



Sun™ Grid Engine 5.3 (企业版) 管理基础

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

部件号: 816-7483-10
2002年7月, 修订版 A

请将关于此文档的意见或建议发送至: docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版权所有。

Sun Microsystems, Inc. 对此文档描述的产品中所包含的相关技术拥有知识产权。在特殊且不受限制的情况下，这些知识产权可能包括 <http://www.sun.com/patents> 上列出的一个或多个美国专利，以及美国和其它国家的一个或多个其它专利或待决的专利申请。

此文档及其所属产品按照限制其使用、复制、分发和反编译的许可证进行分发。未经 Sun 及其许可证颁发机构的事先书面授权，不得以任何方式、任何形式复制本产品或本文档的任何部分。

第三方软件，包括字体技术，由 Sun 供应商提供许可和版权。

本产品的某些部分从 Berkeley BSD 系统派生而来，经 University of California 许可授权。UNIX 是在美国和其它国家注册的商标，经 X/Open Company, Ltd. 独家许可授权。

Sun、Sun Microsystems、Sun 徽标、AnswerBook2、docs.sun.com 和 Solaris 是 Sun Microsystems, Inc. 在美国和其它国家的商标或注册商标。

所有的 SPARC 商标均按许可证使用，是 SPARC International, Inc. 在美国和其它国家的商标或注册商标。带有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

OPEN LOOK 和 Sun™ 图形用户界面是由 Sun Microsystems, Inc. 为其用户和许可证持有人开发的。Sun 承认 Xerox 在为计算机行业研究和开发可视或图形用户界面方面所作出的先行努力。Sun 以非独占方式从 Xerox 获得 Xerox 图形用户界面的许可证，该许可证涵盖实施 OPEN LOOK GUI 且遵守 Sun 的书面许可协议的许可证持有人。

本资料按“现有形式”提供，不承担明确或隐含的条件、陈述和保证，包括对特定目的的商业活动和适用性或非侵害性的任何隐含保证，除非这种不承担责任的声明是不合法的。



请回收



目录

关于网络计算和 Sun Grid Engine 5.3（企业版）软件	2
分布式资源管理的职能	3
策略、主机和守护程序	4
普通管理任务的命令行指导	5
▼ 如何从主机添加或删除管理权限	5
▼ 如何添加一台执行主机	5
▼ 如何删除一台执行主机	6
▼ 如何添加或删除一台提交主机	6
▼ 如何显示当前主机的名称	6
▼ 如何管理队列	7
▼ 如何更改主控主机	8
▼ 如何设置影像主控主机	8
使用脚本和文件管理任务	9
▼ 如何用文件添加或修改对象	9
▼ 如何用文件修改队列、主机和环境	10
▼ 如何用文件修改全局配置或调度程序	14
▼ 如何微调网格环境	15
策略	17

职能策略设置 17

- ▼ 如何创建基于用户的职能调度 17
- ▼ 如何创建基于项目的职能调度 18
- ▼ 如何创建基于部门的职能调度 18
- ▼ 如何用 FCFS 在每个项目内创建基于项目的份额树调度 19
- ▼ 如何创建基于项目的份额树调度并且使每位用户份额均等 20
- ▼ 如何以每位用户单独的份额在每个项目内创建基于项目的份额树调度 20

常见问题诊断 21

诊断问题 26

暂挂作业未分配 27

报告作业或队列处于错误状态 E 27

Sun Grid Engine 5.3（企业版）管理基础

本文档为快速参考手册，为 Sun™ Grid Engine 5.3（企业版）系统的管理员提供帮助。除了提供对普通管理任务的指导外，本文档还包含了对常规网格计算和专门针对 Sun Grid Engine 5.3（企业版）产品的网格计算的简短讨论。本文档得出了几个微调 and 错误诊断的技巧。

本手册可用作《Sun Grid Engine 5.3（企业版）管理和用户指南》的补充资料，但本手册无法替代后者，因为在后者中您可以找到 Sun Grid Engine 5.3（企业版）概念和过程的详细解释。

本文档中包含完成以下任务的指导。

- 第 5 页的“如何从主机添加或删除管理权限”
- 第 5 页的“如何添加一台执行主机”
- 第 6 页的“如何删除一台执行主机”
- 第 6 页的“如何添加或删除一台提交主机”
- 第 6 页的“如何显示当前主机的名称”
- 第 7 页的“如何管理队列”
- 第 8 页的“如何更改主控主机”
- 第 8 页的“如何设置影像主控主机”
- 第 9 页的“如何用文件添加或修改对象”
- 第 10 页的“如何用文件修改队列、主机和环境”
- 第 14 页的“如何用文件修改全局配置或调度程序”
- 第 15 页的“如何微调网格环境”
- 第 17 页的“如何创建基于用户的职能调度”
- 第 18 页的“如何创建基于项目的职能调度”
- 第 18 页的“如何创建基于部门的职能调度”
- 第 19 页的“如何用 FCFS 在每个项目内创建基于项目的份额树调度”
- 第 20 页的“如何创建基于项目的份额树调度并且使每位用户份额均等”

- 第 20 页的 “如何以每位用户单独的份额在每个项目内创建基于项目的份额树调度”
- 第 21 页的 “常见问题诊断”
- 第 26 页的 “诊断问题”

注意 – 本文档中的大部分最初出现在 Sun Grid Engine 项目网站的 “How-To” 部分。由于经常更新，此网站对 Sun Grid Engine 5.3（企业版）系统管理员有着特殊意义，并且非常值得时常访问。此网站的 URL 为：
<http://gridengine.sunsource.net/project/gridengine/howto/howto.html>

关于网格计算和 Sun Grid Engine 5.3 （企业版）软件

若您对网格计算管理不甚了解，那么您处于绝大多数专业计算机管理员之列。迄今为止，只有极少数人能称为真正有经验的网格计算管理员，因为启用软件技术的历史还不足十年。根据 Ian Foster 博士（网格计算领域的先驱之一）的说法，有关网格计算的概念从 1995 年才开始研究。因此，即使此技术最早的采纳者也只有不足十年的经验。

但是，尽管您没有网格计算的丰富经验，但您有使用网格——虽然是另一种网格——的广泛经验。每次将电子开关置为“开”，您就使用了网格资源。在大多数工业化国家，电通过“电力网格”提供给家庭和企业，该网格由许多彼此独立的电力资源生产者提供，他们将其产品输入公用的资源池，这就是网格。

资源汇聚对消费者和生产者都有益处。电力生产者获得的益处是，可以按其最大能力运行发电厂，即使其紧邻的地理区域上的消费者不需要如此多的电力。例如，远离城市的农村消费者可能不需要一般发电厂能生产的全部电力。然而，电力网格可以让发电厂将“多余电量”出售给网格，由此为远处那些本地发电厂不能满足当地需求的消费者服务。

这种无论何处有需求，都可以将未充分利用资源转化为产品，并交付那些资源以满足需求的能力，即电力网络的定义特性。通过采用“网格”这个名称，网格计算概念的先驱们创造了相应的类比。网格计算的基本目标为：汇聚数百（甚至数千）闲置或近似闲置的计算机的未充分利用资源，并将组合后的极强的计算能力用于满足需要大量计算的消费者。

分布式资源管理的职能

此类比在交付细节上失效。电力网格管理通过把所有网格的生产者成员与电力消费者最终的家庭、工厂、农场和办公室切实连上线路来汇聚资源。一旦连入并输入电力网格，电力将根据需要流动。一开灯，电力就从某处的发电厂经过电力网格，流入灯泡。网格计算环境中的计算资源也可称为“连”在一起，虽然使用的是网络技术：**Internet**（主要示例）和**Intranet**。但是并不是简单打开网络计算机的开关就期望能使用（或提供）计算网格的资源。

只有在安装专门的**中间件**软件后才能访问计算网格。中间件为位于计算机操作系统（如 Sun 的 Solaris™ 操作环境或 Linux 操作系统）和应用程序软件（如三维图形绘制程序）之间的软件层。Sun Grid Engine 软件即为启用参与计算网格的中间件。

作为早期“开放源”网格计算项目的切实结果，**Sun Grid Engine** 是**分布式资源管理**软件，它在多个用户和多个计算资源（通常为远程）之间调节，为用户提供复杂作业所需的巨大计算资源。通过**Sun Grid Engine** 软件，闲置的计算资源将加入网格，变得可以为远程用户（常常是）所用，由此提高用户和资源的生产力。例如，通常只以其实际能力的 1/4 运行的计算资源通过加入网格，已经表明接近于 100% 运行。

可计算设备是“分布式资源管理”术语中的“分布式资源”。**Sun Grid Engine** 软件则是术语中不可缺少的“管理”部分。没有这样的管理，就会产生混乱。作为网格管理员，**Sun Grid Engine** 软件接受用户提交的作业，并且基于**资源管理策略**（这些策略由该组织的管理和技术人员设定）调度它们在网格中的相应系统上执行。

任何两个网格都不可能如出一辙；一种尺寸大小不会适合所有的情况。存在三种主要的网格级别，其规模从单个系统一直到使用数千处理器的超型计算机级计算区域。

- **群集网络**（由基本 Sun Grid Engine 5.3 软件启用）是最简单的一种，它包含了协同工作的多台计算机**主机**，为单一项目或部门的用户提供单点访问。

Sun Grid Engine 5.3（企业版）软件扩展了此基本模型，创建了另外两个更复杂、更强大的网格级别。

- **公司网络**使得一个组织内的多个项目或部门能够共享计算资源。各个组织可以运用公司网络处理许多种类的任务，从循环事务处理到绘图、数据采集等等。
- **全球网络**是一组公司网络的集合，它跨越组织界线形成一个非常庞大的虚拟系统。用户可获得的计算能力远远超出了其组织内部的可用资源。

Sun Grid Engine 5.3（企业版）软件为公司网络提供了所需的能力和灵活性。此产品对由基本 Sun Grid Engine 软件启用的现有群集网络极其有用。这是因为它能将公司内所有现存的 Sun Grid Engine 群集网络进行联合后创建出公司网络，并且保证了整个切换过程的顺利完成。此产品还是企业级公司首次向网格运算模式迈进的良好起点。

策略、主机和守护程序

Sun Grid Engine 5.3（企业版）软件根据系统管理员维护的企业资源策略协调可计算资源的交付。Sun Grid Engine 5.3（企业版）系统运用这些策略检查并收集公司网络内可用的计算资源，然后自动地按整个公司网络的最佳用法分配和交付这些资源。

为使公司网络内达成合作关系，使用网络的项目拥有者们需要洽谈策略，能灵活地运用策略越权手动控制独特的项目请求，并能使策略得以自动监控和执行。

作为管理员，您可以定义高级的使用策略，按照任何适合站点的方式进行自定义。有 4 个这样的策略可用，它们在《*Sun Grid Engine 5.3（企业版）管理和用户指南*》中进行了详细的说明。

- 职能
- 基于共享
- 限期
- 越权

Sun Grid Engine 5.3（企业版）策略管理自动控制群集中共享资源的使用以达到您的目标。在与其它低优先级的作业争夺资源时，高优先级作业将首先分配资源并获得更高的 CPU 配额。Sun Grid Engine 5.3（企业版）软件监控所有作业的进度，并根据进度和策略中定义的相关目标相应地调节作业的优先级别。

无论站点的用法规则松散还是严格，Sun Grid Engine 5.3（企业版）策略模块都可以适应。这种基于策略的资源分配可在累积时间内（如一周、一月或一季度）为每位用户、每个团队、每个部门和所有项目分配共享的系统资源。

对 Sun Grid Engine 5.3（企业版）系统而言，最基本的主机类型有四种。

- 主控
- 执行
- 管理
- 提交

此外，管理员最好创建一个**影像主控主机**。虽然只有一个主控主机，群集中的其它机器也可以被指定为影像主控主机，从而提供更大的可用性。影像主控主机可持续监控主控主机，并自动、明显地在主控主机失败时进行控制。已经在群集中的作业不会受到主控主机失败的影响。

每一主机类型均在《*Sun Grid Engine 5.3（企业版）管理和用户指南*》中进行了详细的说明。

4 个守护程序提供了 Sun Grid Engine 5.3（企业版）功能。

- `sge_qmaster` 为主控守护程序。
- `sge_schedd` 为调度程序守护程序。
- `sge_execd` 为执行守护程序。

- sge_commd 为通信守护程序。

这些守护程序中的每一个均在 《Sun Grid Engine 5.3 (企业版) 管理和用户指南》中进行了详细的论述。

本文档其它部分提供了如何完成普通 Sun Grid Engine 5.3 (企业版) 管理任务的指导。其它的背景信息和有关如何完成此处没有提及的任务的指导, 请参见 《Sun Grid Engine 5.3 (企业版) 管理和用户指南》、《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》或手册页。

普通管理任务的命令行指导

注意 – 下列指导仅基于在命令行使用命令。所有命令都在 Sun Grid Engine 5.3 (企业版) 图形用户界面 QMON 中有相应的功能。有关如何使用 QMON 完成以下任务的说明, 请参见 《Sun Grid Engine 5.3 (企业版) 管理和用户指南》。

▼ 如何从主机添加或删除管理权限

- 要添加管理权限, 输入以下命令。

```
qconf -ah
```

- 要删除管理权限, 输入以下命令。

```
qconf -dh
```

▼ 如何添加一台执行主机

1. 输入以下命令, 将新主机建为管理主机。

```
qconf -ah
```

2. 作为新主机上的 root 用户，从 `$SGE_ROOT` 运行以下脚本。

```
install_execd
```

▼ 如何删除一台执行主机

- 输入以下命令，删除主机相关的队列。

```
qconf -dq
```

- 输入以下命令删除主机。

```
qconf -de
```

▼ 如何添加或删除一台提交主机

- 要将一台主机指定为提交主机，在主机的命令行输入以下命令。

```
qconf -as
```

- 要删除一台主机的提交主机职能，在主机的命令行输入以下命令。

```
qconf -ds
```

▼ 如何显示当前主机的名称

- 要显示管理主机名，输入以下命令。

```
qconf -sh
```

- 要显示提交主机名，输入以下命令。

```
qconf -ss
```

- 要显示执行主机名，输入以下命令。

```
qconf -sel
```

▼ 如何管理队列

- 要添加队列，输入以下命令。

```
qconf -aq
```

- 要从文件添加队列，输入以下命令。

```
qconf -Aq
```

- 要删除队列，输入以下命令。

```
qconf -dq
```

- 要修改队列，输入以下命令。

```
qconf -mq
```

- 要更改多个队列的个别属性，输入以下命令。

```
qconf -mqattr
```

▼ 如何更改主控主机

1. 输入以下命令，停止当前主控主机上的主控守护程序和调度程序守护程序。

```
qconf -ks -km
```

2. 根据以下指导，编辑 `$SGE_ROOT/default/common/act_qmaster` 文件。

- a. 在 `act_qmaster` 文件中，用新的主控主机名替代当前主机名。

此名称应与 `gethostname` 实用程序返回的名称相同。要获得此名称，在新的主控主机上输入以下命令。

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

- b. 用 `gethostname` 实用程序返回的名称替代 `act_qmaster` 文件中的旧名称。

3. 在新的主控主机上，执行以下脚本。

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

这将在新的主控主机上启动 `sge_qmaster` 和 `sge_schedd`。

▼ 如何设置影像主控主机

1. 根据以下指导，创建 `shadow_masters` 文件。

- 在 `$SGE_ROOT/default/common` 中创建该文件。
- 将主要主控主机放在第一行，再按您想要的顺序列出被选为担任主控责任的其它主机。例如：

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

以上示例中，`host1` 为主要主控主机。若 `host1` 发生故障，`host2` 将在大约 10 分钟以后担任主控主机。另外，若 `host2` 也发生故障，`host3` 将担任主控主机。

2. 验证正确的权限。

所有主控影像主机必须具有对 `qmaster` 假脱机目录的读 / 写权限。

3. 用 `sge5` 启动脚本启动影像守护程序。在每台主机上以 `root` 用户身份输入以下命令。

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

完成这些步骤后，就会激活 Sun Grid Engine 5.3（企业版）群集的主控影像。

使用脚本和文件管理任务

注意 – 可以使用 QMON 图形用户界面执行所有的 Sun Grid Engine 5.3（企业版）管理任务，并且使用它可以给您带来另一好处，即了解系统的所有能力。不过，您也可以通过在 shell 提示符下发布命令以及从 shell 脚本内调用命令来管理 Sun Grid Engine 5.3（企业版）网格。许多有经验的管理员都发现这是一条更灵活、更快捷和更强大的更改设定的捷径。以下三组指导阐述了这些方法。

▼ 如何用文件添加或修改对象

- 根据文件中创建的说明，使用 `qconf` 命令添加或修改对象。

此命令有以下语法：`qconf -{A,M}<对象><文件名>`

在以上语法中，`-A` 表示添加，`-M` 表示修改。`<对象>` 变量可为以下任何一个：

- `c` – 属性组
- `ckpt` – 检查点环境
- `e` – 执行主机
- `p` – 并行环境
- `q` – 队列
- `u` – 用户组

可以将此选项与 `qconf` 命令的 `show` 选项（即 `qconf -s<对象>`）结合使用，获得现有对象，修改它，再更新现有对象或创建新对象。

示例

以下为 shell 脚本的示例，它修改了现有检查点环境的迁移命令。

```
#!/bin/sh
# ckptmod.sh: modify the migration command
# of a checkpointing environment
# Usage: ckptmod.sh <checkpoint-env-name> <full-path-to-
command>
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
```

▼ 如何用文件修改队列、主机和环境

可以按以下两种不同方法之一来使用 `qconf` 命令并结合其它命令，根据此命令的自变量，可从命令行修改各个队列、主机、并行环境和检查点环境。以下指导涵盖了两种方法。

- 若已经准备了文件，输入 `qconf` 命令和相应选项。

此命令有以下语法；尤其要注意选项字母的大小写，因为大写字母和小写字母的差别很大。

```
qconf -M{q,e,p,ckpt} <文件名>
```

选项有以下含义。

- -M – 从现有文件（<文件名>）修改
- q – 队列
- e – 执行主机
- p – 并行环境
- ckpt – 检查点环境

- 若没有准备文件，输入 `qconf` 命令和相应选项。

此命令有以下语法；尤其要注意选项字母的大小写，因为大写字母和小写字母的差别很大。

```
qconf -m{q,e,p,ckpt} <对象名>
```

选项有以下含义。

- -m – 通过打开文本编辑器创建修改文件来修改
- q – 队列
- e – 执行主机
- p – 并行环境
- ckpt – 检查点环境

讨论

两个命令的差别 — 第一个命令中为大写字母 M，第二个命令中为小写字母 m — 控制着命令的结果。-M 和 -m 选项都表示 *修改*，但大写字母 M 表示从 *现有文件* 修改，而小写字母 m 则不。小写字母 -m 表示：在编辑器中打开一个临时文件，保存对此文件的更改，然后退出编辑器，系统能马上反映那些更改。

不过，若想一次更改多个对象或者想非交互地更改对象配置，请使用 `qconf -...attr` 命令组。

此命令组根据 *文件* 中的说明进行更改：

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

对应的命令组根据 *命令行* 中的说明进行更改：

```
qconf -{a,m,r,d}attr queue|exechost|pe|ckpt <属性> <值> <队列列表> | <主机列表>
```

两组命令中，选项含义如下：

- -A/a – 添加属性
- -M/m – 修改属性
- -R/r – 替换属性
- -D/d – 删除属性
- <属性> – 要更改的队列或主机属性
- <值> – 影响的属性值
- <文件名> – 包含属性值偶的文件

a、m 和 d 选项可对值列表中的单个值进行操作，而 r 将用命令行或文件中指定的新值代替整个值列表。

示例

- 将 tcf27-e019.q 的队列类型改为仅适用于批处理:

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

- 在 new.cfg 文件内容的基础上, 修改 tcf27-e019.q 的队列类型和 shell 启动操作:

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

- 将名称为 storage 和 license 的属性组附加到主机 tcf27-e019:

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

- 添加值为 1000M、名称为 scratch1 的资源, 以及值为 2 的 long 资源:

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

- 将名称为 short、值为 4 的资源附加到主机

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

- 将 scratch1 的值更改为 500M, 而保持其它值不变:

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

- 删除资源 long:

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


- 为点检查环境 sph 添加 tcf27-b011.q 到队列列表:

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

- 将并行环境 make 中位置数的数目更改为 50:

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

用 qselect 命令选定队列

qselect 命令可输出队列列表。若带选项调用, 它仅列出与给定说明匹配的队列。您可以将它与 qconf -...attr 命令集组合以充分使用此命令, 选定您想修改的特定队列。

示例

- 列出 Linux 机器上的所有队列:

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

- 列出带有双 CPU 的机器上的所有队列:

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

- 列出所有 4 CPU、64 位的 Solaris 机器上的所有队列:

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

- 列出提供应用程序许可证的队列 (以前已配置):

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

结合 qselect 与 qconf, 可以用一个命令行作大范围的更改。要做到这一点, 将整个 qselect 命令包含在反斜线以内 (/), 并用它代替 qconf 命令行中的 <队列列表> 变量。

示例

- 将 Solaris 机器上的所有队列的 `prolog` 脚本设置为 `sol_prolog.sh`:

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

- 将双处理器系统上所有队列的 `fluent_license` 属性设置为 2:

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

将 `qconf` 命令与 `qselect` 命令一起使用提供了最灵活的方法，可使 Sun Grid Engine 5.3（企业版）队列的配置自动化，并允许建立您自己的自定义管理脚本。

▼ 如何用文件修改全局配置或调度程序

- 要更改全局配置，请按照下节的指导使用 `qconf -mconf` 命令。
- 要更改调度程序，请按照下节的指导使用 `qconf -msconf` 命令。

讨论

这两个命令将在编辑器中打开一个临时文件。退出编辑器时，系统将处理保存到临时文件的所有更改，而且更改立即生效。打开临时文件的编辑器程序由 `EDITOR` 环境变量指定。若没有定义此变量，则将使用 `vi`。

可以充分利用 `EDITOR` 环境变量使 `qconf -m...` 命令的执行自动化。更改此变量的值，使其指向修改文件（文件名由第一个自变量给定）的程序。程序修改临时文件并退出后，系统将读取所作的修改并立即更新。

注意 – 若文件的修改时间未在编辑操作后更改，系统可能错误地假定文件未修改。因此，写入文件之前，应插入 `sleep 1` 指令以确保创建不同的修改时间。

示例

以下示例中的脚本修改了调度程序的调度时间间隔。

```
#!/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
```

以上的脚本修改了 EDITOR 环境以指向其本身，然后调用 `qconf -msconf` 命令。脚本的第二个嵌套调用修改了第一个自变量指定的临时文件，然后退出。然后，Sun Grid Engine 5.3（企业版）系统自动读入更改，第一次调用结束。以上技巧可与 `qconf -m...` 命令结合使用。但是，该技巧对调度程序以及全局配置的管理显得更加有用，因为没有其它方法可使这些操作自动化。

▼ 如何微调网格环境

Sun Grid Engine 5.3（企业版）为一个全功能、多用途的分布式资源管理 (DRM) 工具。系统中的调度程序组件支持大范围的不同计算区域方案。要从计算环境中获取最大性能，回顾一下启用了哪些功能以及确实需要哪些功能来解决负荷管理问题是非常有价值的。禁用某些功能可以提高群集吞吐量。

调度程序监控

调度程序监控可以帮助查找某些作业没有分配的原因。但是，为所有作业随时提供此信息可能消耗资源，通常不需要这样做。

- 要禁用调度程序监控，在调度程序配置 `sched_conf(5)` 中将 `schedd_job_info` 设置为 `false`。

已完成的作业

若为阵列作业，`qmaster` 的已完成作业列表可能非常大。将其关闭既可节省内存，又可加快 `qstat` 进程，`qstat` 进程也取用已完成的作业列表。

- 要关闭已完成的作业列表功能，可在全局配置 `sge_conf(5)` 中将 `finished_jobs` 设置为 `0`。

作业验证

作业提交时强制验证是很有价值的过程，可防止未分配的作业一直保留为暂挂状态。不过，验证作业也很消耗时间，当不同环境中带有不同的执行节点和可用资源，每位用户都有其自己的作业配置文件时尤其费时。在仅有几个不同作业的环境中，可以省略常规的作业验证。

- 要禁用作业验证，在群集范围的默认请求中添加 `qsub(1)` 选项 `-wn`（参见 `sge_request(5)`）。

加载阈值和暂停阈值

若有意多预订了机器并且需要一种机制防止系统负荷过大，则需要加载阈值。暂停阈值同样适用于这种情况。另外一种需要加载阈值的情况是，执行节点仍然对交互式负荷开放（交互式负荷不在 Sun Grid Engine 5.3（企业版）系统的控制之下），而您想防止节点负荷过大。

若计算区域具有多重目的——例如，每个计算节点的 CPU 仅由一个队列位置代表，且这些节点无交互式负荷——那么您可以省略 `load_thresholds`。

- 要禁用两个阈值，将 `load_thresholds` 设置为 `none` 并将 `suspend_thresholds` 设置为 `none`（参见 `queue_conf(5)`）。

负荷调整

负荷调整实际上是在作业分配后用于增加精确的负荷。此机制在多预订了机器以按负荷阈值校准时很有用。若非需要，不要打开负荷调整，因为它们会强加给调度程序一些附加的工作，这些工作排序主机和负荷阈值验证有关。

- 要禁用负荷调整，在调度程序配置 `sched_conf(5)` 中将 `job_load_adjustments` 设置为 `none` 并且将 `load_adjustment_decay_time` 设置为 `0`。

按需求调度

Sun Grid Engine 5.3（企业版）系统的缺省值为按固定时间间隔启动调度（参见 `schedd_conf(5)` 中的 `schedule_interval`）。固定时间间隔的优点是可限制 `qmaster/scheduler` 的 CPU 时间消耗。缺点是它抑制了调度程序，人为导致了吞吐量的限制。许多计算区域中有专门用于 `qmaster/scheduler` 的机器，在这样的设置中，调度程序不应受到抑制。

- 使用全局群集配置 `sge_conf(5)` 中 `schedd_params` 部分的 `FLUSH_SUBMIT_SEC` 和 `FLUSH_FINISH_SEC` 设定，配置按需求调度。

若激活了按需求调度，则计算区域的吞吐量仅受 `qmaster/scheduler` 所驻留机器能力的限制。

策略

Sun Grid Engine 5.3（企业版）软件最重要的功能之一是它通过策略（由系统管理员设置）的方式分配运算能力的概念。请参见第 4 页的“策略、主机和守护程序”一节以获得背景信息。

下节介绍如何为您的站点创建基本的“职能”策略设置和“基于份额”策略设置。对“限期”和“越权”策略设置，以及附加的职能和基于份额设置的介绍，均包含在《*Sun Grid Engine 5.3（企业版）管理和用户指南*》中。

职能策略设置

本节中的所有设置均使用职能策略作为主要策略。此类型的设置确保已定义的份额对每一用户、项目或部门都能随时保证。系统给闲置资源分派作业时，会优先考虑其所占资源少于预想资源的用户、项目或部门的作业。

同时保证了完全资源利用，因为未使用的份额部分在那些需要资源的用户、项目和部门之间进行分配。过去的资源消耗不考虑在内。

▼ 如何创建基于用户的职能调度

此设置的目的是将 Sun Grid Engine 5.3（企业版）群集中结合的所有资源分成特定的份额分配给不同用户。FCFS 调度用于在同一用户的作业中进行。

1. 在全局配置 (`sge_conf(5)`) 的 `schedd_params` 部分, 使用 `SHARE_FUNCTIONAL_SHARES=1`。
2. 在调度程序配置 (`sched_conf(5)`) 中指定职能票券数目 (例如, 1000000)。
3. 为每位与调度程序相关的用户添加一位用户 (`user(5)`)。
4. 将职能份额分配给每位用户。
将份额按照整体的百分比进行分配。例如:
 - 用户 A (10)
 - 用户 B (20)
 - 用户 C (20)
 - 用户 D (50)

▼ 如何创建基于项目的职能调度

此设置的目的是将 Sun Grid Engine 5.3 (企业版) 群集中结合的所有资源分成特定的份额分配给不同项目。FCFS 调度用于在同一项目的作业中进行。

1. 在全局配置 (`sge_conf(5)`) 的 `schedd_params` 部分, 使用 `SHARE_FUNCTIONAL_SHARES=1`。
2. 在调度程序配置 (`sched_conf(5)`) 中指定职能票券数目 (例如, 1000000)。
3. 为每个与调度程序相关的项目添加一个项目 (`project(5)`)。

您可通过 `acl` 和 `xacl` 功能的方式控制对较高优先级项目的访问。请参见 *《Sun Grid Engine 5.3 (企业版) 管理和用户指南》* 或 *《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》* 以获得有关这些功能的信息。

4. 将职能份额分配给每个项目。
将份额按照整体的百分比进行分配。例如:
 - 项目 A (55)
 - 项目 B (45)

▼ 如何创建基于部门的职能调度

此设置的目的是将 Sun Grid Engine 5.3 (企业版) 群集中结合的所有资源分成一定的份额分配给不同部门。FCFS 调度用于在同一部门的作业中进行。

1. 在全局配置 (`sge_conf(5)`) 的 `schedd_params` 部分, 使用 `SHARE_FUNCTIONAL_SHARES=1`。

2. 在调度程序配置 (`sched_conf(5)`) 中指定职能票券数目 (例如, 1000000)。
3. 添加每个与调度相关的部门。
4. 将职能份额分配给每个部门。

将份额按照整体的百分比进行分配。例如:

- DepartmentA (90)
- DepartmentB (5)
- DepartmentC (5)

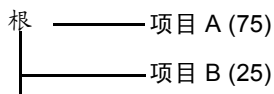
本节中的所有设置均使用份额树策略作为主要策略。此类型的设置确保已定义的份额在一段时间内保证份额树中配置的实例。当系统给闲置资源分派作业时,与那些过去所用资源少于预定份额树份额的份额树“分支”相关联的作业优先考虑。同时保证了完全利用资源,因为未使用的份额部分仍可用于与其他份额树分支相关联的暂挂作业。

▼ 如何用 FCFS 在每个项目内创建基于项目的份额树调度

此设置的目的是保证 Sun Grid Engine 5.3 (企业版) 群集中结合的所有资源在一段时间内以一定的份额分配给不同项目。FCFS 调度用于在同一项目的作业中进行。

注意 – 本节中的一些指导 (与本文档其它各节中的指导不同) 需要使用 `qmon` 图形用户界面。请参见 《Sun Grid Engine 5.3 (企业版) 管理和用户指南》以获得有关使用 `qmon` 的背景信息和介绍。

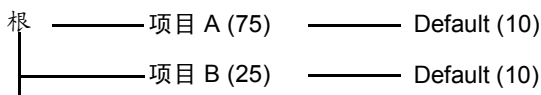
1. 在调度程序配置 (`sched_conf(5)`) 中指定份额树票券的数目 (例如, 1000000)。
2. 为每个与调度程序相关的项目添加一个项目 (`project(5)`)。
3. 使用 `qmon` 图形用户界面, 设置一个将所有与调度相关的项目的结构反映为节点的份额树。
4. 将份额树份额分配给项目。
简单的结构可能类似于下例。



▼ 如何创建基于项目的份额树调度并且使每位用户份额均等

此设置的目的是保证 Sun Grid Engine 5.3（企业版）群集中结合的所有资源在一段时间内有一定的份额分配给不同项目。在同一项目内竞争资源的所有作业份额目标均等。

1. 在调度程序配置 (`sched_conf(5)`) 中指定份额树票券的数目（例如，1000000）。
2. 为每位与调度相关的用户添加一位用户 (`user(5)`)。
3. 为每位与调度相关的项目添加一个项目 (`project(5)`)。
4. 使用 `qmon` 图形用户界面，设置一个将所有与调度相关的项目的结构反映为节点的份额树。
5. 将份额树份额分配给项目。
简单的结构可能类似于下例。



6. 在树中每一个项目下将用户 `default` 作为叶节点添加。

▼ 如何以每位用户单独的份额在每个项目内创建基于项目的份额树调度

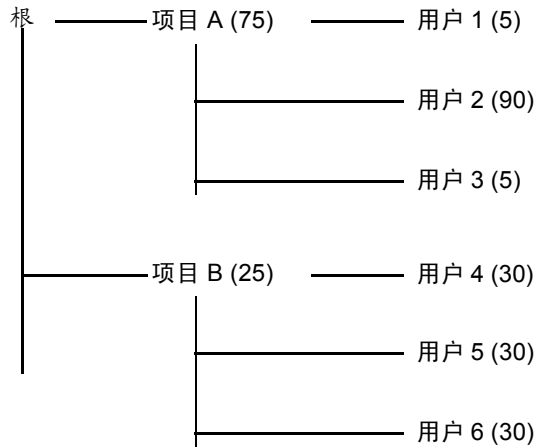
此设置的目的是保证 Sun Grid Engine 5.3（企业版）群集中结合的所有资源在一段时间内有一定的份额分配给不同项目。需要给每位用户分配单独的份额。

1. 在调度程序配置 (`sched_conf(5)`) 中指定份额树票券的数目（例如，1000000）。
2. 为每位与调度相关的用户添加一位用户 (`user(5)`)。
3. 为每位与调度相关的项目添加一个项目 (`project(5)`)。
4. 使用 `qmon` 图形用户界面，设置一个将所有与调度相关的项目的结构反映为节点的份额树。

5. 将份额树份额分配给项目。

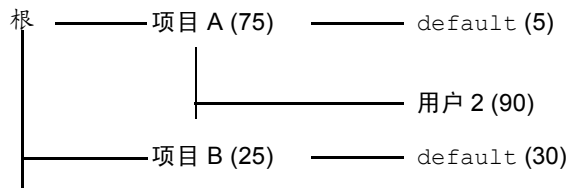
将每位用户在其有访问权限并分配单独份额的项目下作为叶节点添加。

简单的结构可能类似于下例。



讨论

若只有几位用户需要分配单独的份额，则您可在项目节点下将指定用户 default 与单独的用户结合使用。例如，上面所示的树可精简如下。



常见问题诊断

以下各节能帮助您诊断出常见问题的原因并做出正确响应。

- **问题** – 作业的输出文件显示: Warning: no access to tty; thus no job control in this shell....
 - **可能原因** – 一个或多个登录文件包含了 stty 命令。这些命令仅在存在终端时有用。

- **可能的解决方案** – 在 Sun Grid Engine 5.3（企业版）批处理作业中，没有与这些作业关联的终端。您必须从登录文件中删除所有 stty 命令，或者用一条在处理前检查终端的 if 语句把它们括起来。以下即为此例。

```
/bin/csh:
stty -g          # checks terminal status
if ($status == 0) # succeeds if a terminal is present
< 将所有 stty 命令放在这 >
endif
```

- **问题** – 作业标准错误日志文件显示：`tty`: Ambiguous。但是，用户的 shell 没有涉及在作业脚本中调用的 tty。
 - **可能原因** – shell_start_mode 默认为 posix_compliant；因此，所有作业脚本以在队列定义中指定的，而不是在作业脚本第一行中指定的，shell 运行。
 - **可能的解决方案** – 将 -s 标志用于 qsub 命令，或将 shell_start_mode 更改为 unix_behavior。
- **问题** – 您可以从命令行运行作业脚本，但通过 qsub 命令运行时失败。
 - **可能原因** – 可能对作业设置了进程限制。要测试这一点，写出执行 limit 和 limit -h 功能的测试脚本。分别在 shell 提示下和通过 qsub 命令交互执行两个脚本，比较结果。
 - **可能的解决方案** – 确保将配置文件中用于在 shell 中设置限制的所有命令删除。
- **问题** – 执行主机报告负荷为 99.99。
 - **可能原因** – 有三种可能。
 1. execd 守护程序未在主机上运行。
 2. 未正确指定缺省域。
 3. qmaster 主机所见的执行主机名与执行主机自身所见不同。
 - **可能的解决方案** – 根据不同原因，以下解决方案之一可行。（以下解决方案编号与“可能原因”的编号相匹配。）
 1. 作为 root 用户，在执行主机上通过运行 \$SGE_ROOT/default/common/'rcsge' 脚本来启动 execd 守护程序。
 2. 作为 Sun Grid Engine（企业版）管理员，运行 qconf -mconf 命令并将 default_domain 变量更改为 none。
 3. 若您正使用 DNS 解析运算群集的主机名，则请配置 /etc/hosts 和 NIS 以返回完全合格的域名 (FQDN) 作为主要主机名。当然，您仍可以定义和使用快捷别名；例如：168.0.0.1 myhost.dom.com myhost

若您并未使用 DNS，则请确保所有 /etc/hosts 文件和 NIS 表格均一致；例如：168.0.0.1 myhost.corp myhost 或 168.0.0.1 myhost

- **问题** – 每隔 30 秒，类似于以下的警告出现在 <单元>/spool/<host>/messages 中：

```
Tue Jan 23 21:20:46 2001|execd|meta|W|local
未定义配置 meta - 正在使用全局配置
```

但是每台主机都有一个文件在 <单元>/common/local_conf/ 中，每一个都带有 FQDN。

- **可能原因** – 在你的机器上解析的主机名 meta 返回短名称，但在你的主控主机上，返回 meta 和 FQDN。
- **可能的解决方案** – 确保所有的 /etc/hosts 文件和 NIS 表在这方面均一致。此例中，主机 meta 的 /etc/hosts 文件中可能存在一错误的命令行，如下所示：

```
168.0.0.1 meta meta.your.domain
```

但是，这一行应该为：

```
168.0.0.1 meta.your.domain meta.
```

- **问题** – 有时在守护程序的 messages 文件中可以看见 CHECKSUM ERROR、WRITE ERROR 或 READ ERROR 消息。
 - **可能原因** – 只要这些消息不以 1 秒的时间间隔出现（它们通常每天出现 1 到 30 次），就不需要理会这个问题。
- **问题** – 作业在某个特定队列中完成，并且在 qmaster/messages 中返回以下消息：

```
Wed Mar 28 10:57:15 2001|qmaster|masterhost|I| 作业 490.1
在主机 execheost 上完成
```

但是接下来会在执行主机的 execheost/messages 文件上看到以下错误消息：

```
Wed Mar 28 10:57:15 2001|execd|execheost|E| 无法找到目录
"active_jobs/490.1" 以收尾作业 490.1
```

```
Wed Mar 28 10:57:15 2001|execd|execheost|E| 无法删除目录
"active_jobs/490.1": opendir(active_jobs/490.1) 失败:
输入 / 输出错误
```

- **可能原因** – 自动装入的 `$SGE_ROOT` 目录被取消装入，导致 `sgexecd` 守护程序失去其 `cwd`。
- **可能的解决方案** – 使用 `execd` 主机的本地假脱机目录。使用 `qmon` 或 `qconf` 命令设置参数 `execd_spool_dir`。
- **问题** – 用 `qrsh` 实用程序提交交互式作业时，收到以下错误消息：

```
% qrsh -l mem_free=1G error: 错误: 无适合队列
```

然而使用 `qsub` 实用程序可为批处理作业获得队列，并且可用 `qhost -l mem_free=1G` 和 `qstat -f -l mem_free=1G` 查询队列。

- **可能原因** – 消息：错误：无适合队列，由 `-w e submit` 选项产生，缺省情况下，此选项对交互式作业都处于激活状态，如 `qrsh`（在 `qrsh(1)` 中查找 `-w e`）。若根据当前群集配置，`qmaster` 不能确定作业是否可分配，此选项将导致 `submit` 命令失败。此种机制的目的是一旦作业请求不能满足，就提前拒绝作业请求。
- **可能的解决方案** – 在这种情况下，`mem_free` 被配置为可用资源，但您未指定每台主机上可用的内存量。此检查中有意没有考虑内存负荷值，因为它们各不相同，不能被视为群集配置的一部分。要克服这一点，可执行以下操作之一。

一般用覆盖 `qrsh` 的缺省设置 `-w e` 的方法省略检查，即以 `-w n` 提交。也可以将此选项放入 `$SGE_ROOT/<单元>/common/cod_request`。

若打算将 `mem_free` 作为可用资源管理，用 `qconf -me <主机名>` 命令在 `host_conf(5)` 的 `complex_values` 中指定主机的 `mem_free` 容量。

若不想将 `mem_free` 作为可用资源管理，用 `qconf -mc 主机` 命令在 `complex(5)` 的 `consumable` 栏中将其再次变为不可用资源。

- **问题** – `qrsh` 不分配到其所在的同一节点。以下消息来自于 `qsh shell`：

```
host2 [49]% qrsh -inherit host2 hostname
错误: 执行作业 1 的任务失败:

host2 [50]% qrsh -inherit host4 hostname
host4
```

- **可能原因** – `gid_range` 不足。它应该定义为一个范围，而不是单个的数字。Sun Grid Engine 5.3（企业版）系统给主机上的每个作业分配一个不同的 `gid`。

- **可能的解决方案** – 用 `qconf -mconf` 或 `qmon` 图形用户界面调整 `gid_range`。建议的范围如下。

<code>gid_range</code>	20000-20100
------------------------	-------------

- **问题** – 在并行作业内部使用时，`qrsh -inherit -v` 不工作。并收到以下消息。

无法连接至 "qlogin_starter"

- **可能原因** – 此问题伴随嵌套的 `qrsh` 调用出现，由 `-v` 开关选项引起。第一个 `qrsh -inherit` 调用将设置环境变量 `TASK_ID`（并行作业内与任务紧密结合的 ID）。第二个 `qrsh -inherit` 调用将使用此环境变量登记任务，因其尝试启动的任务 ID 与已在运行的第一项任务的 ID 相同，故将失败。
- **可能的解决方案** – 可以在调用 `qrsh -inherit` 前，不设置 `TASK_ID`，或选择不使用 `-v` 开关选项而使用 `-v`，并且仅输出真正需要的环境变量。
- **问题** – `qrsh` 似乎根本不工作。您收到类似于以下的消息。

```

host2$ qrsh -verbose hostname
未定义本地配置 host2 - 正在使用全局配置
正在等待调度交互式作业 ...
交互式作业 88 已成功调度。
正在建立至主机 exehost 的会话
/share/gridware/utilbin/solaris64/rsh ...
rcmd: socket: 权限已拒绝
/share/gridware/utilbin/solaris64/rsh 已退出，并返回退出代码 1
正在从 shepherd 读取退出代码 ...
错误: 等待用于客户机连接的套接字时出错: 系统调用已中断
错误: 读取远程命令的返回代码时出错
在 /share/gridware/utilbin/solaris64/rsh 异常退出后清除
host2$

```

- **可能原因** – `qrsh` 的权限设置不当。
- **可能的解决方案** – 检查以下文件的权限，它们位于 `$SGE_ROOT/utilbin/`。（注意 `rlogin` 和 `rsh` 需要 `setuid` 并且要为 `root` 所拥有。）

```
-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*
```

注意 – \$SGE_ROOT 目录也需要带有 setuid 选项的 NFS 装入。若从提交客户机装入，带有 nosuid 选项，那么 qrsh（和相关命令）将不工作。

- 问题 – 尝试启动分布式建立时，qmake 退出，显示以下错误消息。

```
qrsh_starter: 执行子进程失败: 无此文件或目录
```

- 可能原因 – Sun Grid Engine 5.3（企业版）系统将在执行主机上启动一个 qmake 实例。若 Sun Grid Engine 5.3（企业版）环境（特别是 PATH 变量）未在用户的 shell 资源文件（.profile/.cshrc）中设置，此 qmake 调用将失败。
- 可能的解决方案 – 用 -v 选项将 PATH 环境变量输出到 qmake 作业。典型的 qmake 调用如下。

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

- 问题 – 使用 qmake 工具时，收到以下错误消息。

```
正在等待调度交互式作业 ... 超时 (4 秒)
等待套接字 fd 5 时已过期

无法调度 "qrsh" 请求，请稍后再试。
```

- 可能原因 – ARCH 环境变量可能在调用 qmake 的 shell 中设置错误。
- 可能的解决方案 – 将 ARCH 变量正确地设置为与群集中的可用主机匹配的支持值，或者在提交时指定正确值，如 qmake -v ARCH=solaris64 ...

诊断问题

Sun Grid Engine 5.3（企业版）系统提供了几种报告方法，帮助您诊断问题。以下内容简要介绍了它们的用法。

暂挂作业未分配

有时，暂挂作业显然可以运行，但没有获得分配。要诊断此原因，Sun Grid Engine 5.3（企业版）提供了两个实用程序及其选项，即 `qstat -j <作业 ID>` 和 `qalter -w v <作业 ID>`。

■ `qstat -j <作业 ID>`

启用后，`qstat -j <作业 ID>` 将为用户提供原因列表，说明为什么某个作业在上次调度运行时没有分配。这种监控可以启用也可以禁用，因为它可能在 `schedd` 守护程序和 `qmaster` 之间产生不必要的通信系统开销（参见 `sched_conf(5)` 中的 `schedd_job_info`）。以下是 ID 为 242059 的作业输出示例。

```
% qstat -j 242059
调度信息: 队列 "fangorn.q" 由于暂时不可用而终止
           队列 "lolek.q" 由于暂时不可用而终止
           队列 "balrog.q" 由于暂时不可用而终止
           队列 "saruman.q" 由于满而终止
           无法在队列 "bilbur.q" 中运行，因为未包含在其必须队列列表 (-q) 中
           无法在队列 "dwain.q" 中运行，因为未包含在其必须队列列表 (-q) 中
           没有对主机 "ori" 的权限
```

此信息由 `schedd` 守护程序直接产生，并考虑了群集的当前利用状况。有时这并不正好是您想要的，例如，若所有队列位置已经被其它用户的作业占据，则不会产生您感兴趣的作业的详细消息。

■ `qalter -w v <作业 ID>`

此命令列出了原则上作业不能分配的原因。为此目的，将执行干调度运行。干调度运行的特殊性在于所有可用资源（包括位置）都视为对作业完全可用。类似地，由于所有负荷值各不相同，将忽略它们。

报告作业或队列处于错误状态 E

作业或队列错误在 `qstat` 输出中通过大写字母 E 表示。在 Sun Grid Engine 5.3（企业版）系统尝试执行队列中的作业但由于此作业特有的原因失败时，此作业进入错误状态。在 Sun Grid Engine 5.3（企业版）系统尝试执行队列中的作业但由于此队列特有的原因失败时，此队列进入错误状态。

Sun Grid Engine 5.3（企业版）系统提供了一旦在作业执行错误时，收集诊断信息的一组便利途径。因为队列和作业的错误状态都因作业执行失败引起，所以诊断途径对两种类型的错误状态都适用。

- 用户异常终止邮件

若作业带着 `submit` 选项 `-m a` 提交，则异常终止邮件将发送到 `-M 用户[@ 主机]` 选项指定的地址。异常终止邮件包含有关作业错误的诊断信息，是推荐的用户信息来源。

- `qacct` 统计

若无可用的异常终止邮件，用户可运行 `qacct -j` 命令从 Sun Grid Engine 5.3（企业版）系统的作业统计功能获得作业错误的信息。

- 管理员异常终止邮件

管理员可以通过指定合适的电子邮件地址，订阅有关作业执行问题的管理员邮件（参见 `sge_conf(5)` 中的 `administrator_mail`）。管理员邮件比用户异常终止邮件所含的诊断信息更详细，若频繁出现作业执行错误，建议使用此方法。

- 消息文件

若无可用的管理员邮件，应首先调查 `qmaster messages` 文件。您可以搜索相应的作业 ID，查找与某一作业有关的记录。缺省安装中，`qmaster messages` 文件为 `$SGE_ROOT/default/spool/qmaster/messages`。

有时可在启动作业的 `execd` 守护程序的消息中找到附加信息。用 `qacct -j < 作业 ID >` 命令找到启动作业的主机，然后在 `$SGE_ROOT/default/spool/< 主机 >/messages` 中搜索作业 ID。