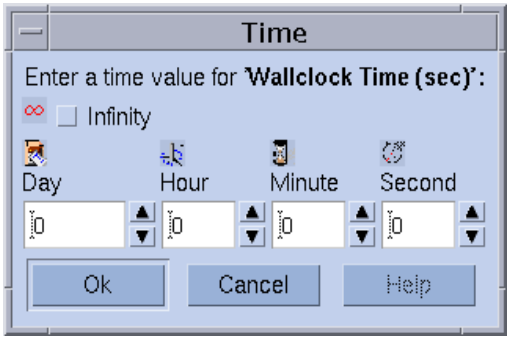


図 7-7　「時間」入力ダイアログボックス

さまざまなオペレーティングシステム別の制限関係の個々のパラメータとその意味についての詳細は、`queue_conf` および `setrlimit` にマニュアルページを参照してください。

▼ ユーザー複合を設定する

- 「ユーザー複合」パラメータセットを選択します。

  図 7-8 に示すような画面が表示されます。

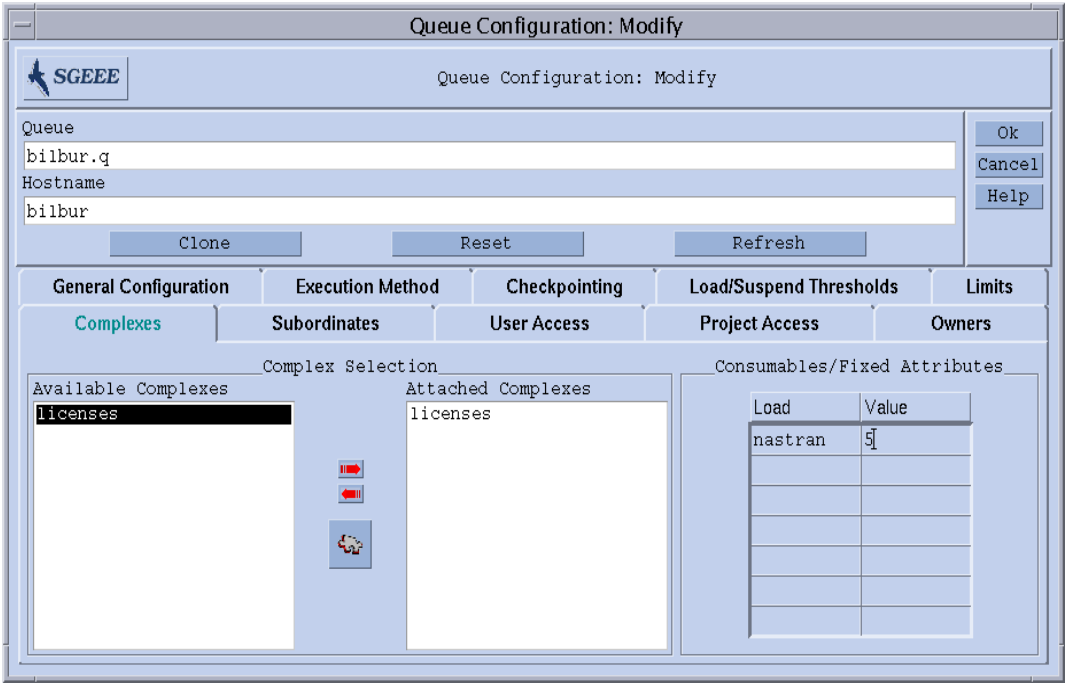


図 7-8 キュー構成 - ユーザー複合

提供されているフィールドで、、以下のパラメータを設定できます。

- キューに関連付けるユーザー定義の複合セット (197 ページの「ユーザー定義の複合」を参照)

  「複合選択」ボックス中央の赤い矢印を使用して、ユーザー定義の複合をキューに関連付けたり、関連付け解除したりできます。

- キューで使用可能な複合パラメータセットから選択したパラメータの値定義

  使用可能な複合パラメータは、デフォルトでは、グローバル複合、ホスト複合、関連付けられているユーザー定義の複合から集められます。属性は消費可能または固定パラメータのいずれかです。キューの値の定義では、消費可能属性の場合はキューが管理する資源能力、固定属性の場合は単に固定のキュー固有の値を定義します (詳細は、191 ページの「複合」を参照)。値が明示的に定義された属性は、「消費可能 / 固定属性」表示に表示されます。既存の属性は、対応する「値」フィールドをダブルクリックすることによって選択、変更することができます。新しい属性定義を追加するには、表の最上部の「名前」または「値」ボタンをクリックします。この操作によって、キューに関連付けられている有効な属

性のすべてを列挙した選択リストが開きます。図 6-6 は、「属性の選択」ダイアログボックスを示しています。どれか属性を選択し、「了解」ボタンをクリックして選択を確定すると、その属性が対応するしきい値表の「名前」列に追加され、その「値」フィールドにポインタが移動します。選択したエントリを削除するには、Ctrl-D を押すか、マウスの右ボタンをクリックして削除ボックスを開き、削除を確定します。

これらのパラメータについての詳細は、queue_conf のマニュアルページを参照してください。

「複合構成」アイコンボタンをクリックすると、「複合構成」ダイアログボックスが開きます (第 8 章、191 ページの「複合の概念」の図 8-5 の例を参照)。「複合構成」ダイアログボックスを使用して、キューにユーザー定義の複合を関連付けまたは関連付け解除する前に現在の複合構成を確認したり、変更したりできます。

▼ 従属キューを設定する

- **「従属」パラメータセットを選択します。**

  図 7-9 に示すような画面が表示されます。

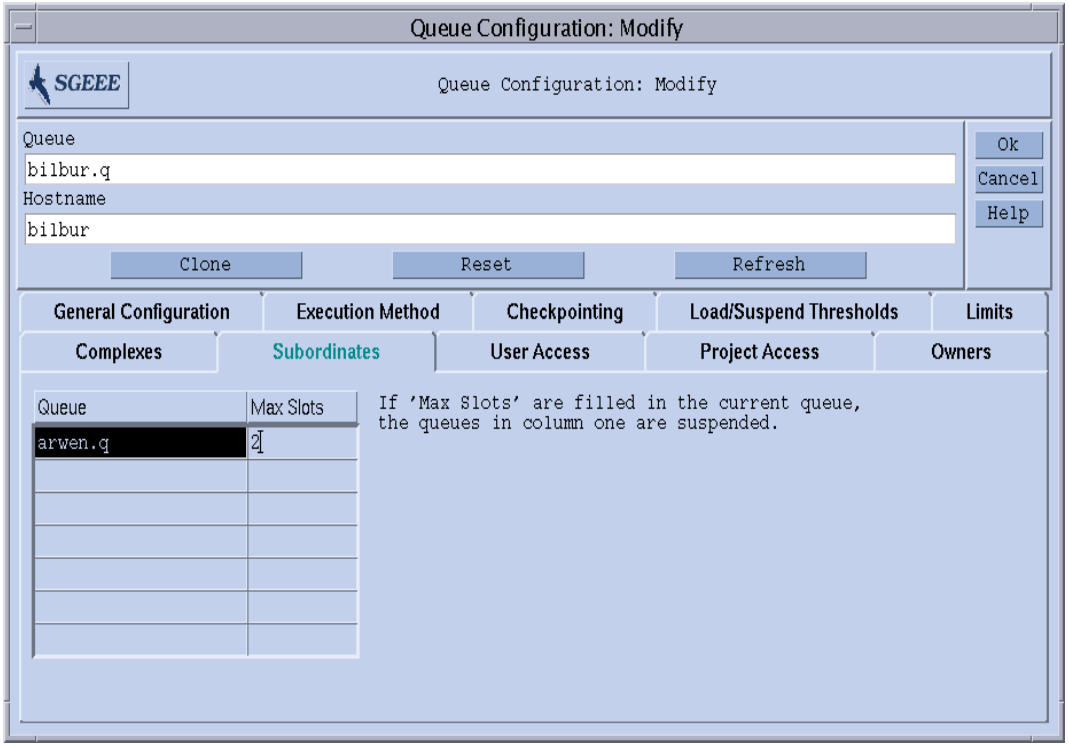


図 7-9　　キュー構成 - 従属

提供されているフィールドで、、以下のパラメータを設定できます。

- 現在のキューに従属するキュー

  従属キューは、構成しているキューが「使用中」になると一時停止され、使用中でなくなると停止解除されます。任意の従属キューに、一時停止を開始するために、最低限、構成しているキューで占有される必要があるジョブスロット数を設定することができます。ジョブスロット値が指定されていない場合、キューの一時停止を開始するには、すべてのスロットが埋まる必要があります。

これらのパラメータについての詳細は、queue_conf のマニュアルページを参照してください。

従属キュー機能は、スタンドアロンのキューばかりでなく、優先順付けしたキューを実現する場合に使用してください。

▼ ユーザーアクセスの設定をする

- **「ユーザーアクセス」パラメータセットを選択します。**

  図 7-10 に示すような画面が表示されます。

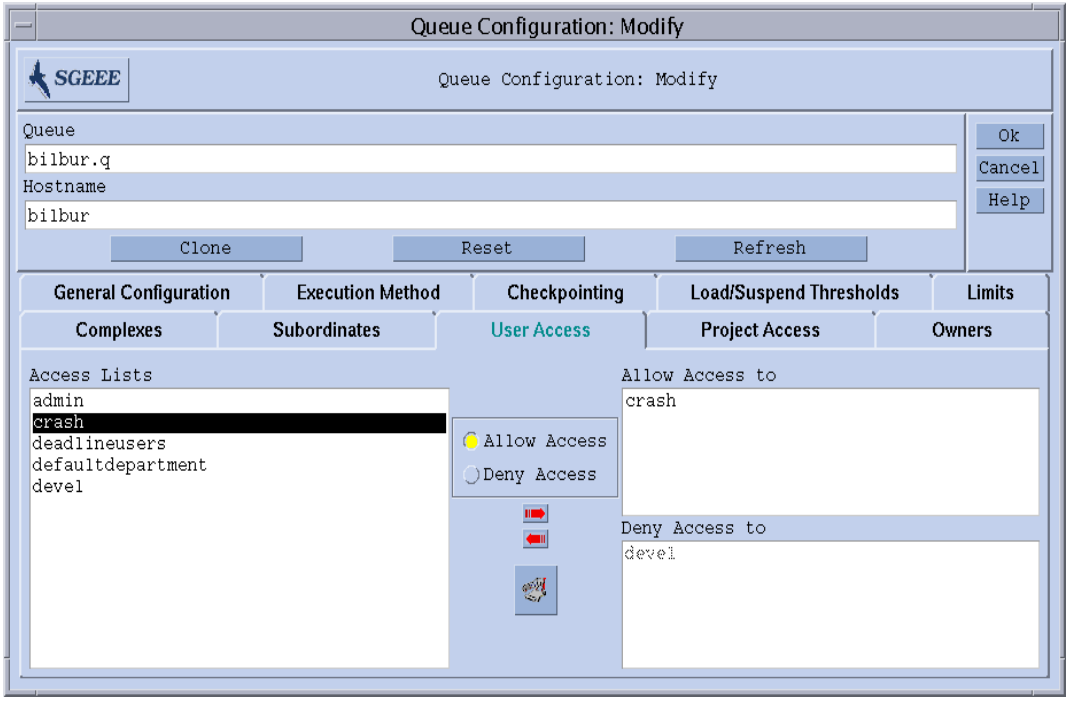


図 7-10　　キュー構成 - ユーザーアクセス

提供されているフィールドで、以下のパラメータを設定できます。

- キューへのアクセス許可または拒否リストに関連付けるユーザーアクセスリスト

  許可リストに登録されているアクセスリストに属するユーザーまたはユーザーグループは、キューにアクセスできます。拒否リストに関連付けられているユーザーまたはユーザーグループはキューにアクセスできません。許可リストが空の場合は、拒否リストに明示的に登録されていなくても、アクセスは無制限になります。

これらのパラメータについての詳細は、`queue_conf` のマニュアルページを参照してください。

中央のボタンをクリックすることによって、「アクセスリスト構成」ダイアログボックスを開くことができます (68 ページの「ユーザーのアクセス権」を参照)。

▼ プロジェクトアクセスの設定をする

- 「プロジェクトアクセス」パラメータセットを選択します。

  図 7-11 に示すような画面が表示されます。

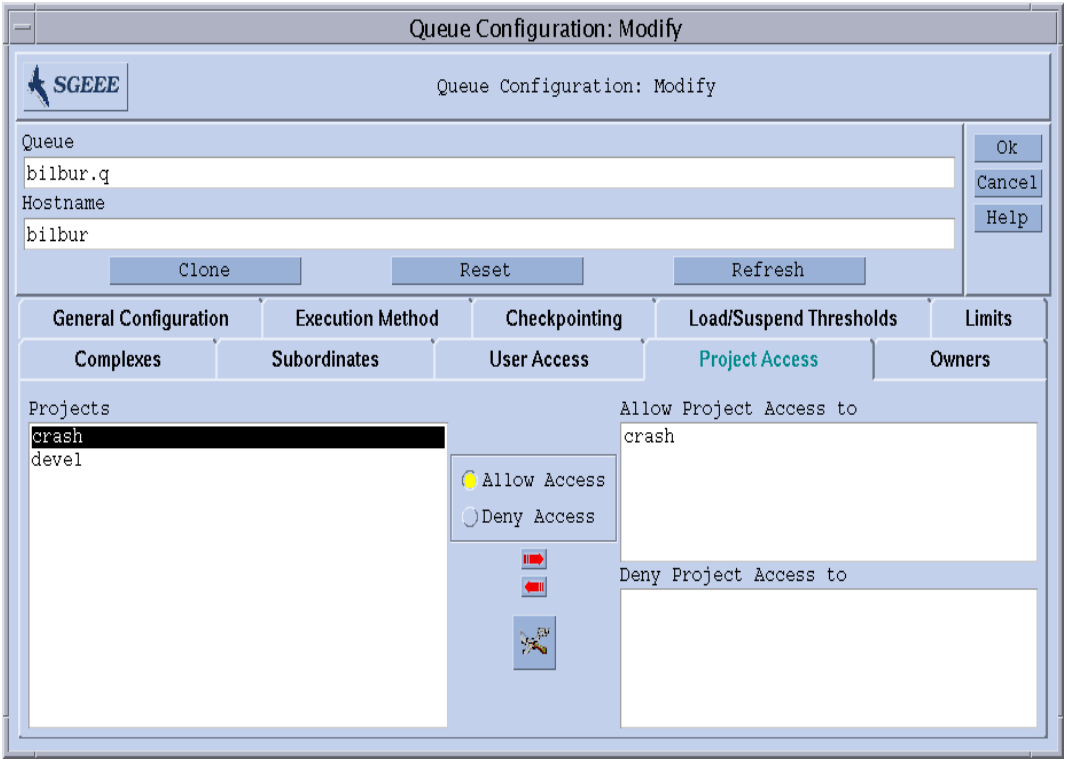


図 7-11 キュー構成 - プロジェクトアクセス

提供されているフィールドで、以下のパラメータを設定できます。

- キューへのアクセスを許可または拒否するプロジェクト

  許可されたプロジェクトリストに登録されたプロジェクトに実行依頼されたジョブは、キューを利用することができます。拒否されたプロジェクトに実行依頼されたジョブは、キューにディスパッチされません。

これらのパラメータについての詳細は、`queue_conf` のマニュアルページを参照してください。

中央のボタンをクリックすることによって、「プロジェクト構成」ダイアログボックスを開くことができます (233 ページの「プロジェクト」を参照)。

▼ 所有者を設定する

- 「所有者」パラメータセットを選択します。
  図 7-12 に示すような画面が表示されます。

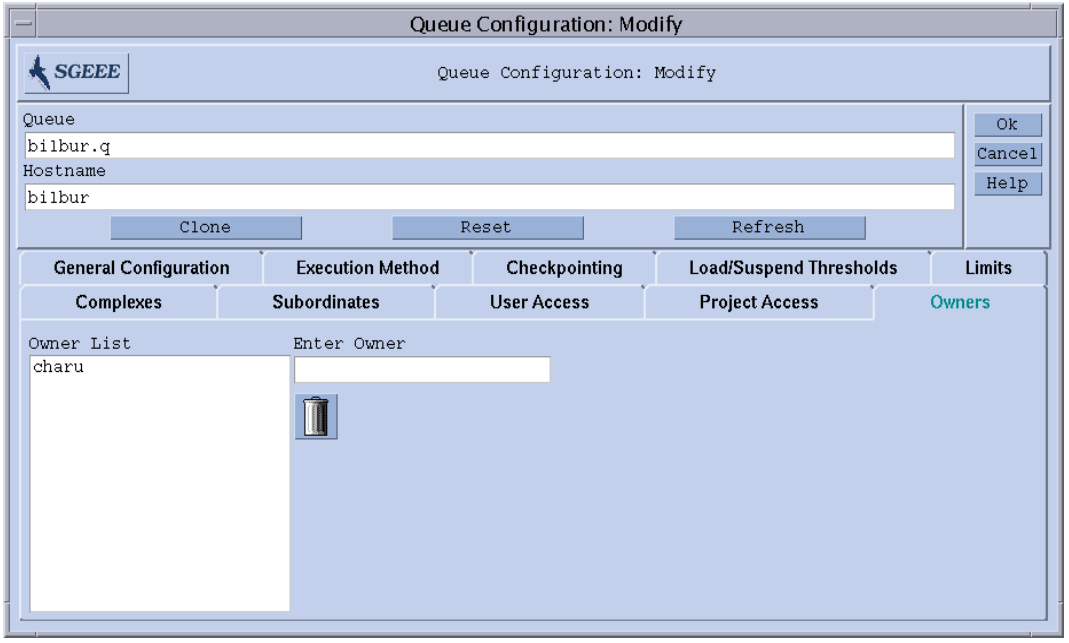


図 7-12　　キュー構成 - 所有者

提供されているフィールドで、以下のパラメータを設定できます。

- キューの所有者のリスト

  キューの所有者には、キューを一時停止 / 停止解除、あるいは使用不可 / 使用可能にする権限が付与されます。キュー所有者リストには、正当な任意のユーザーアカウントを登録することができます。リストからユーザーアカウントを削除するには、「所有者リスト」欄でユーザーアカウントを選択し、ダイアログボックスの右下にあるゴミ箱アイコンをクリックします。

これらのパラメータについての詳細は、`queue_conf` のマニュアルページを参照してください。

▼ コマンド行からキューを構成する

- **目的のキュー構成作業に応じて適切な引数を付けて次のコマンドを入力します。**

```
# qconf options
```

qconf コマンドには以下のオプションがあります。

- qconf -aq *[queue_name]*

  キューの追加 - このコマンドは、エディタ (デフォルトの vi か、$EDITOR 環境変数に指定されたエディタ) を使用して、キュー構成用のテンプレートを開きます。省略可能なパラメータの *queue_name* が指定された場合は、そのキューの構成がテンプレートとして使用されます。テンプレートの内容を変更し、ディスクに保存することによって、キューを構成してください。変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の queue_conf の項を参照してください。

- qconf -Aq *file_name*

  キューの追加 - ファイル *file_name* を使用してキューを定義します。この定義ファイルは、qconf -sq *queue_name* によって生成されたファイルでもかまいません (下記を参照)。

- qconf -cq *queue_name[,...]*

  キューの後処理 - 指定されたキュー (複数指定可能) のステータスをクリアして、実行中のジョブのない休止状態にします。ステータスは、現在のステータスに関係なくリセットされます。このオプションはエラー状態を解除するときに便利ですが、通常の運用モードでは使用しないでください。

- qconf -dq *queue_name[,...]*

  キューの削除 - 使用可能なキューのリストから、引数リストに指定されたキューを削除します。

- qconf -mq *queue_name*

  キューの変更 - 指定されたキューを変更します。エディタ (デフォルトの vi か、$EDITOR 環境変数に指定されたエディタ) を使用し、キューの構成が表示されます。この構成を変更し、ディスクに保存することによって、キューを変更します。

- qconf -Mq *file_name*

  キューの変更 - ファイル *file_name* を使用してキューの構成を変更します。この定義ファイルは、qconf -sq *queue_name* によって生成されたファイルでもかまいません (下記を参照)。

- `qconf -sq` *[queue_name[,...]]*

  キューの表示 - デフォルトのキュー構成用テンプレートを表示するか (引数が省略された場合)、コンマ区切りの引数リストに指定されたキューの現在の構成を表示します。

- `qconf -sql`

  キューのリストの表示 - 構成済みのすべてのキューのリストを表示します。

キューカレンダ

キューカレンダは、日付や曜日、1 日の時刻、あるいはそれらの組み合わせに基づいて Sun Grid Engine, Enterprise Edition のキューの可用性を定義します。任意の時点でそのステータスが変更されるようキューを構成することができます。キューのステータスは、使用不可、使用可能、一時停止、停止解除 (再開) に変更することができます。

Sun Grid Engine, Enterprise Edition には、サイトに固有のカレンダセットを定義する機能があります。それぞれのカレンダには、任意のステータス変更とその変更が発生するカレンダイベントが含まれます。キューはそうしたカレンダを参照することができます。すなわち、各キューを 1 つのカレンダに関連付け (関連付けなくても可)、そのカレンダに定義されている可用性プロファイルを適用することができます。この逆に、キューをカレンダの関連付けから除外することもできます。

カレンダ形式の構文は、`calendar_conf` のマニュアルページで詳しく説明しています。以下では、対応する機能を説明するとともに、いくつか例を紹介します。

▼ QMON からキューカレンダを構成する

1. **QMON のメインメニューから「カレンダ構成」をクリックします。**

   図 7-13 に示すような「キューカレンダ構成」ダイアログボックスが表示されます。

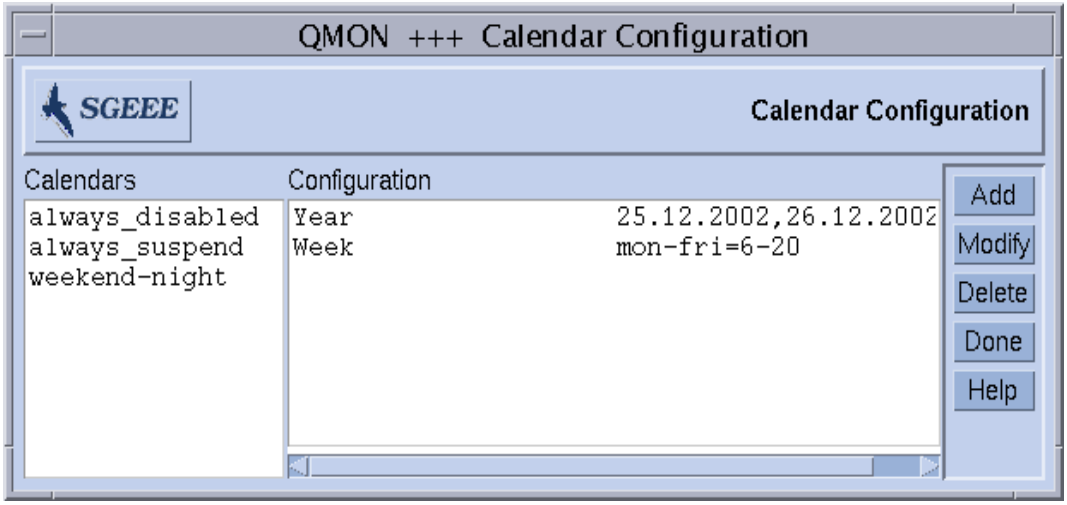


図 7-13　　カレンダ構成

画面の左側は「カレンダ」選択リストで、選択可能なカレンダが表示されます。

2. 「カレンダ」選択リストで変更または削除するカレンダをクリックします。
3. 以下のいずれか適切な操作を行います。
   a. 選択したカレンダを削除する場合は、画面右側にある「削除」ボタンをクリックします。
   b. 選択したカレンダを変更する場合は、「変更」ボタンをクリックします。
   c. カレンダを追加する場合は、「追加」ボタンをクリックします。

   どの場合も、図 7-14 に示すような「カレンダの定義」ダイアログボックスが開き、カレンダを削除、変更あるいは追加することができます。

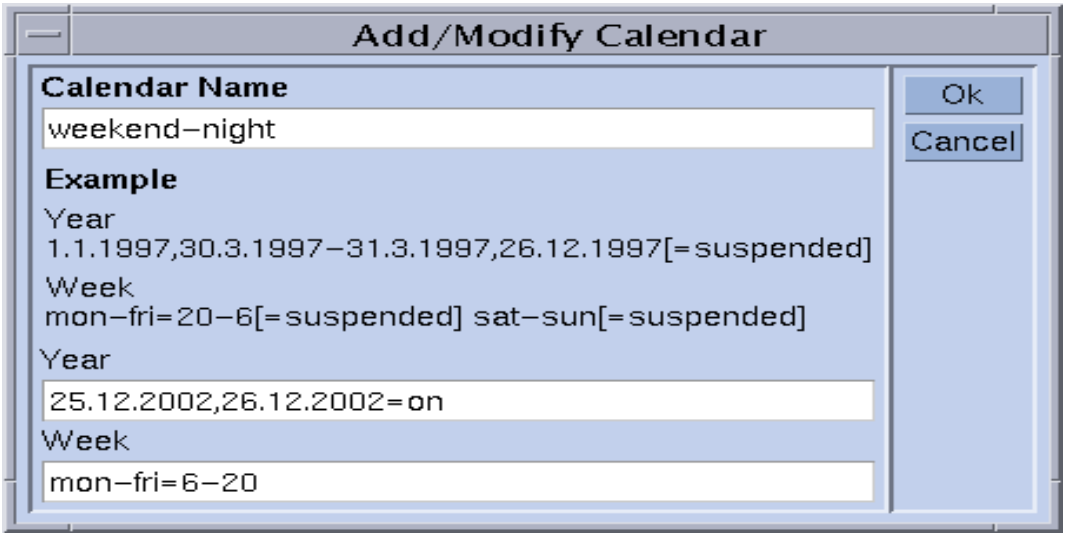


図 7-14　　カレンダの追加 / 削除 / 変更

4. **以下の説明に従って操作を進めます。**

「カレンダ名」入力フィールド - 変更の場合は、選択されたカレンダ名が表示されます。追加の場合は、このフィールドを使用して定義するカレンダの名前を入力することができます。「年」および「週」入力フィールドには、`calendar_conf` のマニュアルページで説明している構文を使用してカレンダイベントを定義することができます。

上記のカレンダ構成例は、営業時間外と週末に使用可能なキューに適しています。また、週末と同様に扱うようにクリスマス休暇を定義しています。

この構文についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `calendar_conf` の項を参照してください。実際の例も紹介しています。

キューにカレンダ構成を関連付けることによって、そのカレンダに定義されている可用性プロファイルがキューに割り当てられます。こうしたキューへのカレンダの関連付けは、図 7-15 に示すキュー構成の一般パラメータ画面で行います。「カレンダ」入力フィールドに関連付けるカレンダ名を指定します。フィールド横のアイコンボタンをクリックすると、構成されているカレンダのリストからなる選択ダイアログが表示されます。キューの構成についての詳細は、169 ページの「キューの構成」の節を参照してください。

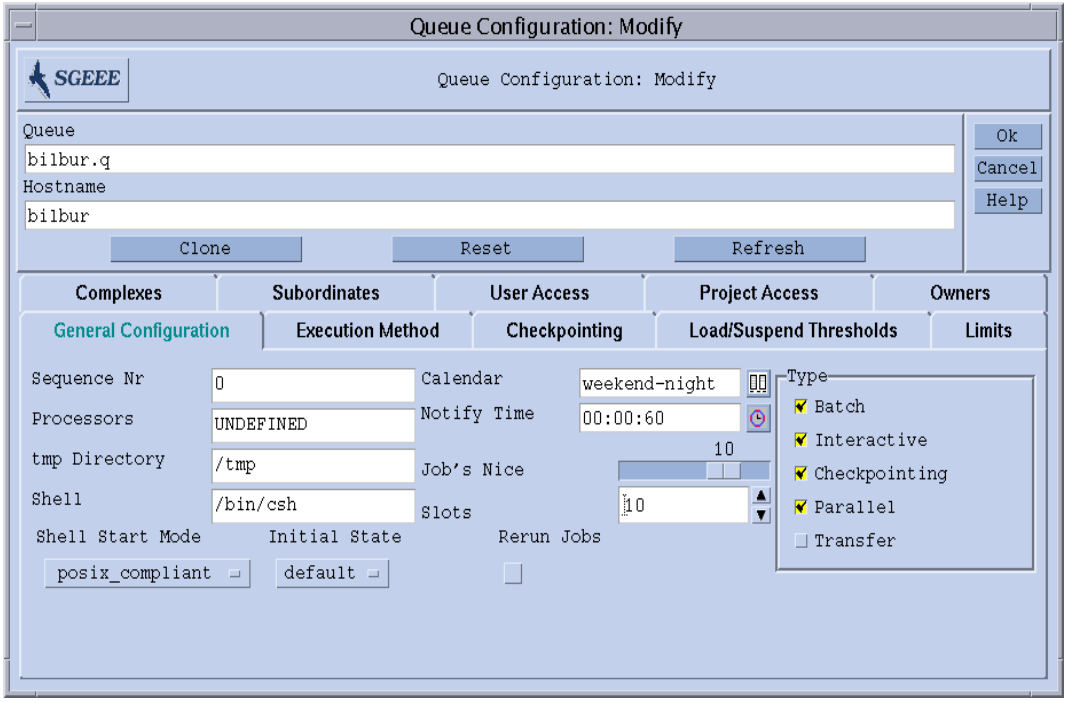


図 7-15　　キュー構成の一般パラメータ画面のカレンダ指定例

▼ コマンド行からカレンダを構成する

- 適切なスイッチを付けて次のコマンドを入力します。

```
% qconf switches
```

以下の 4 つのスイッチを使用できます。

- `qconf -Acal` または `-acal`

  カレンダの追加 - Sun Grid Engine, Enterprise Edition クラスタに新しいカレンダ構成を追加します。追加するカレンダはファイルから読み取るか (`-Acal`)、エディタを使用して構成用テンプレートを開き、カレンダを入力することができます。

- `qconf -dcal`

  カレンダの削除

- qconf -Mcal または -mcal

  カレンダの変更 - 既存のカレンダ構成を変更します。変更するカレンダはファイルから読み取るか (-Mcal)、エディタを使用してカレンダ構成を開き、新しい定義を入力することができます (-mcal)。

- qconf -scal または -scall

  カレンダの表示 - 既存のカレンダ構成を表示するか (-scal)、構成されているすべてのカレンダのリストを表示します (-scall)。

第8章

複合の概念

この章では、複合という、Sun Grid Engine, Enterprise Edition 5.3 の重要な概念について説明します。また、こうした複合に関する予備知識的な情報と関連する概念とともに、以下の作業を行う方法を詳しく説明します。

複合

複合の定義では、`qsub` または `qalter` の `-l` オプションを使用することによって、Sun Grid Engine, Enterprise Edition ジョブでユーザーが要求可能な資源属性と Sun Grid Engine, Enterprise Edition システムにおけるそれらパラメータの解釈方法に関するすべての情報を提供します。

複合はまた、Sun Grid Engine, Enterprise Edition の消費可能資源機能の枠組みを提供します。消費可能資源機能とは、関連付けられた能力とともに資源を識別する、クラスタ全体 (グローバル)、ホスト別、あるいはキュー関連の属性の定義を可能にする機能です。スケジューリングでは、Sun Grid Engine, Enterprise Edition ジョブの要求とともに資源の可用性が考慮されます。Sun Grid Engine, Enterprise Edition はまたブックキーピングを行い、能力利用計画をたてることによって、消費可能資源の過剰な予約を防ぎます。代表的な消費可能属性としては、たとえば使用可能な空きメモリー、使用されていないソフトウェアパッケージのライセンス、空きディスク容量、ネットワーク接続の使用可能な帯域幅などがあります。

もっと一般的な意味では、Sun Grid Engine, Enterprise Edition の複合は、キューやホスト、クラスタ属性の解釈方法を記述する手段として使用され、この記述には、属性名、属性の参照に使用可能なショートカット、属性値の型 (`STRING`、`TIME` など)、複合属性に割り当てる事前定義値、Sun Grid Engine, Enterprise Edition スケジューラの `sge_schedd` が使用する関係演算子、ジョブで属性が要求可能であるか

どうかを制御する要求可能フラグ、属性が消費可能属性であることを示す消費可能フラグ、消費可能属性に対する要求がジョブに明示的に指定されていない場合にそうした属性について考慮するデフォルト要求値などが含まれます。

図 8-1 に示す「QMON 複合構成」ダイアログボックスは、複合属性を定義例です。

▼ 複合構成を追加または変更する

1. **QMON メインメニューで「複合構成」ボタンをクリックします。**

   図 8-1 に示すような「複合構成」ダイアログボックスが表示されます。

2. **以下の節の説明に従って複合構成を追加または変更します。**
   - 194 ページの「キュー複合」
   - 194 ページの「ホスト複合」
   - 196 ページの「グローバル複合」
   - 197 ページの「ユーザー定義の複合」

   「複合構成」ダイアログボックスでは、既存の複合の定義を変更したり、ユーザー複合を新しく定義したりできます。

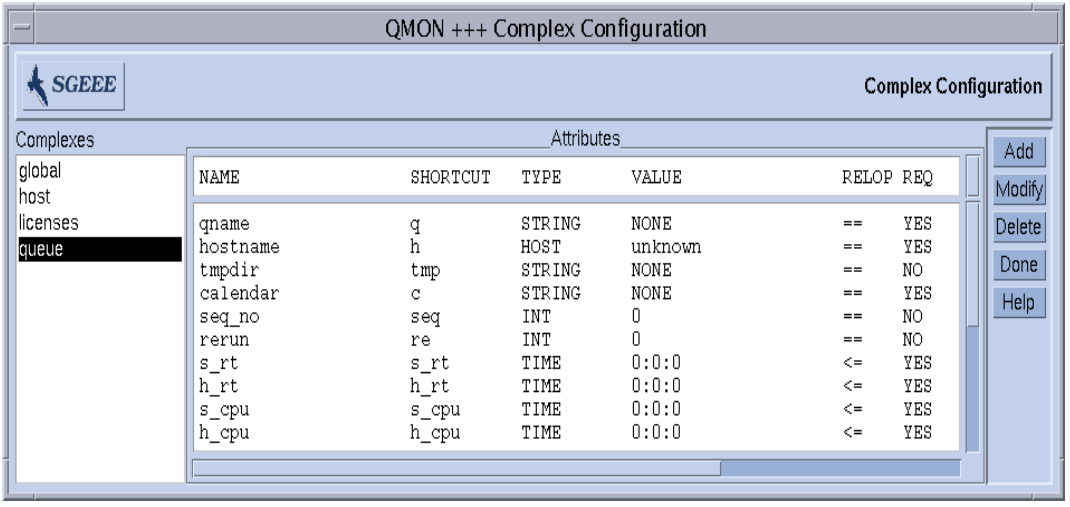


図 8-1 「複合構成」ダイアログボックス - キュー

ダイアログボックスの左側には、システムが認識しているすべての複合の選択リストが表示されます。複合を変更または削除する場合は、このリストを利用することができます。画面の右側には、それぞれの操作 (追加、変更、削除) を行うためのボタンがあります。新規の作成、あるいは既存の複合の変更では、図 8-2 に示すようなダイアログボックスが開きます。

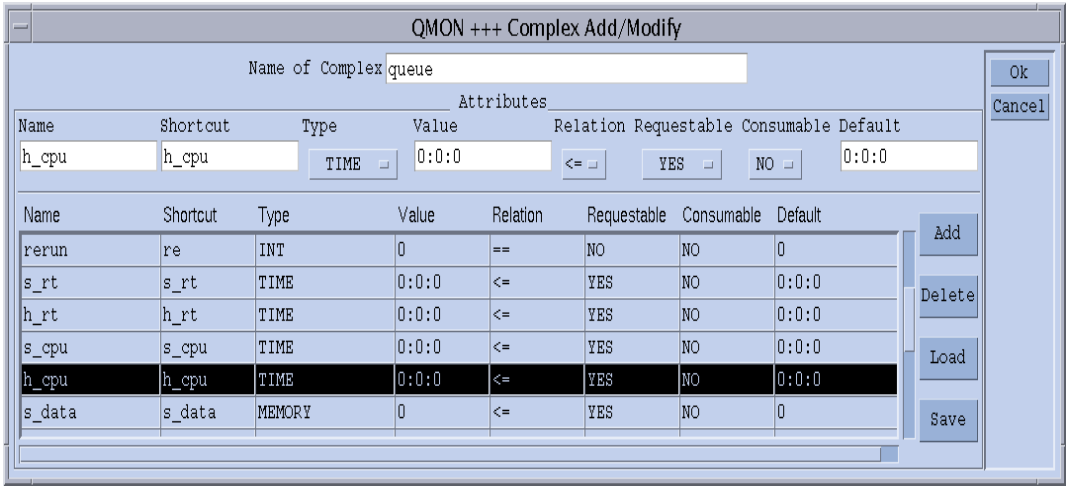


図 8-2　「複合の追加 / 変更」ダイアログボックス

このダイアログボックス最上部の「複合名」入力フィールドでは、複合名を入力または選択する必要があります。「複合の定義」表内の複合属性は、マウスの左ボタンで行を選択することによって変更することができます。「属性」ボックスの上部の定義フィールドとセレクタには、選択されたエントリの定義内容が表示されます。定義を変更して、「追加」ボタンをクリックすると、定義表にその変更が反映されます。

新規エントリは、定義フィールドをすべて埋めて、セレクタを使用し、最後に「追加」ボタンをクリックすることによって追加することができます。新しい属性を追加する場合は、属性表の行を選択しないでください。

「読み込み」および「保存」ボタンは、通常のファイルから複合構成を読み込んだり、通常のファイルに複合構成を保存したりします。このときファイル選択用のダイアログボックスが開いて、ファイルを選択することができます。「削除」ボタンは、複合構成内の選択されている行を削除します。

属性表の行と列の意味についての詳細は、`complex` のマニュアルページを参照してください。ダイアログボックス右上隅の「了解」ボタンは、新規または変更された複合構成を `sge_qmaster` に登録します。

複合の種類

Sun Grid Engine, Enterprise Edition の複合オブジェクトは、次の 4 種類の複合を統合します。

- キュー複合
- ホスト複合

- グローバル複合
- ユーザー定義の複合

以下では、種類別にこれらの複合を詳しく説明します。

キュー複合

キュー複合は、特殊名の `queue` で参照します。

デフォルトでは、キュー複合には、`queue_conf` に定義されているキュー構成のパラメータ群が含まれています。このキュー複合の第一の目的は、それらパラメータの解釈方法を定義するとともに、すべてのキューに適用できるようにすることを意図した追加属性用のコンテナを提供することにあります。つまり、キュー複合はユーザー定義の属性で拡張することができます。

特定のキューのコンテキストでキュー複合が参照されている場合、キュー複合の属性値は、そのキュー構成の対応する値によって置き換えられます (「**値**」列が書き換えられる)。

たとえば *big* というキューにキュー複合が設定された場合、キュー複合の属性 `qname` の「値」列 の値 `unknown` (図 8-1 を参照) は、big に設定されます。

この暗黙の値設定は、キュー構成で `complex_values` パラメータを使用することによって書き換えることができます (169 ページの「キューの構成」を参照)。この書き換えは、通常、消費可能属性に対して行います (201 ページの「消費可能資源を構成する」の節を参照)。たとえば仮想メモリーのサイズ制限の場合、キュー構成値はジョブ 1 つあたりの総メモリー使用量の制限に使用されるのに対し、`complex_values` リストの対応するエントリは、ホスト上またはキューに割り当てられている使用可能な仮想メモリーの合計量を定義するといった具合です。

管理者によってキュー複合に属性が追加された場合、特定のキューとの関連付けでは、その値は、キューの `complex_values` パラメータを使用して定義されるか、定義されていない場合は、デフォルトでキュー複合構成の `value` 列の値が使用されます。

ホスト複合

ホスト複合は特殊名の `host` で参照され、ホスト単位で管理することを意図したすべての属性の特性定義が含まれます (図 8-3 を参照)。ホスト関連の属性の標準セットは 2 つのカテゴリから構成されますが、上記のキュー複合と同様、拡張することができます。最初のカテゴリは、特にホスト単位の管理に適したいくつかのキュー構成の属性で構成されます。それらの属性は以下のとおりです。

- スロット
- `h_vmem`
- `s_fsize`
- `h_fsize`

(詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の queue_conf の項を参照してください。)

注 – キュー構成でこれらの属性があることと、ホスト複合でこれらの属性を定義することとの間に矛盾はありません。そうすることによって、ホストレベルとキューレベルの両方で対応する資源を管理することができます。たとえば、ホストに関して使用可能な仮想メモリー全体 (h_vmem) を管理できるとともに、その一部をそのホストのキューに関連付けることができます。

標準のホスト複合の 2 つ目の属性カテゴリはデフォルトの負荷値です。各 sge_execd は定期的に負荷を sge_qmaster に報告します。報告される負荷値は、CPU 負荷平均などの、Sun Grid Engine, Enterprise Edition 標準の負荷値、または Sun Grid Engine, Enterprise Edition 管理者が定義した負荷値のいずれかです (214 ページの「負荷パラメータ」の節を参照)。標準の負荷値の特性定義がデフォルトのホスト複合に含まれるのに対して、管理者定義の負荷値はホスト複合の拡張を必要とします。

一般にホスト複合は、標準以外の負荷パラメータを追加する目的ばかりでなく、ホストに割り当てるソフトウェアライセンス数やホストのローカルファイルシステムの使用可能なディスク容量などのホスト関連の資源を管理する目的でも拡張されます。

ホスト複合が特定のホストまたはそのホスト上のキューに関連付けられた場合、特定のホスト複合属性の値は、以下のいずれかによって決まります。

- キュー構成から属性を取得できる場合はキュー構成
- 報告された負荷値
- 対応するホスト構成の complex_values エントリの値の明示的な定義 (148 ページの「ホストの構成」の節を参照)。

上記のどれも該当しない場合は (値は負荷パラメータとされているのだが、sge_execd からその負荷値の報告がないなど)、ホスト複合構成の「**値**」フィールドが使用されます。

たとえば 全体の空き仮想メモリー属性の h_vmem はキュー構成では制限として定義され、標準の負荷パラメータとしても報告されます。ホスト上、およびそのホストのキューに関連付けられている仮想メモリーの使用可能量の合計は、そのホストの complex_values リストとそのキュー構成で定義することができます。h_vmem を消費可能資源として定義するとともに (201 ページの「消費可能資源」を参照)、このように定義することによって、メモリーの過剰予約 (しばしばスワップによるシステムパフォーマンスの低下の原因になる) を招くことなく、マシンのメモリーを効率よく利用することができます。

注 – システムのデフォルトの負荷属性では、「**ショートカット**」「**値**」「**関係**」「**要求可能**」「**消費可能**」「**デフォルト**」列のみ変更することができます。デフォルトの属性を削除しないでください。

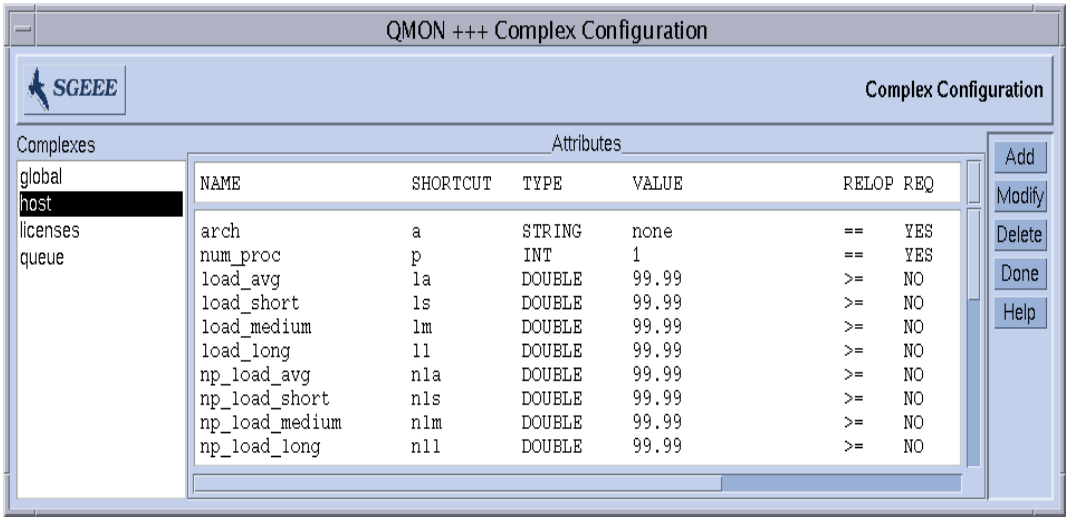


図 8-3　「複合構成」ダイアログボックス - ホスト

グローバル複合

グローバル複合は、特殊複合名の `global` で参照します。

グローバル複合構成のエントリは、ファイルサーバーの使用可能なネットワーク帯域幅、ネットワーク全体で使用可能なファイルシステムの空きディスク容量などのクラスタ全体の資源属性を表します (図 8-4 を参照)。負荷レポートに `GLOBAL` 識別子を含めることによって (214 ページの「負荷パラメータ」の節を参照)、グローバル資源属性を負荷レポートに関連付け、クラスタ内の任意のホストからグローバル負荷値を報告させることもできます。デフォルトでは、Sun Grid Engine, Enterprise Edition が報告するグローバル負荷値はないため、デフォルトのグローバル複合構成はありません。

グローバル複合属性の具体的な値は、グローバル負荷ポートか、`global` ホスト構成の `complex_values` パラメータにおける明示的な定義 (148 ページの「ホストの構成」の節を参照)、特定のホストまたはキューとの関連付けと対応する

complex_values リストにおける明示的な定義のいずれかで決定されます。上記のどれにも該当しない場合は (負荷値が報告されていないなど)、グローバル複合構成の「値」フィールドが使用されます。

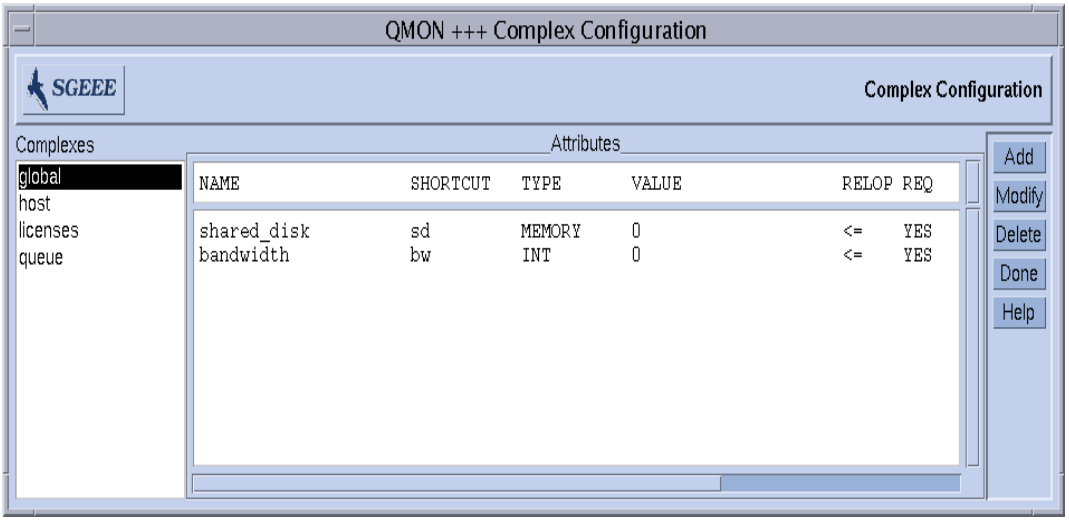


図 8-4　「複合構成」ダイアログボックス - グローバル

ユーザー定義の複合

Sun Grid Engine, Enterprise Edition 管理者は、ユーザ定義の複合を構成することによって、Sun Grid Engine, Enterprise Edition が管理する属性セットを拡張したり、特定のキューかホスト、またはその両方に対するそれら属性の影響を制限したりすることができます。ユーザー複合は、単に属性と、Sun Grid Engine, Enterprise Edition におけるそれら属性の取り扱い方法に関する定義の、名前付きの集合です。coplex_list キューとホスト構成パラメータを使用して、ユーザー定義の複合をキューかホスト、またはその両方に関連付けることができます (169 ページの「キューの構成」と 148 ページの「ホストの構成」を参照)。デフォルトの複合属性のほかに、関連付けられた複合に定義されているすべての属性が、キューおよびホストから利用できるようになります。

キューあるいはホストに関連付けられたユーザー定義の複合の属性の具体的な値は、キューまたはホスト構成の complex_values パラメータで設定する必要があります。設定されていていない場合は、ユーザー複合構成の「値」フィールドが使用されます。

一例として、ユーザー定義の複合 licenses を定義してみましょう。

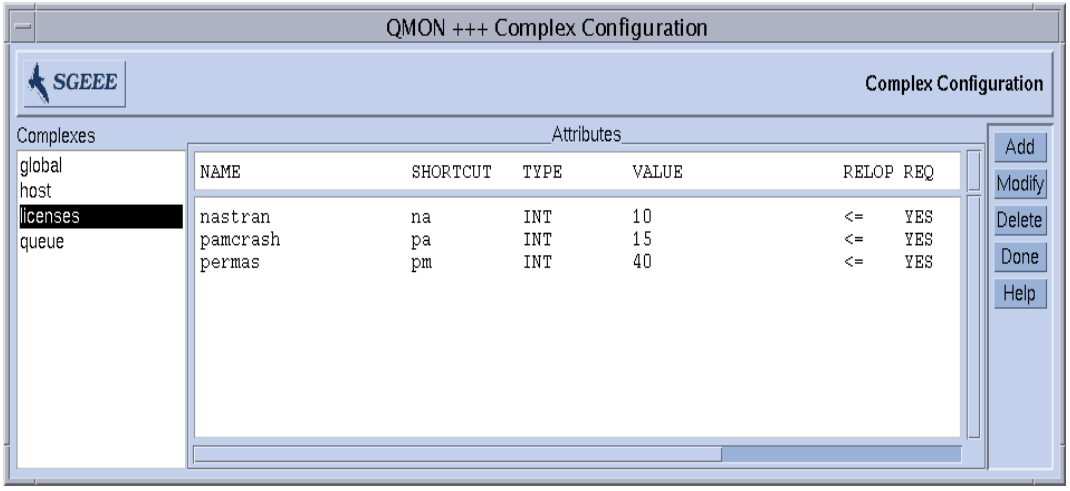


図 8-5 「複合構成」ダイアログボックス - licenses

図 8-6 のキュー構成の「複合」サブダイアログボックスで示しているように、関連付けられているユーザー定義の複合のリストに、この licenses 複合を追加します (キューの構成方法についての詳細は、169 ページの「キューの構成」と関連する節を参照)。

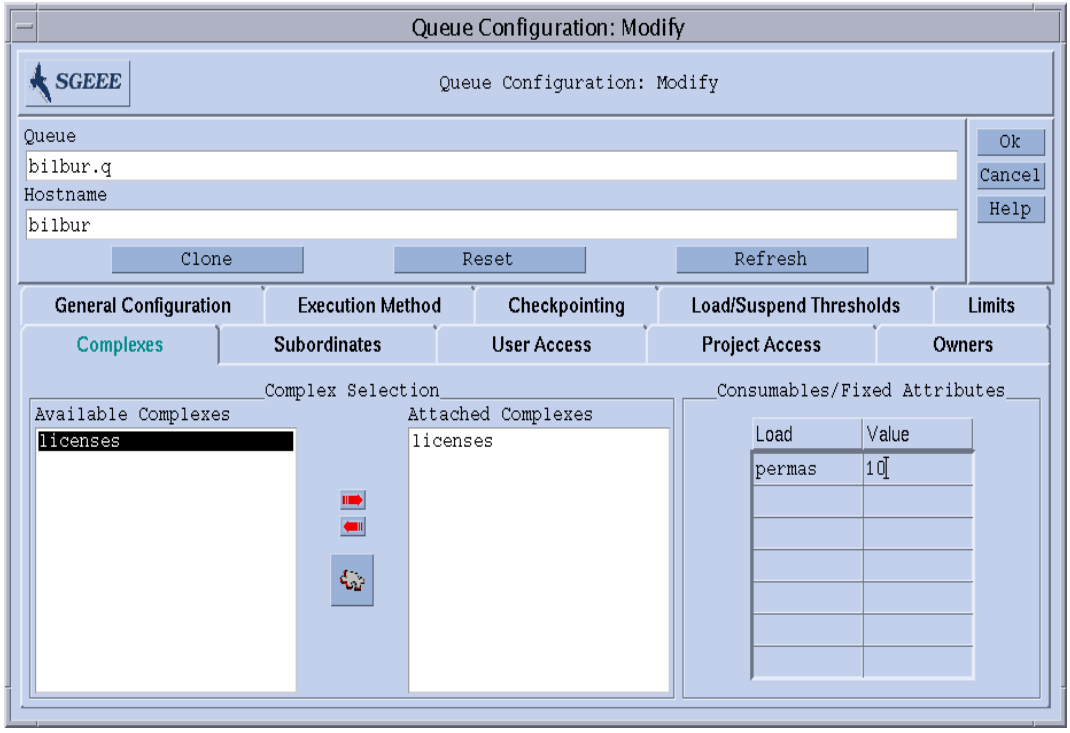


図 8-6　　キュー構成 - ユーザー定義の複合

表示されたキューが、ソフトウェアパッケージ `permas` のライセンスを 10 個まで管理するように設定します。これで、`licenses` 複合属性の `parms` が、Sun Grid Engine, Enterprise Edition ジョブで要求可能になり、図 8-7 に示すように、「実行依頼」ダイアログボックスの「要求資源」サブダイアログボックスの「使用可能な資源」リストに現れます (ジョブの実行依頼方法についての詳細は、第 4 章、71 ページの「ジョブの実行依頼」を参照)。

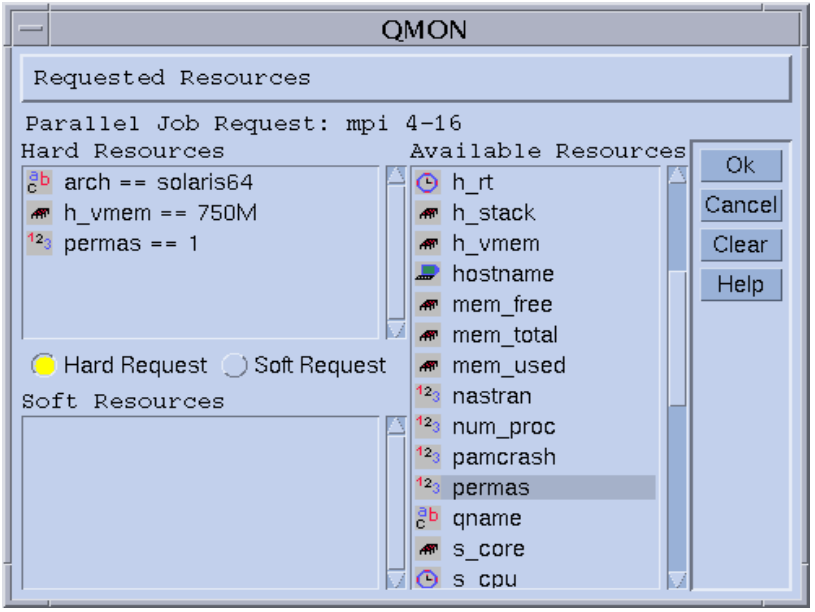


図 8-7　　実行依頼の「要求資源」サブダイアログボックス

次のようにコマンド行からジョブの実行依頼をして、`licenses` 属性を要求することもできます。

```
% qsub -l pe=1 permas.sh
```

注 – `permas` という完全な属性名のかわりに、`pm` というショートカットを使用することができます。

こうした構成を行って、類似のジョブ要求をした場合、そのジョブの実行資格があるキューは、ユーザー定義の `licenses` 複合に関連付けられているキューだけになります。すなわち、`permas` ライセンスが設定され、そのライセンスを使用可能なキューです。

不正なユーザー定義の複合名

以下は予約されている複合名で、ユーザー定義の複合名として使用することはできません。

- `global`

- host
- queue

消費可能資源

消費資源とも呼ばれる消費可能資源は、使用可能なメモリー、ファイルシステムの空き容量、ネットワーク帯域幅、包括的ソフトウェアライセンスなどの限られた資源を管理する効率的な手段です。消費可能資源の総合能力は Sun Grid Engine, Enterprise Edition 管理者が定義し、Sun Grid Engine, Enterprise Edition 内部のブックキーピングによって対応する資源の消費量が監視されます。Sun Grid Engine, Enterprise Edition は、実行中のすべてのジョブについて消費可能資源の消費量を把握し、使用可能な消費可能資源が十分にあることが、その内部ブックキーピングによって明らかである場合にのみジョブがディスパッチされるようにします。

消費可能資源はデフォルトまたはユーザー定義の負荷パラメータと結合することができます (214 ページの「負荷パラメータ」を参照)。すなわち、消費可能資源に関する負荷値を報告させたり、その逆に負荷属性に消費可能フラグを設定したりすることができます。その場合、Sun Grid Engine, Enterprise Edition の消費可能資源の管理機能は、負荷 (資源の可用性の測定によって得られる) と内部ブックキーピングの両方を考慮して、どちらも指定されている制限を超えないようにします。

消費可能資源の管理を有効にするには、資源の総合能力を定義する必要があります。この定義は、クラスタ全体のグローバルレベル、ホストレベル、キューレベルで行うことができ、これらのカテゴリは、列挙したのとは逆の順序で別のカテゴリに優先することができます。すなわち、ホストはクラスタ資源を制限することができ、キューはホストおよびクラスタ資源を制限することができます。資源の能力の定義は、キューおよびホスト構成の `complex_values` エントリを使用して行います。詳細は、このマニュアルの 169 ページの「キューの構成」、148 ページの「ホストの構成」のほか、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `host_conf` および `queue_conf` の項を参照してください。`global` ホストの `complex_values` の定義では、クラスタ全体のグローバル消費可能資源の設定をします。`complex_values` リスト内の消費可能資源の複合属性ごとに、その資源の最大使用可能量を表す値を割り当てます。内部ブックキーピングは、この全体量から、ジョブの資源要求から推定される、実行中のすべてのジョブの資源消費量を差し引きます。

▼ 消費可能資源を構成する

消費可能資源として構成できるのは、数値型の複合属性 (`INT`、`MEMORY`、`TIME` 型の属性) だけです。

1. `QMON` **メインメニューで「複合構成」ボタンをクリックします。**

   図 8-1 に示すような「複合構成」ダイアログボックスが表示されます。

2. 属性に対して Sun Grid Engine, Enterprise Edition の消費資源の管理を有効にするには、図 8-8 の `virtual_free` メモリー資源の例で示しているように複合構成で属性の消費可能フラグをセットします。

3. 次の節で説明している例に従って他の消費可能資源を構成します。
   - 203 ページの「例 1: 包括的ソフトウェアライセンスの管理」
   - 207 ページの「例 2: 仮想メモリーのスペースシェアリング」
   - 210 ページの「例 3: 使用可能なディスク容量の管理」

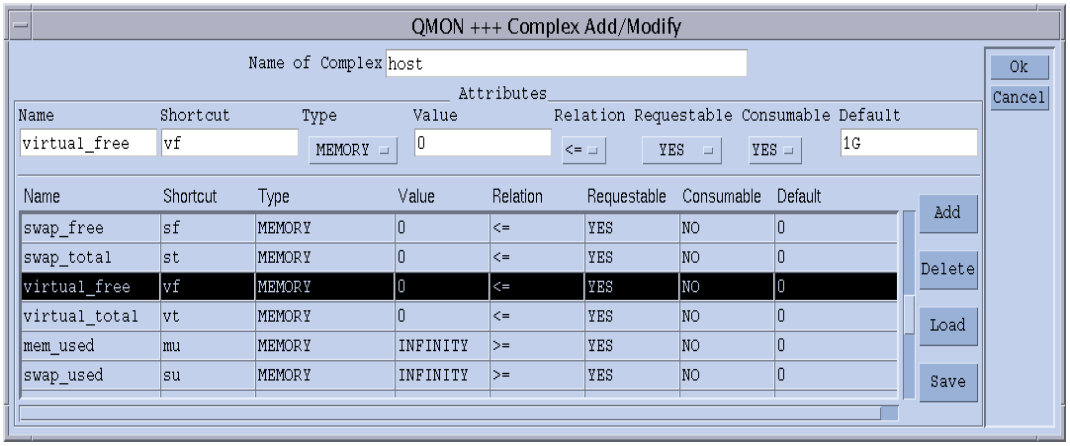


図 8-8 「複合構成」ダイアログボックス - `virtual_free`

構成を終えたら、Sun Grid Engine, Enterprise Edition に必要な能力利用計画を立てさせようとするキューまたはホストごとに、`complex_value` リストで能力を定義します。図 8-9 はその一例で、現在のホストの能力値として 1G バイトの仮想メモリーを定義しています。

こうして、このホストで現在使用可能な仮想メモリーは、ホスト (この場合、そのホストのどのキューであるかは問題ではない) で並行して実行中のすべてのジョブの仮想メモリー要求値を集計し、その集計値を 1G バイトの能力から差し引くことによって求められます。`virtual_free` に対するジョブの要求が使用可能な量を超える場合、そのジョブはそのホストのキューにディスパッチされません。

注 – 要求可能パラメータの強制値を使用して、ジョブに強制的に資源要求させ、推定消費量を指定することができます (図 8-8 を参照)。

注 – Sun Grid Engine, Enterprise Edition の管理者は、ジョブが消費可能資源属性を明示的に要求していない場合にデフォルトで使用する資源消費量を事前に定義しておくことができます (たとえば図 8-8 の例では、デフォルトを 200M バイトに設定)。これは、前述したように、属性の要求が行われていない場合にのみ意味があります。

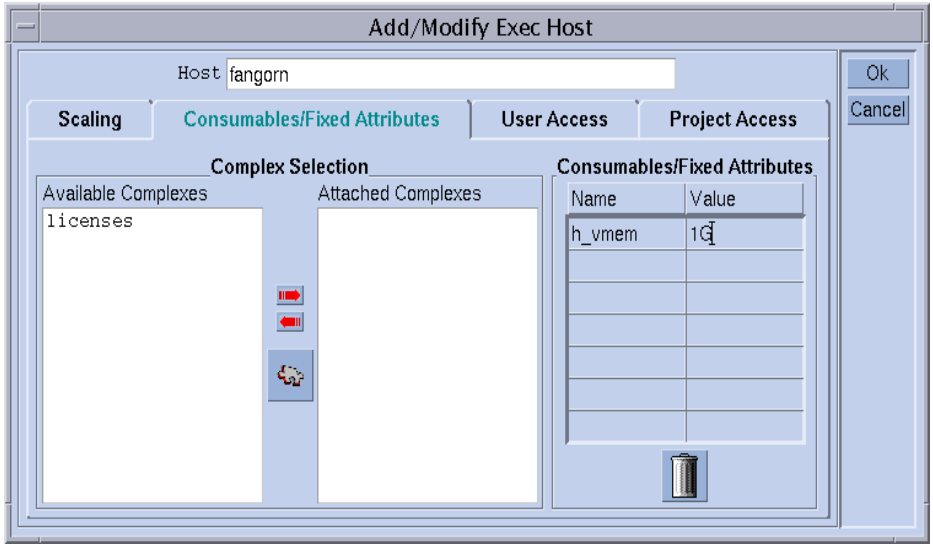


図 8-9　実行ホスト構成 - `virtual_free`

消費可能資源の構成例

ここでは、サイトでの消費可能資源の構成例を紹介します。

例 *1:* 包括的ソフトウェアライセンスの管理

クラスタで pam-crash というソフトウェアパッケージを使用していて、その包括ライセンス数が 10 であると仮定します。すなわち、アクティブな個数の合計が 10 を超えない限り、あらゆるシステムで `pam-crash` を使用することができます。目標は、実行中の pam-crash ジョブによって 10 個のライセンスがすべて占有されない限り、pam-crash ジョブのスケジューリングが妨げられないように Sun Grid Engine, Enterprise Edition を構成するということです。

この目標は、Sun Grid Engine, Enterprise Edition の消費可能資源を使用して簡単に実現することができます。最初に行う必要があるのは、図 8-10 に示すように、使用可能な pam-crash ライセンス数を消費可能資源としてグローバル複合構成に追加することです。

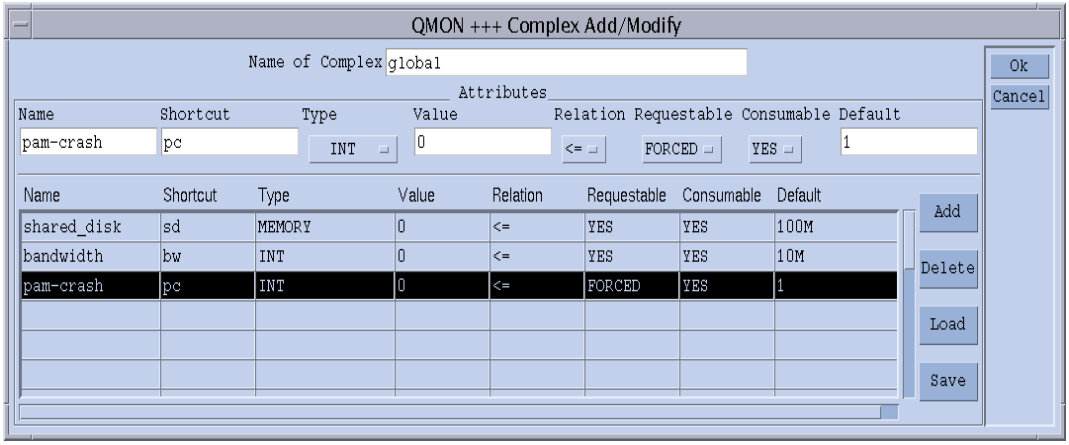


図 8-10 「複合構成」ダイアログ - pam-crash

この消費可能資源に pam-crash という名前を付け、qalter、qselect、qsh、qstat、qsub の -l オプションでは、そのショートカットとして pc を使用することができます。属性の型は、整数カウンタと定義します。消費可能資源は、complex_values リストを使用してグローバルかホスト、キュー構成からその値を得るため、「値」フィールドの設定は関係ありません。要求可能フラグは強制 (FORCED) に設定して、ジョブを実行依頼する際にそのジョブが占有する pam-crash ライセンス数を指定する必要があることを示します。消費可能フラグは、この属性を消費可能資源と定義しますが、「**デフォルト**」の設定は、「**要求可能**」が**強制**に設定されているため、関係ありません。つまり、どのジョブについても、この属性の要求値が受け取られます。

この属性およびクラスタの資源利用計画をアクティブにするには、図 8-11 に示すように、グローバルホスト構成で使用可能な pam-crash ライセンス数を定義する必要があります。包括ライセンス数は 10 ですから、属性 pam-crash の値を 10 に設定します。

注 – 「**消費可能 / 固定属性**」表は、ホスト構成のファイル形式 (host_conf) で説明している complex_values エントリに対応しています。

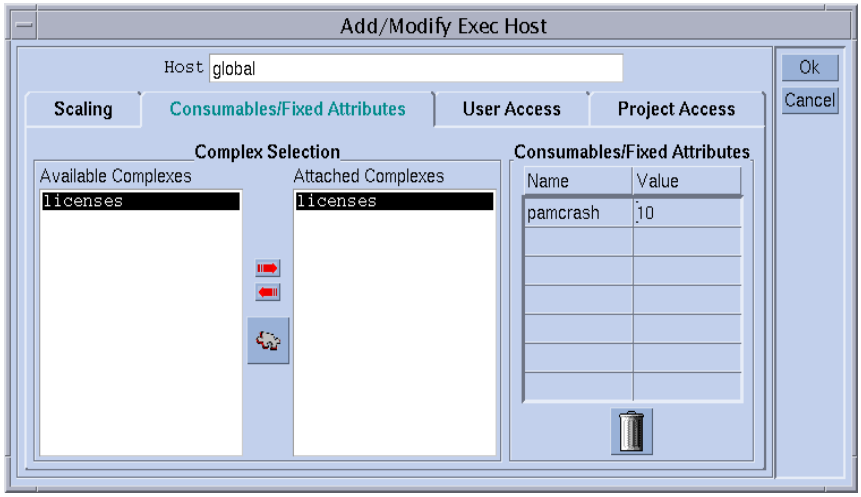


図 8-11　グローバルホスト構成 - pam-crash

ユーザーから次のジョブの実行依頼があったと仮定します。

```
% qsub -l pe=1 permas.sh
```

このジョブは、そのときに占有されている pam-crash ライセンス数が 10 未満の場合にのみ開始されます。ただし、このジョブはクラスタ内の任意の場所で実行することができ、その実行時間を通して pam-crash ライセンスを 1 つ占有します。

pam-crash バイナリがないなどの理由でクラスタ内のホストを包括ライセンス対象にできない場合は、消費可能属性 pam-crash に関するそのホストの能力を 0 に設定することによって、pam-crash ライセンスの管理対象からそのホストを除外することができます。この設定は、図 8-12 に示すようにそのホストの「実行ホスト構成」ダイアログボックスで行う必要があります。

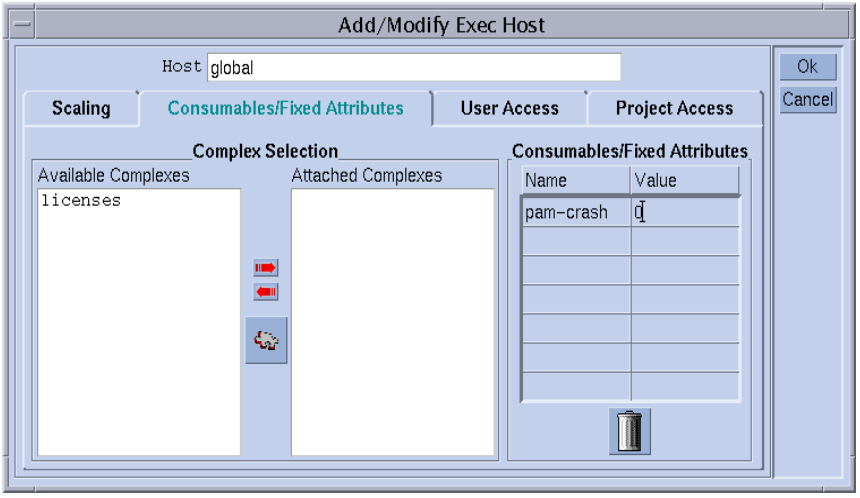


図 8-12　実行ホスト構成 - `pam-crash`

注 – `global` 複合のすべての属性はあらゆる実行ホストに継承されるため、実行ホストは暗黙で `pam-crash` 属性を利用できます。同様に能力を 0 に設定することによって、クラスタの全ライセンスの一部として特定のホストが管理するライセンス数を、2 などのゼロ以外の値に制限することもできます。この場合、そのホストに共存できる `pam-crash` ジョブ数は最大で 2 つです。

同様に、たとえばメモリーや CPU 時間制限が `pam-crash` に適していないキューが `pam-crash` ジョブを実行しないように設定することもできます。この場合は、図 8-13 に示すようにキュー構成で対応する能力を 0 に設定すればよいだけです。

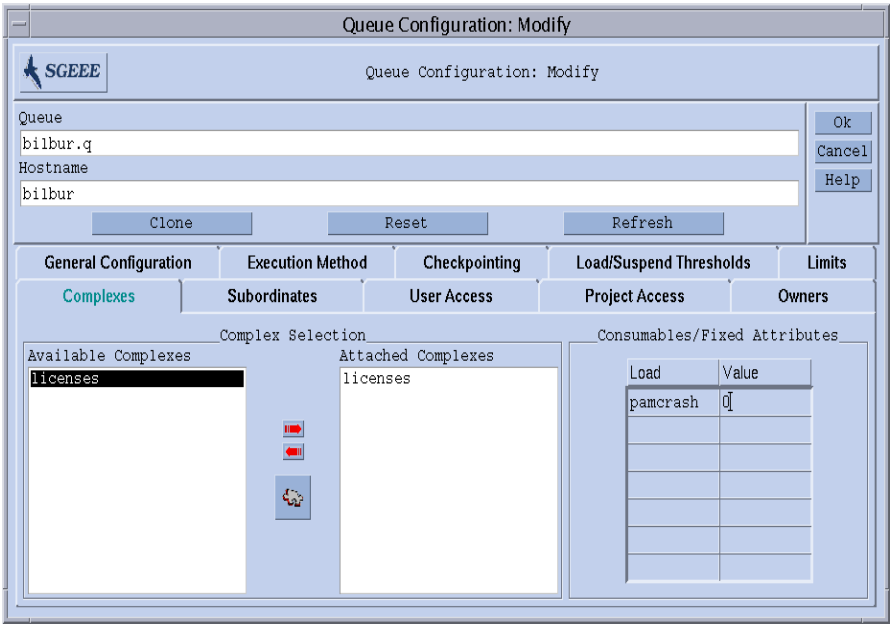


図 8-13　キュー構成 - pam-crash

注 – global 複合のすべての属性はあらゆるキューに継承されるため、キューは暗黙で pam-crash 属性を利用できます。

例 *2:* 仮想メモリーのスペースシェアリング

メモリーの過剰予約を原因とするパフォーマンスの低下、そしてその結果としてマシンでスワップが起きないようにシステムをチューニングすることは、システム管理者がよく行う仕事です。Sun Grid Engine, Enterprise Edition ソフトウェアは、消費可能資源機能でこの仕事の支援をします。

標準の負荷パラメータ virtual_free は、使用可能な仮想メモリー、すなわち、使用可能なスワップ空間と使用可能な物理メモリーを組み合わせた値を報告します。スワップを回避するには、スワップ空間の使用を極力抑える必要があります。理想は、ホストで動作する各プロセスの要求するすべてのメモリーが物理メモリーに収まるようにすることです。

Sun Grid Engine, Enterprise Edition ソフトウェアでは、以下のことを前提に、開始されるすべてのジョブに対してこのことが保証されるようにすることができます。

- virtual_free が消費可能資源として設定され、各ホストのその能力が使用可能な物理メモリー以下に設定されている。
- ジョブでメモリー使用の予測量が指定され、実行中に要求値を超えない。

図 8-8 の考えられるホスト複合の構成例と、その構成に対応する、1G バイトの主メモリーを搭載したホストの実行ホスト構成例を参照してください。

注 – 直前のグローバル複合構成の例では、**要求可能**フラグを**強制**に設定したのに対し、このホスト構成例では、要求可能フラグを YES (はい) に設定しています。このことは、ジョブでメモリー要求を指定する必要はないが、明示的なメモリー要求がない場合は、「**デフォルト**」フィールドの値が使用されることを意味します。この場合のデフォルト要求の 1G バイトという値は、要求のないジョブは使用可能なすべて物理メモリーを占有すると見なされることを意味します。

注 – virtual_free は、Sun Grid Engine, Enterprise Edition 標準の負荷パラメータの 1 つです。仮想メモリーの能力利用計画では、Sun Grid Engine, Enterprise Edition は使用可能なメモリーに関する最新の統計を自動的に考慮します。空き仮想メモリーの負荷報告値が Sun Grid Engine, Enterprise Edition の内部ブックキーピングで得られた値を下回っている場合は、その負荷値を使用して過剰なメモリー予約が回避されます。負荷報告値と内部ブックキーピング値の違いは、Sun Grid Engine, Enterprise Edition を使用しないでジョブが開始された場合によく発生することがあります。

---

1 つのマシンでメモリー要求が異なるさまざまなクラスのジョブを実行する場合は、それらのジョブクラス全体で使用できるようそのマシンのメモリーを分割した方がよいことがあります。「スペースシェアリング」とも呼ばれるこの機能は、それぞれのジョブクラスにキューを設定し、そのキューにホストのメモリーの一部を割り当てることによって実現することができます。

図 8-14 に示す、この例のキュー構成では、ホスト bilbur で使用可能なメモリー全体の半分、500M バイトをキュー bilbur.q に関連付けていますこれで、キュー billbur.q で実行されるすべてのジョブの累積メモリー消費量が 500M バイトを超えることができなくなります。他のキューのジョブは考慮されませんが、ホスト bilbur で実行されるすべてのジョブの総メモリー消費量が 1G バイトを超えることもできません。

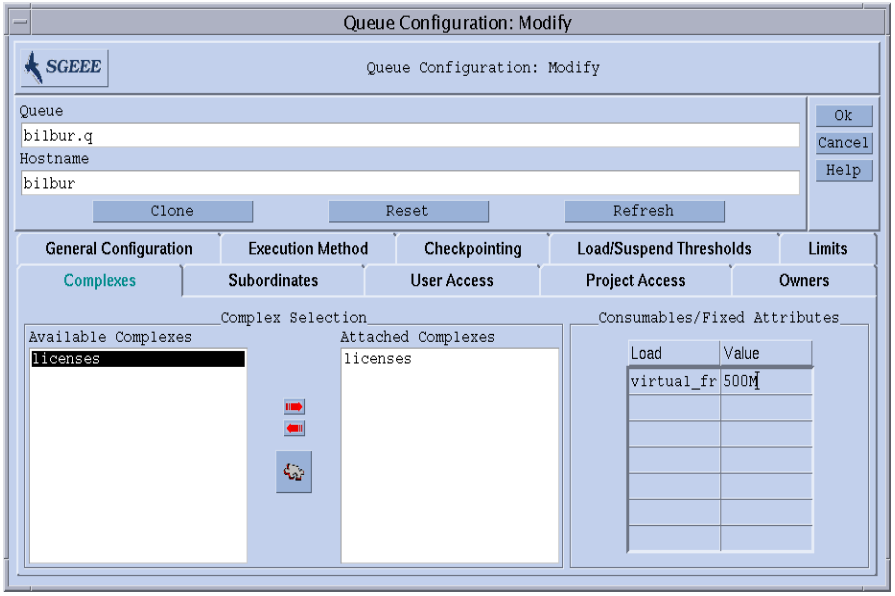


図 8-14　キュー構成 - `virtual_free`

注 – `virtual_free` 属性はホスト複合から継承されるため、すべてのキューで利用することができます。

次のいずれかの形式で、上記の例に似た構成のシステムにジョブの実行依頼をしたと仮定します。

```
% qsub -l vf=100M honest.sh
% qsub dont_care.sh
```

最初のコマンドで実行依頼されたジョブは、少なくとも 100M バイトのメモリーが使用可能である限りただちに開始され、その使用量が、`virtual_free` 消費可能資源の能力利用計画で考慮されます。2 つ目のジョブは、暗黙で使用可能なすべてのメモリーを要求しているため、他のジョブがシステムに存在しない場合にのみ実行されます。また、キューのメモリー能力を超えるため、`bilbur.q` キューで実行することはできません。

例 *3:* 使用可能なディスク容量の管理

アプリケーションには、ファイルに格納された巨大なデータセットの操作を必要とするものがあり、そうしたアプリケーションでは、実行中ずっと十分なディスク容量を利用できる必要があります。この条件は、前述の例で説明した使用可能なメモリーのスペースシェアリングに似ています。大きな違いは、Sun Grid Engine, Enterprise Edition 標準の負荷パラメータとして、空きディスク容量が存在しないことです。これは、通常、ディスクはサイトに固有の方法でパーティションに分割されて、使用されているファイルシステムが異なり、操作するファイルシステムを自動的に識別できないためです。

しかし、使用可能なディスク容量は、Sun Grid Engine, Enterprise Edition の消費可能資源機能を使用して効率的に管理することができます。このためには、この節で後ほど挙げる理由からホスト複合属性の `h_fsize` を使用することを推奨します。最初に、たとえば図 8-15 に示すように、属性を消費可能資源として設定する必要があります。

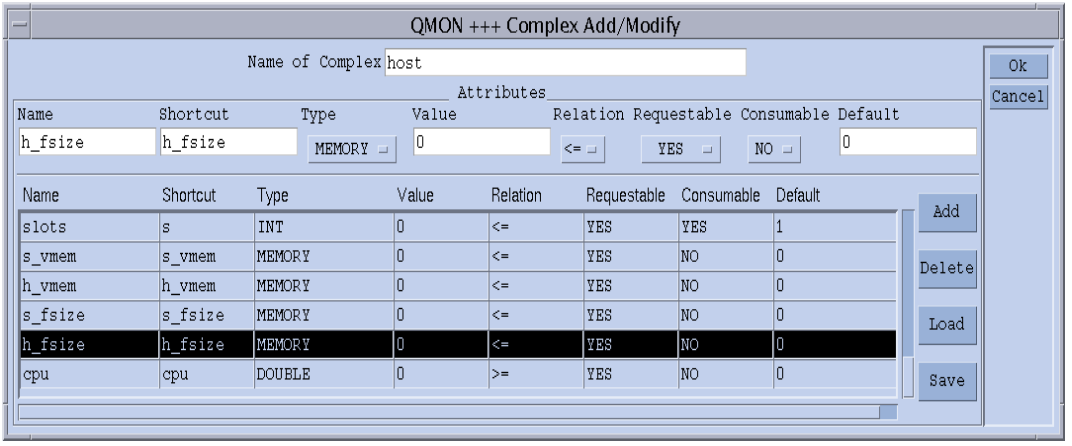


図 8-15　　複合構成 ―`h_fsize`

ファイルシステムがホストにローカルなシステムであると仮定すると、図 8-16 に示すようにディスク容量消費可能資源の能力の定義をホスト構成に追加します。

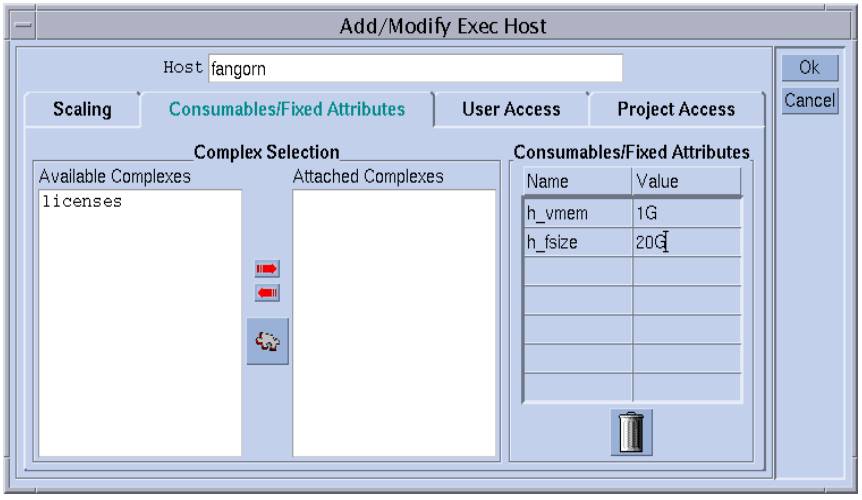


図 8-16 実行ホスト構成 - `h_fsize`

このように構成された Sun Grid Engine, Enterprise Edition システムへのジョブの実行依頼は、前述の例に似た仕組みで動作します。

```
% qsub -l hf=5G big_sort.sh
```

この例で `h_fsize` 属性を推奨する理由は、`h_fsize` がキュー構成のハードファイル制限としても使用されることにあります。ファイルサイズ制限は、ジョブの実行依頼で指定されたサイズ (この例では 20G バイト)、あるいはジョブがこの属性を要求していない場合は、キュー構成内の対応する値より大きなファイルを作成するジョブの権限を制限します、この例では、`h_fsize` の**要求可能**フラグを強制 (FORCED) に設定しているため、要求はつねに存在することになります。

消費可能資源としてキュー制限を使用することによって、ジョブスクリプトによる実際の資源消費量に基づいてユーザーが指定した要求を自動的に制御することができます。この制限に対する違反は制裁され、最終的にジョブは実行中止されます (詳細は、`queue_conf` と `setrlimit` のマニュアルページを参照)。このようにして、Sun Grid Engine, Enterprise Edition 内部の能力利用計画に基づく資源要求を信頼性の高いものにすることができます。

注 – オペレーティングシステムには、プロセス単位のファイルサイズ制限しかサポートしていないものがあります。その場合、ジョブは制限までのサイズのファイルを複数作成することができます。これに対し、ジョブ単位のファイルサイズ制限をサポートしているシステムでは、Sun Grid Engine, Enterprise Edition は `h_fsize` 属性でこの機能を利用します (詳細は、`queue_conf` のマニュアルページを参照)。

Sun Grid Engine, Enterprise Edition に実行依頼されないアプリケーションが並行してディスク領域を使用すると予想される場合、ディスク容量の不足を原因とする問題がアプリケーションで発生するのを防ぐには、Sun Grid Engine, Enterprise Edition 内部のブックキーピングでは不十分かもしれません。この問題を回避するには、定期的な方法でディスク容量使用統計を得ると良いかもしれません。そうした統計によって、Sun Grid Engine, Enterprise Edition の外部で発生しているものも含めて、全体のディスク領域の消費量が分かります。

Sun Grid Engine, Enterprise Edition の負荷センサーインタフェース (214 ページの「サイトに固有の負荷パラメータの追加」を参照) では、特定のファイルシステムで使用可能なディスク容量などのサイトに固有の情報で、Sun Grid Engine, Enterprise Edition 標準の負荷パラメータの機能を強化することができます。

`h_fsize` 用の適切な負荷センサー追加し、空きディスク容量を報告させることによって、消費可能資源の管理と資源の可用性統計を組み合わせることができます。Sun Grid Engine, Enterprise Edition は、ディスク容量に関するジョブの要求と、自身の内部の資源利用計画から得られる使用可能能力および報告された最新の負荷値とを比較し、両方の条件が満たされた場合にのみジョブをホストにディスパッチします。

複合の構成

Sun Grid Engine, Enterprise Edition の複合の定義および管理は、192 ページの「複合構成を追加または変更する」の節で説明しているように「QMON 複合構成」ダイアログボックスを使用してグラフィカルに行うことも、コマンド行から行うこともできます。

▼ コマンド行から複合構成を変更する

適切なオプションを付けて次のコマンドを入力します。

```
% qconf options
```

qconf のコマンド形式および有効な値フィールドの構文についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の complex の項または complex のマニュアルページを参照してください。

便利なオプションとしては以下があります。

- -ac
- -mc
- -Ac
- -Mc

qconf の -Ac および -Mc オプションは引数として複合構成ファイルをとるのに対し、-ac および -mc オプションは、エディタで複合構成用のテンプレートまたは既存の複合の構成を開きます。

これらのオプションの意味は次のとおりです。

- qconf -Ac または -ac

  使用可能な複合のリストに新しい複合を追加します。

- qconf -Mc または -mc

  既存の複合を変更します。

qconf コマンドの使用例

次のコマンドは、

```
% qconf -sc licenses
```

complex(5) のマニュアルページに定義されているファイル形式で標準出力ストリームに nastran 複合 (図 8-5 を参照) を出力します。表 8-1 は、licenses 複合の出力例を示しています。

```
#name        shortcut    type     value    relop    requestable    consumable    default
#-------------------------------------------------------------------------------------
nastran      na          INT      10       <=       YES            NO            0
pam-crash    pc          INT      15       <=       YES            YES           1
permas       pm          INT      40       <=       FORCED         YES           1
#---- # start a comment but comments are not saved across edits ----------------------
```

表 8-1 qconf -sc の出力例

負荷パラメータ

この節では、Sun Grid Engine, Enterprise Edition 5.3 の負荷パラメータの概念と独自の負荷センターの作成方法を説明します。

デフォルトの負荷パラメータ

デフォルトでは、sge_execd はいくつかの負荷パラメータとその値を定期的に sge_qmaster に報告し、その報告は sge_qmaster の内部ホストオブジェクトに格納されます (147 ページの「デーモンとホスト」の節を参照)。しかし、そうした報告は、対応する名前を持つ複合属性が定義されている場合に内部的に使用されるだけです。そうした複合属性には、負荷値の解釈方法に関する定義が含まれています (詳細は、193 ページの「複合の種類」の節を参照)。

Sun Grid Engine, Enterprise Edition の基本インストールを終えると、標準の負荷パラメータセットが報告されます。標準の負荷パラメータに必要なすべての属性は、ホスト複合に定義されています。以降の Sun Grid Engine, Enterprise Edition のリリースでは、デフォルトの負荷パラメータの拡張セットが提供されることが考えられます。このため、デフォルトで報告される負荷パラメータセットについては、*<sge_root>*/doc/load_parameters.asc ファイルで説明をしています。

注 – 負荷属性を利用可能かどうかは、その負荷属性が定義されている複合で決まります。グローバル複合で負荷属性を定義すると、クラスタ全体およびあらゆるホストでその属性を利用できるようになります。ホスト複合で定義した場合は、あらゆるホストにその属性が提供されますが、クラスタ全体にグローバルには提供されません。ユーザー定義の複合で定義すると、そのユーザー複合をホストに関連付けたり、関連付け解除したりすることによって負荷属性の表示を制御することができます。

注 – キュー複合で負荷属性を定義しないでください。ホストおよびクラスタのどちらからも利用できなくなります。

サイトに固有の負荷パラメータの追加

デフォルトの負荷パラメータセットは、特にサイトに固有のポリシーやアプリケーション、構成の観点からすると、クラスタの負荷状況を完全に表すのに十分ではないかもしれません。このため、Sun Grid Engine, Enterprise Edition ソフトウェアには、任意の方法で負荷パラメータセットを拡張する手段が用意されています。sge_execd は、現在の負荷値とともに負荷パラメータを sge_execd に供給するイ

ンタフェースが用意されています。供給されたパラメータはデフォルトの負荷パラメータがまったく同様に扱われます。定義した負荷パラメータを有効にするには、デフォルト同様、負荷複合で対応する属性を定義する必要があります (214 ページの「デフォルトの負荷パラメータ」の節を参照)。

▼ 独自の負荷センサーを作成する

追加の負荷情報を `sge_execd` に供給するには、負荷センサーを用意する必要があります。この負荷センサーはスクリプトでも、バイナリ形式の実行可能ファイルでもかまいません。どちらの場合も、その標準入出力ストリームの処理および制御の流れは次の規則に従っている必要があります。

負荷センサーは、特定の地点で `STDIN` からの入力を待つ無限ループとして作成する必要があります。`STDIN` から文字列 `quit` を読み取ったら、負荷センサーを終了します。`STDIN` から行の終わりを読み取ったら、ただちに負荷データの読み出しサイクルを開始します。このサイクルでは、負荷センサーは目的の負荷値の計算に必要なあらゆる処理を行い、サイクルの終わりで結果を `stdout` に書き込みます。

規則

負荷センサーの形式は次のとおりです。

- 負荷値レポートは、`begin` という文字列だけを含む行で開始します。
- 負荷値はそれぞれ改行で区切ります。
- 1 つの負荷値は、空白なしのコロン (`:`) で区切られた 3 つの部分で構成します。
- 負荷値の最初の部分は、その負荷の情報元のホスト名か特殊名 `global` です。
- 2 つ目の部分は、ホストまたはグローバル構成リストに定義されている負荷値のシンボリック名です (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `complex(5)` の項を参照)。ホストまたはグローバル複合リストにエントリのない負荷値が報告された場合、その負荷値は使用されません。
- 3 つ目の部分は負荷測定値です。
- 負荷値レポートは、`end` という文字列の行で終了します。

スクリプト例

コード例 8-1 は、Bourn シェルスクリプトの負荷センサーの例です。

```
#!/bin/sh
myhost=`uname -n`
while [ 1 ]; do
   # wait for input
   read input
   result=$?
   if [ $result != 0 ]; then
      exit 1
   fi
   if [ $input = quit ]; then
      exit 0
   fi
   #send users logged in
   logins=`who | cut -f1 -d" " | sort | uniq | wc -l` | sed "s/^ *//"
   echo begin
   echo "$myhost:logins:$logins"
   echo end
done
# we never get here
exit 0
```

コード例 8-1　　Bourne シェルスクリプトの負荷センサー

このコード例を `load.sh` というファイル名で保存し、`chmod` で実行可能権限を割り当てると、`load.sh` を起動し、キーボードの Return キーを繰り返し押すことによって、コマンド行から対話形式でテストを行うことができます。

テストがうまくいったら、クラスタ、グローバル、あるいは実行ホスト別の構成に `load_sensor` パラメータとして負荷センサーのパスを設定することによって、任意の実行ホスト用に負荷センサーを組み込むことができます (162 ページの「基本クラスタ構成」または `sge_conf` のマニュアルページを参照)。

対応する QMON 画面は図 8-17 の例のようになります。

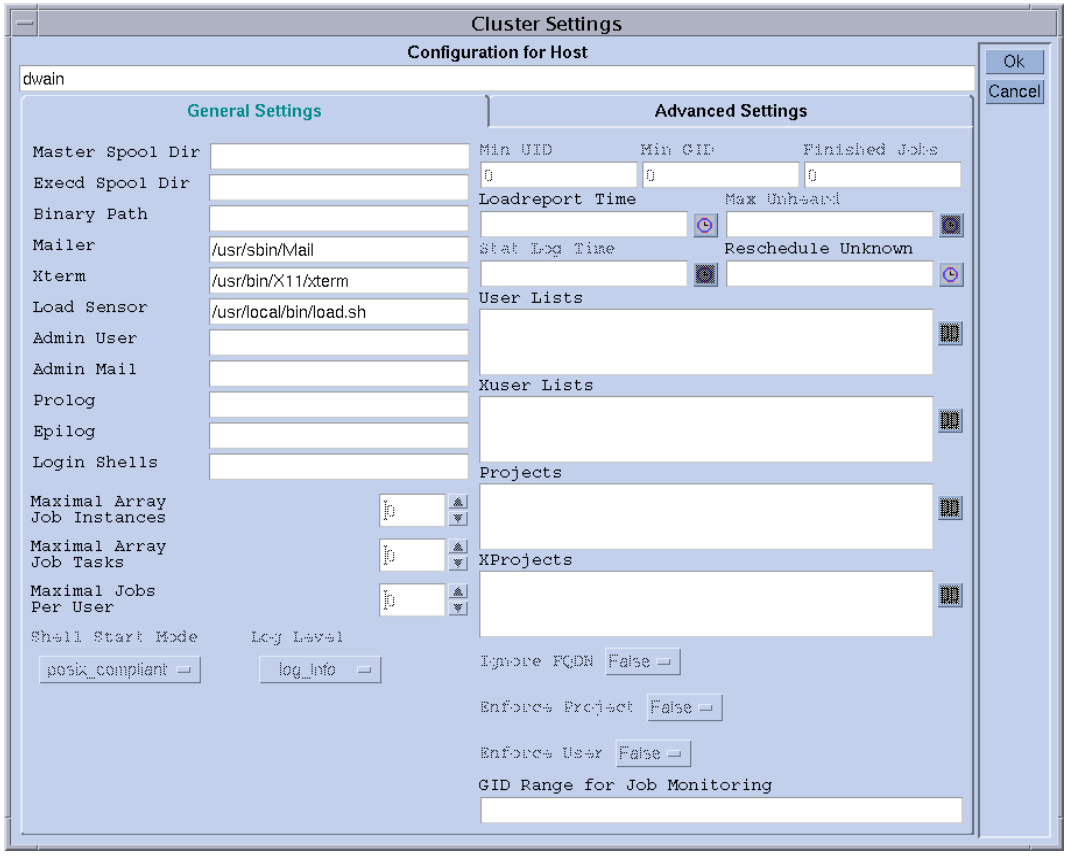


図 8-17　ローカル構成 - 負荷センサー

報告される負荷パラメータの logins は、対応する属性をホスト複合に追加するとすぐに使用できるようになります。また、必要な定義は、図 8-18 の「QMON 複合構成」ダイアログボックスの表の最後のエントリのようになります。

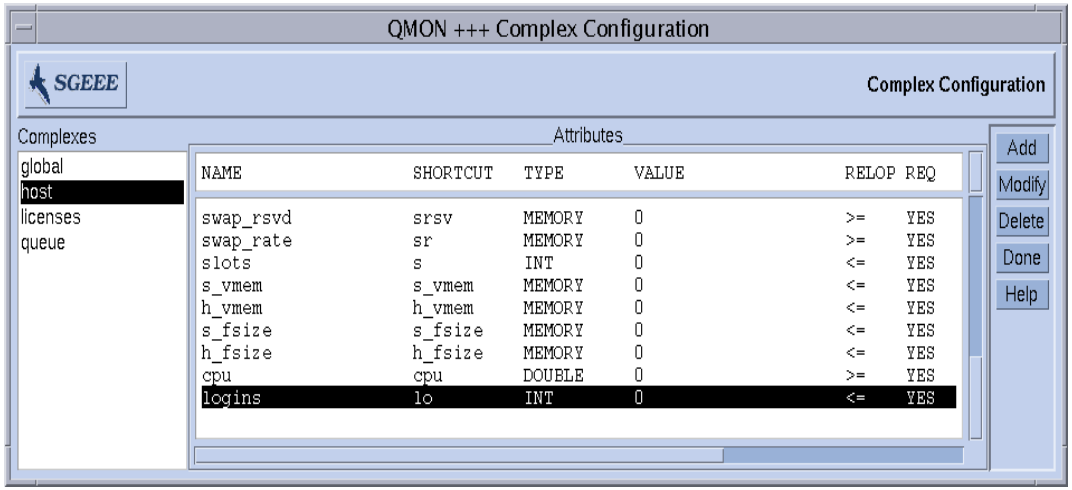


図 8-18　「複合構成」ダイアログボックス - logins

第9章

ユーザーアクセスとポリシーの管理

この章では、Sun Grid Engine, Enterprise Edition システムにおけるユーザーとそれに関係するアカウントとポリシーの管理に関する重要な情報を提供します。具体的には、ユーザーアクセスやプロジェクト、パスの別名設定、デフォルトの要求、アカウンティングと利用統計、チェックポイント機能のサポートなどです。

予備知識の他に、それらの作業を行う方法を詳細に説明します。

ユーザーの構成

ここでは、Sun Grid Engine, Enterprise Edition のユーザー構成で必要な作業と行うことができる作業をまとめています。

- 必要なログインアカウント

  ホスト *A* からホスト *B* でジョブを実行するように依頼するには、ユーザーはホスト *A* と *B* に同じアカウント (すなわち、同じユーザー名) を持っている必要があります。`sge_qmaster` が動作しているマシンにログインアカウントを持っている必要はありません。

- Sun Grid Engine, Enterprise Edition のアクセス権の設定

  Sun Grid Engine, Enterprise Edition ソフトウェアには、クラスタ全体やキュー、並列環境に対するユーザーアクセスを制限する機能が用意されています。詳細は、226 ページの「ユーザーのアクセス権」の節を参照してください。

  また、Sun Grid Engine, Enterprise Edition システムのユーザーは、特定のキューを一時停止または使用可能にする権限を持つことができます (183 ページの「所有者を設定する」を参照)。

- Sun Grid Engine, Enterprise Edition ユーザーの定義

  ユーザに対する基本割当ツリーにノードを含めるか、ユーザーに対する業務優先または一時優先ポリシーを定義する場合は (249 ページの「QMON からポリシー / チケットに基づく高度な資源管理を実施する」の節を参照)、Sun Grid Engine, Enterprise Edition システムにユーザーを定義する必要があります。詳細は、230 ページの「QMON からユーザーオブジェクトを構成する」を参照してください。

- Sun Grid Engine, Enterprise Edition プロジェクトへのアクセス

  Sun Grid Engine, Enterprise Edition プロジェクトを使用して、基本割当、業務優先、または一時優先ポリシーのいずれかを定義する場合は (249 ページの「QMON からポリシー / チケットに基づく高度な資源管理を実施する」の節を参照)、1 つ以上のプロジェクトに対するアクセス権をユーザーに付与する必要があります。このアクセス権がない場合、ユーザーのジョブは優先順位の最も低いクラスになり、資源へのアクセス権を受けるチャンスはほとんどなくなります。

- ファイルアクセス制限

  Sun Grid Engine, Enterprise Edition ユーザーは、<*sge_root*>`/cell/common` ディレクトリに対する読み取りアクセス権を持っている必要があります。

  Sun Grid Engine, Enterprise Edition の実行デーモン (*root* で動作) は、Sun Grid Engine, Enterprise Edition ジョブを開始する前にそのジョブ用に一時作業ディレクトリを作成し、そのディレクトリの所有権をジョブの所有者に移します。

この一時作業ディレクトリは、キュー構成パラメータ tmpdir に指定されたパスの下に作成され、 ジョブが終了するとただちに削除されます (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の queue_conf の項を参照)。

必ず、tmpdir の場所に下に一時ディレクトリを作成して、Sun Grid Engine, Enterprise Edition ユーザーの所有権を設定できるようし、ユーザーが後でその一時ディレクトリに書き込みを行えるようにしてください。

- サイト依存

  定義上、バッチジョブには端末接続はありません。このため、コマンドインタプリタの起動リソースファイル (csh に対する .cshrc など) に stty のような UNIX コマンドが含まれていると、エラーになることがあります。49 ページの「インストールの検証」の説明に従って、そのようなコマンドがないか調べ、使用されないようにしてください。

  通常 Sun Grid Engine, Enterprise Edition のバッチジョブはオフラインで実行されるため、エラーイベントなどをジョブの所有者に通知する方法は 2 つしかありません。1 つはファイルにエラーメッセージを記録する方法、もう 1 つは電子メールを送信する方法です。めったにはありませんが、エラーログファイルを開けないなどの理由で、電子メールがユーザーに直接通知する唯一の手段になることもあります (この場合でも、そうしたエラーメッセージは Sun Grid Engine, Enterprise Edition システムのログファイルに記録されますが、通常、ユーザーはシステムログファイルの内容を見ません)。このため、Sun Grid Engine, Enterprise Edition のユーザーのために電子メールシステムを正しくインストールしておくことを推奨します。

- Sun Grid Engine, Enterprise Edition の定義ファイル

  Sun Grid Engine, Enterprise Edition ユーザー用に次の定義ファイルを作成することができます。

  - qmon - Sun Grid Engine, Enterprise Edition GUI 用のリソースファイル。14 ページの「QMON のカスタマイズ」の節を参照。
  - sge_aliases - 現在の作業ディレクトリのパスの別名。276 ページの「パスの別名設定」の節を参照。
  - sge_request - デフォルトの要求定義ファイル。278 ページの「デフォルト要求の構成」の節を参照。

ユーザーカテゴリ

Sun Grid Engine, Enterprise Edition システムには、4 つのユーザーカテゴリがあります。

- **マネージャー** - Sun Grid Engine, Enterprise Edition の運用に関する全権を持つユーザーです。デフォルトでは、マスターホストとキューのホストとなるマシンのスーパーユーザーがマネージャー特権を持ちます。
- **オペレータ** - キューの追加や削除、変更ができないことを除けば、マネージャーが実行するコマンドの多くを実行できるユーザーです。
- **所有者** - キューの所有者のことで、所有するキューの一時停止 / 停止解除、あるいは使用不可 / 使用可能操作だけを行えるユーザーです。`qidle` を使用するには、これらの特権が必要です。一般にユーザーは、使用しているデスクトップワークステーション上のキューの所有者として定義されます。
- **ユーザー** - ユーザーは 226 ページの「ユーザーのアクセス権」で説明しているようないくつかのアクセス権を持ちますが、クラスタやキューの管理を行うことはできません。

これらのカテゴリについては、以降の節でさらに詳しい説明があります。

▼ QMON からアカウントを構成する

1. **`QMON` のメインメニューで「ユーザー構成」ボタンをクリックします。**
2. **行おうとする作業に従って適切なタブセレクタをクリックします。**
   - マネージャーアカウント構成 (図 9-1 を参照)。
   - オペレータアカウント構成 (図 9-2 を参照)。
   - ユーザーセットアクセス / 部署リスト構成 (図 9-3 を参照)。
   - ユーザー構成 (図 9-5 を参照)。
3. **以下の適切な節に進みます。**

注 – デフォルトでは、「ユーザー構成」ボタンを初めてクリックすると、「マネージャーアカウント構成」ダイアログボックスが開きます。

▼ QMON からマネージャーアカウントを構成する

「マネージャー」タブを選択すると、「マネージャー構成」ダイアログボックスが表示され (図 9-1 を参照)、このダイアログボックスから、あらゆる Sun Grid Engine, Enterprise Edition 管理コマンドを実行する権限を付与するアカウントを定義することができます。画面の下半分は選択リストで、すでに管理権限を持つことが定義されているアカウントが表示されます。

- **削除** - ダイアログボックスの選択リストで既存のアカウント名をクリックし、右側の「削除」ボタンをクリックすると、リストからそのマネージャーアカウントが削除されます。

- **追加** - 選択リストの上にある入力フィールドにアカウント名を入力して、「追加」ボタンをクリックするか、キーボードの Return キーを押すと、その名前のマネージャーアカウントが追加されます。

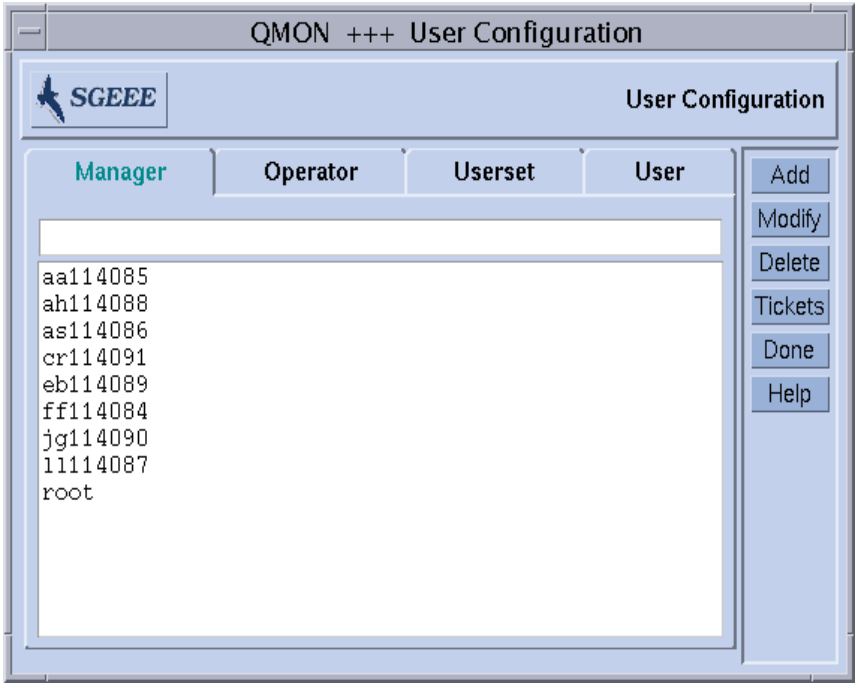


図 9-1　「マネージャー構成」ダイアログボックス

▼ コマンド行からマネージャーアカウントを構成する

- 適切なスイッチを付けて次のコマンドを入力します。

```
# qconf switches
```

使用可能なスイッチ

- `qconf -am` *user_name*`[,...]`

  マネージャーの追加 - Sun Grid Engine, Enterprise Edition のマネージャーリストに指定されたユーザー (複数指定可能) を追加します。デフォルトでは、Sun Grid Engine, Enterprise Edition によって信任 (トラスト) されたホストの root アカウントは、Sun Grid Engine, Enterprise Edition マネージャーになります。

- `qconf -dm` *user_name[,...]*

  マネージャーの削除 - Sun Grid Engine, Enterprise Edition のマネージャーリストから指定されたユーザー (複数指定可能) を削除します。

- `qconf -sm`

  マネージャーの表示 - Sun Grid Engine, Enterprise Edition のマネージャーリストを表示します。

▼ QMON からオペレータアカウントを構成する

「オペレータ」タブを選択すると、「オペレータ構成」ダイアログボックスが表示され (図 9-2 を参照)、このダイアログボックスから、限られた Sun Grid Engine, Enterprise Edition 管理コマンドを実行する権限を付与するアカウントを定義することができます (ただし、定義するアカウントは、マネージャーアカウントとして定義されていない必要があります。222 ページの「QMON からマネージャーアカウントを構成する」を参照)。画面の下半分は選択リストで、すでにオペレータ権限を持つことが定義されているアカウントが表示されます。

- **削除** - ダイアログボックスの選択リストで既存のアカウント名をクリックし、右側の「削除」ボタンをクリックすると、リストからそのオペレータアカウントが削除されます。
- **追加** - 選択リストの上にある入力フィールドにアカウント名を入力して、「追加」ボタンをクリックするか、キーボードの Return キーを押すと、その名前のオペレータアカウントが追加されます。

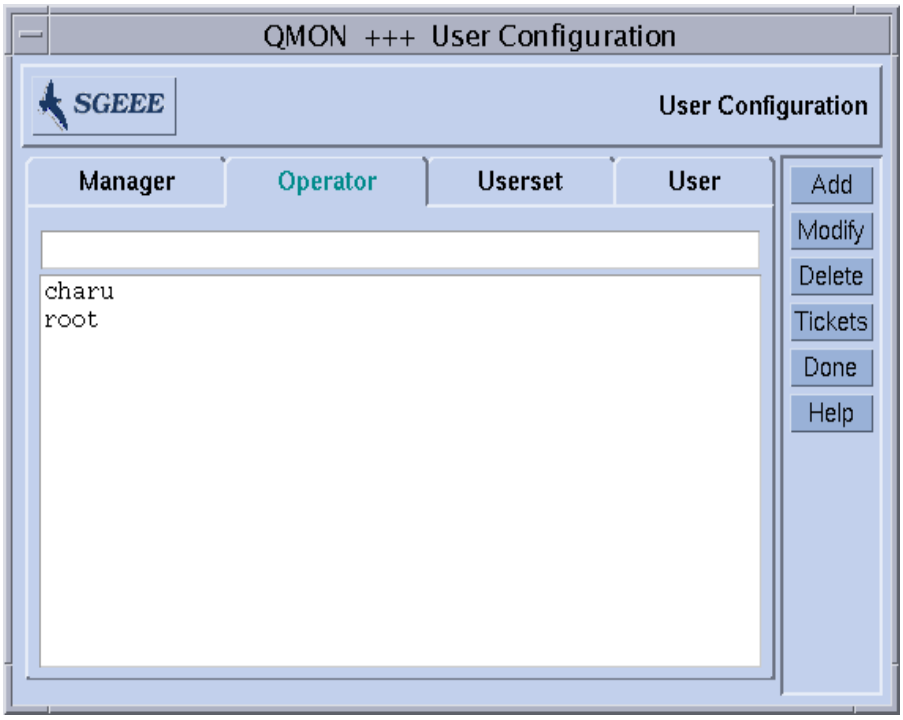


図 9-2　「オペレータ構成」ダイアログボックス

▼ コマンド行からオペレータアカウントを構成する

- 適切なスイッチを付けて次のコマンドを入力します。

```
# qconf switches
```

使用可能なスイッチ

- `qconf -ao` *user_name[,...]*

  オペレータの追加 - Sun Grid Engine, Enterprise Edition オペレータリストに指定されたユーザー (複数指定可能) を追加します。

- `qconf -do` *user_name[,...]*

  オペレータの削除 - Sun Grid Engine, Enterprise Edition オペレータリストから指定されたユーザー (複数指定可能) を削除します。

- `qconf -so`

  オペレータの表示 - Sun Grid Engine, Enterprise Edition オペレータリストを表示します。

キュー所有者のアカウント

キューの所有者は、Sun Grid Engine, Enterprise Edition キューの構成または変更中に定義されます。170 ページの「QMON からキューを構成する」と 184 ページの「コマンド行からキューを構成する」を参照してください。キューの所有者は以下のことを行うことができます。

- **一時停止** - キューで実行されているすべてのジョブの実行を停止して、キューを閉じます。
- **停止解除** - キューでの実行を再開して、キューを開きます。
- **使用不可** - キューを閉じます。ただし、実行中のジョブには影響しません。
- **使用可能** - キューを開きます。

注 – キューの一時停止中に明示的に一時停止されたジョブは、キューが停止解除されても実行再開されません。明示的に停止解除する必要があります。

一般に、ユーザーが大切な仕事のためにときどき特定のマシンを必要とする場合、あるいはユーザーがバックグラウンドで動作している Sun Grid Engine, Enterprise Edition ジョブの強い影響を受ける場合は、そのユーザーを特定のキューの所有者として定義します。

ユーザーのアクセス権

少なくとも 1 つの実行依頼ホストと実行ホストに正当なログインアカウントを持つユーザーは誰でも、Sun Grid Engine, Enterprise Edition システムを利用することができます。ただし、Sun Grid Engine, Enterprise Edition のマネージャーは、特定のユーザーが一部またはすべてのキューにアクセスするのを禁止することができます。特定の並列環境などの機能の利用を制限することもできます (289 ページの「並列環境」の節を参照)。

アクセス権を定義するには、ユーザーアクセスリストを定義する必要があります。ユーザーアクセスリストは、内容の重複が可能な名前付きのユーザーセットです。ユーザーアクセスリストの定義には、ユーザー名と UNIX グループ名を使用することができます。定義したユーザーアクセスリストは、クラスタ構成 (162 ページの「基本クラスタ構成」の節を参照) やキュー構成 (179 ページの「従属キューを設定する」の節を参照)、あるいは並列環境インタフェースの構成 (290 ページの「QMON から並列環境を構成する」の節を参照) で、特定の資源へのアクセスの拒否や許可の指定に使用されます。

▼ QMON からユーザーアクセスリストを構成する

「ユーザーセット」タブを選択すると、図 9-3 に示すような「ユーザーセット構成」ダイアログボックスが表示されます。

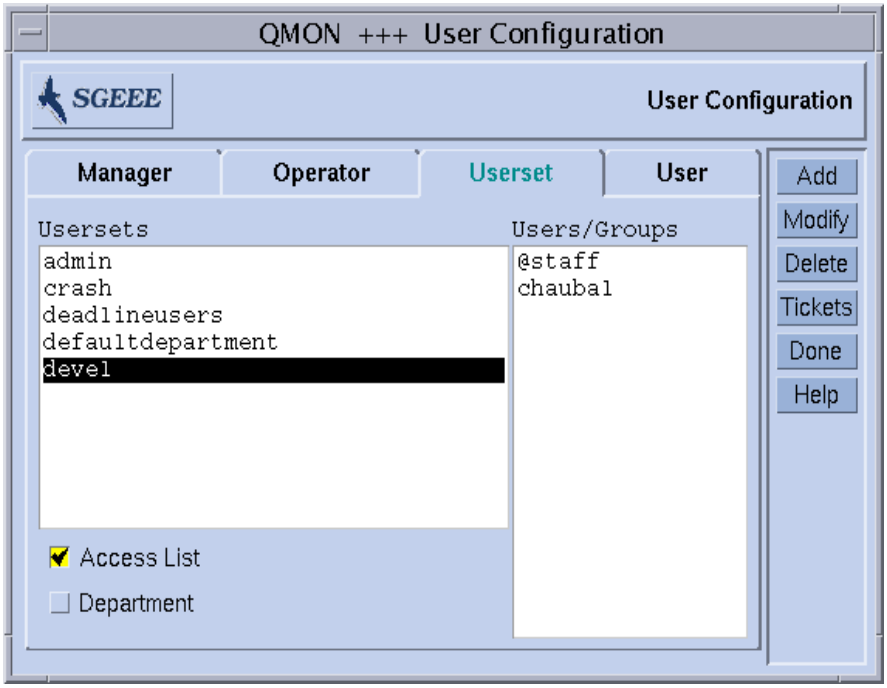


図 9-3　「ユーザーセット構成」ダイアログボックス

画面の左側は「ユーザーセット」選択リストで、選択可能なアクセスリストが表示されます。「アクセスリスト」選択リストでアクセスリストをクリックすると、「ユーザー / グループ」表示域にそのリストの内容が表示されます。

注 – グループはユーザーと区別され、先頭に @ 記号が付いています。

Sun Grid Engine, Enterprise Edition では、ユーザーセットはアクセスリストのこともあれば、部署あるいはその両方のこともあります。どちらのタイプのユーザーセットであるかは、「ユーザーセット」選択リストの下にある 2 つのフラグで示されます。この節では、すべてのユーザーセットがアクセスリストであると仮定します。部署については、230 ページの「ユーザーセットを使用したプロジェクトと部署の定義」の節を参照してください。

「ユーザーセット構成」ダイアログボックスでは、以下のことを行うことができます。

- **削除** - ダイアログボックスの「ユーザーセット」選択リストで既存のアクセスリスト名をクリックし、右側の「削除」ボタンをクリックすると、リストからそのアクセスリストが削除されます。
- **追加** - 「追加」ボタンをクリックすると、新しいユーザーセットが追加されます。
- **変更** - 「変更」ボタンをクリックすると、選択されたアクセスリストが変更されます。

追加および変更の場合は、図 9-4 に示すような「アクセスリストの定義」ダイアログボックスが開き、対応する機能が提供されます。

図 9-4　「アクセスリストの定義」ダイアログボックス

「アクセスリストの定義」ダイアログボックスの各部

- **「ユーザーセット名」**入力フィールド - 変更の場合は、選択されたアクセスリスト名が表示されます。追加の場合は、このフィールドを使用して定義するアクセスリストの名前を入力することができます。
- **「ユーザー / グループ」**表示域 - それまでに定義されているアクセスリストエントリが表示されます。
- **「ユーザー / グループ」**入力フィールド - アクセスリストへの新規エントリの追加に使用します。

キーボードの Return キーを押すと、入力したユーザーまたはグループ名 (グループ名には先頭に @ が付く) が「ユーザー / グループ」表示域のリストの最後に追加されます。表示域で項目を選択し、ゴミ箱のアイコンをクリックすることによって、項目を削除することもできます。

Sun Grid Engine, Enterprise Edition におけるアクセスリストの定義では、必ず「アクセスリスト」フラグが選択されていることを確認してください。「部署」フラグについては、230 ページの「ユーザーセットを使用したプロジェクトと部署の定義」の節を参照してください。

変更したか、新規定義したアクセスリストは、「**了解**」ボタンをクリックすると登録され、「キャンセル」ボタンをクリックすると廃棄されます。どちらの場合も、「アクセスリストの定義」ダイアログボックスは閉じます。

▼ コマンド行からユーザーアクセスリストを構成する

- **適切なオプションを付けて次のコマンドを入力します。**

```
# qconf switches
```

使用可能なオプション

- `qconf -au` *user_name[,...] access_list_name[,...]*

  ユーザーの追加 - 指定されたアクセスリストにユーザーを追加します (どちらも複数指定可能)。

- `qconf -Au` *filename*

  ファイルからのユーザーアクセスリストの追加 - 構成ファイル *filename* を使用してアクセスリストを追加します。

- `qconf -du` *user_name[,...] access_list_name[,...]*

  ユーザーの削除 - 指定されたアクセスリストからユーザーを削除します (どちらも複数指定可能)。

- `qconf -dul` *access_list_name [,...]*

  ユーザーリストの削除 - ユーザセットリストを完全に削除します。

- `qconf -mu` *access_list_name*

  ユーザーアクセスリストの変更 - 特定のアクセスリストを変更するときに使用します。

- `qconf -Mu` *filename*

  ファイルからのユーザーアクセスリストの変更 - 構成ファイル *filename* を使用して、指定されたアクセスリストを変更します。

- `qconf -su` *access_list_name[,...]*

  ユーザーアクセスリストの表示 - 指定されたアクセスリストを表示します。

- `qconf -sul`

  ユーザーアクセスリストの表示 - 現在定義されているすべてのアクセスリストを一覧表示します。

ユーザーセットを使用したプロジェクトと部署の定義

ユーザーセットは、Sun Grid Engine, Enterprise Edition プロジェクト (233 ページの「プロジェクト」を参照) と部署の定義にも使用されます。部署は、Sun Grid Engine, Enterprise Edition のポリシーの業務優先 (261 ページの「業務優先ポリシー」を参照) と一時優先 (270 ページの「一時優先ポリシー」を参照) の構成に使用されます。部署はアクセスリストとは異なります。アクセスリストでは、同じユーザーを複数のリストに登録できるのに対し、部署では、ユーザーは 1 つの部署のメンバーにしかなれません。また、予約名の `deadlineusers` を持つユーザーセットは、Sun Grid Engine, Enterprise Edition ソフトウェアを使用して締め切り優先ジョブを実行依頼できるすべてのユーザーが含まれます。

ユーザーセットが部署であることは、図 9-3 と 図 9-4 に示す「部署」フラグで示されます。ユーザーセットが部署の場合は、そのユーザーセットをアクセスリストとして使用、定義することができます。ただし、その場合は、複数の部署にユーザーを登録できないという制限が適用されます。

ユーザーオブジェクトの構成

個別ユーザーに基本割当、業務優先、一時優先ポリシーのいずれかを定義する場合は (249 ページの「QMON からポリシー / チケットに基づく高度な資源管理を実施する」を参照)、前もってそれらユーザーを定義しておく必要があります。ユーザーを定義するには、「ユーザー構成」ダイアログボックスを使用します。

▼ QMON からユーザーオブジェクトを構成する

1. QMON のメインメニューで「ユーザー構成」ボタンをクリックします。

2. 画面上部の「ユーザー」タブを選択します。

図 9-5 に示すような「ユーザー構成」ダイアログボックスが表示されます。

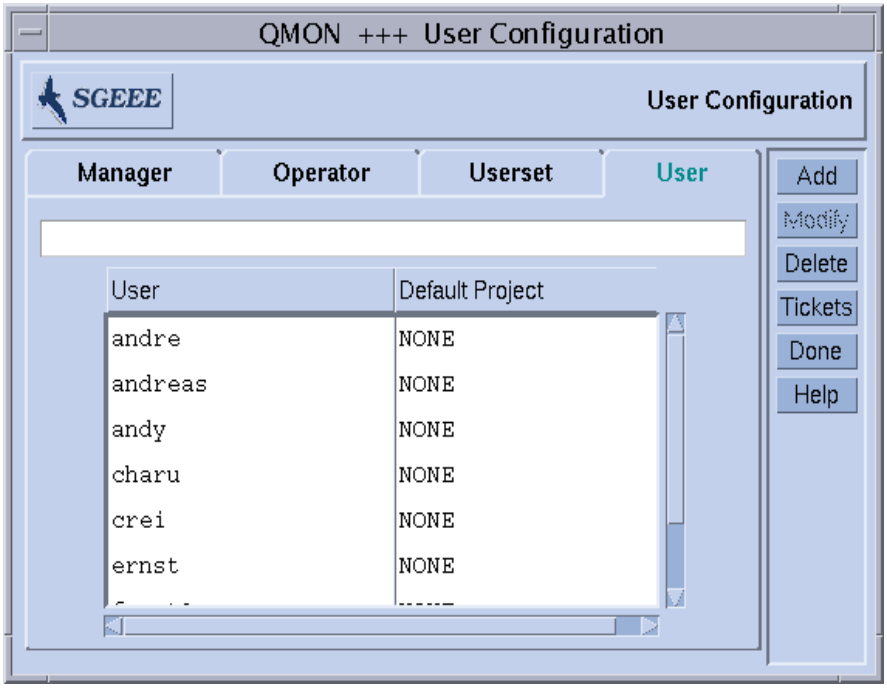


図 9-5　「ユーザー構成」ダイアログボックス

3. 行おうとする作業に従って、ダイアログボックス上部の入力フィールドにユーザー名を入力するか、リストからユーザー名を選択して、以下の適切な操作を行います。

追加または削除

- **新規ユーザー名の追加** - ユーザー名を入力した後、「追加」ボタンをクリックするか、キーボードの Return キーを押します。
- **ユーザー名の削除** - ユーザー名を選択した後、「削除」ボタンをクリックします。

▼ ユーザーにデフォルトのプロジェクトを割り当てる

各ユーザーにデフォルトのプロジェクトを割り当てることができます (233 ページの「プロジェクト」を参照)。割り当てられたデフォルトプロジェクトはユーザーのす

べてのジョブに関連付けられ、そのユーザーは実行依頼で別のプロジェクトを要求する必要がなくなります。

1. デフォルトのプロジェクトを割り当てるには、ユーザーエントリをクリックして選択状態にします。
2. リストの上の「デフォルトプロジェクト」ボタンをクリックします。

   図 9-6 に示すような「プロジェクト選択」ダイアログボックスが表示されます。

図 9-6　「プロジェクト選択」ダイアログボックス

3. 選択状態のユーザーエントリに対して適切なプロジェクトを選択します。
4. 「了解」をクリックしてデフォルトのプロジェクトを割り当て、ダイアログボックスを閉じます。

▼ コマンド行からユーザーオブジェクトを構成する

- 適切なオプションを付けて次のコマンドを入力します。

```
# qconf options
```

使用可能なオプション

- `qconf -auser`

  ユーザーの追加 - `$EDITOR` に指定されたエディタまたはデフォルトの `vi` でユーザー構成用のテンプレートが開き、このテンプレートを編集することができます (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `user` 項を参照)。変更内容を保存してエディタを閉じると、変更内容が `sge_qmaster` に登録されます。

- `qconf -Auser` *filename*

  ファイルからのユーザーの追加 - 指定されたファイルを構文解析して、ユーザー構成を追加します。ファイルは、ユーザー構成用のテンプレート形式である必要があります。

- `qconf -duser` *user_name[,...]*

  ユーザーの削除 - ユーザーオブジェクト (複数指定可能) を削除します。

- `qconf -muser` *user_name*

  ユーザーの変更 - 既存のユーザーエントリを変更します。`$EDITOR` に指定されたエディタまたはデフォルトの `vi` にユーザー構成が読み込まれて、編集することができます。変更内容を保存してエディタを閉じると、変更内容が `sge_qmaster` に登録されます。

- `qconf -Muser` *filename*

  ファイルからのユーザーの変更 - 指定されたファイルを構文解析して、ユーザー構成を変更します。ファイルは、ユーザー構成用のテンプレート形式である必要があります。

- `qconf -suser` *user_name*

  ユーザーの表示 - 特定のユーザーの構成を表示します。

- `qconf -suserl`

  ユーザーリストの表示 - 現在定義されているすべてのユーザーを一覧表示します。

プロジェクト

Sun Grid Engine, Enterprise Edition プロジェクトは、複数のユーザーの計算業務を共同業務として 1 つにまとめ、その共同業務に関係するすべてのジョブに対して資源利用ポリシーを定義する手段です。プロジェクトは、3 つのスケジューリングポリシーで使用されます。

- 基本割当 - プロジェクトに資源が配分されます (250 ページの「基本割当ポリシー」を参照)。

- 業務優先 - プロジェクトは業務優先チケットの一定割合を受け取ります (261 ページの「業務優先ポリシー」を参照)。
- 一時優先 - 管理者によってプロジェクトに一時優先チケットが付与されます (270 ページの「一時優先ポリシー」を参照)。

注 – これらのポリシーでプロジェクトを使用するには、前もってプロジェクトを定義しておく必要があります。

Sun Grid Engine, Enterprise Edition のマネージャーは、プロジェクトに名前を付け、いくつかの属性を設定することによって Sun Grid Engine, Enterprise Edition プロジェクトを定義します。Sun Grid Engine, Enterprise Edition ユーザーは、ジョブの実行依頼で ジョブにプロジェクトを関連付けることができます。基本割当、業務優先、一時優先チケットの配分はプロジェクトによって異なるため、ジョブとプロジェクトの関連付けはジョブのディスパッチの優先順位に影響します。

▼ QMON からプロジェクトを定義する

Sun Grid Engine, Enterprise Edition のマネージャーは、「プロジェクト構成」ダイアログボックスを使用して、Sun Grid Engine, Enterprise Edition プロジェクトを定義、更新することができます。

1. QMON のメインメニューでプロジェクト構成のアイコンをクリックします。
   図 9-7 に示すような「プロジェクト構成」ダイアログボックスが表示されます。

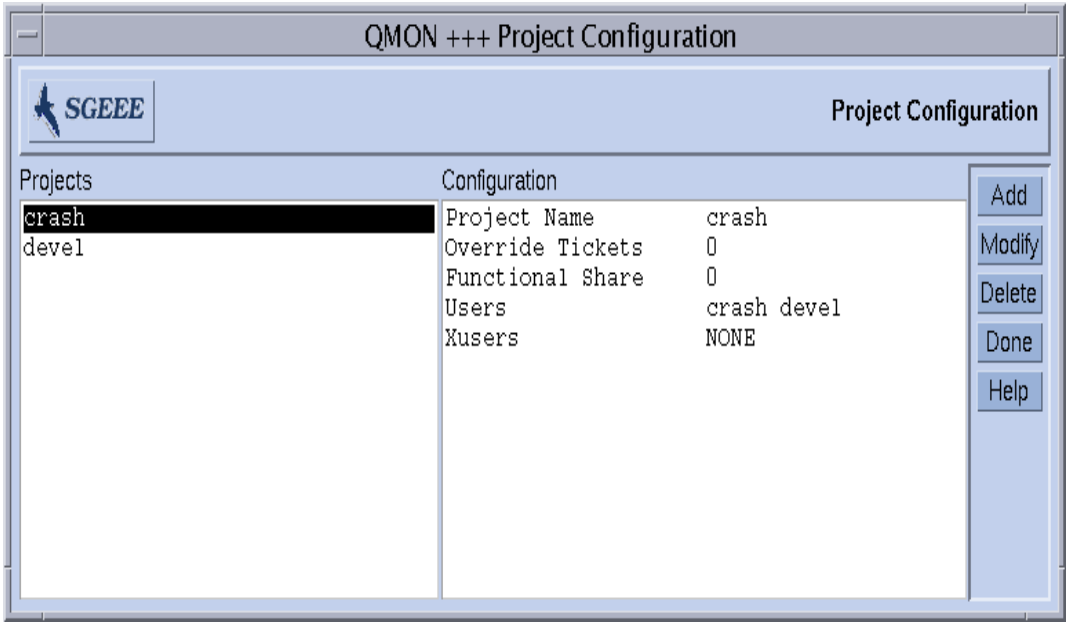


図 9-7　「プロジェクト構成」ダイアログボックス

画面の左側は「プロジェクト」選択リストで、定義済みのプロジェクトが表示されます。

2. センタクリストでプロジェクト名をクリックします。
   「構成」ウィンドウにプロジェクトの定義が表示されます。
3. 以下のいずれか適切な操作を行います。
   a. 選択したプロジェクトを削除する場合は、「削除」をクリックします。

b. **新規プロジェクトを追加する場合は「追加」、選択したプロジェクトを変更する場合は「変更」をクリックします。**

「追加」または「変更」をクリックすると、図 9-8 に示すような「プロジェクトの追加 / 変更」ダイアログボックスが表示されます。

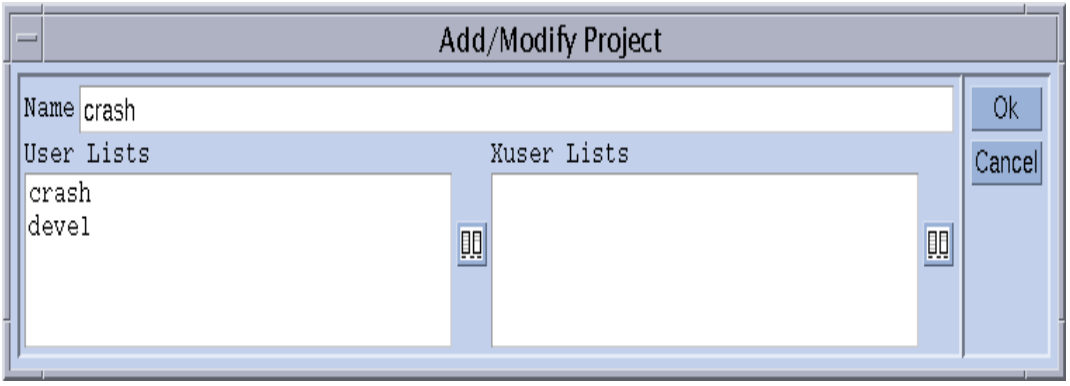


図 9-8　「プロジェクトの追加 / 変更」ダイアログボックス

c. **次の説明に従って「プロジェクトの追加 / 変更」ダイアログボックスでの操作を行います。**

- **プロジェクトの追加または変更** - ダイアログボックスの上部の「名前」入力フィールドはプロジェクト名を示します。 プロジェクトは、それへのアクセスが許可または拒否されているユーザーによって定義されます。
- **許可または拒否の指定** - ユーザーリスト (アクセス許可) または X ユーザーリスト (アクセス拒否) にユーザーアクセスリスト (226 ページの「ユーザーのアクセス権」の節を参照) を関連付けることによってアクセスを許可するかどうかを指定します。ユーザーリストに関連付けられたアクセスリストに登録されているユーザーまたはユーザーグループは、そのプロジェクトのジョブの実行依頼が許可されます。X ユーザーリストに登録されているユーザーまたはユーザーグループは、プロジェクトの使用が拒否されます。両方のリストが空の場合は、あらゆるユーザーがそのプロジェクトを利用できます。1 人のユーザーが複数のアクセスリストに登録されていて、それらのリストがユーザーリストと拒否ユーザーリストの両方に関連付けられている場合、そのユーザーのアクセスは拒否されます。
- **ユーザーリストおよび X ユーザーリストのユーザーの追加または削除** - ユーザーリストまたは X ユーザーリストの右側にあるアイコンのボタンをクリックすると、図 9-9 に示すような「アクセスリストの選択」ダイアログボックスが表示されます。

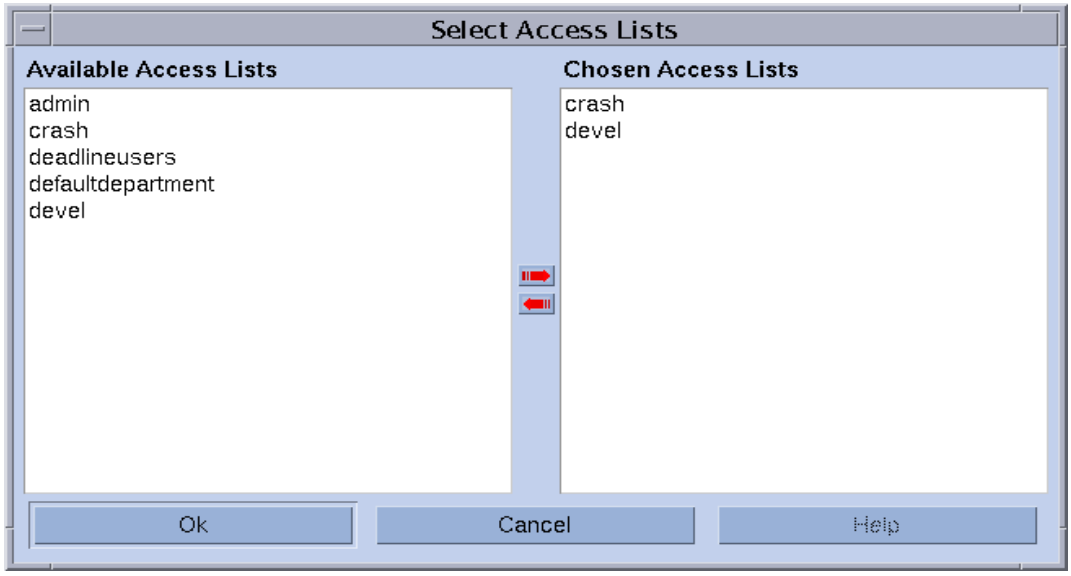


図 9-9　「アクセスリストの選択」ダイアログボックス

このダイアログボックスの「使用可能なアクセスリスト」区画には、定義されているすべてのアクセスリスト、「選択されているアクセスリスト」区画には、関連付けられているすべてのアクセスリストが表示されます。どちらの区画でもアクセスリストを選択して、矢印アイコンを使用して区画間を移動させることができます。

d. 「了解」ボタンをクリックして変更を確定し、ダイアログボックスを閉じます。

▼ コマンド行からプロジェクトを定義する

- 適切なオプションを付けて次のコマンドを入力します。

```
# qconf options
```

使用可能なオプション

- `qconf -aprj`

  プロジェクトの追加 - `$EDITOR` に指定されたエディタまたはデフォルトの `vi` でプロジェクト構成用のテンプレートが開き、このテンプレートを編集することができます (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `project` 項を参照)。変更内容を保存してエディタを閉じると、変更内容が `sge_qmaster` に登録されます。

- `qconf -Aprj` *filename*

  ファイルからのプロジェクトの追加 - 指定されたファイルを構文解析して、プロジェクト構成を追加します。ファイルは、プロジェクト構成用のテンプレート形式である必要があります。

- `qconf -dprj` *project_name[,...]*

  プロジェクトの削除 - プロジェクト (複数指定可能) を削除します。

- `qconf -mprj` *project_name*

  プロジェクトの変更 - 既存のプロジェクトエントリを変更します。`$EDITOR` に指定されたエディタまたはデフォルトの `vi` にプロジェクト構成が読み込まれて、編集することができます。変更内容を保存してエディタを閉じると、変更内容が `sge_qmaster` に登録されます。

- `qconf -mprj` *filename*

  ファイルからのプロジェクトの変更 - 指定されたファイルを構文解析して、既存のプロジェクト構成を変更します。ファイルは、プロジェクト構成用のテンプレート形式である必要があります。

- `qconf -sprj` *project_name*

  プロジェクトの表示 - 特定のプロジェクトの構成を表示します。

- `qconf -sprjl`

  プロジェクトリストの表示 - 現在定義されているすべてのプロジェクトを一覧表示します。

スケジューリング

Sun Grid Engine, Enterprise Edition システムのジョブスケジューリングでは、以下の処理が行われます。

- **ディスパッチ前の決定** - 実行キューが一杯か過負荷の場合にそのキューを削除したり、待機領域で現在の実行対象になっていないジョブをスプールしたりする処理です。

- **ディスパッチ** - 保留中および実行中の他のすべてジョブとの比較でジョブの重要性を決定し、クラスタ内のすべてのマシンの負荷を調べて、設定されている選択条件に従って選択されたマシン上の実行キューにジョブを送信します。
- **ディスパッチ後の監視** - 資源の取得状況、さらにはそれぞれに相対的な重要性を持つ他のジョブのシステムの入出状況に応じて、ジョブの相対的な重要性を調整します。

Sun Grid Engine, Enterprise Edition ソフトウェアは、次の条件に基づき異機種コンピュータからなるクラスタ全体にわたってジョブをスケジューリングします。

- クラスタの現在の負荷
- ジョブの相対的な重要性
- ホストの相対的な性能
- ジョブが必要とする資源条件 (CPU、メモリー、入出力帯域幅など)

スケジューリングの決定は、現場の戦略とクラスタを構成する各コンピュータの瞬間的な負荷に基づいて行われます。現場のスケジューリング戦略は、Sun Grid Engine, Enterprise Edition システムの構成パラメータを使用して表現します。負荷特性は、動作中のシステムのパフォーマンスデータを収集することによって確認されます。

スケジューリング戦略

Sun Grid Engine, Enterprise Edition 管理者は、次のスケジューリング業務について戦略を立てることができます。

- **動的資源管理** - Sun Grid Engine, Enterprise Edition システムは、実行中のジョブに割り当てられている資源利用資格を動的に制御、調整します (CPU 配分の変更)。
- **キューのソート** - キューを埋める順番に従ってクラスタ内のキューをランク付けします。
- **ジョブのソート** - Sun Grid Engine, Enterprise Edition システムがジョブをスケジューリングする順番を決定します。

動的資源管理

Sun Grid Engine, Enterprise Edition ソフトウェアでは、4 つのポリシーの重みの組み合わせに基づくジョブスケジューリング戦略の自動化を実現します。

- 基本割当
- 業務優先 (プライオリティともいう)
- 締め切り優先
- 一時優先

Sun Grid Engine, Enterprise Edition システムが日常的に基本割当ポリシーか業務優先ポリシー、またはその両方を使用するように構成することができます。これらのポリシーは、0 から 1 の範囲で重み与えたり、2 つ目だけ使用して両方に同じ重みを与えたりなどの任意の比率で組み合わせることができます。

これらの定期ポリシーのほかに、締め切り優先でジョブの実行を依頼することもできます。締め切り優先ジョブは、定期スケジューリングに影響します。管理者はまた、緊急を要する場合などに一時的に、あるいは恒久的に基本割当や業務優先、締め切り優先スケジューリングを無効にすることもできます。一時優先は、特定の 1 つジョブに対して適用することも、特定のユーザー、部署、プロジェクト、ジョブクラス (すなわち、キュー) に関連付けられているすべてのジョブに適用することもできます。

Sun Grid Engine, Enterprise Edition には、すべてのジョブ間の調停をするためのこれら 4 つのポリシーのほかに、ユーザーが自分のジョブに優先順位を設定する機能もあります。たとえばユーザーが 3 つのジョブを実行依頼しようとしていて、ジョブ 3 が最重要で、ジョブ 1 と 2 の重要性は同じであると仮定します。この重要性に基づくスケジューリングは、基本割当か業務優先、またはその両方のポリシーとジョブへの業務優先チケットの付与を組み合わせることによって実現することができます。

スケジューリングポリシーは、チケットを使用して実現されます。各ポリシーにはチケットプールがあり、そこから、複数マシンからなる Sun Grid Engine, Enterprise Edition システムに入るジョブにチケットが割り当てられます。定期ポリシーを有効にすると、新規ジョブの 1 つ 1 つにチケットが割り当てられ、スケジューリングのたびに実行中のジョブへのチケットの再割り当てが試みられます。以下では、各ポリシーがチケットを割り当てるときに使用する基準を説明します。

チケットは 4 つのポリシーに重みを付けます。たとえば業務優先ポリシーにチケットが割り当てられていない場合、業務優先ポリシーは使用されません。業務優先と基本割当チケットプールに同数のチケットが割り当てられている場合、両方のポリシーはジョブの重要性の決定に際して同等の重みを持ちます。

システム構成時、Sun Grid Engine, Enterprise Edition のマネージャーは定期ポリシーにチケットを割り当てます。その後、マネージャーおよびオペレータはいつでもチケット割当量を変更して、すぐに有効にすることができます。締め切り優先または一時優先を指示するには、システムに一時的に追加チケットを注入します。ポリシーはチケットの割り当てによって組み合わせられます。複数のポリシーにチケットが割り当てられている場合、ジョブは有効な各ポリシーにおけるその重要性に応じて割当分のチケットを受け取ります。

Sun Grid Engine, Enterprise Edition は、システムに入るジョブにチケットを付与することによって、有効な各ポリシーにおけるその重要性を指示します。実行中のジョブは、スケジューリングのたびにチケットが増えることもあれば (たとえば一時優先が行われたり、締め切りが迫っていたりするなどの理由)、減ることもあり (たとえば、正当な量を超える資源配分を受けているなどの理由)、同じチケット数が維持されることもあります。ジョブが保持するチケット数は、Sun Grid Engine, Enterprise Edition がスケジューリング中にそのジョブに付与しようとする資源配分量を表します。

現場の動的資源管理戦略は、Sun Grid Engine, Enterprise Edition のインストール中に、基本割当と業務優先ポリシーにチケットを割り当てて、基本割当ツリーと業務優先の配分量を定義し、締め切り優先チケットの最大数設定することによって構成します。基本割当と業務優先チケットの割当量と締め切り優先チケットの最大数は、いつでも自動的に変更されることがあります。一時優先チケットは、管理者が手動で割り当てまたは割り当て解除します。

キューのソート

Sun Grid Engine, Enterprise Edition には、以下の方法を使用してキューを埋める順番を決めることができます。

- **負荷報告** - Sun Grid Engine, Enterprise Edition 管理者は、ホストおよびそのキューの負荷状態の比較に使用する負荷パラメータを選択することができます。214 ページの「負荷パラメータ」の節では、標準で用意されている広範囲の各種負荷パラメータと、現場に固有の負荷センサーを利用してその標準のパラメータセットを拡張するためのインタフェースを説明しています。
- **負荷スケーリング** - さまざまなホストからの負荷レポートを正規化して、類似の状況を比較することができます (153 ページの「QMON から実行ホストを構成する」を参照)。
- **負荷調整** - ホストにジョブをディスパッチしたときに、前回報告された負荷を Sun Grid Engine, Enterprise Edition が自動的に修正するよう構成することができます。修正された負荷は、最近のジョブの実行開始で発生すると予想される負荷の増加を表します。この人為的に増加された負荷は、それらのジョブの負荷に対する影響が明らかになると、自動的に減少させることができます。
- **連続番号** - 厳密な順序でキューをソートすることができます。
- **ホストの能力** - クラスタを構成するマシンの相対的な能力を定義することによって、能力インジケータに基づいてホストとそのキューをソートすることができます。

ジョブのソート

ディスパッチを開始する前、Sun Grid Engine, Enterprise Edition は優先順位の高い順にジョブをソートし、その順番でジョブに適切な資源を見つけようとします。管理者の介入がなければ、この順番は FIFO (先入れ先出し) 方式です。管理者は、このジョブの順番を以下の方法で制御することができます。

- **チケットに基づくジョブ優先順位** - Sun Grid Engine, Enterprise Edition では、ジョブはつねに、それが所有するチケット数で決まる相対的な重要性に応じた扱いを受けます。保留中のジョブはチケットの多い順にソートされ、管理者がチケットポリシーの変更を行うと、このソート順が変更されます。

- **ユーザー / グループのジョブの最大数** - 1 人のユーザーまたは 1 つの UNIX ユーザーグループが Sun Grid Engine, Enterprise Edition で並行して実行できるジョブの最大数を制限することができます。ユーザーのジョブがこの制限を超えないことが優先されるため、保留中のジョブリストのソート順はその影響を受けます。

スケジューリング時に行われる処理

スケジューラは間隔を置いてスケジューリングします。Sun Grid Engine, Enterprise Edition は、次のスケジューリングまで、ジョブの実行依頼や完了、取り消し、クラスタ構成の変更、クラスタへの新規マシンの登録などの重要なイベントに関する情報を保持します。スケジューリング時間になると、スケジューラは以下のことを行います。

- すべての重要イベントの考慮
- 管理者の指定に応じたジョブとキューのソート
- すべてのジョブの資源要求内容の考慮

この後、Sun Grid Engine, Enterprise Edition システムは必要に応じて以下のことを行います。

- 新規ジョブのディスパッチ
- 実行中のジョブの一時停止
- 実行中のジョブに割り当てられている資源の増減
- 現状の維持

Sun Grid Engine, Enterprise Edition システムで基本割当スケジューリングが使用されている場合は、そのユーザーまたはプロジェクトに対してすでに発生した資源利用が考慮されます。少なくとも一部でもスケジューリングが基本割当スケジューリングでない場合は、クラスタ内の資源 (CPU、メモリー、入出力帯域幅) が最大限に利用されるよう、実行中および実行待ちのすべてのジョブがランク付けされて、重要性の高い順にジョブが処理されます。

スケジューラ監視

ジョブが開始されず、その理由が不明な場合は、`-w v` オプションを付けて `qalter` を実行します。Sun Grid Engine, Enterprise Edition ソフトウェアはクラスタが空であるとみなし、使用できるキューでそのジョブに合ったキューがないかどうかを調べます。

`qstat -j` *job_id* を実行することによって、さらに詳しい情報を得ることもできます。このコマンドは、前回のスケジューリングでジョブがスケジューリングされなかった理由を含めて、ジョブの要求プロファイルの要約情報を表示します。ジョブ ID なしで `qstat -j` を実行すると、前回のスケジューリングでスケジューリングされなかったすべてのジョブに関する要約情報が表示されます。

注 – この機能を利用するには、スケジューラ構成 `sched_conf` でスケジューリング理由情報の収集を有効にする必要があります。『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `schedd_job_info` パラメータか、このマニュアルの 246 ページの「`QMON` からスケジューラ構成を変更する」の節を参照してください。

`qcon` コマンドの `-tsm` オプションを使用することによって、Sun Grid Engine, Enterprise Edition のスケジューラ `sge_schedd` の決定に関するさらに詳細な情報を得ることもできます。このコマンドを使用すると、`sge_schedd` が強制的にファイルにトレース出力を書き込みます。

スケジューラ構成

Sun Grid Engine, Enterprise Edition のチケットに基づく資源共有ポリシーのスケジューリング管理についての詳細は、249 ページの「`QMON` からポリシー / チケットに基づく高度な資源管理を実施する」を参照してください。この節では、スケジューラ構成の `sched_conf` の管理とそれに関連する問題に焦点を当てます。

デフォルトのスケジューリング

Sun Grid Engine, Enterprise Edition のデフォルトのスケジューリングは FIFO 方式です。すなわち、先に実行依頼のあったジョブが、先にスケジューラによって調べられ、キューにディスパッチされます。保留中のジョブリストの先頭のジョブに対する休止中で適切なキューが見つかると、スケジューリングでそのジョブが最初に処理されます。保留中のジョブリストの先頭のジョブより先に他の 2 番目以降のジョブが処理されるのは、先頭ジョブに対する適切な未使用資源が見つからなかった場合だけです。

ジョブに対するキューの選択に関するデフォルト戦略では、ジョブの資源要求内容に合ったサービスが提供されるのである限り、 Sun Grid Engine, Enterprise Edition は最も負荷の小さいホストのキューを選択します。適切なキューが複数あり、それらの負荷が同じ場合は、どのキューが選択されるか予測できません。

その他のスケジューリング方法

Sun Grid Engine, Enterprise Edition には、ジョブのスケジューリングとキュー選択方法を変更するさまざまな方法が用意されています。

- スケジューリングアルゴリズムの変更
- システム負荷のスケーリング
- 連続番号によるキューの選択
- 配分量によるキューの選択

■ ユーザー 1 人またはグループ 1 つあたりのジョブ数の制限

以下では、デフォルトに替わるこれらのスケジューリング方法を詳しく説明します。

スケジューリングアルゴリズムの変更

スケジューラ構成パラメータの `algorithm` は、使用するスケジューリングアルゴリズムの選択を可能にするパラメータです (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』 の `sched_conf` 項を参照)。ただし、現在指定できる設定は `default` だけです。

システム負荷のスケーリング

Sun Grid Engine, Enterprise Edition システムは、キューのホストであるマシンのシステム負荷情報に基づいて、ジョブに対する実行キューを選択します。このキュー選択方式によって負荷均衡状態が確立され、クラスタ内の使用可能な資源のより優れた利用が保証されます。

ただし、システム負荷がつねに真実を伝えるとは限りません。たとえば複数 CPU のマシンと単一 CPU のマシンを比較すると、通常、マルチプロセッサシステムが報告する負荷値の方が大きくなります。これは、たいていマルチプロセッサシステムの方が実行しているプロセス数が多く、システム負荷が CPU へのアクセス権を取得しようとするプロセス数に大きく左右される測定値であるためです。しかし、複数 CPU システムは、単一 CPU マシンに比べてずっと大きな負荷に対処することができます。この問題は、`sge_execd` からデフォルトで報告される負荷値をプロセッサ数で調整することによって対処します (詳細は、214 ページの「負荷パラメータ」の節と *<sge_root>*`/doc/load_parameters.asc` ファイルを参照)。すなわち、生の負荷値ではなく、負荷パラメータを使用することによって、上記の問題に対処することができます。

負荷値が正しく判断されない可能性があるもう 1 つの例として、潜在的な性能あるいは価格性能費に大きな差があるシステムがあります。どちらの場合も、負荷値が同じであるからといって、ジョブの実行用にどちらのホストを選択してもよいわけではありません。こうした場合は、問題のホストと負荷パラメータに対する負荷スケーリング係数を定義することを推奨します (153 ページの「`QMON` から実行ホストを構成する」と関係する節を参照)。

注 – スケーリングした負荷パラメータは、負荷しきい値リストの *load_thresholds* および *migr_load_thresholds* との負荷パラメータの照合にも使用されます (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `queue_conf` 項を参照)。

負荷パラメータには、さらにもう 1 つ問題があります。それは、業務および現場によって負荷値とその相対的な重要性の解釈を変える必要があることです。あるサイトで一般的なある種の業務では CPU 負荷が圧倒的であるのに対し、一般に計算クラスタが特定業務プロファイル専用になっている別のサイトではメモリー負荷がずっと重要であることがあります。Sun Grid Engine, Enterprise Edition では、スケジューラ構成ファイル `sched_conf` にいわゆる負荷の式 (*load formula*) を指定することによって、この問題に対処することができます (詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の対応する節を参照)。すなわち、負荷の式でサイト定義の負荷パラメータ (214 ページの「サイトに固有の負荷パラメータの追加」の節を参照) と消費可能資源 (201 ページの「消費可能資源」の節を参照) を使用することによって、資源利用と能力利用計画に関するサイト固有の情報を考慮することができます。

システム負荷の問題として、最後に負荷パラメータの時間依存も考慮する必要があります。システムで実行中の Sun Grid Engine, Enterprise Edition ジョブによって課される負荷は時間とともに変化し、しばしば、オペレーティングシステムが適切な報告するのにそれなりの時間が必要になることがあります (たとえば CPU 負荷のため) このため、ジョブの開始直後の場合、報告される負荷は、ジョブによってホストにすでに課されている負荷を十分に表していないことがあります。報告される負荷は時間とともに実際の負荷に近づいていきますが、低すぎる間は、そのホストが過剰な実行依頼を受ける可能性があります。Sun Grid Engine, Enterprise Edition 管理者は、Sun Grid Engine, Enterprise Edition スケジューラがこの問題の補正に使用する負荷調整係数を指定することができます。負荷調整係数の設定方法についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』およびスケジューラ構成ファイル `sched_conf` を参照してください。

連続番号によるキューの選択

デフォルトのキュー選択方法を変更するもう 1 つの方法は、Sun Grid Engine, Enterprise Edition のグローバルクラスタ構成パラメータの `queue_sort_method` をデフォルトの `load` から `seq_no` に変更する方法です (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `sched_conf` 項を参照)。この設定にすると、キュー選択の第 1 手段としてシステム負荷が使用されなくなります。キュー構成パラメータ `seq_no` ですべてのキューに割り当てられた連続番号が (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `queue_conf` 項を参照) 、キューの一定の優先順位を定義する第一の手段になり、キューが検討対象のジョブに適切で未使用の場合は、その優先順位でキューが選択されます。

このキュー選択方法は、たとえばマシン A であるジョブを実行すると 1 単位のコストがかかるのに対し、マシン B では 10 単位、マシン C では 100 単位のコストがかかるというように、ジョブ 1 つあたりのコストでバッチサービスを提供するマシンをランク付けしている場合に役立つかもしれません。つまり、この場合の望ましいスケジューリング方法は、最初にホスト A を一杯にして、次にホスト B、そして代わりが残っていない場合だけホスト C を使用するという方法です。

注 – キュー選択方法を `seq_no` に変更して、検討対象のすべてのキューの連続番号が同じ場合、キューはデフォルトの負荷 (`load`) に基づいて選択されます。

配分量によるキューの選択

この方法の目標は、すべてのジョブに対して目標とするグロバールシステム資源配分が達成されるようにジョブを配置することにあります。この方法では、すべてのシステム資源との関係で各ホストが表す資源能力を考慮し、各ホストの Sun Grid Engine, Enterprise Edition チケットの割合 (すなわち、ホストで実行中のすべてのジョブの総チケット数) と各ホストが表す資源能力の割合のバランスをとろうとします。ホストの能力の定義方法については、153 ページの「`QMON` から実行ホストを構成する」を参照してください。

重要さでは劣りますが、この方法のソーティングでは、ホストの負荷も考慮されます。これは、基本割当ポリシーを採用しているサイトに推奨するソーティング方法です。

ユーザー *1* 人またはグループ *1* つあたりのジョブ数の制限

Sun Grid Engine, Enterprise Edition 管理者は、1 人のユーザーまたは 1 つの UNIX グループが任意の時点で実行できるジョブ数を制限することができます。この機能を使用するには、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `shed_conf` の節で説明しているように、`maxujobs` と `maxgjobs` を設定します。

▼ QMON からスケジューラ構成を変更する

1. **`QMON` のメインメニューから「スケジューラ構成」をクリックします。**

   「スケジューラ構成」ダイアログボックスが表示されます。このダイアログボックスには、「一般パラメータ」と「負荷調整」の 2 つのタブがあります。行おうとする作業に従って、いずれか適切なタブを選択します。

a. 一般的なスケジューリングパラメータを変更するには、「一般パラメータ」タブをクリックします。

図 9-10 に示すような「一般パラメータ」ダイアログボックスが表示されます。

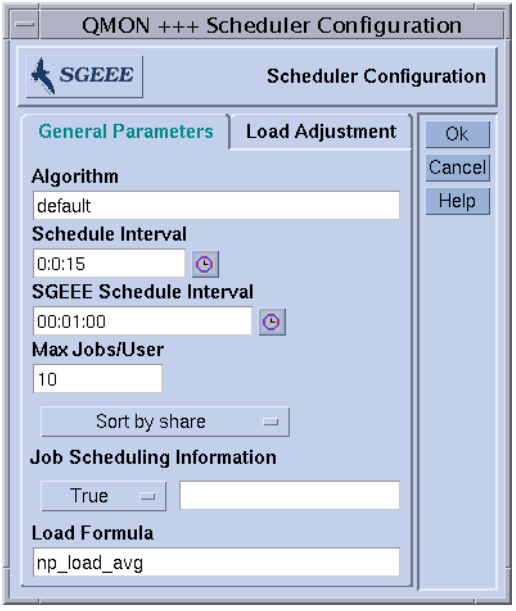


図 9-10　「スケジューラ構成」ダイアログボックス - 一般パラメータ

「一般パラメータ」ダイアログボックスでは、以下のパラメータを設定できます。

- スケジューリングアルゴリズム (244 ページの「スケジューリングアルゴリズムの変更」を参照)
- スケジューラの定期実行間隔
- Sun Grid Engine, Enterprise Edition スケジューラの定期実行間隔 - 資源共有ポリシーに基づいてチケットが再分配されます。
- 1 人のユーザーまたは 1 つの UNIX グループが並行して実行可能なジョブの最大数 (246 ページの「ユーザー 1 人またはグループ 1 つあたりのジョブ数の制限」を参照)。
- キューのソート方法 - 負荷、連続番号、配分量によるソートのいずれか (245 ページの「連続番号によるキューの選択」と 246 ページの「配分量によるキューの選択」を参照)
- `qstat -j` によるジョブスケジューリング情報へのアクセス、または入力フィールドに指定されたジョブ ID 範囲のジョブスケジューリング情報の収集。ジョブスケジューリング情報の一般的な収集は、保留中のジョブ数がきわめて多い場合にのみ一時的に使用することを推奨します。
- ホストおよびキューのソートに使用する負荷の式

b. 負荷調整パラメータを変更するには、「負荷調整」タブを選択します。

図 9-11 に示すような「負荷調整」ダイアログボックスが表示されます。

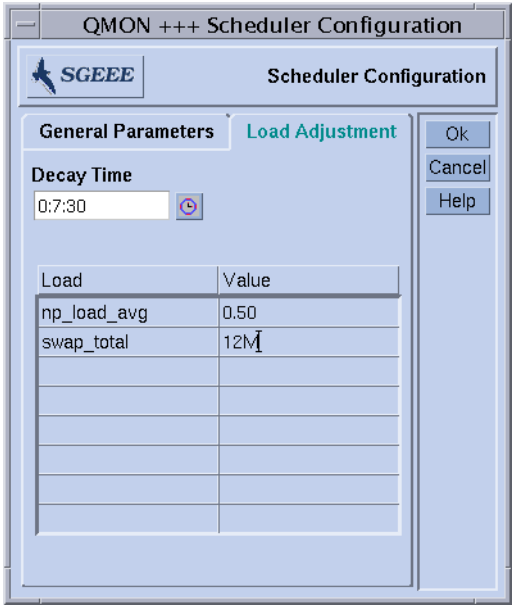


図 9-11 「スケジューラ構成」ダイアログボックス - 負荷調整

「負荷調整」ダイアログボックスでは、以下のパラメータを設定することができます。

- 負荷調整減少時間
- ダイアログボックスの左半分の負荷調整値表 - この表は、現在調整値が定義されているすべての負荷および消費可能属性のリストです。このリストは、上部の「負荷」または「値」ボタンをクリックすることによって拡張することができます。ボタンをクリックすると、ホストに関連付けられているすべての属性 (すなわち、グローバル、ホスト、管理者定義の複合に設定されているすべての属性をまとめたもの) の入った選択リストが開きます (図 6-6 の「属性選択」ダイアログボックスを参照)。どれか属性を選択し、「**了解**」ボタンをクリックして選択を確定すると、その属性が「消費可能 / 固定属性」表の「負荷」列に追加され、対応する「値」フィールドにポインタが移動します。「値」フィールドをダブルクリックすると、既存の値を変更することができます。属性を削除するには、対応する表の行を選択して、Ctrl-D を押すか、マウスの右ボタンをクリックして削除ボックスを開き、削除を確定します。

予備知識的な情報については、244 ページの「システム負荷のスケーリング」を参照してください。スケジューラ構成についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `sched_conf` マニュアルページを参照してください。

▼ QMON からポリシー / チケットに基づく高度な資源管理を実施する

1. QMON のメインメニューで「チケット構成」ボタンをクリックします。

   図 9-12 に示すような「チケット概要」ダイアログボックスが表示されます。

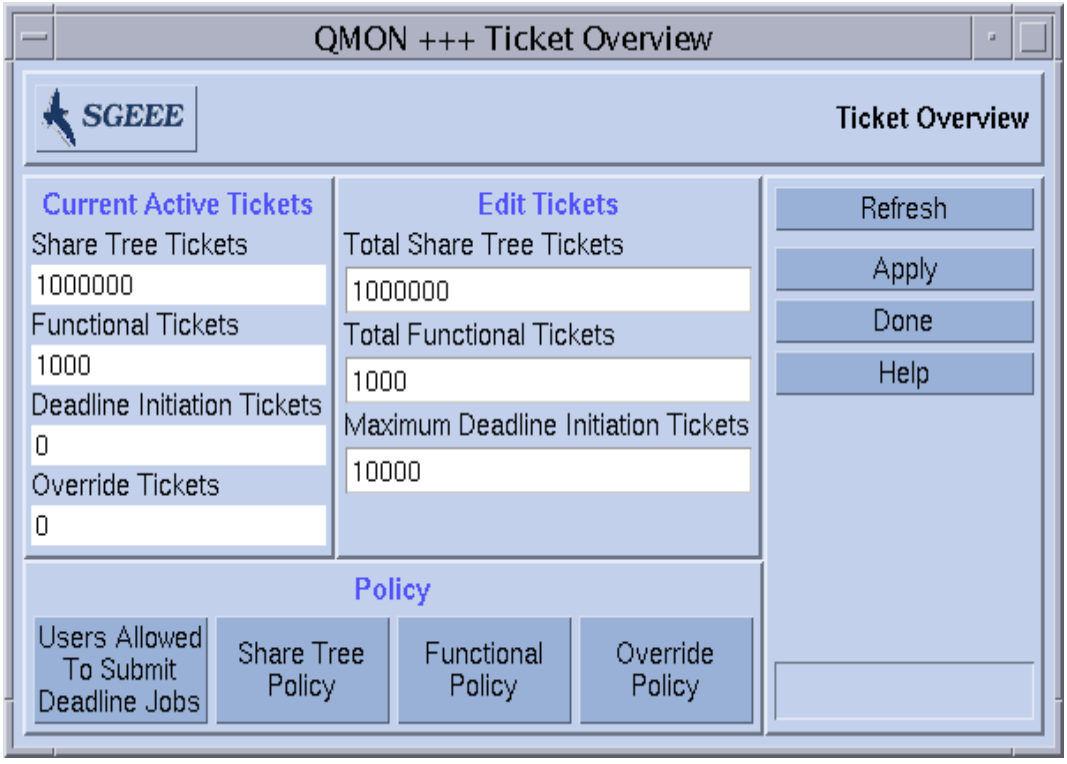


図 9-12　「チケット概要」ダイアログボックス

2. 以下の適切な節に進みます。

   「チケット概要」ダイアログボックスは、チケットに基づくポリシー間の現在のチケット分配量を表示し、このダイアログボックスから、ポリシー関連のチケットを再調整したり、チケットに基づくポリシーに対応する構成ダイアログボックスを開いたりすることができます。

   個々のポリシーに現在割り当てられているチケットは、左側の「現在アクティブなチケット」表示域に表示されます。この数はポリシーの相対的な重要性を表し、特定のポリシーが現在クラスタを優先使用しているかどうか、あるいはポリシー間のバランスが取られているかどうかが分かります。チケットは量的な目安を提供し、このこと

は、たとえば基本割当ポリシーのチケット割当量が業務優先ポリシーの 2 倍の場合、基本割当ポリシーには業務優先ポリシーの 2 倍の資源利用資格があることを意味します。この意味では、チケットは企業の株式に非常によく似ています。

総チケット数に特別な意味はありません。重要なのはポリシー間の関係だけです。つまり、ポリシーの相対的な重要性を細かく調整できるよう、総チケット数は通常かなり大きな数字になります。

チケットの編集

「チケットの編集」区画では、優勢指定以外の各ポリシーに割り当てるチケットを変更することができます。一時優先チケットは、一時優先ポリシー構成から直接割り当てるのに対し、その他のチケットプールは、それらポリシーに関連付けられているジョブ間で実際のポリシー構成に基づいて自動的に分配されます。

注 – 基本割当および業務優先チケットは、いつでも、それらポリシーに関連付けられているジョブ間ですべて分配されます。締め切り優先チケットは、締め切りが迫ってきた場合にのみ締め切り優先ジョブに割り当てられます。現在アクティブなジョブに一時優先チケットを適用することはできないため、一時優先ポリシーにチケットが定義されていても、アクティブな一時優先チケット数がゼロのことがあります。

ポリシーのボタン

「ポリシー」区画には以下のボタンがあります。

- 「ユーザー構成」ダイアログボックスを開くボタン - Deadlineusers ユーザーセット構成に簡単にアクセスできます。
- 基本割当、業務優先、一時優先ポリシー構成ダイアログボックスを開くボタン - 締め切り優先ポリシーには、構成ダイアログボックスはありません。

ダイアログボックスの右側にあるボタンを使用して、画面を再表示したり、変更を適用、廃棄したりすることができます。

基本割当ポリシー

基本割当スケジューリング (基本割当ツリースケジューリングともいう) は、週、月、四半期などの累積期間中にユーザーおよびプロジェクトのそれぞれに事前に決められた配分でシステム資源を付与しようとするスケジューリング方式です。このため、次のスケジューリングまでの短い時間の間に各ユーザーおよびプロジェクトに予定されている資源配分が調整されます。基本割当スケジューリングは、ユーザーまたはプロジェクトごとに定義します。

各ユーザー / プロジェクトにできる限り目標に近い配分を付与することによって、部署などのユーザー / プロジェクトの集団も目標の配分を得られるようになります。すなわち、累積期間中に資源利用資格を持つすべてのエンティティが資源を得ようとした場合にのみ、すべてのエンティティに対する公平な配分を達成できます。ユーザー / プロジェクトまたはその集団が特定の期間中にジョブの実行依頼をしなかった場合、資源は実行依頼をした者の間で分配されます。

基本割当スケジューリングは「フィードバック方式」です。任意のユーザー / ユーザーグループおよび任意のプロジェクト / プロジェクトグループがシステムの利用資格があるかどうかは、Sun Grid Engine, Enterprise Edition の構成パラメータが 1 つで定義されます。また、任意のジョブにシステムの利用資格があるかどうかは、次の要素に基づいて決定されます。

- そのジョブのユーザーまたはプロジェクトへの割り当て分
- 各ユーザー / ユーザーグループ、プロジェクト / プロジェクトグループの累積された過去の使用 - この使用は「減少係数」で調整され、「かなり以前」の使用ほど影響は小さくなります。

Sun Grid Engine, Enterprise Edition はユーザー / プロジェクトの過去の使用量を記録し、スケジューリングのたびにすべてのジョブの資源配分を調整して、すべてのユーザー / ユーザーグループおよびプロジェクト / プロジェクトグループが累積期間の間にできる限りシステムの公平な配分を受けられるようにします。言い替えれば、資源利用を許可したり、拒否したりすることによって、誰もがほぼ目標に近い配分を受けられるようにします。

半減期係数

半減期とは、システムがユーザーの資源消費を「忘れる」速さです。システム管理者は、6 ヶ月前あるいは 6 日前のユーザーの大きな資源消費にペナルティを科すかどうか、科すとすれば、どのようにして科すのかを決定することができます。Sun Grid Engine, Enterprise Edition ソフトウェアは、基本割当ツリーのすべてのノードについてユーザーの資源消費記録を維持します。

基本割当ポリシーを作成する際、システム管理者はこの記録に基づいて、ユーザーの過少利用または過大利用を判断する際、どのくらい過去にさかのぼるかを決定することができます。この意味での資源利用量は、「フレックスな時間枠」で消費されたすべてのコンピュータ資源の数学的な全体 (合計) です。

この時間枠の長さは「半減期」係数で決まり、そこでは Sun Grid Engine, Enterprise Edition システムは内部減少関数です。この減少関数は、経時的な資源消費の影響を小さくします。半減期が短いほど、資源の過大消費の影響は短期間に小さくなり、半減期が長いほど、資源の過大消費の影響は徐々に小さくなります。

Sun Grid Engine, Enterprise Edition システムでは、この半減期減少関数は指定された時間単位に基づきます。たとえば 1,000 単位の資源消費に 7 日の半減期を適用した場合は、経時で次の使用「ペナルティ」による調整が行われます。

- 7 日後 500

- 14 日後 250
- 21 日後 125
- 28 日後 62.5

半減期に基づく減少は、ペナルティ効果が非常に小さくなり、無視できるまで経時のユーザーの資源消費の影響を小さくします。ユーザーが一時優先チケットを受け取った場合、そのチケットが過去の利用ペナルティの影響を受けることはないことに注意してください。一時優先チケットは別のポリシー系のチケットです。減少関数は基本割当ポリシーに固有です。

補正係数

比較でユーザー / プロジェクトの実際の利用量が目標利用量をずっと下回っていることが明らかになった場合は、その資源配分を調整することによって、ユーザー / プロジェクトがシステムを優先使用し、目標の配分の達成を図られるようにすることができます。ただし、この優先使用は望ましいことではないかもしれません。「補正係数」を使用し、管理者は、累積使用量が非常に少ないユーザー / プロジェクトが、指定されている利用目標の達成を図るときに短期に資源を優先使用させる度合いを制限することができます。

たとえば補正係数 2 は、ユーザー / プロジェクトの現在の配分を目標配分の 2 倍に制限します。すなわち、ユーザー / プロジェクトが累積期間中にシステム資源の 20% を受けられるようになっていて、現在実際に受けている量がそれより非常に低い場合は、短期に 40% だけを受けることができます。

基本割当ツリーに従ってユーザー / プロジェクトの長期の資源利用資格が定義される基本割当ポリシーとこの補正係数を組み合わせることによって、利用資格の自動的な調整が行われます。

特定のユーザー / プロジェクトが目標の利用量を下回っているか、上回っている場合、Sun Grid Engine, Enterprise Edition システムは、そのユーザー / プロジェクトの短期の利用資格を長期目標よりも上げるか、下げることによって補正をします。この補正は、Sun Grid Engine, Enterprise Edition システムの基本割当ツリーアルゴリズムの計算によって行われます。

補正係数は、Sun Grid Engine, Enterprise Edition システムが割り当てる補正量を制御するもう 1 つの仕組みを提供します。この別の補正係数 (CF) 計算は、次の条件が満たされる場合にのみ行われます。

- 短期資格 > 長期資格 * CF
- CF > 0

上記の条件の一方または両方が満たされていない場合は、基本割当ツリーアルゴリズムによって定義されているとおりの補正が適用されます。

補正係数の設定の一般的な規則として、CF 値が小さいほど、その効果は大きくなります。CF 値が 1 より大きい場合、Sun Grid Engine, Enterprise Edition システムは補正はしますが、限られた効果しかありません。補正の上限は、長期資格 * CF で求められます。上記で定義されているように、補正係数に基づく操作が行われるには、短期資格がこの上限を超えている必要があることにも注意してください。

CF 値が 1 の場合、Sun Grid Engine, Enterprise Edition システムはそのままの基本割当ツリーアルゴリズムと同じ方法で補正をします。このため、値 1 は値 0 と似た効果になります。唯一の違いは、CF = 0 の場合は、CF 計算が抑止されるのに対して、CF = 1 の場合は、CF 計算を行われる実装仕様になっていることだけです。

値が 1 より小さい場合、Sun Grid Engine, Enterprise Edition システムは「過剰補正」します。ジョブは、基本割当ツリーアルゴリズムに基づいて得られるよりもずっと多くの補正量を受けます。また、補正の実施条件の「短期資格 > 長期資格 * CF」が低い短期資格値で満たされるため、早期にこの過剰補正を受けることになります。

階層形式の基本割当ツリー

基本割当ポリシーは、移動累積期間中にすべてのユーザー / プロジェクトの間でどのようにシステム資源を配分するかということを規定した階層形式の「基本割当ツリー」を使用して実現されます。この累積期間の長さは、設定変更が可能な減少定数で決まります。Sun Grid Engine, Enterprise Edition は、基本割当ツリー内の各親ノードのその累積上限への到達度に基づいて、ジョブが配分を受ける資格を決定します。ジョブのこの資格は、そのリーフノードの割当量に基づいて決まり、リーフノードの割当量はその親ノードの割り当てに依存します。リーフノードの割当量は、そのノードに関連付けられているすべてのジョブの間で分配されます。

ジョブの最終的なシステム資源利用資格は、基本割当ツリーから得られる利用資格と、締め切り優先あるいは業務優先ポリシーなどから得られる他の利用資格を組み合わせることによって決定されます。基本割当ツリーには、基本割当スケジューリングに対する全チケットが割り当てられます。この数によって、4 通りあるスケジューリングポリシーにおける基本割当スケジューリングの重みが決まります。

基本割当ツリーは、Sun Grid Engine, Enterprise Edition のインストール中に定義し、いつでも変更することができます。基本割当ツリーを編集すると、次のスケジューリングで新しい割り当てが有効になります。

▼ QMON から基本割当ポリシーを編集する

1. QMON の「チケット概要」ダイアログボックスの株にある「基本割当ポリシー」ボタンをクリックします。

   図 9-13 に示すような「基本割当ポリシー」ダイアログボックスが表示されます。

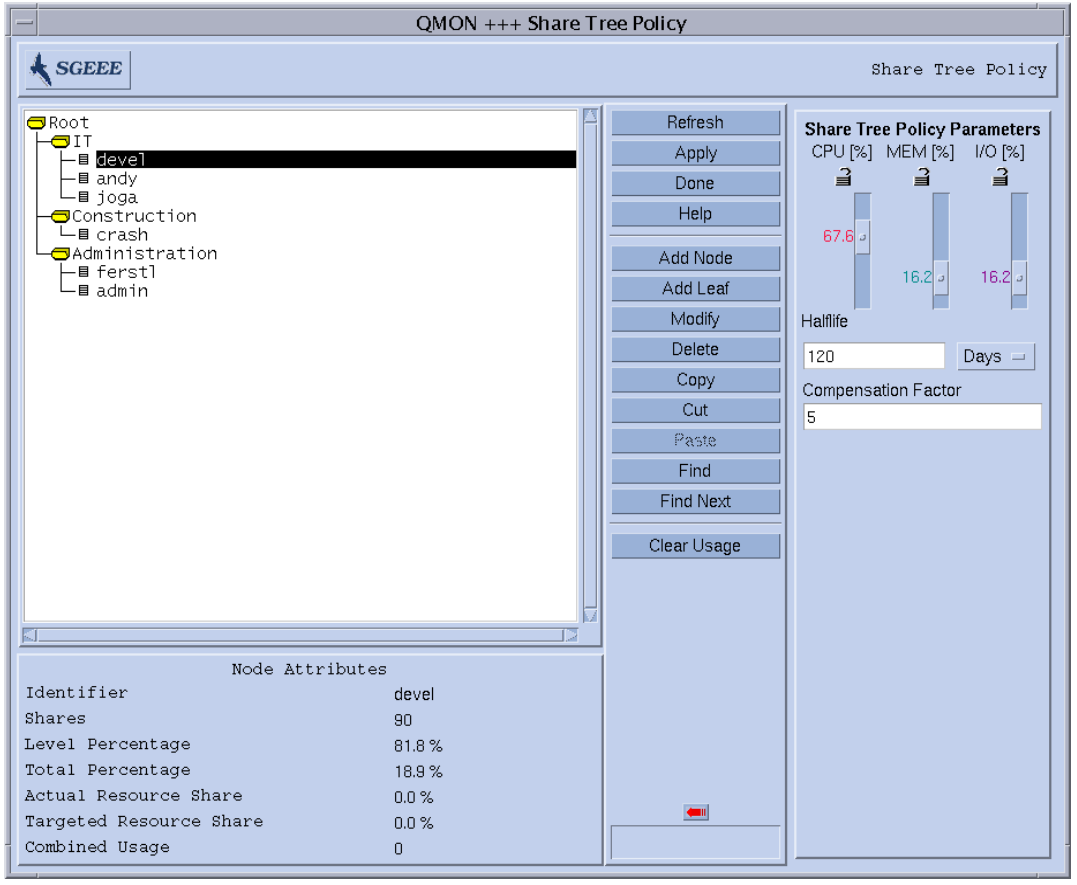


図 9-13　「基本割当ポリシー」ダイアログボックス

2. 以下の説明に従ってポリシーの編集に進みます。

ノード属性

この区画は、選択されているノードの属性を表示します。

- **識別名** - ユーザーかプロジェクト、またはその集団の名前です。
- **配分** - このユーザーまたはプロジェクトに割り当てられている配分量です。

注 – 配分は相対的な重要性を定義します。百分率値ではありません。また、数量的な意味もありません。相対的な重要性を細かく調整できるため、一般には、百または千台の数字を使用することを推奨します。

- **同階層での割合 (%)** - このツリーの同じ親ノードレベルでの配分合計に対して、このノードが占める割合です。
- **全体での割合 (%)** - 基本割当ツリー全体の配分合計に対してこのノードが占める割合です。これが、基本割当ポリシーにおけるノードの長期資源配分目標になります。
- **実際の資源利用** - 累積期間中にこのノードがこれまでに消費した、システムの全資源に対する割合です。この割合は、基本割当ツリーのすべてのノードに対する割合です。
- **資源利用目標** - 上記と同じですが、基本割当ツリーで現在アクティブなノードだけが考慮されます。アクティブなノードとはシステムにジョブがあるノードです。短期には、Sun Grid Engine, Enterprise Edition はアクティブなノードの間で利用資格のバランスをとろうとします。
- **利用合計** - このノードの利用量の合計です。利用合計は、このノードで累積された利用量の合計です。リーフノードは、その下で実行されるすべてのジョブの利用量を累積します。内部ノードは、すべての子孫ノードの利用量を累積します。利用合計は、「基本割当ポリシー」ダイアログボックスで指定された割合に従って CPU やメモリー、 入出力利用から構成され、同じダイアログボックスで指定された半減期減少率で減少します。

リーフノードのユーザー / プロジェクトノードを削除して、同じ場所または別の場所に追加し直しても、その利用量は残ります。ユーザー / プロジェクトノードを追加し直す前に利用量をゼロにするには、Sun Grid Engine, Enterprise Edition で構成したユーザー / プロジェクトからそのユーザー / プロジェクトを削除してから、追加し直す必要があります。

基本割当ツリーに登録されていないユーザー / プロジェクトがジョブを実行していて、そのユーザー / プロジェクトを基本割当ツリーに追加した場合、その利用量はゼロ以外の値になります。ここでもまた、ツリーに追加する前にそのユーザー / プロジェクトの利用量をゼロにするには、Sun Grid Engine, Enterprise Edition で構成したユーザー / プロジェクトからそのユーザー / プロジェクトを削除してから、基本割当ツリーに追加します。

再表示

QMON グラフィカルユーザーインタフェースは、その表示情報を定期的に更新します。このボタンは、そうした再表示をただちに行います。

適用

このボタンをクリックすると、行われた追加や削除、ノード変更のすべてが適用されます。ウィンドウは開いたままです。

完了

このボタンをクリックすると、ウィンドウが閉じます。それまでに行われた追加や削除、ノード変更は適用されません。

ヘルプ

このボタンをクリックすると、オンラインヘルプが開きます。

ノードを追加

選択したノード内にノードを追加するには、このボタンをクリックします。空の「ノード情報」ダイアログボックスが開き、ノード名や配分を入力することができます。このノード名や配分は自由に指定することができます。

リーフを追加

選択したノードの下にリーフノードを追加するには、このボタンをクリックします。空の「ノード情報」ダイアログボックスが開き、ノード名や配分を入力することができます。このノード名は、既存の Sun Grid Engine, Enterprise Edition ユーザー名 (230 ページの「QMON からユーザーオブジェクトを構成する」) か、Sun Grid Engine, Enterprise Edition プロジェクト名 (233 ページの「プロジェクト」) である必要があります。

以下の規則が適用されます。

- 基本割当ツリー内の各ノードが固有のパスを持つこと。
- 基本割当ツリーで同じプロジェクトを複数回参照しないこと。
- プロジェクトサブツリーに同じユーザーが複数回現れないこと。
- プロジェクトサブツリーの外部に同じユーザーが 1 回しか現れないこと。
- ユーザーがリーフ以外のノードに現れないこと。
- プロジェクトサブツリー内のあらゆるリーフノードが既知のユーザーか予約名の「default」を参照すること。(この特殊なユーザーについての詳細は、259 ページの「特殊ユーザー default」の節を参照してください。)
- プロジェクトサブツリー内にサブプロジェクトを含まないこと。

- プロジェクトサブツリー以外のあらゆるリーフノードが既知のユーザーかプロジェクトを参照すること。
- プロジェクトサブツリー内のあらゆるユーザーリーフノードにプロジェクトへのアクセス権があること。

変更

選択したノードを編集するには、このボタンをクリックします。ノード名と配分の入った「ノード情報」ダイアログボックスが開きます。

削除

このボタンをクリックすると、選択されているノードとそのすべての子孫が削除されます。

コピー

このボタンをクリックすると、選択されているノードがその子孫とともにペーストバッファーにコピーされます。

カット

このボタンをクリックすると、選択されているノードとその子孫が基本割当ツリーから切り取られ、ペーストバッファーにコピーされます。

ペースト

このボタンをクリックすると、前回コピーされたノードが選択されているノードに下にペーストされます。

検索

このボタンをクリックすると、入力ボックスが開き、検索文字列を入力して、基本割当ツリー内でその文字列を含む名前を検索することができます。検索文字列は大文字と小文字が区別され、その文字列から始まる「ノード名」が表示されます。

次を検索

検索文字列に一致する名前の検索を繰り返します。

利用クリア

このボタンをクリックすると、基本割当ツリー階層構造全体で累積された利用量のすべてがゼロに戻されます。このボタンは、基本割当ポリシーを予算編成時期に合わせて、最初からやり直す必要がある場合に特に有効です。利用クリア機能は、Sun Grid Engine, Enterprise Edition テスト環境を構成したり、変更したりする際にも便利です。

大きな矢印のナビゲータ

この矢印をクリックすると、このウィンドウの「基本割当ポリシーのパラメータ」区画が開きます。

基本割当ポリシーのパラメータ

- **CPU (%) スライダ** - CPU が利用合計に占める割合を示します。このスライダを動かすと、CPU の割合の変化に合わせて MEM および I/O スライダも変化します。
- **MEM (%) スライダ** - メモリーが利用合計に占める割合を示します。このスライダを動かすと、メモリーの割合の変化に合わせて CPU および I/O スライダも変化します。
- **I/O (%) スライダ** - 入出力が利用合計に占める割合を示します。このスライダを動かすと、入出力の割合の変化に合わせて CPU および MEM スライダも変化します。

注 – CPU(%)、MEM(%)、I/O(%) の合計はつねに 100% になります。

- **錠のシンボル** - 錠が開いていると、対応するスライダが自由に動くことができます。直接操作されることによって動くことも、別のスライダが動かされたために、動くこともあります。

  錠が閉じていると、対応するスライダは動きません。2 つの錠が閉じていて 1 つが開いている場合は、どのスライダも動かせません。
- **半減期** - この入力フィールドを使用して、資源利用に対する半減期を指定します。資源利用量は、スケジューリングのたびに減少します。累積利用量に関わる資源利用があると、半減期の経過後にその値が半分になります。
- **「日数 / 時間数」選択メニュー** - 半減期の単位を日数または時間数のどちらにするか選択します。
- **補正係数** - 正の整数値の補正係数を受け付ける入力フィールドです。適切な値は 2 ～ 10 の範囲です。

  補正係数は、実際の利用が利用目標をずっと下回っているユーザー / プロジェクトが、初めて資源を取得するときに資源を優先使用するのを防ぎます (上記の説明を参照)。

特殊ユーザー default

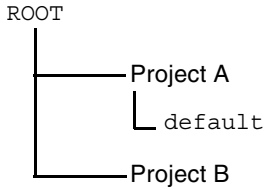
多数のユーザーがいるサイトでは、ユーザー default を使用して、基本割当ツリーを保守する作業を減らすことができます。この特殊ユーザーは、基本割当ツリーにおいてユーザーが Sun Grid Engine, Enterprise Edition プロジェクトの下位に位置する、いわゆる「混成」基本割当ツリー、あるいは同じプロジェクトの大部分のユーザーに同じ資源利用資格 (同等の割り当てスケジューリング) を割り当てる場合にのみ利用することができます。

ユーザー default は、基本割当ツリー内のプロジェクトノード (既存の Sun Grid Engine, Enterprise Edition プロジェクトノード) の下のリーフノードとしてのみ現れることができます。default リーフノードがあると、対応するプロジェクトノードの下位に既存のすべて Sun Grid Engine, Enterprise Edition ユーザーエントリを構成するものと解釈され、同量の配分が付与されます。そうしたプロジェクトにアクセスして、ジョブの実行依頼をするユーザーはすべて、default ユーザーエントリに設定されたのと同じ利用資格を受けます。特定のユーザーに対してこの機能を有効にするには、Sun Grid Engine, Enterprise Edition システムユーザーリストにそのユーザーを追加する必要があります。

default ユーザーの短期利用資格は、消費する資源量がそれぞれに異なるため互いに異なることに注意してください。ただし、長期利用資格は同じです。

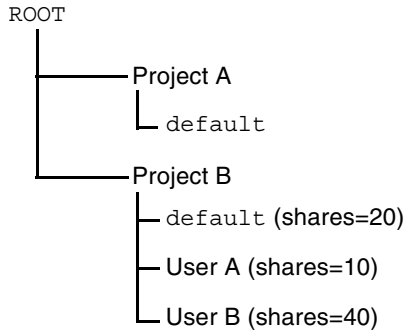
一部のユーザーにだけ特別な (低いまたは高い) 利用資格を割り当てて、他のすべてのユーザーには同じ長期利用資格を維持する場合は、default ユーザーと同レベルに、その特別な資格を持つ個々のユーザーのエントリを含む基本割当ツリーを構成することができます。

以下を例 A とします。



例 A では、プロジェクト A に実行依頼するすべてのユーザーが同じ長期利用資格を得るのに対し、プロジェクト B に実行依頼するユーザーは単にそのプロジェクトの累積資源消費量に関係します。プロジェクト B のユーザーの利用資格は管理されません。

例 A を例 B と比較してみてください。



例 B のプロジェクト A に対する扱いは、例 A のときと同じです。しかし、プロジェクト B では、そこに実行依頼する大部分のユーザーが同等の長期利用資格を得るものの、ユーザー A の受ける利用資格はそれら大部分のユーザーの半分であり、ユーザー B は 2 倍の利用資格を得ます。

▼ コマンド行から基本割当ポリシーを構成する

注 – 基本割当ツリーの構成には、QMON を使用することを推奨します。階層形式のツリーは本来グラフィカル表示と編集によく適しています。ただし、たとえばシェルスクリプトから基本割当ツリーの変更を行う必要が生じた場合は、qconf コマンドとそのオプションを利用して、そのようにすることもできます。

- **以下のガイドラインに従って qconf コマンドを使用してください。**
  - qconf の -astree、-mstree、-dstree、-sstree は、それぞれ新規基本割当ツリー全体の追加、既存の基本割当ツリー構成の変更、基本割当ツリーの削除、基本割当ツリー構成を表示するためのオプションです。これらのオプションについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qconf 項を参照してください。share_tree マニュアルページに、基本割当ツリー構成の形式に関する説明があります。
  - qconf の -astnode、-mstnode、-dstnode、-sstnode オプションは、基本割当ツリー全体ではなく、単一のノードだけを扱うオプションです。ディレクトリのパス同様、ノードは親ノードから基本割当ツリーを下方向にたどったパスとして参照できます。上記のオプションでは、それぞれノードを追加、変更、削除、表示することができます。ノードに含まれる情報は、ノード名と割り当てられている配分で構成されます。
  - CPU、メモリー、入出力資源利用の重み付けパラメータと半減期、補正係数は、それぞれスケジューラ構成で usage_weight_list、halftime、compenstation_factor を使用して設定します。. スケジューラ構成は、コマンド行から qconf の -msconf および -ssconf オプションを使用してアクセスで

きます。形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の sched_conf 項を参照してください。

業務優先ポリシー

プライオリティスケジューリングと呼ばれることもある業務優先スケジューリングは、実行依頼するユーザー、プロジェクト、部署、ジョブクラスとの関係でジョブの重要性を決定する、フィードバックのないスケジューリング方式です。ジョブの最終的なシステム資源利用資格は、業務優先ポリシーから得られる利用資格と、締め切り優先あるいは基本割当ポリシーなどから得られる他の利用資格を組み合わせることによって決定されます。

4 通りあるスケジューリングポリシーにおける業務優先ポリシーの重みは、業務優先ポリシーに割り当てられたチケットの合計によって決まります。管理者は、Sun Grid Engine, Enterprise Edition のインストール中に業務優先チケット全体をユーザー、部署、プロジェクト、ジョブ、ジョブクラスという業務カテゴリに振り分けます。

業務優先配分

業務優先配分は、業務カテゴリ (ユーザー、部署、プロジェクト、ジョブ、ジョブクラス) のあらゆるメンバーに割り当てられます。そして、それらの配分は、カテゴリのメンバーに関連付けられている各ジョブが受ける資格を持つ、そのカテゴリ用のチケット全体に占める割合を示します。ユーザー davidson が 200、ユーザー donlee が 100 の配分の場合、davidson が実行依頼するジョブは donlee のジョブよりも 2 倍多くの user-functional-tickets を得ることができます (200、100 というのはチケット数ではない)。

各カテゴリに割り当てられた業務優先チケットは、そのカテゴリに関連付けられているすべてのジョブ間で分配されます。

share_functional_shares パラメータ

業務優先ポリシーでは、業務カテゴリのユーザー、プロジェクト、部署、ジョブクラス (キュー)、ジョブに対する利用資格の配分、そしてその各カテゴリのすべてのメンバーに対する配分を定義します。この意味で業務優先ポリシーは 2 階層の基本割当ツリーに似ていますが、1 つのジョブを同時に複数のカテゴリに関連付けられるという違いがあります。たとえば特定のユーザーのジョブが、プロジェクトや部署、ジョブクラスに属してもかまいません。

ただし、基本割当ツリー同様、業務カテゴリからジョブが受ける配分資格は、そのジョブのカテゴリメンバー (たとえば、その特定のプロジェクト) に定義されている配分と、そのカテゴリ (プロジェクトとユーザー、部署など) そのものに割り当てら

れている配分によって決まります。 share_functional_shares パラメータ (クラスタ構成内の schedd_params にある) は、ジョブの配分の決定にカテゴリメンバーの配分をどのように反映させるかを定義します。カテゴリメンバー (たとえば特定のユーザーまたは特定のプロジェクト) に割り当てられた配分をすべてのジョブに繰り返すことも、カテゴリメンバーのジョブの間で配分を振り分けることもできます。

- share_functional_shares=false は繰り返しを意味します。
- share_functional_shares=true は分配を意味します。

こうした配分は株式に例えることができます。同じカテゴリメンバーに属するジョブには何の意味もなく、どちらの場合も、同じカテゴリメンバーのすべてのジョブは同じ量の配分を受けます。しかし、メンバーではなく、同じカテゴリ内の配分量の比較では、配分数は意味を持ちます。share_functional_shares が true に設定されている場合、関係する多数のジョブが同じカテゴリメンバーに属するジョブは、その配分のうちの比較的小さい配分を受けることになります。
share_functional_shares が false の場合は、そうはならずに、関係するすべてのジョブがそのカテゴリメンバーと同じ配分量を受けます。

システムに存在するジョブ数に関係なく、カテゴリメンバーがそのジョブ全体に対して一定のレベルの役割優先資格を受けるようにする場合は、
share_functional_shares=true を使用します。その場合、関係するジョブが多数ある場合、個々のジョブが受ける資格はごくわずかになるかもしれません。システムに存在する関係ジョブ数に関係なく、そのカテゴリメンバーの資格に基づいて各ジョブに同じレベルの資格を与える場合は、share_functional_shares=false を使用します。この場合、多数のジョブを抱えるカテゴリメンバーは、業務優先ポリシーで圧倒的な地位を占めることになるかもしれないことに注意してください。

役割優先チケットの全体の量が、share functional shares の設定で左右されることはありません。この全体量は、つねに、業務優先ポリシーチケットプールに管理者が定義した量です。share functional shares パラメータは、単に業務優先ポリシー内での業務優先チケットの分配方法に影響するだけです。

▼ QMON から業務優先ポリシーを構成する

1. QMON の「チケット概要」ダイアログボックスの下部にある「業務優先ポリシー」ボタンをクリックします。

   図 9-14 に示すような「業務優先ポリシー」ダイアログボックスが表示されます。

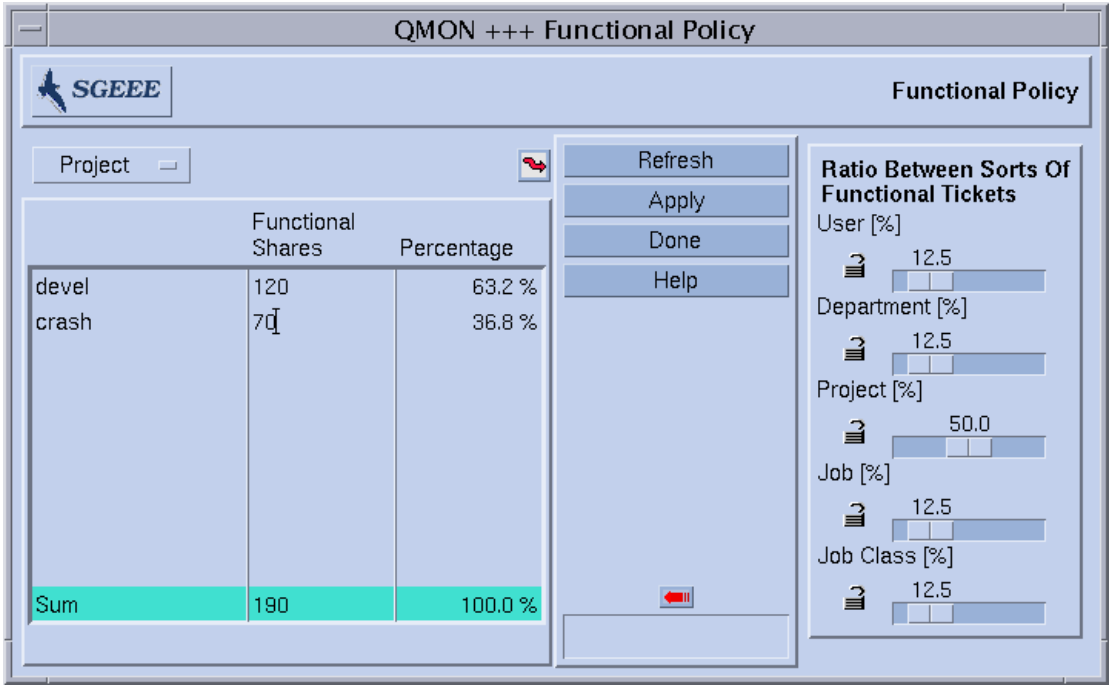


図 9-14　「業務優先ポリシー」ダイアログボックス

2. 以下の適切な節に進みます。

カテゴリ選択メニュー

業務優先配分を定義する業務カテゴリを選択します。業務カテゴリは、ユーザー、プロジェクト、部署、ジョブ、ジョブクラス (キューで定義) のいずれかです。

メンバー情報の表示

スクロール可能な区画で、以下を表示します。

- 業務優先配分を定義するカテゴリ (ユーザー、プロジェクト、部署、ジョブ、ジョブクラスのいずれか) のメンバーのリスト

- カテゴリの各メンバーの業務優先配分数。配分は、業務カテゴリの各メンバーの相対的な重要性を表します。このフィールドは編集可能です。
- このカテゴリの業務優先チケット (ユーザー、ユーザーセットなど) 全体に占める、業務優先配分数が表すチケット割当量の割合。このフィールドは編集できません。パーセント値がフィードバックされます。

ねじれ矢印のナビゲータ

この矢印をクリックすると、構成ダイアログボックスが開きます。

- ユーザーカテゴリの場合は、「ユーザー構成」ダイアログボックスが開きます。「ユーザー」タブを使用して、適切な構成変更モードに切り替え、Sun Grid Engine, Enterprise Edition ユーザーの構成を変更することができます。
- 部署カテゴリの場合も、「ユーザー構成」ダイアログボックスが開きます。「ユーザーセット」タブを使用しして、適切な構成変更モードに切り替え、Sun Grid Engine, Enterprise Edition ユーザーセットで表される部署の構成を変更することができます。
- プロジェクトカテゴリの場合は、「プロジェクト構成」ダイアログボックスが開きます。
- ジョブカテゴリの場合は、「ジョブ制御」ダイアログボックスが開きます。
- ジョブクラスカテゴリの場合は、「キュー制御」ダイアログボックスが開きます。

再表示

QMON グラフィカルユーザーインタフェースは、その表示情報を定期的に更新します。このボタンは、そうした再表示をただちに行います。

適用

このボタンをクリックすると、行われた追加や削除、ノード変更のすべてが適用されます。ウィンドウは開いたままです。

完了

このボタンをクリックすると、ウィンドが閉じます。変更は適用されません。

ヘルプ

このボタンをクリックすると、オンラインヘルプが開きます。

大きな矢印のナビゲータ

この矢印をクリックすると、このウィンドウの**「業務優先チケットのカテゴリ別比率」**区画が開きます。

業務優先チケットのカテゴリ別比率

ユーザー (%)、部署 (%)、プロジェクト (%)、ジョブ (%)、ジョブクラス (%) の合計はつねに 100% になります。

ユーザー *(%)* スライダ

このスライダの設定は、業務優先チケット全体からユーザーカテゴリに割り当てる割合を示します。このスライダを動かすと、ユーザーの割合の変化に合わせて、ロックされていない他のスライダも変化します。

部署 *(%)* スライダ

このスライダの設定は、業務優先チケット全体から部署カテゴリに割り当てる割合を示します。このスライダを動かすと、部署の割合の変化に合わせて、ロックされていない他のスライダも変化します。

プロジェクト *(%)* スライダ

このスライダの設定は、業務優先チケット全体からプロジェクトカテゴリに割り当てる割合を示します。このスライダを動かすと、プロジェクトの割合の変化に合わせて、ロックされていない他のスライダも変化します。

ジョブ *(%)* スライダ

このスライダの設定は、業務優先チケット全体のうちのジョブカテゴリに割り当てる割合を示します。このスライダを動かすと、ジョブの割合の変化に合わせて、ロックされていない他のスライダも変化します。

ジョブクラス *(%)* スライダ

このスライダの設定は、業務優先チケット全体からジョブクラスカテゴリに割り当てる割合を示します。このスライダを動かすと、ジョブクラスの割合の変化に合わせて、ロックされていない他のスライダも変化します。

錠のシンボル

錠が開いていると、対応するスライダが自由に動くことができます。直接操作されることによって動くことも、別のスライダが動かされたために、動くこともあります。

錠が閉じていると、対応するスライダは動きません。

4 つの錠が閉じていて 1 つが開いている場合は、どのスライダも動かせません。

▼ コマンド行から業務優先ポリシーを構成する

- **以下のガイドラインに従って `qconf` コマンドを使用してください。**
  - ユーザーカテゴリの場合は、`qconf -muser` コマンドを使用して `fshare` パラメータを変更します (`user` ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - 部署カテゴリの場合は、`qconf -mu` コマンドを使用して `fshare` パラメータを変更します (部署を表す `access_list` ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - プロジェクトカテゴリの場合は、`qconf -mprj` コマンドを使用して `fshare` パラメータを変更します (`project` ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - ジョブクラスカテゴリの場合は、`qconf -mq` コマンドを使用して `fshare` パラメータを変更します (ジョブクラスを表す `queue` ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - これらのカテゴリの重みは、スケジューラ構成の `sched_conf` で定義し、`qconf -msconf` で変更することができます。変更するパラメータは `weight_user`、`weight_department`、`weight_project`、`weight_job`、`weight_jobclass` です。これらのパラメータの値の範囲は 0 から 1 で、合計で 1 になる必要があります。

注 – ジョブへは、`QMON` でしか業務優先配分を割り当てられません。現在のところ、コマンド行インタフェースにこの機能はありません。

締め切り優先ポリシー

締め切り優先スケジューリングは、締め切りに間に合うようにジョブをただちに開始し、十分な資源を与えることによって、特定の日時までにジョブが完了するようにします。締め切り優先ジョブを実行依頼する場合は、次の情報を指定します。

- **開始時間** - ジョブが実行対象になる時間です。通常、開始時間はジョブの実行依頼直後ですが、QMON の「ジョブ実行依頼」ダイアログボックスの**「開始時間」**パラメータまたは qsub の -a オプションを使用して遅らせることもできます (詳細は、71 ページの「ジョブの実行依頼」を参照)。
- **締め切り優先時間** - ジョブの重要性が最高度に達して、そのジョブが受ける資格のある締め切り優先チケットのすべてを取得し、最大のシステム資源配分を受ける時間です。実行依頼するユーザーは、ジョブの実際の締め切りに間に合わせるのに締め切り優先時間が適切であるかどうかを判断する必要があります。

締め切り優先チケット

Sun Grid Engine, Enterprise Edition は、締め切り優先時間の前に低いレベルの重要性で締め切り優先ジョブを開始することによって、使用可能なシステム資源を利用することができます。締め切り優先ジョブは、締め切り優先時間に近づくに従って自動的に追加のチケットを受け取ります。締め切り優先ジョブに与えられる締め切り優先チケットは、ジョブが実行対象になってから締め切り優先時間に達するまでリニアに増加していきます。複数の締め切り優先ジョブが締め切り優先時間に達する場合、締め切り優先チケットは、その時間に基づいてそれぞれのジョブに比例配分されます。

share_deadline_tickets パラメータ

管理者は、締め切り優先ポリシーに特定の数のチケットを割り当てます。締め切り優先ジョブのそれぞれに割り当てられるチケット数は、このポリシー全体のチケット数と、実行依頼時間から締め切り優先時間までの間のジョブの相対的な時間位置によって決まります。share_deadline_tickets パラメータ (クラスタ構成の schedd_params にある) は、締め切り優先ジョブに割り当てる締め切り優先チケット数の計算に影響を及ぼす 3 つ目の要素です。

share_deadline_tickets=true の設定では、締め切り優先ポリシーに割り当てられたチケット全体が、すべての締め切り優先ジョブ間で分配され、その後、それぞれのジョブのチケット分は、その締め切り優先時間までの位置に応じて減少します。share_deadline_tickets=false の設定では、どの締め切り優先ジョブもその締め切り優先時間に到達すると、締め切り優先ポリシーに割り当てられているチケット全部を取得し、締め切りに近づくのに比例してチケットが減少します。

締め切り優先ポリシーで分配するチケット量全体を制御する場合 (特に分配できるチケットが一定量しかない基本割当ポリシーや業務優先ポリシーとの関係で制御する場合) は、share_deadline_tickets=true を使用してください。ただし、この場

合、多数の締め切り優先ジョブが同時にシステムにあると、1 つのジョブに割り当てられるチケット量は締め切りに近づくにつれて非常に少量になる可能性があることに注意してください。

他のポリシーに用意されているチケットプールとの関係で個々の締め切り優先ジョブの重要性を制御する場合は、`share_deadline_tickets=false` を使用します。この設定では、システムに存在する締め切り優先ジョブ数は重要ではありません。どのジョブもつねに締め切り優先チケットの最大量を得ることができます。ただし、システムに多数の締め切り優先ジョブがある場合、他のポリシーの重要性が低下する可能性があることに注意してください。

締め切り優先チケットの構成

システム管理者は、締め切り優先ジョブに割当可能な締め切り優先チケットの最大数を設定します。この数によって、4 つあるポリシー間の締め切り優先スケジューリングの重みが決まります。「チケット概要」ダイアログボックス (図 9-12 を参照) を使用して最大数を設定してください。このダイアログボックスには、システムでアクティブな締め切り優先チケット数も表示されます。

Deadlineusers の構成

クラスタ構成では、締め切り優先ジョブの実行依頼を許可するユーザーに関するポリシーも制御します。締め切り優先チケットは、ユーザーアクセスリストに登録されているユーザー (deadlineusers) にのみ付与されます。図 9-15 は、「ジョブ実行依頼」ダイアログボックスの締め切り優先の表示例です。

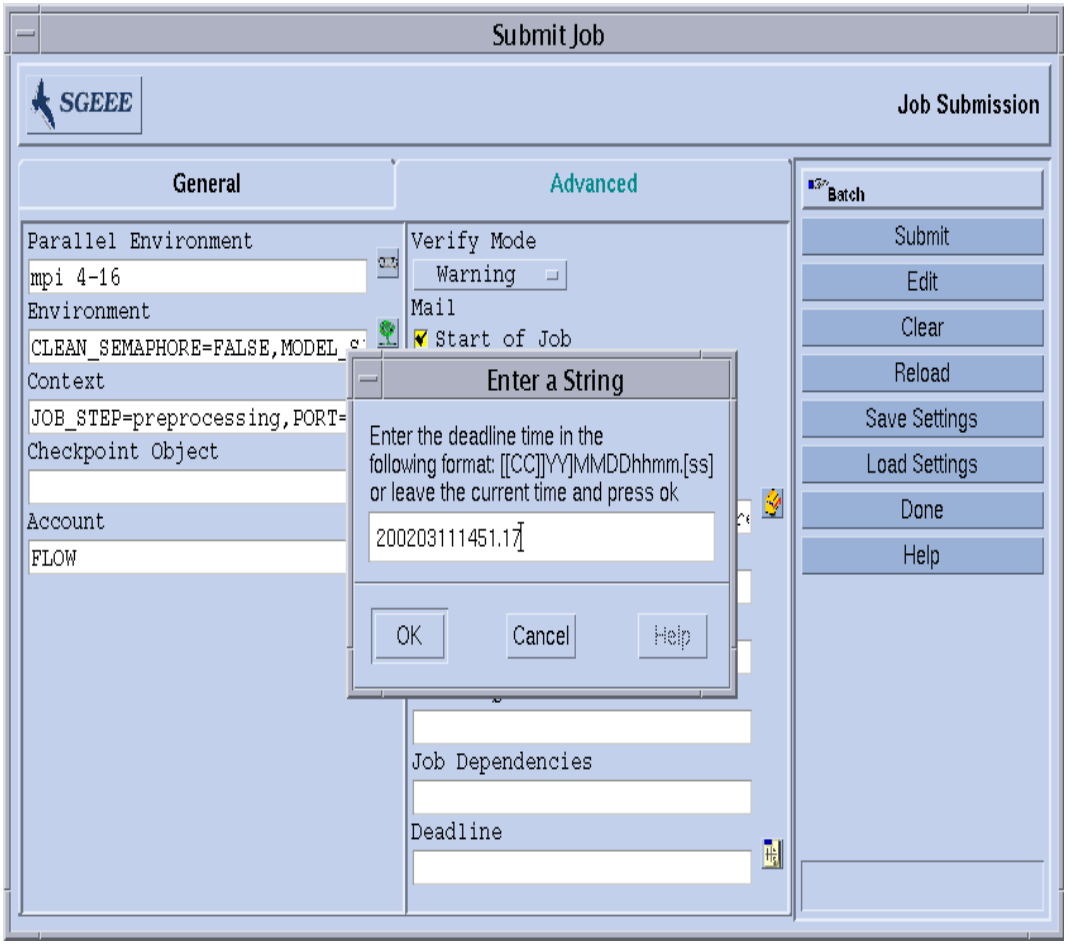


図 9-15　「ジョブ実行依頼」ダイアログボックスの締め切り優先の表示

コマンド行から `qsub` の `-dl` オプションを使用して、Sun Grid Engine, Enterprise Edition システムに締め切り優先ジョブを渡すこともできます。ダイアログボックスの実行依頼方法についての詳細は、第 4 章を参照してください。

一時優先ポリシー

一時優先スケジューリングでは、Sun Grid Engine, Enterprise Edition のマネージャーまたはオペレータは、ジョブ、ユーザー、部署、プロジェクト、ジョブクラスにチケットを追加することによって、個々のジョブまたはユーザー、部署、プロジェクト、ジョブクラスに関連付けられているすべてのジョブの相対的な重要性を動的に調整することができます。一時優先チケットを追加すると、ユーザー、部署、プロジェクト、ジョブクラス、またはジョブが受け取るチケットの合計数、すなわち、それらが全体として受け取る資源配分が増加します。

また、システム全体のチケットの合計数も増加します。チケットの追加によって、あらゆるジョブのチケットの価値が低下します。

一時優先チケットは、主として 2 通りの用途を意図しています。

- 基本割当や業務優先、締め切り優先などのポリシーの構成を変更しないで、それらの自動的なチケット割当ポリシーに一時的に優先する指定をする。
- 一定量のチケットからなる資源利用資格レベルを設ける。ジョブをレベル分けしたり (大 / 中 / 小)、優先クラスなどを設けたりする場合に適しています。

ジョブに直接割り当てられた一時優先チケットは、そのジョブが完了すると消滅し、他のすべてのチケットは元の価値に戻ります。ユーザー、部署、プロジェクト、ジョブクラスに割り当てられた一時優先チケットは、管理者が明示的に削除しない限り、システムに留まります。

図 9-12 の「チケット概要」画面には、システムで現在アクティブな一時優先チケット数も示されています。

注 – 不必要になった時点でオペレータが明示的に削除しなかった場合、一時優先エントリは「一時優先」ダイアログボックスに残り、以降の運用に影響が出ることがあります。

---

share_override_tickets パラメータ

管理者は、ユーザー、プロジェクト、部署、ジョブクラス、ジョブという一時優先カテゴリのさまざまなメンバーにチケットを割り当てます。「ジョブ」カテゴリを除き、このことは、特定のカテゴリメンバーの個々のジョブに割り当てられるチケット値は、そのメンバーに定義されているチケット量によって決まることを意味します。たとえばユーザー A に付与されているチケット数によって、ユーザー A のすべてのジョブに割り当てられるチケット数が決まります。

share_override_tickets パラメータ (クラスタ構成の schedd_params にある) は、カテゴリメンバーのチケット値からジョブのチケット値を得る方法を制御します。share_override_tickets=true の設定では、カテゴリメンバーのチケットはそのメンバーのすべてのジョブの間で平等に分配されます。

share_override_tickets=false の設定では、それぞれのジョブはそのカテゴリメンバーに定義されているチケット量を継承します。すなわち、カテゴリメンバーのチケットが、そのジョブのすべてのジョブに繰り返されます。

一時優先ポリシーで分配するチケット量全体を制御する場合 (特に分配できるチケットが一定の量しかない基本割当ポリシーや業務優先ポリシーとの関係で制御する場合) は、share_overide_tickets=true を使用してください。ただし、share_override_tickets を true に設定していて、1 つのカテゴリメンバー (たとえば特定のユーザー) に多数のジョブがある場合、個々のジョブに割り当てられるチケットはごく少量になる可能性があることに注意してください。

他のポリシーと一時優先カテゴリに用意されているチケットプールとの関係で個々の一時優先ジョブの重要性を制御する場合は、share_override_tickets=false を使用します。この設定では、システムに存在する一時優先ジョブ数は重要ではありません。この場合は、ジョブが受けるチケット量はつねに同じで、一時優先チケットを受け取る権利を持つジョブが多いほど、システム内の一時優先チケットの合計数は多くなります。このため、他のポリシーの重要性が低下する可能性があります。

▼ QMON から一時優先ポリシーを構成する

1. **「チケット概要」ダイアログボックスから「一時優先ポリシー」をクリックします。**
   図 9-16 に示すような「一時優先ポリシー」ダイアログボックスが表示されます。

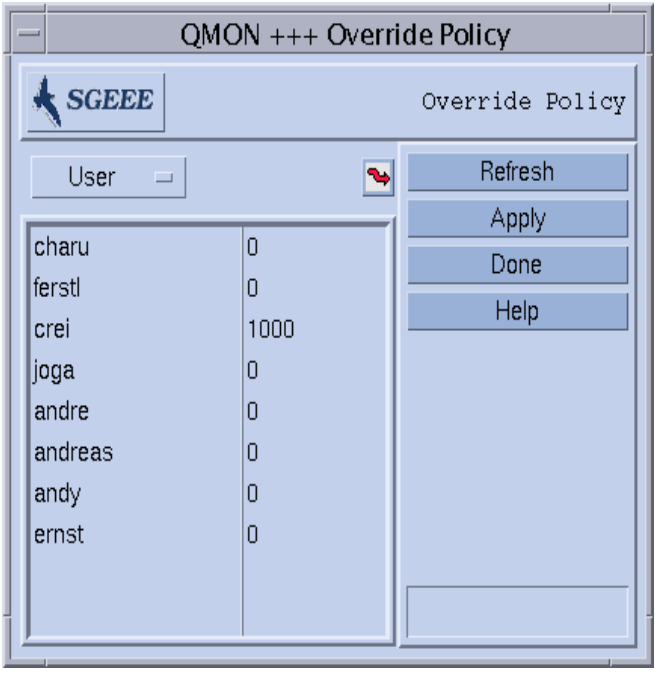


図 9-16 「一時優先ポリシー」ダイアログボックス

2. **以下の説明に従ってジョブ、ユーザー、部署、プロジェクト、またはジョブクラスに一時優先チケットを割り当てます。**

カテゴリ選択メニュー

一時優先チケットを定義するカテゴリ (ユーザー、プロジェクト、部署、ジョブ、ジョブクラスのいずれか) を選択します。

メンバー情報の表示

スクロール可能な区画で、以下を表示します。

- 一時優先チケットを定義するカテゴリ (ユーザー、プロジェクト、部署、ジョブ、ジョブクラスのいずれか) のメンバーのリスト

- カテゴリの各メンバーの一時優先チケット数 (整数)。このフィールドは編集可能です。

ねじれ矢印のナビゲータ

この矢印をクリックすると、構成ダイアログボックスが開きます。

- ユーザーカテゴリの場合は、「ユーザー構成」ダイアログボックスが開きます。「ユーザー」タブを使用して、適切な構成変更モードに切り替え、Sun Grid Engine, Enterprise Edition ユーザーの構成を変更することができます。
- 部署カテゴリの場合も、「ユーザー構成」ダイアログボックスが開きます。「ユーザーセット」タブを使用しして、適切な構成変更モードに切り替え、Sun Grid Engine, Enterprise Edition ユーザーセットで表される部署の構成を変更することができます。
- プロジェクトカテゴリの場合は、「プロジェクト構成」ダイアログボックスが開きます。
- ジョブカテゴリの場合は、「ジョブ制御」ダイアログボックスが開きます。
- ジョブクラスカテゴリの場合は、「キュー制御」ダイアログボックスが開きます。

再表示

QMON グラフィカルユーザーインタフェースは、その表示情報を定期的に更新します。このボタンは、そうした再表示をただちに行います。

適用

このボタンをクリックすると、行われた追加や削除、ノード変更のすべてが適用されます。ウィンドウは開いたままです。

完了

このボタンをクリックすると、ウィンドウが閉じます。それまでに行われた追加や削除、ノード変更は適用されません。

ヘルプ

このボタンをクリックすると、オンラインヘルプが開きます。

▼ コマンド行から一時優先ポリシーを構成する

- **以下の説明に従って** `qconf` **コマンドを使用します。**
  - **ユーザーカテゴリの場合は、**`qconf -muser` **コマンド**を使用して `oticket` パラメータを変更します (user ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - **部署カテゴリの場合は、**`qconf -mu` **コマンド**を使用して `oticket` パラメータを変更します (部署を表す access_list ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - **プロジェクトカテゴリの場合は、**`qconf -mprj` **コマンド**を使用して `oticket` パラメータを変更します (project ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。
  - **ジョブクラスカテゴリの場合は、**`qconf -mq` **コマンド**を使用して `oticket` パラメータを変更します (ジョブクラスを表す queue ファイル形式についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。

注 – ジョブへは、`QMON` でしか一時優先チケットを割り当てられません。現在のところ、コマンド行インタフェースにこの機能はありません。

ポリシー階層

ポリシー階層は、特に保留中のジョブに関するポリシー間の衝突の問題を解決する手段を提供します。この種の問題は、基本割当ポリシーあるいは業務優先ポリシーとの組み合わせでいくつか発生するケースがあります。この 2 つのポリシーにはともに、同じリーフレベルのエンティティに属するジョブは、それらに割り当てられている優先順位 (配分を受ける資格) を基準に FIFO 順になるという特徴があります。リーフレベルのエンティティとは、基本割当ツリーではユーザー / プロジェクトリーフ、業務優先ポリシーでは、「ジョブ」カテゴリを除いて業務優先カテゴリの任意の「メンバー」(特定のユーザー、プロジェクト、部署、キューのいずれか) です。このため、たとえば同じユーザーの複数のジョブの優先順位は、最初のジョブが最高、2 番目のジョブがその次というようになります。

衝突が発生する可能性があるのは、別のポリシーでこれとは異なる優先順位が定義されている場合です。たとえば一時優先ポリシーで、3 番目のジョブが最重要で、最初に実行依頼されたジョブの重要性は最低、というように定義されているケースです。

ポリシー階層は、基本割当ツリーあるいは業務優先ポリシーよりも前に一時優先ポリシーを位置付けることによって、基本割当ツリーの同じリーフレベルのエンティティ(ユーザーまたはプロジェクト) である限り、一時優先ポリシーで最も重要なジョブが、基本割当 / 業務優先ポリシーで最高の利用資格を得るようにします。

policy_hierarchy パラメータ (クラスタ構成の schedd_params にある) には、4 つあるポリシーの先頭文字 (基本割当の S、業務優先の F、締め切り優先の D、一時優先の O) を 4 文字まで組み合わせ、こうして、最初の文字が最上位のポリシー、最後の文字が最下位のポリシーというポリシー階層を作成することができます。ポリシー階層にないポリシーは、階層に影響しません。ただし、そのポリシーも依然ジョブのチケットの供給源になります。そうしたポリシーのチケットは、他のポリシーのチケット計算で考慮されませんが、各ジョブの総合的な利用資格の定義に際しては、すべてのポリシーのすべてのチケットが合計されます。

以下では、2 通りの設定を例に、保留中のジョブがどのような影響を受けるのかを説明します。

```
policy_hierarchy=OS
```

- 最初に、一時優先ポリシーに従って、保留中の各ジョブに適切な数のチケットが割り当てられます。
- 2 つのジョブが同じユーザーまたは同じリーフレベルのプロジェクトに属している場合、このチケット数は基本割当ツリーの利用資格の割り当てに影響します。保留中のジョブに対する基本割当ツリーのチケットが計算されます。
- 一時優先ポリシーと基本割当ポリシーのチケットと、ポリシー階層にない他のすべてのアクティブなポリシーのチケットとが合計されます。チケット数の最も多いジョブが最高の利用資格を受けます。

```
policy_hierarchy=DO
```

- 保留中のすべての締め切り優先ジョブの締め切り優先チケットが計算されます。
- 一時優先ポリシーに従って、保留中の各ジョブに適切な数のチケットが割り当てられ、締め切り優先と一時優先ポリシーのチケットが合計されます。
- 2 つのジョブが同じ業務優先カテゴリメンバーに属している場合、このチケット合計値は業務優先ポリシーの利用資格の割り当てに影響します。すなわち、合計値に基づいて、保留中のジョブに対する業務優先チケットが計算されます。この計算結果が、締め切り優先および一時優先ポリシーのチケット数に加算されます。
- 2 つのジョブが同じユーザーまたは同じリーフレベルのプロジェクトに属している場合、次に、これらのチケット値は基本割当ツリーの利用資格の割り当てに影響します。保留中のジョブに対応する基本割当ツリーチケットが計算され、締め切り優先、一時優先、業務優先ポリシーのチケット集計値に加算されます。

■ チケット数の最も多いジョブが最高の利用資格を受けます。

4 つの文字はどのように組み合わせることもできますが、意味がある、あるいは現実的に妥当なのは一部の組み合わせだけです。最後の文字はつねに S または F にします。これは、上記の例で説明した特徴があるため、影響を受ける可能性があるポリシーは、この 2 つしかないためです。D と O が並ぶ場合は、動作が変わることはないため、どちらが先でもかまいません。

もっと一般には、policy_hierarchy の設定は以下の順番にすることを推奨します。

```
[O|D][O|D][S|F][S|F]
```

最初と 2 番目の文字には、影響を与える可能性があるポリシー (締め切り優先と一時優先ポリシー) だけ、3 番目と 4 番目の文字には、影響を受ける可能性があるポリシー (基本割当と業務優先ポリシー) だけ指定するようにします。

OFD などの設定も完全に有効ですが、これは OF と同じことです。OFDS などの設定も有効で、たとえば ODFS と少し異なる結果が得られますが、ODFS に比べて OFDS が求めるものというのは、かなり不自然に思われます。

パスの別名設定

Solaris および他のネットワーク UNIX 環境で NFS アクセス可能にしている場合、1 人のユーザーがいくつかのマシンに同じホームディレクトリ (あるいはその一部が同じ) を持つことはかなりよくあることです。しかし、そのすべてのマシンでホームディレクトリのパスが完全には同じでないことがあります。

たとえば NFS とオートマウンタを使用してアクセス可能なユーザーのホームディレクトリを考えてみましょう。ユーザーが NFS サーバー上に /home/foo というホームディレクトリを持っている場合、そのユーザーは、オートマウンタが動作しているインストール済みの NFS クライアントからこのパスのホームディレクトリにアクセスすることができます。しかしながら、クライアントの /home/foo は、NFS サーバーに物理的に存在する /tmp_mnt/home/foo のシンボリックリンクにすぎないことを認識することが重要です。オートマウンタはそこからディレクトリをマウントしています。

こうした状況で、ユーザーが qsub -cwd コマンド (現在の作業ディレクトリでジョブを実行) を使用して、クライアント上のホームディレクトリツリー内のどこかからジョブを実行依頼した場合、Sun Grid Engine, Enterprise Edition システムは、実行ホストで現在の作業ディレクトリを見つけられないという問題に直面する可能性があります (実行ホストが NFS サーバーの場合)。これは、qsub コマンドが実行依頼ホスト上の現在の作業ディレクトリにアクセスし、/tmp_mnt/home/foo/ (実行依頼ホ

スト上の物理的な場所) を得るためです。このパスは実行ホストに渡されますが、実行ホストが /home/foo という物理的なホームディレクトリパスを持つ NFS サーバーの場合、解決できません。

その他、これに似た問題を引き起こすケースとしては、マシンによってマウントポイントのパスが異なる 固定 (非自動マウント) NFS マウント (たとえば、あるホストでは /usr/people にホームディレクトリをマウントし、別のホストで /usr/users の下にホームディレクトリをマウントするなど)、ネットワークから利用可能なシステムへの外部からのシンボリックリンクなどがあります。

こうした問題に対する対策として、Sun Grid Engine, Enterprise Edition ソフトウェアでは、管理者およびユーザーのどちらも「パス別名設定ファイル」を構成することができます。パス別名設定ファイルは、以下の場所にあります。

- *<sge_root>*/*<cell>*/common/sge_aliases - クラスタ全体のグローバルパス別名設定ファイルです。
- $HOME/.sge_aliases - ユーザー別のパス別名設定ファイルです。

注 – クラスタ全体のグローバルパス別名設定ファイルの編集は、認定された管理者だけが行ってください。

ファイル形式

2 つのファイルのファイル形式は同じです。

- 空白行と先頭文字位置に # 記号がある行は無視されます。
- 空白行と # で始まる行以外の各行には、任意の数の空白文字またはタブで区切った 4 つの文字列が含まれる必要があります。

  最初の文字列がソースパス、2 つ目が実行依頼ホスト、3 つ目が事項ホスト、4 つ目がソース置換パスを表します。
- 実行依頼ホストおよび実行ホストのエントリは、任意のホストを意味する * 記号だけで構成することができます。

パス別名設定ファイルの解釈のされ方

パス別名設定ファイルは、次のように解釈されます。

- クラスタ全体のグローバルパス名設定ファイルが存在する場合は、qsub が物理的な現在の作業ディレクトリのパスを検索した後で、そのファイルが読み取られます。ユーザー別のパス別名設定ファイルは、グローバルファイルの最後に付加されているかのように、後で読み取られます。

- ファイルの先頭から 1 行ずつ読み取られ、必要に応じて、それらの行に指定された置換内容が保存されます。
- 置換内容が保存されるのは、実行依頼ホストのエントリが qsub コマンドの実行されるホストに一致し、ソースパスが、すでに保存されている現在の作業ディレクトリまたはソース置換パスの先頭部分の構成要素になっている場合だけです。
- 両方のファイルの読み取りを終えるとただちにと、保存されているパス別名設定情報が実行依頼されたジョブとともに渡されます。
- 実行ホストで、別名設定情報が評価されます。パス別名の実行ホストエントリが実行ホストに一致する場合は、現在の作業ディレクトリの先頭部分が、ソース置換パスに置き換えられます。この場合は現在の作業ディレクトリ文字列が変更されること、また以降のパス別名が適用する新しい作業ディレクトリパスに一致する必要があることに注意してください。

パス別名設定ファイルの例

コード例 9-1 は、上記の NFS/ オートマウンタの問題の解決に使用可能な別名設定ファイルの例です。

```
# cluster global path aliases file
# src-path    subm-host      exec-host       dest-path
/tmp_mnt/     *              *               /
```

コード例 9-1　　パス別名設定ファイルの例

デフォルト要求の構成

通常、Sun Grid Engine, Enterprise Edition システムは、ユーザーが定義した要求プロファイルを基にバッチジョブをキューに割り当てます。ユーザーは、ジョブを正しく実行するために必要な要求のプロファイルを作成し、Sun Grid Engine, Enterprise Edition スケジューラは、そのジョブの割当先として、プロファイルに定義された要求を満たすキューだけを検討します。

ジョブに要求の指定がない場合、スケジューラは、その割当先として、そのジョブのユーザーがアクセス可能なあらゆるキューを検討します。この場合、キューがアクセス可能であること以外の制約はありません。Sun Grid Engine, Enterprise Edition ソフトウェアでは、ジョブの資源要求を定義した「デフォルトの要求」を構成することができ、ユーザーが資源要求を明示的に指定しなくても、その要求を使用することができます。

デフォルト要求は、Sun Grid Engine, Enterprise Edition クラスタのユーザー全員にグローバルに構成することも、任意のユーザーに対して個人用として構成することもできます。デフォルト要求の構成は、デフォルト要求ファイルで表します。グローバル要求ファイルは *<sge_root>*/*<cell>*/common/sge_request にあり、ユーザー別の要求ファイル (.sge_request) は、ユーザーのホームディレクトリまたは qsub コマンドが実行される現在の作業ディレクトリのいずれかに置くことができます。

これらのファイルが存在する場合は、あらゆるジョブでその評価が行われます。この評価の順序は以下のとおりです。

1. グローバルデフォルト要求ファイル
2. ユーザーのホームディレクトリにあるユーザー別デフォルト要求ファイル
3. 現在の作業ディレクトリにあるユーザー別デフォルト要求ファイル

注 – ジョブスクリプトまたは qsub コマンド行で要求が指定された場合は、デフォルト要求ファイルの要求より、その指定された要求の方が優先されます (ジョブに対する明示的な資源要求方法についての詳細は、第 4 章を参照)。

注 – qsub -clear オプションを使用して、デフォルト要求ファイルによって意図に反した影響が出るのを防ぐことができます。このオプションは、それまでのすべての要求指定を廃棄します。

デフォルト要求ファイルの形式

ここでは、ローカルおよびグローバル両方のデフォルト要求ファイルの形式をまとめています。

- デフォルト要求ファイルには、任意の数の行を含むことができます。空白行と先頭文字位置に # 記号がある行は無視されます。
- 無視する行以外の各行には、任意の qsub オプションを含めることができます (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照)。1 行に複数のオプションを指定することができます。バッチスクリプトファイルとバッチスクリプトに対する引数オプションは、qsub オプションとみなされません。このため、デフォルト要求ファイルでは使用できません。
- qsub の -clear オプションは、現在評価されている要求ファイルまたは以前に処理された要求ファイルのあらゆる要求指定を廃棄します。

デフォルト要求ファイルの例

一例として、あるユーザーのローカルのデフォルト要求ファイルがコード例 9-2 のスクリプト、test.sh と同じ構成であると仮定します。

```
# Local Default Request File
# exec job on a sun4 queue offering 5h cpu
-l arch=solaris64,s_cpu=5:0:0
# exec job in current working dir
-cwd
```

コード例 9-2　　デフォルト要求ファイルの例

このスクリプトを実行するには、次のコマンドを実行します。

```
% qsub test.sh
```

この test.sh スクリプトを実行したのと同じ結果を得るには、コマンド行から直接以下のような qsub オプションを指定します。

```
% qsub -l arch=solaris64,s_cpu=5:0:0 -cwd test.sh
```

注 – qsub で実行依頼したバッチジョブ同様、デフォルト要求ファイルは、qsh で実行依頼した対話形式のジョブでも考慮されます。また、QMON から実行依頼した対話形式とバッチジョブでも、デフォルト要求ファイルが考慮されます。

アカウンティングおよび資源利用統計の収集

Sun Grid Engine, Enterprise Edition コマンドの qacct を使用して、英数字からなるアカウンティング統計を生成することができます。スイッチなしで実行された場合、qacct は、完了したすべてのジョブによって生成され、クラスタアカウンティングファイル *<sge_root>*/*<cell>*/common/accounting. に含まれている、Sun Grid Engine, Enterprise Edition クラスタのすべてのマシンに関する総利用情報を表示します。この場合 qacct は、秒単位で 3 つの時間を報告するだけです。

- REAL - ジョブの開始と終了までの時計時間
- USER - ユーザープロセスで費やされた CPU 時間
- SYSTEM - システムコールで費やされた CPU 時間

いくつかのスイッチを使用して、すべてまたは特定のキュー、あるいはユーザーなどに関するアカウンティング情報を得ることができます。たとえば、すでに完了していて、ジョブの実行依頼の `qsub` コマンドに使用されたのと同じ `-l` 構文を使用した資源要求指定に一致するすべてのジョブに関する情報を要求することができます。詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `qacct` の項を参照してください。

`qacct` のオプション、`-j` *[job_id|job_name]* は、Sun Grid Engine, Enterprise Edition システムによって保存された、`getrusage` システムコール提供情報などの資源利用情報の全体に直接アクセスすることを可能にします (『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の対応する項を参照)。

このオプションは、ジョブ ID が *[job_id]* またはジョブ名が *[job_name]* のジョブに関する資源利用の報告をします。引数なしで実行された場合は、参照したアカウンティングファイルに含まれるすべてのジョブが表示されます。ジョブ id が選択され、複数のエントリが表示された場合は、ジョブ id 番号 (ジョブ id の範囲は 1 ～ 999999) が折り返されるか、移動したチェックポイントジョブが表示されます。

チェックポイント機能のサポート

チェックポイントは、実行中の状態またはアプリケーションの状態を凍結して、その状態 (いわゆるチェックポイント) をディスクに保存し、システム停止などの原因で、そのジョブまたはアプリケーションの実行を最後まで行えなかった場合にそのチェックポイントから再開できるようにする機能です。チェックポイントを別のホストに移動できる場合は、チェックポイント機能を使用して、計算資源をあまり失うことなく、クラスタ内のアプリケーションまたはジョブを移動することができます。つまり、チェックポイント機能の助けを借りて、動的な負荷均衡を実現することができます。

Sun Grid Engine, Enterprise Edition システムは、2 つのレベルのチェックポイント機能をサポートしています。

- **ユーザーレベルのチェックポイント機能**

  このレベルでのチェックポイント生成機能は、完全にユーザーまたはアプリケーションの責任で実現します。ユーザーレベルのチェックポイント機能としては、たとえば以下があります。

- チェックポイントファイルの定期的な書き込み - アプリケーションでは、重要なアルゴリズムステップでそれらのファイルを符号化し、アプリケーション再起動時にファイルが正しく処理されるようにします。
- アプリケーションとリンクする必要があるチェックポイントライブラリの利用 - この方法でチェックポイント機能をインストールします。

注 – サン以外のさまざまなアプリケーションに、チェックポイントファイルの書き込みに基づく組み込み型のチェックポイント機能が用意されています。チェックポイントライブラリは、パブリックドメイン (たとえば ウィスコンシン大学の Condor プロジェクト) またはハードウェアベンダーから入手できます。

- **カーネルレベルの透過的チェックポイント機能**

  このレベルのチェックポイント機能は、任意のジョブに適用できるようオペレーティングシステム (またはその拡張機能として) によって実現する必要があります。カーネルレベルのチェックポイント機能を使用するために、ソースコードを変更したり、アプリケーションを再リンクしたりする必要はありません。

カーネルレベルのチェックポイント機能が、ジョブ全体、すなわち、ジョブによって作成されたプロセス階層に適用できるのに対し、ユーザーレベルのチェックポイント機能は通常単一プログラムに制限されます。つまり、そうしたプログラムが埋め込まれているジョブは、ジョブ全体を再開した場合に、この問題に正しく対処する必要があります。

チェックポイントライブラリに基づくチェックポイント機能ばかりでなく、カーネルレベルのチェックポイント機能は、チェックポイント生成時にジョブまたはアプリケーションが使用している仮想アドレス空間全体をディスクにダンプするため、非常に多くの資源を消費する可能性があります。これに比べて、チェックポイントファイルに基づくユーザーレベルのチェックポイント機能では、チェックポイントに書き込むデータを重要情報にだけ制限することができます。

チェックポイント環境

チェックポイントの実行方法は、オペレーティングシステムのアーキテクチャによってさまざまな種類があり、またそれらの方法から派生する方法もさまざま登場する可能性があります。このため、Sun Grid Engine, Enterprise Edition では、使用するチェックポイントの実行方法に関する属性定義を行えるようにしています。

この属性定義を「チェックポイント環境」といいます。Sun Grid Engine, Enterprise Edition には、デフォルトのチェックポイント環境が用意されており、必要に応じてサイトで変更することができます。

基本的に新しいチェックポイント実行方法を組み込むこともできますが、これは難しい作業になる可能性があり、そうした作業は、経験の豊富なスタッフまたは Sun Grid Engine, Enterprise Edition サポートチームだけが行うようにしてください。

▼ QMON からチェックポイント環境を構成する

1. QMON メインメニューから「チェックポイント構成」のアイコンをクリックします。

   図 9-17 に示すような「チェックポイント構成」ダイアログボックスが表示されます。

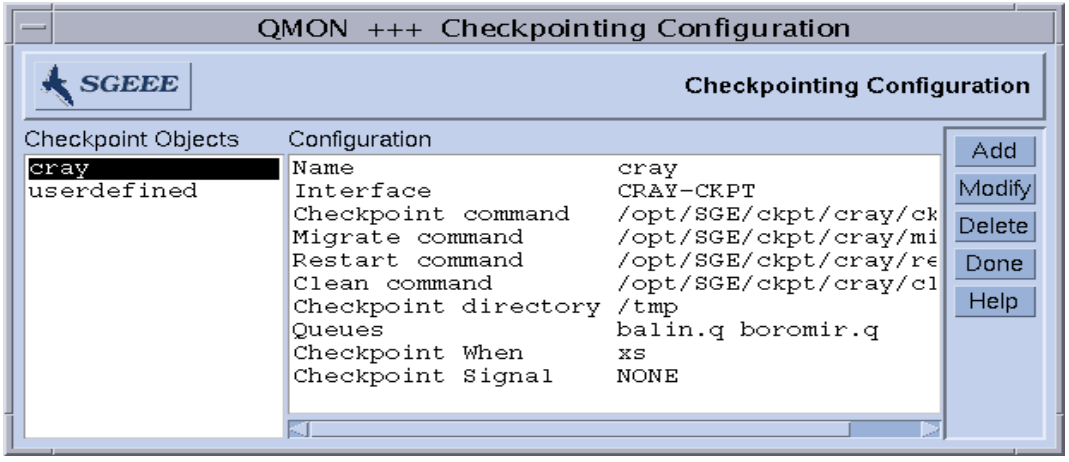


図 9-17　「チェックポイント構成」ダイアログボックス

2. 「チェックポイント構成」ダイアログボックスで、以下のうちの目的の操作を行います。

構成済みチェックポイント環境の表示

- 以前に構成したチェックポイント環境を表示するには、「チェックポイントオブジェクト」欄からチェックポイント環境名を選択します。

  「構成」欄に、選択された環境の構成が表示されます。

構成済みチェックポイント環境の削除

- 構成済みチェックポイント環境を削除するには、「チェックポイントオブジェクト」欄から環境名を選択し、「削除」ボタンをクリックします。

構成済みチェックポイント環境の変更

1. **「チェックポイントオブジェクト」欄で変更する構成済みチェックポイント環境名を選択し、「変更」をクリックします。**

   選択したチェックポイント環境の現在の構成情報が入った、図 9-18 に示すような「チェックポイントオブジェクトの変更」ダイアログボックスが表示されます。

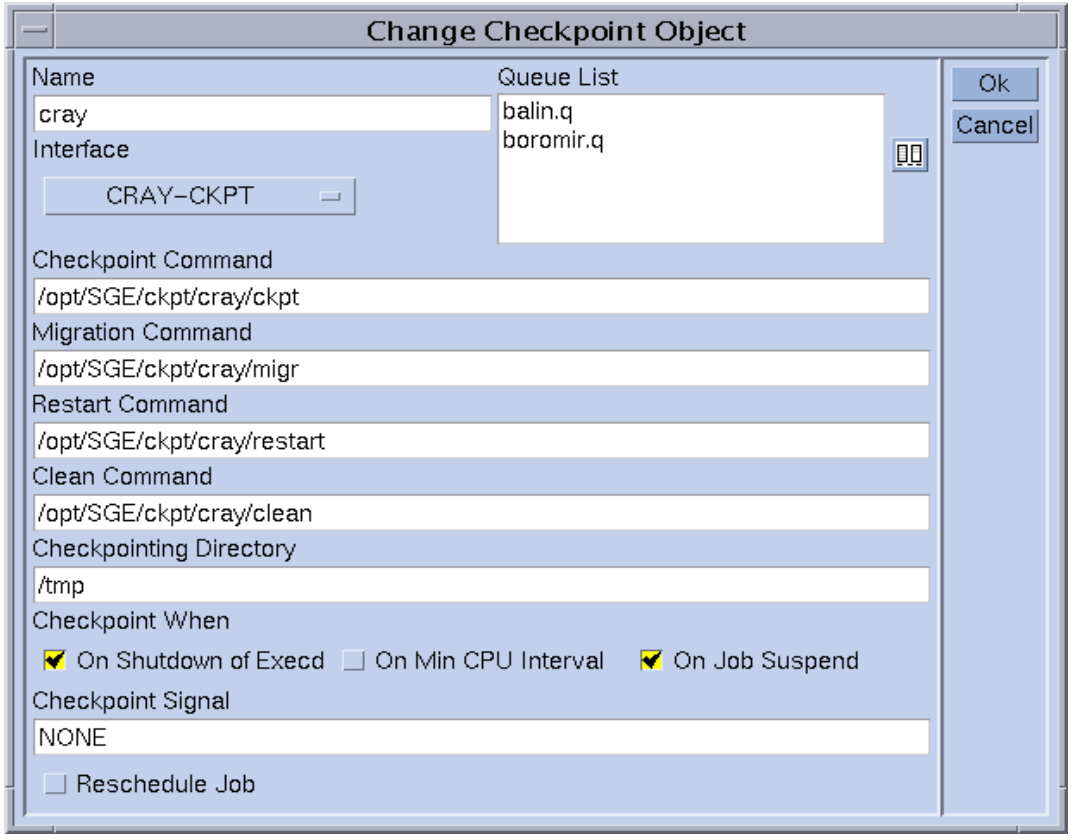


図 9-18　「チェックポイントオブジェクトの変更」ダイアログボックス

2. **次の説明に従って選択したチェックポイント環境を変更します。**

   「チェックポイントオブジェクトの変更」ダイアログボックスでは、以下を変更することができます。

   - 名前
   - **チェックポイント、移動、再開、後処理**コマンド文字列
   - チェックポイントファイル保存ディレクトリ
   - チェックポイントの開始時期
   - チェックポイント開始時にジョブまたはアプリケーションに送信するシグナル

注 – これらのパラメータについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `checkpoint` の項を参照してください。また、使用するインタフェース (チェックポイント実行方法) も定義する必要があります。対応する選択リストに用意されている方法のいずれかを選択してください。さまざまなインタフェースの意味についての詳細は、同じく `checkpoint` の項を参照してください。

3. **重要 - Sun Grid Engine, Enterprise Edition が提供するチェックポイント環境の場合は、「名前」と「チェックポイントディレクトリ」、「キューリスト」パラメータのみ変更できます。**

   「キューリスト」パラメータを変更する場合は、手順 a に進んでください。それ以外の場合は、手順 a を飛ばして、手順 4 に進みます。

   a. **「キューリスト」区画の右側のアイコンをクリックします (図 9-18 を参照)。**

      図 9-19 に示すような「キューの選択」ダイアログボックスが表示されます。

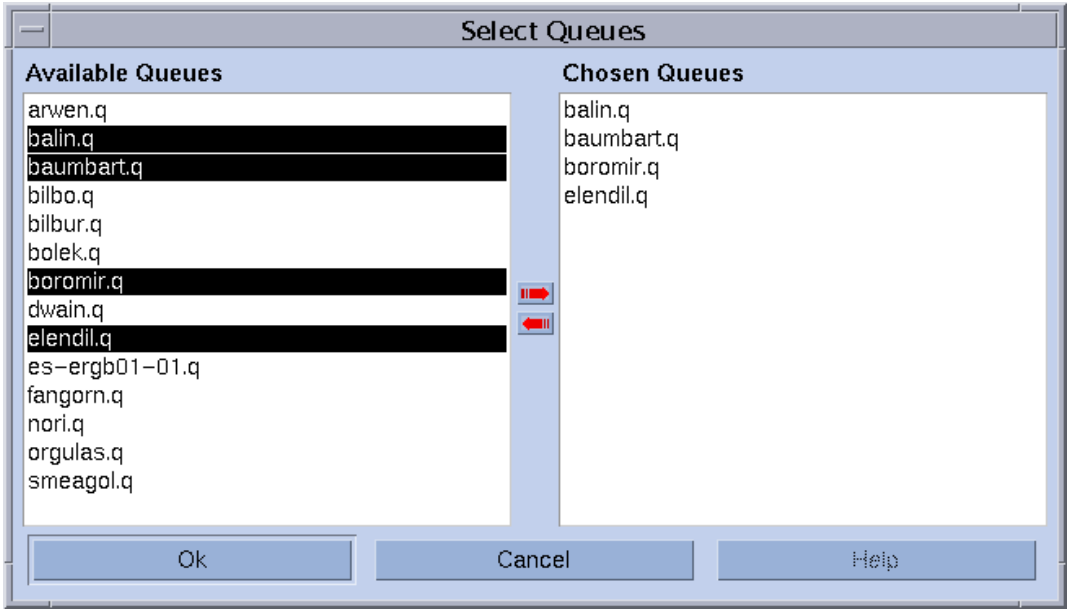


図 9-19 チェックポイントの「キューの選択」ダイアログボックス

   b. **「使用可能なキュー」のリストからチェックポイント環境に含めるキューを選択して、「選択されているキュー」リストに追加します。**

   c. **「了解」をクリックします。**

      「チェックポイントオブジェクトの変更」ダイアログボックスの「キューリスト」ウィンドウに選択したキューが表示されます。

4. sge_qmaster に変更を登録する場合は「了解」、変更を廃棄する場合は「キャンセル」をクリックします。

チェックポイント環境の登録

1. 「チェックポイント構成」ダイアログボックスで「追加」をクリックします。

   編集可能な構成テンプレートの入った、図 9-18 に示すような「チェックポイントオブジェクトの変更」ダイアログボックスが表示されます。

2. テンプレートに必要な情報を入力して、完成します。

3. sge_qmaster に変更を登録する場合は「了解」、変更を廃棄する場合は「キャンセル」をクリックします。

▼ コマンド行からチェックポイント環境を構成する

- 以下の説明に従って qconf コマンドと適切なオプションを入力します。

qconf のチェックポイント用オプション

- qconf -ackpt *ckpt_name*

  チェックポイント環境の追加 - このコマンドは、エディタ (デフォルトの vi か、$EDITOR 環境変数に指定されたエディタ) を使用して、チェックポイント環境構成用のテンプレートを開きます。*ckpt_name* パラメータはチェックポイント環境名で、テンプレートの対応するフィールドに事前に入力されています。テンプレートの内容を変更し、ディスクに保存することによって、チェックポイント環境を構成してください。変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の checkpoint の項を参照してください。

- qconf -Ackpt *filename*

  ファイルからのチェックポイント環境の追加 - 指定されたファイルを構文解析して、チェックポイント環境構成を追加します。ファイルは、チェックポイント環境構成用のテンプレート形式である必要があります。

- qconf -dckpt *ckpt_name*

  チェックポイント環境の削除 - 指定されたチェックポイント環境を削除します。

- qconf -mckpt *ckpt_name*

  チェックポイント環境の変更 - このコマンドは、エディタ (デフォルトの vi か、$EDITOR 環境変数に指定されたエディタ) を使用し、指定されたチェックポイント環境を構成用テンプレートとして開きます。テンプレートの内容を変更し、ディスクに保存することによって、チェックポイント環境を構成してください。

変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `checkpoint` の項を参照してください。

- `qconf -Mckpt` *filename*

  ファイルからのチェックポイント環境の変更 - 指定されたファイルを構文解析して、既存のチェックポイント環境構成を変更します。ファイルは、チェックポイント環境構成用のテンプレート形式である必要があります。

- `qconf -sckpt` *ckpt_name*

  チェックポイント環境の表示 - 指定されたチェックポイント環境の構成を標準出力に出力します。

- `qconf -sckptl`

  チェックポイント環境リストの表示 - これまでに構成されているすべてのチェックポイント環境名を一覧表示します。

第 10 章

並列環境の管理

この章では、並列環境 (PE) の管理運用について説明します。

予備知識的な情報を提供するばかりでなく、次の作業を行う方法を詳しく説明します。

並列環境

「並列環境 (PE) 」は、ネットワーク環境または並列プラットフォームにおける並行コンピューティング用に設計されたソフトウェアパッケージです。この何年もの間にさまざまなシステムが発展を遂げ、さまざまなハードウェアプラットフォームで分散・並列処理技術が実用的なものになってきました。そうした環境として特に一般的なものとして、Oak Ridge National Laboratories の PVM (Parallel Virtual Machine,) と Message Passing Interface Forum の MPI (Message Passing Interface) という 2 つのメッセージ引き渡し環境があります。両方のツールとも、ハードウェアベンダー提供のものばかりでなく、パブリックドメインのものもあります。

これらのシステムはどれも異なる特徴を持ち、要求される使用条件がそれぞれに異なります。そうしたシステム上で動作する任意の並列ジョブに対応できるよう、Sun Grid Engine, Enterprise Edition システムには、さまざまなニーズを満たす柔軟で強力なインタフェースが用意されています。

Sun Grid Engine, Enterprise Edition システムは、PVM や MPI などの任意のメッセージ引き渡し環境を使用して並列ジョブを実行したり (詳細は、『PVM User's Guide』および『MPI User's Guide』を参照)、単一キューの複数スロットで、あるいは複数のキューまたはマシンに分散された共有メモリー並列プログラム (マシン分散の場合は分散メモリー並列ジョブ) を実行したりする手段を提供します。任意の数のさまざまな並列環境インタフェースを同時並行的に構成することができます。

298 ページの「並列環境の起動プロシージャ」と 300 ページの「並列環境の終了」の節で説明しているように、適切な並列環境の起動および停止プロシージャが用意されている限り、Sun Grid Engine, Enterprise Edition は任意の並列環境に接続することができます。

▼ QMON から並列環境を構成する

1. **QMON のメインメニューで「並列環境構成」ボタンをクリックします。**

   図 10-1 に示すような「並列環境構成」ダイアログボックスが表示されます。

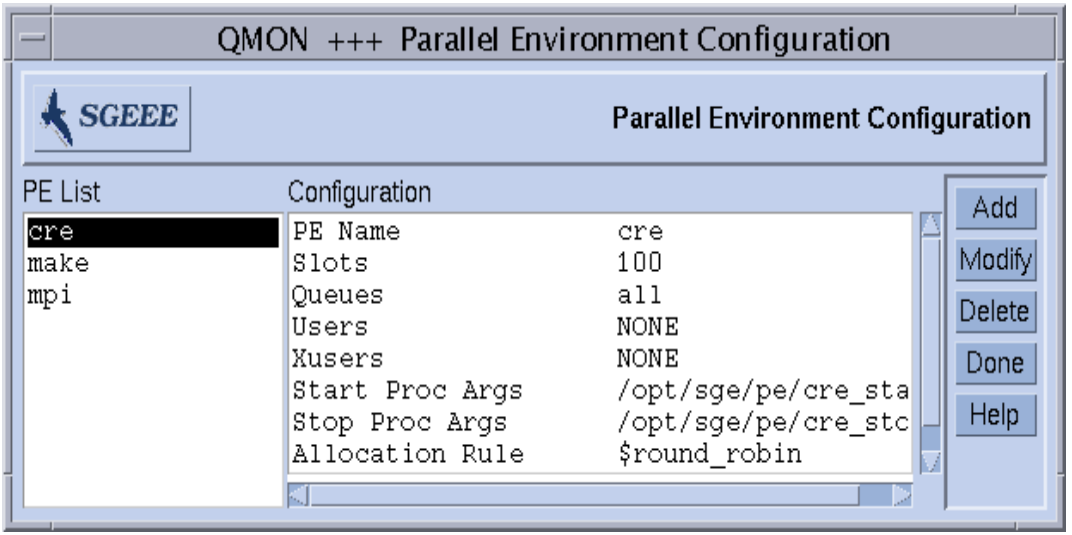


図 10-1 「並列環境構成」ダイアログボックス

画面の左側は「並列環境リスト」選択リストで、すでに構成されている並列環境が表示されます。

2. **「並列環境構成」ダイアログボックスで、以下の目的の操作を行います。**

▼ 並列環境の構成を表示する

- **並列環境の構成を表示するには、「並列環境リスト」選択リストで並列環境名をクリックします。**

  「構成」表示区画に、選択した並列環境の構成が表示されます。

▼ 並列環境を削除する

- **並列環境を削除するには、「並列環境リスト」選択リストで並列環境名を選択して、ダイアログボックスの右側にある「削除」をクリックします。**

▼ 並列環境を変更する

1. **並列環境を変更するには、並列環境名を選択して、「変更」ボタンをクリックします。**

   図 10-2 に示すような「並列環境の定義」ダイアログボックスが表示されます。

2. **292 ページの「並列環境定義パラメータの説明」の節の説明に従って、並列環境の定義を変更します。**

3. **「了解」をクリックして変更内容を保存するか、「キャンセル」をクリックして変更内容を廃棄します。**

   どちらの場合も、ダイアログボックスが閉じます。

▼ 並列環境を追加する

1. **新しい並列環境を追加するには、「追加」ボタンをクリックします。**

   図 10-2 に示すような「並列環境の定義」ダイアログボックスが表示されます。

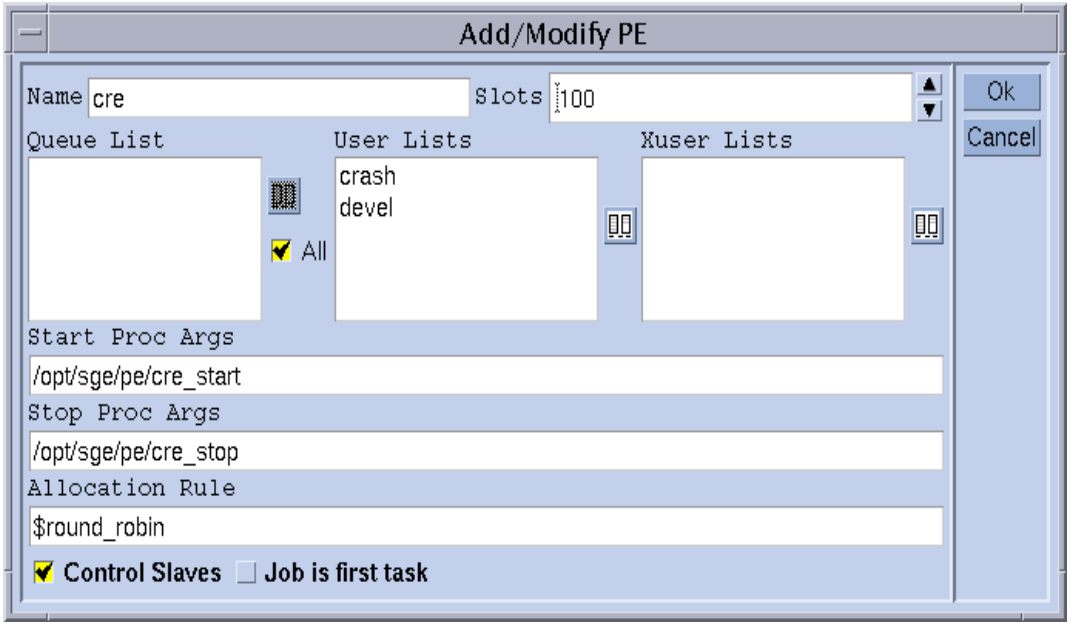


図 10-2　「並列環境の定義」ダイアログボックス

2. **292 ページの「並列環境定義パラメータの説明」の節の説明に従って、並列環境の定義を追加します。**
3. **「了解」をクリックして変更内容を保存するか、「キャンセル」をクリックして変更内容を廃棄します。**

   どちらの場合も、ダイアログボックスが閉じます。

並列環境定義パラメータの説明

- 「名前」入力フィールド - 変更の場合は、選択された並列環境名が表示されます。追加の場合は、このフィールドを使用して定義する並列環境の名前を入力することができます。
- 「スロット」スピンボックス - 並行して実行するすべての並列環境ジョブが占有すると考えられる総ジョブスロット数を入力します。

- 「キューリスト」区画は - 並列環境が使用可能なキューを示します。「キューリスト」区画右側にあるアイコンボタンをクリックすると、図 10-3 に示すような「キューの選択」ダイアログボックスが表示され、並列環境用のキューリストを変更することができます。「すべて」チェックボックスを使用して、並列環境がすべての並列キューを使用するように指定することもできます。

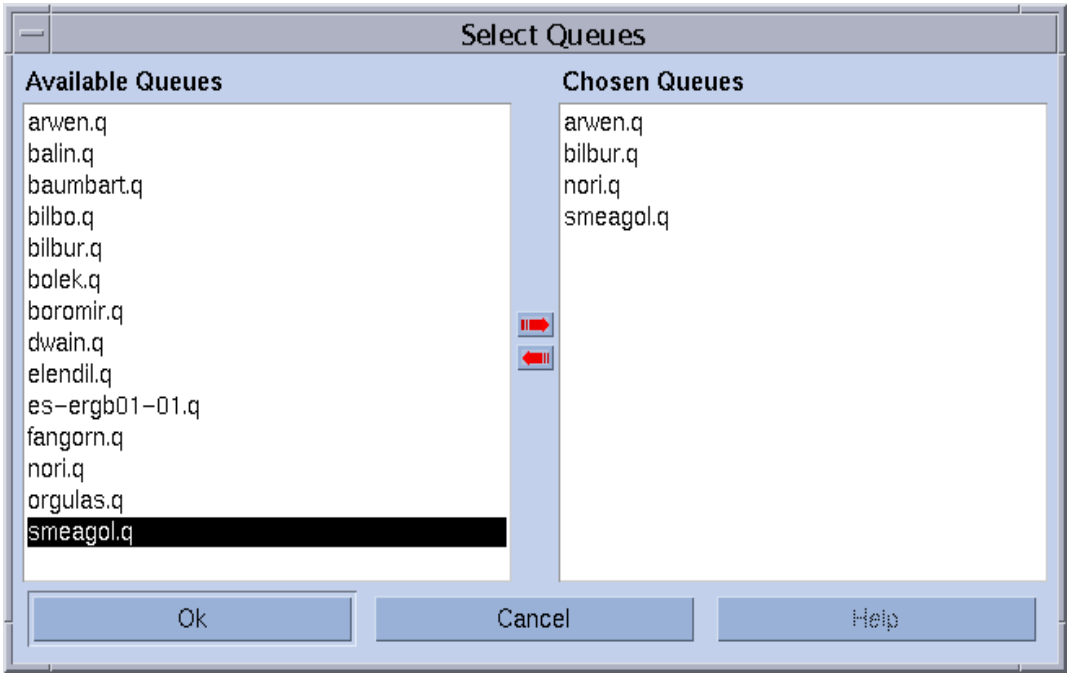


図 10-3　「キューの選択」ダイアログボックス

- 「ユーザーリスト」区画 - 並列環境へのアクセスを許可するユーザーアクセスリストが表示されます (226 ページの「ユーザーのアクセス権」の節を参照)。
- 「X ユーザーリスト」区画 - アクセスを拒否するアクセスリストが表示されます。

両方の区画とも、関連付けられているアイコンボタンをクリックすると、図 10-4 に示すような「アクセスリストの選択」ダイアログボックスが表示されます。これらのダイアログボックスを使用して、アクセスリストの表示内容を変更することができます。

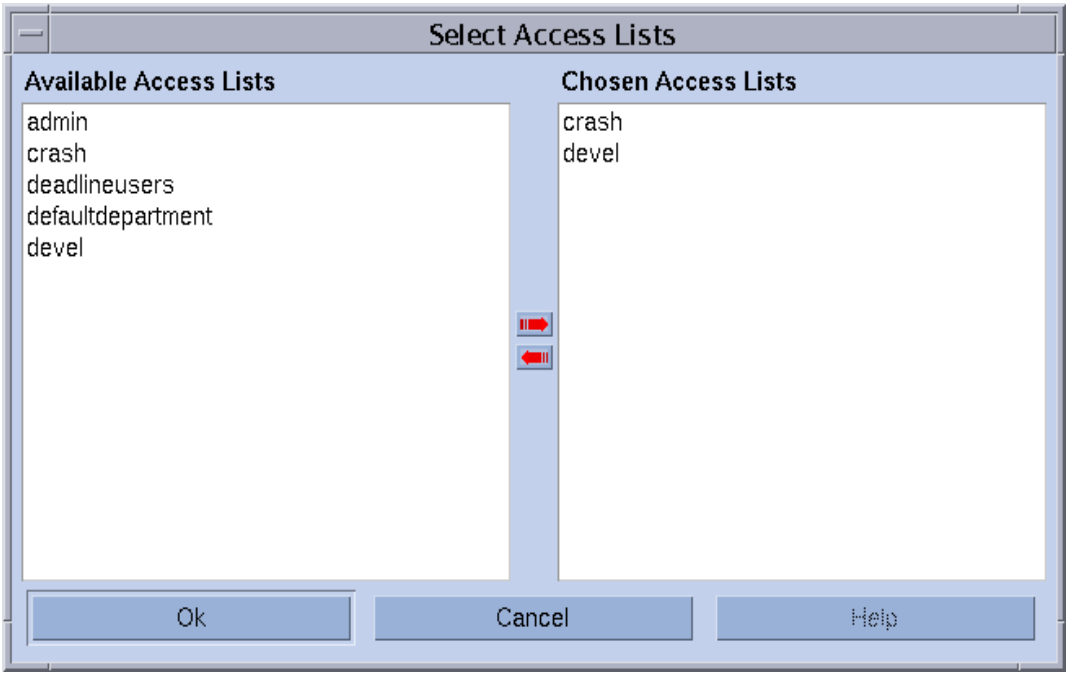


図 10-4　「アクセスリストの選択」ダイアログボックス

- 「起動プロシージャの引数」と「停止プロシージャの引数」入力フィールド - 並列環境の起動および停止プロシージャの正確な起動シーケンスを入力します (それぞれ 298 ページの「並列環境の起動プロシージャ」と 300 ページの「並列環境の終了」の節を参照)。これらのパラメータは必須ではないことに注意してください。並列環境にそうしたプロシージャが必要がない場合は、フィールドを空にしておくことができます。

  通常、先頭の引数には、起動または停止プロシージャそのものを指定します。残りのパラメータは、そのプロシージャに対するコマンド行引数です。

  Sun Grid Engine, Enterprise Edition の内部実行時情報をプロシージャに渡すための各種の特殊な識別子 ($ 接頭辞から始まる) が用意されてます。『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `sge_pe` の項に、使用可能な全パラメーター一覧が含まれています。

- 「割り当て規則」入力フィールド - 並列環境で使用する各マシンに割り当てる並列プロセス数を定義します。正の整数を指定すると、適切な各ホストに割り当てられるプロセスがその数に固定されます。特殊なデノミネータ `$pe_slots` を使用すると、ジョブの全範囲のプロセスを単一のホスト (SMP) に割り当てることができます。デノミネータ `$fill_up` および `$round_robin` を使用すると、各ホストに不均等にプロセスを割り当てることができます。

これらの割り当て規則についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の sge_pe の項を参照してください。

- 「スレーブを制御する」トグルボタン - Sun Grid Engine, Enterprise Edition (すなわち、sge_execd および sge_shepherd) を使用して並列タスクを生成するかどうかを指定します。Sun Grid Engine, Enterprise Edition を使用しない場合は、並列環境が独自のプロセス生成を行います。Sun Grid Engine, Enterprise Edition システムがスレーブタスクを完全に制御すると、適切なアカウンティングと資源制御が行われるというメリットがありますが、この機能を使用できるのは、Sun Grid Engine, Enterprise Edition 専用にカスタマイズされた並列環境インタフェースだけです。詳細は、300 ページの「並列環境と Sun Grid Engine, Enterprise Edition ソフトウェアの密統合」を参照してください。
- 「ジョブは最初のタスクです」トグルボタン - 「スレーブを制御する」が有効な場合にのみ意味を持ち、ジョブスクリプトまたはその子プロセスの 1 つが並列アプリケーションの並列タスクの 1 つとして働くことを示します (通常、これは PVM などに該当します)。このトグルボタンが無効の場合、ジョブスクリプトは並列アプリケーションの実行を開始しますが、参加しません (たとえば、mpirun を使用したときの MPI など)。

▼ コマンド行から並列環境を構成する

- **以下の説明に従い、適切なオプションを付けて qconf コマンドと適切なオプションを入力します。**

qconf の並列環境関係のオプション

- qconf -ap *pe_name*

  並列環境の追加 - このコマンドは、エディタ (デフォルトの vi か、$EDITOR 環境変数に指定されたエディタ) を使用して、並列環境構成用のテンプレートを開きます。*pe_name* パラメータは並列環境名で、テンプレートの対応するフィールドに事前に入力されています。テンプレートの内容を変更し、ディスクに保存することによって、並列環境を構成してください。変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の sge_pe の項を参照してください。

- qconf -Ap *filename*

  ファイルからの並列環境の追加 - 指定されたファイルを構文解析して、並列環境構成を追加します。ファイルは、並列環境構成用のテンプレート形式である必要があります。

- qconf -dp *pe_name*

  並列環境の削除 - 指定された並列環境を削除します。

- `qconf -mp` *pe_name*

  並列環境の変更 - このコマンドは、エディタ (デフォルトの `vi` か、`$EDITOR` 環境変数に指定されたエディタ) を使用し、指定された並列環境を構成用テンプレートとして開きます。テンプレートの内容を変更し、ディスクに保存することによって、並列環境を変更してください。変更するテンプレートのエントリについての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `sge_pe` の項を参照してください。

- `qconf -Mp` *filename*

  ファイルからの並列環境の変更 - 指定されたファイルを構文解析して、並列環境構成を追加します。ファイルは、並列環境構成用のテンプレート形式である必要があります。

- `qconf -sp` *pe_name*

  並列環境の表示 - 指定された並列環境の構成を標準出力に出力します。

- `qconf -spl`

  並列環境リストの表示 - これまでに構成されているすべての並列環境名を一覧表示します。

▼ コマンド行から既存の並列環境インタフェースを表示する

- **次のコマンドを入力します。**

```
% qconf -spl
% qconf -sp pe_name
```

最初のコマンドは、現在使用可能な並列環境インタフェース名を一覧表示します。2 つ目のコマンドは、特定の並列環境インタフェースの構成を表示します。並列環境の構成についての詳細は、`sge_pe` のマニュアルページを参照してください。

▼ QMON から既存の並列環境インタフェースを表示する

- **QMON メインメニューで「並列環境構成」ボタンをクリックします。**

「並列環境構成」ダイアログボックスが表示されます (290 ページの「QMON から並列環境を構成する」の節を参照)。

並列ジョブの定義例は、84 ページの「高度な設定」の節ですでに紹介しています。その並列ジョブでは、少なくとも 4 個、望ましくは最高で 16 個のプロセスで並列環境インタフェースの mpi (メッセージ引き渡しインタフェース) を使用するよう要求していました。高度な実行依頼画面の「並列環境 (PE) の指定」フィールド右横のボタンを使用すると、ダイアログボックスが表示され、使用可能な並列環境のリストから目的の並列環境を選択することができます (図 10-5 を参照)。また、そのフィールドに指定した並列環境名の後ろには、ジョブが開始する並列タスク数を範囲指定することができます。

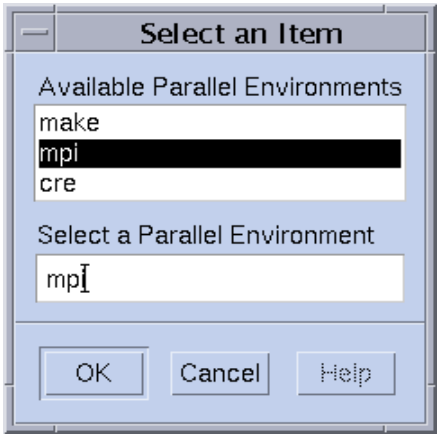


図 10-5　並列環境の選択

上記の並列ジョブの指定に対応するコマンド行の実行依頼コマンドについては、96 ページの「コマンド行からジョブの実行依頼をする」を参照してください。qsub コマンドで同等の要求を表す -pe オプションの用法を示しています。-pe 構文についての詳細は、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の qsub の項を参照してください。

並列ジョブに合った適切な並列環境インタフェースを選択することが重要です。並列環境インタフェースはメッセージ引き渡しシステムを利用しないこともあれば、利用するシステムが異なることもあります。単一のホストにプロセスを割り当てることもあれば、複数のホストに割り当てることもあります。また、一部のユーザーの並列環境へのアクセスを拒否されるケースもあります。特定のキューしか利用しない並列環境インタフェースもあり、いつでも特定の個数のキュースロットしか占有しない並列環境インタフェースもあります。このため、使用可能な並列環境インタフェースで自分の並列ジョブに最適なインタフェースについては、Sun Grid Engine, Enterprise Edition の管理者にお尋ねください。

89 ページの「資源要求の定義」の説明しているように、並列環境要求とともに資源要求を指定することができます。この指定を行うと、並列環境インタフェースに適したキューが、指定された資源要求に合うキューにさらに絞られます。たとえば、次のコマンドでジョブの実行依頼をしたと仮定します。

```
% qsub -pe mpi 1,2,4,8 -l nastran,arch=osf nastran.par
```

このジョブに適したキューは、並列環境構成で並列環境インタフェース mpi に関連付けられていて、かつ -l オプションで指定された資源要求を満たすキューになります。

注 – Sun Grid Engine, Enterprise Edition の並列環境インタフェースは、構成の自由度が大きい機能です。Sun Grid Engine, Enterprise Edition の管理者は、サイトに固有のニーズに合わせて並列環境の起動および停止プロシージャを構成することができます (sge_pe のマニュアルページを参照)。ジョブの実行依頼をするユーザーは、環境変数をエクスポートする qsub の -v および -V オプションを使用して、並列環境の起動および停止プロシージャに情報を渡すことができます。特定の環境変数をエクスポートする必要があるかどうかについて不明な点がある場合は、Sun Grid Engine, Enterprise Edition の管理者にお尋ねください。

並列環境の起動プロシージャ

Sun Grid Engine, Enterprise Edition システムは、exec システムコールで起動プロシージャを実行することによって並列環境を起動します。起動用の実行可能ファイル名とそのファイルに渡すパラメータは、Sun Grid Engine, Enterprise Edition システムの中から設定することができます。Sun Grid Engine, Enterprise Edition ディストリビューションツリーには、PVE 環境用のサンプル起動プロシージャが含まれています。このプロシージャは、シェルスクリプト 1 つとそのスクリプトによって実行される C プログラム 1 つで構成されています。シェルスクリプトは C プログラムを使用して、 PVM をクリーンに起動します。その他の必要な処理はすべて、シェルスクリプトが行います。

このシェルスクリプトのパスは <*sge_root*>/pvm/startpvm.sh、C プログラムファイルのパスは <*sge_root*>/pvm/src/start_pvm.c です。

注 – 起動プロシージャが、C プログラム 1 つだけであってもかまいません。シェルスクリプトを使用することによって、サンプルの起動プロシージャのカスタマイズが容易になります。

サンプルスクリプトの startpvm.sh には、次の 3 つのパラメータが必要です。

- Sun Grid Engine, Enterprise Edition ソフトウェアによって生成されたホストファイル (起動する PVM が存在するホスト名が含まれ) のパス
- `startpvm.sh` プロシージャの起動元のホスト
- PVM ルートディレクトリのパス (通常は、`PVM_ROOT` 環境変数に含まれる)

これらのパラメータは、290 ページの「`QMON` から並列環境を構成する」で説明している方法を使用して、起動スクリプトに渡すことができます。実行中、Sun Grid Engine, Enterprise Edition によって並列環境の起動および停止スクリプトに提供されるパラメータは、この他にもあります。たとえば必要なホストファイルは Sun Grid Engine, Enterprise Edition によって生成され、そのファイル名は、並列環境構成で特殊パラメータ名 `$sge_hostfile` を使用して起動プロシージャに渡すことができます。使用可能なすべてのパラメータについては、『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』の `sge_pe` の項を参照してください。

ホストファイルの形式は次のとおりです。

- ファイルの各行は、並列プロセスを実行するホストを表します。
- 各行の最初のエントリは、キューのホスト名を示します。
- 2 つ目のエントリは、キューで実行する並列プロセス数を示します。
- 3 つ目のエントリは キューの名前です。
- 4 つ目のエントリは、マルチプロセッサマシンの場合に使用するプロセッサ範囲を示します。

Sun Grid Engine, Enterprise Edition は、つねにこの形式でホストファイルを生成します。異なるファイル形式を必要とする並列環境 (PVM など) の場合は、起動プロシージャ内で形式を変換する必要があります (`startpvm.sh` ファイルを参照)。

Sun Grid Engine, Enterprise Edition システムによって並列環境の起動プロシージャが実行されると、ただちに並列環境が起動されます。起動プロシージャは、ゼロの終了ステータスで終了するようにします。起動プロシージャの終了ステータスがゼロ以外の場合、Sun Grid Engine, Enterprise Edition ソフトウェアはエラーを返し、並列ジョブの実行を開始しません。

注 – Sun Grid Engine, Enterprise Edition の枠組みに起動プロシージャを組み込むと、エラーが発生した場合に、その原因を突き止めることが難しくなることがあります。このため、最初は Sun Grid Engine, Enterprise Edition なしで、コマンド行から起動プロシージャをテストして、すべてのエラーを取り除いておくことを推奨します。

並列環境の終了

並列ジョブが正常終了するか、qdel を使用して途中で打ち切られると、その並列環境を停止するプロシージャが呼び出されます。このプロシージャの定義と構文は、起動プロシージャで説明したものと非常によく似ています。停止プロシージャは、並列環境構成で定義することもできます (たとえば、290 ページの「QMON から並列環境を構成する」を参照)。

停止プロシージャの目的は、並列環境を停止して、関係するすべてのプロセスを刈り取ることにあります。

注 – 停止プロシージャが並列環境プロセスの後処理に失敗すると、Sun Grid Engine, Enterprise Edition システムが並列環境の制御下で動作しているプロセスの情報を得られずに、後処理を行えないことがあります。当然、Sun Grid Engine, Enterprise Edition ソフトウェアは、自身が起動したジョブスクリプトに直接関係するすべてのプロセスの後処理をします。

Sun Grid Engine, Enterprise Edition ディストリビューションツリーには、PVM 並列環境用のサンプル停止プロシージャも含まれています。このサンプルのパスは <*sge_root*>/pvm/stoppvm.sh で、次の 2 つの引数をとります。

- Sun Grid Engine, Enterprise Edition システムによって生成されたホストファイルのパス
- 停止プロシージャの実行元ホスト名

起動プロシージャ同様、停止プロシージャは、実行成功時にゼロの終了ステータス、失敗時にゼロ以外の終了ステータスを返すとみなされます。

注 – Sun Grid Engine, Enterprise Edition の枠組みに停止プロシージャを組み込むと、エラーが発生した場合に、その原因を突き止めることが難しくなることがあります。このため、最初は Sun Grid Engine, Enterprise Edition なしで、コマンド行から停止プロシージャをテストして、すべてのエラーを取り除いておくことを推奨します。

並列環境と Sun Grid Engine, Enterprise Edition ソフトウェアの密統合

290 ページの「QMON から並列環境を構成する」の節で「スレーブを制御する」パラメータの説明では、Sun Grid Engine, Enterprise Edition コンポーネントの sge_execd および sge_shepherd で並列タスクを作成する方が、並列環境で独自のプロセス作成を行うよりメリットが多いとしていました。これは、UNIX オペレーティングシステムでは、プロセス階層の作成者だけが信頼性の高い資源管理を行える

ためです。並列アプリケーションのための適切なアカウンティング、資源の利用制限、プロセス制御などの機能は、すべての並列タスクの作成者だけが適用することができます。

大部分の並列環境には、こうした機能は実装されていません。このため、Sun Grid Engine, Enterprise Edition などの資源管理システムと統合するための十分なインタフェースが用意されていません。この問題を克服するには、Sun Grid Engine, Enterprise Edition システムの側で、並列環境と密に統合するための高度な並列環境インタフェースを提供し、タスクの生成の仕事を並列環境から Sun Grid Engine, Enterprise Edition に移します。

Sun Grid Engine, Enterprise Edition ディストリビューションには、パブリックドメイン版 PVM と Argonne National Laboratories の MPICH MPI 実装版用のそうした密統合例が用意されており、それぞれ *<sge_root>*/pvm と *<sge_root>*/mpi ディレクトリに含まれています。それのディレクトリにはまた、使用方法と現時点での制限事項を記述した README ファイルも含まれています。詳細は、それら README ファイルを参照してください。

また、比較用として、*<sge_root>*/mpi/sunhpc/loose-integration ディレクトリには、Sun HPC ClusterToos™ ソフトウェアとの疎の統合例が含まれており、*<sge_root>*/mpi ディレクトリにも比較用に疎統合バージョンの並列環境インタフェースが含まれています。

注 – 並列環境との密統合は高度な作業であり、その並列環境と Sun Grid Engine, Enterprise Edition の並列環境インタフェースに関する専門的な知識が必要になることがあります。必要な場合は、サンからサポートを受けることができます。

第 11 章

エラーの通知と障害追跡

この章では、Sun Grid Engine, Enterprise Edition 5.3 のエラーの通知方法を説明するとともに、よくある問題の解決方法に関するヒントを提供します。

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアからのエラーの報告

Sun Grid Engine, Enterprise Edition ソフトウェアは、特定のファイルにメッセージを記録するか、電子メールでエラーや警告を報告します。使用されるログファイルは以下のとおりです。

- Messages ファイル:

  `sge_qmaster` と `sge_schedd` `sge_execd` のそれぞれに `messages` ファイルがあります。ファイル名は共通で messages です。それぞれ、`sge_qmaster` のログファイルはマスターのスプールディレクトリ、`sge_schedd` の messages ファイルはスケジューラのスプールディレクトリ、実行デーモンのログファイルは実行デーモンのスプールディレクトリにあります (スプールディレクトリについての詳細は、24 ページの「ルートディレクトリ内のスプールディレクトリ」の節を参照)。

  messages ファイルの形式は以下のとおりです。

  - 1 行に 1 つのメッセージ。
  - 1 つのメッセージは縦線 (|) で区切られた 5 つの要素で構成される。
  - 最初の要素はメッセージのタイムスタンプ。
  - 2 つ目の要素はメッセージを生成した Sun Grid Engine, Enterprise Edition のデーモン名。
  - 3 つ目の要素はデーモンが動作しているホスト名。

- 4 つ目の要素はメッセージの種類。通知の N、情報の I (最初の 2 つは通常の通知目的)、警告 の W、エラーの E (エラー状態検出)、重大の C (プログラムの異常終了になる可能性あり) のいずれか。

  クラスタ構成で loglevel パラメータを使用して、記録するメッセージの種類をグローバルまたはローカルのどちらに傾かせるか指定することができます。

- 5 つ目の要素はメッセージ本文。

注 – 何らかの理由でエラーログファイルにアクセスできない場合、Sun Grid Engine, Enterprise Edition は対応するホストの /tmp/sge_qmaster_messages、/tmp/sge_schedd_messages、/tmp/sge_execd_messages のいずれかにエラーメッセージを記録しようとします。

- ジョブの STDERR 出力:

  ジョブが開始されると、ジョブスクリプトの標準エラー (STDERR) 出力はただちにファイルにリダイレクトされます。このファイル名と場所はデフォルトに準じていてもかまいませんし、いくつか qsub コマンド行スイッチを使用して指定することもできます。詳細は、『*Sun Grid Engine User's Guide*』および『Sun Grid Engine 5.3/Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』を参照してください。

状況によっては、Sun Grid Engine, Enterprise Edition は、電子メールでユーザーか管理者、またはその両方にエラーイベントの発生を通知します。Sun Grid Engine, Enterprise Edition が送信するこうしたメールメッセージには、メッセージ本文は含まれません。メッセージテキストは、メールの件名フィールドにすべて含まれます。

さまざまなエラーまたは終了コードの意味

表 11-1 は、ジョブ関連のさまざまエラーまたは終了コードをまとめています。これらのコードは、あらゆる種類の Sun Grid Engine, Enterprise Edition ジョブに該当します。

表 11-1 ジョブ関連のエラーまたは終了コード

スクリプト / 方法	終了 / エラーコード	意味
ジョブスクリプト	0	成功
	99	再キューイング
	その他	成功: アカウンティングファイル内の終了コード

表 11-1　ジョブ関連のエラーまたは終了コード (続き)

スクリプト / 方法	終了 / エラーコード	意味
プロローグ / エピローグ	0	成功
	99	再キューイング
	その他	キューのエラー状態: ジョブの再キューイング

表 11-2 は、並列環境 (PE) 構成関連のジョブのエラーまたは終了コードをまとめています。

表 11-2　並列環境関連のエラーまたは終了コード

スクリプト / 方法	終了 / エラーコード	意味
pe_start	0	成功
	その他	キューをエラー状態に設定: ジョブの再キューイング
pe_stop	0	成功
	その他	キューをエラー状態に設定: ジョブの再キューイング

表 11-3 は、キュー構成関連のジョブのエラーまたは終了コードをまとめています。これらのコードは、対応する方法が書き換えられた場合にのみ該当します。

表 11-3　キュー関連のエラーまたは終了コード

スクリプト / 方法	終了 / エラーコード	意味
ジョブ開始	0	成功
	その他	成功。他の特別な意味なし。
一時停止	0	成功
	その他	成功。他の特別な意味なし。
再開	0	成功
	その他	成功。他の特別な意味なし。

表 11-3　キュー関連のエラーまたは終了コード (続き)

スクリプト / 方法	終了 / エラーコード	意味
終了	0	成功
	その他	成功。他の特別な意味なし。

表 11-4 は、チェックポイント関連のジョブのエラーまたは終了コードをまとめています。

表 11-4　チェックポイント関連のエラーまたは終了コード

スクリプト / 方法	終了 / エラーコード	意味
チェックポイント	0	成功
	その他	成功。ただし、カーネルチェックポイントの場合は、特別な意味があり、チェックポイントの実行不成功で、行われなかった。
移動	0	成功
	その他	成功。ただし、カーネルチェックポイントの場合は、特別な意味があり、チェックポイントの実行不成功で、行われなかった。移動は行われる。
再開	0	成功
	その他	成功。他の特別な意味なし。
後処理	0	成功
	その他	成功。他の特別な意味なし。

デバッグモードでの Sun Grid Engine, Enterprise Edition の実行

重大なエラー状態が発生した場合は、問題の特定に必要な情報がエラー記録機構によって生成されないことがあります。このため、Sun Grid Engine, Enterprise Edition には、ほぼあらゆる補助プログラムとデーモンをデバッグモードで実行する

機能が用意されています。デバッグのレベルは、提供される情報の量および深さに応じて、0 から 10 のレベルがあり、10 は最も詳細な情報を提供するレベル、0 はデバッグ無効です。

Sun Grid Engine, Enterprise Edition ディストリビューションには、ユーザーの .cshrc または .profile リソースファイルに、デバッグレベルを設定する機能を付加するファイルが用意されています。csh または tcsh の場合は *<sge_root>*/<util>/dl.csh というファイル、sh または ksh の場合は *<sge_root>*/util/dl.sh というファイルです。標準のリソースファイルに、これらのうちの適切なファイルをそのまま取り込む必要があります。csh または tcsh を使用している場合は、自分の .cshrc ファイルに次の行を含めてください。

```
source <sge_root>/util/dl.csh
```

sh または ksh を使用している場合は、.profile ファイルに次の行を追加します。

```
. <sge_root>/util/dl.sh
```

いったんログアウトして、ログインし直すと、次のコマンドを使用してデバッグレベルの *level* を設定できるようになります。

```
% dl level
```

level が 0 より大きい場合、以降 Sun Grid Engine, Enterprise Edition のコマンドを実行すると、強制的にトレース出力は STDOUT に書き込まれます。このトレース出力には、有効なデバッグレベルによっては、警告やステータス、エラーメッセージばかりでなく、内部的に呼び出されたプログラムモジュール名がソースコードの行番号情報 (エラーを報告するさいに役立つ) とともに含まれます。

注 – かなりのサイズのスクロール行バッファー (たとえば 1000 行) を持つウィンドウでデバッグトレースを監視することを推奨します。

注 – xterm ウィンドウを使用している場合は、xterm のログ記録機能を使用してトレース出力を調べることを推奨します。

デバッグモードで Sun Grid Engine, Enterprise Edition デーモンを実行すると、デーモンが端末接続を維持して、トレース出力を書き出すようになります。こうした端末接続は、使用している端末エミュレーションの、Control-C などの割り込み文字を入力することによって打ち切ることができます。

注 – デバッグモードを無効にするには、デバッグレベルを 0 に戻します。

問題の診断

Sun Grid Engine, Enterprise Edition 5.3 システムには、問題の診断に役立つ報告を受ける手段がいくつか用意されています。以下では、それらの手段の使用方法を簡単に説明します。

保留中のジョブがディスパッチされない

保留中のジョブが実行可能な状態であることが明らかであるにもかかわらず、ディスパッチされないことがあります。Sun Grid Engine, Enterprise Edition 5.3 には、その理由を調べる手段として `qstat -j <ジョブ id>` と `qalter -w v <ジョブ id>` のユーティリティとオプションのペアがあります。

- `qstat -j <ジョブ id>`

`qstat -j <ジョブ id>` は、最後のスケジューリングで特定のジョブがディスパッチされなかった理由のリストを提供します。この監視機能は、schedd デーモンと qmaster との間の通信で望ましくないオーバーヘッドを生む可能性があるため、有効または無効にすることができます。以下は、id が 242059 のジョブに関する出力例です。

```
% 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"
```

この情報は、schedd デーモンによって直接生成され、クラスタの現在の利用状況が考慮されます。ただし、無関係の情報が得られることもあります。たとえば、他のユーザーのジョブによってすべてのキュースロットがすでに占有されている場合、問題のジョブに関する詳細なメッセージは生成されません。

- `qalter -w v <ジョブ id>`

このコマンドは、ドライスケジューリングを行うことによって、基本的にジョブがディスパッチ不可能な理由を一覧表示します。このドライスケジューリングが特別な点は、スロットを含めて消費可能なすべての資源がそのジョブ用に完全に利用可能であるとみなしてスケジューリングが行われることです。同様に、負荷は変化するため、すべての負荷値は無視されます。

ジョブまたはキューがエラー状態 E と報告される

`qstat` の出力では、ジョブまたはキューのエラーは大文字の `E` で示されます。ジョブがエラー状態になるのは、Sun Grid Engine, Enterprise Edition 5.3 システムがキュー内のジョブを実行しようとして、そのジョブに固有の理由で実行に失敗した場合です。キューがエラー状態になるのは、Sun Grid Engine, Enterprise Edition 5.3 システムがキュー内のジョブを実行しようとして、そのキューに固有の理由で実行に失敗した場合です。

Sun Grid Engine, Enterprise Edition 5.3 システムには、ジョブ実行エラーが発生した場合に、ユーザーおよび管理者がその診断情報を収集するための一群の機能が用意されています。キューおよびジョブのエラーの状態のどちらも、原因はジョブの実行失敗にあるため、診断結果はその両方のエラー状態に適用することができます。

- ユーザー宛て中止メール

  `submit` オプションの `-m a` を使用してジョブが実行依頼された場合は、`-M` *user*[`@`*host*] オプションで指定されたアドレスに中止メールが送信されます。ユーザー宛て中止メールには、ジョブのエラーに関する診断情報が含まれており、情報源として利用することを推奨します。

- `qacct` のアカウンティング情報

  中止メールが得られない場合、ユーザーは `qacct -j` コマンドを実行して、Sun Grid Engine, Enterprise Edition 5.3 システムのジョブアカウンティング機能からジョブのエラーに関する情報を入手することができます。

- 管理者宛て中止メール

  管理者は、適切な電子メールアドレスを指定することによってジョブ実行時の問題に関するメールを送信するよう指示することができます (`sge_conf(5)` の `administrator_mail` の下を参照)。管理者宛てのメールには、ユーザー宛ての中止メールよりも詳しい診断情報が含まれ、ジョブ実行エラーがよく発生する場合に利用することを推奨します。

- Messages ファイル:

  管理者宛てメールが得られない場合は、`qmaster` の `messages` ファイルをまず調べてください。適切なジョブ ID を検索することによって特定のジョブに関するログを得ることができます。デフォルトの設定でインストールした場合、`qmaster` の `messages` ファイルは、`$SGE_ROOT/default/spool/qmaster/messages` にあります。

ジョブの起動元の execd デーモンのメッセージに、補足情報が含まれていることもあります。qacct -j <**ジョブ** *id*> を使用してジョブの起動元のホストを確認し、$SGE_ROOT/default/spool/<**ホスト**>/messages で適切なジョブ ID を検索します。

よくある問題の解決

ここでは、よくある問題の原因の究明と解決に役立つ情報をまとめています。

- **問題** - ジョブの出力ファイルに「Warning: no access to tty; thus no job control in this shell...」というメッセージが出力される。
  - **原因** - 少なくとも 1 つのログインファイルに stty コマンドが含まれていることが考えられます。stty コマンドが役立つのは、端末が存在する場合だけです。
  - **対策** - Sun Grid Engine, Enterprise Edition 5.3 では、バッチジョブは端末に関連付けられません。ログインファイルから stty コマンドを削除するか、処理する前に端末の有無を確認する if 文で stty コマンドを囲ってください。以下は、この例です。

```
/bin/csh:
stty -g             # checks terminal status
if ($status == 0)  # succeeds if a terminal is present
< ここにすべての stty コマンドを入れる >
endif
```

- **問題** - ジョブの標準エラーログファイルに「'tty': Ambiguous」というメッセージが出力される。しかし、ジョブスクリプトで呼び出されるユーザーのシェルには、tty に対する参照はない。
  - **原因** - デフォルトでは、shell_start_mode は posix_compliant です。このため、すべてのジョブスクリプトは、ジョブスクリプトの先頭行に指定されたシェルではなく、キュー定義に指定されたシェルで実行されます。
  - **対策** - qsub コマンドに -S フラグを使用するか、shell_start_mode を unix_behavior に変更してください。
- **問題** - コマンド行から実行できるジョブスクリプトを qsub コマンドを使用して実行しようとすると、問題が発生する。
  - **原因** - ジョブに対するプロセス数が制限されている可能性があります。この確認をするには、limit および limit -h 関数を実行するテストスクリプトを作成し、シェルプロンプトから対話形式による方法と qsub コマンドを使用する方法の両方でテストスクリプトを実行し、結果を比較します。

- **対策** - 構成ファイルから、シェルで制限を設けるすべてのコマンドを削除してください。

- **問題** - 実行ホストから負荷 `99.99` が報告される。
  - **原因** - 3 つの原因が考えられます。

1. ホストで `execd` デーモンが動作していない。
2. デフォルトデーモンの指定に誤りがある。
3. `qmaster` ホストが認識している実行ホスト名と、その実行ホストが認識している自身の名前とが異なる。
   - **対策** - 原因によって、以下のいずれかの対策が考えられます (以下の対策の番号は上記の「原因」の番号と対応しています)。

1. 実行ホストで `root` になり、 `$SGE_ROOT/default/common/'rcsge'` スクリプトを実行することによって `execd` デーモンを起動する。
2. Sun Grid Engine, Enterprise Edition の管理者として `qconf -mconf` コマンドを実行し、`default_domain` 変数の値を `none` に変更する。
3. クラスタのホスト名の解決に DNS を使用している場合は、主ホスト名として絶対パスによるドメイン名 (FQDN) が返されるように `/etc/host` と NIS を構成する (このように構成しても、当然、`168.0.0.1 myhost.dom.com myhost` というように短い別名を定義、使用することができます) 。

   DNS を使用していない場合は、`/etc/hosts` のすべてのファイルと NIS テーブルに矛盾がないようにする (例: `168.0.0.1 myhost.corp myhost`、`168.0.0.1 myhost`)。

   問題 - 次のような警告が 30 秒おきに *<cell>*`/spool/<host>/messages` に出力される。

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

しかし、*<sell>*`/common/local_conf/` には、各ホスト用のファイルがあり、それぞれに FDQN が存在する。

- **原因** - 使用しているマシン `meta` では、ホスト名解決でショート名が返されるのに対し、マスターマシンでは、FDQN 付きの `meta` が返されます。
- **対策** - この点に関して、`/etc/hosts` のすべてのファイルと NIS テーブルの間に矛盾がないようにしてください。この例では、ホスト `meta` の `/etc/hosts` ファイルに次のような行が含まれている可能性があります。

  ```
  168.0.0.1 meta meta.your.domain
  ```

  正しくは、この行は次のようにします。

  ```
  168.0.0.1 meta.your.domain meta.
  ```

- **問題** - デーモンの messages ファイルに CHECKSUM ERROR や WRITE ERROR、READ ERROR というメッセージが出力されることがある。
  - **原因** - 一般にこれらのメッセージは、1 日に 1 回から 30 回の出力されます。1 秒間隔で出力されるのでない限り、何もする必要はありません。
- **問題** - ジョブが特定のキューで完了し、qmaster/messages に次のメッセージを返す。

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

しかし、実行ホストの exechost/messages ファイルには次のエラーメッセージが出力される。

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

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

  - **原因** - 自動マウントされる $SGE_ROOT ディレクトリがマウント解除されたために、sge_execd デーモンがその cwd を失った可能性があります。
  - **対策** - execd ホストにローカルのスプールディレクトリを使用してください。このためには、qmon または qconf を使用して、execd_spool_dir パラメータを設定します。
- **問題** - qrsh ユーティリティを使用して対話形式のジョブを実行依頼しようとすると、次のエラーメッセージが表示される。

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

  しかし、qsub ユーティリティを使用してバッチジョブに対してキューを使用可能にすることができ、qhost -l mem_free=1G および qstat -f -l mem_free=1G で照会できる。
  - **原因** - 「error: no suitable queues」というメッセージの原因は、qrsh などの対話形式のジョブに対してデフォルトで有効になる submit の -w e オプションにあります (grsh(1) の -w e を参照)。現在のクラスタ構成に従ってジョブがディスパッチ可能であるかどうかを qmaster が確実に判断できない場合、このオプションがあると、submit コマンドで問題が発生します。この仕組みの意図は、許可できないジョブ要求を事前に拒否することにあります。

- **対策** - この場合は、mem_free が消費可能な資源に設定されているにもかかわらず、そのホストで使用可能にするメモリーサイズが指定されていなかったことが原因です。メモリー負荷値はそれぞれに異なるため、この検査では、意図的に検討されず、このため、クラスタ構成で表示することはできません。この問題を解決するには、次のいずれかを行います。

  qrsh のデフォルト設定の -w e を無効にするか、明示的に -w n を使用して実行依頼することによって、この検査を省略する。この指定は、$SGE_ROOT/<*cell*>/common/cod_request で行うこともできます。

  mem_free を消費可能な資源として管理する場合は、qconf -me <**ホスト名**> を使用して、complex_values of host_conf(5) のホストに mem_free 値を指定する。

  mem_free を消費可能な資源として管理しない場合は、qconf -mc *host* を使用して、complex(5) の consumable 列で mem_free を消費不可資源に戻す。

- **問題** - qrsh が、自身が動作しているのと同じノードにディスパッチしない。このとき qsh シェルから 以下のメッセージが返される。

```
host2 [49]% qrsh -inherit host2 hostname
error: executing task of job 1 failed:

host2 [50]% qrsh -inherit host4 hostname
host4
```

  - **原因** - gid_range が十分ではないことが考えられます。1 つの数字ではなく、範囲を指定してください。Sun Grid Engine, Enterprise Edition 5.3 システムは、ホスト上の各ジョブに固有の gid を割り当てます。
  - **対策** - qconf -mconf または qmon グラフィカルユーザーインタフェースを使用して、gid_range を調整してください。推奨する範囲は以下のとおりです。

```
gid_range                    20000-20100
```

- **問題** - 並列ジョブ内で使用すると、qrsh -inherit -V が機能しないで、次のメッセージが返される。

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

  - **原因** - この問題は入れ子にされた qrsh 呼び出しで発生し、原因は -V スイッチにあります。最初の qrsh -inherit 呼び出しでは、環境変数 TASK_ID (並列ジョブ内で密統合されたタスクの ID) を設定します。2 つ目の qrsh -inherit

呼び出しでは、このタスクの登録に TASK_ID 環境変数を使用して、すでに実行中の最初のタスクと同じ ID を持つタスクを開始しようとするため、タスクの開始は失敗します。

- **対策** - qrsh -inherit を呼び出す前に TASK_ID を設定解除するか、-V スイッチではなく -v を使用して、実際に必要な環境変数だけエクスポートしてください。

- **問題** - qrsh がまったく機能していないように見えて、次のようなメッセージが返される。

```
host2$ qrsh -verbose hostname
local configuration host2 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
host2$
```

- **原因** - qrsh に対する権限が正しく設定されていない可能性があります。
- **対策** - $SGE_ROOT/utilbin/ にある次のファイルの権限を調べてください。(root が rlogin と rsh を setuid し、所有している必要があることに注意してください。)

```
-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*
```

注 – $SGE_ROOT ディレクトリも、setuid 付きで NFS マウントされている必要があります。実行依頼クライアントから nosuid でマウントされている場合、qrsh と関係するコマンドは機能しません。

- **問題** - 分散 make を起動しようとすると、次のエラーメッセージで qmake が終了する。

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

  - **原因** - Sun Grid Engine, Enterprise Edition 5.3 システムは、実行ホストで qmake のインスタンスを起動します。この Sun Grid Engine, Enterprise Edition 5.3 環境 (特に PATH 変数) がユーザーのシェルリソースファイル (.profile/.cshrc) に設定されていない場合、この qmake の呼び出しは失敗します。
  - **対策** - -v オプションを使用して、PATH 環境変数を qmake ジョブにエクスポートしてください。以下は、一般的な qmake の呼び出し例です。

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

- **問題** - qmake ユーティリティを使用する際、次のエラーメッセージが返される。

```
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.
```

  - **原因** - qmake の呼び出し元であるシェルに ARCH 環境変数が正しく設定されていない可能性があります。
  - **対策** - クラスタで使用可能なホストに一致するサポート値を ARCH 変数に設定するか、実行依頼時に適切な値を指定してください (例: qmake -v ARCH=solaris64 ...)。