



# Sun™ Grid Engine, Enterprise Edition 5.3 管理基本マニュアル

---

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

Part No. 816-7481-10  
2002 年 7 月, Revision A

コメントの宛先: [docfeedback@sun.com](mailto:docfeedback@sun.com)

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします)は、本書に記述されている製品に採用されている技術に関する知的所有権を有しています。これら知的所有権には、<http://www.sun.com/patents>に掲載されているひとつまたは複数の米国特許、および米国ならびにその他の国におけるひとつまたは複数の特許または出願中の特許が含まれています。

本書およびそれに付随する製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および本書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品のフォント技術を含む第三者のソフトウェアは、著作権法により保護されており、提供者からライセンスを受けているものです。

本製品のの一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

本製品は、株式会社モリサワからライセンス供与されたリュウミン L-KL (Ryumin-Light) および中ゴシック BBB (GothicBBB-Medium) のフォント・データを含んでいます。

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェースマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun, Sun Microsystems, AnswerBook2, docs.sun.com は、米国およびその他の国における米国 Sun Microsystems 社の商標もしくは登録商標です。サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

ATOK は、株式会社ジャストシステムの登録商標です。ATOK8 は、株式会社ジャストシステムの著作物であり、ATOK8 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。ATOK Server/ATOK12 は、株式会社ジャストシステムの著作物であり、ATOK Server/ATOK12 にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPENLOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザーインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the Sun Microsystems, Inc. license agreements and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1998), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本書には、技術的な誤りまたは誤植のある可能性があります。また、本書に記載された情報には、定期的に変更が行われ、かかる変更は本書の最新版に反映されます。さらに、米国サンまたは日本サンは、本書に記載された製品またはプログラムを、予告なく改良または変更することがあります。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典:	Sun™ Grid Engine, Enterprise Edition 5.3 Basics of Administration Part No: 816-7409-11 Revision A
-----	---



Adobe PostScript

# 目次

---

グリッドコンピューティングと Sun Grid Engine, Enterprise Edition 5.3 ソフトウェア	2
分散資源管理の役割	3
ポリシー、ホスト、デーモン	4
コマンド行からの主な管理機能の使用方法	6
▼ ホストからの管理特権の追加または削除	6
▼ 実行ホストの追加	6
▼ 実行ホストの削除	7
▼ 実行依頼ホストの追加または削除	7
▼ 現在のホスト名の表示	7
▼ キューの管理	8
▼ マスターホストの変更	8
▼ シャドウマスターの設定	9
スクリプトとファイルを使用した管理業務の実施方法	10
▼ ファイルを使用したオブジェクトの追加または変更	10
▼ ファイルを使用したキュー、ホスト、環境の変更	11
▼ ファイルを使用したグローバル構成またはスケジューラの変更	15
▼ グリッド環境の調整	17
ポリシー	19
業務優先ポリシーの構成	19

- ▼ ユーザーに対する業務優先スケジュールを作成する 19
- ▼ プロジェクトに対する業務優先スケジュールを作成する 20
- ▼ 部署に対する業務優先スケジュールを作成する 21

#### 基本割当ポリシーの構成 21

- ▼ プロジェクトに対する基本割当スケジューリングを作成する - プロジェクト別に FCFS 処理する場合 21
- ▼ プロジェクトに対する基本割当スケジューリングを作成する - 全ユーザーに同じ配分をする場合 22
- ▼ プロジェクトに対する基本割当スケジューリングを作成する - プロジェクト別にユーザーごとの配分をする場合 23

#### よくある問題の解決 24

##### 問題の診断 30

- 保留中のジョブがディスパッチされない 30
- ジョブまたはキューがエラー状態 E と報告される 31

# Sun Grid Engine, Enterprise Edition

## 5.3 管理の基本

---

このマニュアルは、Sun Grid Engine, Enterprise Edition 5.3 システムの管理者向けに要点を簡潔にまとめた手引き書です。主な管理機能の使用方法ばかりでなく、グリッドコンピューティング全般、Sun Grid Engine, Enterprise Edition 5.3 製品そのものについても簡単に説明しています。また、マニュアルの最後では、調整や問題解決のためのヒントも取り上げています。

このマニュアルは、Sun Grid Engine, Enterprise Edition 5.3 の概念と使用手順を詳細に説明している『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』に取って代わるものではなく、あくまで補足としてご利用ください。

具体的には、このマニュアルでは以下の管理機能を使用する方法を説明します。

- 6 ページの「ホストからの管理特権の追加または削除」
- 6 ページの「実行ホストの追加」
- 7 ページの「実行ホストの削除」
- 7 ページの「実行依頼ホストの追加または削除」
- 7 ページの「現在のホスト名の表示」
- 8 ページの「キューの管理」
- 8 ページの「マスターホストの変更」
- 9 ページの「シャドウマスターの設定」
- 10 ページの「ファイルを使用したオブジェクトの追加または変更」
- 11 ページの「ファイルを使用したキュー、ホスト、環境の変更」
- 15 ページの「ファイルを使用したグローバル構成またはスケジューラの変更」
- 17 ページの「グリッド環境の調整」
- 19 ページの「ユーザーに対する業務優先スケジュールを作成する」
- 20 ページの「プロジェクトに対する業務優先スケジュールを作成する」
- 21 ページの「部署に対する業務優先スケジュールを作成する」
- 21 ページの「プロジェクトに対する基本割当スケジューリングを作成する - プロジェクト別に FCFS 処理する場合」
- 22 ページの「プロジェクトに対する基本割当スケジューリングを作成する - 全ユーザーに同じ配分をする場合」
- 23 ページの「プロジェクトに対する基本割当スケジューリングを作成する - プロジェクト別にユーザーごとの配分をする場合」

- 24 ページの「よくある問題の解決」
- 30 ページの「問題の診断」

---

注 - このマニュアルの情報の多くは、元々、Sun Grid Engine プロジェクト Web サイトの「How-To」セクションに掲載されていたものです。この Web サイトは、Sun Grid Engine, Enterprise Edition 5.3 システムの管理者にとって特に有用な情報を提供しており、頻繁に更新されています。この Web サイトの URL は次のとおりです。  
<http://gridengine.sunsource.net/project/gridengine/howto/howto.html>

---

## グリッドコンピューティングと Sun Grid Engine, Enterprise Edition 5.3 ソフトウェア

コンピュータ管理の専門家であっても、グリッドコンピューティング管理の経験がある方はまずいないでしょう。グリッドコンピューティングに対応するソフトウェア技術の歴史は 10 年にもなりませんから、真の意味でグリッドコンピューティング環境の管理経験があると主張できる人は、いたとしてもごくわずかです。グリッドコンピューティング分野の先駆者の 1 人である Ian Foster 博士は、グリッドコンピューティングに関する概念の「探究」は 1995 年頃に始まったばかり、と述べています。つまり、グリッド技術を最も早くから採用していたとしても、その経験は 10 年にもなりません。

しかし、グリッドコンピューティングの経験がどれほどわずかであったとしても、別の種類ではありますが、私たちにはグリッドを使用した経験があります。電気のスイッチを「入れる」たびに、私たちはグリッドの資源を利用しています。工業先進国では、グリッドと呼ばれる共同資源プールにその製品 (電力) を供給する、独立した多数の電力会社 (電力の生産者) から構成される「電力グリッド」を通じて家庭や企業に電力が供給されます。

この資源プールは、消費者と生産者の両方にメリットがあります。電力の生産者は、地理的に隣接した地域の消費者がその発電所が生産できるほど多くの電気を必要としなくても、発電所をフル操業できるメリットがあります。これに対し、たとえば都市から遠く離れた地域の消費者は、一般には、通常発電所が生産できるほど多くの電力を必要としません。しかし、電力グリッドの取り決めによって、発電所は「余分な」生産能力をグリッドに販売し、こうして、地元の発電所では需要を賅えない遠く離れた地域の消費者にサービスを提供することができます。

需要がどこにあるかに関係なく、需要を満たすために、十分に活用されていない資源を生産に回して供給できるという能力が電力グリッドの特性を明らかにしています。「グリッド」という名前を冠することによって、グリッドコンピューティングの概念の先駆者は適切なアナロジーを生み出しました。グリッドコンピューティングの根本目標は、ほぼまたは完全に休止している数百、数千のコンピュータという、十分に活用されていない資源をプールし、結合されたコンピューティングパワーを最大限に利用できるようにすることによって、集約的な計算能力を必要とする消費者の需要を満たせるようにすることにあります。

## 分散資源管理の役割

このアナロジーの少し異なる点は、供給の細部の部分においてです。電力グリッドは、文字通りグリッドの全生産者メンバー、そして最終的には、電気の消費者である家庭、工場、農場、事務所のすべてを電線で結ぶことによって資源をプールします。電線で結ばれ、電力グリッドに供給された電気は、需要に従って流れていきます。たとえば照明のスイッチを「入れる」と、ある場所の発電所から電力グリッドを経由し、電球に電気が流れます。グリッドコンピューティング環境におけるコンピューティング資源もまた、電線ではなくネットワーク技術によってですが、「結ばれる」ということができます(この最大の例はインターネットですが、イントラネットによっても結ばれます)。しかし、ネットワークコンピュータのスイッチを入れただけでは、コンピューティンググリッドの資源を利用したり、コンピューティンググリッドに資源を奉仕させたりすることはできません。

コンピューティンググリッドは、特別な**ミドルウェア**ソフトウェアをインストールしてはじめて利用できるようになります。ミドルウェアは、Sun の Solaris™ オペレーティング環境や Linux オペレーティングシステムなどのコンピュータのオペレーティングシステムと、三次元グラフィックスプログラムなどのアプリケーションソフトウェアとの間に位置するソフトウェア階層です。そして、Sun Grid Engine ソフトウェアは、コンピューティンググリッドへの参加を可能にするミドルウェアです。

初期の「オープンソース」のグリッドコンピューティングプロジェクトである Sun Grid Engine の明白な成果は、しばしば遠く離れた場所に位置する、複数のユーザーや計算資源間の調停をして、複雑なジョブ用にはるかに強力な計算能力を提供する**分散資源管理**ソフトウェアです。Sun Grid Engine ソフトウェアを介して、休止しているコンピューティング資源がグリッドに結合して、遠隔地のユーザーから利用できるようになり、ユーザーと資源の両方の生産性が向上します。たとえば普段は実際の能力の 4 分の 1 の力で動作していたコンピューティング資源が、グリッドに参加することによってほぼ 100% の力で動作するようになることが明らかになっています。

計算デバイスは、「分散資源管理」という用語の「分散資源」にあたります。Sun Grid Engine は、この用語の必要不可欠な部分、「管理」にあたります。こうした管理がなければ、大混乱をきたすことでしょう。グリッド管理者として Sun Grid Engine ソフトウェアはユーザーから実行依頼されたジョブを受け付け、組織の技術および管理スタッフによって決められた**資源管理ポリシー**に基づいて、適切なシステムで実行されるようそのジョブをスケジューリングします。

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した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簡単なグリッドです。

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアは、この基本モデルを、残る2つのより複雑で強力なクラスのグリッドに拡張します。

- **構内グリッド** - 同じ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** - 構内グリッドを組織の枠を越えて結んで、非常に大規模な仮想システムを実現するグリッドです。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手にすることができます。

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、基本 Sun Grid Engine ソフトウェア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、企業が初めて構内グリッドという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。

## ポリシー、ホスト、デーモン

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアは、システム管理者が管理する社内資源利用ポリシーに基づいて計算パワーの供給を調整します。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、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の間で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。

管理者は、現場に合った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。こうしたポリシーについては、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』で詳しく説明しています。

- 業務優先
- 基本割当
- 締め切り優先



- 一時優先

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、その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れらジョブの相対的な優先順位を調整します。

Sun Grid Engine, Enterprise Edition 5.3 のポリシーモジュールは、現場の緩やかな資源利用規則あるいは厳密な利用規則のどちらにも対応し、ポリシーに基づいて、週、月、四半期などの累積期間中に各ユーザー、チーム、部署、プロジェクトに事前に決められた配分でシステム資源を付与します。

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

また、これらのホストのほかにシャドウマスターホストも作成することを推奨します。クラスタ内のマスターホストは 1 つだけですが、他のマシンをシャドウマスターホストとして指定することによって、可用性を高めることができます。シャドウマスターホストはマスターホストを絶えず監視し、マスターホストで問題が発生した場合に自動的にそしてトランスペアレントに制御を受け取ります。この結果、マスターホストに問題が発生しても、クラスタ内のジョブがその影響を受けることはありません。

これらのホストについては、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』で詳しく説明しています。

Sun Grid Engine, Enterprise Edition 5.3 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
- `sge_schedd` - スケジューラデーモン
- `sge_execd` - 実行デーモン
- `sge_commd` - 通信デーモン

これらのデーモンについては、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』で詳しく説明しています。

以下では、Sun Grid Engine, Enterprise Edition 5.3 の主な管理機能の使用方法を説明します。他の予備知識および、このマニュアルでは取り上げていない機能の使用方法については、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』、『Sun Grid Engine 5.3 and Sun Grid Engine, Enterprise Edition 5.3 リファレンスマニュアル』、マニュアルページを参照してください。

---

# コマンド行からの主な管理機能の使用方 法

---

注 - ここでは、コマンド行からコマンドを使用して管理業務を行う方法を機能別にグループ分けして説明します。Sun Grid Engine, Enterprise Edition 5.3 のグラフィカルユーザーインターフェースの QMON には、これらのコマンドに対応する機能がすべて用意されています。QMON を使用して管理業務を行う方法については、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』を参照してください。

---

## ▼ ホストからの管理特権の追加または削除

- 管理特権を追加するには、次のコマンドを入力します。

```
qconf -ah
```

- 管理特権を削除するには、次のコマンドを入力します。

```
qconf -dh
```

## ▼ 実行ホストの追加

1. 次のコマンドを入力することによって新しいホストを管理ホストにします。

```
qconf -ah
```

2. 管理ホストにしたホスト上でスーパーユーザーになって、\$SGE\_ROOT から次のスクリプトを実行します。

```
install_execd
```

## ▼ 実行ホストの削除

- 次のコマンドを入力することによって、実行ホストに関連づけられているキューを削除します。

```
qconf -dq
```

- 次のコマンドを入力することによって実行ホストを削除します。

```
qconf -de
```

## ▼ 実行依頼ホストの追加または削除

- ホストを実行依頼ホストに指定するには、そのホストのコマンド行で次のコマンドを入力します。

```
qconf -as
```

- ホストの実行依頼ホスト指定を削除するには、そのホストのコマンド行で次のコマンドを入力します。

```
qconf -ds
```

## ▼ 現在のホスト名の表示

- 管理ホストの名前を表示するには、次のコマンドを入力します。

```
qconf -sh
```

- 実行依頼ホストの名前を表示するには、次のコマンドを入力します。

```
qconf -ss
```

- 実行ホストの名前を表示するには、次のコマンドを入力します。

```
qconf -sel
```

## ▼ キューの管理

- キューを追加するには、次のコマンドを入力します。

```
qconf -aq
```

- ファイルからキューを追加するには、次のコマンドを入力します。

```
qconf -Aq
```

- キューを削除するには、次のコマンドを入力します。

```
qconf -dq
```

- キューを変更するには、次のコマンドを入力します。

```
qconf -mq
```

- 複数のキューの同じ属性を変更するには、次のコマンドを入力します。

```
qconf -mqattr
```

## ▼ マスターホストの変更

1. 次のコマンドを入力することによって現在のマスターホスト上のマスターデーモンとスケジューラデーモンを停止します。

```
qconf -ks -km
```

2. 次の手順に従って `$SGE_ROOT/default/common/act_qmaster` ファイルを編集します。

a. `act_qmaster` ファイル内の現在のホスト名を新しいマスターホスト名に置き換えます。

新しいホスト名には、`gethostname` ユーティリティによって返されたのと同じ名前を使用します。この名前を調べるには、新しくマスターホストにするホスト上で次のコマンドを入力します。

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

b. `act_qmaster` ファイル内の古い名前を、`gethostname` ユーティリティによって返された名前に置き換えます。

3. 新しいマスターホスト上で、次のスクリプトを実行します。

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

このスクリプトによって、新しいマスターホスト上で `sge_qmaster` と `sge_schedd` が起動されます。

## ▼ シャドウマスターの設定

1. 次の手順に従って `shadow_masters` ファイルを作成します。

- `$SGE_ROOT/default/common` に `shadow_masters` ファイルを作成します。
- 1 行目に主マスターホスト名を入力し、2 行目以降にマスターの役割を引き継がせる他のホストを、望ましい順番で列挙します。

例:

```
system% cat shadow_masters
host1
host2
host3
```

この例では、`host1` が主マスターホストです。`host1` で問題が発生すると、約 10 分の時間の経過後、`host2` がマスターホストを引き継ぎます。さらに、`host2` でも問題が発生した場合は、`host3` が引き継ぎます。

## 2. 適切な権限があることを確認します。

すべてのマスターシャドウホストが、qmaster スプールディレクトリに対する読み取り・書き込み権限を持っている必要があります。

## 3. 各ホストで sge5 起動スクリプトを実行することによってシャドウデーモンを起動します。このためには、スーパーユーザーになって次のコマンドを入力します。

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

この手順を完了すると、Sun Grid Engine, Enterprise Edition 5.3 クラスタに対するマスターシャドウ機能がアクティブになります。

# スクリプトとファイルを使用した管理業務の実施方法

---

注 – Sun Grid Engine, Enterprise Edition 5.3 での管理業務はすべて、QMON グラフィカルユーザーインターフェースを使用して行うことができます。また、QMON を使用することには、システムのすべての機能を学べるというメリットもあります。しかし、そうした管理業務は、シェルプロンプトでコマンドを発行したり、シェルスクリプトからコマンドを呼び出したりすることによって行うこともできます。経験を積んだ多くの管理者は、設定変更方法としては、柔軟性やスピード、強力さの点でこの方が勝っていることに気がきます。以下では、この方法を使用した手順を 3 つ紹介します。

---

## ▼ ファイルを使用したオブジェクトの追加または変更

- qconf コマンドを使用し、ファイルに指定された内容に従ってオブジェクトを追加または変更します。

qconf コマンドの構文: qconf -{A,M}<オブジェクト> <ファイル名>

上記の構文の -A は追加、-M は変更を意味します。<オブジェクト> 変数には以下のどれでも指定できます。

- c - 複合
- ckpt - チェックポイント環境
- e - 実行ホスト
- p - 並列環境
- q - キュー

## ■ u - ユーザーセット

このオプションを `qconf` コマンドの `show` オプションと組み合わせることによって (`qconf -s<オブジェクト>`)、既存のオブジェクトを受け取って、変更、更新したり、新しいオブジェクトを作成したりできます。

## 例

以下は、既存のチェックポイント環境の移動コマンドを変更するシェルスクリプトの例です。

```
#!/bin/sh
# ckptmod.sh: modify the migration command
# of a checkpointing environment
# Usage: ckptmod.sh <チェックポイント環境名> <コマンドのフルパス>
TMPFILE=/tmp/ckptmod.$$

CKPT=$1
MIGMETHOD=$2

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

## ▼ ファイルを使用したキュー、ホスト、環境の変更

コマンド行からキューやホストを個別に変更したり、並列およびチェックポイント環境の両方を変更したりできます。このためには、コマンドの引数に従って2つある方法のうちの1つで `qconf` コマンドを使用したり、他のコマンドと `qconf` コマンドを組み合わせたりします。以下では、この両方のケースを説明します。

### ● 変更用のファイルがすでにある場合

この場合のコマンドの構文は次のとおりです。文字オプションの大文字と小文字の違いに注意してください。この違いに重要な意味があります。

```
qconf -M{q,e,p,ckpt} <ファイル名>
```

オプションには次の意味があります。

- -M - 既存のファイルの <ファイル名> から変更します。
- q - キュー
- e - 実行ホスト
- p - 並列環境

- ckpt - チェックポイント環境

## ● 変更用のファイルがない場合

この場合のコマンドの構文は次のとおりです。文字オプションの大文字と小文字の違いに注意してください。この違いに重要な意味があります。

```
qconf -m{q,e,p,ckpt} <オブジェクト名>
```

オプションには次の意味があります。

- -m - テキストエディタを開き、変更用のファイルを作成することによって変更します。
- q - キュー
- e - 実行ホスト
- p - 並列環境
- ckpt - チェックポイント環境

## 説明

最初のコマンド (大文字の M) と 2 つ目のコマンド (小文字の m) の違いは、コマンドの結果に影響します。-M および -m オプションはともに変更を意味しますが、大文字の M が既存のファイルからの変更を意味するのに対し、小文字の m はそうではありません。小文字の -m はエディタで一時ファイルを開き、そのファイルに変更内容を保存して、エディタを終了すると、ただちにその変更がシステムに反映されます。

ただし、多数のオブジェクトを一度に変更するか、非対話形式でオブジェクト構成を変更する場合は、「qconf -...attr」コマンドセットを使用します。

一方のコマンドセット形式では、**ファイル内**の指定内容に従って変更を加えます。

```
qconf -{A,M,R,D}attr queue|exechost|pe|ckpt <ファイル名>
```

もう一方のコマンドセット形式では、**コマンド行**の指定内容に従って変更を加えます。

```
qconf -{a,m,r,d}attr queue|exechost|pe|ckpt <属性> <値> <キューリスト>|<ホストのリスト>
```

両方のコマンドセットのオプションには以下の意味があります。

- -A/a - 属性を追加
- -M/m - 属性を変更
- -R/r - 属性を置換
- -D/d - 属性を削除
- <属性> - 変更するキューまたはホスト属性



- <値> - 設定する属性値
- <ファイル名> - 属性とその値の対を含むファイル

a、m、d オプションでは、値リストの値を個別に操作するのに対し、r オプションは値リスト全体を、コマンド行またはファイルに指定された新しいリストに置き換えます。

## 例

- tcf27-e019.q のキュータイプをバッチ専用に変更します。

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

- ファイル new.cfg の内容に従って tcf27-e019.q のキュータイプとシェル起動時の動作を変更します。

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

- storage と license という名前の複合をホスト tcf27-e019 に接続します。

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

- scratch1 と long という名前の資源を、それぞれ値 1000M、値 2 で追加します。

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

- short という名前の資源を値 4 で ホストに追加します。

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

- scratch1 の値を 500M に変更します。他の値は変更しません。

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

- 資源 long を削除します。

```
% qconf -dattrib execest complex_values long tcf27-e019
```

- チェックポイント環境 sph 用のキューリストに tcf27-b011.q を追加します。

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

- 並列環境 make のスロット数を 50 に変更します。

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

## qselect コマンドを使用したキューの特定

qselect コマンドはキューのリストを出力します。オプションを指定して呼び出されると、その指定に一致するキューだけを出力します。このため、qconf -...attr コマンドセットとこのコマンドを組み合わせてすることによって、変更対象のキューを特定することができます。

### 例

- Linux マシン上のすべてのキューを一覧表示：

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

- 2 つの CPU が搭載されたマシンのすべてのキューを一覧表示：

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

- CPU が 4 つのすべての Solaris マシン上のすべてのキューを一覧表示：

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

- アプリケーションライセンスを提供するキューを一覧表示 ( 事前に設定されていること ):

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

qselect と qconf を組み合わせることによって、コマンド行 1 行で広範囲の変更を加えることができます。このためには、qconf コマンド行の <キューリスト> の代わりに全体を円記号 (¥) で囲った qselect コマンドを使用します。

## 例

- Solaris マシンのすべてのキューで prolog スクリプトを sol\_prolog.sh に変更する。

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

- プロセッサ 2 つのシステムのすべてのキューで fluent\_license 属性を 2 に変更する。

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

qconf コマンドと qselect コマンドの組み合わせは、独自のカスタム管理スクリプトを作成して、Sun Grid Engine, Enterprise Edition 5.3 キューの構成を自動化する最も柔軟な手段です。

## ▼ ファイルを使用したグローバル構成またはスケジューラの変更

- グローバル構成を変更するには、次節の説明に従って qconf -mconf コマンドを使用します。
- スケジューラを変更するには、次節の説明に従って qconf -msconf コマンドを使用します。

## 説明

これらのコマンドとともに、エディタで一時ファイルを開きます。エディタを終了すると、その一時ファイルに保存した変更内容が処理され、ただちに有効になります。一時ファイルが開かれるエディタプログラムは、EDITOR 環境変数に指定されたエディタです。この環境変数が定義されていない場合は、vi が使用されます。

EDITOR 環境変数を活用することによって、qconf -m... コマンドの動作を自動化することができます。コマンドの最初の引数で指定した名前のファイルを変更するプログラムを参照するように、この変数の値を変更してください。このプログラムが一時ファイルを変更して終了すると、その変更内容がただちにシステムに読み込まれて、反映されます。

---

**注** – 編集操作の終了後にファイルの修正日時が変更されていない場合は、ファイルが変更されていないとみなされることがあります。この誤りを防ぐには、ファイルを書き込む前に sleep 1 命令を挿入して、修正日時が変更されるようにしてください。

---

## 例

次の例のスクリプトは、スケジューラのスケジュール間隔を変更します。

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

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

このスクリプトは自身を参照するように EDITOR 環境変数の設定を変更して、qconf -msconf コマンドを呼び出します。2 つ目の入れ子の呼び出しでは、最初の引数に指定された一時ファイルが変更され、そこで終了します。Sun Grid Engine, Enterprise Edition 5.3 システムはこの変更内容を自動的に読み込み、それでスクリプ

トの最初の呼び出しが終了します。この方法は、あらゆる `qconf -m...` コマンドで利用することができます。ただし、この方法が特に役立つのは、これ以外に自動化する方法のない、スケジューラとグローバル構成の管理においてです。

## ▼ グリッド環境の調整

Sun Grid Engine, Enterprise Edition 5.3 は、あらゆる機能を装備した汎用の分散資源管理 (DRM) ツールです。このシステムのスケジューラコンポーネントは、さまざまな種類の広範囲のコンピュータセンター環境をサポートします。実際のコンピューティング環境で最高のパフォーマンスを得るには、どの機能を使用可能にして、負荷管理の問題を解決するためにどの機能が本当に必要かを調査することを推奨します。一部の機能を使用不可にするによって、クラスタのスループットのパフォーマンスが向上することがあります。

### スケジューラ監視

スケジューラ監視は、ジョブがディスパッチされない理由を調べる際に役立ちます。ただし、すべてのジョブに関していつでもこの情報が提供されるようにすると、資源が大量に消費されることがあり、通常、そのようにする必要はありません。

- スケジューラ監視を無効にするは、スケジューラ構成 `sched_conf (5)` の `schedd_job_info` を `false` に設定します。

### 完了したジョブ

配列ジョブの場合、`qmaster` 内の完了したジョブリストが非常に大きくなる場合があります。完了したジョブのリストは、`qstat` プロセスもフェッチするため、この機能を無効にすると、メモリーの節約になるばかりでなく、`qstat` プロセスの動作が速くなります。

- 完了したジョブリスト機能を無効にするには、グローバル構成 `sge_conf (5)` の `finished_jobs` を `0` に設定します。

### ジョブ検証

実行依頼時の強制的なジョブ検証は、ディスパッチ不可能なジョブがずっと保留状態のまま置かれるのを防止するための有効な手段になります。しかし、さまざま実行ノードと消費可能な資源を持ち、個々のユーザーが専用のジョブファイルを持つ異機種環境では特に、ジョブ検証は時間のかかる作業です。ジョブが 2, 3 個しかない同一機種環境では、通常、一般的なジョブ検証は省略することができます。

- ジョブ検証を無効にするには、クラスタ全体のデフォルト要求に `qsub(1)` オプションの `-wn` を追加します (`sge_request(5)` を参照)。

## 負荷しきい値と一時停止しきい値

負荷しきい値は、意図的にマシンに過度の実行依頼をしたときにシステムに多大の負荷がかかるのを防止する仕組みが必要な場合に使用します。一時停止しきい値もまた、この目的のために使用します。負荷しきい値が必要となるもう 1 つのケースは、Sun Grid Engine, Enterprise Edition 5.3 の管理下でない対話形式の負荷のための実行ノードが開いたままになっていて、そのノードが過負荷になるのを防止したい場合です。

たとえば計算ノードの各 CPU が単一のキューロットで表され、そうしたノードでは対話形式の負荷は発生しないというように、コンピュータセンターがほぼ単一目的の用途に限定される場合は、`load_thresholds` を省略することができます。

- 負荷および一時停止両方のしきい値を無効にするには、`load_thresholds` と `suspend_thresholds` をそれぞれ `none` に設定します (`queue_conf(5)` を参照)。

## 負荷調整

負荷調整は、ジョブのディスパッチ後、測定された負荷を実質的に大きくする目的に使用されます。この仕組みは、過度の実行依頼があるマシンを負荷しきい値に合わせる場合に有用です。ただし、ホストのソートや負荷しきい値の確認に関連してスケジューラに余分な負荷がかかりますから、必要ない場合は、負荷調整を無効にしてください。

- 負荷調整を無効にするは、スケジューラ構成 `sched_conf(5)` で `job_load_adjustments` を `none`、`load_adjustment_decay_time` を 0 に設定します。

## 要求に応じたスケジューリング

デフォルトでは、Sun Grid Engine, Enterprise Edition 5.3 は、一定のスケジューリング間隔でスケジューリング実行を開始します (`schedd_conf(5)` の `schedule_interval` を参照)。一定の間隔で開始することの利点は、`qmaster/scheduler` の CPU 時間の消費が押さえられることです。この逆に欠点は、スケジューラの動作が遅くなり、この結果スループットが向上しないことです。多くのコンピュータセンターでは `qmaster/scheduler` 専用のマシンを用意しており、そうした環境では、スケジューラの動作を遅くする理由はありません。

- 要求に応じてスケジューリングを設定するには、グローバルクラスタ構成 `sge_conf(5)` の `schedd_params` セクションの `FLUSH_SUBMIT_SEC` および `FLUSH_FINISH_SEC` 設定を使用します。

要求に応じてスケジューリングを有効にした場合、コンピュータセンターのスループットを制限するのは、`qmaster/scheduler` のホストになっているマシンのパワーだけになります。

---

## ポリシー

Sun Grid Engine, Enterprise Edition 5.3 ソフトウェアの特に重要な特長の 1 つに、システム管理者によって決められたポリシーに基づいて計算パワーを分配するという考え方があります。この考え方に関する予備的な情報については、4 ページの「ポリシー、ホスト、デーモン」を参照してください。

この節では、実際に基本的な「業務優先 (Funcitonal)」および「基本割当 (Share-based)」ポリシーの設定をする方法を説明します。「締め切り優先 (Deadline)」および「一時優先 (Override)」ポリシーの設定方法と、他の機能優先および基本割当の設定方法については、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』を参照してください。

### 業務優先ポリシーの構成

この節で紹介する構成では、メインポリシーとして業務優先 (Funcitonal) ポリシーを使用します。この種の構成を行うことによって、事前の配分がいつでも各ユーザー、プロジェクト、部署に確保されるようになります。すなわち、割り当てられている資源が予定より少ないユーザー、プロジェクト、部署のジョブは、休止している資源へのジョブのディスパッチで優先されます。

同時に、資源を必要とするユーザー、プロジェクト、部署の間で配分の未使用分が分配されるため、完全な資源利用が保証されます。過去の資源消費は考慮されません。

## ▼ ユーザーに対する業務優先スケジュールを作成する

この構成の目的は、Sun Grid Engine, Enterprise Edition 5.3 クラスタにまとめられている全資源のうちの一定の割合をさまざまなユーザーに配分することにあります。同じユーザーの複数のジョブに対しては FCFS スケジューリングが行われます。

1. グローバル構成 (`sge_conf(5)`) の `schedd_params` セクションで、`SHARE_FUNCTIONAL_SHARES=1` を使用します。
2. スケジューラ構成 (`sched_conf(5)`) で業務優先チケット数 (たとえば 1000000) を指定します。
3. スケジューリング対象のユーザーそれぞれに 1 ユーザー分 (`user(5)`) 追加します。
4. 各ユーザーに対する業務優先配分を割り当てます。  
全体に対する割合を示す百分率で配分を割り当ててください。

例:

- ユーザー A (10)
- ユーザー B (20)
- ユーザー C (20)
- ユーザー D (50)

## ▼ プロジェクトに対する業務優先スケジュールを作成する

この構成の目的は、Sun Grid Engine, Enterprise Edition 5.3 クラスタにまとめられている全資源のうちの一定の割合をさまざまなプロジェクトに配分することにあります。同じプロジェクトの複数のジョブに対しては FCFS スケジューリングが行われず。

1. グローバル構成 (`sge_conf(5)`) の `schedd_params` セクションで、`SHARE_FUNCTIONAL_SHARES=1` を使用します。
2. スケジューラ構成 (`sched_conf(5)`) で業務優先チケット数 (たとえば 1000000) を指定します。
3. スケジューリング対象のプロジェクトのそれぞれに 1 プロジェクト分 (`project(5)`) 追加します。  
`acl` および `xacl` 関数を使用して、特権レベルが高いプロジェクトに対するアクセスを制御することができます。これらの関数については、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』または『Sun Grid Engine 5.3 and 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。
4. 各プロジェクトに対する業務優先配分を割り当てます。  
全体に対する割合を示す百分率で配分を割り当ててください。

例:

- プロジェクト A (55)
- プロジェクト B (45)



## ▼ 部署に対する業務優先スケジュールを作成する

この構成の目的は、Sun Grid Engine, Enterprise Edition 5.3 クラスタにまとめられている全資源のうちの一定の割合をさまざまな部署に配分することにあります。同じ部署の複数のジョブに対しては FCFS スケジューリングが行われます。

1. グローバル構成 (`sge_conf(5)`) の `schedd_params` セクションで、`SHARE_FUNCTIONAL_SHARES=1` を使用します。
2. スケジューラ構成 (`sched_conf(5)`) で業務優先チケット数 (たとえば 1000000) を指定します。
3. スケジューリング対象の部署を 1 つ 1 つ追加します。
4. 各部署に対する業務優先配分を割り当てます。  
全体に対する割合を示す百分率で配分を割り当ててください。

例:

- 部署 A (90)
- 部署 B (5)
- 部署 C (5)

## 基本割当ポリシーの構成

この節で紹介する構成では、メインポリシーとして基本割当 (Share-tree) ポリシーを使用します。この種の構成を行うことによって、一定の時間の間に基本割当に設定されたインスタンスに事前の配分が確実に確保されるようになります。すなわち、過去の資源消費が予定 (基本割当配分) より少なかった基本割当ツリーの「枝」のジョブは、休止している資源へのジョブのディスパッチで優先されます。同時に、配分の未使用分が他の基本割当ツリーの枝の保留中のジョブに使用できるため、完全な資源利用が保証されます。

## ▼ プロジェクトに対する基本割当スケジュールを作成する - プロジェクト別に FCFS 処理する場合

この構成の目的は、Sun Grid Engine, Enterprise Edition 5.3 クラスタにまとめられている全資源のうちの一定の割合が一定の期間の間にさまざまなプロジェクトに確保されるようにすることにあります。同じプロジェクトの複数のジョブに対しては FCFS スケジューリングが行われます。

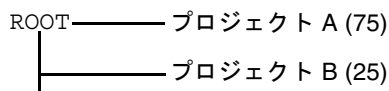
---

注 - このマニュアルの他の節と異なり、この節では、一部、qmon グラフィカルユーザーインターフェースを使用する必要があります。qmon に関する予備知識および使用方法については、『Sun Grid Engine, Enterprise Edition 5.3 管理マニュアル』を参照してください。

---

1. スケジューラ構成 (sched\_conf (5)) で基本割当チケット数 (たとえば 1000000) を指定します。
2. スケジューリング対象のプロジェクトのそれぞれに 1 プロジェクト分 (project (5)) 追加します。
3. qmon グラフィカルユーザーインターフェースを使用して、すべてのスケジューリング対象プロジェクト (ノード) の構成を反映した基本割当ツリーを作成します。
4. 各プロジェクトに基本割当配分を割り当てます。

以下は、簡単な基本割当ツリーの構成例です。



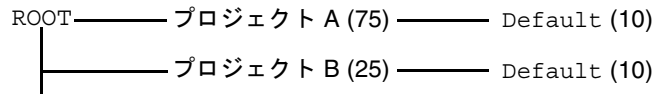
## ▼ プロジェクトに対する基本割当スケジューリングを作成する - 全ユーザーに同じ配分をする場合

この構成の目的は、Sun Grid Engine, Enterprise Edition 5.3 クラスタにまとめられている全資源のうちの一定の割合が一定の期間の間にさまざまなプロジェクトに確保されるようにすることにあります。同じプロジェクト内でジョブの資源競合が発生した場合は、それぞれに等しい配分が行われようにします。

1. スケジューラ構成 (sched\_conf (5)) で基本割当チケット数 (たとえば 1000000) を指定します。
2. スケジューリング対象のユーザーそれぞれに 1 ユーザー分 (user (5)) 追加します。
3. スケジューリング対象のプロジェクトのそれぞれに 1 プロジェクト分 (project (5)) 追加します。
4. qmon グラフィカルユーザーインターフェースを使用して、すべてのスケジューリング対象プロジェクト (ノード) の構成を反映した基本割当ツリーを作成します。

5. 各プロジェクトに基本割当配分を割り当てます。

以下は、基本的な構成例です。



6. 基本割当ツリーの各プロジェクトの下に枝としてユーザー default を追加します。

## ▼ プロジェクトに対する基本割当スケジューリングを作成する - プロジェクト別にユーザーごとの配分をする場合

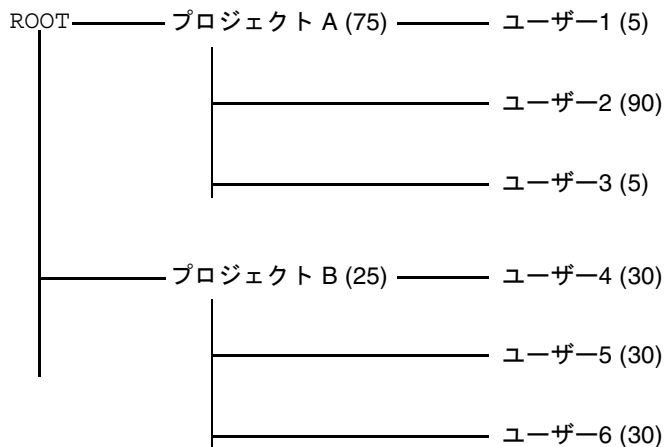
この構成の目的は、Sun Grid Engine, Enterprise Edition 5.3 クラスタにまとめられている全資源のうちの一定の割合が一定の期間の間にさまざまなプロジェクトに確保されるようにすることにあります。ユーザーごとに配分を割り当てる必要があります。

1. スケジューラ構成 (sched\_conf (5)) で基本割当チケット数 (たとえば 1000000) を指定します。
2. スケジューリング対象のユーザーそれぞれに 1 ユーザー分 (user (5)) 追加します。
3. スケジューリング対象のプロジェクトのそれぞれに 1 プロジェクト分 (project (5)) 追加します。
4. qmon グラフィカルユーザーインターフェースを使用して、すべてのスケジューリング対象プロジェクト (ノード) の構成を反映した基本割当ツリーを作成します。

## 5. 各プロジェクトに基本割当配分を割り当てます。

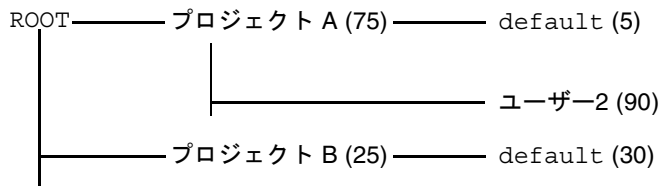
すべてのユーザーについて、そのユーザーがアクセスするプロジェクトに葉としてユーザーを追加し、配分を割り当てます。

以下は、簡単な基本割当ツリーの構成例です。



## 説明

個別に配分を割り当てる必要があるユーザーが 2, 3 人しかない場合は、プロジェクトノードの下に個々のユーザーの他にユーザー default を使用することができます。たとえば、そうすることによって基本割当ツリーを以下のように簡略化することができます。



---

## よくある問題の解決

ここでは、よくある問題を診断して解決する方法をまとめています。

- 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 `sttdy` コマンドが含まれていることが考えられます。 `sttdy` 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づけられません。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          # 端末のチェック
if ($status == 0) # 端末が存在する場合に成功
<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、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. DNS を、クラスタのホスト名解決のために使用している場合は、主ホスト名として完全修飾ドメイン名 (FQDN) を返すように、`/etc/hosts` と NIS を設定します (168.0.0.1 myhost.dom.com myhost のような短いエイリアス名を付けてもかまいません)。

DNS を使用していない場合は、たとえば次のように、すべての `/etc/hosts` ファイルと NIS テーブルが一致していることを確認します。

```
168.0.0.1 myhost.crop myhost または 168.0.0.1 myhost
```

- **問題** - 次のような警告が 30 秒おきに `<セル>/spool/<ホスト>/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、`<セル>/common/local_conf/` には、各ホスト用のファイルがあり、それぞれに FQDN 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、FQDN 付きの `meta` が返される。
- **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 execheost
```

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、`sgc_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 (`qrsh(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/<セル>/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 <ホスト名> 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ー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。(スーパーユーザーが rlogin と rsh が 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、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。一般的な `qmake` 呼び出しは次のとおりです。

```
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 ...`)。

---

## 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機能がいくつか用意されています。以下では、それらの使用方法を説明します。

### 保留中のジョブがディスパッチされない

保留中のジョブがあり、実行可能であることは明らかなのですが、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、この原因を探るためのユーティリティとして一組のユーティリティとオプション、`qstat -j <ジョブ ID>` と `qalter -w v <ジョブ ID>` が用意されています。

- `qstat -j <ジョブ ID>`

`qstat -j <ジョブ ID>` を有効にすると、特定のジョブが前回のスケジューリング実行でディスパッチされなかった理由一覧を得ることができます。この監視機能が有効または無効にできるようになっているのは、`schedd` デーモンと

qmaster 間の通信で不要なオーバーヘッドを生む可能性があるためです (sched\_conf(5) の schedd.job\_info の下を参照)。以下は、id 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、クラスタの現在の使用状況が考慮されます。ただし、正確にはこれは目的の情報ではないことがあります。たとえばすべてのキューロットが他のユーザーのジョブによってすでに占有されている場合、目的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、ジョブがディスパッチ不可能な一般的な理由を一覧表示します。この目的のために、このコマンドはドライスケジューリング実行を実施します。このドライスケジューリング実行で特別なのは、消費可能なすべての資源 (スロットを含む) がこのジョブに完全に利用可能であるとみなされることです。同様に、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

qstate の出力では、ジョブまたはキューのエラーが大文字の E で示されます。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できなかった場合、ジョブはエラー状態になります。また、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できなかった場合、キューはエラー状態になります。

Sun Grid Engine, Enterprise Edition 5.3 システムには、ジョブの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ことを可能にする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状態はジョブの実行に失敗したことが原因であるため、診断情報は、両方の種類のエラー状態に当てはまります。

#### ■ ユーザー宛て中止メール

ジョブが submit オプションの `-m a` 付きで実行依頼された場合は、`-M <ユーザー名>[@ホスト]` 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おり、情報源として利用することを推奨します。

- qacct アカウンティング

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)。管理者宛てメールには、ユーザー宛て中止メールより詳細な診断情報が含まれており、ジョブの実行エラーがよく発生する場合の情報源として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。デフォルトのインストールでは、`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。