



Sun™ ONE Grid Engine 管理和用户指南

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

部件号: 816-7465-10
2002年9月, 修订版 A

请将有关本文档的意见或建议发送至: docfeedback@sun.com

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

本产品或文档按照限制其使用、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、注册商标或服务标记。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 商标的产品均以 Sun Microsystems, Inc. 开发的体系结构为基础。“能源之星”徽标是 EPA 的注册商标。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



Adobe PostScript

目录

| | |
|---------------|-------|
| 前言 | xv |
| 本书结构 | xv |
| 使用 UNIX 命令 | xvi |
| 排印约定 | xvi |
| Shell 提示符 | xvii |
| 相关文档资料 | xvii |
| 访问 Sun 联机文档资料 | xvii |
| Sun 欢迎您提出宝贵意见 | xviii |

第一部分 . 背景和定义

| | |
|----------------------------|---|
| 1. Sun Grid Engine 5.3 简介 | 1 |
| 什么是网格计算? | 1 |
| 通过管理资源和策略管理工作负荷 | 2 |
| 系统是如何运作的 | 2 |
| 使资源与请求相匹配 | 2 |
| 作业和队列: Sun Grid Engine 的世界 | 3 |
| Sun Grid Engine 5.3 组件 | 3 |
| 主机 | 4 |

| | |
|-----------------------------|---|
| 主控主机 | 4 |
| 执行主机 | 4 |
| 管理主机 | 4 |
| 提交主机 | 4 |
| 守护程序 | 5 |
| sge_qmaster – 主控守护程序 | 5 |
| sge_schedd – 调度守护程序 | 5 |
| sge_execd – 执行守护程序 | 5 |
| sge_commd – 通讯守护程序 | 5 |
| 队列 | 5 |
| 客户端命令 | 6 |
| Sun Grid Engine 图形用户界面 QMON | 8 |
| 自定义 QMON | 8 |
| Sun Grid Engine 术语词汇表 | 9 |

第二部分 . 入门

2. 安装 13

| | |
|----------------------------------|----|
| 快速启动安装概述 | 13 |
| 快速启动安装的先决条件 | 14 |
| 安装帐户 | 14 |
| 安装目录 | 14 |
| ▼ 如何创建安装目录 | 14 |
| 通讯端口号 | 15 |
| 读取发行媒体 | 15 |
| ▼ 如何解压缩 Sun Grid Engine 5.3 发行文件 | 15 |
| 为群集安装缺省的 Sun Grid Engine 系统 | 16 |
| ▼ 如何安装主控主机 | 16 |

| | |
|----------------|----|
| ▼ 如何安装执行主机 | 17 |
| 缺省系统配置 | 18 |
| 完全安装概述 | 19 |
| 阶段 1 – 规划 | 20 |
| 阶段 2 – 安装软件 | 20 |
| 阶段 3 – 验证安装 | 20 |
| 规划安装 | 21 |
| 先决任务 | 21 |
| 安装目录 <sgc 根目录> | 21 |
| 根目录下的假脱机目录 | 21 |
| 目录的组织形式 | 22 |
| 磁盘空间需求 | 23 |
| 安装帐户 | 23 |
| 文件访问权限 | 23 |
| 网络服务 | 24 |
| 主控主机 | 24 |
| 影像主控主机 | 24 |
| 执行主机 | 25 |
| 管理主机 | 25 |
| 提交主机 | 25 |
| 单元 | 25 |
| 用户名 | 25 |
| 队列 | 25 |
| ▼ 如何规划安装 | 26 |
| ▼ 如何读取发行媒体 | 27 |
| ▼ 如何安装主控主机 | 27 |
| ▼ 如何安装执行主机 | 28 |

- ▼ 如何安装管理和提交主机 29
 - 高安全性的安装 29
 - 所需的附加设置 30
- ▼ 如何安装和设置基于 CSP 的加密系统 30
- ▼ 如何为用户生成证书和私用密钥 39
- ▼ 如何检查证书 40
 - 显示证书 40
 - 查看颁发人 41
 - 查看主题 41
 - 显示证书的电子邮件 41
 - 显示有效期 42
 - 显示指纹 42
- ▼ 如何验证安装 42

第三部分 . 使用 Sun Grid Engine 5.3 软件

- 3. 浏览 Sun Grid Engine 5.3 程序 49
 - Sun Grid Engine 的用户类型和操作 49
 - 队列和队列特性 50
 - QMON 浏览器 50
 - ▼ 如何启动 QMON 浏览器 51
 - 队列控制 QMON 对话框 51
 - ▼ 如何显示队列的列表 52
 - ▼ 如何显示队列特性 52
 - 使用 QMON 浏览器 52
 - 从命令行 54
 - 解释队列特性信息 54
 - 主机功能 55

- ▼ 如何找到主控主机的名称 55
- ▼ 如何显示执行主机列表 55
- ▼ 如何显示管理主机列表 55
- ▼ 如何显示提交主机列表 56
- 可请求的属性 56
 - ▼ 如何显示可请求属性列表 57
 - 用户访问权限 60
 - 管理人员、操作人员和所有者 61
- 4. 提交作业 63
 - 运行简单作业 63
 - ▼ 如何从命令行运行简单作业 63
 - ▼ 如何从图形用户界面 QMON 提交作业 65
 - 提交批处理作业 69
 - 关于 Shell 脚本 69
 - 脚本文件示例 70
 - 用 QMON 提交扩展作业和高级作业 70
 - 扩展作业示例 70
 - 高级示例 74
 - 资源需求定义 78
 - Sun Grid Engine 系统如何分配资源 80
 - 常规 Shell 脚本的扩展 81
 - 如何选择命令解释器 81
 - 输出重定向 81
 - 有效的 Sun Grid Engine 注释 82
 - ▼ 如何从命令行提交作业 84
 - 缺省请求 85

| | |
|---------------------------|------------|
| 阵列作业 | 86 |
| ▼ 如何从命令行提交阵列作业 | 87 |
| ▼ 如何用 QMON 提交阵列作业 | 87 |
| 提交交互式作业 | 88 |
| 用 QMON 提交交互式作业 | 89 |
| ▼ 如何用 QMON 提交交互式作业 | 89 |
| 用 qsh 提交交互式作业 | 92 |
| ▼ 如何用 qsh 提交交互式作业 | 92 |
| 用 qlogin 提交交互式作业 | 92 |
| ▼ 如何用 qlogin 提交交互式作业 | 92 |
| 并行作业 | 93 |
| 如何调度 Sun Grid Engine 作业 | 93 |
| 作业优先级 | 93 |
| 等份额调度 | 93 |
| 队列选择 | 94 |
| 透明的远程执行 | 94 |
| 使用 qrsh 进行远程执行 | 94 |
| qrsh 用法 | 95 |
| 用 qtcsh 进行透明的作业分配 | 95 |
| qtcsh 用法 | 96 |
| 用 qmake 执行并行的 Makefile 处理 | 97 |
| qmake 用法 | 98 |
| 5. 点检查、监视和控制作业 | 101 |
| 关于点检查作业 | 101 |
| 用户级别的点检查 | 102 |
| 内核级别的点检查 | 102 |

- 点检查作业的迁移 102
- 编写点检查作业脚本 103
- ▼ 如何从命令行提交、监视或删除点检查作业 104
- ▼ 如何用 QMON 提交点检查作业 104
- 文件系统需求 105
- 监视和控制 Sun Grid Engine 作业 106
- ▼ 如何用 QMON 监视和控制作业 106
- 用 QMON 对象浏览器查看附加信息 115
- ▼ 如何用 qstat 监视作业 116
- ▼ 如何用电子邮件监视作业 118
- 从命令行控制 Sun Grid Engine 作业 118
- ▼ 如何从命令行控制作业 119
- 作业从属性 120
- 控制队列 120
- ▼ 如何用 QMON 控制队列 120
- ▼ 如何用 qmod 控制队列 124
- 自定义 QMON 125

第四部分 . 管理

- 6. 主机和群集配置 129
- 关于主控和影像主控配置 130
- 关于守护程序和主机 131
- 关于配置主机 132
- ▼ 如何用 QMON 配置管理主机 132
- ▼ 如何删除管理主机 134
- ▼ 如何添加管理主机 134
- ▼ 如何从命令行配置管理主机 134

- ▼ 如何用 QMON 配置提交主机 134
- ▼ 如何删除提交主机 136
- ▼ 如何添加提交主机 136
- ▼ 如何从命令行配置提交主机 136
- ▼ 如何用 QMON 配置执行主机 136
- ▼ 如何删除执行主机 138
- ▼ 如何关闭执行主机守护程序 138
- ▼ 如何添加或修改执行主机 138
- ▼ 如何从命令行配置执行主机 142
- ▼ 如何用 qhost 监视执行主机 143
- ▼ 如何从命令行中止守护程序 143
- ▼ 如何从命令行重新启动守护程序 144
- 基本群集配置 144
 - ▼ 如何从命令行显示基本群集配置 145
 - ▼ 如何从命令行修改基本群集配置 145
 - ▼ 如何用 QMON 显示群集配置 146
 - ▼ 如何用 QMON 删除群集配置 146
 - ▼ 如何用 QMON 显示全局群集配置 147
 - ▼ 如何使用 QMON 修改全局配置和主机配置 147

7. 配置队列和队列日历 149

- 关于配置队列 149
 - ▼ 如何用 QMON 配置队列 150
 - ▼ 如何配置常规参数 151
 - ▼ 如何配置“执行方法”参数 152
 - ▼ 如何配置“点检查”参数 153
 - ▼ 如何配置负荷和暂停阈值 154

- ▼ 如何配置“限制” 155
- ▼ 如何配置用户“属性组” 157
- ▼ 如何配置“从属队列” 158
- ▼ 如何配置“用户访问权限” 159
- ▼ 如何配置“拥有者” 160
- ▼ 如何从命令行配置队列 161
- 关于队列日历 162
- ▼ 如何用 QMON 配置队列日历 162
- ▼ 如何从命令行配置日历 165
- 8. 属性组概念 167**
 - 关于属性组 167
 - ▼ 如何添加或修改属性组配置 168
 - 属性组类型 169
 - 队列属性组 170
 - 主机属性组 170
 - 全局属性组 172
 - 用户定义的属性组 173
 - 可使用的资源 177
 - ▼ 如何设置可使用资源 177
 - 设置可使用资源的示例 179
 - 配置属性组 188
 - ▼ 如何从命令行修改属性组配置 188
 - qconf 命令示例 189
 - 负荷参数 189
 - 缺省负荷参数 189
 - 添加特定于站点的负荷参数 190

- ▼ 如何写您自己的负荷传感器 190
 - 规则 190
 - 脚本示例 191

9. 管理用户访问权限和策略 195

- 关于设置用户 195
- 关于用户访问权限 197
 - ▼ 如何用 QMON 配置帐户 197
 - ▼ 如何用 QMON 配置管理人员帐户 197
 - ▼ 如何从命令行配置管理人员帐户 198
 - 可用开关选项 198
 - ▼ 如何用 QMON 配置操作人员帐户 199
 - ▼ 如何从命令行配置操作人员帐户 200
 - 可用开关选项 200
 - 关于队列拥有者帐户 200
 - 关于用户访问权限 201
 - ▼ 如何用 QMON 配置用户访问列表 201
 - ▼ 如何从命令行配置用户访问列表 203
 - 可用选项 203
- 关于调度 203
 - 调度策略 204
 - 队列排序 204
 - 作业排序 204
 - 发生于调度间隔内的操作 205
 - 调度程序监视 205
 - 缺省调度 205
 - 调度方案 206

| | |
|---------------------|------------|
| ▼ 如何用 QMON 更改调度程序配置 | 209 |
| 关于路径别名 | 212 |
| 文件格式 | 213 |
| 如何解释路径别名文件 | 213 |
| 路径别名文件示例 | 213 |
| 关于配置缺省请求 | 214 |
| 缺省请求文件的格式 | 214 |
| 缺省请求文件的示例 | 215 |
| 关于收集帐户信息和利用统计信息 | 215 |
| 关于点检查支持 | 216 |
| 点检查环境 | 217 |
| ▼ 如何用 QMON 配置点检查环境 | 217 |
| 查看已配置的点检查环境 | 218 |
| 删除已配置的点检查环境 | 218 |
| 修改已配置的点检查环境 | 219 |
| 添加点检查环境 | 221 |
| ▼ 如何从命令行配置点检查环境 | 221 |
| qconf 点检查选项 | 221 |
| 10. 管理并行环境 | 223 |
| 关于并行环境 | 223 |
| ▼ 如何用 QMON 配置 PE | 224 |
| ▼ 显示 PE 内容 | 224 |
| ▼ 删除 PE | 224 |
| ▼ 修改 PE | 224 |
| ▼ 添加 PE | 225 |
| ▼ 如何从命令行配置 PE | 228 |

| | |
|--------------------------------|-----|
| qconf PE 选项 | 228 |
| ▼ 如何从命令行显示已配置的 PE 接口 | 228 |
| ▼ 如何用 QMON 显示已配置的 PE 接口 | 229 |
| PE 启动过程 | 230 |
| 终止 PE | 231 |
| PE 和 Sun Grid Engine 软件的紧密集成 | 232 |
| 11. 错误消息 | 233 |
| Sun Grid Engine 5.3 软件如何检索错误报告 | 233 |
| 不同错误或退出代码的后果 | 234 |
| 在调试模式下运行 Sun Grid Engine 程序 | 236 |

前言

《*Sun Grid Engine 5.3 管理和用户指南*》是一部内容全面的手册，它提供了产品的背景信息、安装指导以及充分利用该产品的指导。

本书结构

由于本指南是为 Sun Grid Engine 5.3 产品的用户和系统管理员准备的，而系统管理员的产品责任与一般用户并非总是一样，因此本指南分为四个部分。每一章都包含了对用户或管理员尤为重要的信息。

下面是每部分的描述及其针对的读者。

- 第一部分 – 背景和定义

《*Sun Grid Engine 5.3 管理和用户指南*》的这个部分同时针对用户和管理员，详细描述产品的用法、组件、术语等等。

- 第二部分 – 入门

本指南的这个部分针对安装产品的人员（通常是管理员），包含有关“快速启动”、全新安装和升级安装的详细指导。

- 第三部分 – 使用 Sun Grid Engine 5.3 软件

本指南的这个部分同时针对用户和管理员。其中包含关于许多任务的指导和背景信息。

- 第四部分 – 管理

本指南的这个部分中包含的背景信息和指导是针对资深系统管理员的。

使用 UNIX 命令

本文档可能不包括有关基本的 UNIX[®] 命令和过程（如关闭系统、引导系统和配置设备）的信息。

有关基本的 UNIX 命令和过程的信息，请参见以下文档：

- 《*Solaris Handbook for Sun Peripherals*》
- Solaris[™] 操作环境的 AnswerBook2[™] 联机文档资料
- 所用系统附带的其它软件文档资料

排印约定

| 字体 | 含义 | 示例 |
|------------------|-----------------------------------|--|
| AaBbCc123 | 命令、文件和目录的名称；计算机屏幕上的输出 | 编辑 .login 文件。 使用 ls -a 列出所有文件。 % You have mail. |
| AaBbCc123 | 所键入的内容，与计算机屏幕输出相区别。 | % su Password: |
| <i>AaBbCc123</i> | 书名、新词或术语以及要强调的词。请用实际名称或值来替代命令行变量。 | 请阅读 《 <i>用户指南</i> 》中的第六章。这些被称为类选项。 您 必须 是超级用户才能执行此操作。 要删除文件，请输入 rm <i>文件名</i> 。 |

Shell 提示符

| Shell | 提示符 |
|--------------------------------|-------|
| C shell | 机器名 % |
| C shell 超级用户 | 机器名 # |
| Bourne shell 和 Korn shell | \$ |
| Bourne shell 和 Korn shell 超级用户 | # |

相关文档资料

| 应用 | 书名 | 部件号 |
|----|--|-------------|
| 参考 | 《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》 | 816-7478-10 |

访问 Sun 联机文档资料

请在以下网址查询关于 Sun 系统的各种文档资料：

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

请在以下网址查询关于 Solaris 的全套文档资料以及其它多种书目：

<http://docs.sun.com>

在此站点，还可找到有关如何订购本指南印刷稿的信息。

Sun 欢迎您提出宝贵意见

Sun 致力于提高文档资料的质量，欢迎您提出宝贵意见和建议。您可以将意见通过电子邮件发送给 Sun，地址如下：

`docfeedback@sun.com`

请在电子邮件的主题行中注明文档的部件号 (816-7465-10)。

第一部分 背景和定义

《*Sun Grid Engine 5.3 管理和用户指南*》中的第一部分只包含一章：

- 第一章 – 第 1 页的 “Sun Grid Engine 5.3 简介”。

本章的简洁可能会使读者误认为本章对用户和管理员不太重要，但是熟悉本章的内容会使二者都受益匪浅。本章包含以下内容。

- Sun™ Grid Engine 5.3 软件在复杂计算环境中的主要职能的说明
- 本产品的主要组件的列表以及每个组件的功能的定义
- Sun Grid Engine 5.3 环境中需要了解的重要术语的词汇表

Sun Grid Engine 5.3 简介

本章提供了有关 Sun Grid Engine 5.3 系统的背景信息，这些信息对用户和管理员都有用。本章描述了如何使用该产品管理计算机群集，使之不会杂乱无章。除此之外，本章还包括以下主题：

- 网格计算的简单解释
- QMON — Sun Grid Engine 5.3 用户图形界面的概述
- 该产品各个重要组件的解释
- 可供用户和管理员使用的客户端命令的详细列表
- Sun Grid Engine 5.3 术语的完整词汇表

什么是网格计算？

从概念上讲，网格很简单。它是执行任务的计算资源的集合。对用户而言，它就是一个大系统，提供单个切入点，以访问强大而分散的资源。用户将网格视作单个计算资源。资源管理软件（如 Sun Grid Engine）接受由用户提交的作业，并根据资源管理策略将作业安排在网格内适当的系统上执行。用户可以一次提交数千个作业，而不必考虑它们在何处运行。

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。主要有三种类型的网格，其规模可以小到单个系统，也可以大到使用数千个处理器的超型计算机级的计算区域。*群集网格*是由许多协同工作的计算资源组成的，为某一项目或部门的用户提供单个的访问切入点，由 Sun Grid Engine 5.3 系统帮助您创建和管理。

（另外两种更复杂的网格类型（*公司网格*和*全球网格*）则由 Sun、Sun Grid Engine（企业版）的相关产品创建和管理。）

Sun Grid Engine 5.3 软件根据组织内的技术人员和管理人员所设置的企业资源策略，协调指挥计算资源的交付。Sun Grid Engine 系统运用这些策略检查并收集网格内的可用计算资源，然后按整个网格内的最优化使用方案自动地分配和交付这些资源。

通过管理资源和策略管理工作负荷

Sun Grid Engine 软件为用户提供了向 Sun Grid Engine 系统提交要求计算的任务的方法，以透明地分配相关的工作负荷。用户可以向 Sun Grid Engine 系统提交批处理作业、交互式作业和并行作业。

本产品也支持点检查程序。点检查作业可从一个工作站迁移到另一个工作站，而不需要用户干涉负荷需求。

对管理员而言，此软件提供了监视和控制 Sun Grid Engine 作业的一系列工具。

系统是如何运作的

Sun Grid Engine 系统从外部接受作业（用户对计算机资源的请求），将这些作业存放在等候区域直至可以执行它们，将它们从等候区域送往执行设备，随后在执行过程中进行管理，最后在整个过程结束后将执行情况记录下来。

可以想象一家位于世界上某个金融城市的巨大的“货币中心”银行。

使资源与请求相匹配

在银行大楼的大厅里有许许多多的客户，每个客户的需求不尽相同，他们都在等候服务。其中一个客户只想从自己的帐户中取出一小笔钱。而紧跟在他后面的客户则预约了银行的某位投资专员；她想在进行复杂的风险投资之前寻求一些建议。在长队中，排在这两位之前的另一位客户想申请一大笔贷款；排在她前面的八位客户也是抱此目的而来。

不同的客户和不同的意向需要不同类型和级别的银行资源。也许当天银行有许多员工并且有充足的时间来处理客户从帐户取钱的简单交易。但也许那天只有一两个负责放贷的官员为众多的贷款申请人服务。另一天，也许情况会相反。

结果当然是客户们只得等待服务，即使对于很多客户来说，只要他们的需求能立即被识别并有与之相匹配的可用资源，他们就可以立即得到服务。

如果把 Sun Grid Engine 系统当作银行经理，那么它提供服务的方式将完全不同。

- 客户一进入银行大厅就需要通报姓名、从属关系（如代表某公司）及其需求。
- 客户到达的时间将被记录下来。
- 根据客户在大厅中提供的信息，那些所提需求能立即得到满足的客户（具有适当的资源）、所提需求优先级最高的客户以及在大厅里等候多时的客户将获得服务。
- 当然，在“Sun Grid Engine 银行”中，每位银行职员可能同时为多个客户提供帮助。Sun Grid Engine 系统尽量将新的客户分配给工作量最小并且最为适合的银行职员。

作业和队列：Sun Grid Engine 的世界

在 Sun Grid Engine 系统中，*作业*可对应于银行客户，作业在计算机的等候区域而非大厅等待，位于计算机服务器上的*队列*相当于银行的职员，它们为作业提供服务。如类比中的银行客户一样，每个作业的需求（通常包括可用内存、执行速度、可用软件许可证及类似的需求）可能大相径庭，并且可能只有某些队列才能提供相应的服务。

与类比相对应，Sun Grid Engine 软件将用以下方式调解可用资源和作业需求。

- 通过 Sun Grid Engine 系统提交作业的用户描述出作业需求的概况。此外，系统还要检索用户的身份以及他或她与*项目*或*用户组*的从属关系。用户提交作业的时间也将存储起来。
- 准确地说，在队列被定为可以对新作业执行操作的那一刻，Sun Grid Engine 系统就决定了适合该队列的作业，并立即分派具有最高优先级或等待时间最长久的作业。
- Sun Grid Engine 队列允许同时执行许多作业。Sun Grid Engine 系统将尽量在负荷最小且最适合的队列中开始新的作业。

Sun Grid Engine 5.3 组件

图 1-1 显示 Sun Grid Engine 最重要的组件和它们在系统中的相互作用。以下各节介绍各个组件的功能。

主机

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

- 主控
- 执行
- 管理
- 提交

主控主机

主控主机是所有群集活动的中心。它可运行主控守护程序 `sge_qmaster` 和调度守护程序 `sge_schedd`。两个守护程序都控制 Sun Grid Engine 的所有组件（如队列和作业），并维护组件状态表和用户访问权限表，等等。

缺省情况下，主控主机还是管理主机和提交主机。请参见有关这些主机的部分。

执行主机

执行主机是有权执行 Sun Grid Engine 作业 的节点。因此，该主机上有 Sun Grid Engine 队列，并运行 Sun Grid Engine 执行守护程序 `sge_execd`。

管理主机

可以赋予主机权限，使之执行任何种类的 Sun Grid Engine 系统管理活动。

提交主机

提交主机仅允许提交和控制*批处理作业*。而且，登录到提交主机的用户可以使用 `qsub` 提交作业，使用 `qstat` 控制作业状态，并使用 Sun Grid Engine OSF/1 Motif 图形用户界面 QMON（在第 8 页的“Sun Grid Engine 图形用户界面 QMON”一节中对它进行了描述）。

注意 – 一台主机可能属于上述的一个或多个类别。

守护程序

四种守护程序提供了 Sun Grid Engine 5.3 系统的功能。

`sge_qmaster` – 主控守护程序

`sge_qmaster` 是群集管理和调度活动的中心，它维护主机表、队列表、作业表、系统负荷表以及用户权限表。它从 `sge_schedd` 接收调度决定，并请求从适当执行主机上的 `sge_execd` 进行处理。

`sge_schedd` – 调度守护程序

调度守护程序在 `sge_qmaster` 的帮助下，维护群集状态的最新视图。它所作的调度决定有：

- 将哪些作业分派到哪些队列

然后，守护程序将这些决定转发至 `sge_qmaster`，后者将启动所需的操作。

`sge_execd` – 执行守护程序

执行守护程序负责其主机上的队列，以及这些队列中的作业的执行。它会定期将信息（如主机上的作业状态或负荷）转发给 `sge_qmaster`。

`sge_commd` – 通讯守护程序

通讯守护程序通过公知的 TCP 端口进行通讯。用于 Sun Grid Engine 组件之间的所有通讯。

队列

Sun Grid Engine 队列是一个容器，它包含了可以在某主机上同时执行的同类别作业。队列决定作业的某些属性；例如此作业是否可迁移。运行的作业在其有效期内一直与它们的队列相关联。这种与队列的关联性会在某些方面影响作业。例如，若队列暂停，与该队列关联的所有作业也将暂停。

在 Sun Grid Engine 系统中，没有必要直接将作业提交至队列。只需指定作业的需求概况（如内存、操作系统、可用软件等），然后 Sun Grid Engine 软件会自动将作业分派给低负荷的主机上的适当队列。当作业提交到某个队列后，作业将绑定到此队列及其主机，这时，Sun Grid Engine 守护程序将无法为其选择负荷更低或更为适合的设备。

客户端命令

Sun Grid Engine 命令行用户界面是一组辅助程序（命令），运用这些命令可以管理队列、提交和删除作业、检查作业状态以及暂停 / 启用队列和作业。Sun Grid Engine 系统使用下列一组辅助程序。

- qacct – 此命令从群集日志文件中抽取任意帐户信息。
- qalter – 此命令更改已提交但正处于暂挂状态的作业的属性。
- qconf – 此命令为群集和队列配置提供用户界面。
- qdel – 用户、操作人员或管理人员可使用此命令向作业或其子集发送信号。
- qhold – 此命令阻止已提交作业的执行。
- qhost – 此命令显示 Sun Grid Engine 执行主机的状态信息。
- qlogin – 此命令启动 telnet 或类似的登录会话，并自动选择负荷较低并且较为适合的主机。
- qmake – 此命令可取代标准的 UNIX make 命令工具。它扩充了 make 的功能，能够将相互独立的 make 步骤分配到一组适合的机器。
- qmod – 此命令使拥有者可以暂停或启用队列（将信号发送给当前与此队列相关的所有活动进程）。
- qmon – 此命令提供了 X-windows Motif 命令界面和监视工具。
- qresub – 此命令通过复制正在运行或暂挂的作业，创建新的作业。
- qrls – 此命令释放先前被阻止执行的作业，例如通过 qhold（见上）阻止执行。
- qrsh – 此命令用途很多，比如：
 - 提供通过 Sun Grid Engine 系统执行的远程交互应用程序 — 与标准的 UNIX 命令工具 rsh 相似
 - 允许提交批处理作业，一经执行便可支持终端 I/O（标准 / 错误输出和标准输入）以及终端控制
 - 提供批处理作业提交客户机，该客户机在作业完成之前一直保持活动状态
 - 允许 Sun Grid Engine 软件控制并行作业的任务远程执行
- qselect – 此命令显示与指定选择标准相对应的队列名称列表。qselect 的输出结果通常送往其它 Sun Grid Engine 命令，以便对选定的一组队列执行操作。

- `qsh` – 此命令在负荷较低的主机上打开交互式 shell（在 `xterm` 中）。所有类型的交互式作业均可以在此 shell 内运行。
- `qstat` – 此命令列出所有与群集相关的作业和队列的状态。
- `qsub` – 此命令是将作业提交到 Sun Grid Engine 系统的用户界面。
- `qtcsch` – 此命令与众所周知并普遍使用的 Unix C-Shell (`csh`) 派生物 `tcsch` 完全兼容，并能替代它。它扩展了命令 shell 的功能，即通过 Sun Grid Engine 软件将指定应用程序的执行透明地分配给适合的并且负荷较低的主机。

所有程序通过 `sge_commd` 与 `sge_qmaster` 进行通讯。这一点可以从 Sun Grid Engine 系统内的组件相互作用示意图（如图 1-1 所示）中反映出来。

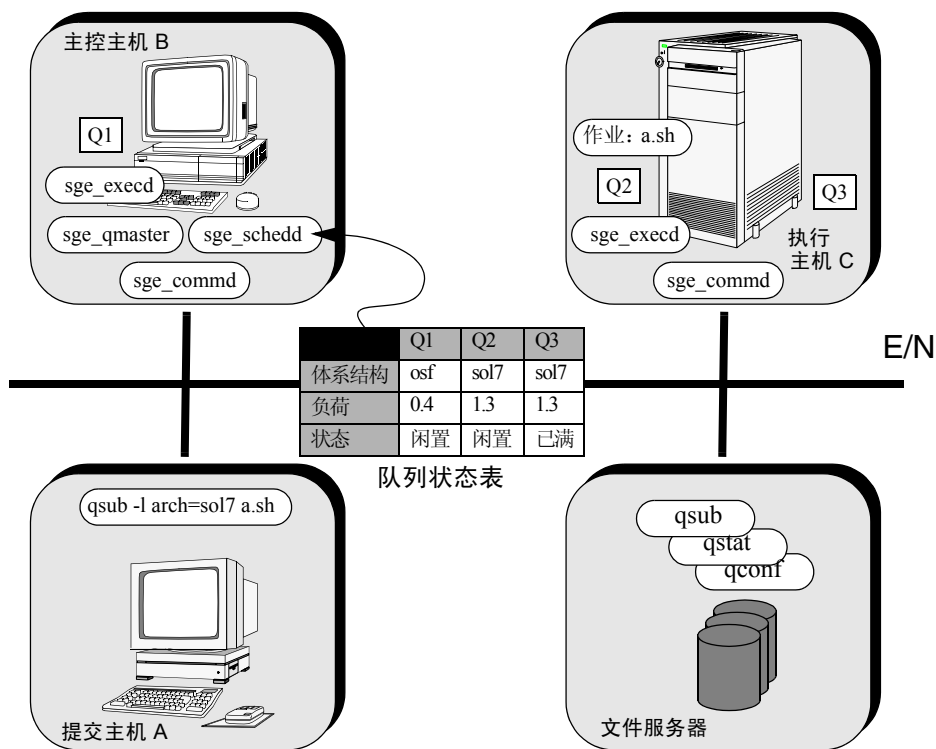


图 1-1 Sun Grid Engine 系统内的组件相互作用

Sun Grid Engine 图形用户界面 QMON

使用 QMON 图形用户界面 (GUI) 工具，可以完成大多数以至全部 Sun Grid Engine 5.3 任务。图 1-2 显示 QMON 的主菜单，它通常是用户和管理员功能的入口。主菜单上的每个图标都是一个 GUI 按钮，可点击这些按钮启动各种任务。当将鼠标置于屏幕的按钮上时，会显示该按钮的名称，该名称同时也是按钮功能的描述。

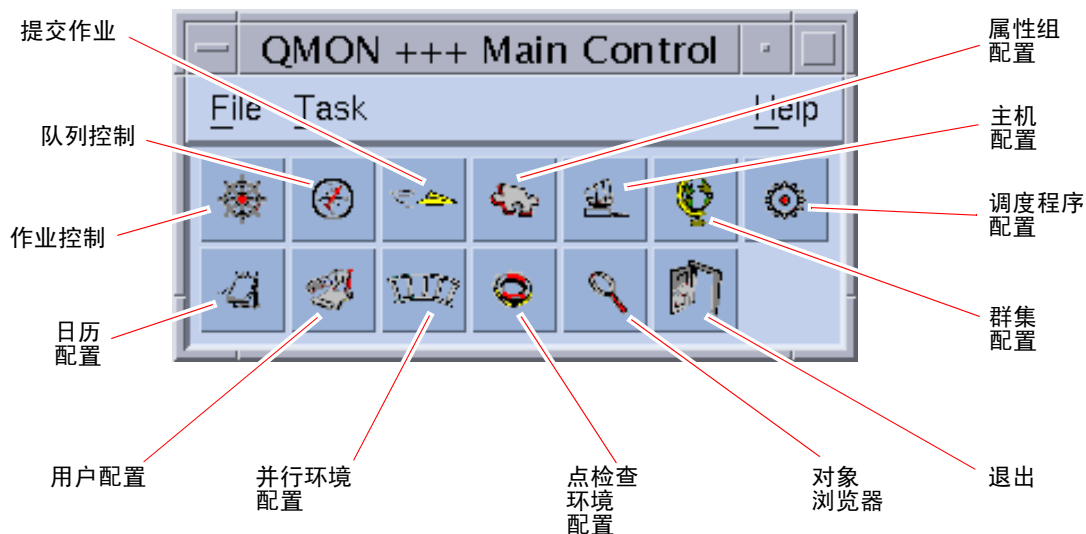


图 1-2 QMON 主菜单定义

自定义 QMON

qmon 的外观主要由专门的资源文件定义。已应用了合理的缺省值，并且在 `<sge 根目录>/qmon/Qmon` 下有样本资源文件。

通过将 qmon 的特定资源定义放入标准的 `.Xdefaults` 或 `.Xresources` 文件，或将站点的特定 Qmon 文件放入标准搜索路径（如 `XAPPLRESDIR`）所引用的位置，群集管理者可以将站点的特定缺省值安装在标准位置，如 `/usr/lib/X11/app-defaults/Qmon`。遇到上述情况，请向管理员查询。

除此之外，用户还可以配置个人首选项，方法之一是：将 Qmon 文件复制到主目录（或私有 XAPPLRESDIR 搜索路径指向的另一个位置）内并修改它；方法之二是：将必要的资源定义包含到用户的私有 .Xdefaults 或 .Xresources 文件中。也可以在操作或启动 X11 环境（比如执行 .xinitrc 资源文件）时，使用 xrdb 命令安装私有 Qmon 资源文件。

有关可能的自定义的详细信息，请参见样本 Qmon 文件内的注释行。

还可以在图 5-3 和图 5-13 所示的作业控制和队列控制自定义对话框内，对 qmon 进行自定义。不论在哪个对话框，都可以使用“保存”按钮，将通过自定义对话框配置的有关过滤和显示的定义存储到位于用户主目录下的 .qmon_preferences 文件中。一旦重新启动，qmon 将读取此文件并重新激活先前定义的运作方式。

Sun Grid Engine 术语词汇表

词汇表对 Sun Grid Engine 环境和常规资源管理中经常用到的术语进行了简短概括。到目前为止，还有许多术语尚未使用过，但会出现在 Sun Grid Engine 文档资料的其它部分。

| | |
|---------------|--|
| 访问列表 | 被允许或被拒绝访问某资源（如队列或某主机）的用户和 UNIX 组的列表。用户和组可以属于多个访问列表，而且同一个访问列表可以用于不同的环境。 |
| 单元 | 具有独立配置和主控主机的独立 Sun Grid Engine 群集。可以用单元松散地联合独立的管理设备。 |
| 点检查 | 将作业的执行状态保存到所谓的检查点的过程，以使中止的作业过一段时间后恢复执行，而不丢失任何信息和已完成的作业。若检查点在作业恢复执行前移动到了另一台主机，则此进程称为迁移。 |
| 点检查环境 | Sun Grid Engine 配置实体，它定义与某种点检查方法相关的事件、界面和操作。 |
| 群集 | 一组机器（即所谓的主机），在其上执行 Sun Grid Engine 功能。 |
| 属性组 | 与队列、主机或整个群集相关的一组属性。 |
| 组 | UNIX 组。 |
| 硬性资源需求 | 必须在作业启动前分配的资源。与软性资源需求相对。 |
| 主机 | 执行 Sun Grid Engine 功能的机器。 |

| | |
|---------------|---|
| 作业 | 批处理作业是 UNIX shell 脚本，它的执行无需用户的干预，而且无需访问终端。 交互式作业是用 Sun Grid Engine 命令 qssh、qsh 或 qlogin 启动的一个会话，这些命令会打开 <i>xterm</i> 窗口以供用户交互式操作或提供相当于远程登录会话的界面。 |
| 作业阵列 | 包含一系列相互独立的相同任务的作业。每个任务都非常类似于一个单独的作业。唯一的区别是作业阵列任务有一个唯一的任务标识符（一个整数）。 |
| 作业类别 | 从某种意义上说相同且待遇相似的一组作业。在 Sun Grid Engine 中，作业类别是由相应作业的相同请求和适用于那些作业的队列特性共同定义的。 |
| 管理人员 | 可以全面控制 Sun Grid Engine 的用户。主控主机的超级用户以及其他任何充当管理主机的机器的超级用户都有管理人员特权。管理人员特权也可以分配给非 root 用户帐户。 |
| 迁移 | 在作业恢复执行前，将检查点从一台主机移动到另一台主机的进程。 |
| 操作人员 | 此用户不能改变配置而只能进行维护操作，除此之外，其可执行的命令与管理人员相同。 |
| 拥有者 | 这种用户可以暂停 / 取消暂停，并禁用 / 启用其拥有的队列。通常，用户是其工作站上的队列的拥有者。 |
| 并行环境 | Sun Grid Engine 的配置实体，它定义必要接口，以使 Sun Grid Engine 正确处理并行作业。 |
| 并行作业 | 这种作业包含多个密切相关的任务。任务可以分布在多台主机上。并行作业通常使用通讯工具（如共享内存或消息传递 (MPI, PVM)）来同步和联络任务。 |
| 策略 | 一组规则和配置，Sun Grid Engine 管理员可用它定义 Sun Grid Engine 的运作方式。策略由 Sun Grid Engine 自动执行。 |
| 优先级 | 与其它作业相比，某一 Sun Grid Engine 作业的相对重要级别。 |
| 队列 | 它包含了某一类别的多个作业，这些作业可以同时执行在 Sun Grid Engine 执行主机上。 |
| 资源 | 由正在运行的作业消耗或占用的计算设备。典型的例子有：内存、CPU、I/O 带宽、文件空间、软件许可证等等。 |
| 软性资源需求 | 作业所需的资源，但这些资源并不必在作业启动前进行分配。当资源可用时才分配给作业。与 硬性资源需求 相对。 |
| 暂停 | 阻止作业运行，但仍将其保留在执行主机上（这与中止作业的点检查不同）。暂停的作业仍会消耗一些资源，如交换内存或文件空间。 |
| 用户 | 可提交作业至 Sun Grid Engine 并用它来执行作业，条件是他或她有权登录至少一台提交主机和一台执行主机。 |
| 用户组 | 访问列表（见上）。 |

第二部分 入门

《*Sun Grid Engine 5.3 管理和用户指南*》中的这个部分只包含一章。

- 第二章 – 第 13 页的“安装”

本章将对 Sun Grid Engine 5.3 产品的初次安装进行指导，并介绍如何将现有旧版本升级为新版本。

安装

本章描述并提供了三种安装任务的详细指导：

- “快速启动”安装 — 不推荐用于所有站点（请参见第 13 页的“快速启动安装概述”）
- Sun Grid Engine 5.3 软件的全新安装（请参见第 19 页的“完全安装概述”）
- 使用特殊加密功能的安全安装（请参见第 30 页的“如何安装和设置基于 CSP 的加密系统”）

注意 – 本章中的指导假定您是在运行 Solaris™ 操作环境的计算机上安装本软件。Sun Grid Engine 在其它操作系统体系结构上运行所导致的功能差异，均在以 `arc_depend_` 开头的文件中列出，这些文件位于 `<sgе 根目录>/doc` 目录。文件名的其余部分则指明这些文件中的注释所适用的操作系统体系结构。

快速启动安装概述

注意 – 本节说明要使快速启动安装步骤生效而必须在您的站点上满足的条件。若您的环境不包含所描述的所有先决条件，则无法使用快速启动安装步骤。在这种情况下，请转至第 19 页的“完全安装概述”一节，以获得有关如何在更多的限制条件下安装 Sun Grid Engine 5.3 软件的详细信息。

快速启动安装的先决条件

注意 – 本节的指导仅适用于 Sun Grid Engine 5.3 的 *全新安装*。有关如何升级现有旧版本的 Sun Grid Engine 产品的指导，请参见 《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 发行说明*》。

本节说明在执行快速启动安装之前必须在您的站点上已存在或要创建的几个条件。

- 安装账户
- 安装目录
- 通讯端口号

安装帐户

对于快速启动安装，必须存在 *管理员* 账户，或根据以下指导创建一个账户。

- 管理员可以是一个现有的管理登录账户，也可以是一个新的登录账户，例如 sgeadmin。

此账户将拥有 Sun Grid Engine 安装和假脱机目录中的所有文件，并且一旦安装就可用于配置和管理群集。

- 此用户不应为 *root 用户*。

不过，若您想要对文件拥有权使用 *root 用户*，则 *root 用户* 必须对安装 Sun Grid Engine 程序的目录下的所有主机有写权限。通常情况下，不为 *root 用户* 导出共享的 (NFS) 文件系统以允许写权限。

- 此账户在安装之前必须存在。

安装目录

要为管理员账户的使用作准备，您应创建安装目录，它最好位于网络范围内共享的文件系统上。

▼ 如何创建安装目录

- 请使用以下一系列命令。

```
% mkdir -p <安装目录>
% chown <管理员用户> <安装目录>
% chmod 755 <安装目录>
```

注意 – 按照此步骤或类似步骤创建的目录在本手册的其余部分称作 Sun Grid Engine 根目录。

通讯端口号

Sun Grid Engine 系统使用 TCP 端口进行通讯。群集中的*所有*主机必须使用相同的端口号。可将端口号置于几个位置中的任何一个。以下为两个示例。

- NIS 服务或 NIS+ 数据库

添加如下语句至服务数据库。

```
% sge_commd535/tcp
```

- 每台计算机上的 /etc/services

若您的站点上并未运行 NIS，则可添加上述服务至每台主机上的 /etc/services 文件中。

最好使用 600 以下的特权端口，以免与动态地将端口绑定在接近或低于 1024 或高于 1024 的应用程序发生冲突。

读取发行媒体

Sun Grid Engine 5.3 软件通过 Internet 下载以存档文件的形式发行。该发行文件包含一个直接写入媒体的磁带存档文件 (tar)。

▼ 如何解压缩 Sun Grid Engine 5.3 发行文件

1. 以您所选择的安装账户（请参见第 14 页的“快速启动安装的先决条件”一节），登录到您打算要读取 Sun Grid Engine 发行媒体的主机。
2. 将工作目录切换至 Sun Grid Engine 根目录。
3. 请使用以下命令读取发行媒体。

```
% cd sge 根目录  
% tar -xvpf 源发行文件
```

在上述命令中，*sge 根目录* 是 Sun Grid Engine 根目录的路径名，而*源发行文件* 是硬盘上的磁带存档文件名。

为群集安装缺省的 Sun Grid Engine 系统

缺省的 Sun Grid Engine 系统由一台*主控主机* 和任意数目的*执行主机* 组成。主控主机控制整个群集的活动，而执行主机控制由主控主机分配给它们的作业的执行。一台主机可同时充当主控主机和执行主机。

注意 – 首先安装主控主机，然后以任意顺序安装执行主机并结束。

▼ 如何安装主控主机

1. 选择一台计算机作为主控主机。

在选择主控主机时请遵循以下条件。

- 所选的系统不会因其它任务而负荷超载。
- 主控主机应能提供足够的可用主内存，以运行必需的 Sun Grid Engine 守护程序。

所需的主内存量极大地依赖于群集的大小和系统中预期的作业数。对于多达数十台主机和数百项作业的群集，10 MB 的空闲内存应该足够了。

对于有数千台或更多主机以及数万项作业的大型群集，也许需要 1 GB 内存。

2. 登录到所选的计算机。

对于包含所有功能的安装，必须使用 *root 用户* 账户进行安装（文件仍由在第 14 页的“快速启动安装的先决条件”一节中创建的管理员所拥有）。

对于测试安装，您也可以作为管理员用户进行安装，但这样就只有管理员可以运行作业，并且 Sun Grid Engine 系统在监视系统负荷和系统控制方面的功能将受到限制。

3. 切换目录 (cd) 至 Sun Grid Engine 根目录。

4. 通过以下命令执行主控主机安装程序。

```
% ./install_qmaster
```

如果出现错误，则安装程序显示错误情况说明。

5. 回答安装脚本您要安装的主机名。

安装脚本要您提供最初想要安装的主机列表。列出所有这样的主机，这些主机将作为*提交*主机和*管理*主机添加。

执行主机的安装（请参见下节）要求所有主机均为管理主机。若您想要在多台主机上安装 Sun Grid Engine 程序，则安装脚本允许您提供包含所有主机名列表（每台主机一行）的文件的途径。

6. 回答安装脚本所有余下的提问。

安装程序需要一些附加信息。大多数问题将提供有用的缺省值，可通过按回车键确认。

▼ 如何安装执行主机

1. 根据以下指导，选择安装所使用的账户。

- 与主控主机的安装一样，您应使用 *root* 用户账户来安装执行主机，从而对所有 Sun Grid Engine 工具均有访问权限。

作为 *root* 用户账户安装仍保留管理员账户（在第 14 页的“快速启动安装的先决条件”一节中创建）的文件拥有权。

- 通过管理员账户进行安装的方式只可用于测试目的，并且禁止除管理员以外的其他用户运行作业。它不允许 Sun Grid Engine 程序提供完整的系统监视和控制功能。

2. 以选定的安装账户登录到某一在主控主机安装程序中指定的执行主机。

3. 切换目录 (cd) 至 Sun Grid Engine 根目录。

4. 输入以下命令启动执行主机安装程序：

```
% ./install_execd
```

任何错误均会由安装程序指出。

5. 回答安装脚本的提问。

安装程序将询问您是否要为主机配置缺省队列。这种情况下，定义的队列将具有如第 18 页的“缺省系统配置”一节中所述的特征。

6. 当安装脚本通知您执行主机安装程序成功完成时，请对在主控主机安装过程中指定的每台执行主机继续执行类似操作。

完成该列表中所有主机的安装后，缺省的 Sun Grid Engine 系统就已在群集中配置并准备就绪可供使用。要进行一些使用该程序的练习，请参见第 63 页的“如何从命令行运行简单作业”一节。

以下各节是安装的缺省配置的概述。

缺省系统配置

注意 – 以下是 Sun Grid Engine 系统的说明，由快速启动安装过程在环境中配置。它是用于测试目的的最小设置，可在以后随时更改或扩展。

主控主机和执行主机安装完成之后，群集中已配置了以下基本的 Sun Grid Engine 系统。

■ 主控主机

运行主控主机安装程序的主机配置为群集的主控主机。并未配置在发生故障的情况下接管主控主机的任务的影像主控主机。

■ 执行主机

在主控主机的安装过程中，将要求您提供要安装 Sun Grid Engine 执行代理的计算机的列表。在这些执行主机的安装过程中，您可以让安装程序在这些主机上自动创建队列。队列说明可在该主机上运行的作业的概况（一个属性和需求列表）。缺省情况下，为执行主机配置的队列具有以下重要特征。

- 队列名：< 无限制的主机名 >.q
- 位置数（同时运行的作业）：< 处理器数 >
- 队列为作业提供无限制的系统资源（例如内存、CPU 时间等）。
- 队列不对特定用户或用户组执行任何访问限制。任何具有有效账户的用户均可在队列中运行作业。
- 将配置负荷阈值为 1.75（每个 CPU）（即平均有 1.75 个进程试图访问每个 CPU）。

注意 – 如任何其它 Sun Grid Engine 配置一样，队列配置可在以后的系统运行中随时更改。

注意 – 若在主控主机上也调用了执行主机安装程序，则该主控主机既充当主控主机，也充当执行主机。

■ 管理帐户和主机

主控主机和所有执行主机均配置为允许执行 Sun Grid Engine 管理命令。只允许 *root* 用户和如第 14 页的“快速启动安装的先决条件”一节中所述的管理员账户可管理 Sun Grid Engine。若一位非特权用户安装 Sun Grid Engine 程序，则他或她将被添加到 Sun Grid Engine 管理员列表中。

■ 提交帐户和主机

若您在 *root* 用户账户下安装程序，则任何具有有效账户的用户均可提交和控制 Sun Grid Engine 作业。否则，则只有安装 Sun Grid Engine 软件时使用的用户帐户才有访问权限（请参见第 14 页的“快速启动安装的先决条件”）。从主控主机或任何执行主机均可执行提交作业、控制 Sun Grid Engine 系统活动或删除作业等任务。

■ 守护程序

以下守护程序分别在系统安装过程中在不同主机上启动，或可能在正常系统操作过程中调用。

- `sgc_qmaster` 只在主控主机上运行。它是重要的群集活动控制守护程序。
- `sgc_schedd` 也只能在主控主机上调用。此守护程序用于在 Sun Grid Engine 群集中分配工作负荷。
- `sgc_execd` 用于在执行主机上执行作业，因此它运行在所有执行主机上。
- 主机上实际执行的每项作业运行一个 `sgc_shepherd` 的实例。
`sgc_shepherd` 控制作业进程分层结构并在作业结束后收集帐户统计数据。
- `sgc_commd` 运行在每台执行主机和主控主机上。所有 `sgc_commd` 的网络组成了 Sun Grid Engine 群集网络通讯的主干。

完全安装概述

注意 – 本节的指导仅适用于 Sun Grid Engine 5.3 的 *全新* 安装。有关如何升级现有旧版本的 Sun Grid Engine 产品的指导，请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）发行说明*》。

完全安装包含以下几项主要任务。

- 规划 Sun Grid Engine 配置和环境
- 将 Sun Grid Engine 发行文件从外部媒体读取到工作站
- 在 Sun Grid Engine 系统内的主控主机和所有执行主机上运行安装脚本
- 注册有关管理和提交主机的信息
- 验证安装

应由熟悉 Solaris 操作环境的人员进行安装。整个安装程序分三个阶段。

阶段 1 – 规划

注意 – 若您想使安装的系统更加安全，在继续安装前，请参见第 29 页的“高安全性的安装”。

安装的规划阶段包含以下任务。

- 决定将 Sun Grid Engine 环境设置为单个群集，还是一组子群集（称作单元）的集合
- 选择用作 Sun Grid Engine 主机的机器。决定每台机器的主机种类 — 主控主机、影像主控主机、管理主机、提交主机、执行主机或它们的组合
- 确保所有的 Sun Grid Engine 用户在所有提交和执行主机上拥有相同的用户 ID
- 决定 Sun Grid Engine 目录的组织形式。例如，可以决定在每台工作站上将目录组织为一棵完整的树、或交叉装入目录、或在某些工作站上建立部分目录树。以及每个 Sun Grid Engine 根目录的位置
- 决定站点的队列结构
- 决定是将网络服务定义为 NIS 文件，还是将其放在每个工作站的本地 `/etc/services` 中
- 填写安装工作清单（请参见第 26 页的步骤 1，“开始安装前，在与下表相似的表内填写安装规划。”），以备在后续安装步骤中使用

阶段 2 – 安装软件

安装阶段包括以下任务。

- 创建安装目录，并在此目录内加载发行文件
- 安装主控主机
- 安装所有执行主机
- 注册所有管理主机
- 注册所有提交主机

阶段 3 – 验证安装

验证安装阶段包括以下任务。

- 检查守护程序是否正在主控主机上运行
- 检查守护程序是否正在所有执行主机上运行
- 检查 Sun Grid Engine 执行简单命令的情况
- 提交测试作业

规划安装

开始安装 Sun Grid Engine 5.3 软件以前，必须仔细地规划如何才能取得适合环境的最佳效果。本节将帮助您做重要决定，这些决定将影响安装程序的其余部分。

先决任务

以下各节将描述安装 Sun Grid Engine 系统所需的信息。

安装目录 *<sge 根目录>*

准备一个目录，用于读取 Sun Grid Engine 发行媒体上的内容。此目录称作 Sun Grid Engine *根目录*，以后在 Sun Grid Engine 系统的操作过程中，它将用来存储当前群集的配置和所有需假脱机到磁盘的新增数据。

为目录使用一个可以被所有主机正确引用的路径名。例如，如果文件系统是通过自动装入程序装入的，则将 *<sge 根目录>* 设置为 `/usr/SGE`，而不是 `/tmp_mnt/usr/SGE`。（本文中，每当提及安装目录时，都使用 *<sge 根目录>* 环境变量。）

<sge 根目录> 是 Sun Grid Engine 目录树的最顶层。启动时，单元（请参见第 25 页的“单元”一节）中的所有 Sun Grid Engine 组件都必须有读取 *<sge 根目录>* /*<单元>*/common 的权限。有关所需权限的描述，请参见第 23 页的“文件访问权限”一节。

为便于安装和管理，所有将要执行 Sun Grid Engine 安装步骤的主机都必须可以读取此目录。例如，可以选择一个能够通过网络文件系统（如 NFS）访问的目录。如果选择主机本地的文件系统，就必须在每台机器上开始执行安装程序前，将安装目录复制到本地。

根目录下的假脱机目录

- 在 Sun Grid Engine 主控主机上，假脱机目录位于 *<sge 根目录>* /*<单元>* /spool/qmaster 和 *<sge 根目录>* /*<单元>* /spool/schedd 下。
- 所有执行主机上都有称作 *<sge 根目录>* /*<单元>* /spool/*<执行主机>* 的假脱机目录。

不必向其它机器导出这些目录。然而，若将整个 *<sge 根目录>* 树导出，并使主控主机和所有执行主机都拥有对此树的写权限，会便于管理。

目录的组织形式

决定 Sun Grid Engine 目录的组织形式（例如，每个工作站上一棵完整的树、交叉装入目录或在某些工作站上建立部分目录树），并决定每个 Sun Grid Engine 根目录（即 <code>sge 根目录</code>）的位置。

注意 – 由于变更安装目录和 / 或假脱机目录的位置将要求重新安装系统（尽管上一次安装的重要信息能够保留下来），因此应格外谨慎地选择适当的安装目录。

缺省情况下，Sun Grid Engine 安装程序会按照目录分层结构（请参见第 22 页的图 2-1，“目录分层结构示例”）将 Sun Grid Engine 系统、手册页、假脱机区域和配置文件安装到安装目录下。如果接受这种缺省设置，则应该安装 / 选择一个有访问权限的目录（请参见第 23 页的“文件访问权限”）。

在安装的初始阶段，可以选择将假脱机区域放在其它位置（相关指导，请参见第 129 页的“主机和群集配置”）。

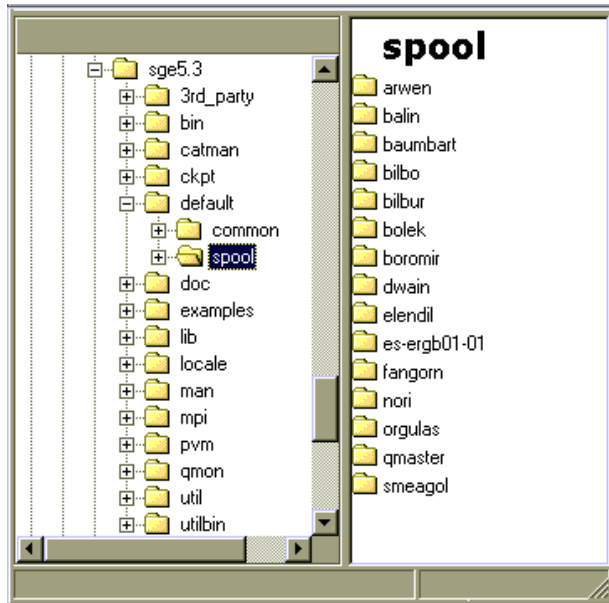


图 2-1 目录分层结构示例

磁盘空间需求

Sun Grid Engine 目录树有如下所示的一些固定的磁盘空间要求。

- 安装套件（包括文档资料，但不含二进制文件）需 40 MB 空间
- 每组二进制文件需要 10 到 15 MB 空间（体系结构 Cray 除外，其二进制文件大约需要占用 35 MB）

Sun Grid Engine 日志文件的理想磁盘空间如下。

- 主控主机假脱机目录需 30-200 MB 磁盘空间，这取决于群集的大小
- 每台执行主机需 10-20 MB 空间

注意 – 主控主机和执行主机的假脱机目录是可配置的，而且不必位于缺省位置 `<sgengine根目录>` 下。应该在初步安装完成后，再更改假脱机目录的位置（相关指导，请参见第六章第 129 页的“主机和群集配置”）。

安装帐户

安装 Sun Grid Engine 时既可以使用 root 帐户也可以使用非特权（如私用的）帐户。如果以非特权帐户安装，这种安装将只允许该用户运行 Sun Grid Engine 作业。其它所有帐户的访问将被拒绝。以 root 帐户安装可解除这种限制；但必须以 root 权限完成整个安装程序。

文件访问权限

若以 root 身份进行安装，则可能会在为共享文件系统上的所有主机配置根目录读 / 写访问权限时出现问题，致使在网络文件系统中放置 `<sgengine根目录>` 时遇到麻烦。可以强制 Sun Grid Engine 软件通过非 root 管理用户帐户（例如 `sgadmin`）执行 Sun Grid Engine 组件的全部文件操作。这样，您只需此用户的共享的 root 文件系统的读 / 写访问权限。Sun Grid Engine 安装程序会询问是否在管理用户帐户下处理文件。若回答 **yes**，并提供有效的 *用户名*，则将通过此用户名处理文件。否则将使用执行安装程序中所使用的用户名。

必须确保在所有情况下，用于文件处理的帐户在所有主机上都拥有读 / 写 Sun Grid Engine 根目录。同样，安装程序假设读取 Sun Grid Engine 发行媒体的主机也有权访问此目录。

网络服务

决定是将网络服务定义为 NIS 文件，还是将其放在每个工作站的本地 /etc/services 中。如果您的站点使用 NIS，请找到 NIS 服务器主机，以便在服务 NIS 映射中添加项。

Sun Grid Engine 服务是 sge_commd。要将服务添加到 NIS 映射，应选择一个预留的尚未使用的端口号（小于 1024）。以下是 sge_commd 项的一个示例。

```
sge_commd 536/tcp
```

主控主机

这是控制 Sun Grid Engine 的主机。它运行主控守护程序 sge_qmaster。对于那些包含数百乃至数千台主机而且系统内总是有数万个作业的大型群集，也许需要 1 GB 或更多的可用主内存，并且最好有两个 CPU。

- 必须是一个稳定的平台。
- 不得过多地忙于处理其它事情。
- 必须至少有 20 MB 尚未使用的主内存，以运行 Sun Grid Engine 守护程序。
- 可选地，它应有位于本地的 Sun Grid Engine 目录 <sge 根目录>，这样可以减少网络流量。

影像主控主机

在主控主机或主控守护程序出现故障时，这些主机可行使 sge_qmaster 的功能。作为影像主控主机的机器必须具有以下特征。

- 必须运行 sge_shadowd。
- 必须共享那些记录在磁盘中的有关 sge_qmaster 状态、作业和队列配置的信息。尤其是，影像主控主机必须拥有读/写根目录的权限，或拥有可以访问 sge_qmaster 假脱机目录和 <sge 根目录>/<单元>/common 目录的管理用户权限。
- <sge 根目录>/<单元>/common/shadow_masters 文件内必须有将主机定义为影像主控主机的语句。

一旦满足这些条件，即激活影像主控主机功能。因此，要使主机变为影像主机，并不需要重启 Sun Grid Engine 守护程序。

执行主机

这类主机运行提交至 Sun Grid Engine 的作业。您将在每台执行主机上运行安装脚本。

管理主机

Sun Grid Engine 操作人员和管理人员在这类主机上执行管理任务，如重新配置队列或添加 Sun Grid Engine 用户。主控主机安装脚本自动将主控主机设置为管理主机。

提交主机

可以从提交主机提交并控制 Sun Grid Engine 作业。主控主机安装脚本自动将主控主机设置为提交主机。

单元

可以将 Sun Grid Engine 设置为单个群集或松散联合的群集（称作单元）的集合。SGE_CELL 环境变量指明正在引用的群集。若 Sun Grid Engine 是作为单个群集安装的，将不设置 SGE_CELL，并假定单元的值为 default。

用户名

为了使 Sun Grid Engine 能够验证提交作业的用户是否有权提交，以及是否有权使用他们所需的执行主机，在涉及到的提交主机和执行主机上该用户的用户名必须相同。为满足此要求可能需要更改某些机器的用户 ID。

注意 – 主控主机的用户 ID 与检验权限无关，不必相互匹配，甚至不必存在。

队列

按照站点的需求规划队列的结构。也就是说，确定什么样的队列应该放在哪些执行主机上，是否需要有益于依次式的、交互式的、并行的以及其它类型作业的队列，每个队列需要多少作业位置，以及其它配置决策。

管理员也可以让 Sun Grid Engine 安装程序创建缺省的队列结构，这种结构适合初学的使用者，并可作为日后调整的基础。

注意 – 除了 Sun Grid Engine 的安装目录， Sun Grid Engine 安装程序创建的大多数设置都可在系统的操作过程中随时更改。

如果您对 Sun Grid Engine 已经很熟悉，或者已经确定了要对群集实施的队列结构，则不应让安装程序为您安装缺省队列结构。但是，您应准备一份详细说明队列结构的文档，且安装一完毕，就参照第七章第 149 页的“配置队列和队列日历”继续执行。

▼ 如何规划安装

1. 开始安装前，在与下表相似的表内填写安装规划。

| 参数 | 值 |
|----------------|---|
| <sg_e 根目录> | |
| 管理用户 | |
| 管理组 | |
| sg_e_commd 端口号 | |
| 主控主机 | |
| 影像主控主机 | |
| 执行主机 | |
| 管理主机 | |
| 提交主机 | |

2. 确保通过上述定义的访问权限可正确设置含有 Sun Grid Engine 发行文件、假脱机文件和配置文件的文件系统和目录。

▼ 如何读取发行媒体

Sun Grid Engine 软件以存档文件的形式发行，可通过 Internet 下载。Web 发行文件还以 tar 文件格式提供，它最终用 compress（扩展名为 .Z）或用 gzip（扩展名为 .gz）来压缩。在继续执行以下步骤之前，您必须解压缩文件（使用 uncompress 或 gunzip）。

1. 提供访问发行媒体的权限，并登录系统。最好是直接连至文件服务器的系统。
2. 按照第 21 页的“安装目录 <sge 根目录>”中的描述创建安装目录，以读取 Sun Grid Engine 安装套件，确保正确设置了安装目录的访问权限。
3. 从命令提示符执行以下命令。

```
% cd 安装目录
% tar -xvpf 源发行文件
```

其中，*安装目录* 是安装目录的路径名，*源发行文件* 是硬盘上已解压缩的磁带存档文件名。执行以上两命令即可读入 Sun Grid Engine 安装套件。

▼ 如何安装主控主机

1. 以 root 身份登录主控主机。
2. 取决于是否能从主控主机上看到安装套件所处的目录，执行以下操作之一。
 - a. 如果 *能* 从主控主机上看到安装套件所处的目录，将目录更换到 (cd) 安装目录，然后执行步骤 3。
 - b. 如果 *不能* 看到该目录，则执行以下操作。
 - i. 在主机上创建一个本地的安装目录。
 - ii. 通过网络（如，使用 ftp 或 rcp）将安装套件复制到本地安装目录下。
 - iii. 将目录更换到 (cd) 本地安装目录。
3. 执行以下指令。

注意 – 若通过证书安全协议的方法进行安装，则必须在以下命令中加上 `-csp` 标志（请参见第 30 页的“如何安装和设置基于 CSP 的加密系统”）。

```
% ./install_qmaster
```

这将启动主控主机安装程序。您需要回答几个问题，而且可能会被要求执行一些管理操作。问题和操作项都是自解释的。

注意 – 为方便起见，可激活另一个终端会话来执行管理任务。

主控主机安装程序会创建 `sgc_qmaster` 和 `sgc_schedd` 所需的适当目录分层结构。安装程序将在主控主机上启动 Sun Grid Engine 组件 `sgc_commd`、`sgc_qmaster` 和 `sgc_schedd`。主控主机同时也被注册为具有管理和提交权限的主机。

如果您觉得有不正确的地方，可以随时中止并重新执行安装程序。

▼ 如何安装执行主机

1. 以 `root` 身份登录执行主机。
2. 同主控主机的安装一样，将安装套件复制到本地安装目录或使用网络安装目录。
3. 将目录更换到 (`cd`) 安装目录，并执行以下命令。

注意 – 若通过证书安全协议的方法进行安装，则必须在以下命令中加上 `-csp` 标志（请参见第 30 页的“如何安装和设置基于 CSP 的加密系统”）。

```
% ./install_execd
```

这将启动执行主机的安装程序。执行主机安装程序与主控主机安装程序的操作和处理很相似。

4. 回答安装脚本的提问。

注意 – 您也可以使用主控主机执行作业。只需在主控主机上进行执行主机的安装。

注意 – 如果作为主控主机的这台机器速度很低，或者群集极其庞大，则应让主控主机仅仅执行主控任务。

执行主机安装程序创建 `sge_execd` 所需的适当目录分层结构。安装程序将在执行主机上启动 Sun Grid Engine 组件 `sge_commd` 和 `sge_execd`。

▼ 如何安装管理和提交主机

主控主机默认为可以执行管理任务，并可以用于提交、监视和删除作业。它不需要进行其它安装以成为管理或提交主机。与此相反，*纯粹*的管理主机和执行主机却一定需要注册。

- 在管理主机（如主控主机）上通过管理帐户（如超级用户帐户）的身份，输入以下命令。

```
% qconf -ah 管理主机名 [...]  
% qconf -as 提交主机名 [...]
```

有关配置不同类型主机的更多信息以及其它方法，请参见第 131 页的“关于守护程序和主机”。

高安全性的安装

遵守以下指导可使系统更加安全。这些指导将帮助您建立基于*证书保密协议 (CSP)* 加密的系统。

Sun Grid Engine 5.3 产品和 Sun Grid Engine 5.3（企业版）产品都可以利用这种安全设置，这些指导对这两种产品都适用。为简便起见，这些指导中仅提及 Sun Grid Engine 产品。

在这种更加安全的系统中，不再用纯文本方式传送信息，而是用密钥对信息进行加密。密钥是通过公用 / 私用密钥协议进行交换的。用户在 Sun Grid Engine 系统中出示可以证明他或她身份的证书，并从 Sun Grid Engine 系统接收确定其通讯对象是否正确的证书。完成初始的通告阶段后，通讯将以加密的形式继续透明地进行。会话仅在某个时期内有效，过期后必须重新认证才能继续会话。

所需的附加设置

设置证书安全协议增强版的 Sun Grid Engine 系统的所需步骤与标准版很相似。大体上按第 26 页的“如何规划安装”、第 27 页的“如何读取发行媒体”、第 27 页的“如何安装主控主机”、第 28 页的“如何安装执行主机”和第 29 页的“如何安装管理和提交主机”中所述的指导操作即可。

但是，必须执行以下附加操作。

- 在主控主机上生成证书授权 (CA) 系统密钥和证书
调用带 `-csp` 标志的安装脚本即可完成此任务。
- 向执行和提交主机分发系统密钥和证书
这项工作须由系统管理员以安全的方式完成；也就是说，密钥必须以安全的方式（如通过 `ssh`）传递给执行主机和提交主机。
- 生成用户密钥和证书
此任务可在完成主控主机的安装后，由系统管理员自动完成。
- 系统管理员准许新用户的进入

▼ 如何安装和设置基于 CSP 的加密系统

1. 按照第 19 页的“完全安装概述”、第 21 页的“规划安装”、第 27 页的“如何安装主控主机”、第 28 页的“如何安装执行主机”和第 29 页的“如何安装管理和提交主机”中的描述安装 Sun Grid Engine 系统，除了这一点：在调用各个安装脚本时，需使用附加标志 `-csp`。

例如，若主控主机的基本安装指导告诉您输入 `./install_qmaster` 以调用脚本，则您须更改此指令，加上 `-csp` 标志。因此，要安装基于 CSP 的加密系统，必须改变主控主机的安装程序，输入：

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

2. 回答安装脚本的提问。

必须提供以下信息才能生成 CSP 证书和密钥。

- 两个字母的国家代码。例如，US 代表美国
- 州 / 省
- 位置，如城市
- 组织
- 部门
- CA 电子邮件地址

随着安装的进行，证书授权即被创建。在主控主机上创建了特定于 Sun Grid Engine 的 CA。包含相关安全信息的目录如下。

- 在目录 `$SGE_ROOT/{缺省值|$SGE_CELL}/common/sgeCA` 下，存储的是公用 CA 和守护程序证书。
- 在目录 `/var/sgeCA/{sge 服务|端口 $COMM_PORT}/{缺省值|$SGE_CELL}/private` 下，存储的是相应的私用密钥。
- 在目录 `/var/sgeCA/{sge 服务|端口 $COMM_PORT}/{缺省值|$SGE_CELL}/userkeys/$USER` 下，存储的是用户密钥和证书。

在此进程中，脚本的输出与代码示例 2-1 的显示很相似。

代码示例 2-1 CSP 安装脚本 — 目录的创建

```
正在为 OpenSSL 安全性框架结构初始化证书授权 (CA)
```

```
-----  
正在创建 /scratch2/eddy/sge_sec/default/common/sgeCA  
正在创建 /var/sgeCA/port6789/default  
正在创建 /scratch2/eddy/sge_sec/default/common/sgeCA/certs  
正在创建 /scratch2/eddy/sge_sec/default/common/sgeCA/crl  
正在创建 /scratch2/eddy/sge_sec/default/common/sgeCA/newcerts  
正在创建 /scratch2/eddy/sge_sec/default/common/sgeCA/serial  
正在创建 /scratch2/eddy/sge_sec/default/common/sgeCA/index.txt  
正在创建 /var/sgeCA/port6789/default/userkeys  
正在创建 /var/sgeCA/port6789/default/private  
单击回车键以继续 >>
```

建立目录后，特定于 CA 的证书和私用密钥便生成了。Sun Grid Engine 使用专门文件的伪随机数据，或者使用 `/dev/random`（如果可用）产生伪随机数生成器 (PRNG)。（有关随机数的详细信息，请参见 <http://www.openssl.org/support/faq.html> 和 <http://www.cosy.sbg.ac.at/~andi>。）

安装 CA 基础架构后，CA 就为管理用户、伪守护程序用户和 root 用户创建和签署应用程序证书、用户证书和私用密钥。脚本（其输出信息与代码示例 2-2 中的示例相似）首先询问站点信息。

代码示例 2-2 CSP 安装脚本 — 信息的收集

```
正在创建 CA 证书和私用密钥
-----

请指定一些基本参数以便创建证书的
识别名称 (DN)。

将询问

    - 两个字母的国家代码
    - 州
    - 位置，例如城市或建筑代码
    - 机构（例如，您的公司名称）
    - 机构单位，例如，您的部门
    - CA 管理员（您！）的电子邮件地址

单击回车键以继续 >>

请输入两个字母的国家代码，例如 >US< >> DE
请输入州 >> Bavaria
请输入您的位置，例如城市或建筑代码 >> Regensburg
请输入机构名称 >> Myorg
请输入机构单位，例如您的部门 >> Mydept
请输入 CA 管理员的电子邮件地址 >> admin@my.org

您已选择了以下基本数据用于证书的
识别名称:

国家代码:          C=DE
州:                ST=Bavaria
位置:              L=Regensburg
机构:              O=Myorg
机构单位:          OU=Mydept
CA 电子邮件地址:  emailAddress=admin@my.org

是否要使用这些数据 (y/n) [y] >>
```

确认提供的信息是正确的之后，安装程序继续 CA 证书和私用密钥的生成，首先设置 CA 基础架构。脚本输出与代码示例 2-3 中的示例相似。

代码示例 2-3 CSP 安装脚本 — CA 基础架构的创建

```
正在从 >/var/sgeCA/port6789/default/private/rand.seed< 中的 >/kernel/genunix< 创建 RANDFILE

1513428 semi-random bytes loaded
正在创建 CA 证书和私用密钥

Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '/var/sgeCA/port6789/default/private/akey.pem'
-----
单击回车键以继续 >>
```

安装 CA 基础架构后，CA 为伪守护程序用户和 root 用户创建并签署应用程序和用户证书以及私用密钥。脚本输出与示例中所示（接下页）相似。请注意本例中的某些行有省略，以使之能够在本页的一个单行内显示。省略部分用 (...) 表示。

代码示例 2-4 CSP 安装脚本 — 证书和私用密钥的创建

```
正在创建守护程序证书和密钥
-----

正在从 >/var/sgeCA/(...)/rand.seed< 的 >/kernel/genunix< 创建 RANDFILE

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
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName         :PRINTABLE:'Regensburg'
```

代码示例 2-4 CSP 安装脚本 — 证书和私用密钥的创建 (接上页)

```
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
正在从 >/var/(...)/userkeys/root/rand.seed< 中的 >/kernel/genunix< 创建 RANDFILE

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<
正在从 >/(...)/userkeys/eddy/rand.seed< 中的 >/kernel/genunix< 创建 RANDFILE
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
```

代码示例 2-4 CSP 安装脚本 — 证书和私用密钥的创建 (接上页)

```
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<
正在从 >/var/(...)/userkeys/eddy/rand.seed< 中的 >/kernel/genunix< 创建 RANDFILE

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:'root'
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<
正在从 >/var/(...)/userkeys/eddy/rand.seed< 中的 >/kernel/genunix< 创建 RANDFILE

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
```

代码示例 2-4 CSP 安装脚本 — 证书和私用密钥的创建 (接上页)

```
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:'root'
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<
单击回车键以继续 >>
```

主控主机 (sge_qmaster) 的相关安全设置完成后, 脚本将提示您继续执行安装的其余步骤, 与代码示例 2-5 的示例类似。

代码示例 2-5 CSP 安装脚本 — 继续安装

```
SGE startup script
-----

Your system wide SGE startup script is installed as:

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

单击回车键以继续 >>
```

3. 执行以下操作之一。

- a. 若您认为共享的文件系统不够安全, 以至于不宜将 CSP 安全信息放在执行守护程序可以访问的地方, 请执行步骤 4。

- b. 若您认为共享的文件系统是足够安全的，则继续执行第 28 页的“如何安装执行主机”一节中的基本安装步骤。

安装执行主机时，不要忘记在调用 `./install_execd` 脚本时加上“-csp”标志。

完成其余的所有步骤后，依照第 39 页的“如何为用户生成证书和私用密钥”中的指导继续进行。

4. (可选) 若共享的文件系统不够安全，以至于不宜将 CSP 安全信息放在执行守护程序可以访问的地方，则必须将包含守护程序的私用密钥的目录和随机文件转移到执行主机上。

- a. 以主控主机的 `root` 用户身份输入以下命令，准备将私用密钥复制到将设置为执行主机的机器上。

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

- b. 在每台执行主机上以 `root` 用户身份输入以下命令以复制文件。

```
# umask 077
# cd /
# scp <主控主机>:/var/sgeCA/port6789.tar .
# umask 022
# tar xvpf /port6789.tar
# rm /port6789.tar
```

- c. 输入以下命令可验证文件权限。

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

输出应与代码示例 2-6 中的示例相似。

代码示例 2-6 文件权限验证

```
/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. 输入以下命令继续安装 Sun Grid Engine。

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

e. 按照第 28 页的“如何安装执行主机”一节中自步骤 3 开始的其余安装指导，请切记添加 `-csp` 标志以启动安装脚本。

完成所有剩余的安装步骤后，继续按照第 39 页的“如何为用户生成证书和私用密钥”一节中的指导执行。

▼ 如何为用户生成证书和私用密钥

为了能够使用 CSP 安全系统，用户必须有权访问特定于用户的证书和私用密钥。最简便的方法就是创建一个标识用户身份的文本文件。

1. 创建并保存标识用户身份的文本文件。

使用下例所示文件 `myusers.txt` 的格式。（文件的字段是 *UNIX 用户名:Gecos 字段: 电子邮件地址*。）

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

2. 以主控主机的 root 用户身份输入以下命令。

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

3. 输入以下命令进行确认。

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

目录列表的输出应与下例相似。

```
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. 通知文件（如本例中的 `myusers.txt`）中列出的用户将与安全相关的文件安装到其 `$HOME/.sge` 目录。方法是输入以下命令。

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

用户应该看到以下确认信息（本例中用户名为 `eddy`）。

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

每次安装 Sun Grid Engine 时，都会安装一个与 COMMD_PORT 端口号对应的子目录。下例基于 myusers.txt 文件，是执行位于所示输出之前的命令而产生的结果。

```
% 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
```

▼ 如何检查证书

- 根据需要，输入以下一条或多条命令。

显示证书

将以下命令作为一个字符串输入（该命令太长，无法在本指南的一行中完全显示），`-in` 和 `~/.sge` 两部分之间有一个空格。

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

查看颁发人

将以下命令作为一个字符串输入（该命令太长，无法在本指南的一行中完全显示），`-in` 和 `~/sge` 两部分之间有一个空格。

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

查看主题

将以下命令作为一个字符串输入（该命令太长，无法在本指南的一行中完全显示），`-in` 和 `~/sge` 两部分之间有一个空格。

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

显示证书的电子邮件

将以下命令作为一个字符串输入（该命令太长，无法在本指南的一行中完全显示），`-in` 和 `~/sge` 两部分之间有一个空格。

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

显示有效期

将以下命令作为一个字符串输入（该命令太长，无法在本指南的一行中完全显示），`-in` 和 `~/sge` 两部分之间有一个空格。

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

显示指纹

将以下命令作为一个字符串输入（该命令太长，无法在本指南的一行中完全显示），`-in` 和 `~/sge` 两部分之间有一个空格。

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

▼ 如何验证安装

在主控主机上

1. 登录到主控主机。
2. 根据运行的操作系统，执行以下命令之一。
 - a. 在基于 BSD 的 UNIX 系统上，输入以下命令。

```
% ps -ax
```

- b. 在运行基于 UNIX System 5 的操作系统（如 Solaris 操作环境）下，输入以下命令。

```
% ps -ef
```

3. 浏览输出，找到类似于下例的 `sge` 的字符串。

在基于 BSD 的 UNIX 系统上，应该看到类似于下例的输出。

```
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
```

若是基于 UNIX System 5 的系统，则应该看到类似于下例的输出。

```
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
```

若没有看到相应的字符串，则表明这台机器上未运行主控主机所需的一个或多个 Sun Grid Engine 守护程序（可以查看 < sge 根目录 > / < 单元 > / common / act_qmaster 文件，以确定登录的是否是主控主机）。继续下一步。

4. (可选) 手动地重新启动守护程序。

有关继续操作的指导，请参见第 131 页的“关于守护程序和主机”一节。

在执行主机上

1. 登录运行 Sun Grid Engine 执行主机安装程序的执行主机。
2. 参考主控主机安装程序中的步骤 2，以确定适合系统的 ps 命令，然后输入该命令。
3. 在输出中找到 sge 字符串。

在基于 BSD 的 UNIX 系统上，应该看到类似于下例的输出。

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

若是基于 UNIX System 5 的系统（如 Solaris 操作环境），则应该看到类似于下例的输出。

```
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
```

若没有看到类似的输出，则表明执行主机上所需的一个或多个守护程序没有运行。继续下一步。

4. (可选) 手动地重新启动守护程序。

有关继续操作的指导，请参见第 131 页的“关于守护程序和主机”一节。

试用命令

若主控主机和执行主机上均已在运行必要的守护程序，则 Sun Grid Engine 系统应该可以运作。发出试用命令进行检测。

1. 登录到主控主机或另一台管理主机。

务必将安装 Sun Grid Engine 二进制文件的路径包含在标准搜索路径内。

2. 在命令行上输入以下命令。

```
% qconf -sconf
```

此 `qconf` 命令显示当前的全局群集配置（请参见第 144 页的“基本群集配置”一节）。若此命令失败，则很可能是由于 `SGE_ROOT` 环境变量设置不正确，或 `qconf` 联络与 `sge_qmaster` 相关的 `sge_commd` 时失败。继续下一步。

3. 检查脚本文件 `<sge 根目录>/<单元>/common/settings.csh` 或 `<sge 根目录>/<单元>/common/settings.sh` 是否设置了环境变量 `COMMD_PORT`。

如果已设置，请在重试以上命令前，确保环境变量 `COMMD_PORT` 已设置为合适的值。如果设置文件中未使用 `COMMD_PORT` 变量，那么执行命令的主机上的服务数据库（如 `/etc/services` 或 NIS 服务映射）必须提供一个 `sge_commd` 项。若情况并非如此，则请在该机器的服务数据库中添加这一项，并使其值与 Sun Grid Engine 主控主机上的配置值相同。然后执行下一步。

4. 重试 `qconf` 命令。

准备提交作业

在向 Sun Grid Engine 系统提交批处理脚本前，检查站点的标准和私用 shell 资源文件（`.cshrc`、`.profile` 或 `.kshrc`）是否包含诸如 `stty` 的命令（缺省情况下，批处理作业并不与终端相连，因此调用 `stty` 会导致错误）。

1. 登录到主控主机。
2. 请输入以下命令。

```
% rsh 某执行主机 date
```

某执行主机 指的是您计划使用的、已安装的一台执行主机（若您的登录名或主目录在主机与主机之间不相同，则必须在所有执行主机上进行检查）。`rsh` 命令的输出应与在主控主机本地执行 `date` 命令的输出相似。如果另外还有一些包含错误消息的行，则必须先消除错误起因，之后才能成功运行批处理作业。

对于所有命令解释器，您可以在执行命令（如 `stty`）之前，在实际的终端连接上检查。下面以 Bourne-/Korn-Shell 为例解释如何检查：

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

C-Shell 的语法也很相似：

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

3. 提交位于 `<sgc 根目录>/examples/jobs` 目录中的某个示例脚本。

请输入以下命令。

```
% qsub 脚本路径
```

4. 使用 Sun Grid Engine `qstat` 命令监视作业的操作。

关于提交和监视批处理作业的更多信息，请参见第 69 页的“提交批处理作业”。

5. 作业的执行完成后，在您的主目录中检查重定向的 `stdout/stderr` 文件 `<脚本名>.e <作业 ID>` 和 `<脚本名>.o<作业 ID>`，其中，`<作业 ID>` 是分配给每个作业的唯一连续整数。

若出现问题，请参见第十一章“错误诊断”。

第三部分 使用 Sun Grid Engine 5.3 软件

《*Sun Grid Engine 5.3 管理和用户指南*》中的这个部分包括三章，主要针对用户，即不履行系统管理员（请参见第四部分，第 127 页的“管理”）职责的人员。

- **第三章 – 第 49 页的“浏览 Sun Grid Engine 5.3 程序”**

本章介绍有关 Sun Grid Engine 5.3 的一些基础知识，并指导您如何列出各种资源。

- **第四章 – 第 63 页的“提交作业”**

本章提供运用 Sun Grid Engine 5.3 系统提交作业的详尽指导，从“练习”作业的提交开始让您熟悉整个过程。

- **第五章 – 第 101 页的“点检查、监视和控制作业”**

本章解释作业控制的概念，并指导您完成各种作业控制任务。

第三部分的每一章都涵盖有关利用 Sun Grid Engine 5.3 系统完成各种任务的背景信息和详细指导。

浏览 Sun Grid Engine 5.3 程序

本章介绍有关 Sun Grid Engine 5.3 的一些基本概念和术语，有助于您初学该软件的使用。有关本产品的详尽的背景信息，包括完整的词汇表，请参见第一章第 1 页的“Sun Grid Engine 5.3 简介”。

本章还包含完成以下任务的指导。

- 第 51 页的 “如何启动 QMON 浏览器”
- 第 52 页的 “如何显示队列的列表”
- 第 52 页的 “如何显示队列特性”
- 第 55 页的 “如何找到主控主机的名称”
- 第 55 页的 “如何显示执行主机列表”
- 第 55 页的 “如何显示管理主机列表”
- 第 56 页的 “如何显示提交主机列表”
- 第 57 页的 “如何显示可请求属性列表”

Sun Grid Engine 的用户类型和操作

Sun Grid Engine 将用户分成四类。

- **管理人员** – 管理人员能够全面操控 Sun Grid Engine。缺省情况下，任何作为队列主机的计算机的超级用户均有管理人员特权。
- **操作人员** – 操作人员能够执行管理人员所执行的大多数命令，但操作人员不能修改配置（例如添加、删除或修改队列）。
- **拥有者** – 队列拥有者可以暂停或启用自身所拥有的队列或作业，但没有其它管理权限。
- **用户** – 如第 60 页的“用户访问权限”所述，用户有一定的访问权限，但不能管理群集或队列。

表 3-1 列出了不同类型的用户可使用的 Sun Grid Engine 5.3 命令。

表 3-1 用户种类及其可执行的相关命令

| 命令 | 管理人员 | 操作人员 | 拥有者 | 用户 |
|---------|------|--------------|---------------------|-------------|
| qacct | 全部 | 全部 | 仅适于自己的作业 | 仅适于自己的作业 |
| qalter | 全部 | 全部 | 仅适于自己的作业 | 仅适于自己的作业 |
| qconf | 全部 | 无系统设置 修改权 | 仅可显示配置和访问 权限 | 仅可显示配置和访问权限 |
| qdel | 全部 | 全部 | 仅适于自己的作业 | 仅适于自己的作业 |
| qhold | 全部 | 全部 | 仅适于自己的作业 | 仅适于自己的作业 |
| qhost | 全部 | 全部 | 全部 | 全部 |
| qlogin | 全部 | 全部 | 全部 | 全部 |
| qmod | 全部 | 全部 | 仅适于自己的作业和 所拥有的队列 | 仅适于自己的作业 |
| qmon | 全部 | 无系统设置 修改权 | 无配置 更改权 | 无配置更改权 |
| qrexec | 全部 | 全部 | 全部 | 全部 |
| qselect | 全部 | 全部 | 全部 | 全部 |
| qsh | 全部 | 全部 | 全部 | 全部 |
| qstat | 全部 | 全部 | 全部 | 全部 |
| qsub | 全部 | 全部 | 全部 | 全部 |

队列和队列特性

为了能在您的站点上充分利用 Sun Grid Engine 系统，最好先熟悉为 Sun Grid Engine 系统配置的队列结构和队列特性。

QMON 浏览器

Sun Grid Engine 提供了图形用户界面 (GUI) 命令工具 QMON 浏览器。QMON 浏览器提供许多 Sun Grid Engine 功能，包括作业提交、作业控制和重要信息的收集。

▼ 如何启动 QMON 浏览器

- 在命令行输入以下命令。

```
% qmon
```

显示一个消息窗口后，随之跳出一个如下所示的 QMON 主控制面板（有关各图标的含义，请参见图 1-2）。



图 3-1 QMON 主控制面板

本书中的许多指导都要求使用 QMON 浏览器。当把鼠标置于图标按钮上时即可显示出此图标的名称，也就是其功能的描述。

（有关如何自定义 QMON 浏览器的信息，请参见第 8 页的“自定义 QMON”。）

队列控制 QMON 对话框

第 120 页的“如何用 QMON 控制队列”中显示并描述了“QMON 队列控制”对话框，此对话框概述了已安装队列及其当前状态。

▼ 如何显示队列的列表

- 请输入以下命令。

```
% qconf -sql
```

▼ 如何显示队列特性

可以使用 QMON 或命令行显示队列特性。

使用 QMON 浏览器

1. 从 QMON 主菜单，单击“浏览器”按钮。
2. 单击“队列”按钮。

3. 在“队列控制”对话框，将鼠标移至适当队列的图标上。

图 3-2 举例显示了部分队列特性的信息。

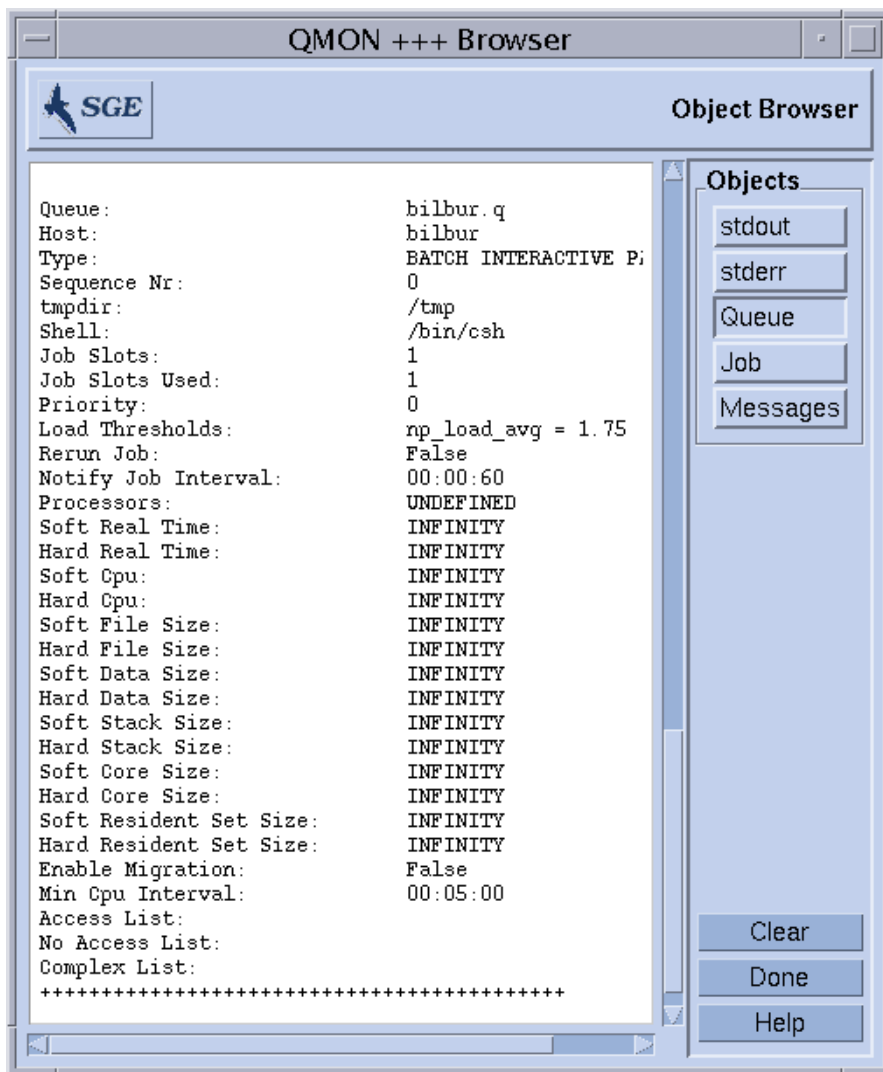


图 3-2 队列特性的 QMON 浏览器显示

从命令行

- 请输入以下命令。

```
% qconf -sq 队列名
```

显示的信息与图 3-2 所示信息相似。

解释队列特性信息

可以在 `queue_conf` 手册页和 《*Sun Grid Engine 5.3 和 Sun Grid Engine, 5.3 (企业版) 参考手册*》的 `queue_conf` 一节中找到每个队列特性的详细描述。

下面列出了一些最重要的参数。

- `qname` – 所请求的队列名称。
- `hostname` – 队列所处的主机。
- `processors` – 多处理器系统中此队列有权访问的处理器。
- `qtype` – 允许在此队列中运行的作业类型。目前包括批处理作业、交互式作业、点检查作业、并行作业或它们的任意组合或相互转换的作业。
- `slots` – 可在队列上同时执行的作业数量。
- `owner_list` – 队列的拥有者，请参见第 61 页的“管理人员、操作人员和拥有者”的解释。
- `user_lists` – 此参数下所列的用户访问列表（请参见第 60 页的“用户访问权限”）中标识的用户或组有权访问此队列。
- `xuser_lists` – 此参数下所列的用户访问列表（请参见第 60 页的“用户访问权限”）中标识的用户或组无权访问此队列。
- `complex_list` – 列于此参数下的属性组与队列相关联，而且这些属性组中所含的属性构成此队列的可请求属性组（请参见第 56 页的“可请求的属性”）。
- `complex_values` – 提供给此队列的某些属性组属性的赋值（请参见第 56 页的“可请求的属性”）。

主机功能

单击 QMON 主菜单中的“主机配置”按钮，可显示与 Sun Grid Engine 群集中的主机相关的功能的概述。然而，若没有 Sun Grid Engine 管理人员特权，则无法对现有配置进行任何更改。

第 131 页的“关于守护程序和主机”一节对主机配置对话框进行了描述。以下各节提供从命令行检索此类信息的命令。

▼ 如何找到主控主机的名称

由于主控主机可能会随时可能在当前的主控主机和某个影像主控主机之间切换，主控主机的位置对用户来说应该是透明的。

- 使用文本编辑器，打开 `<sgc 根目录>/<单元>/common/act_qmaster` 文件。此文件中有当前主控主机的名称。

▼ 如何显示执行主机列表

要显示群集中配置为执行主机的主机列表，请使用命令：

```
% qconf -sel  
% qconf -se 主机名称  
% qhost
```

第一条命令显示当前配置为执行主机的所有主机的列表。第二条命令显示指定的执行主机的详细信息。第三条命令显示执行主机的状态和负荷信息。关于通过 `qconf` 显示的信息的详细情况，请参见 `host_conf` 手册页；关于其输出和更多的选项，请参见 `qhost` 手册页。

▼ 如何显示管理主机列表

可用以下命令显示有管理权限的主机列表：

```
% qconf -sh
```

▼ 如何显示提交主机列表

可用以下命令显示提交主机列表。

```
% qconf -ss
```

可请求的属性

提交一个 Sun Grid Engine 作业时，可指定该作业的需求概况。用户可以指定作业所需的主机或队列的属性或特性以保证作业成功运行。Sun Grid Engine 将这些作业需求映射到 Sun Grid Engine 群集的主机和队列的配置，从而找到适合该作业的主机。

可用于指定作业需求的属性可与 Sun Grid Engine 群集（如网络共享磁盘的空间）、主机（如操作系统的体系结构）或队列（如允许的 CPU 时间）相关，属性还可以从站点策略（如已安装的软件只能在某些主机上使用）派生而来。

可用的属性包括队列特性列表（请参见第 50 页的“队列和队列特性”）、全局属性和主机相关属性的列表（请参见第 169 页的“属性组类型”），以及管理员定义的属性。但是，为方便起见，Sun Grid Engine 管理员通常只将一个所有可用属性的子集定义为可请求。

当前可请求的属性列在“QMON 提交”对话框的“请求的资源”子对话框（请参见图 3-3）中。（有关如何提交作业的信息，请参见第 69 页的“提交批处理作业”一节。）它们列在“可用资源”选择列表内。

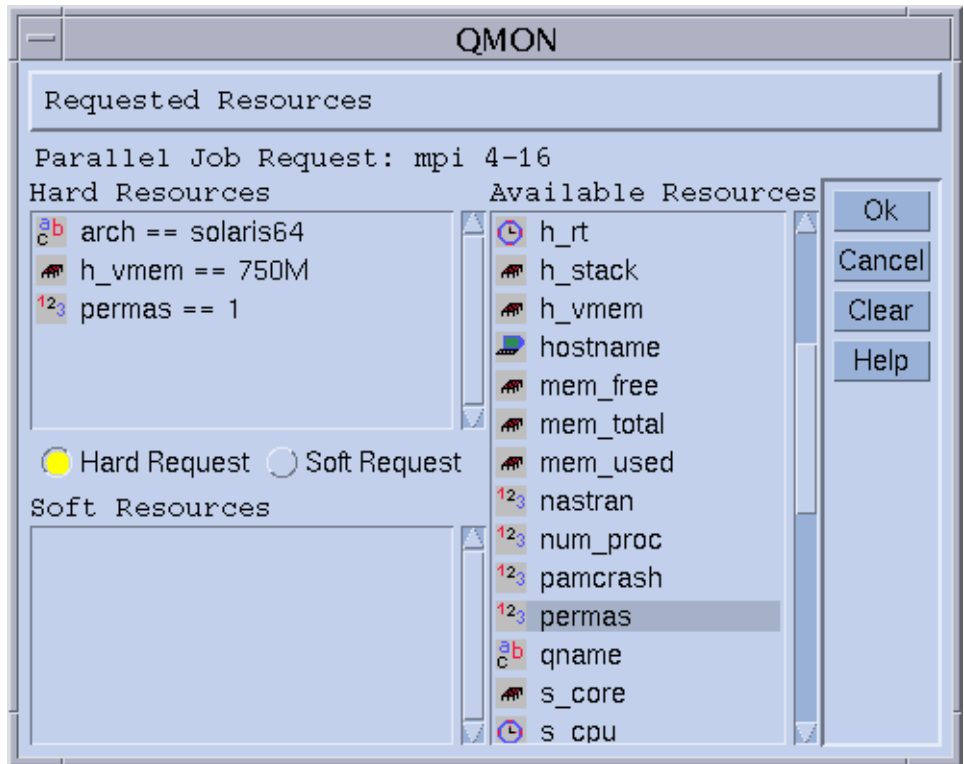


图 3-3 “请求的资源”对话框

▼ 如何显示可请求属性列表

1. 在命令行中输入以下命令，可显示已配置的属性组列表：

```
% qconf -scl
```

属性组包含一组属性的定义。有三种标准属性组：

- global – 针对群集全局属性（可选）
- host – 针对主机特有的属性
- queue – 针对队列特性的属性

如果输入以上命令后还显示出其它属性组名称，则那些属性组是管理员定义的（有关属性组的更多信息，请参见第八章第 167 页的“属性组概念”或《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中有关属性组格式的描述）。

2. 通过输入以下命令，显示某个特定属性组的属性：

```
% qconf -sc 属性组名称 [...]
```

例如，队列属性组的输出可能类似于表 3-2 中所示。

表 3-2 显示的 queue 属性组属性

| # 名称 | 缩写名 | 类型 | 值 | 关系 | 可否请求 | 可否使用 | 缺省值 |
|------------------|---------|--------|---------|----|------|------|-------|
| 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 |
| 优先级 | 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 |
| min_cpu_interval | mci | TIME | 0:0:0 | <= | NO | NO | 0:0:0 |

表 3-2 显示的 queue 属性组属性 (接上页)

| # 名称 | 缩写名 | 类型 | 值 | 关系 | 可否请求 | 可否使用 | 缺省值 |
|---------------|-----|--------|---------|----|------|------|-------|
| 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 |
| 优先级 | pr | INT | 0 | >= | NO | NO | 0 |
| seq_no | seq | INT | 0 | == | NO | NO | 0 |
| max_migr_time | mmt | TIME | 0:0:0 | <= | NO | NO | 0:0:0 |
| max_no_migr | mmn | TIME | 0:0:0 | <= | NO | NO | 0:0:0 |

名称一栏中的显示与 `qconf -sq` 命令显示的第一栏基本相同。队列属性涵盖大部分 Sun Grid Engine 队列特性。缩写名一栏包含可由管理员定义的第一栏中全名的缩写。用户可在 `qsub` 命令的请求选项中指定全名或缩写名。

可否请求一栏表明是否可将相应的项用于 `qsub`。比方说，这样管理员就可以仅仅通过将项 `qname` 和 / 或 `qhostname` 设置为不可请求，来阻止群集用户直接请求某些机器 / 队列为其作业服务。这样做意即大体上可以由多个队列来满足可行的用户请求，从而强制执行 Sun Grid Engine 的负荷平衡功能。

关系栏定义关系运算以用于计算队列是否满足用户请求。执行的比较是：

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

如果比较的结果为假，则用户作业将无法在当前考虑的队列中执行。例如，为队列 `q1` 配置了 100 秒的软性 CPU 时间限制（有关用户操作限制的信息，请参见 `queue_conf` 和 `setrlimit` 手册页），而为队列 `q2` 配置了 1000 秒的软性 CPU 时间限制。

可否使用栏和缺省值栏对管理员极其有用，管理员可以用它们声明“可使用资源”（请参见第 177 页的“可使用的资源”一节）。用户可以像请求其它属性一样请求可使用资源。但是，Sun Grid Engine 对资源的内部簿记并不相同。

假定用户提交以下请求。

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

`s_cpu=0:5:0` 请求（请参见 `qsub` 手册页以获得它的详细语法信息）一个授予 CPU 软性限制时间至少为 5 分钟的队列。因此，只有软性 CPU 运行时间限制至少为 5 分钟的队列才被设置为适合运行此作业。

注意 – 若有不止一个队列能够运行此作业，则 Sun Grid Engine 在调度过程中只考虑负荷信息。

用户访问权限

Sun Grid Engine 管理员可以限制某些用户或用户组访问队列以及其它 Sun Grid Engine 工具（比如并行环境接口。请参见第 93 页的“并行作业”）。

注意 – Sun Grid Engine 自动考虑由群集管理者配置的访问权限。如果您想查询个人的访问权限，以下各节很重要。

为了限制访问权限，管理员创建并维护“访问列表”（简称为 *ACL*）。ACL 包含任意用户和 UNIX 组名。然后，ACL 被添加到队列或并行环境接口配置中的 *允许访问* 或 *拒绝访问* 列表（请分别参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》第五节的 `queue_conf` 或 `sge_pe`）。

若用户属于允许访问列表中的 ACL 中的一员，则其有权访问队列或并行环境接口。若用户属于拒绝访问列表中的 ACL 的成员，则其无权访问相关资源。

经由 QMON 主菜单中的“用户配置”图标按钮打开的“用户组配置”对话框，可用来查询您有权访问的 ACL。有关详细信息，请参见第九章第 195 页的“管理用户访问权限和策略”。

在命令行输入以下命令可以获得当前配置的 ACL 列表：

```
% qconf -sul
```

输入以下命令可显示一个或多个访问列表中的项：

```
% qconf -su ACL 名称 [...]
```

ACL 包含用户帐户名和 UNIX 组名，UNIX 组名用前缀“@”标识出来。这样即可确定您的帐户所属的 ACL。

注意 – 若您有权使用 `newgrp` 命令切换主要 UNIX 组，则您的访问权限有可能会改变。

现在即可查询您有权或无权访问的队列或并行环境接口。请按照第 50 页的“队列和队列特性”和第 224 页的“如何用 QMON 配置 PE”中的描述查询队列或并行环境接口配置。允许访问列表的名称是 `user_lists`。拒绝访问列表的名称是 `xuser_lists`。若您的用户帐户或主要 UNIX 组与允许访问列表相关联，那么您有权访问相关资源。若您与拒绝访问列表相关联，则无权访问队列或并行环境接口。如果两个列表都为空，那么使用有效帐户的每一个用户均可访问相关资源。

管理人员、操作人员和拥有者

使用以下命令可获得 Sun Grid Engine 管理人员的列表：

```
% qconf -sm
```

要显示操作人员的列表：

```
% qconf -so
```

注意 – Sun Grid Engine 管理主机的超级用户默认为管理人员。

如第 50 页的“队列和队列特性”一节中所述，拥有某个队列的用户被包含在该队列配置数据库中。执行以下命令可以检索数据库：

```
% qconf -sq 队列名
```

相关的队列配置项称为 `owners`。

提交作业

本章提供了使用 Sun Grid Engine 5.3 提交作业以供处理的背景信息和指导。本章从运行一个简单作业的示例着手，然后提供运行更复杂的作业的指导。

本章中包含完成以下任务的指导。

- 第 63 页的 “如何从命令行运行简单作业”
- 第 65 页的 “如何从图形用户界面 QMON 提交作业”
- 第 84 页的 “如何从命令行提交作业”
- 第 87 页的 “如何从命令行提交阵列作业”
- 第 87 页的 “如何用 QMON 提交阵列作业”
- 第 89 页的 “如何用 QMON 提交交互式作业”
- 第 92 页的 “如何用 qsh 提交交互式作业”
- 第 92 页的 “如何用 qllogin 提交交互式作业”

运行简单作业

运用本节中的信息和指导，可熟悉提交 Sun Grid Engine 5.3 作业所涉及的基本步骤。

注意 – 如果您已经用一个非特权帐户安装了 Sun Grid Engine 程序，则必须以此用户登录，才能运行作业（详细信息请参见第 21 页的“先决任务”）。

▼ 如何从命令行运行简单作业

执行任何 Sun Grid Engine 命令之前，必须设置适当的可执行搜索路径和其它环境条件。

1. 根据您所使用的命令解释器，输入以下命令。

a. 如果正在用 `csch` 或 `tcsh` 作为命令解释器：

```
% source sgc 根目录 /default/common/settings.csh
```

sgc 根目录 是指在安装过程之初为 Sun Grid Engine 选择的根目录的位置。

b. 如果正在用 `sh`、`ksh` 或 `bash` 作为命令解释器：

```
# . sgc 根目录 /default/common/settings.sh
```

注意 – 可以将以上命令添加到 `.login`、`.cshrc` 或 `.profile` 文件中（选择合适的），以保证稍后将启动的交互式会话的 Sun Grid Engine 设置正确。

2. 将以下简单作业脚本提交给 Sun Grid Engine 群集。

您可以在 Sun Grid Engine 根目录下的 `examples/jobs/simple.sh` 文件中找到以下作业。

```
#!/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
```

输入以下命令，此处假定 `simple.sh` 为存储以上脚本的脚本文件名，且文件位于当前工作目录下。

```
% qsub simple.sh
```

`qsub` 命令应确认作业已成功提交，如下所示。

```
您的作业 1 ("simple.sh") 已提交
```

3. 输入以下命令检索作业的状态信息。

```
% qstat
```

您应收到一份状态报告，其中包括 Sun Grid Engine 系统当前已知的所有作业的信息、每个作业的所谓 *作业 ID*（提交确认信息中包含的唯一编号）、作业脚本名、作业所有者、状态信息（*r* 表示正在运行）、提交或启动时间以及作业在其中执行的队列名。

若 `qstat` 命令没有产生输出，则系统实际上无已知作业。例如，您的作业可能已经完成。可以检查已完成作业的 `stdout` 和 `stderr` 重定向文件，以控制它们的输出。缺省情况下，生成的这些文件存放在执行此作业的主机上的作业所有者主目录中。文件名由作业脚本文件名、句点加“o”（对于 `stdout` 文件）或“e”（对于 `stderr` 文件），以及唯一的作业 ID 组成。因此，如果作业是在一新安装的 Sun Grid Engine（企业版）系统上首次执行，您的作业的 `stdout` 和 `stderr` 文件名分别为 `simple.sh.o1` 和 `simple.sh.e1`。

▼ 如何从图形用户界面 QMON 提交作业

提交和控制 Sun Grid Engine 作业以及获得 Sun Grid Engine 系统概述的更方便的方法是图形用户界面 QMON。QMON 为提交和监视作业的任务提供了一个作业提交菜单和一个“作业控制”对话框，以及其它工具。

在命令行提示符下，键入以下命令。

```
% qmon
```

启动过程中，将出现消息窗口，然后出现 QMON 主菜单。

4. 单击“作业控制”按钮，然后单击“提交作业”按钮。



图 4-1 QMON 主菜单

“作业提交”和“作业控制”对话框出现（分别参见图 4-2 和图 4-3）。鼠标移至按钮上时，会显示出按钮名称（如“作业控制”）。

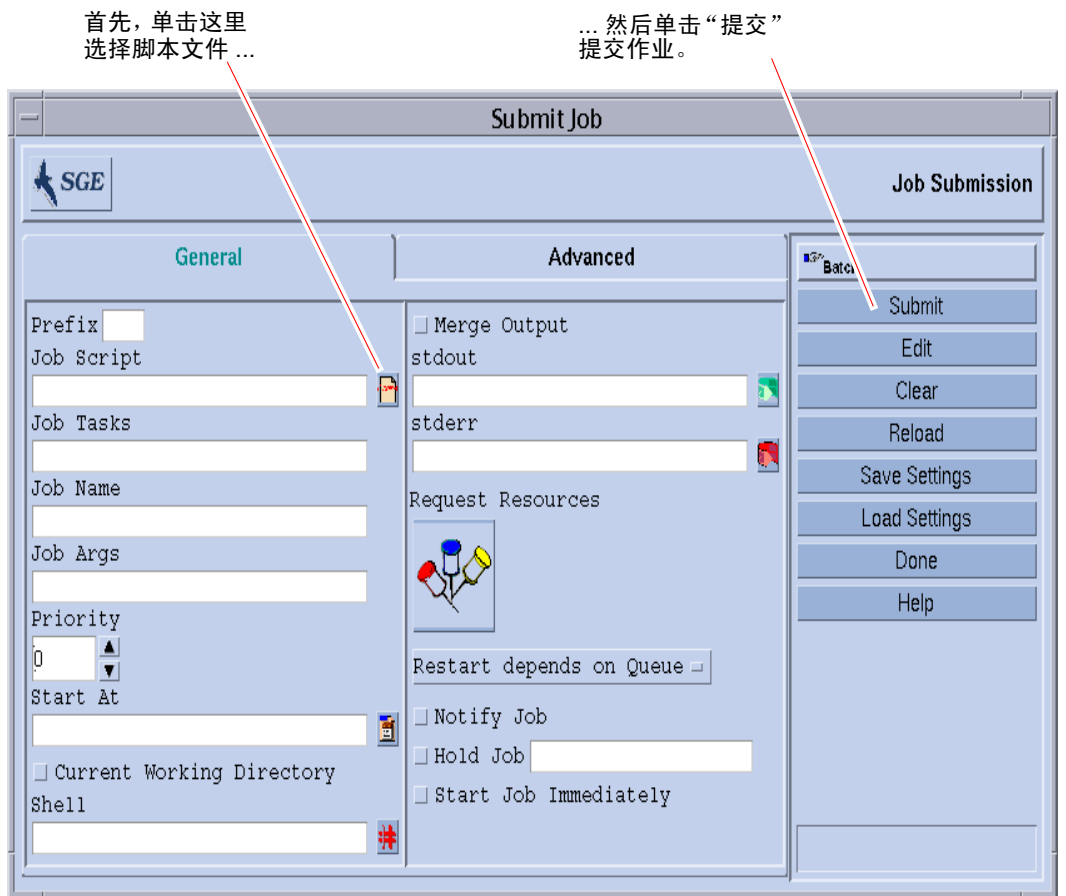


图 4-2 QMON 的“作业提交”对话框

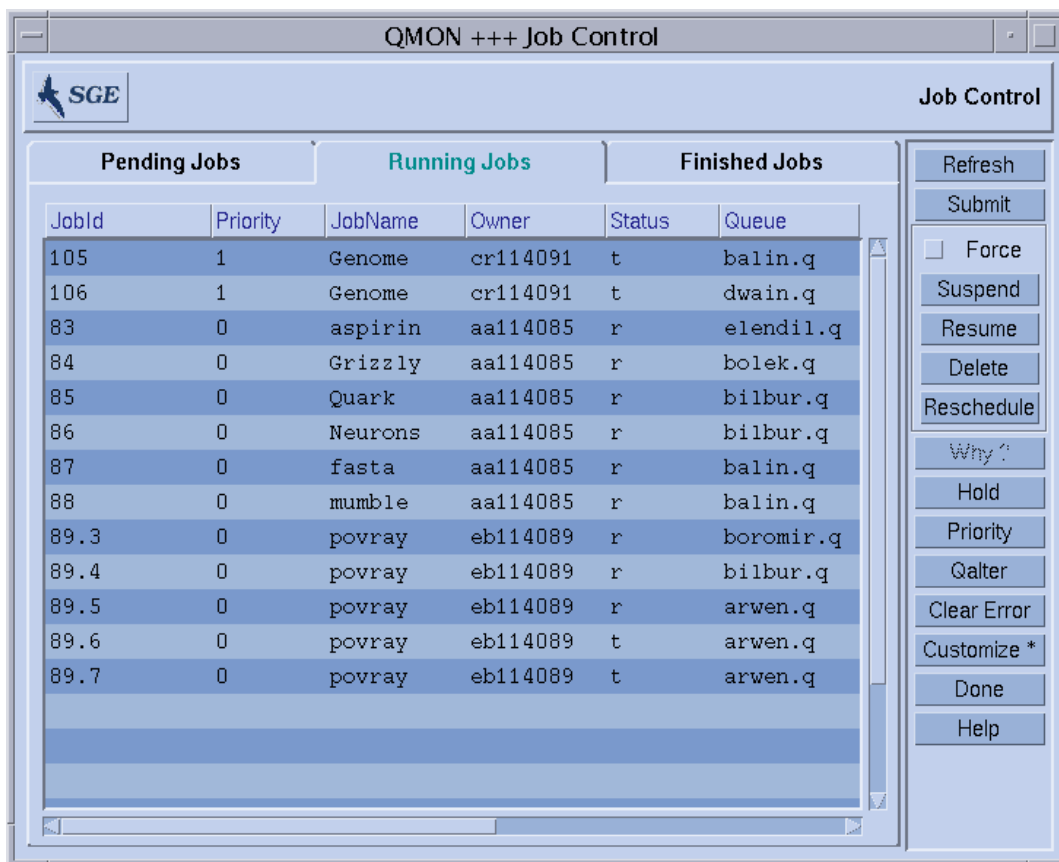


图 4-3 QMON 的“作业控制”对话框

5. 在“作业提交”菜单中，单击“作业脚本”文件选择图标打开文件选择框。
6. 单击相应的文件名选择脚本文件（例如，命令行示例中的 *simple.sh*）。
7. 单击“作业提交”菜单下面的“提交”按钮。

几秒钟以后，您就可以在“作业控制”面板中监视作业了。该作业首先将显示在“暂挂的作业”列表中，启动后将很快移至“正运行的作业”列表。

提交批处理作业

以下各节介绍了如何通过 Sun Grid Engine 5.3 程序提交更复杂的作业。

关于 Shell 脚本

Shell 脚本即批处理作业，主要指集成到一个文件中的一系列命令行指令。chmod 命令可使脚本文件变成可执行文件。一旦调用脚本，即可启动相应的命令解释器（例如，csh、tcsh、sh 或 ksh），解释每条指令，其结果等同于执行脚本的用户手动输入这些指令。您可以在一个 shell 脚本内调用任意命令、应用程序和其它 shell 脚本。

相应的命令解释器是否作为 login-shell 调用，取决于其名称（csh、tcsh、sh、ksh、...）是否包含在执行该作业的特定主机和队列所使用的 Sun Grid Engine 配置的 login_shells 项值列表中。

注意 – 群集中配置的各个主机和队列的 Sun Grid Engine 配置可能有所不同。可以通过 qconf 命令的 -sconf 和 -sq 选项显示有效配置（详细信息请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》）。

若命令解释器作为 login-shell 被调用，则作业环境将如同您登录并执行此脚本。例如，若使用 csh，除系统缺省启动资源文件（例如，/etc/login）外，还将执行 .login 和 cshrc；但是若 csh 不是作为 login-shell 被调用的，则只执行 .cshrc。有关作为和不作为 login-shell 调用的差别的描述，请参见所使用的命令解释器的手册页。

脚本文件示例

代码示例 4-1 是简单 shell 脚本的示例，它首先编译应用程序 flow 的 Fortran77 源文件，然后执行它。

```
#!/bin/csh

# This is a sample script file for compiling and
# running a sample FORTRAN program under Sun Grid
Engine.

cd TEST

# Now we need to compile the program 'flow.f' and
# name the executable 'flow'.

f77 flow.f -o flow
```

代码示例 4-1 简单 Shell 脚本

您的本地系统用户指南将提供有关建立和自定义 shell 脚本的详细信息（也可查看 sh、ksh、csh 或 tcsh 手册页）。以下各节的重点是：为 Sun Grid Engine 准备批处理脚本而要考虑的特殊事项。

一般来说，您可以从命令提示符下手动把所有您可执行的 shell 脚本提交给 Sun Grid Engine，只要它们不需要终端连接（标准错误和输出设备除外，它们会被自动重定向），且不需要交互式的用户干预。这样，代码示例 4-1 可提交到 Sun Grid Engine，并将执行所需操作。

用 QMON 提交扩展作业和高级作业

尝试更复杂的作业提交形式（*扩展作业* 或 *高级作业*）之前，最好了解一些有关此过程的重要背景信息。以下各节即提供这些信息。

扩展作业示例

“作业提交”对话框的标准形式（请参见图 4-2）提供了为扩展作业配置以下参数的方法。

- 前缀字符串，用于脚本内嵌的 Sun Grid Engine 提交选项（有关详细信息，请参见第 82 页的“有效的 Sun Grid Engine 注释”）

- 要使用的作业脚本
 - 按下相关文件按钮即可打开文件选择框（请参见图 4-3）。
- 提交阵列作业的任务 ID 范围（请参见第 86 页的“阵列作业”）
- 作业名（会在选定作业脚本后设置一个缺省值）
- 作业脚本自变量
- 作业的初始优先级值
 - 没有管理人员或操作人员权限的用户只能降低其初始优先级值。
- 可考虑执行作业的时间
 - 若按下相关文件按钮，将出现一个辅助对话框，可在此输入格式正确的时间（请参见图 4-4）。
- 一个标志，用于指明作业是否在当前工作目录中执行（仅适用于提交主机和可能的执行主机之间目录分层结构相同的情况）
- 命令解释器，用于执行作业脚本（请参见第 81 页的“如何选择命令解释器”）
 - 若按下相关按钮，将出现一个辅助对话框，可在此输入打开的作业的命令解释器（请参见图 4-5）。
- 一个标志，用于指明是否将作业的标准输出和标准错误输出合并成标准输出流
- 要使用的标准输出重定向（请参见第 81 页的“输出重定向”）
 - 若不加指定，将使用缺省值。若按下相关文件按钮，将出现一个辅助对话框，可在此输入其它的输出重定向（请参见第 81 页的“输出重定向”）。
- 要使用的标准错误输出重定向。与标准输出重定向极其相似
- 作业的资源需求
 - 要定义作业所需资源，请按相应的图标按钮。若资源已被某作业请求，图标按钮的颜色会改变。
- 选择列表按钮，用于定义当系统崩溃或有类似事件导致作业异常中止时，作业是否能重新启动；以及重新启动操作是取决于队列还是遵照作业的要求
- 一个标志，用于指明作业将被暂停或取消时，是由 SIGUSR1 信号还是由 SIGUSR2 信号进行通知
- 一个标志，用于表明是为作业指定用户等候，还是指定作业从属性
 - 只要作业被指定了任何类型的等候，就不会执行（有关等候的更多信息，请参见第 106 页的“监视和控制 Sun Grid Engine 作业”）。“等候”标志对应的输入字段可用于将等候限制为只针对阵列作业的特定范围的任务（请参见第 86 页的“阵列作业”）。
- 一个标志，用于强制作业尽可能立即启动或被拒绝
 - 若选择此标志，作业不进入队列。

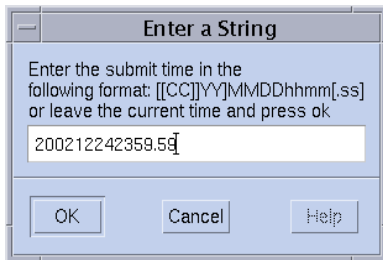


图 4-4 “时间输入”对话框

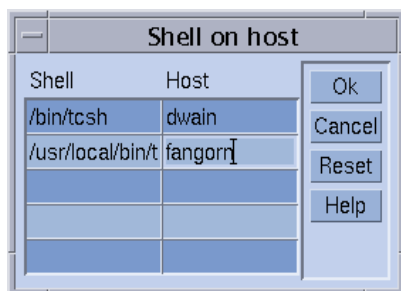


图 4-5 “Shell 选择”对话框

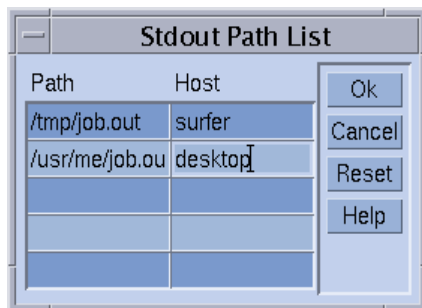


图 4-6 “输出重定向”对话框

“作业提交”屏幕右边的按钮可以启动各种操作：

- **提交** – 按照对话框的指定，提交作业。
- **编辑** – 在 X-terminal 中，用 vi 或 \$EDITOR 环境变量中定义的编辑器编辑选定的脚本文件。
- **清除** – 清除“作业提交”对话框中的所有设定，包括所有指定的资源请求。

- **重新加载** – 重新加载指定的脚本文件，分析所有脚本内嵌的选项（请参见第 82 页的“有效的 Sun Grid Engine 注释”一节），分析缺省设定（请参见第 85 页的“缺省请求”）并放弃对这些设定间接的手动更改。此操作相当于对前一脚本文件执行“清除”操作并重新进行指定。只有已选定脚本文件时此选项才生效。
- **保存设置** – 保存对文件的当前设置。文件选择框将打开以选择文件。保存后的文件可以在将来明确加载（参见下文），也可以用作缺省请求（请参见第 85 页的“缺省请求”）。
- **加载设置** – 加载以前用“保存设置”按钮保存的（参见上文）设置。加载的设置将覆盖当前设置。
- **完成** – 关闭“作业提交”对话框。
- **帮助** – 显示与此对话框相关的帮助。

图 4-7 显示了带有大部分参数设定的“作业提交”对话框。

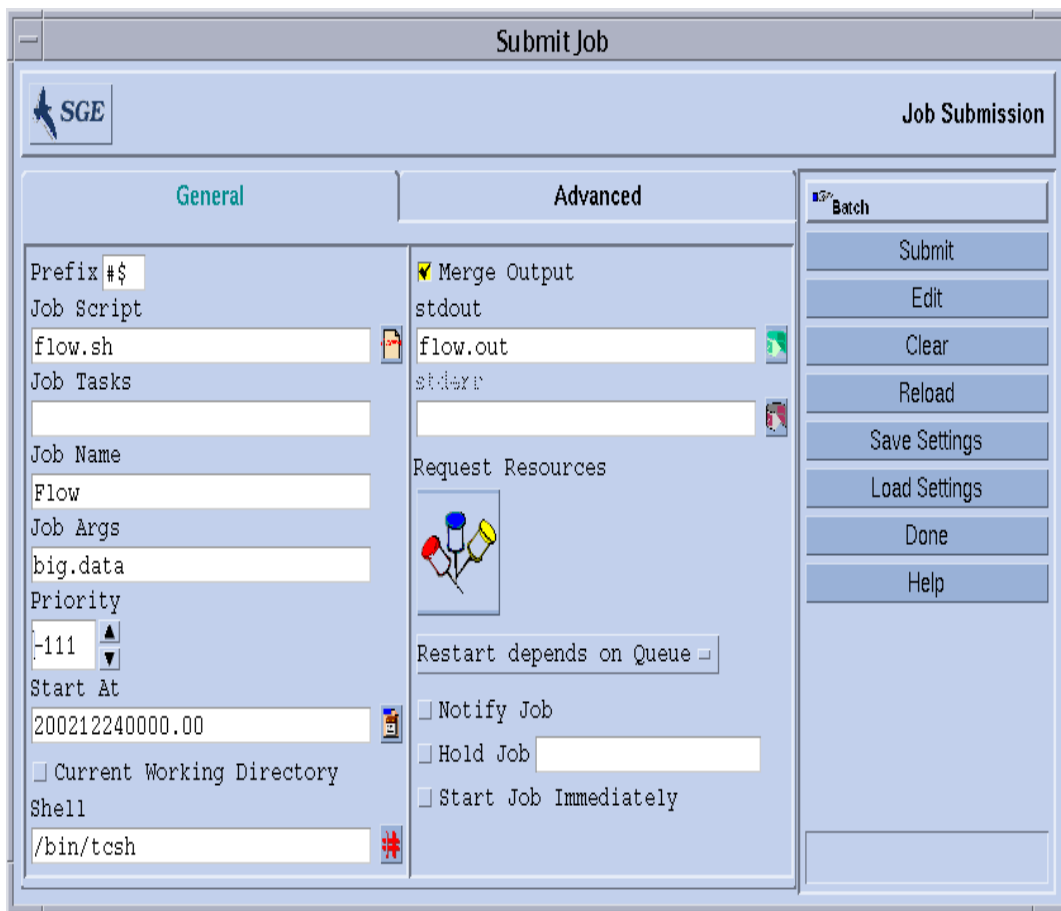


图 4-7 扩展作业提交示例

示例中配置的作业有脚本文件 `flow.sh`，该文件必须位于 QMON 的工作目录中。作业名为 `Flow`，且脚本文件使用单个自变量 `big.data`。作业启动时优先级为 `-111`，并且将在 2002 年 12 月 24 日之后执行。作业将在提交工作目录之中执行，并将使用 `tcs` 命令解释器。最后，标准输出和标准错误输出将被合并至文件 `flow.out`，该文件也将创建在当前工作目录中。

高级示例

“高级”提交屏幕可用于定义以下附加参数：

- 要使用的并行环境接口及所需的进程范围（请参见第 93 页的“并行作业”一节）
- 在作业执行前为作业设置的一组环境变量
若按下相关图标按钮，将出现一个辅助对话框，用以定义待导出的环境变量（请参见图 4-8）。环境变量可从 QMON 的运行时环境中获得，也可以定义任意环境变量。
- 名为“背景”的名称 / 值对应的列表（请参见图 4-9），用于存储和交流可从 Sun Grid Engine 群集内任何地方访问的作业相关信息
背景变量可从命令行通过 `qsub`、`qrsh`、`qsh`、`qlogin` 或 `qalter` 的 `-ac/-dc/-sc` 选项修改，也可以用 `qstat -j` 命令检索。
- 点检查环境，用于需要并且适合进行点检查的作业（请参见第 101 页的“关于点检查作业”）
- 与作业相关的帐户字符串
帐户字符串将被添加至作业的帐户记录中，并将用于将来的帐户分析。
- 验证标志，决定作业的一致性检查模式
为了检查作业请求的一致性，Sun Grid Engine 假定有一个空的未加载的群集，并试图查找至少一个可运行此作业的队列。可能的检查模式为：
 - 跳过 - 根本不进行一致性检查。
 - 警告 - 报告不一致性，但仍接受作业（若群集配置将在作业提交后更改，可使用此模式）。
 - 错误 - 报告不一致性，若遇到任何问题，将拒绝此作业。
 - 仅验证 - 不提交作业，但会生成有关群集中每个主机和队列是否适合于此作业的详细报告。
- 将用电子邮件通知用户的事件
当前定义了作业的启动 / 结束 / 中止 / 暂停事件。
- 接收电子邮件通知的电子邮件地址列表
若按下相关按钮，将出现一个辅助对话框，用于定义邮件列表（图 4-10）。

- 队列名称列表，是执行作业的必不可少的选项
“必需队列列表”和“可选队列列表”的处理方式与相应的资源需求相同，如第71页的“作业的资源需求”中的项目符号列表项所述。
- 能够充当并行作业的*主控队列*的队列名称列表
并行作业在*主控队列*中启动。作业派生的并行任务所在的所有其它队列称为*从属队列*。
- 作业 ID 列表，成功完成此列表后才能启动作业的提交。
新创建的作业*依赖于*那些作业的成功完成。

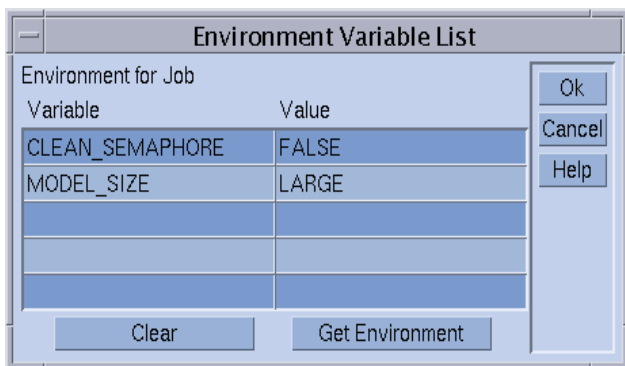


图 4-8 作业环境定义

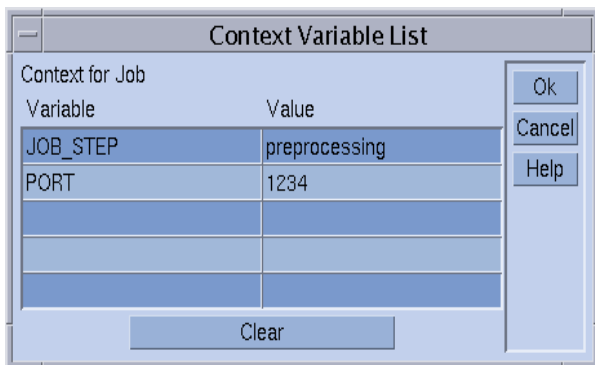


图 4-9 作业背景定义

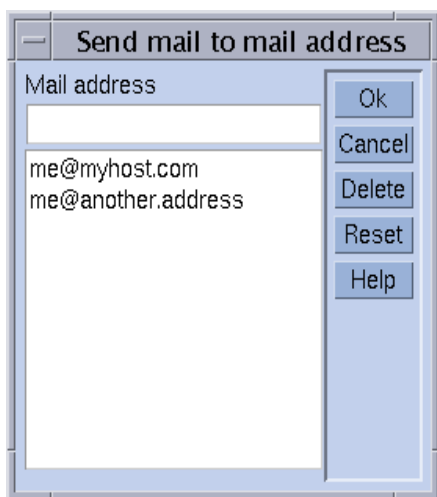


图 4-10 邮件地址指定

与图 4-11 一节中的作业定义相比，第 70 页的“扩展作业示例”中定义的作业具有以下附加特性。

- 作业要求使用并行环境 `mpi`。至少需要创建 4 个并行进程，最多可利用 16 个进程。
- 为作业设定并导出 2 个环境变量。
- 设定 2 个背景变量。
- 帐户字符串 `FLOW` 将被添加至作业帐户记录。
- 若作业因系统崩溃而失败，将重新启动。
- 若检测到作业请求与群集配置不一致，将给出警告。
- 作业一经启动和完成，邮件将发送给 2 个电子邮件地址。
- 最好是在队列 `big_q` 中执行此作业。

图 4-11 显示了高级作业提交的示例。

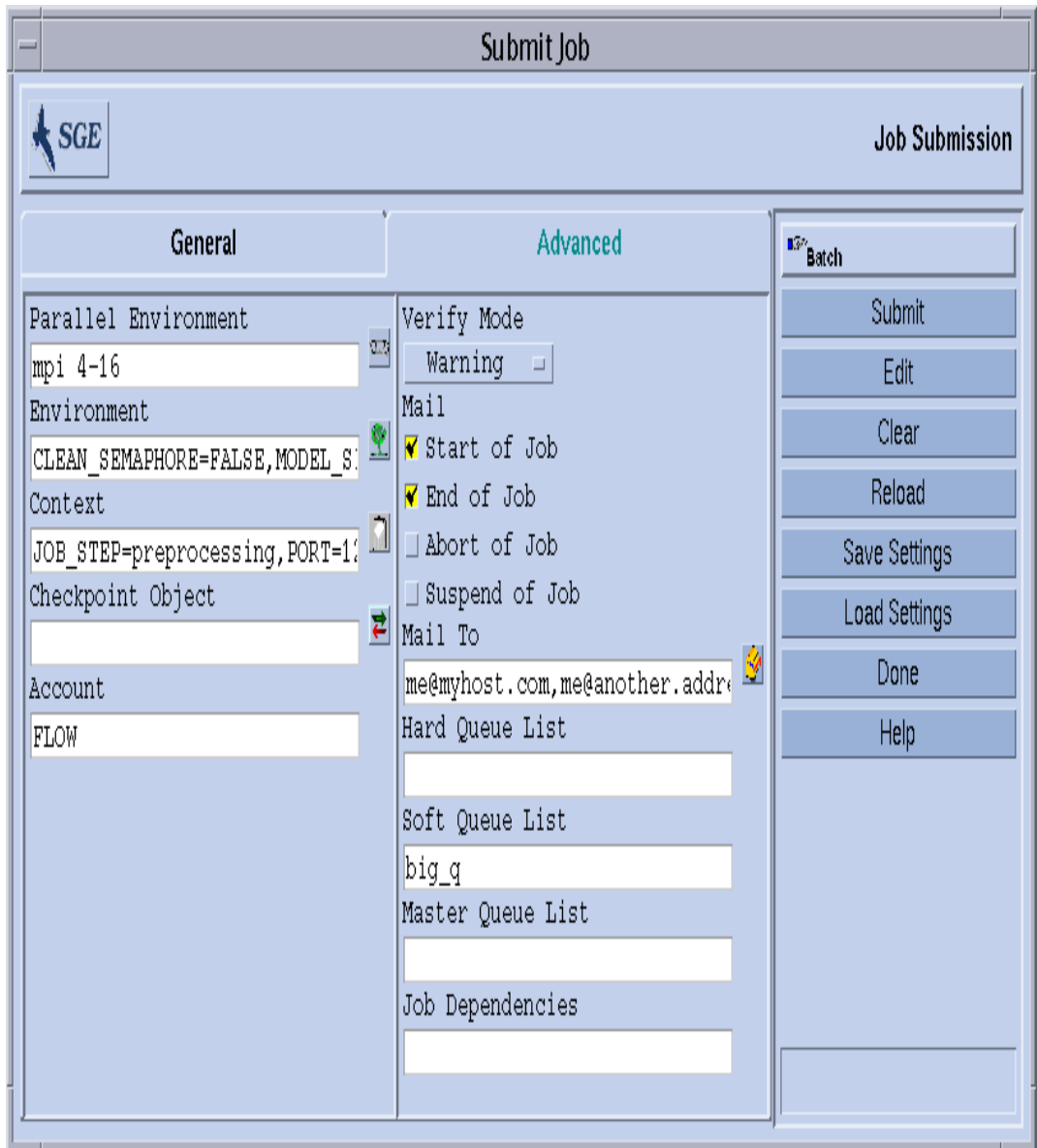


图 4-11 高级作业提交示例

资源需求定义

到目前为止的例子中，提交选项没有指出对执行作业的主机的需求。Sun Grid Engine 假定此类作业可在任何主机上运行。然而，实际上要执行主机成功完成作业，大部分作业要求一些先决条件。这些先决条件包括足够的可用内存、安装所需软件或某种操作系统体系结构。而且，群集管理者也经常对群集中的机器加上一些使用限制。例如，作业可使用的 CPU 时间就常常受到限制。

Sun Grid Engine 为用户提供了一种方式，用户无需对群集设备及其所用策略有详细了解，即可找到适合于用户作业的主机。用户只需指定用户作业的要求，然后让 Sun Grid Engine 管理相应任务，即查找合适的且负荷较轻的主机的任务。

资源需求可通过第 56 页的“可请求的属性”一节中介绍的*可请求的属性*来指定。QMON 提供了一种指定作业需求的简便方法。“请求的资源”对话框（按“作业提交”对话框中的“请求的资源”按钮即可打开，请参见图 4-12 的示例）仅列出了“可用资源”选择列表中当前可用的那些属性。双击某个属性，该属性将添加到作业的“必需资源”或“可选资源”列表（参见下文）中，并且会出现辅助对话框，指导您输入相关属性的值。BOOLEAN 类型属性除外，它们将设为 True。

图 4-12 中显示的“请求的资源”对话框示例指明了一个作业的资源概况，其中要求至少有 750 MB 内存的 solaris64 主机，且该主机要有可用的 permass 许可证。若找到多个满足此条件的队列，任何软性资源需求都可考虑（本例中无）。不过，若找不到硬性资源和软性资源需求都满足的队列，任何满足硬性资源需求的队列都认为是合适的。

注意 – 只有在多个队列适用于作业时，才使用负荷标准确定在何处启动作业。

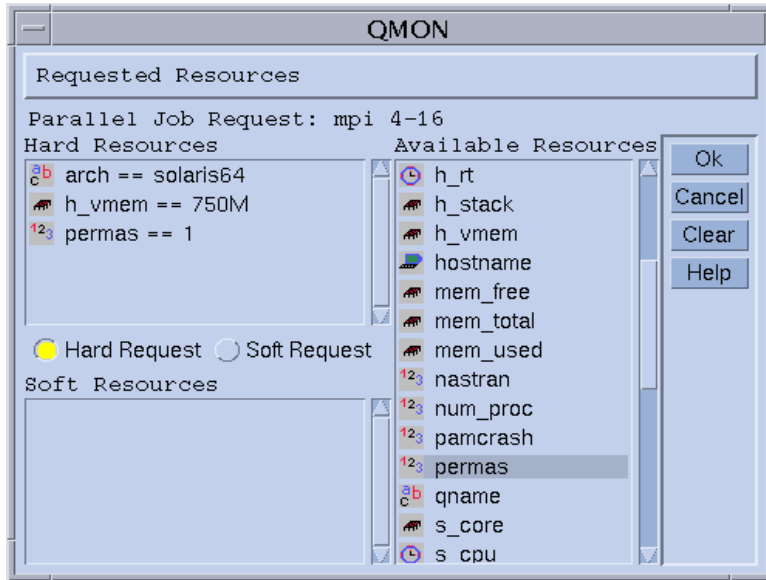


图 4-12 “请求的资源”对话框

注意 – “整数”属性 `permas` 通过管理员扩展引入到“全局”属性组，“字符串”属性 `arch` 是从“主机”属性组导入的，而“内存”属性 `h_vmem` 是从“队列”属性组中导入的。

相当的资源需求概况也可以从 `qsub` 命令行提交：

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

注意 – 第一个 `-l` 选项之前暗含的 `-hard` 开关选项被省略。

符号 `750M` 表示 750 兆字节，是 Sun Grid Engine 数量语法的示例。对那些请求内存使用的属性，可以用十进制整数、十进制浮点数、八进制整数和十六进制整数后跟所谓的乘数来指定：

- `k` – 将值乘以 1000。
- `K` – 将值乘以 1024。
- `m` – 将值乘以 1000 的平方。
- `M` – 将值乘以 1024 的平方。

指定八进制常数应以 0（零）开头，数字范围仅限于从 0 到 7。指定十六进制常数应在数字前面加上 0x，数字为 0 到 9、a 到 f 和 A 到 F。若其后未跟乘数，则值为字节数。若使用十进制浮点数，结果值将被取整为整数。

对于那些有时间限制的属性，可以按照时、分或秒及其任何组合来指定时间值。时、分和秒用十进制表示，用冒号分隔。时间 3:5:11 将被转换为 11111 秒。若时、分、秒的指定值为 0，有冒号时 0 可省略。因此，:5: 会解释为 5 分钟。以上“请求的资源”对话框中使用的形式为扩展形式，仅在 QMON 中有效。

Sun Grid Engine 系统如何分配资源

如上节所示，了解 Sun Grid Engine 软件如何处理请求和如何分配资源十分重要。以下简要提供 Sun Grid Engine 软件的资源分配算法。

1. 读入并分析所有缺省的请求文件（请参见第 85 页的“缺省请求”）。
2. 处理脚本文件的内嵌选项（请参见第 82 页的“有效的 Sun Grid Engine 注释”）。
3. 提交作业时读取所有脚本的内嵌选项，而不考虑其在脚本文件中的位置。
4. 从命令行读取和分析所有请求。

一旦收集了所有 qsub 请求，*硬性*和*软性*请求将分别处理（硬性优先）。根据以下优先顺序评估请求：

1. 脚本 / 缺省请求文件从左到右
2. 脚本 / 缺省请求文件从上到下
3. 命令行从左到右

换句话说，命令行可用来覆盖嵌入的标志。

分配所请求的硬性资源。若请求无效，将拒绝提交。若提交时无法满足一个或多个请求（例如被请求的队列正忙），作业将假脱机，稍后重新调度。若所有硬性请求都能满足，将分配这些资源，作业可以运行。

检查所请求的软性资源。即使部分或全部请求无法满足，作业仍可运行。若多个队列（已满足硬性资源请求）提供了部分软性资源列表（重叠或不同），Sun Grid Engine 软件将选择满足最多软性请求的队列。

作业将会启动，并占用已分配的资源。

可通过用执行 UNIX 命令（如 hostname 或 date）的小测试脚本文件，来积累一些有关自变量列表选项和内嵌选项或硬性和软性请求是如何互相影响的经验，这样做十分有益。

常规 Shell 脚本的扩展

常规 shell 脚本有一些扩展，它们在 Sun Grid Engine 的控制下运行会影响脚本的运行。以下各节描述这些扩展。

如何选择命令解释器

用于处理作业脚本文件的命令解释器可在提交时指定（请参见图 4-7 的示例）。不过，若没有指定，配置变量 `shell_start_mode` 将决定如何选择命令解释器：

- 若 `shell_start_mode` 设为 `unix_behavior`，脚本文件的第一行（若以“#!”序列开始）将被评估，以确定命令解释器。若第一行没有“#!”序列，缺省情况下将使用 Bourne Shell `sh`。
- 对于 `shell_start_mode` 的所有其它设定，则使用启动作业的队列的 `shell` 参数配置值，来作为缺省命令解释器（请参见第 50 页的“队列和队列特性”和 `queue_conf` 手册页）。

输出重定向

由于批处理作业无终端连接，因此，必须将其标准输出和标准错误输出重定向到文件。Sun Grid Engine 允许用户定义输出重定向的文件的位置，但若不指定位置，将使用缺省值。

文件的标准位置是执行作业的当前工作目录。缺省标准输出文件名为 `< 作业名 >.o< 作业 ID >`，缺省的标准错误输出将重定向至 `< 作业名 >.e< 作业 ID >`。`< 作业名 >` 可从脚本文件名创建或由用户定义（请参见 `qsub` 手册页中 `-N` 选项的示例）。`< 作业 ID >` 为 Sun Grid Engine 分配给作业的唯一标识符。

如果是阵列作业任务（请参见第 86 页的“阵列作业”），这些文件名后会加上任务标识符，用句点分隔。因此最后得出的标准重定向路径为 `< 作业名 >.o< 作业 ID >.< 任务 ID >` 和 `< 作业名 >.e< 作业 ID >.< 任务 ID >`。

若标准位置不合适，用户可用 `QMON` 指定输出方向（如图 4-11 和图 4-6 中所示），或者用 `-e` 和 `-o qsub` 选项指定输出方向。标准输出和标准错误输出可以合并为一个文件，并可在每一台执行主机上指定重定向。即，根据执行作业的主机的不同，输出重定向文件的位置也不同。要自定义唯一的重定向文件路径，可使用伪环境变量，这些变量可与 `qsub -e` 和 `-o` 选项一起使用。以下为此类变量的列表。

- `$HOME` – 执行主机上的主目录
- `$USER` – 作业拥有者的用户 ID
- `$JOB_ID` – 当前作业的 ID
- `$JOB_NAME` – 当前作业名（参见 `-N` 选项）
- `$HOSTNAME` – 执行主机名

- \$TASK_ID – 阵列作业任务索引号

作业运行时这些变量替换为实际值，重定向路径由此建立。

有关详细信息，参见《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》中的 qsub 项。

有效的 Sun Grid Engine 注释

shell 脚本中以“#”符号开头的行是注释行。不过，Sun Grid Engine 能识别特殊的注释行，并以特殊的方法使用它们：此类脚本行的其余部分将视作 Sun Grid Engine 提交命令 qsub 的命令行自变量列表的一部分进行处理。特殊注释行内的 qsub 选项也由 QMON 的“作业提交”对话框解释，而且相应参数会在选择脚本文件时预先设置。

缺省情况下，特殊注释行用“#\$”前缀字符串标识。前缀字符串可用 qsub -C 选项重新定义。

所描述的机制称为提交自变量的脚本嵌入。以下为脚本文件的示例，该文件使用内嵌脚本的命令行选项。

```
#!/bin/csh
#Force csh if not Sun Grid Engine default shell
#$ -S /bin/csh
# This is a sample script file for compiling and
# running a sample FORTRAN program under Sun Grid Engine.
# We want Sun Grid Engine 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
```

代码示例 4-2 使用脚本内嵌的命令行选项

环境变量

当 Sun Grid Engine 作业运行时，许多变量会预先设置到作业的环境中，如下所列：

- ARC – 作业在其上运行的节点的 Sun Grid Engine 体系结构名，该名称将编译到 `sgc_execd` 二进制文件中
- COMMD_PORT – 指定 `sgc_commd(8)` 将在其上侦听通讯请求的 TCP 端口
- SGE_ROOT – 启动前为 `sgc_execd` 设置的 Sun Grid Engine 根目录或缺省的 `/usr/SGE`
- SGE_CELL – 作业在其中执行的 Sun Grid Engine 单元
- SGE_JOB_SPOOL_DIR – `sgc_shepherd(8)` 在作业执行时用于存储作业的相关数据的目录
- SGE_O_HOME – 作业拥有者在提交作业的主机上的主目录
- SGE_O_HOST – 作业提交的主机
- SGE_O_LOGNAME – 作业拥有者在提交作业的主机上的登录名
- SGE_O_MAIL – 作业提交命令的背景中 MAIL 环境变量的内容
- SGE_O_PATH – 作业提交命令的背景中 PATH 环境变量的内容
- SGE_O_SHELL – 作业提交命令的背景中 SHELL 环境变量的内容
- SGE_O_TZ – 作业提交命令背景中 TZ 环境变量的内容
- SGE_O_WORKDIR – 作业提交命令的工作目录
- SGE_CKPT_ENV – 指定在其中执行点检查作业的点检查环境（同 `qsub -ckpt` 选项所选定的值）
- SGE_CKPT_DIR – 仅为点检查工作设置；包括点检查接口的路径 `ckpt_dir`（请参见 checkpoint 手册页）
- SGE_STDERR_PATH – 作业的标准错误流转向的文件路径名；通常用于以前导程序、收尾程序、并行环境启动 / 停止或点检查脚本中的错误消息扩充输出。
- SGE_STDOUT_PATH – 作业的标准输出流转向的文件路径名；通常用于以前导程序、收尾程序、并行环境启动 / 停止或点检查脚本中的错误消息扩充输出。
- SGE_TASK_ID – 阵列作业中此任务代表的任务标识符
- ENVIRONMENT – 总是设为 BATCH；此变量表示脚本以批处理模式运行
- HOME – 来自 `passwd` 文件的用户主目录路径
- HOSTNAME – 作业在其上运行的节点的主机名
- JOB_ID – 作业提交时，由 `sgc_qmaster` 分配的唯一标识符，作业 ID 的范围为 99999 以内的十进制整数
- JOB_NAME – 作业名，由 `qsub` 脚本文件名、一个句点和作业 ID 的数字组成；此缺省值可以被 `qsub -N` 覆盖。
- LOGNAME – 来自 `passwd` 文件的用户登录名

- NHOSTS – 并行作业正在使用的主机数
- NQUEUES – 分配给作业的队列数（对于串行作业，总为 1）
- NSLOTS – 并行作业正使用的队列位置数
- PATH – 缺省的 shell 搜索路径，即：
/usr/local/bin:/usr/ucb:/bin:/usr/bin
- PE – 执行作业的并行环境（仅用于并行作业）
- PE_HOSTFILE – 文件路径，该文件含有 Sun Grid Engine 为并行作业分配的虚拟并行机的定义
此文件格式的详细信息，请参见 sge_pe 中 \$pe_hostfile 参数的描述。此环境变量仅可用于并行作业。
- QUEUE – 在其中运行作业的队列名
- REQUEST – 作业请求名称，可以是作业脚本文件名或通过 qsub -N 选项明确地分配给作业的名称
- RESTARTED – 指出是否启动了点检查作业；如果已设置（为 1），则表明作业至少中断了一次并且已重新启动
- SHELL – 来自 passwd 文件的用户登录 shell

注意 – 不一定是作业正在使用的 shell。

- TMPDIR – 作业临时工作目录的绝对路径
- TMP – 与 TMPDIR 相同；提供此环境变量是为了与 NQS 兼容
- TZ – 从 sge_execd 导入的时区变量（如果已设置）
- LOGNAME – 来自 passwd 文件的用户登录名

▼ 如何从命令行提交作业

- 输入 qsub 命令及其相应自变量。

例如，使用脚本文件名 flow.sh 的简单作业（如第 63 页的“如何从命令行运行简单作业”中所述）可以用此命令提交：

```
% qsub flow.sh
```


若要产生与扩展的 QMON 作业提交相同的结果（如图 4-7 中所示），应使用以下命令：

```
% qsub -N Flow -p -l11 -a 200012240000.00 -cwd \  
-S /bin/tcsh -o flow.out -j y flow.sh big.data
```

可进一步添加命令行选项，组成更复杂的请求。例如，图 4-11 中所示的高级作业请求应类似于以下命令：

```
% qsub -N Flow -p -l11 -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.sh big.data
```

缺省请求

上节的最后一个示例说明高级作业请求可能变得非常复杂且不易操作，尤其是在需要经常提交类似请求的时候。为避免输入这些繁琐和易于出错的命令行，用户可以在脚本文件中嵌入 qsub 选项（请参见第 82 页的“有效的 Sun Grid Engine 注释”）或使用所谓的缺省请求。

群集管理者可为所有 Sun Grid Engine 用户设置缺省请求文件。另一方面，用户可在用户主目录下创建私用的缺省请求文件，也可以在工作目录下创建应用程序专用的缺省请求文件。

缺省请求文件只是在一行或几行中包括 qsub 选项，这些选项将被缺省用于 Sun Grid Engine 作业中。群集全局缺省文件的位置为 *<sge 根目录>/<单元>/common/sge_request*。私用的一般缺省请求文件位于 *\$HOME/.sge_request* 之下，而特定于应用程序的缺省请求文件位于 *\$cwd/.sge_request* 下。

若有多个此类文件可用，它们将按照以下优先顺序合并为一个缺省请求文件：

1. 全局缺省请求文件。
2. 通用私用缺省请求文件。
3. 特定于应用程序的缺省请求文件。

注意 – 脚本的内嵌项和 qsub 命令行的优先顺序高于缺省请求文件。因此，脚本的内嵌项将覆盖缺省请求文件设定，而 qsub 命令行选项又能覆盖这些设定。

注意 – 可以随时在缺省请求文件、内嵌的脚本命令和 `qsub` 命令行中使用 `qsub -clear` 选项，来废弃以前的设定。

以下列出了私用缺省请求文件的示例。

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

若非覆盖，指定用户的所有作业的帐户字符串都是 `myproject`，作业将在当前工作目录下执行，邮件通知将在作业的开始和结束时发送到 `me@myhost.com`，系统崩溃后作业将重新启动，标准输出和标准错误输出将合并，`ksh` 将被用作命令解释器。

阵列作业

同一组操作（包含于一个作业脚本中）参数化的和重复的执行是 **Sun Grid Engine 阵列作业** 工具的最理想应用。此类应用的典型示例可以在数字内容生成行业的任务（如绘制）中找到。本例中，动画计算分帧进行，每一帧独立进行同样的绘制计算。

阵列作业工具提供提交、监视和控制此类应用的简便方法。另一方面，**Sun Grid Engine** 提供了阵列作业的有效执行，能将一组独立的任务当作组合成单个作业的方式来处理计算。阵列作业的任务可以通过阵列索引引用。所有任务的索引横跨整个阵列作业的索引范围，该索引范围是在提交阵列作业时通过单个 `qsub` 命令定义的。

阵列作业可以作为一个整体、或按单个任务或任务子集的形式来监控（如暂停、恢复或取消），若不作为整体监控，相应索引号将加到作业 ID 之后，以引用这些任务。任务执行时（与常规作业相似），可以使用环境变量 `$SGE_TASK_ID` 获得其自身的索引号，访问指定给此任务标识符的输入数据集。

▼ 如何从命令行提交阵列作业

- 输入 `qsub` 命令，并带上相应的自变量。

以下为提交阵列作业的示例。

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

`-t` 选项定义了任务索引范围。此例中，`2-10:2` 说明 2 为最小索引号，10 为最大索引号，使用的每个索引号相差 2（`:2` 部分）。因此，阵列作业由 5 个任务组成，任务索引为 2、4、6、8 和 10。每个任务请求 45 分钟（`-l` 选项）的硬性 CPU 时间限制，并且一旦由 Sun Grid Engine 分配和启动，将执行 `render.sh` 作业脚本。任务可以用 `$SGE_TASK_ID` 查出任务是 2、4、6、8 或 10，并且可以用索引号查找数据文件 `data.in` 中的输入数据记录。

▼ 如何用 QMON 提交阵列作业

- 遵照第 65 页的“如何从图形用户界面 QMON 提交作业”中的指导，另请考虑以下注意事项。

注意 – 从 QMON 提交阵列作业实际上与第 65 页的“如何从图形用户界面 QMON 提交作业”中描述的很相似。唯一差别是图 4-7 中的“作业任务”输入窗口需要任务范围值，范围可用与 `qsub -t` 选项相同的语法来指定。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `qsub` 项，获取阵列索引语法的详细信息。

第 106 页的“监视和控制 Sun Grid Engine 作业”和第 118 页的“从命令行控制 Sun Grid Engine 作业”，以及《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中有关 `qstat`、`qhold`、`qrls`、`qmod` 和 `qdel` 的章节，包含了监控 Sun Grid Engine 的一般作业和特殊的阵列作业的相关信息。

注意 – 阵列作业能够完全访问常规作业已知的所有 Sun Grid Engine 工具。特别是，它们同时可以是并行作业，或与其它作业相互依存。

提交交互式作业

在作业需要您的直接输入以影响其结果时，提交交互式作业（而不是批处理作业）特别有用。典型的情况是定义为交互式的 X-windows 应用程序，或是那些需要您对直接结果进行解释以控制进一步计算的任务。

Sun Grid Engine 系统中存在 3 种创建交互式作业的方法。

- `qlogin` – 类似于 `telnet` 会话，在 Sun Grid Engine 软件选定的主机上启动。
- `qrsh` – 这是一个等效于标准的 UNIX `rsh` 的工具。命令在 Sun Grid Engine 系统选定的主机上远程执行，若未指定要执行的命令，会在远程主机上启动远程登录 (`rlogin`) 会话。
- `qsh` – 这是一个 `xterm`，它从执行作业的计算机启动，其显示设置与您的指定值或 `DISPLAY` 环境变量相对应。若未设定 `DISPLAY` 变量且未明确定义显示目标，Sun Grid Engine 将把 `xterm` 定向到提交交互式作业的主机的 X 服务器的 0.0 屏幕。

注意 – 若要正确运行，所有工具都需要适当配置 Sun Grid Engine 群集参数。必须为 `qsh` 定义正确的 `xterm` 执行路径，必须有适用于此类作业的交互式队列。请与系统管理员联系，询问群集是否准备就绪可以执行交互式作业。

交互式作业的缺省处理不同于批处理作业的处理，若交互式作业未能在提交时执行，它们并不排队。这是为了在交互式作业提交后，立即指明无足够的适用资源分配给该作业。这种情况下用户会立即得到通知，告知 Sun Grid Engine 群集此时非常忙碌。

缺省操作可用 `qsh`、`qlogin` 和 `qrsh` 的 `-now no` 选项更改。若指定了此选项，交互式作业将像批处理作业一样排队。使用 `-now yes`，用 `qsub` 提交的批处理作业也可像交互式作业一样处理，即要么立即被分配执行要么被拒绝。

注意 – 交互式作业只能在 INTERACTIVE 类型（交互式）的队列中执行（详情请参见第 149 页的“关于配置队列”）。

接下来的各节概述了 `qlogin` 和 `qsh` 工具的用法。第 94 页的“透明的远程执行”一节对 `qrsh` 命令作了更多的解释。

用 QMON 提交交互式作业

唯一一种可从 QMON 提交的交互式作业是那些在 Sun Grid Engine 选定的主机上启动 `xterm` 的作业。

▼ 如何用 QMON 提交交互式作业

- 单击“作业提交”对话框右边按钮栏顶部的图标，直到“交互式”图标出现。

这使“作业提交”对话框准备就绪以提交交互式作业（请参见图 4-13 和图 4-14）。

对话框中选择选项的含义和用法与第 69 页的“提交批处理作业”一节中介绍的批处理作业的相同。基本差别在于几个输入字段在此无效，因为它们不适用于交互式作业。

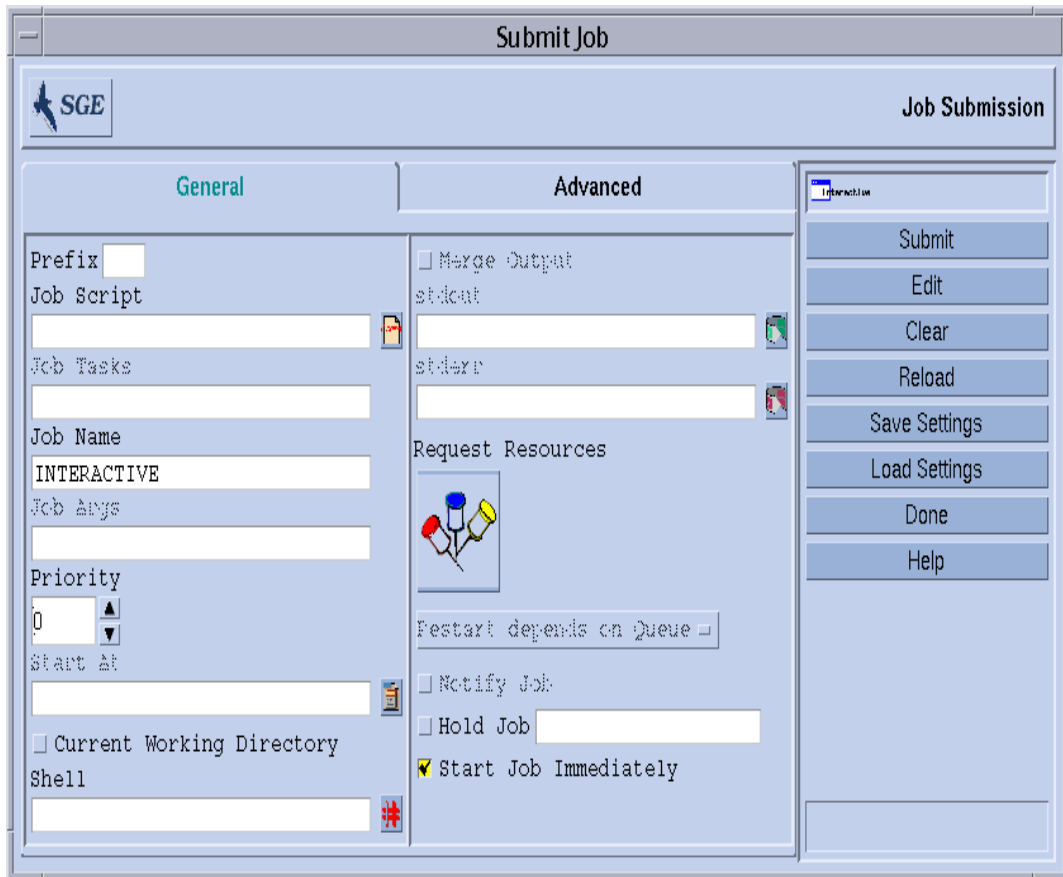


图 4-13 “交互式作业提交”对话框，常规

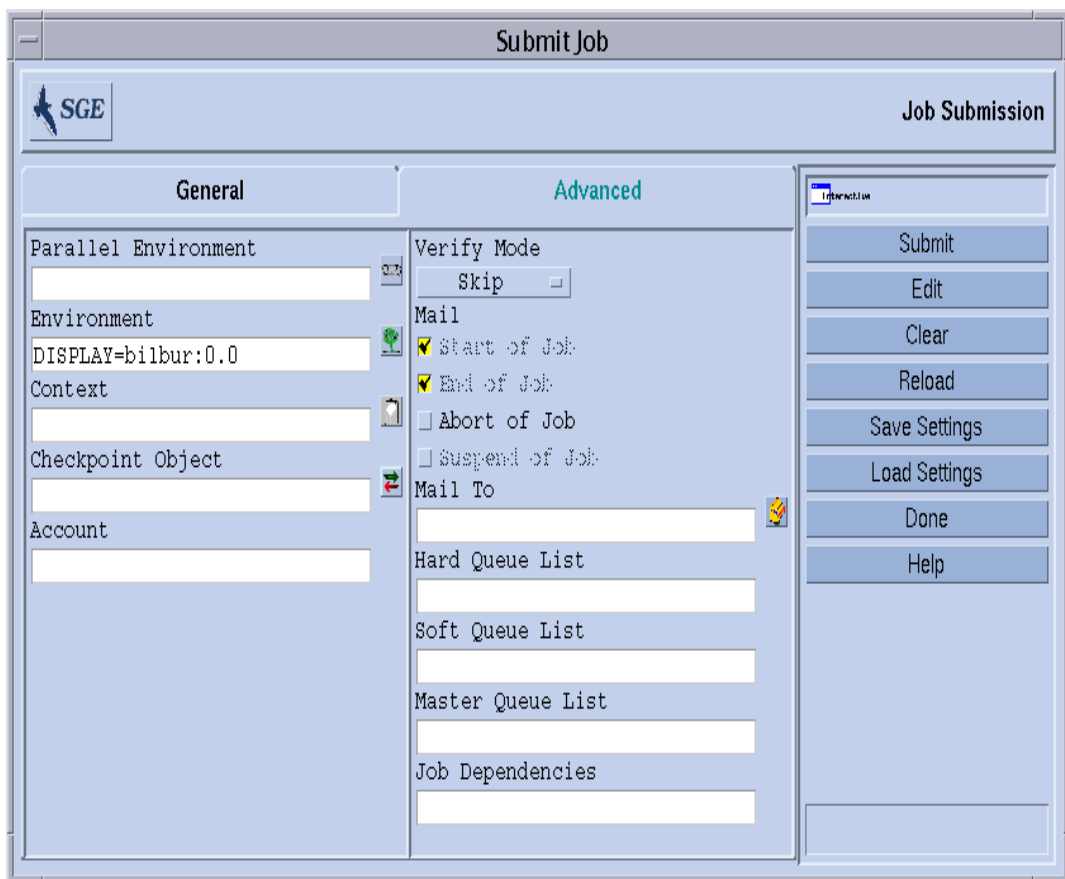


图 4-14 “交互式作业提交”对话框，高级

用 qsh 提交交互式作业

Qsh 与 qsub 非常相似，且支持数个 qsub 选项以及附加的开关选项 `-display` 以便控制要调用的 `xterm` 的显示（有关详情，请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `qsh` 项）。

▼ 如何用 qsh 提交交互式作业

- 在任何可用的 Sun Solaris 64 位操作系统主机上，输入以下命令启动 `xterm`。

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

用 qlogin 提交交互式作业

`qlogin` 命令可用于从任何终端或终端仿真程序启动 Sun Grid Engine 控制下的交互式会话。

▼ 如何用 qlogin 提交交互式作业

- 输入以下命令，找到一台低负荷的主机，它应带有可用的 Star-CD 许可证，且至少拥有一个能提供不少于 6 小时的硬性 CPU 时间限制的队列。

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

注意 – 可能需要在登录提示符下输入用户名、口令或同时输入两者，这取决于为 Sun Grid Engine 系统配置的远程登录工具。

并行作业

Sun Grid Engine 提供执行并行作业的方式，可使用任意消息传递环境，例如 PVM 或 MPI（请参见《*PVM User's Guide*》和《*MPI User's Guide*》以获得细节信息）；或使用共享的内存并行程序，该程序位于单个队列中的多个位置上；或在多个队列和（对分布式内存并行作业）计算机之间分布。可同时配置任意数目的不同并行环境 (PE) 接口。请参见第十章第 223 页的“管理并行环境”以获得有关 PE 的细节。

如何调度 Sun Grid Engine 作业

基本上，Sun Grid Engine 5.3 软件使用两套标准调度作业：

- 作业优先级
- 等份额

作业优先级

关于不同作业的调度过程顺序，缺省情况下应用 *先进先出* (fifo) 规则。所有 *暂挂*（尚未调度）作业均插入到一个列表中，第一个提交的作业在列表的头部，接下来是第二项提交的作业，等等。首先尝试调度先提交的作业。若至少有一个可用的合适队列，则作业将得到调度。Sun Grid Engine 软件此后将试图调度第二项作业，而不论第一项作业是否已分配。

暂挂作业中的此优先顺序可通过由群集管理员分配给作业的 *优先级值* 覆盖。实际的优先级值可通过使用 `qstat` 命令显示（优先级值包含在标题为 `P` 的暂挂作业显示的最后一栏，请参考第 116 页的“如何用 `qstat` 监视作业”以获得细节）。提交时分配给作业的缺省优先级值为 0。优先级值为正整数和负整数，并且暂挂作业列表按相应优先级值的降序排列。通过分配给作业一个相对较高的优先级值，可使作业移动到暂挂作业列表的顶端。优先级值为负的作业甚至会置于刚提交的作业之后。若几项作业有相同的优先级值，则在该优先级值范围内应用 *fifo* 规则。

等份额调度

fifo 规则有时会导致问题，尤其是若用户试图同时提交一系列作业的情况下（例如，通过 `shell` 脚本发出一个接一个的提交）。所有后来提交的和指定到同一组队列的作业均不得等待很长的时间。*等份额调度* 将已拥有运行作业的用户作业排列在优先级列表的结尾，从而避免了这一问题。排序仅在同一优先级值范围的作业中执行。若 Sun Grid Engine 调度程序配置项 `user_sort` 设置为 `TRUE`（请参考 `sched_conf` 手册页以获得细节），激活等份额调度。

队列选择

Sun Grid Engine 系统不分配请求非特定队列的作业，如果它们不能即时启动的话。此类作业将在 `sge_qmaster` 中标记为假脱机，该命令会不时尝试重新调度它们。于是，作业将分配给下一个可用的合适队列。

与此相反，在请求中指明队列名的作业将直接进入该队列，无论它们是否能启动或者处于假脱机状态。因此，将 Sun Grid Engine 队列视为计算机科学中的 *批处理队列* 仅对用名称请求的作业有效。用非特定请求提交的作业使用 `sge_qmaster` 的假脱机机制排队，从而使用更抽象和更灵活的排队概念。

若作业被调度，并且有多个空闲队列满足其资源请求，则在合适的队列中，通常作业将被分配给负荷最轻的主机上的队列。通过将 Sun Grid Engine 调度程序配置项 `queue_sort_method` 设置为 `seq_no`，群集管理者可将此依赖于负荷的方案更改为固定的顺序算法：用队列配置项 `seq_no` 定义队列的优先顺序，具有最高优先级的队列的序列号最低。

透明的远程执行

Sun Grid Engine 提供了一套紧密相关的工具，支持某些计算任务的透明远程执行。实现此功能的核心工具为第 94 页的“使用 `qrsh` 进行远程执行”中描述的 `qrsh` 命令。建立于 `qrsh` 之上的两个高层工具 `qtcsh` 和 `qmake` 允许通过 Sun Grid Engine 进行隐含的计算任务的透明分配，从而增强标准的 UNIX 工具 `make` 和 `csh`。`qtcsh` 在第 95 页的“用 `qtcsh` 进行透明的作业分配”一节中讲述，而 `qmake` 在第 97 页的“用 `qmake` 执行并行的 Makefile 处理”一节中讲述。

使用 `qrsh` 进行远程执行

`Qrsh` 是基于 `rsh` 工具创建的（请参见 `<sge 根目录>/3rd_party` 提供的信息，以了解有关 `rsh` 的详情），且具有多种用途。

- 通过 Sun Grid Engine 提供交互式应用程序的远程执行，类似于标准 UNIX 工具 `rsh`（在 HP-UX 中又称为 `remsh`）。
- 通过 Sun Grid Engine 提供交互式登录会话功能，类似于标准 UNIX 工具 `rlogin`（注意，仍然需要 `qlogin` 作为 UNIX `telnet` 工具的 Sun Grid Engine 代表）。
- 允许提交批处理作业，这些作业一开始执行就支持终端 I/O（标准 / 错误输出和标准输入）以及终端控制。
- 提供一种提交未嵌入 `shell` 脚本中的独立程序的方法。

- 提供批处理作业提交客户端程序，该程序在作业暂挂或执行时始终处于活动状态，且仅在作业完成或取消时结束。
- 允许在并行作业分配的分布式资源的框架之内，在 Sun Grid Engine 系统控制下远程执行作业任务（如一个并行作业的并发任务）（请参见第 232 页的“PE 和 Sun Grid Engine 软件的紧密集成”一节）。

凭借这些功能，qrsh 成为使 qtcsh 和 qmake 工具得以实现的主要基础架构，对于 Sun Grid Engine 与并行环境（如 MPI 或 PVM）的所谓紧密集成也是如此。

qrsh 用法

qrsh 命令的常规格式为：

```
% qrsh [选项] program|shell-script [自变量] \
    [> stdout 文件] [>&2 stderr 文件] [< stdin 文件]
```

qrsh 几乎能理解所有的 qsub 选项，且提供了几个附加选项。

- `-now yes|no` – 此选项控制若无合适的可用资源，作业是否立即被调度并拒绝（这是缺省值，因为通常这正是交互式作业所需要），或者若作业不能于提交时启动，是否像批处理作业一样排队。
- `-inherit` – qrsh 不会仔细检查 Sun Grid Engine 调度进程以启动作业任务，但它假定其已嵌入到并行作业背景内部，而该并行作业已经在指定的远程执行主机上分配了合适的资源。这种形式的 qrsh 通常在 qmake 内部以及紧密集成的并行环境内部使用。缺省值为不继承外部作业资源。
- `-verbose` – 此选项显示有关调度进程的输出。其主要用途是调试，因此缺省情况下此选项是关闭的。

用 qtcsh 进行透明的作业分配

qtcsh 为众所周知且广为使用的 UNIX C-Shell (csh) 的派生物 tcsh 的完全兼容的替代品（qmake 是基于 tcsh 创建的。请参见 *<sge 根目录>/3rd_party* 中提供的信息，以获取有关 tcsh 的详细信息）。它为命令 shell 提供扩展功能，通过 Sun Grid Engine 将指定应用程序的执行透明地分配到适合的且负荷较低的主机。远程执行哪些应用程序，哪些需求适用于执行主机的选择，均在称为 .qtask 的配置文件中定义。

这类应用程序可通过 `qrsh` 工具提交给 Sun Grid Engine 执行。这对用户是透明的。`qrsh` 提供了标准输出、错误输出和标准输入处理，以及到远程执行应用程序的终端控制连接，所以，与在 `shell` 所在主机上执行此类应用程序相比，远程执行此类应用程序仅有三个显著差别。

- 远程主机可能比本地主机更合适（更加强大、负荷更低、安装了所需的硬件 / 软件资源），后者可能根本不允许执行应用程序。当然，这正是我们想要的差别。
- 作业远程启动以及通过 Sun Grid Engine 处理可能会导致少许延迟。
- 管理员可通过交互式作业 (`qrsh`) 以及 `qtcsh` 限制资源的用量。若无足够的适用资源供应用程序通过 `qrsh` 工具启动，或者若所有合适系统均超负荷，隐含的 `qrsh` 提交将失败，并返回相应的错误消息 (Not enough resources ... try later)。

除了标准用途外，`qtcsh` 还是第三方代码和工具集成的合适平台。在集成环境内以单一应用程序执行形式 `qtcsh -c 应用程序名` 使用 `qtcsh` 能提供一个几乎永远不需要更改的永久接口。所有所需的应用程序、工具、集成、站点甚至用户专用的配置都包含在正确定义的 `.qtask` 文件中。一个更大的优点是此接口可从任意类型的 `shell` 脚本、C 程序甚至 Java 应用程序中使用。

qtcsh 用法

`qtcsh` 的调用与 `tcsh` 完全相同。`qtcsh` 扩展了 `tcsh`，增加了对 `.qtask` 文件的支持，还提供了一组专门的 `shell` 内置模式。

`.qtask` 文件定义如下。文件中每行均遵守以下格式：

```
% [!] 应用程序名 qrsh 选项
```

句首可选的惊叹号 (!) 定义了群集全局 `.qtask` 文件和 `qtcsh` 用户的个人 `.qtask` 文件之间存在相冲突的定义时，这二者的优先顺序。若群集全局文件中没有惊叹号，则最终用户文件中冲突的定义有效。若群集全局文件中有惊叹号，则该文件中的相应定义有效。

命令行的其余部分指定应用程序名（在 `qtcsh` 命令行中输入时，该应用程序名将被提交给 Sun Grid Engine 进行远程执行）和 `qrsh` 工具的选项（应用程序将会使用这些选项，而且这些选项会为应用程序定义资源需求）。

注意 – 出现在命令行中的应用程序名必须与 `.qtask` 文件中的定义完全相同。若其前面加上了绝对路径或相对路径，则假定寻址的是本地二进制文件且不需要远程执行。

注意 – 不过，Csh 别名将在与应用程序名进行比较之前扩展。要远程执行的应用程序可出现在 qtcsh 命令行的任何地方，特别是在标准 I/O 重定向的前后。

因此，以下示例是有效且有意义的语法：

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

若给定此 .qtask 文件，以下 qtcsh 命令行：

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

就意味等同以下命令行：

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

qtcsh 可以运行在多种模式下，这些模式由开关选项控制，其中每个开关选项都可设置为可开关：

- 命令在本地或在远程执行（缺省为远程）
- 即时或远程批处理执行（缺省为即时）
- 冗余或非冗余输出（缺省为非冗余）

这些模式的设定可以在启动时用 qtcsh 的选项自变量更改，或在运行时用 shell 内置命令 qrshmode 更改。有关更多信息，参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 qtcsh 项。

用 qmake 执行并行的 Makefile 处理

qmake 是标准 UNIX make 工具的替代品。它能够将独立的 make 步骤在群集中的多台合适的计算机间进行分配，从而扩展了 make。qmake 是基于流行的 GNU make 工具 gmake 建立的。有关涉及 gmake 的细节，请参见 *<sgc 根目录>/3rd_party* 中提供的信息。

为确保所分配的复杂 make 进程能完成运行，qmake 首先用类似于并行作业的形式分配所需资源。然后 Qmake 管理此资源集，无需与 Sun Grid Engine 调度进一步交互作用。它在资源可用时通过启用 `-inherit` 选项的 `qrsh` 工具，分配 make 步骤。

由于 `qrsh` 提供了标准输出、错误输出和标准输入处理，以及远程执行 make 步骤的终端控制连接，本地执行 make 过程或使用 qmake 之间仅有三个显著差别：

- 假设单个 make 步骤需一定运行时间，并且有足够多的独立 make 步骤需要处理，则 make 进程的并行处理将明显加速。当然，这正是我们想要的差别。
- 执行远程启动的 make 步骤时，存在隐含的少量的系统开销，如由 `qrsh` 和远程执行所引起的开销。
- 若要利用 qmake 对 make 步骤的分配，用户须指定最小的并行程度，即，可同时执行的 make 步骤的数目。此外，用户可指定 make 步骤所需的资源特性，如可用的软件许可证、机器体系结构、内存或 CPU 时间要求。

一般而言，make 最常见的用途是复杂软件包的编译。然而，这可能不是 qmake 的主要应用。程序文件通常都很小（良好的编程理应如此），因此，单个程序文件的编译（作为一个 make 步骤）通常只需要几秒钟。此外，编译通常意味着众多的文件访问（内嵌的包含文件），如果并行处理多个 make 步骤，这种访问并不会加快，因为文件服务器可能成为有效串行处理所有文件访问的瓶颈。所以有时不能期待编译进程能有令人满意的提速。

qmake 其它潜在的应用会更合适。比如，控制通过 `make-file` 的复杂分析任务的互相依赖性以及工作流。这在某些领域（如 EDA）中很常见，并且此类环境中的每个 make 步骤一般为具有不可忽略资源和计算时间需求的仿真或数据分析操作。此类情况下能显著加速。

qmake 用法

qmake 的命令行语法看上去与 `qrsh` 的一种语法非常相似：

```
% qmake [-pe PE 名称 PE 范围][更多的选项] \  
-- [GNU make 选项][目标]
```

注意 – 如本节稍后所述，qmake 也支持 `-inherit` 选项。

必须特别注意 `-pe` 选项的用法及其与 `gmake -j` 选项的关系。两个选项都能用于表示将要完成的并行量。差别在于 `gmake` 不能用 `-j` 指定诸如将要使用的并行环境等等。因此，qmake 假定：并行 make 的缺省环境已配置好，且该环境称为

make。此外，gmake 的 -j 无法指定范围，只能指定一个编号。Qmake 将 -j 指定的编号解释为 1-<给定编号> 的范围。与此相反，-pe 允许详细指定所有参数。因此，以下两命令行示例是相同的。

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

而以下命令行就不能通过 -j 选项表示：

```
% qmake -pe make 5-10.16 --
% qmake -pe make 1-99999 --
```

除此语法之外，qmake 还支持两种调用模式：从命令行交互调用（不带 -inherit）或在批处理作业内部调用（带有 -inherit）。这两种模式将启动一组不同的操作：

- **交互式** – 当在命令行调用 qmake 时，make 进程将隐含通过 qrsh 提交给 Sun Grid Engine，同时会考虑 qmake 命令行中指定的资源需求。然后 Sun Grid Engine 会选择一台 **主控主机** 来执行与并行的 make 作业相关的并行作业，并且在其上启动 make 过程。这是必要的，因为 make 进程可能与体系结构有关，且所需体系结构会在 qmake 命令行中指定。然后主控主机上的 qmake 进程会将各个 make 步骤的执行委托给其它主机。这些主机已由 Sun Grid Engine 分配给此作业，并且已通过并行环境主机文件传递给 qmake。
- **批处理** – 这种情况下，qmake 将带 -inherit 选项出现在批处理脚本内部（若无 -inherit 选项，将会派生新的作业，如第一种情况所述）。这将导致 qmake 利用已经分配给 qmake 所嵌入的作业的资源。它将直接使用 qrsh -inherit 启动 make 步骤。以批处理模式调用 qmake 时，资源需求的指定值或 -pe 和 -j 选项将被忽略。

注意 – 单 CPU 的作业也需要请求并行环境 (qmake -pe make 1 --)。若不需要并行执行，用 gmake 命令行语法调用 qmake（不带 Sun Grid Engine 选项和 “-- ”），它执行的操作与 gmake 类似。

参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 qmake 项，获取有关 qmake 的更多信息。

点检查、监视和控制作业

用 Sun Grid Engine 5.3 系统提交作业后，您需要有监视和控制它们的能力。本章提供了有关完成这些任务的背景信息和指导。

本章包含以下具体任务的指导。

- 第 104 页的 “如何从命令行提交、监视或删除点检查作业”
- 第 104 页的 “如何用 QMON 提交点检查作业”
- 第 106 页的 “如何用 QMON 监视和控制作业”
- 第 116 页的 “如何用 qstat 监视作业”
- 第 118 页的 “如何用电子邮件监视作业”
- 第 119 页的 “如何从命令行控制作业”
- 第 120 页的 “如何用 QMON 控制队列”
- 第 124 页的 “如何用 qmod 控制队列”

关于点检查作业

本章探讨了作业点检查的两种类型。

- *用户级别*
- *内核级别*

用户级别的点检查

许多应用程序，尤其是那些通常消耗很多 CPU 时间的应用程序，已经运用了点检查和重新启动机制，以增强容错能力。状态信息和所处理数据的重要部分在算法的某些阶段被重复写入一个或多个文件。这些文件（称为重新启动文件）可以在应用程序中止后重新启动时处理，并达到与点检查之前一致的状态。由于用户通常要处理这些重新启动文件以便将其移至合适的位置，所以这种点检查被称为*用户级别*点检查。

对于那些没有集成（用户级别）点检查的应用程序，另一种选择是使用所谓的*点检查库*，该库可由公共域提供（例如，参见 University of Wisconsin 的 *Condor* 项目）或由一些硬件供应商提供。将应用程序与此类库重新链接，即可在该应用程序中安装点检查机制，而无需更改源代码。

内核级别的点检查

某些操作系统在操作系统内核内提供了点检查支持。此类情况下，无需准备应用程序，也无需重新链接应用程序。内核级别的点检查通常既适用于单个进程，也适用于整个进程分层结构。即，可以随时对互相依赖的进程的分层结构进行点检查和重新启动。通常，用户命令和 C 库界面都可用来启动点检查。

Sun Grid Engine 支持操作系统点检查（若可用）。有关当前支持的内核级别点检查工具的信息，请参见《Sun Grid Engine 发行说明》。

点检查作业的迁移

点检查作业可随时中断，因其重新启动功能保证了只需重复极少的已经完成的工作。此功能用于建立 Sun Grid Engine 的迁移和动态负荷平衡机制。若经请求，Sun Grid Engine 的点检查作业将根据要求中止并迁移到 Sun Grid Engine 池中的其它计算机中，因而以动态方式均衡群集中的负荷。以下原因会导致点检查中止并迁移。

- 正在执行的队列或作业被 `qmod` 或 `qmon` 命令明令暂停。
- 正在执行的队列或作业自动暂停，原因是：已经超过队列的暂停阈值（请参见第 154 页的“如何配置负荷和暂停阈值”一节），并且作业的点检查时机说明中包括了暂停情形（请参见第 104 页的“如何从命令行提交、监视或删除点检查作业”一节）。

您可通过 `qstat` 输出中的状态 `m` 来识别将要迁移的作业。迁移的作业移回 `sge_qmaster`，接着被分配给另一个合适的队列（若存在可用队列）。

编写点检查作业脚本

内核级别点检查的 shell 脚本与常规 shell 脚本没有差别。

用户级别点检查作业的 Shell 脚本与常规的 Sun Grid Engine 批处理脚本仅在合理处理重新启动的能力上有所差别。环境变量 RESTARTED 是为重新启动的点检查作业设置的。它可用于跳过作业脚本中只应在初次调用时执行的部分。

因而，透明点检查作业脚本应该与代码示例 5-1 类似。

```
#!/bin/sh
#Force /bin/sh in Sun Grid Engine
#$ -S /bin/csh

# 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
```

代码示例 5-1 点检查作业脚本示例

务必记住：若迁移了用户级别的点检查作业，则作业脚本从开头重新启动。用户要负责将 shell 脚本的程序流引导至作业中断的位置，从而跳过脚本中那些若执行多次会产生重大影响的命令行。

注意 – 内核级别的点检查作业可随时中断，内含的 shell 脚本也将从上次进行点检查之处重新启动。因此，RESTARTED 环境变量与内核级别的点检查作业无关。

▼ 如何从命令行提交、监视或删除点检查作业

输入以下带有相应开关选项的命令。

```
#qsub 选项 自变量
```

除 `qsub -ckpt` 和 `-c` 选项（它们请求点检查机制并且定义对作业进行点检查的时机）外，提交点检查作业的方式与常规批处理脚本相同。`-ckpt` 选项带一个自变量，它是要使用的点检查环境的名称（第 216 页的“关于点检查支持”）。`-c` 选项不是必需的，它也带一个自变量。它可用于覆盖点检查环境配置中 `when` 参数的定义（有关详情，参见《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中的 `checkpoint` 项）。

`-c` 选项的自变量可以是以下单字母选项中的任一个（或其任意组合）或时间值。

- `n` – 不执行点检查。此项优先级最高。
- `s` – 检查点仅在作业主机上的 `sge_execd` 关闭时才产生。
- `m` – 按相应队列配置中定义的最小 CPU 时间间隔产生检查点（参见 `queue_conf` 手册页中的 `min_cpu_interval` 参数）。
- `x` – 作业暂停时产生检查点。
- `interval` – 以给定时间间隔产生检查点，但其频率不高于 `min_cpu_interval` 定义的值（参见上文）。时间值必须以 `hh:mm:ss` 形式指定（小时两位、分钟两位、秒两位，用冒号分开）。

点检查作业的监视方式与常规作业不同，因为这些作业可能不时迁移（通过 `qstat` 输出中的迁移状态 `m` 表示，请参见上述内容），并且因此不会固定于某个队列。不过，唯一的作业标识号以及作业名保持不变。

点检查作业的删除方式与第 118 页的“从命令行控制 Sun Grid Engine 作业”中描述的不同。

▼ 如何用 QMON 提交点检查作业

- 遵照第 74 页的“高级示例”中的指导，并注意以下附加信息。

除需要另外指定合适的点检查环境外，通过 QMON 提交点检查作业与提交常规批处理作业相同。如第 74 页的“高级示例”中的过程所述，“作业提交”对话框为作业相关的点检查环境提供了一个输入窗口。输入窗口旁有一个图标按钮，可打开图 5-1 中显示的“选择”对话框。它可用于从可用点检查环境列表中选择合适的点检查环境。询问系统管理员本站点安装的点检查环境的有关特性信息，或参见第 216 页的“关于点检查支持”一节。

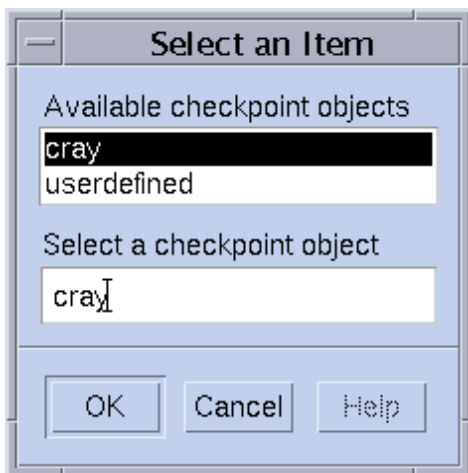


图 5-1 点检查对象的选择

文件系统需求

基于点检查库的用户级别或内核级别点检查写入之后，需要转储要进行点检查的进程或作业所占据的虚拟内存的完整映像。为此，需要有足够的可用磁盘空间。若设置了点检查环境配置参数 `ckpt_dir`，则点检查信息将转储到 `ckpt_dir` 目录下的作业私用位置。若 `ckpt_dir` 设置为 `NONE`，则将使用曾在其中启动点检查作业的目录。关于点检查环境配置的详细信息，请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中的 `checkpoint` 项。

注意 – 若 `ckpt_dir` 设为 `NONE`，则应该用 `qsub -cwd` 脚本启动点检查作业。

有关文件系统如何组织的附加需求产生于这样一个事实，即点检查文件和重新启动文件必须在所有计算机上可见，以保证成功迁移和重新启动作业。因此需要 NFS 或相似的文件系统。询问群集管理者站点是否满足需求。

若您的站点不运行 NFS 或出于某种原因不适合使用 NFS，则对于用户级别的点检查作业，应确保能在 shell 脚本开始处就明令传输重新启动文件（例如，通过 `rcp` 或 `ftp`）。

监视和控制 Sun Grid Engine 作业

原则上，有三种方法可监视提交的作业。

- 使用 Sun Grid Engine 图形用户界面 QMON
- 在命令行使用 `qstat` 命令
- 通过电子邮件

▼ 如何用 QMON 监视和控制作业

Sun Grid Engine 图形用户界面 QMON 提供了专门为控制作业而设计的对话框。

- 在 QMON 主菜单中，按“作业控制”按钮，然后根据以下各部分的详细附加信息继续执行。

此对话框的主要目的是提供一种方法，用于监视系统已知的所有正运行的、暂挂的、以及一定数目（此数可配置）的已完成作业，或其中的一部分。对话框还可用于控制作业，如更改其优先级、暂停、恢复和取消它们。对话框中有三个列表环境，一个是针对正在运行的作业，一个是针对暂挂的等待分配给合适资源的作业，一个是针对最近完成的作业。单击屏幕上部相应的选项卡标签，可以从这三个列表环境中选择。

缺省表单（参见图 5-2）显示各个正运行的以及暂挂的作业的作业 ID、优先级、作业名称和队列。

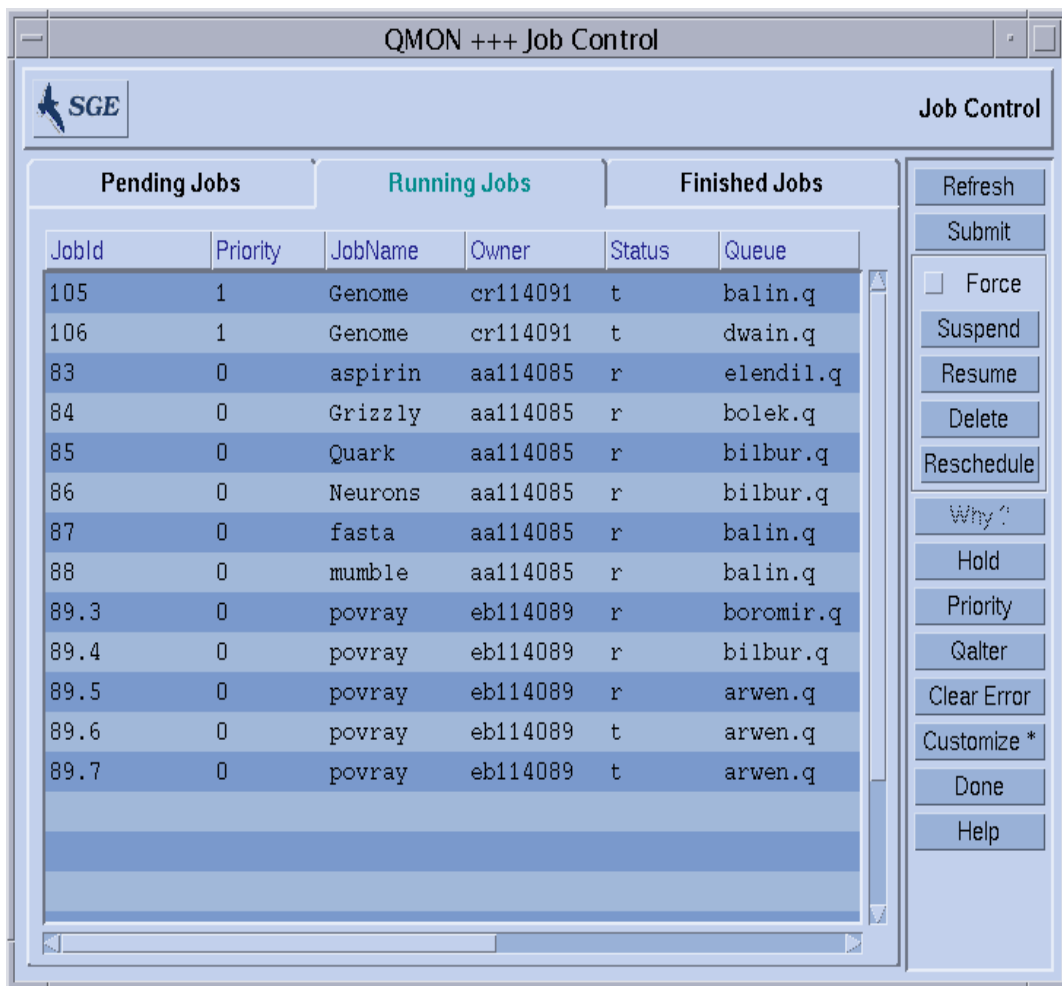


图 5-2 “作业控制”对话框 — 标准表单

可以在“自定义”对话框中配置所显示的这组信息（参见图 5-3），单击“作业控制”对话框中的“自定义”按钮即可打开该对话框。

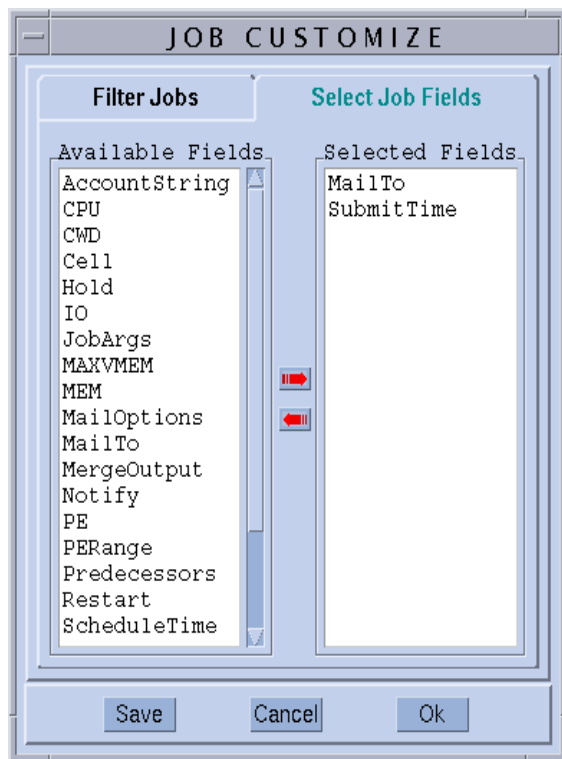


图 5-3 作业控制的自定义对话框

有了“自定义”对话框，就可以选择显示 Sun Grid Engine 作业对象的其它项，并且可以根据需要过滤作业。图 5-3 中的示例选择了附加字段“收件人”和“提交时间”。

图 5-4 中的“作业控制”对话框显示了对“已完成的作业”列表进行自定义后的扩充界面。

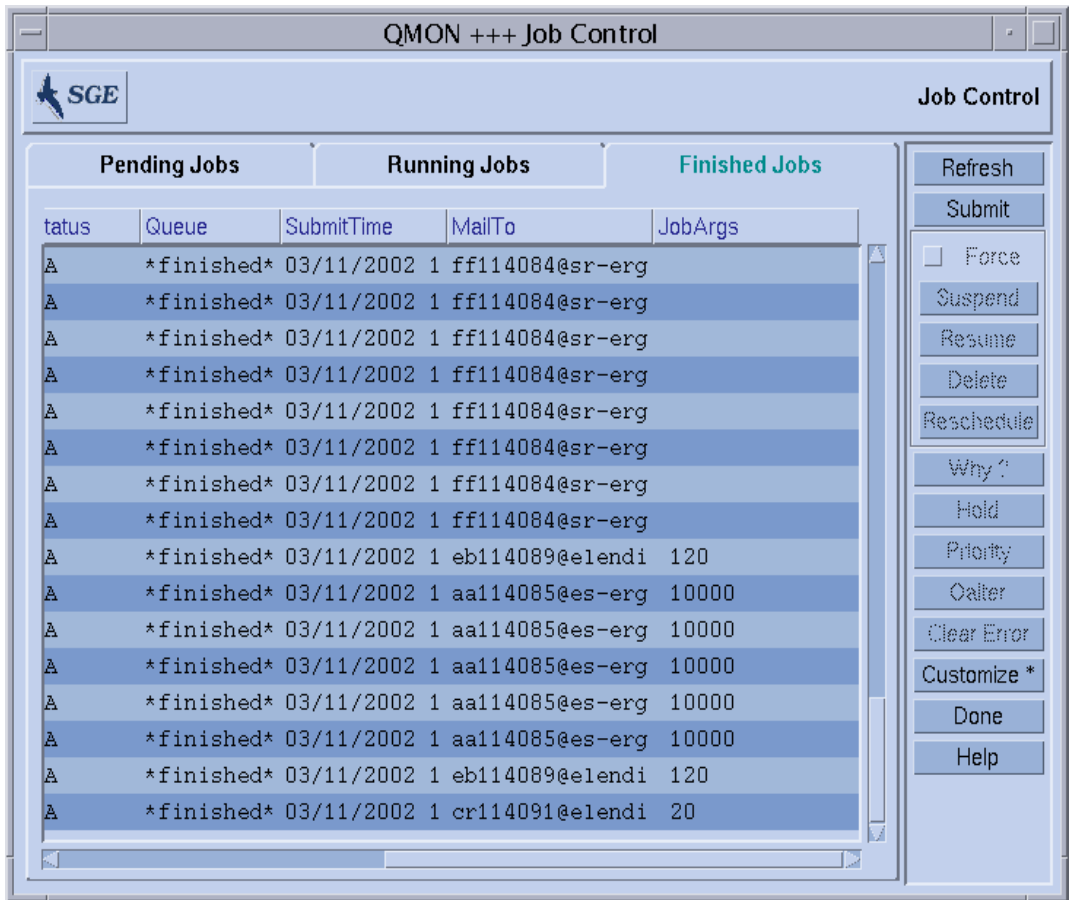


图 5-4 “作业控制”的“已完成的作业”对话框 — 已扩充

图 5-5 中过滤工具的示例仅选择了那些为 `ferst1` 所拥有并且运行或适合于体系结构 `solaris64` 的作业。

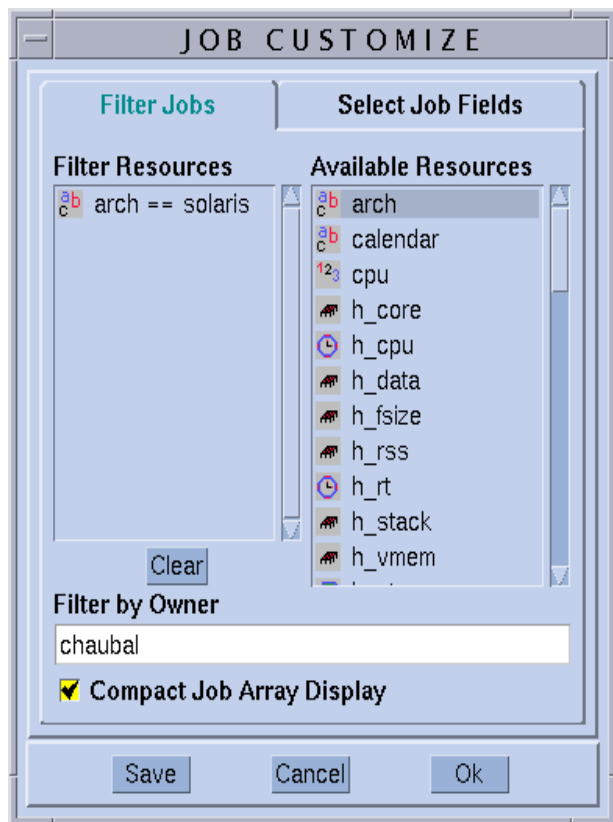


图 5-5 作业控制过滤

图 5-6 是过滤后的“作业控制”对话框，其中显示正运行的作业。

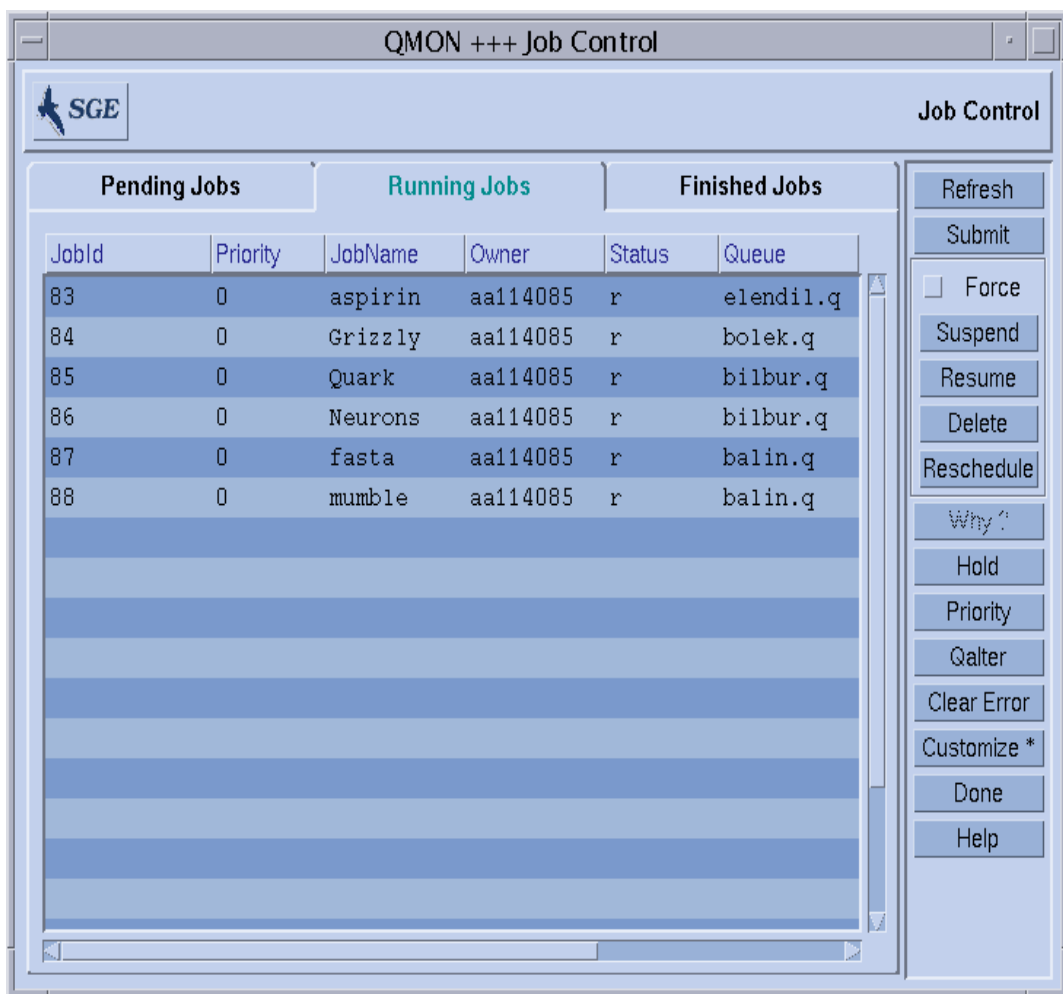


图 5-6 作业控制对话框 — 过滤后

注意 – 例如，图 5-3 中的“自定义”对话框内显示的保存按钮可以将自定义内容保存至用户主目录下的 `.qmon_preferences` 文件中，从而重新定义“作业控制”对话框的缺省界面。

图 5-6 中的“作业控制”对话框也例示了阵列作业在 QMON 中的显示方式。

可以用下列鼠标 / 键盘组合方式选择作业（以便进一步操作）：

- 按住 Control 键的同时用鼠标左键单击某作业可开始多个作业的选择。
- 按住 Shift 键的同时用鼠标左键单击另一个作业，可选择从开始选择的作业到当前作业之间的所有作业。
- 按住 Control 键的同时用鼠标左键单击作业可切换单个作业的选择状态。

选中的作业可通过屏幕右边的相应按钮暂停、恢复（取消暂停）、删除、阻止（并释放）、重定优先级和修改 (Qalter) 等操作。

诸如暂停、取消暂停、删除、等候、修改优先权和修改作业之类的操作，只能由作业拥有者或 Sun Grid Engine 管理人员以及操作人员（请参见第 61 页的“管理人员、操作人员 and 拥有者”）进行。只能暂停 / 恢复正运行的作业，而且只能阻止和修改暂挂的作业（在优先级和其它属性中）。

暂停作业相当于用 UNIX kill 命令向作业进程组发出 SIGSTOP 信号，该命令将中止作业使其不再占用 CPU 时间。取消暂停作业将发出 SIGCONT 信号，由此恢复作业（有关更多信号进程的信息，参见系统手册页 kill）。

注意 – 暂停、取消暂停和删除操作均可以强制执行；即，向 sge_qmaster 注册，而无需通知控制作业的 sge_execd，以防无法访问 sge_execd（例如，由于网络故障）。为此，可使用 Force 标志。

若对选定的暂挂作业使用等候按钮，则打开“设置等候”子对话框（请参见图 5-7）。

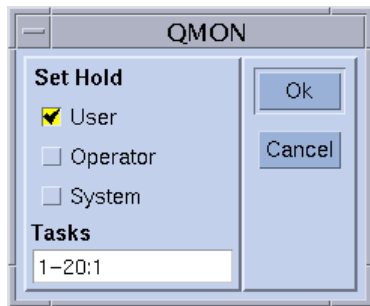


图 5-7 作业控制等候

“设置等候”子对话框用来设置和重新设置用户、系统和操作人员的等候。用户等候可由作业拥有者以及 Sun Grid Engine 操作人员和管理人员设置 / 重新设置。操作人员等候可由管理人员和操作人员设置 / 重新设置，而系统等候只能由管理人员设置 / 重新设置。只要给作业分配了任一种等候，作业就无法执行了。设置 / 重新设置等候的其它方法有：使用 qalter、qhold 和 qrls 命令（请参见《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中相应的项）。

若按下“优先级”按钮，则打开另一个子对话框（图 5-8），可在此输入选定暂挂作业的新优先级。在 Sun Grid Engine 中，优先级确定暂挂的作业列表中的作业顺序和暂挂作业在作业控制对话框中的显示顺序。用户只可将优先级设置在 0 到 -1024 范围内。Sun Grid Engine 操作人员和管理人员还可将优先级级别提高到最大值 1023（请参见第 93 页的“作业优先级”一节以获得有关作业优先级的细节）。

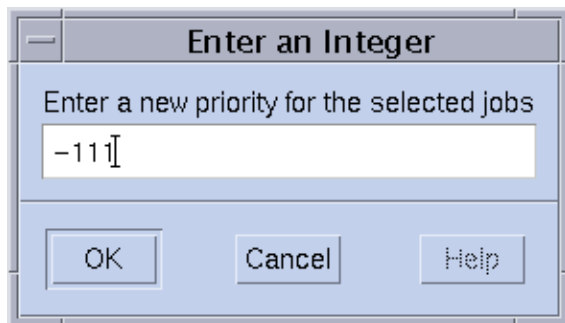


图 5-8 作业控制优先级定义

为暂挂作业按下 `Qalter` 按钮时，会出现第 65 页的“如何从图形用户界面 `QMON` 提交作业”中描述的“作业提交”屏幕，对话框中所有的项会根据作业提交时定义的属性设置。那些不能更改的项被设置为无效。其它项可以编辑，按下“作业提交”对话框中的 `Qalter` 按钮（代替“提交”按钮），即可向 Sun Grid Engine 注册这些更改。

在 `Qalter` 模式中使用“作业提交”屏幕中的“验证”标志时，有特殊含义。您可以检查暂挂的作业的一致性，并调查它们为何还未调度。只需为“验证”标志选择想要的一致性检查模式并按下 `Qalter` 按钮。根据选定的检查模式，系统将在不一致时显示警告。有关更多信息，请参见第 74 页的“高级示例”和 `qalter` 手册页的 `-w` 选项。

检查作业为何仍处于暂挂状态的另一方法是在“作业控制”对话框中先选择作业，然后单击原因按钮。此操作将打开“对象浏览器”对话框并列出最近一次阻碍 Sun Grid Engine 调度程序分配作业的原因。显示此类消息的浏览器屏幕例子如图 5-9 中所示。

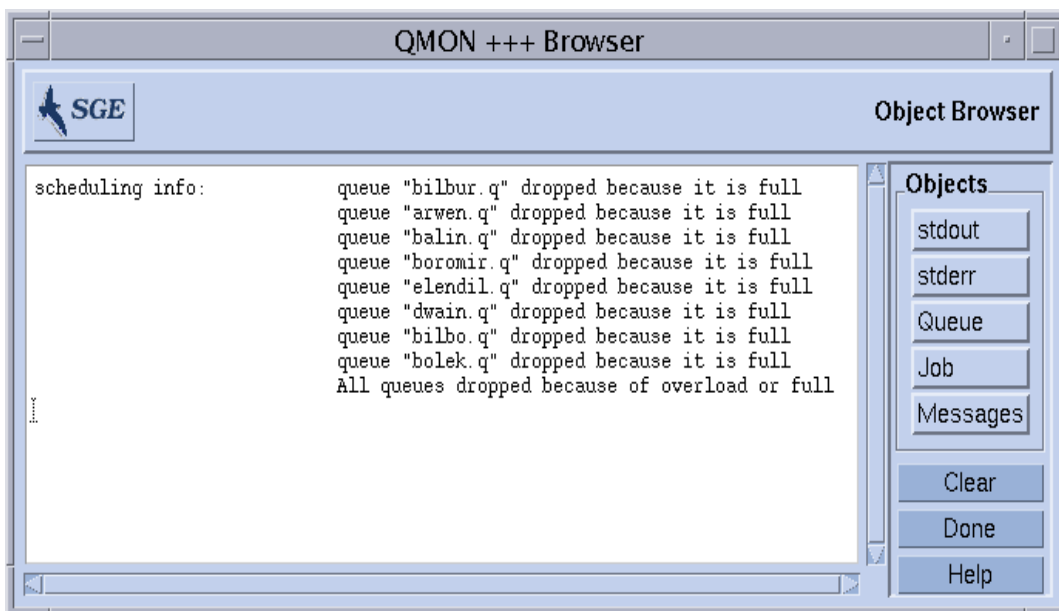


图 5-9 显示调度信息的浏览器

注意 – 若调度程序配置参数 `schedd_job_info` 设置为 `true`，则原因按钮仅提供有意义的输出（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `sched_conf`）。显示的调度程序信息与上一个调度时间间隔相关。在您调查作业为何还未调度时，此信息可能已不准确。

“清除错误”按钮可用于删除所选暂挂作业的错误状态，此作业之前曾尝试启动，后来由于作业相关问题而失败了（例如，对指定的作业输出文件无足够的写权限）。

注意 – 错误状态用红色字体显示在暂挂的作业列表中，只有在纠正错误条件后才能删除；例如，通过 `qalter` 纠正。若作业请求在中止时发送电子邮件，则错误情况将通过电子邮件自动报告（例如，通过 `qsub -m a` 选项）。

为确保总是显示最新的信息，QMON 使用巡回检测方案，从 `sge_qmaster` 检索作业状态。按“刷新”按钮可强行更新。

最后，此按钮还提供到“QMON 作业提交”对话框的链接（请参见图 5-10 中的示例）。

用 QMON 对象浏览器查看附加信息

“QMON 对象浏览器”可快速获取有关 Sun Grid Engine 作业的附加信息，而无需自定义“作业控制”对话框（如第 106 页的“如何用 QMON 监视和控制作业”中所述）。

按下 QMON 主菜单中的“浏览器”图标按钮即可打开“对象浏览器”。若选中浏览器中的“作业”按钮并且鼠标指针移至“作业控制”对话框（请参见图 5-2 中的示例）中的作业行上，浏览器将显示有关 Sun Grid Engine 作业的信息。

图 5-10 中的浏览器屏幕给出了在这种情况下显示的信息的示例。

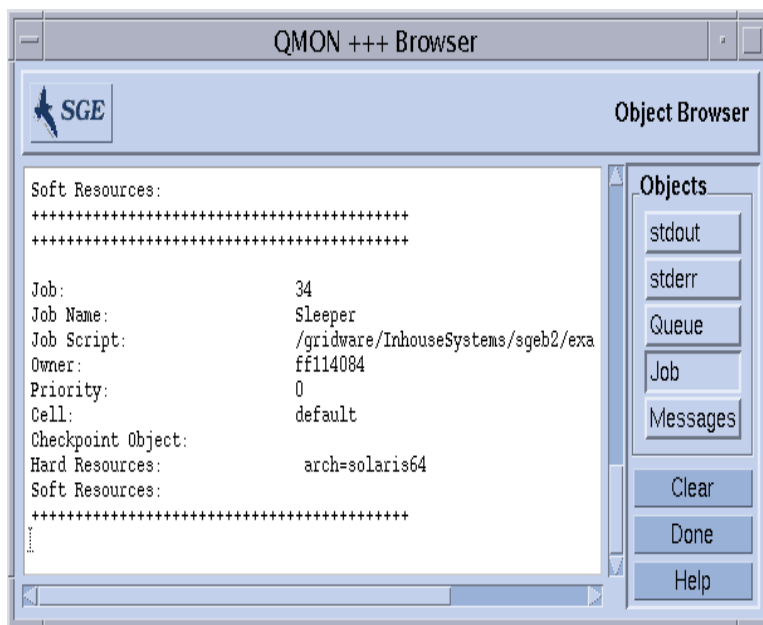


图 5-10 对象浏览器 — 作业

▼ 如何用 qstat 监视作业

- 根据以下各节的详细信息，在命令行中使用以下命令之一。

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

第一张表单仅提供了已提交作业的概述（请参见表 5-1）。第二张表单另外包含了有关当前配置队列的信息（请参见表 5-2）。

第一张表单中，标题行指明每一栏的含义。大部分栏的意图不言自明。不过，state 栏包含的单个字符代码含义如下：r 表示正运行，s 表示已暂停，q 表示已排队，w 表示在等待（请参见《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中的 qstat 项，可获得 qstat 输出格式的详细说明）。

第二张表单分为两个部分，第一部分显示所有可用队列的状态，第二部分（以 - 暂挂作业 - ... 分隔符为标题）显示 sge_qmaster 作业池区域的状态。队列部分的第一行定义了与所列队列相关的每栏的含义。队列以水平线分隔。若作业进入某队列，它们将显示在相关队列之下，其格式与第一张表单中 qstat 命令的输出相同。第二个输出部分中的暂挂的作业格式也与 qstat 的第一张表单相同。

队列描述中的以下栏需要更多解释。

- qtype - 队列类型，B（批处理）、I（交互式）、P（并行）和 C（点检查）中的一个或其任意组合。
- used/free - 队列中已使用的 / 空闲的作业位置数。
- states - 队列状态，u（未知）、a（警报）、s（暂停）、d（禁用）和 E（错误）中的一个或其任意组合。

同样，qstat 手册页含有 qstat 输出格式的更详细的描述。

qstat 命令的各种附加选项在两个版本中都能改善功能。-r 选项可用于显示已提交作业的资源需求。此外，输出可限于某个用户或特定队列，而 -l 选项可用于指定资源需求，如第 78 页的“资源需求定义”中有关 qsub 命令的描述。若使用资源需求，只有那些满足 qstat 命令行中资源需求说明的队列（以及在这些队列中运行的作业）才会显示。

表 5-1 qstat 输出示例

| 作业 ID | 优先级 | 名称 | 用户 | 状态 | 提交 / 启动时间 | 队列 | 职能 |
|-------|-----|-----------|---------|----|----------------------|---------|----|
| 231 | 0 | hydra | craig | r | 07/13/96 20:27:15 | durin.q | 主控 |
| 232 | 0 | compile | penny | r | 07/13/96 20:30:40 | durin.q | 主控 |
| 230 | 0 | blackhole | don | r | 07/13/96 20:26:10 | dwain.q | 主控 |
| 233 | 0 | mac | elaine | r | 07/13/96 20:30:40 | dwain.q | 主控 |
| 234 | 0 | golf | shannon | r | 07/13/96 20:31:44 | dwain.q | 主控 |
| 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 输出示例

| 队列名 | 队列类型 | 已使用的 / 空闲的 | 平均负荷 | 体系结构 | 状态 |
|---------|-------------|------------|-------|----------|-------------|
| 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 主控 |
| 232 | 0 compile | penny | r | 07/13/96 | 20:30:40 主控 |
| dwain.q | BIP | 3/3 | 0.36 | sun4 | |
| 230 | 0 blackhole | don | r | 07/13/96 | 20:26:10 主控 |
| 233 | 0 mac | elaine | r | 07/13/96 | 20:30:40 主控 |
| 234 | 0 golf | shannon | r | 07/13/96 | 20:31:44 主控 |

表 5-2 qstat -f 输出示例 (接上页)

| | | | | |
|--------------------------------------|----------|--------|------|-------------------|
| fq | BIP | 0/3 | 0.36 | sun4 |
| ##### | | | | |
| - 暂挂作业 - 暂挂作业 - 暂挂作业 - 暂挂作业 - 暂挂作业 - | | | | |
| ##### | | | | |
| 236 | 5 word | elaine | qw | 07/13/96 20:32:07 |
| 235 | 0 andrun | penny | qw | 07/13/96 20:31:43 |

▼ 如何用电子邮件监视作业

- 根据以下各节的详细信息，在命令行中输入以下带有相应自变量的命令。

```
% qsub 自变量
```

qsub -m 开关选项请求在发生某些事件时将电子邮件发送到提交作业的用户，或者发送到由 -M 标志指定的电子邮件地址（有关标志的描述，参见 qsub 手册页）。-m 选项的自变量指定事件。有以下自变量供选择：

- b - 作业开始时发电子邮件。
- e - 作业结束时发送电子邮件。
- a - 作业中止时发送电子邮件（例如，被 qdel 命令中止）。
- s - 作业暂停时发送邮件。
- n - 不发送邮件（缺省值）。

一个 -m 选项可以选择多个上述自变量，自变量之间用逗号分隔即可。

同样的电子邮件事件可借助于“QMON 作业提交”对话框来配置。请参见第 74 页的“高级示例”一节。

从命令行控制 Sun Grid Engine 作业

第 106 页的“如何用 QMON 监视和控制作业”一节解释 Sun Grid Engine 如何用 Sun Grid Engine 图形用户界面 QMON 删除、暂停和恢复作业。

如本节所述，相同的功能也可从命令行获得。

▼ 如何从命令行控制作业

- 根据以下各节的详细信息，在命令行中输入以下命令之一及其相应自变量。

```
% qdel 自变量  
% qmod 自变量
```

可以使用 `qdel` 命令取消 Sun Grid Engine 作业，无论它们是正在运行还是处于假脱机状态。`qmod` 命令可以暂停和取消暂停（恢复）已经在运行的作业。

使用这两个命令都需要知道作业标识号，此标识号可由 `qsub` 命令得到。若忘了标识号，可通过 `qstat` 检索（请参见第 116 页的“如何用 `qstat` 监视作业”）。

以下为两个命令的几个示例：

```
% qdel 作业 ID  
% qdel -f 作业 ID 1, 作业 ID 2  
% qmod -s 作业 ID  
% qmod -us -f 作业 ID 1, 作业 ID 2  
% qmod -s 作业 ID. 任务 ID 范围
```

要删除、暂停或取消暂停一项作业，您必须是作业所有者、Sun Grid Engine 管理人员或操作人员（请参见第 61 页的“管理人员、操作人员和所有者”）。

对于这两个命令，都可以用 `-f` 强制选项在 `sge_qmaster` 中注册作业的状态更改，而不用联系 `sge_execd`，以防 `sge_execd` 无法访问（如网络故障）。`-f` 选项应由管理员使用。不过，如果群集配置中的 `qmaster_params` 项设置了标志 `ENABLE_FORCED_QDEL`，用户可使用 `qdel` 命令强制删除自己的作业（有关更多信息，请参见《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中的 `sge_conf` 手册页）。

作业从属性

建立一个复杂任务的最方便的方法常常是将任务划分成子任务。在这些情况下，子任务的启动依赖于其它子任务的顺利完成。例如，前导任务产生一个输出文件，此文件必须由后续任务读取和处理。

Sun Grid Engine 的作业从属性功能可支持互相依赖的任务。作业可被配置为依赖于一个或多个其它作业的成功完成。该功能可由 `qsub -hold_jid` 选项实施。可以指定所提交作业要依赖的作业的列表。作业列表也可包括阵列作业的子集。除非是从属性列表中的所有作业已成功完成，否则，提交的作业无法执行。

控制队列

如第 50 页的“队列和队列特性”中所述，队列的拥有者有权暂停 / 取消暂停或禁用 / 启用队列。若用户不时需要某些计算机来完成重要的工作，而这些机器受在后台运行的 Sun Grid Engine 作业的影响很大，则需要此权限。

暂停或启用队列的方法有两种。

- “QMON 队列控制”对话框
- `qmod` 命令

▼ 如何用 QMON 控制队列

- 在 QMON 主菜单中，单击“队列控制”按钮。

随即出现类似于图 5-11 的“队列控制”对话框。

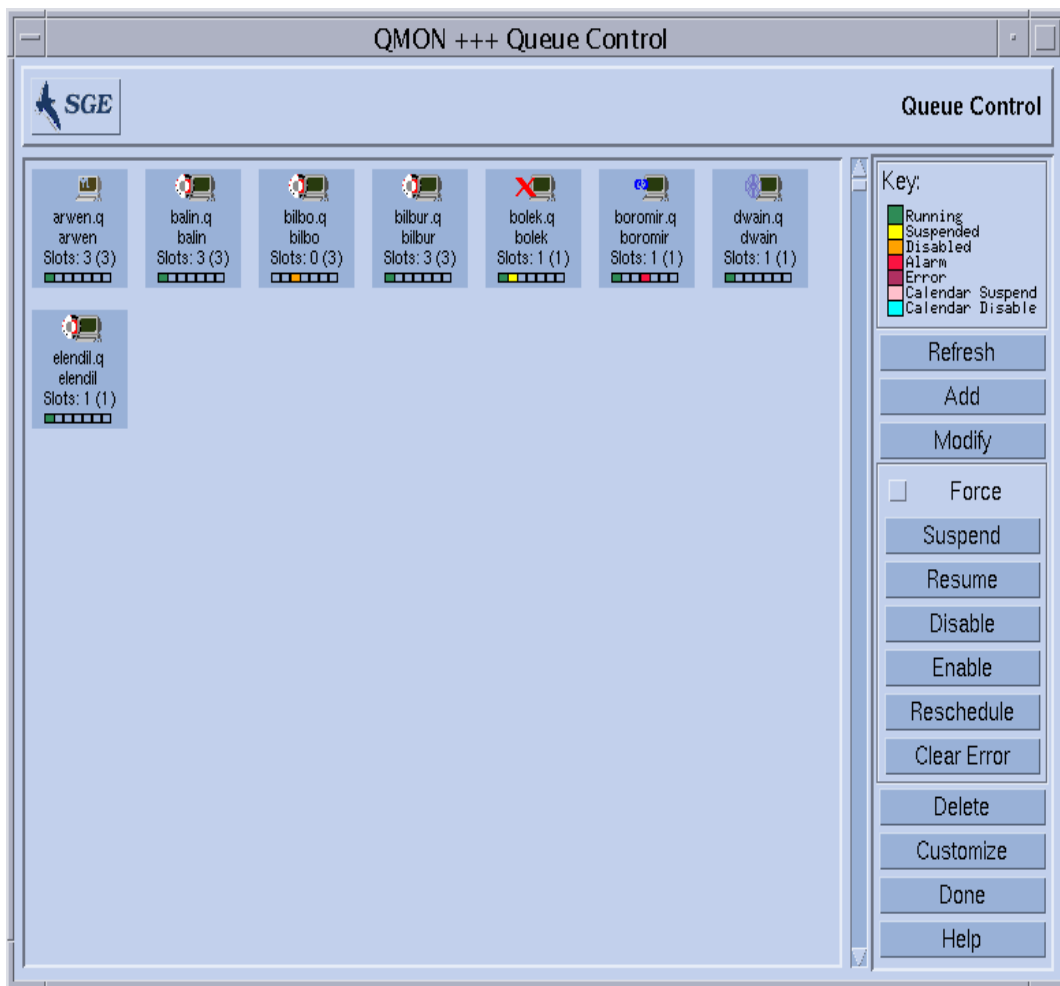


图 5-11 “队列控制”对话框

“队列控制”对话框的用途是提供可用资源和群集活动的简明概述。也提供暂停 / 取消暂停和禁用 / 启用队列以及配置队列的方法。显示的每一个图标代表一个队列。若主显示区为空，则未配置队列。每个队列图标上标有队列名、队列所在主机的名称以及占用的作业位置数。若 `sgc_execd` 在队列主机上运行，并且已经向 `sgc_qmaster` 注册，则队列图标上的图形会指明队列主机的操作系统体系结构，图标底部的彩色条会指明队列的状态。“队列控制”对话框右边的图例显示颜色的含义。

对于这些队列，通过按住 **Shift** 键的同时用鼠标左键单击队列图标，用户可检索到队列的当前属性、负荷和资源使用信息，以及队列所在主机的隐含信息。这样将弹出类似于图 5-12 的信息屏幕。

可通过单击鼠标左键或用围绕队列图标按钮的矩形区域来选中队列。“删除”、“暂停”、“取消暂停”、“禁用”或“启用”按钮都可用于对选定队列执行相应操作。暂停 / 取消暂停和禁用 / 启用操作都需要相应 `sgc_execd` 的通知。若无法通知（如，由于主机关机），倘若“强制”开关打开，则可强制进行 `sgc_qmaster` 的内部状态更改。

若暂停队列，队列将对后续作业关闭，已经在队列中执行的作业也将暂停，如第 106 页的“如何用 QMON 监视和控制作业”中所述。一取消暂停，队列及其作业将立刻恢复。

注意 – 若暂停队列中的作业已经另行明令暂停，则队列取消暂停时作业也不会恢复。该作业需要再次明令取消暂停。

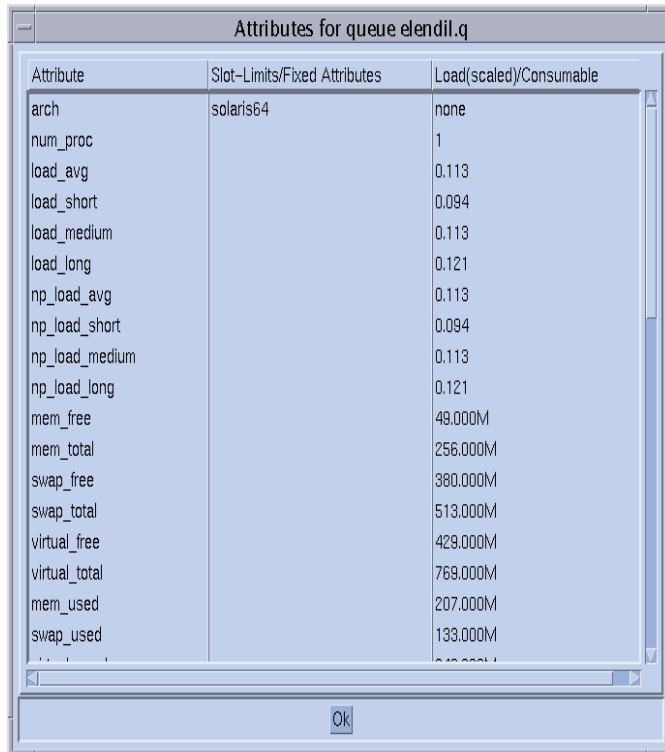
禁用的队列将关闭，不过，在那些队列中执行的作业可以继续。禁用队列常用于“忽略”一个队列。队列启用后，又可执行作业。不会对仍在执行的作业执行任何操作。

暂停 / 取消暂停和禁用 / 启用操作需要队列所有者或 Sun Grid Engine 管理人员或操作人员权限（请参见第 61 页的“管理人员、操作人员和所有者”）。

“队列控制”对话框中显示的信息会定期更新。按“刷新”按钮可强制更新。“完成”按钮将关闭对话框。

使用“自定义”按钮可通过过滤操作选择要显示的队列。图 5-13 中的示例屏幕只显示那些运行在属于体系结构 `osf4`（如 4 版的 Compaq UNIX）的主机上的队列。“自定义”对话框中的“保存”按钮可用来将您的设定存储到主目录中的 `.qmon_preferences` 文件中，以备在稍后调用 QMON 时按标准重激活。

要配置队列，按下“队列控制”屏幕右边的“添加”或“修改”按钮，将打开一个子对话框（有关更多信息，请参见第 150 页的“如何用 QMON 配置队列”）。



| Attribute | Slot-Limits/Fixed Attributes | Load(scaled)/Consumable |
|----------------|------------------------------|-------------------------|
| arch | solaris64 | none |
| num_proc | | 1 |
| load_avg | | 0.113 |
| load_short | | 0.094 |
| load_medium | | 0.113 |
| load_long | | 0.121 |
| np_load_avg | | 0.113 |
| np_load_short | | 0.094 |
| np_load_medium | | 0.113 |
| np_load_long | | 0.121 |
| mem_free | | 49.000M |
| mem_total | | 256.000M |
| swap_free | | 380.000M |
| swap_total | | 513.000M |
| virtual_free | | 429.000M |
| virtual_total | | 769.000M |
| mem_used | | 207.000M |
| swap_used | | 133.000M |

图 5-12 队列属性显示

属于队列的所有属性（包括那些从主机或群集继承的属性）都在“属性”栏列出。“位置限额 / 固定属性”栏显示那些被定义为与队列位置限制相关或固定属性组属性的属性值。“负荷（已调节） / 可使用量”栏显示的信息涉及：报告的（若配置为已调节）负荷参数（请参见第 189 页的“负荷参数”）和基于 Sun Grid Engine 可使用资源工具的可用资源容量（请参见第 177 页的“可使用的资源”）。

注意 – 若负荷属性配置为可使用资源，则负荷报告和可使用容量可能会互相覆盖。将显示两者的较小值，作业分配算法中使用的就是此值。

注意 – 显示的负荷和当前可使用值没有考虑负荷调整更正，如第 25 页的“执行主机”中所述。

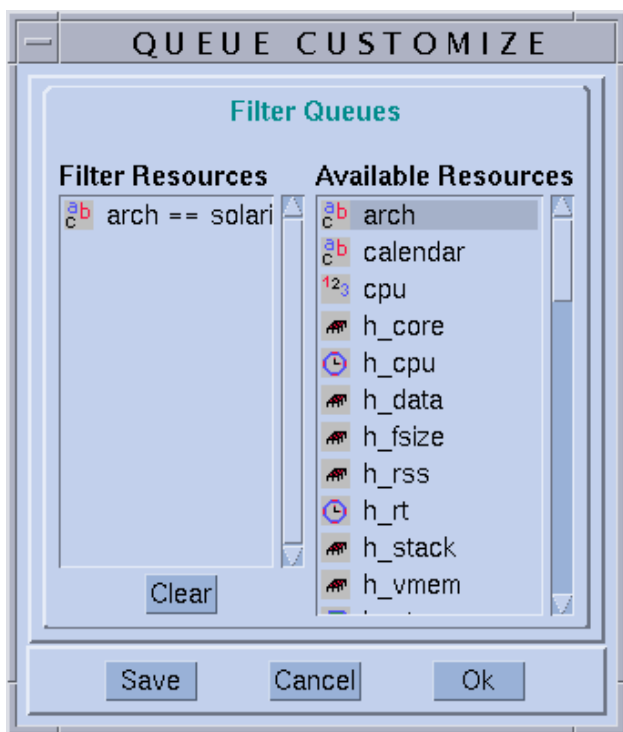


图 5-13 队列控制自定义

▼ 如何用 qmod 控制队列

第 119 页的“如何从命令行控制作业”一节中介绍如何用 Sun Grid Engine 命令 qmod 暂停 / 取消暂停 Sun Grid Engine 作业。不过，qmod 命令还可为用户提供其它暂停 / 取消暂停或禁用 / 启用队列的方法。

- 根据以下各节的详细信息，输入以下带有相应自变量的命令。

```
% qmod 自变量
```


以下命令为 qmod 如何用于此目的的示例：

```
% qmod -s 队列名
% qmod -us -f 队列名1、队列名2
% qmod -d 队列名
% qmod -e 队列名1、队列名2、队列名3
```

前两条命令分别暂停或取消暂停队列，而第 3 条和第 4 条命令禁用和启用队列。另外，第二条命令使用 qmod -f 选项强制在 sge_qmaster 中注册状态的更改，以防 sge_execd 无法访问（如由于网络故障）。

注意 – 暂停 / 取消暂停和禁用 / 启用队列都需要所有者、Sun Grid Engine 管理人员或操作人员权限（请参见第 61 页的“管理人员、操作人员和所有者”）。

注意 – 您可以对 crontab 或 at 作业使用 qmod 命令。

自定义 QMON

QMON 的外观主要由专门设计的资源文件定义。已应用了合理的缺省值，其样本资源文件位于 <sges 根目录>/qmon/Qmon 下。

群集管理者可以将站点专用的缺省值安装在标准位置，如 /usr/lib/X11/app-defaults/Qmon，方法是：将 QMON 专用的资源定义放入标准的 .Xdefaults 或 .Xresources 文件中，或将站点专用的 Qmon 文件放入标准搜索路径（如 XAPPLRESDIR）引用的位置。若您遇到上述相关情况，请咨询管理员。

除此之外，用户还可以配置个人首选项，方法是：将 Qmon 文件复制到主目录（或私人搜索路径 XAPPLRESDIR 指向的另一个位置）内并对其进行修改，或者将必要的资源定义包含到用户的私人 .Xdefaults 或 .Xresources 文件。也可以在操作或启动 X11 环境（如在 .xinitrc 资源文件中）时，使用 xrdb 命令安装私人 Qmon 资源文件。

有关可能的自定义的详细信息，请参见样本文件 Qmon 内的注释行。

图 5-2 和图 5-13 显示的“作业控制”和“队列控制自定义”对话框解释了自定义 QMON 的其它方法。不论在哪个对话框，都可以使用“保存”按钮，将用自定义对话框配置的过滤和显示定义存储到位于用户主目录的 `.qmon_preferences` 文件中。一旦重新启动，QMON 将读取此文件，并重新激活先前已定义的方式。

第四部分 管理

《Sun Grid Engine 5.3 管理和用户指南》的这一部分适用于管理员，共包括六章。

- **第六章 – 第 129 页的 “主机和群集配置”**
本章提供配置 Sun Grid Engine 5.3 主机和群集的一般背景和详细指导。
- **第七章 – 第 149 页的 “配置队列和队列日历”**
本章描述重要概念*队列*。它相当于不同种类的 Sun Grid Engine 5.3 作业的 “容器”。本章还包含了配置队列的详尽指导。
- **第八章 – 第 167 页的 “属性组概念”**
本章介绍 Sun Grid Engine 5.3 系统如何使用*属性组*来定义用户为作业请求的资源属性的所有相关信息。管理员可配置各种属性组以匹配环境需求，本章提供完成此任务的详细指导。
- **第九章 – 第 195 页的 “管理用户访问权限和策略”**
本章提供了有关可从 Sun Grid Engine 5.3 系统获得的用户策略类型的全面背景信息，并提供如何将~~这些策略~~与计算环境匹配的~~指导~~。
- **第十章 – 第 223 页的 “管理并行环境”**
除了描述 Sun Grid Engine 5.3 系统如何适应*并行环境*外，本章还提供如何运用它们的详尽配置指导。
- **第十一章 – 第 233 页的 “错误消息”**
本章描述检索错误消息的 Sun Grid Engine 5.3 过程，并描述如何以调试模式运行该软件。

主机和群集配置

本章提供配置 Sun Grid Engine 5.3 系统各个方面的背景信息和指导。在本章中可找到关于以下任务的指导。

- 第 132 页的 “如何用 QMON 配置管理主机”
- 第 134 页的 “如何删除管理主机”
- 第 134 页的 “如何添加管理主机”
- 第 134 页的 “如何从命令行配置管理主机”
- 第 134 页的 “如何用 QMON 配置提交主机”
- 第 136 页的 “如何删除提交主机”
- 第 136 页的 “如何添加提交主机”
- 第 136 页的 “如何从命令行配置提交主机”
- 第 136 页的 “如何用 QMON 配置执行主机”
- 第 138 页的 “如何删除执行主机”
- 第 138 页的 “如何关闭执行主机守护程序”
- 第 138 页的 “如何添加或修改执行主机”
- 第 142 页的 “如何从命令行配置执行主机”
- 第 143 页的 “如何用 qhost 监视执行主机”
- 第 143 页的 “如何从命令行中止守护程序”
- 第 144 页的 “如何从命令行重新启动守护程序”
- 第 145 页的 “如何从命令行显示基本群集配置”
- 第 145 页的 “如何从命令行修改基本群集配置”
- 第 146 页的 “如何用 QMON 显示群集配置”
- 第 146 页的 “如何用 QMON 删除群集配置”
- 第 147 页的 “如何用 QMON 显示全局群集配置”
- 第 147 页的 “如何使用 QMON 修改全局配置和主机配置”

关于主控和影像主控配置

影像主控主机名文件 `<sg_e 根目录>/<单元>/common/shadow_masters` 包含主控主机（Sun Grid Engine 主控守护程序 `sg_e_qmaster` 最初在其上运行）和影像主控主机的名称。主控主机名文件的格式如下。

- 文件第一行定义主要主控主机
- 第二行开始逐行指定影像主控主机

（影像）主控主机出现的顺序很重要。如果主要主控主机（文件中的第一行）无法继续执行，则第二行定义的影像主机将接替它。若这个主机也无法继续，则第三行定义的主机将接替它，依此类推。

要将一台主机充当 Sun Grid Engine 影像主控主机，则必须满足以下要求：

- 影像主控主机需要运行 `sg_e_shadowd`。
- 影像主控主机要共享记录于磁盘中的 `sg_e_qmaster` 的状态信息、作业和队列信息。尤其是，（影像）主控主机需要对主控主机的假脱机目录和 `<sg_e 根目录>/<单元>/common` 目录的读/写 `root` 访问权限。
- 影像主控主机的主机名文件必须包含将主机定义为影像主控主机的行。

这些需求一经满足，这台主机就激活了影像主控主机功能。不必重新启动 Sun Grid Engine 守护程序就可激活此功能。

影像主控主机上 `sg_e_qmaster` 的自动故障转移启动需要一些时间（大约一分钟）。其间，如果执行 Sun Grid Engine 命令，您就会得到相应的错误消息。

注意 – 文件 `<sg_e 根目录>/<单元>/common/act_qmaster` 包含实际运行 `sg_e_qmaster` 守护程序的主机名。

要启动影像 `sg_e_qmaster`，Sun Grid Engine 必须确保 *旧的* `sg_e_qmaster` 已经停止，或者即将停止，且不会执行干扰刚启动的影像 `sg_e_qmaster` 的操作。这种问题极为常见。出现问题时，相应的错误消息将记录到影像主控主机上的 `sg_e_shadowd` 消息日志文件中（请参见第十一章第 233 页的“错误消息”），并且任何打开到 `sg_e_qmaster` 守护程序的 `tcp` 连接的尝试都会失败。若发生这种情况，确保没有主控守护程序在运行，并手动重新启动任一影像主控主机上的 `sg_e_qmaster`（请参见第 143 页的“如何从命令中止守护程序”一节）。

关于守护程序和主机

Sun Grid Engine 主机分为 4 组，分组取决于哪个守护程序在系统上运行以及主机如何注册到 `sge_qmaster`。

- **主控主机** – 主控主机是一切群集活动的中心。它运行主控守护程序 `sge_qmaster`。`sge_qmaster` 控制所有的 Sun Grid Engine 组件（如队列和作业），并维护关于组件状态和用户访问权限之类的表单。第 27 页的“如何安装主控主机”一节描述如何首次设置主控主机，第 130 页的“关于主控和影像主控配置”一节说明如何配置动态主控主机的更改。主控主机通常运行 Sun Grid Engine 调度程序 `sge_schedd`。除了在安装过程中执行的配置外，主控主机不需要进一步配置。
- **执行主机** – 执行主机是有权执行 Sun Grid Engine 作业的节点。因此，该主机上有 Sun Grid Engine 队列，并运行 Sun Grid Engine 执行守护程序 `sge_execd`。如第 28 页的“如何安装执行主机”一节中所述，执行主机最初是在该执行主机安装过程中设置的。
- **管理主机** – 可为主控主机以外的其它主机赋予权限，以在 Sun Grid Engine 中完成任何种类的管理活动。管理主机可用以下命令设置：

```
qconf -ah 主机名
```

有关详细信息，参见 `qconf` 手册页。

- **提交主机** – 提交主机仅允许提交和控制批处理作业。尤其是登录到提交主机的用户可通过 `qsub` 提交作业，可通过 `qstat` 或运行 Sun Grid Engine 的 OSF/1 Motif 图形用户界面 `QMON` 控制作业状态。提交主机可用以下命令设置：

```
qconf -as 主机名
```

有关详情，参见 `qconf` 手册页。

注意 – 主机可属于一个或多个上述类别。主控主机缺省情况下既是管理主机又是提交主机。

关于配置主机

Sun Grid Engine 维护除主控主机以外的所有类型主机的对象列表。对于管理主机和提交主机，这些列表只提供有关主机是否有管理或提交权限的信息。对于执行主机对象，将存储更多参数，例如，由在主机上运行的 `sge_execd` 所报告的负荷信息以及由 Sun Grid Engine 管理员提供的负荷参数调节系数。

以下各节介绍如何借助 Sun Grid Engine 图形用户界面 QMON 或从命令行配置不同的主机对象。

GUI 管理是由一组主机配置对话框提供的，按下 QMON 主菜单中的“主机配置”图标按钮即可调用它们。可用的对话框有“管理主机配置”（参见图 6-1）、“提交主机配置”（参见图 6-2）和“执行主机配置”（参见图 6-3）。对话框可通过屏幕顶部的选择列表按钮切换。

`qconf` 命令为主机对象管理提供命令行界面。

▼ 如何用 QMON 配置管理主机

1. 单击 QMON 主菜单顶部的“管理主机”选项卡。

随即打开类似于下图的“管理主机配置”对话框。

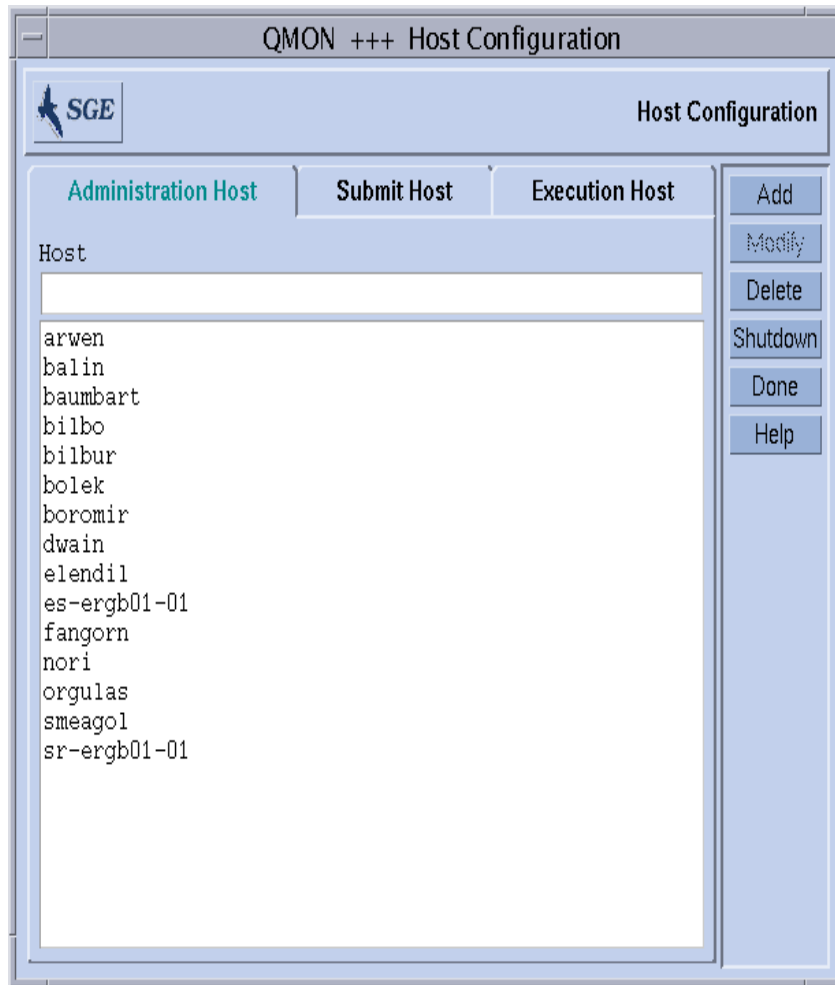


图 6-1 “管理主机配置”对话框

注意 – 第一次按“主机配置”按钮时，会缺省打开“管理主机配置”对话框。

2. 根据想要配置主机的方式，按照以下各节的指导继续。

用此对话框可以配置允许使用 Sun Grid Engine 的管理命令的主机。屏幕中间列出的选项列表显示了已经声明为提供管理权限的主机。

▼ 如何删除管理主机

- 用鼠标左键单击某个现存主机的名称，然后按下对话框底部的“删除”按钮，即可将该主机从列表中删除。

▼ 如何添加管理主机

- 在“主机名”输入窗口中输入新主机的名称，再按“添加”按钮或按回车键，即可添加一台新主机。

▼ 如何从命令行配置管理主机

- 根据想要配置主机的方式，输入以下命令及其相应自变量。

```
% qconf 自变量
```

qconf 命令的自变量及其使用结果如下。

- qconf -ah *主机名*
添加管理主机 — 将指定主机添至管理主机列表。
- qconf -dh *主机名*
删除管理主机 — 将指定主机从管理主机列表中删除。
- qconf -sh
显示管理主机 — 显示所有当前已配置的管理主机的列表。

▼ 如何用 QMON 配置提交主机

1. 单击 QMON 主菜单顶部的“提交主机”选项卡。

随即打开类似于下图的“提交主机配置”对话框。

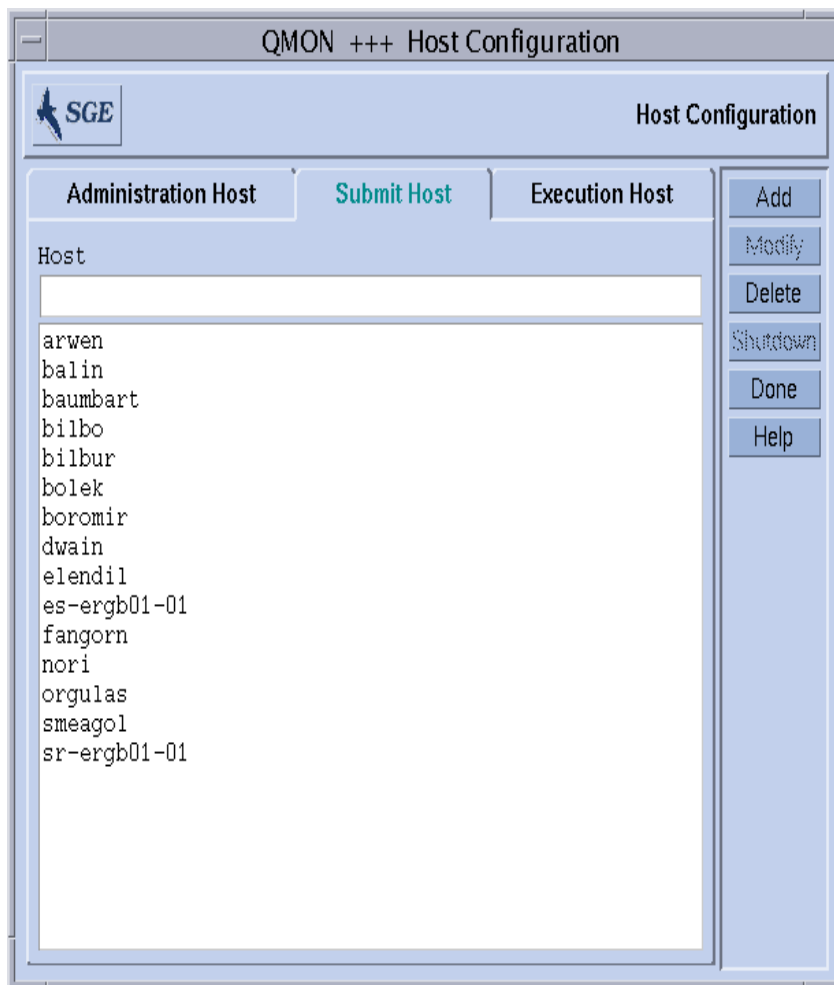


图 6-2 提交主机配置

2. 根据想要配置主机的方式，按照以下各节的指导继续。

使用此对话框，可以声明能从中提交、监视和控制作业的主机。除非还声明为管理主机，否则不允许这些主机使用 Sun Grid Engine 的管理命令（请参见第 132 页的“如何用 QMON 配置管理主机”）。屏幕中间列出的选项列表显示已经声明为提供提交权限的主机。

▼ 如何删除提交主机

- 用鼠标左键单击“提交主机”对话框中某个现存主机的名称，然后按下该对话框底部的“删除”按钮，即可从列表中删除该主机。

▼ 如何添加提交主机

- 在“提交主机”对话框的“主机名”输入窗口中输入主机的名称，再按“添加”按钮或按回车键，即可添加主机。

▼ 如何从命令行配置提交主机

- 根据想要配置主机的方式，输入以下命令及其相应自变量。

```
% qconf 自变量
```

qconf 命令的自变量及其使用结果如下。

- qconf -as *主机名*
添加提交主机 — 将指定主机添至提交主机列表。
- qconf -ds *主机名*
删除提交主机 — 将指定主机从提交主机列表中删除。
- qconf -ss
显示提交主机 — 显示所有当前配置为提供提交权限的主机列表。

▼ 如何用 QMON 配置执行主机

1. 单击 QMON 主菜单顶部的“执行主机”选项卡。
随即打开类似于下图的“执行主机配置”对话框。

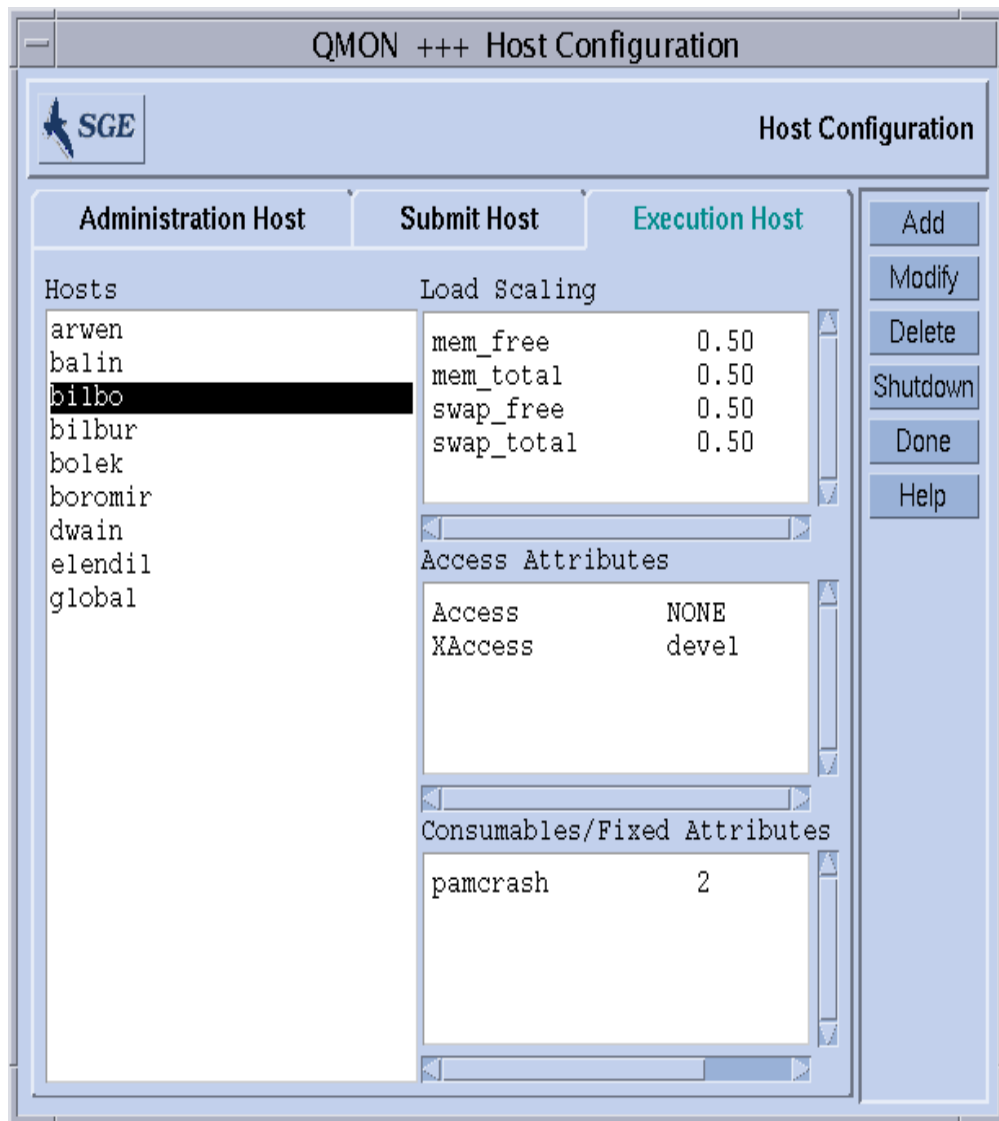


图 6-3 执行主机配置

2. 根据想要配置主机的方式，按照以下各节的指导继续。

Sun Grid Engine 执行主机可从此对话框中配置。除非声明为管理主机或提交主机，否则不会自动允许这些主机使用管理或提交命令（请参见第 132 页的“如何用 QMON 配置管理主机”和第 134 页的“如何用 QMON 配置提交主机”）。

主机选择列表将自动显示已经定义的执行主机。与所选执行主机相关的当前配置的负荷调节系数、访问权限，和可使用的以及固定的属性组属性的资源可用性将分别显示在该主机的“负荷调节”、“访问属性”和“可使用 / 固定属性”窗口中。有关属性组属性、用户访问权限和负荷参数的细节，请参见第 167 页的“关于属性组”、第 60 页的“用户访问权限”和第 189 页的“负荷参数”。

▼ 如何删除执行主机

- 在“执行主机”对话框中，单击要删除的执行主机名，再按对话框右边按钮栏中的“删除”按钮。

▼ 如何关闭执行主机守护程序

- 对于任何选定主机，可按“执行主机”对话框中的“关闭”按钮。

▼ 如何添加或修改执行主机

1. 按“执行主机”对话框中的“添加”或“修改”按钮。

将显示与图 6-4 相似的对话框。

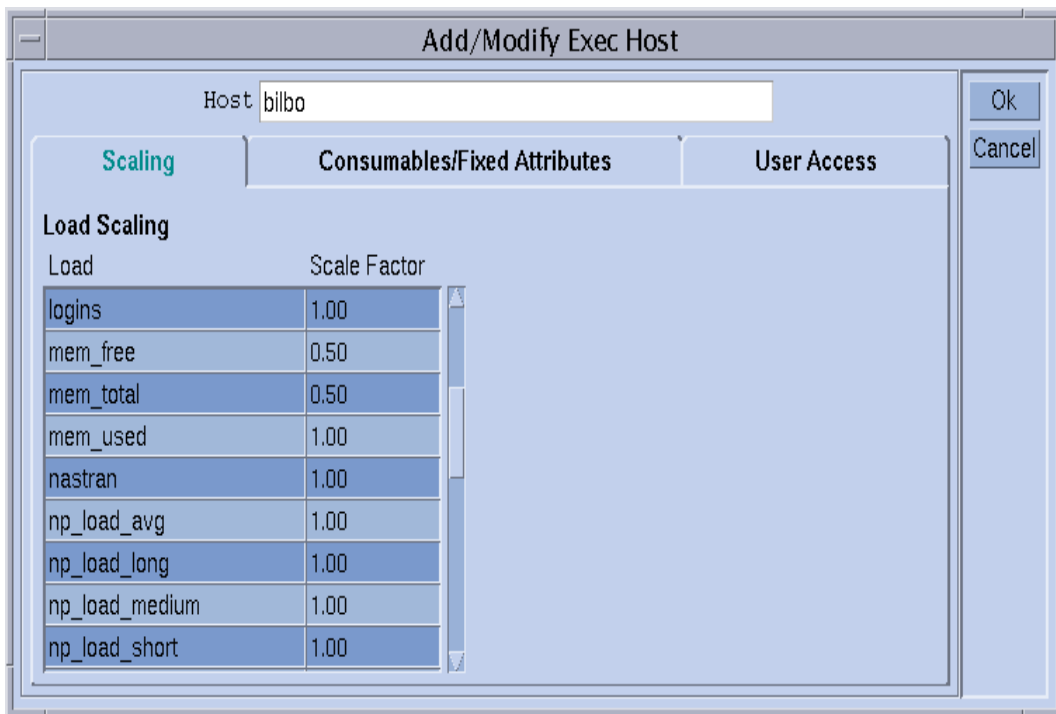


图 6-4 修改负荷调节

2. 根据想要修改主机的方式，按照以下各节的指导继续。

可在用于添加新执行主机或修改已有主机的配置的对话框中，修改与主机相关的所有属性。执行主机名将显示在或可添加到“主机”输入窗口中。可通过选择对话框中的“调节”选项卡定义调节系数（请参见图 6-4）。

所有可用负荷参数将显示在“负荷调节”表的“负荷”栏中，相应的调节定义可在“调节系数”栏中找到。可以编辑“调节系数”栏。有效的调节系数为正的浮点数，可用定点或科学计数法表示。

当选中“可使用/固定属性”选项卡时，可以定义与主机相关的属性组属性（请参见图 6-5）。与主机相关的属性组（请参见第 167 页的“关于属性组”）为全局和主机属性组或管理员定义的属性组，后者是通过对话框左下部的“属性组选项”区附加到主机的。可用的管理员定义的属性组将显示在左边，它们可通过红色箭头附加或分离。如果您需要有关当前属性组配置的更多信息或想修改它，可用“属性组配置”图标按钮打开最顶层的“属性组配置”对话框。

对话框右下部的“可使用/固定属性”表区域会列出当前已赋值的所有属性组属性。此列表可通过单击顶部的“名称”或“值”按钮进行改动。此操作将打开一个选择列表，其中列出附加到主机的所有属性（即，在以下属性组中配置的所有属性的联合：全局属性组、主机属性组以及如上所述附加到主机的管理员定义的属性组）。“属性选择”对话框如图 6-6 所示。选择一个属性，并按“确定”按钮确认，即可将该属性添加至“可使用/固定属性”表的“名称”栏中，且光标会指向相应的“值”字段。用鼠标左键双击“值”字段即可修改现有值。如欲删除某属性，首先要用鼠标左键选择表中的相应行。然后，可以通过键入 CTRL-D 或单击鼠标右键打开删除框并确认删除来删除选定的列表项。

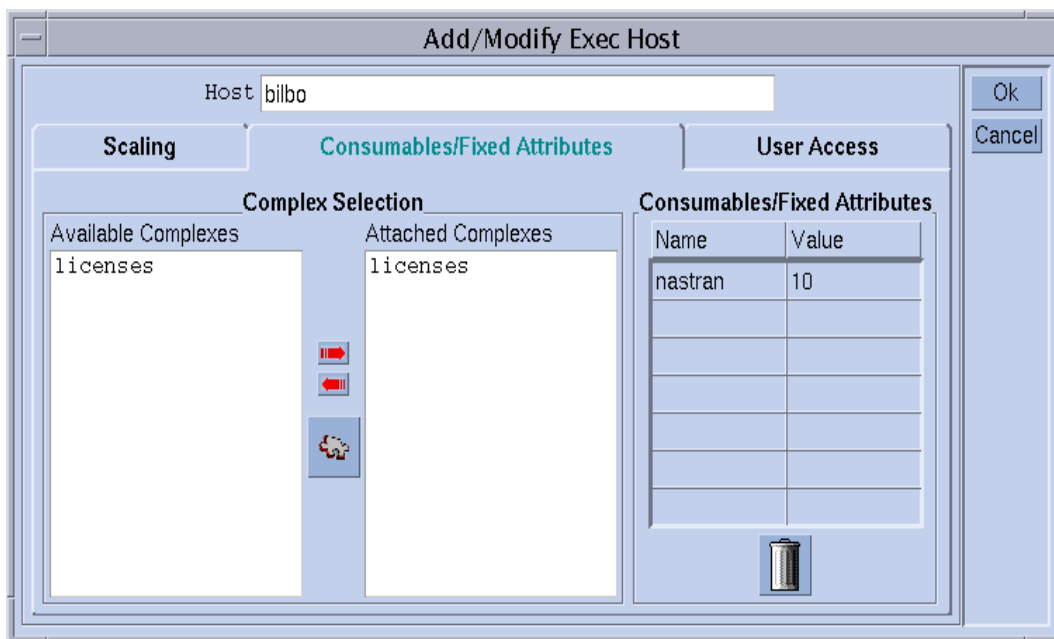


图 6-5 修改可使用/固定属性

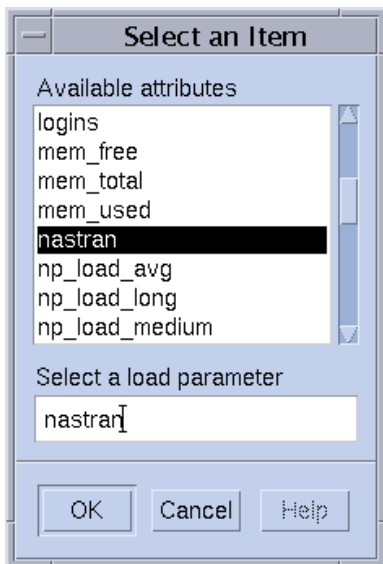


图 6-6 可用属性组属性

通过选择“用户访问权限”选项卡（图 6-7），可以基于之前配置的用户访问权限列表，定义对执行主机的访问权限。

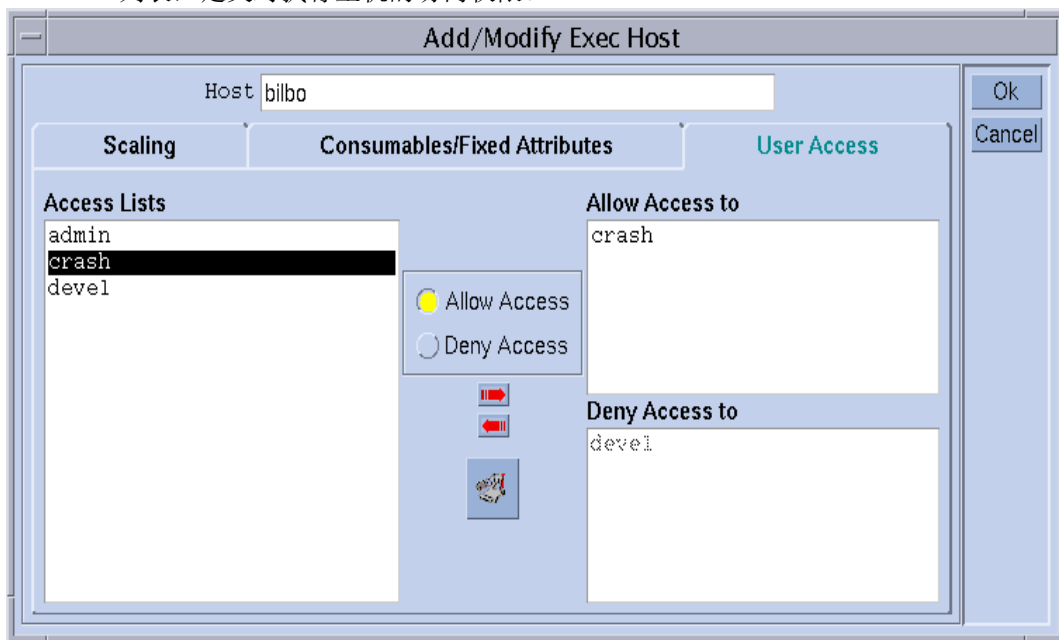


图 6-7 修改用户访问权限

▼ 如何从命令行配置执行主机

- 根据想要配置主机的方式，输入以下命令及其相应自变量。

```
% qconf 自变量
```

维护执行主机列表的命令行界面由 `qconf` 命令的以下选项提供。

- `qconf -ae [执行主机模板]`

添加执行主机 — 此命令启动一个编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量对应的编辑器），其中显示执行主机配置模板。若提供可选参数 *执行主机模板*（已经配置的执行主机名称），此执行主机的配置将用作模板。通过更改模板并将其保存至磁盘来配置执行主机。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `host_conf` 项以获得要更改的模板项的详细说明。

- `qconf -de 主机名`

删除执行主机 — 将指定主机从执行主机列表中删除。执行主机配置中的所有项都将丢失。

- `qconf -me 主机名`

修改执行主机 — 此命令启动一个编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量对应的编辑器），其中显示指定的执行主机配置（即模板）。通过更改模板并将其保存至磁盘来修改执行主机配置。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `host_conf` 手册页以获得要更改的模板项的详细说明。

- `qconf -Me 文件名`

修改执行主机 — 将 *文件名* 的内容用作执行主机配置模板。指定文件中的配置必须关涉现有执行主机。此执行主机的配置将由该文件的内容代替。此 `qconf` 选项对于脱机更改执行主机配置很有用；例如，在 `cron` 作业中，因为它不需要任何手动交互操作。

- `qconf -se 主机名`

显示执行主机 — 显示所指定执行主机的配置（如 `host_conf` 中所定义）。

- `qconf -sel`

显示执行主机列表 — 显示配置为执行主机的主机名列表。

▼ 如何用 qghost 监视执行主机

qghost 提供一种便利的方法来检索有关执行主机状态的简明概述。

- 请输入以下命令。

```
% qghost
```

此命令产生的输出与下例相似。

表 6-1 qghost 输出示例

| 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 |

请参考 《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》中的 qghost 项，以获得输出格式以及更多选项的描述。

▼ 如何从命令行中止守护程序

- 使用以下命令之一。注意，您需要有 Sun Grid Engine 管理人员或操作人员权限才能执行这些操作（请参见第九章第 195 页的“管理用户访问权限和策略”）。

```
% qconf -kej  
% qconf -ks  
% qconf -km
```

- 第一行命令将中止当前所有活动的作业，并关闭所有 Sun Grid Engine 执行守护程序。

注意 – 若用 `qconf -ke` 代替该命令，Sun Grid Engine 执行守护程序将中止，但不会取消活动的作业。直到 `sgexecd` 再次重新启动，系统中在 `sgexecd` 未运行时结束的作业才会报告给 `sgqmaster`。不过，作业报告不会丢失。

- 第二行命令将会关闭 Sun Grid Engine 调度程序 `sgschedd`。
- 第三行命令强制终止 `sgqmaster` 进程。

若有正在运行的作业，并且想等到当前活动的作业结束后再关闭 Sun Grid Engine 过程，可在执行上述 `qconf` 命令行之前对每个队列使用以下命令。

```
% qmod -d 队列名
```

`qmod` 禁用命令阻止将新作业调度到禁用的队列。您应该等到队列中不再有正运行的作业，才中止守护程序。

▼ 如何从命令行重新启动守护程序

1. 以 `root` 用户身份登录到要重新启动 Sun Grid Engine 5.3 守护程序的机器。
2. 执行以下脚本。

```
% <sgc 根目录>/<单元>/common/rcsgc
```

此脚本将寻找通常在此主机上运行的守护程序，然后启动相应的守护程序。

基本群集配置

基本的 Sun Grid Engine 群集配置是一组配置信息，它们反映诸如 `mail` 或 `xterm` 等程序的有效路径一类的依赖于站点的配置，并影响 Sun Grid Engine 的运作。有一个全局配置，提供给 Sun Grid Engine 主控主机和 Sun Grid Engine 池中的每一台主机。此外，Sun Grid Engine 系统还可以配置为使用每台主机本地的配置，以覆盖全局配置中的特定项。

《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》的 `sgconf` 项含有各配置项的详细说明。安装一完成，Sun Grid Engine 群集管理员就应该调整全局和本地配置以适应站点需要，且此后不断更新。

▼ 如何从命令行显示基本群集配置

显示当前配置的 Sun Grid Engine 命令为 `qconf` 程序的显示配置选项。以下为几个示例（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》获得详细描述）。

- 输入以下命令之一。

```
% qconf -sconf
% qconf -sconf global
% qconf -sconf < 主机 >
```

前两条命令的作用相同，都是显示全局配置。第三条命令显示主机的本地配置。

▼ 如何从命令行修改基本群集配置

注意 – 用于更改群集配置的 Sun Grid Engine 命令 `qconf` 仅可由 Sun Grid Engine 管理员使用。

- 输入以下命令之一。

```
% qconf -mconf global
% qconf -mconf < 主机 >
```

- 第一个命令示例修改全局配置。
- 第二个示例对指定的执行主机或主控主机的本地配置进行操作。

以上两条命令为许多可用的 `qconf` 命令中的两个示例。请参考《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》以获得其它命令的信息。

▼ 如何用 QMON 显示群集配置

1. 在 QMON 主菜单中，单击“群集配置”按钮。

显示“群集配置”对话框，与图 6-8 中的示例类似。

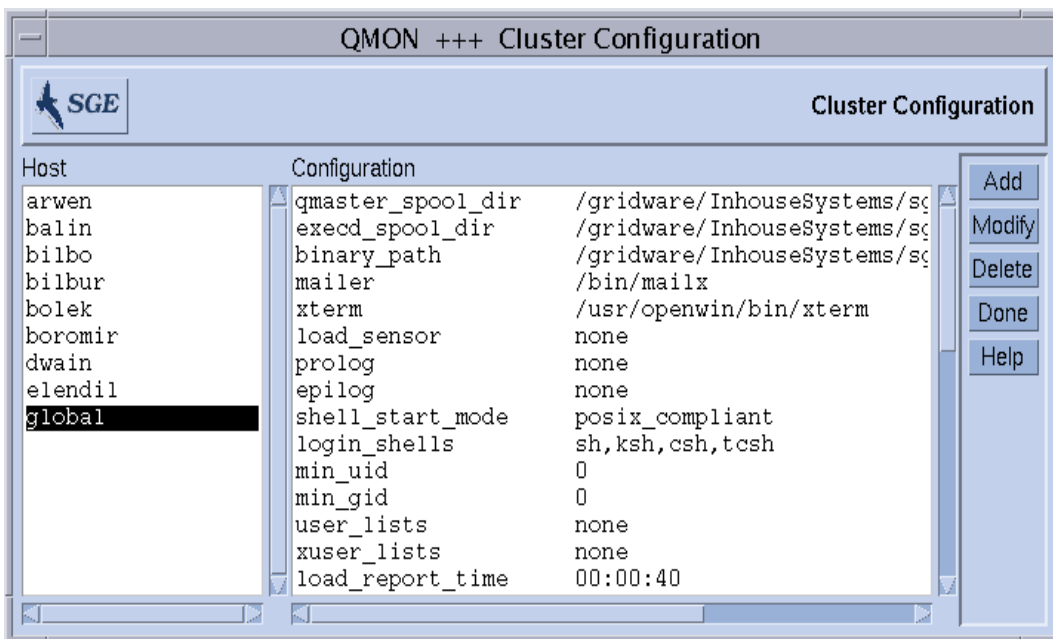


图 6-8 “群集配置”对话框

2. 在屏幕左边的“主机”选择列表中，单击某个主机名，即可显示该主机的当前配置。

▼ 如何用 QMON 删除群集配置

1. 在 QMON 主菜单中，单击“群集配置”按钮。
2. 在屏幕左边的“主机”选择列表中，单击要删除其配置的主机的名称。
3. 按下“删除”按钮。

▼ 如何用 QMON 显示全局群集配置

- 在“主机”选择列表中，选择名称 global。

配置将以 sge_conf 手册页中描述的格式显示。使用“修改”按钮修改选定的全局配置或主机本地配置。使用“添加”按钮为指定的主机添加新配置。

▼ 如何使用 QMON 修改全局配置和主机配置

1. 在“群集配置”对话框中（如第 146 页的“如何用 QMON 显示群集配置”一节中所述），单击“添加”按钮或“修改”按钮。

将打开“群集设置”对话框，与图 6-9 中所示的例子类似。

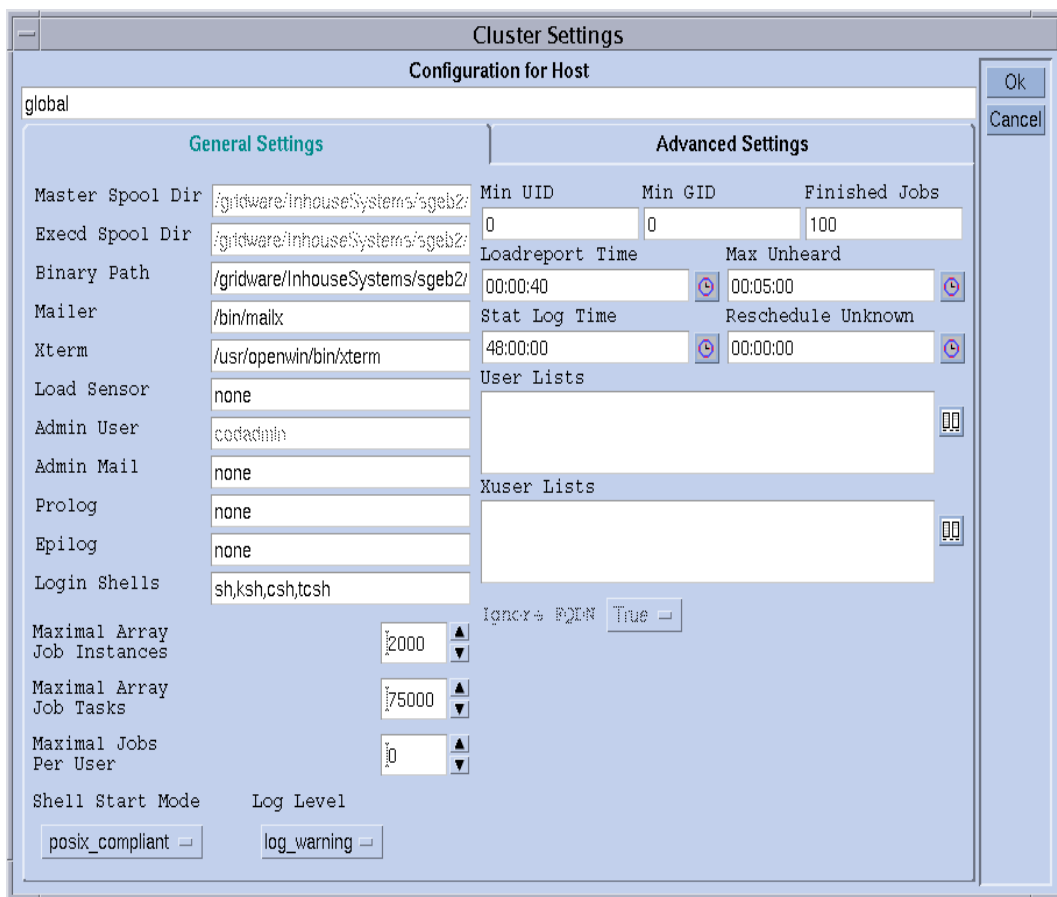


图 6-9 “群集设置”对话框 — 常规设置

2. 根据以下各节中的详细信息指导进行更改。

可以在“群集设置”对话框中更改全局配置或主机本地配置的所有参数。只有更改全局配置时，才能访问所有输入字段；即，从主机列表中选择 **global** 并按“修改”时。若修改常规主机，其实际配置将反映到对话框中，仅有那些可修改的参数可用于主机的本地更改。若添加了新的主机本地配置，对话框的各输入项皆为空字段。

“高级设置”选项卡（图 6-10）根据更改全局配置、主机本地配置还是新配置，显示相应操作。可借此选项卡访问极少使用的群集配置参数。

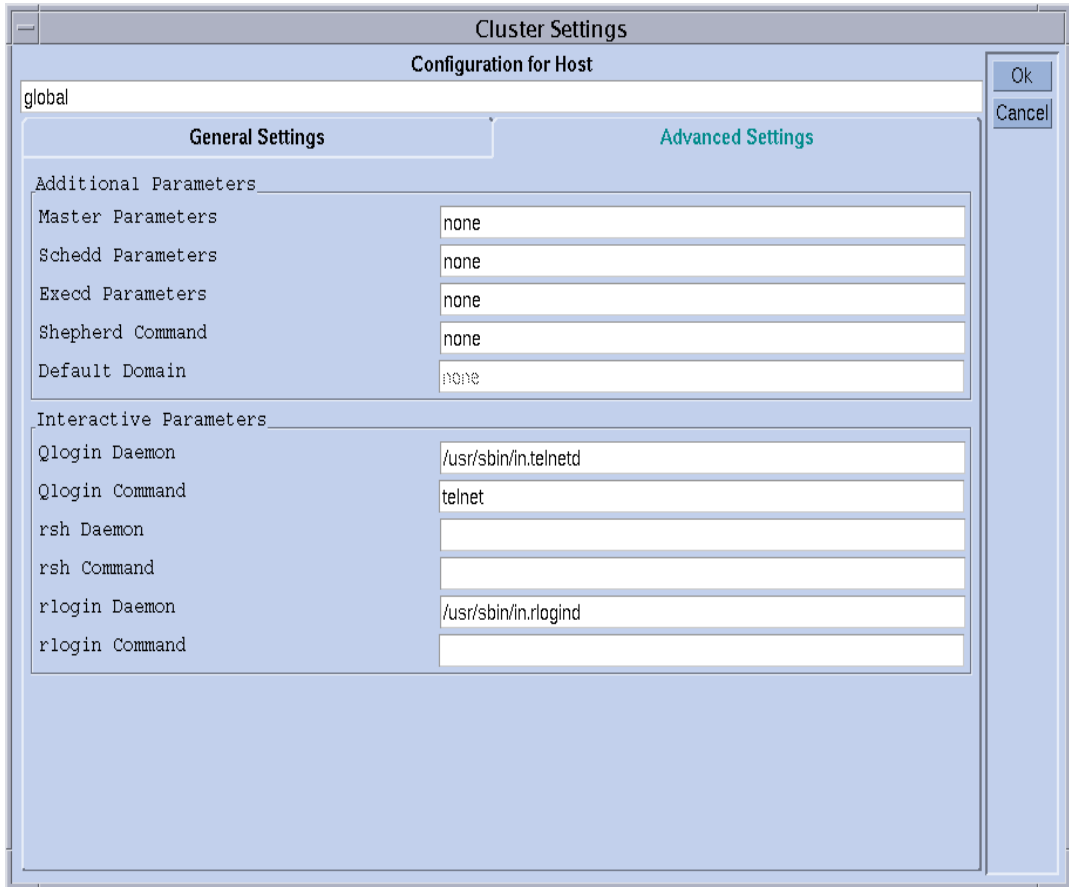


图 6-10 “群集设置”对话框 — 高级设置

完成修改后，按下右上角的“确定”按钮即可注册所修改的配置。按“取消”按钮可放弃所有更改。这两种情况下，对话框均关闭。

请参考 `sge_conf` 手册页，以获取有关所有群集配置参数的详尽描述。

配置队列和队列日历

本章提供有关配置 Sun Grid Engine 5.3 队列的背景信息和指导。

以下是本章所含的具体任务指导的列表。

- 第 150 页的 “如何用 QMON 配置队列”
- 第 151 页的 “如何配置常规参数”
- 第 152 页的 “如何配置 “执行方法” 参数”
- 第 153 页的 “如何配置 “点检查” 参数”
- 第 154 页的 “如何配置负荷和暂停阈值”
- 第 155 页的 “如何配置 “限制””
- 第 157 页的 “如何配置用户 “属性组””
- 第 158 页的 “如何配置 “从属队列””
- 第 159 页的 “如何配置 “用户访问权限””
- 第 160 页的 “如何配置 “拥有者””
- 第 161 页的 “如何从命令行配置队列”
- 第 162 页的 “如何用 QMON 配置队列日历”
- 第 165 页的 “如何从命令行配置日历”

关于配置队列

Sun Grid Engine 队列是各种的作业的 “容器”，并为同一种类的多个作业的并行执行提供相应资源。作业不会在 Sun Grid Engine 队列中等待，一旦得到分配就会立即开始运行。Sun Grid Engine 调度程序的作业暂挂列表是 Sun Grid Engine 作业唯一的等候区域。

配置 Sun Grid Engine 队列将向 `sge_qmaster` 注册队列属性。它们一经配置，就立即对整个群集和属于 Sun Grid Engine 池的全部主机上的所有 Sun Grid Engine 用户都变得可见。

▼ 如何用 QMON 配置队列

1. 在 QMON 主菜单中，按下“队列控制”按钮。
2. 在“队列控制”对话框中，按“添加”或“修改”按钮。

随即打开“队列配置”对话框。第 120 页的“如何用 QMON 控制队列”一节讲述了用于监视和控制队列状态的“队列控制”对话框及其工具。如果是首次打开“队列配置”对话框，它会显示“常规参数”表单（请参见第 151 页的“如何配置常规参数”）。

3. 请根据以下各节的详细信息来决定配置方案。

位于屏幕区域上部的“队列”和“主机名”窗口中显示或定义欲进行的操作将会影响到的队列。如欲修改某个队列，在打开“队列配置”对话框之前，必须在“队列控制”对话框中选定现有队列。如欲添加新队列，必须定义队列名及其所在主机。

为了增加“队列配置”对话框的易用性，在“主机名”窗口下放置了三个直接可用的按钮：“精确复制”按钮，用来通过队列选择列表导入现有队列的所有参数；“复位”按钮，用来加载模板队列的配置；以及“刷新”按钮，用来加载当“队列配置”对话框打开时修改过的其它对象的配置（请参见第 157 页的“如何配置用户“属性组””和第 159 页的“如何配置“用户访问权限””两节，以获得关于“刷新”按钮的详细说明）。

“队列配置”对话框右上角的“确定”按钮用于向 `sge_qmaster` 注册更改，而其下的“取消”按钮则用于放弃所有更改。这两个按钮都会关闭对话框。

可以通过 9 个参数的设置来定义一个队列。

- “常规”（请参见第 151 页的“如何配置常规参数”）
- “执行方法”（请参见第 152 页的“如何配置“执行方法”参数”）
- “点检查”（请参见第 153 页的“如何配置“点检查”参数”）
- “负荷 / 暂停阈值”（请参见第 154 页的“如何配置负荷和暂停阈值”）
- “限制”（请参见第 155 页的“如何配置“限制””）
- “属性组”（请参见第 157 页的“如何配置用户“属性组””）
- “从属队列”（请参见第 158 页的“如何配置“从属队列””）
- “用户访问权限”（请参见第 159 页的“如何配置“用户访问权限””）
- “拥有者”（请参见第 160 页的“如何配置“拥有者””）

可通过“队列参数”选项卡选择所需的参数设置。

▼ 如何配置常规参数

- 选择“常规”参数设置。

出现一个与图 7-1 中所示示例相似的屏幕。

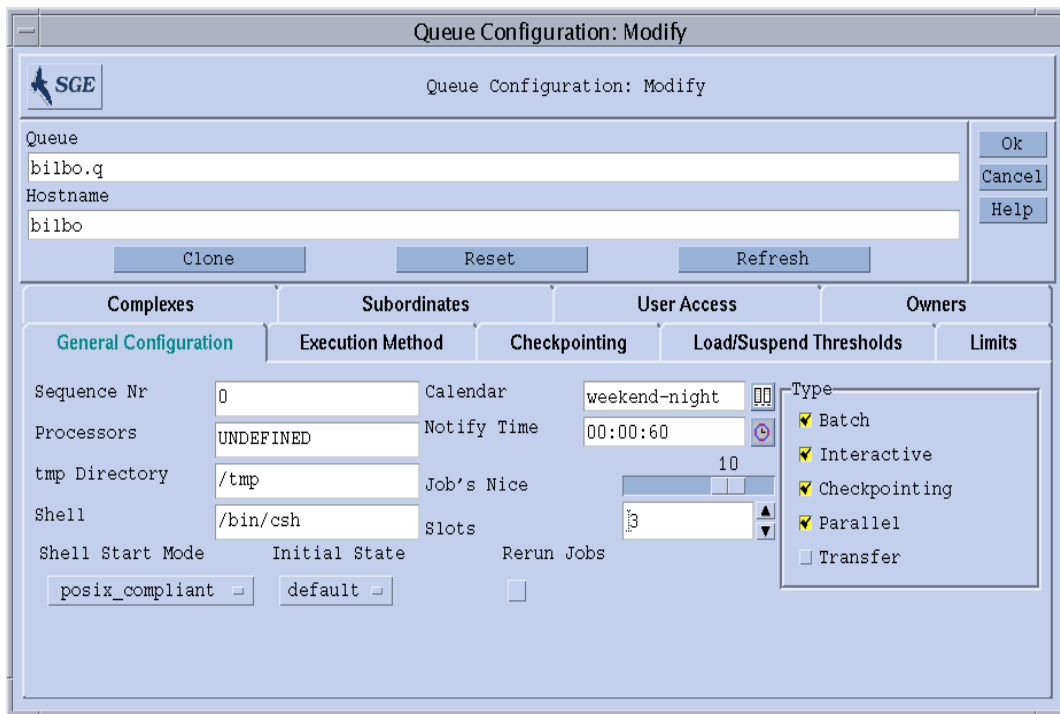


图 7-1 队列配置 — 常规参数

所提供的字段可用来设定以下参数：

- 队列的序列号。
- 处理器 — 该队列中运行的作业将用到的处理器的指示符。对于某些操作系统，该参数可以是一个范围（如 1-4、8、10）或只是处理器设置的一个整数标识符。请参见 Sun Grid Engine 的 doc 目录中的 arc_depend_*.asc 文件，可获得更多信息。
- 临时目录的路径。
- 用来执行作业脚本的缺省命令解释器 (Shell)
- 附加到队列的日历，它定义队列的*工作*和*不工作*时间。
- 发出 SIGUSR1/SIGUSR2 通知信号与发出暂停/中止信号之间的等待时间（通知）。
- 在此队列中启动作业所依据的执行序号值（0 表示使用系统缺省值）。
- 队列中允许同时执行的作业的数目（作业位置数）。

- 队列的类型和允许在此队列中执行的作业的类型。允许选择多项。
- Shell 启动模式；即，启动作业脚本的模式。
- 新添加队列启动时所具有的，或当运行于该队列主机中的 `sgexecd` 重新启动时队列被恢复到的初始状态。
- 队列的缺省重新运行策略，用于强加于那些已被中止（例如，由于系统崩溃）的作业。用户可用 `qsub -r` 选项或通过“作业提交”对话框覆盖这一策略（请参见图 4-7）。

请参阅 `queue_conf` 手册页，以获得有关这些参数的细节。

▼ 如何配置“执行方法”参数

- 选择“执行方法”参数设置。

出现一个与图 7-2 中所示示例相似的屏幕。

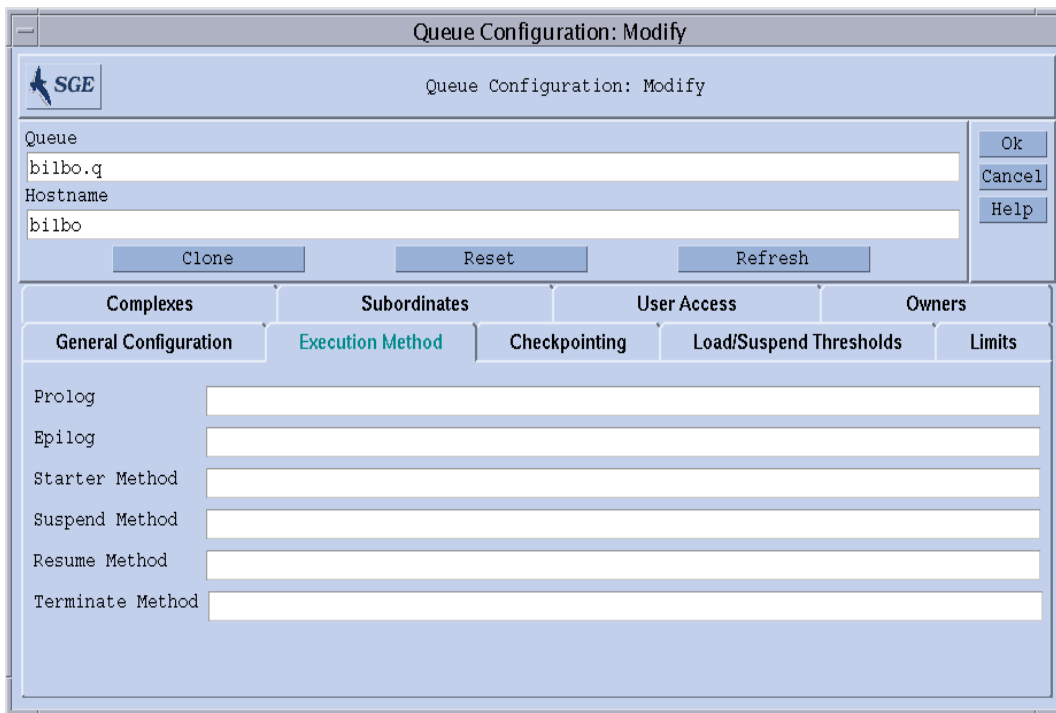


图 7-2 队列配置 — “执行方法”参数

所提供的字段可用来设定以下参数：

- 队列专用的前导脚本和收尾脚本，它们分别在作业脚本启动之前和结束之后执行，其执行环境与作业环境相同。
- 启动 / 暂停 / 恢复 / 终止方法，用于覆盖 Sun Grid Engine 的这些缺省方法，这些操作将应用于作业。

请参阅 `queue_conf` 手册页，以获得有关这些参数的细节。

▼ 如何配置 “点检查” 参数

- 选择 “点检查” 参数设置。

出现一个与图 7-3 中所示示例相似的屏幕。

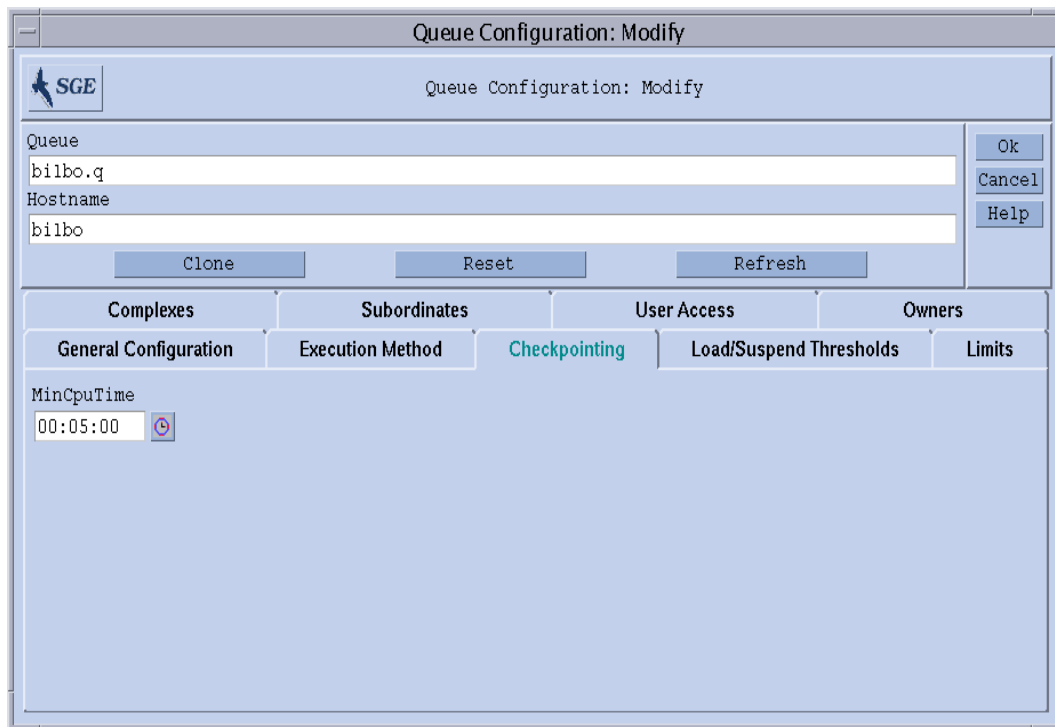


图 7-3 队列配置 — “点检查” 参数

所提供的字段可用来设定以下参数。

- 周期性点检查的时间间隔（最小 CPU 时间）

请参阅 `queue_conf` 手册页，以获得有关这些参数的细节。

▼ 如何配置负荷和暂停阈值

- 选择“负荷 / 暂停阈值”参数设置。

出现一个与图 7-4 中所示示例相似的屏幕。

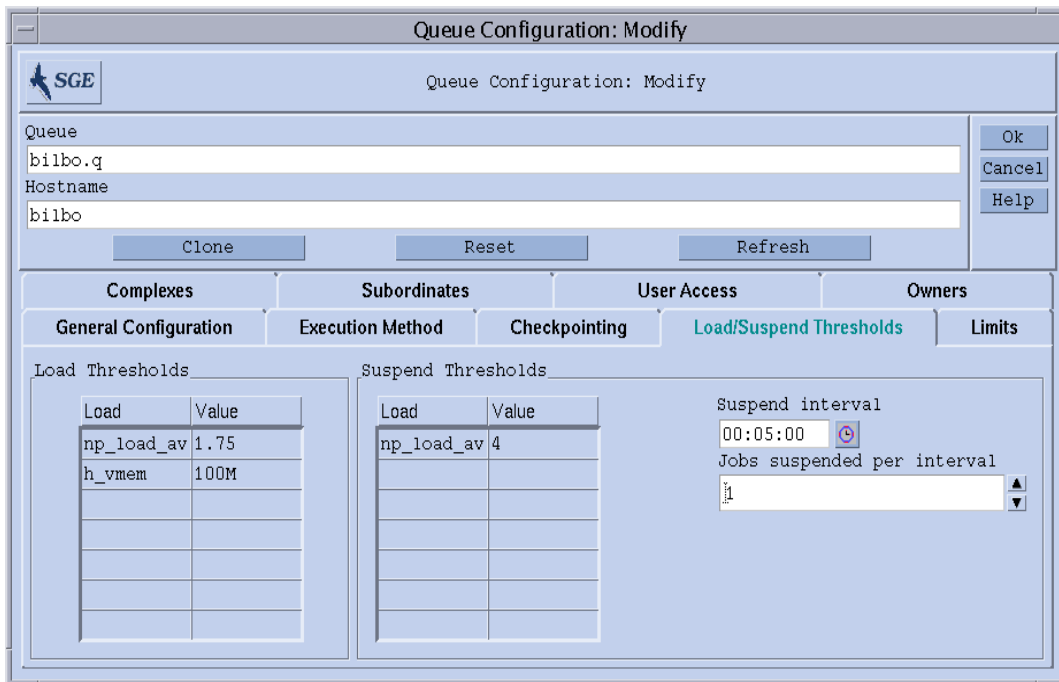


图 7-4 队列配置 — 负荷阈值

所提供的字段可用于设定以下参数。

- “负荷阈值”表和“暂停阈值”表为负荷参数以及可使用属性组属性定义过载阈值（请参见第 167 页的“关于属性组”）。

如果超过负荷阈值的最大值，会导致 Sun Grid Engine 阻止队列接收更多作业。超出一个或多个暂停阈值会引起队列中作业的暂停以减少负荷。表格中显示了当前配置的阈值。用鼠标左键双击相应“值”字段，即可选定并更改现有的阈值。要添加新的阈值，可单击顶部的“名称”或“值”按钮。随之打开一个选择列表，其中列出附加于该队列的所有有效属性。“属性选择”对话框如图 6-6 所示。从中选择一个属性并按“确定”按钮进行确认，会把该属性添加到相应阈值表的“名称”栏，且光标置于它的“值”一栏中。要删除选定的列表项，可按 CTRL-D，或单击鼠标右键打开一个删除框，并确认删除操作。

- 每个时间间隔暂停的作业数量，用以减少被配置队列所在系统的负荷。
- 如果仍然超出暂停阈值，继续暂停作业的时间间隔。

请参阅 queue_conf 手册页，以获得有关这些参数的细节。

▼ 如何配置“限制”

- 选择“限制”参数设置。

出现一个与图 7-5 中所示示例相似的屏幕。

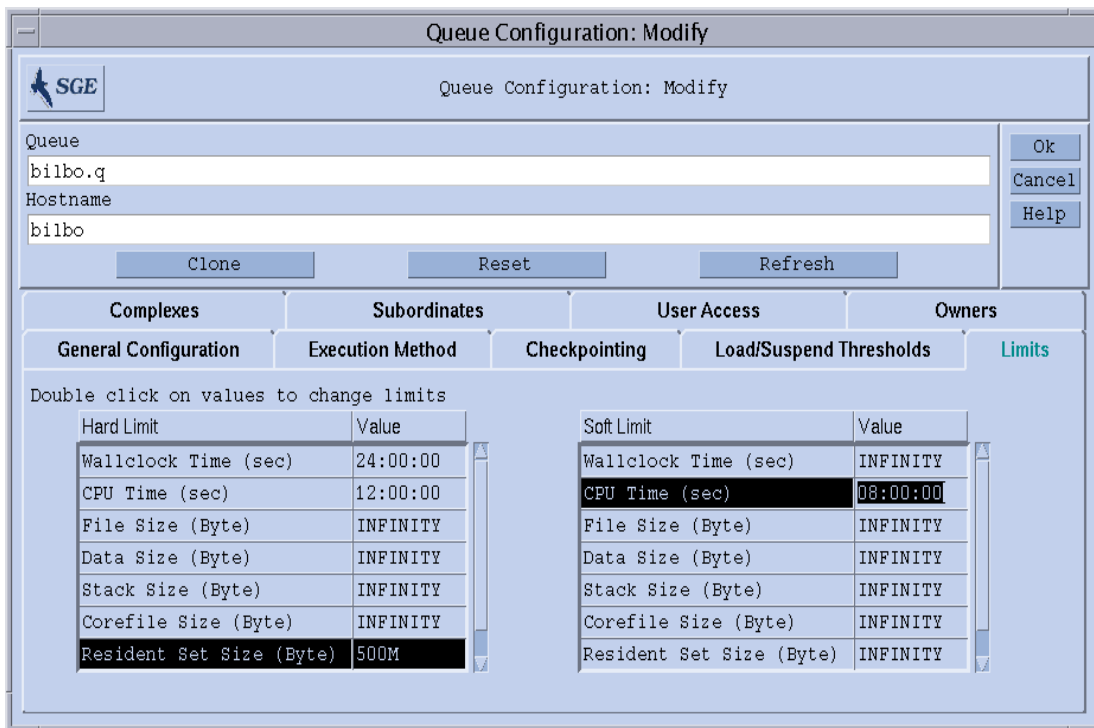


图 7-5 队列配置 — “限制”

所提供的字段可用于设定以下参数。

- 影响队列中运行的作业的硬性和软性限制。

双击某限制项的“值”字段可更改该项限制的值。双击“值”字段两次，可打开“内存”或“时间”限制值快捷输入对话框（参见图 7-6 和图 7-7）。

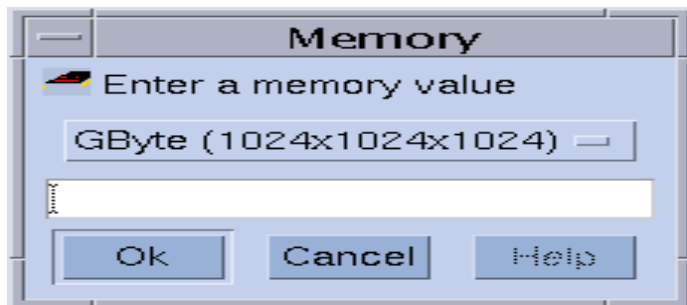


图 7-6 “内存”输入对话框

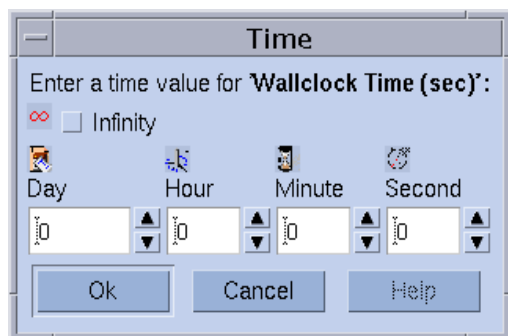


图 7-7 “时间”输入对话框

参阅 `queue_conf` 和 `setrlimit` 手册页，以获得有关各个限制参数及其针对不同操作系统体系结构的解释的详细信息。

▼ 如何配置用户“属性组”

- 选择用户“属性组”参数设置。

出现一个与图 7-8 中所示示例相似的屏幕。

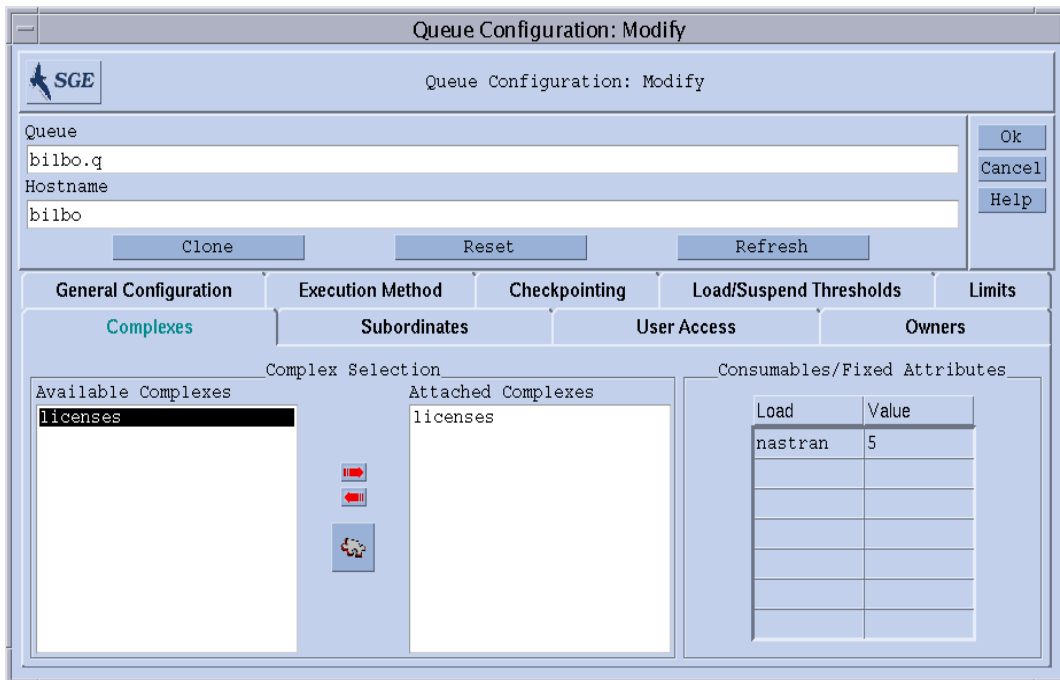


图 7-8 队列配置 — 用户属性组

所提供的字段可用来设定以下参数。

- 用户定义的附加到队列上的一组属性组（请参见第 173 页的“用户定义的属性组”）

“属性组选项”框中间的红色箭头用于将用户定义的属性组附加到队列或从队列分离。

- 值的定义，它针对从该队列中可用的属性组参数集中选出的属性。

缺省情况下，可用的属性组参数由全局属性组、主机属性组和附加的用户定义的属性组组合而成。属性是可使用的或固定的参数。队列值的定义用于定义该队列可管理的容量（针对可使用属性），或仅仅定义一个特定于队列的固定值（针对固定属性）。可参见第 167 页的“关于属性组”以获得更多细节。已明确定义值的属性显示在“可使用/固定属性”表中。双击相应的“值”字段可选择并更改现有属性。要添加新的属性定义，可单击顶部的“名称”或“值”按钮。随之打开一个选择列表，其中列出附加于该队列的所有有效属性。“属性选择”

对话框如图 6-6 所示。从中选择一个属性并按“确定”按钮进行确认，会把该属性添加到属性表的“名称”栏，且光标置于它的“值”一栏中。要删除选定的列表项，可按 CTRL-D，或单击鼠标右键打开一个删除框，并确认删除操作。

请参阅 queue_conf 手册页，以获得有关这些参数的细节。

单击“属性组配置”图标按钮即可打开“属性组配置”对话框（有关示例，请参见第八章的图 8-5，第 167 页的“属性组概念”）。可以在将用户定义的属性组附加到队列或从队列分离之前，检查或修改当前属性组配置。

▼ 如何配置“从属队列”

- 选择“从属队列”参数设置。

出现一个与图 7-9 中所示示例相似的屏幕。

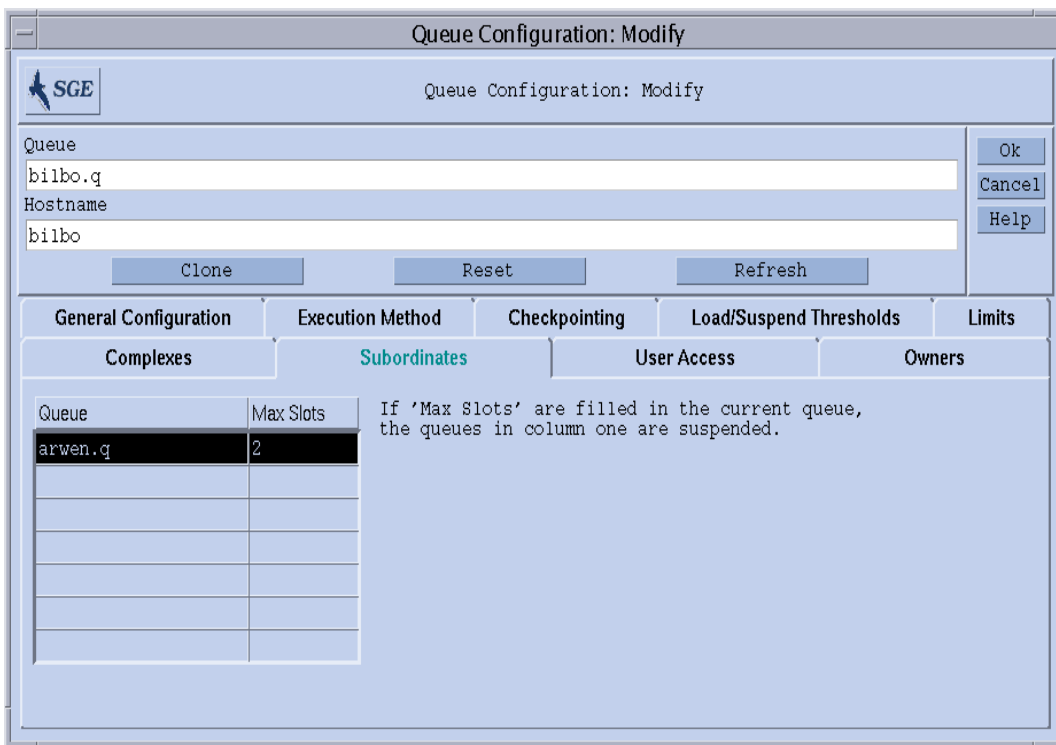


图 7-9 队列配置 — “从属队列”

所提供的字段可用来设定以下参数。

- 从属于已配置队列的队列

已配置的队列繁忙时从属队列会暂停，且已配置的队列不再繁忙时从属队列会取消暂停。对于任一从属队列，可配置作业位置数，当已配置的队列中占用的作业位置数不低于此数，才能引发暂停。如果未指定作业位置数，所有位置数均已占用时才会引发相应队列的暂停。

请参阅 `queue_conf` 手册页，以获得有关这些参数的细节。

从属队列工具可用来实现高优先级队列和低优先级队列以及独立队列。

▼ 如何配置“用户访问权限”

- 选择“用户访问权限”参数设置。

出现一个与图 7-10 中所示示例相似的屏幕。

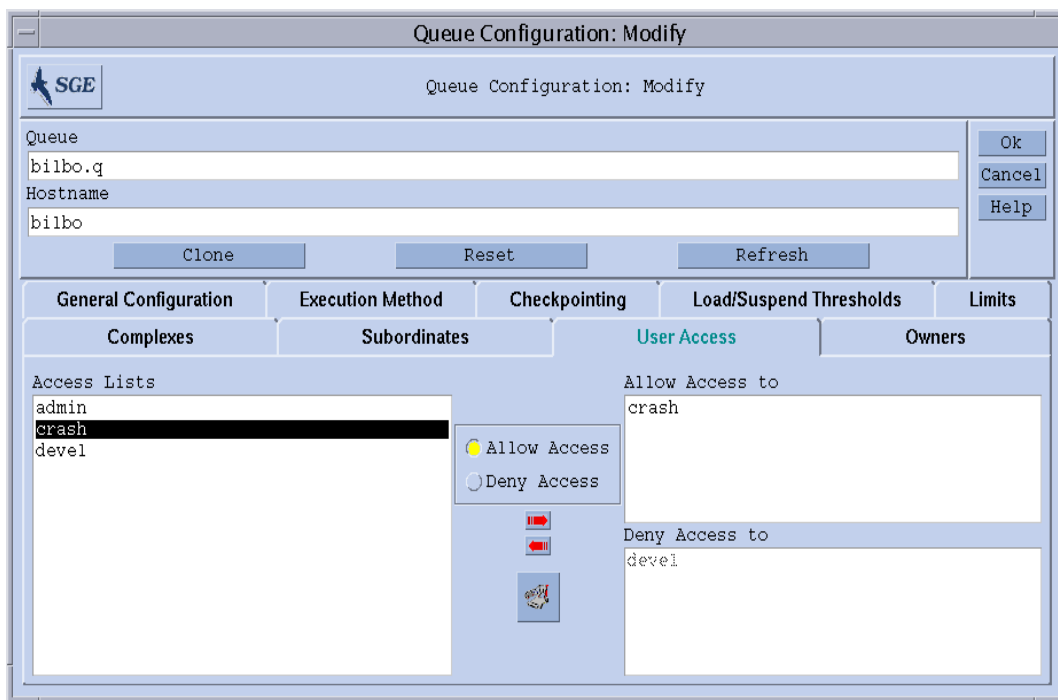


图 7-10 队列配置 — 用户访问权限

所提供的字段可用来设定以下参数。

- 用户访问列表，它附加于队列的允许列表或拒绝列表

访问列表中属于允许列表的用户或用户组有权访问队列。拒绝列表中的用户不能访问队列。如果允许列表为空则访问不受限制，除非在拒绝列表中明确地另行声明。

请参阅 `queue_conf` 手册页，以获得有关这些参数的细节。

单击屏幕中下部的那个按钮可打开“访问列表配置”对话框（参见第 60 页的“用户访问权限”）。

▼ 如何配置“拥有者”

- 选择“拥有者”参数设置。

出现一个与图 7-11 中所示例相似的屏幕。

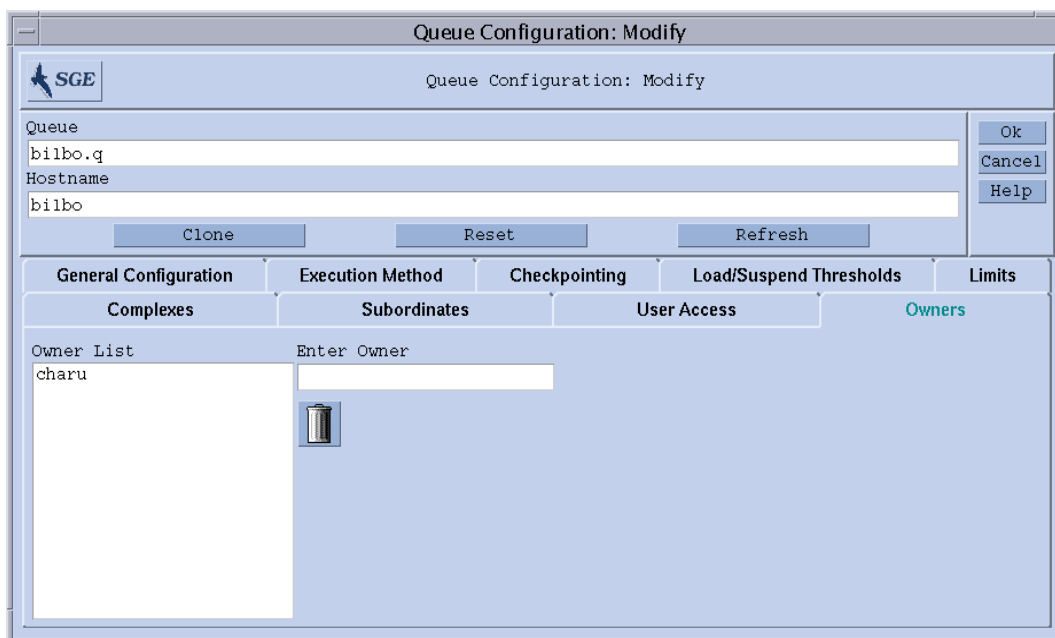


图 7-11 队列配置 — 拥有者

所提供的字段可用于设定以下参数：

- 队列拥有者的列表

队列的拥有者有权暂停 / 取消暂停或禁用 / 启用队列。所有有效用户帐户都可以作为有效值添加到队列拥有者列表中。要从队列拥有者列表中删除某个用户帐户，请在“拥有者列表”中将其选中，然后单击对话框右下角的垃圾桶图标。

请参阅 `queue_conf` 手册页，以获得有关这些参数的细节。

▼ 如何从命令行配置队列

- 根据对队列的配置要求，输入以下命令及其相应选项。

```
# qconf 选项
```

`qconf` 命令有以下选项。

- `qconf -aq [队列名]`

添加队列 — 此命令启动一个编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量对应的编辑器），其中显示队列配置模板。如果提供可选参数 *队列名*，则此队列的配置将用作模板。可通过更改模板并将其保存至磁盘来配置队列。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `queue_conf` 项，以获得要更改的模板项的详细说明。

- `qconf -Aq 文件名`

添加队列 — 用文件 *文件名* 来定义一个队列。定义文件可能已由 `qconf -sq 队列名` 生成（参见下文）。

- `qconf -cq 队列名 [...]`

清除队列 — 清除指定队列的状态，使之闲置，停止运行作业。状态复位，且不考虑当前状态。该选项对于排除错误情形很有用，但不宜在常规操作模式下使用。

- `qconf -dq 队列名 [...]`

删除队列 — 从可用队列列表中删除自变量列表中指定的队列。

- `qconf -mq 队列名`

修改队列 — 修改指定的队列。启动一个编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量对应的编辑器），其中显示欲更改的队列的配置。通过更改配置并保存至磁盘来修改队列。

- `qconf -Mq 文件名`

修改队列 — 用文件 *文件名* 来定义已修改的队列配置。定义文件可能已由 `qconf -sq 队列名` 生成（参见下文）和并进行过后续修改。

- `qconf -sq [队列名 [...]]`

显示队列 — 显示缺省模板队列配置（若不带自变量）或以逗号分隔的自变量列表中所列队列的当前配置。

- `qconf -sql`

显示队列列表 — 显示所有当前已配置队列的列表。

关于队列日历

队列日历以一年中的某天、一周中的某日和 / 或一天中的某时来定义 Sun Grid Engine 队列何时可用。队列可配置成在任意时间更改其状态。可将队列状态更改为已禁用、已启用、已暂停和已恢复（已取消暂停）。

Sun Grid Engine 能够定义一组针对站点的日历，其中每一日历都含有任意状态更改和当其发生时的时间事件。这些日历可供队列引用，即，每个队列附加（或不附加）单个日历，从而采用在附加的日历中定义的可用性配置。

日历格式的语法在手册页 `calendar_conf` 中有详细描述。下面给出了几个示例，并对相应管理工具进行了描述。

▼ 如何用 QMON 配置队列日历

1. 在 QMON 主菜单，单击“日历配置”。

出现类似于图 7-12 的“队列日历配置”对话框。

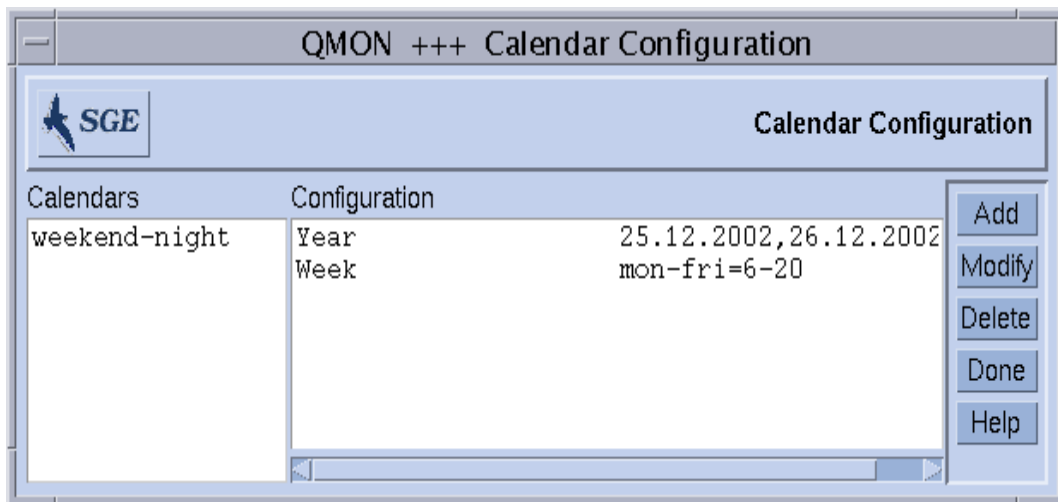


图 7-12 日历配置

屏幕左边的“日历”选择列表中显示可用的访问列表。

2. 在“日历”选择列表中，单击欲修改或删除的日历配置。
3. 根据您想要更改配置的方式，执行以下操作之一。
 - a. 按屏幕右边的“删除”按钮删除所选日历。
 - b. 按“修改”按钮修改所选日历。
 - c. 按“添加”按钮添加访问列表。

在以上所有操作中，都会出现类似于图 7-13 中所示的“日历定义”对话框，它可用于进行删除、修改或添加操作。

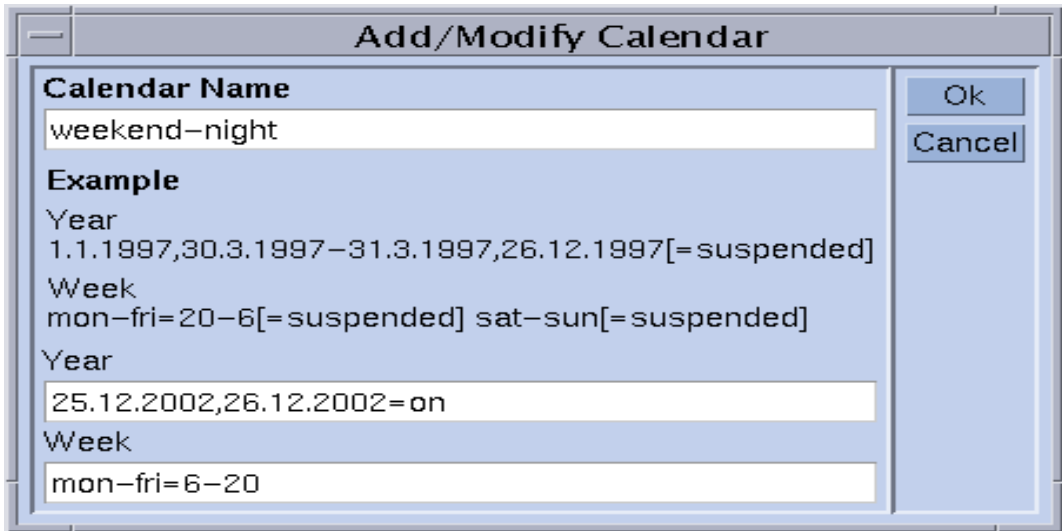


图 7-13 添加、删除或修改日历

4. 根据以下各节的指导继续进行。

若执行的是修改操作，“日历名称”输入窗口会显示所选日历名，您也可用它来输入要声明的日历名称。“年”和“周”输入字段用于定义日历事件，其语法如 `calendar_conf` 手册页所述。

以上日历配置的示例适于那些下班时间以及周末也可用的队列。另外，圣诞期间的假日被定义为视同周末处理。

参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中的 `calendar_conf` 一项，可获得其语法的详细描述和更多示例。

通过将日历配置附加到队列，该日历所定义的可用性配置文件就会成为队列的设置。图 7-14 中显示常规参数队列配置菜单中附加的日历。日历输入字段含有要附加的日历名称，而输入字段旁边的图标按钮可用来打开一个选择对话框，其中列出当前已配置的几个日历。请参见第 149 页的“关于配置队列”一节，以获得有关配置队列的更详细信息。

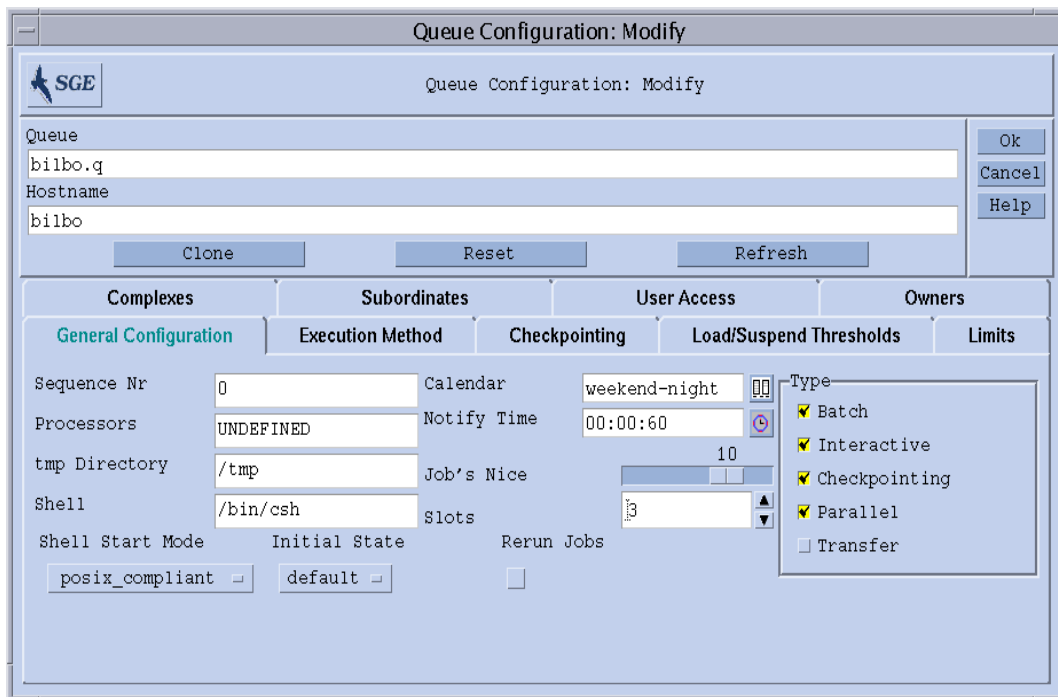


图 7-14 常规参数队列配置菜单中的日历配置

▼ 如何从命令行配置日历

- 请输入以下带有适当开关选项的命令。

```
% qconf 开关选项
```

四个可用的开关选项如下。

- `qconf -Acal, -acal`

添加日历 – 此命令向 Sun Grid Engine 群集添加一个新的日历配置。要添加的日历要么从文件读取，要么是打开一个编辑器，其中显示模板配置，您可在此输入日历。

- `qconf -dcal`

删除 – 日历。

- `qconf -Mcal, -mcal`

修改日历 – 此命令修改现有的日历配置。要修改的日历要么从文件读取 (`-Mcal`)，要么是打开一个编辑器，其中显示原先的配置，您可在此输入新的定义 (`-mcal`)。

- `qconf -scal, -scall`

显示日历 – 此命令显示现有日历配置 (`-scal`)，或显示一份所有已配置日历的列表 (`-scall`)。

属性组概念

本章讲述称为 *属性组* 的重要 Sun Grid Engine 5.3 概念。除了有关属性组及其相关概念的背景信息以外，本章还提供有关如何完成以下各项任务的详细指导。

- 第 168 页的 “如何添加或修改属性组配置”
- 第 177 页的 “如何设置可使用资源”
- 第 188 页的 “如何从命令行修改属性组配置”
- 第 190 页的 “如何写您自己的负荷传感器”

关于属性组

属性组的定义提供了关于用户可能请求的资源属性的所有有关信息，用户请求这些资源属性用于 Sun Grid Engine 作业（通过 `qsub` 或 `qalter -l` 选项）和用于在 Sun Grid Engine 系统内解释这些参数。

属性组还构建了 Sun Grid Engine 系统的 *可使用资源* 功能的框架，可使用此功能定义群集全局属性、特定于主机的属性或与队列相关的属性，这些属性用相关能力来标识资源。在调度时会综合考虑资源的可用性以及 Sun Grid Engine 作业的需求。Sun Grid Engine 还将执行所需的簿记和容量规划，以免过度预订可使用资源。可使用属性的典型例子有：可用空闲内存、未占用的软件包许可证数、空闲磁盘空间或网络连接上的可用带宽。

从广义上讲，Sun Grid Engine 属性组是一种手段，用于说明将如何解释队列、主机和群集的属性。该说明包括属性名称、用于指代它的缩写名、属性值的类型（例如，`STRING` 或 `TIME`）、分配给属性组属性的预定义值、Sun Grid Engine 调度程序 `sge_schedd` 所使用的关系运算符、可否请求标志（该标志决定用户可否为作业请求此属性）、可使用标志（若设置此标志，则将此属性标识为可使用属性），以及缺省请求值（在作业未明确指定可使用属性的请求值，可使用该值）。

图 8-1 中所示的 “QMON 属性组配置” 对话框示例说明如何定义属性组属性。

▼ 如何添加或修改属性组配置

1. 在 QMON 主菜单中，按下“属性组配置”按钮。

显示与图 8-1 中的示例类似的“属性组配置”对话框。

2. 遵循以下各节中的详细指导，添加或修改属性组配置。

- 第 170 页的“队列属性组”
- 第 170 页的“主机属性组”
- 第 172 页的“全局属性组”
- 第 173 页的“用户定义的属性组”

“属性组配置”对话框可用来更改现有属性组的定义和定义新用户属性组。

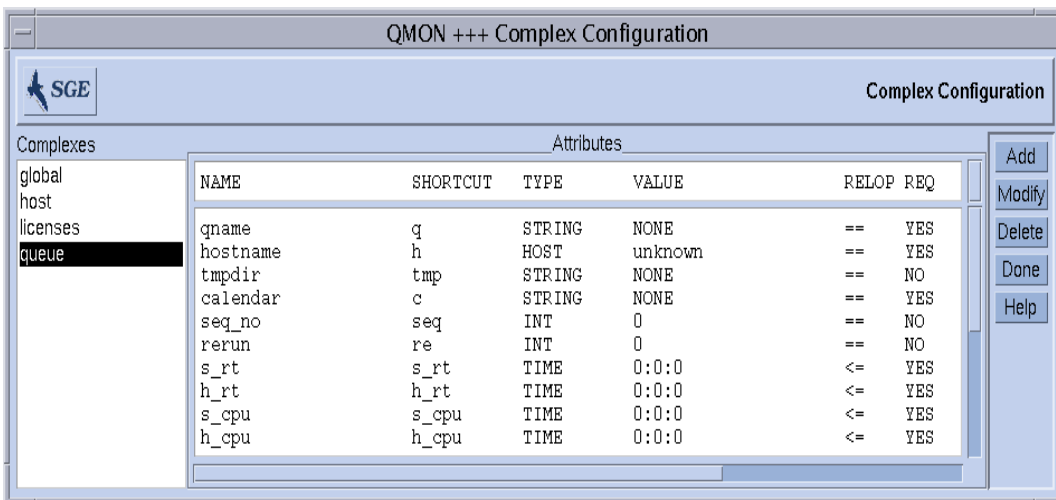


图 8-1 “属性组配置”对话框 — 队列

屏幕的左边会显示系统已知的所有属性组的选择列表。它可用于修改或删除属性组。可通过屏幕右边相应的按钮来选择所需操作（添加、修改或删除）。若要创建一个新属性组或修改现有属性组，则将打开与图 8-2 中示例类似的对话框。

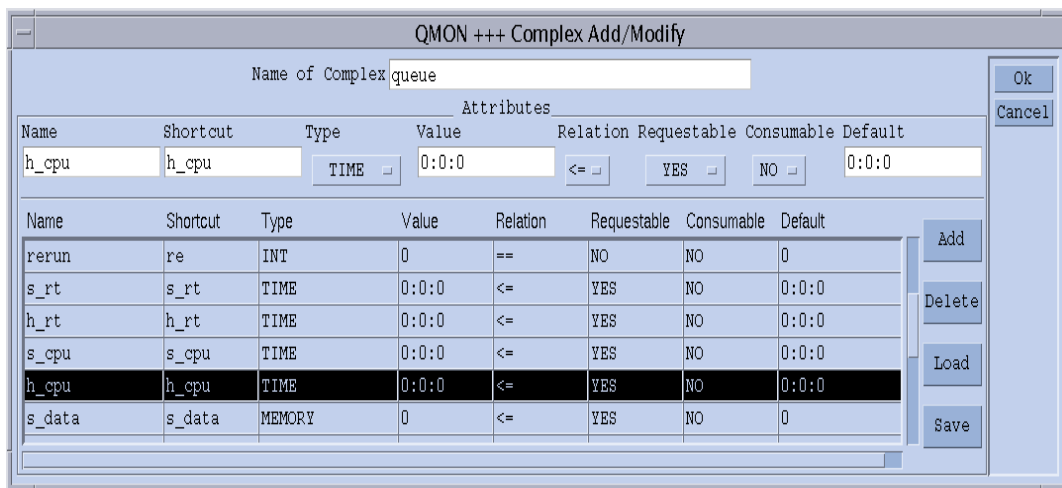


图 8-2 “属性组添加 / 修改”对话框

您必须输入属性组名称或选择它（若其出现在顶端的“属性组名称”输入窗口中）。您可以用鼠标左键在“属性组定义”表中选择一行来修改属性组属性。选定的项将显示在“属性”对话框顶部的定义窗口和选择器中。更改定义并按下“添加”按钮将更新定义表中所作的更改。

填写定义窗口，使用可选择器，然后按下“添加”按钮，即可添加新项。当添加新项时不应选择属性表中的任何一行。

“加载”和“保存”按钮可用于从常规文件加载属性组配置和将属性组配置保存于常规文件中。按这两个按钮会打开一个文件选择框以供选择文件。“删除”按钮可用于删除属性组配置中选定的行。

请参考属性组手册页，以获知有关该表中行和列的具体含义。最后，可用屏幕右下角的“确定”按钮向 sge_qmaster 注册新的或已更改的属性组。

属性组类型

Sun Grid Engine 属性组对象集成了 4 种不同类型的属性组。

- 队列属性组
- 主机属性组
- 全局属性组
- 用户定义的属性组

以下各节详细说明每种类型。

队列属性组

队列属性组通过专用名 `queue` 来引用。

其缺省表格中包含了队列配置中各参数的选择项，这些选择项是在 `queue_conf` 中定义的。队列属性组的主要用途是定义如何解释这些参数，并提供打算用于所有队列的其它属性的容器。因此，队列属性组可通过用户定义的属性来扩展。

若队列属性组是在某一特定队列的背景下引用的，则该队列的相应配置值会替代队列属性组中的属性值（它们覆盖值栏）。

例如，若队列的属性组设置是为名为 *big* 的队列设置的，则队列属性组属性 `qname` 的“值”栏（其缺省值为 `unknown`，请参见图 8-1）将被设置为“*big*”。

这一隐含的值设置可通过队列配置中的 `complex_values` 参数覆盖（请参见第 149 页的“关于配置队列”）。这经常用于*可使用资源*（请参见第 177 页的“可使用的资源”一节）。例如，对于虚拟内存大小的限制，队列配置值 `h_vmem` 将用于限制每项作业所占用的内存总量，而 `complex_values` 列表中相应的项将定义一台主机或指定给一个队列的可用虚拟内存总量。

若管理员添加属性到队列属性组，则与某一特定队列相关联的值或者通过该队列的 `complex_values` 参数来定义，或者缺省使用队列属性组配置中的值栏来定义。

主机属性组

主机属性组通过专用名 `host` 来引用，并且包含所有要基于主机进行管理的属性的典型定义（请参见图 8-3）。与主机有关的属性的标准设置包括两类，但是，同上述队列属性组一样，它是可以增强的。第一类由几种特别适用于基于主机进行管理的队列配置属性组成。这些属性是：

- 位置数
- `seven`
- `h_vmem`
- `s_fsize`
- `h_fsize`

（请参考《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `queue_conf` 项以获得细节。）

注意 – 在主机属性组中定义这些属性与将其也包含在队列配置中并不冲突。允许同时在主机级别和队列级别上维护相应资源。例如，可为某台主机管理虚拟空闲内存的总量 (`h_vmem`)，而此总量的子集也可与该主机上的某队列相关联。

标准主机属性组的第二个属性种类为缺省负荷值。每一 `sgc_execd` 定期向 `sgc_qmaster` 报告负荷。所报告的负荷值或为标准的 Sun Grid Engine 负荷值（如 CPU 平均负荷），或为由 Sun Grid Engine 管理者定义的负荷值（请参见第 189 页的“负荷参数”一节）。标准负荷值的典型定义是缺省主机属性组的一部分，而管理员定义的负荷值需要主机属性组的扩展。

主机属性组通常不仅扩展为包含非标准负荷参数，还管理与主机有关的资源（如分配到一台主机的软件许可证数目或主机本地文件系统上的可用磁盘空间）。

若主机属性组与主机或该主机上的队列相关联，则特定主机属性组属性的具体值由以下之一确定。

- 队列配置（在队列配置派生属性的情况下）
- 报告的负荷值
- 相应主机配置中的 `complex_values` 项中的明确定义值（请参见第 132 页的“关于配置主机”一节）

若以上均不可用（例如，以为该值是负荷参数，但 `sgc_execd` 并未报告其负荷值），则使用主机属性组配置中的值字段。

例如，空闲虚拟内存总量属性 `h_vmem` 在队列配置中定义为限制值，而且作为标准负荷参数报告。主机上的以及附加到该主机的队列的可用虚拟内存总量，可在该主机和队列配置的 `complex_values` 列表中定义。同时再将 `h_vmem` 定义为 *可使用资源*（请参见第 177 页的“可使用的资源”），这使得计算机内存可被有效利用，而不必冒内存过度预订的风险（过度预订经常会导致由 *内存交换* 引起的系统性能的降低）。

注意 – 只可更改系统缺省负荷属性的缩写名、值、关系、可否请求、可否使用和缺省值栏。不可删除缺省属性。

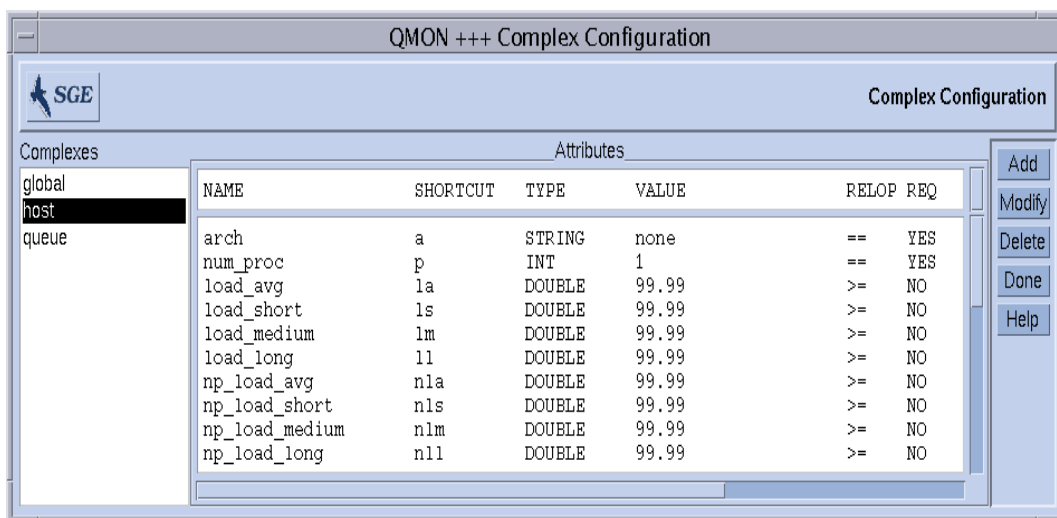


图 8-3 “属性组配置”对话框 — 主机

全局属性组

全局属性组通过专用名 `global` 引用。

全局属性组中配置的各项是指群集范围的资源属性，例如文件服务器的可用网络带宽或网络范围内可用文件系统的空闲磁盘空间（请参见图 8-4）。若相应的负荷报告包含 `GLOBAL` 标识符，则全局资源属性还可与负荷报告相关联（请参见第 189 页的“负荷参数”一节）。全局负荷值可从群集中的任何主机进行报告。缺省情况下，Sun Grid Engine 不报告全局负荷值，因此没有缺省全局属性组配置。

全局属性组属性的具体值或者由全局负荷报告（通过 global 主机配置的 complex_values 参数明确定义，请参见第 132 页的“关于配置主机”一节），或者与特定主机或队列和相应的 complex_values 列表中的明确定义相关联。若非以上情况（例如，负荷值尚未报告），则使用全局属性组配置中的值字段。

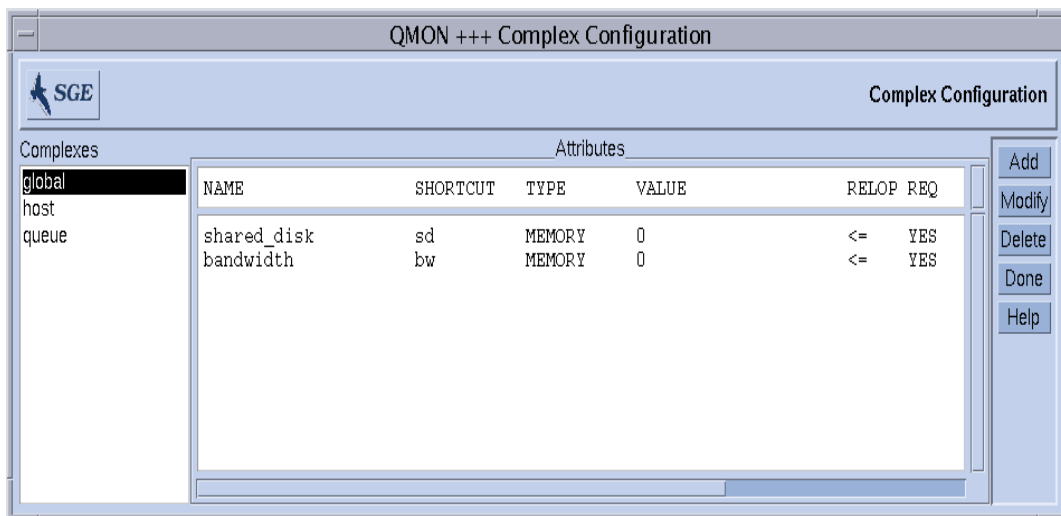


图 8-4 “属性组配置”对话框 — 全局

用户定义的属性组

通过设置用户定义的属性组，Sun Grid Engine 管理者能够扩展由 Sun Grid Engine 管理的属性设置，同时限制那些属性对特定队列和 / 或主机的影响。用户属性组就是一系列已命名的属性以及 Sun Grid Engine 如何处理这些属性的相应定义。可将一个或多个此类用户定义的属性组，通过 complex_list 队列和主机配置参数附加到队列和 / 或主机（请参见第 149 页的“关于配置队列”和第 132 页的“关于配置主机”这两节）。除了缺省属性组属性外，所有指定属性组中定义的属性均可分别用于队列和主机。

与队列和主机相关联的用户定义的属性组的具体值，必须由队列和主机配置中的 complex_values 参数所设置，否则使用用户属性组配置的值字段。

举例来说，定义以下用户定义的属性组 licenses。

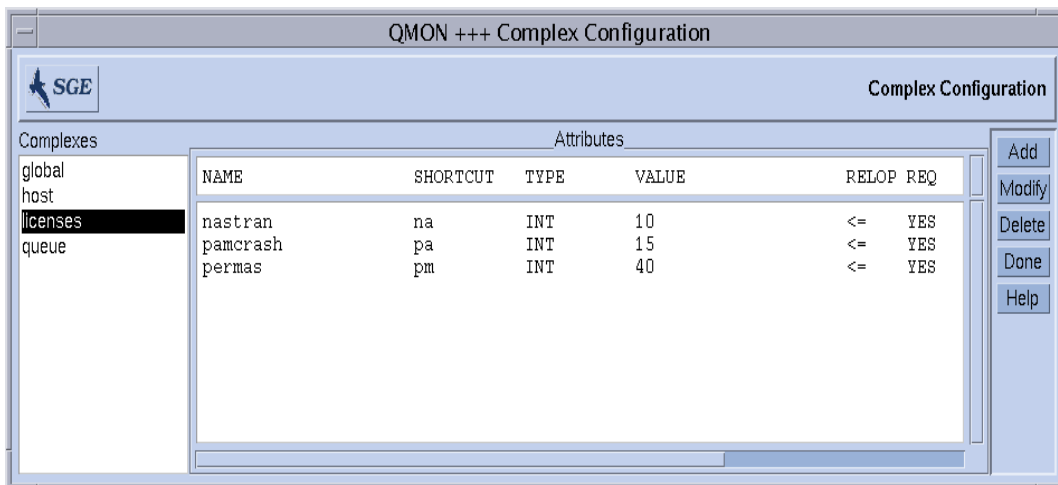


图 8-5 “属性组配置”对话框 — 许可证

并且，如图 8-6 中的队列配置用户属性组子对话框所示，对于至少一个或多个队列，将 licenses 属性组添加到相关联的用户定义属性组中（请参见第 149 页的“关于配置队列”及其相关章节，以获得如何配置队列的细节）。

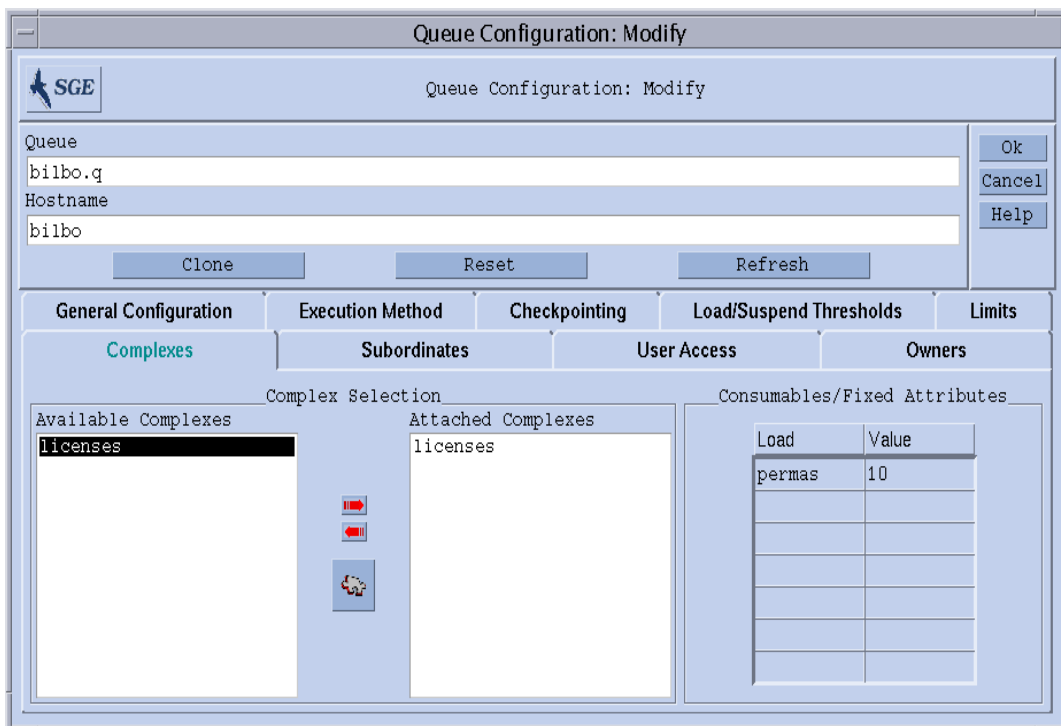


图 8-6 用户定义的属性组队列配置

接下来，所示的队列配置成最多可管理 10 个 permas 软件包许可证。此外，licenses 属性组属性 permas 对 Sun Grid Engine 作业变为可请求，如图 8-7 中的提交对话框的“请求的资源”子对话框中的“可用资源”列表所示（请参见第四章第 63 页的“提交作业”，以获得有关如何提交作业的细节）。

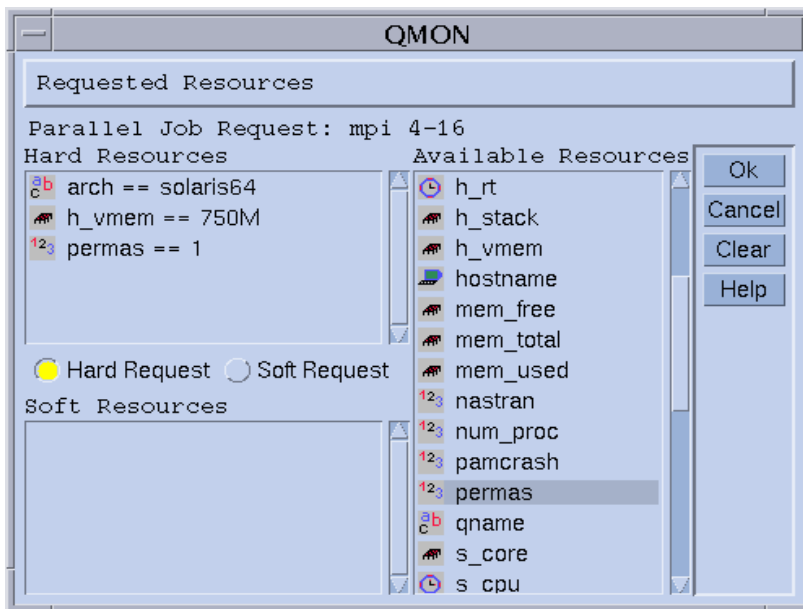


图 8-7 “请求的资源”提交子对话框

或者，用户也可从命令行提交作业，并请求 licenses 属性，如下所示。

```
% qsub -l pe=1 permas.sh
```

注意 – 您可以使用 pm 缩写名代替属性全名 permas。

作为这种配置和类似作业请求的结果，唯一对这些作业合格的队列即那些与用户定义的 licenses 属性组相关联的队列，它们已配置 permas 许可证且能使用该许可证。

无效的用户定义属性组名称

下面是为系统预留、因而不允许指定为用户定义属性组名的属性组名称列表。

- global
- host
- queue

可使用的资源

可使用的资源（又称**可使用资源**）是一种管理有限资源（例如可用内存、文件系统上的空闲磁盘空间、网络带宽或浮动的软件许可证）的有效方式。可使用资源的总可用容量由 Sun Grid Engine 管理员定义，并且相应资源的使用情况由 Sun Grid Engine 内部簿记进行监视。Sun Grid Engine 统计所有运行作业对此资源的使用情况，并确保仅当 Sun Grid Engine 内部簿记表明有足够的可使用资源时才分配作业。

可使用资源可与缺省的或用户定义的负荷参数相结合（请参见第 189 页的“负荷参数”），即，可为可使用属性报告负荷值，或反之为负荷属性设置“可否使用”标志。这种情况下，Sun Grid Engine 可使用资源管理将负荷（测量资源的可用性）和内部簿记均考虑在内，并且确保两者均不超出指定的限制。

要启用可使用资源管理，您必须定义资源的总容量。这可基于群集全局、每台主机及每个队列执行，而这些种类可依指定的顺序相互取代（即，主机可限制群集资源的可用性，而队列可限制主机和群集资源）。资源容量的定义可由队列和主机配置中的 `complex_values` 项执行（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `host_conf` 和 `queue_conf` 这两项，以及第 149 页的“关于配置队列”和第 132 页的“关于配置主机”）。`global` 主机的 `complex_values` 定义指定群集全局可使用资源的设置。对于 `complex_values` 列表中的每一可使用属性组属性，均会赋予一个值，该值表示该资源的最大可用数量。内部簿记将从此总数中减去在作业资源请求中指定的所有运行作业的假定资源使用量。

▼ 如何设置可使用资源

只有数字型的属性组属性（即类型为 `INT`、`MEMORY` 和 `TIME` 的属性组属性）才能配置为可使用。

1. 在 QMON 主菜单中，按下“属性组配置”按钮。

显示与图 8-1 中的示例类似的“属性组配置”对话框。

2. 要执行对某一属性的 Sun Grid Engine 可使用资源管理，请在属性组配置中设置可否使用标志。例如，图 8-8 中对 `virtual_free` 内存资源进行了设置。
3. 遵循以下各节的详细示例，设置其它可使用资源。
 - 第 179 页的“示例 1：浮动软件许可证管理”
 - 第 183 页的“示例 2：虚拟内存的空间共享”
 - 第 186 页的“示例 3：管理可用磁盘空间”

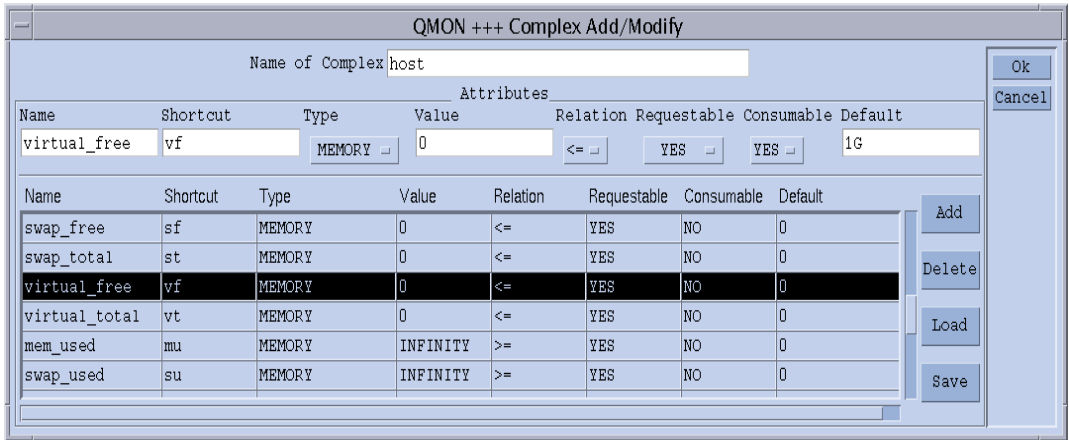


图 8-8 “属性组配置”对话框 — virtual_free

接下来，对于每一个您需要 Sun Grid Engine 执行必需容量规划的队列或主机，您必须在 `complex_values` 列表中定义容量。图 8-9 中的示例将当前主机容量值定义为 1 GB 虚拟内存。

该主机上（任何队列中）所有同时运行的作业的虚拟内存需求均将累加，并从 1 GB 的容量中减去，以确定可用虚拟内存。若某一作业针对 `virtual_free` 的请求超过可用量，则作业不会分配到该主机的队列中。

注意 – 可强制作业请求资源，从而可通过“可否请求”参数的 *强制* 值指定其假定使用量（请参见图 8-8）。

注意 – 对于作业未明确要求的可使用属性，可由管理员预定义缺省的资源使用值（请参见图 8-8 — 缺省设置为 200 MB）。如上所述，预定义的缺省资源使用值仅当未强制请求属性时才有意义。

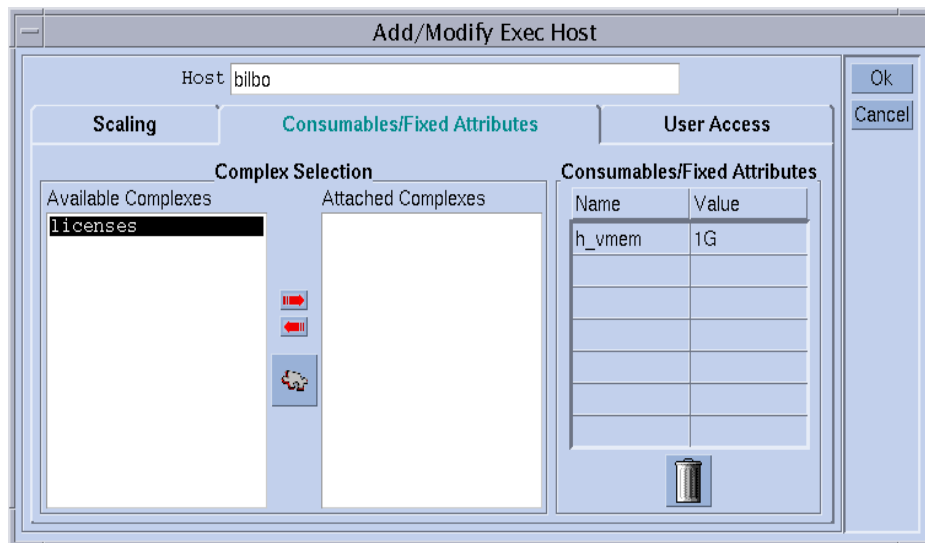


图 8-9 执行主机配置 — virtual_free

设置可使用资源的示例

以下示例可用来指导您设置站点的可使用资源。

示例 1：浮动软件许可证管理

假设群集中使用了 pam-crash 软件包，并且有 10 个浮动许可证，即您可在任何系统中使用 pam-crash，只要该软件的当前调用总数不超过 10。我们的目标是以某种方式配置 Sun Grid Engine，以便只要所有 10 个许可证均被其它正运行的 pam-crash 作业占据时就不再调度 pam-crash 作业。

借助 Sun Grid Engine 可使用资源，可以非常容易地实现此目标。首先，您需要将 pam-crash 许可证的可用数目作为可使用资源添加到全局属性组配置中，如图 8-10 所示。

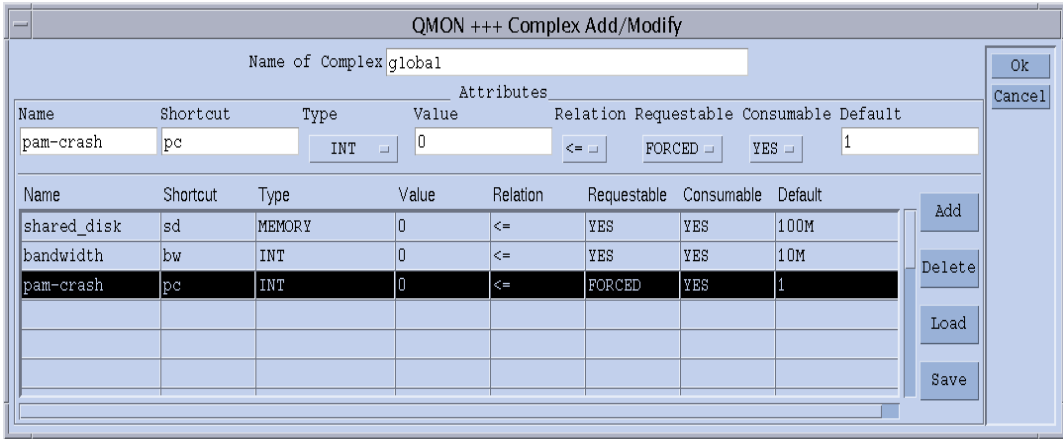


图 8-10 “属性组配置”对话框 — pam-crash

可使用属性的名称设置为 `pam-crash`，而在 `qalter`、`qselect`、`qsh`、`qstat` 或 `qsub -l` 选项中可使用缩写名 `pc`。该属性类型定义为整数。“值”字段的设置与可使用资源无关，因其通过 `complex_values` 列表从全局、主机或队列配置中接收值（请参见下文）。“可否请求”标志设置为强制，表示用户提交作业时必须请求其作业所占用的 `pam-crash` 许可证数。“可否使用”标志最终将该属性定义为可使用资源而与缺省值设置无关，因为可否请求已设置为强制，如此一来，此属性的请求值将随任何作业一道接收。

要激活此属性和群集的资源规划，可用 `pam-crash` 许可证的数目必须在全局主机配置中定义，如图 8-11 所示。属性 `pam-crash` 的值设置为 10，对应 10 个浮动的许可证。

注意 – 可使用 / 固定属性表对应主机配置文件格式 `host_conf` 中所述的 `complex_values` 项。

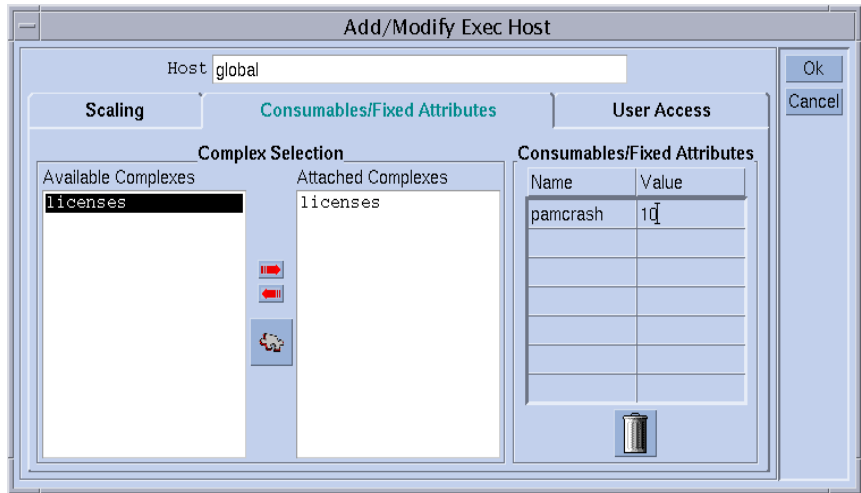


图 8-11 全局主机配置 — pam-crash

假定用户提交以下作业。

```
% qsub -l pc=1 pam-crash.sh
```

该作业将仅在当前占用的 pam-crash 许可证数少于 10 时才启动。不过，该作业可在群集中任何地方运行，且它将在其运行时间内始终为自身占用一个 pam-crash 许可证。

若群集中的某个主机无法包含在浮动许可证中（例如，由于您没有其所用的 pam-crash 二进制程序），则您可以从 pam-crash 许可证管理中排除它，方法是：将与该主机可使用属性 pam-crash 相关的容量设置为 0。此操作必须在“执行主机配置”对话框中执行，如图 8-12 中的主机 bilbo 所示。

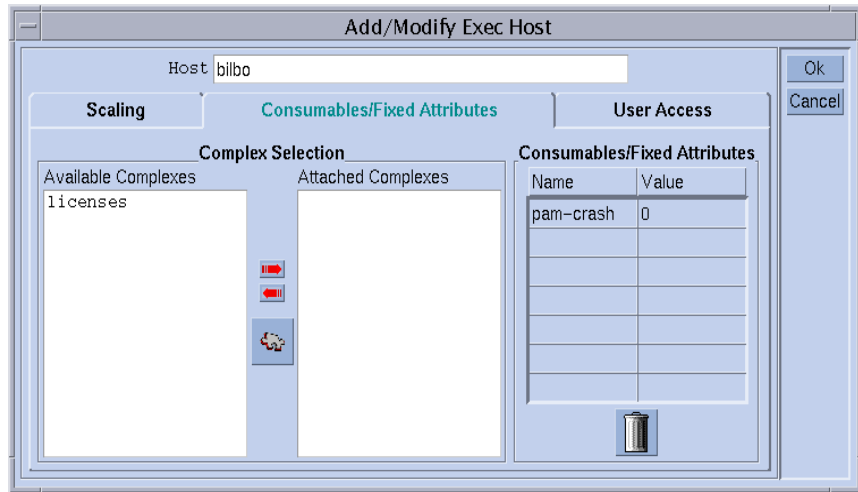


图 8-12 执行主机配置 — pam-crash

注意 — pam-crash 属性对执行主机默认可用，因为 global 属性组的属性可由所有执行主机继承。同样地，通过将容量设置为 0，您还可以将某一特定主机可管理的许可证数目（作为群集全部许可证的一部分）限制为某一非零值（比如 2）。在这种情况下，该主机上最多可同时存在 2 项 pam-crash 作业。

相似地，您可能想要阻止某一队列执行 pam-crash 作业，例如，由于它是有内存和 CPU 时间限制的特快队列，不适合 pam-crash。在这种情况下，您只须在队列配置中将相应的容量设置为 0，如图 8-13 所示。

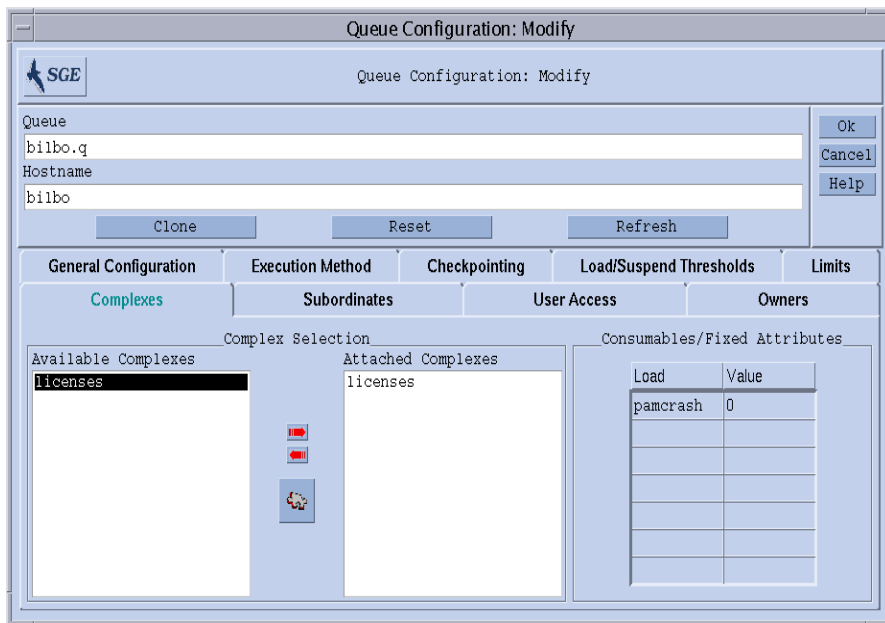


图 8-13 队列配置 — pam-crash

注意 – pam-crash 属性对队列默认可用，因为 global 属性组的属性可由所有队列继承。

示例 2：虚拟内存的空间共享

系统管理员的一项常见任务就是以某种方式调节系统，以避免由于内存过度预订以及随之出现的计算机内存交换导致性能降低。Sun Grid Engine 软件可通过“可使用资源”功能支持您执行这项任务。

标准负荷参数 `virtual_free` 报告可用的空闲虚拟内存，即可用内存交换空间加上可用物理内存。要避免内存交换，则必须将内存交换空间的使用最小化。理想情况下，主机上运行的所有进程所需的所有内存应符合物理内存大小。

在满足以下假定和配置的情况下，Sun Grid Engine 软件可对所有通过它启动的作业确保这一点。

- `virtual_free` 配置为可使用资源，并且其在每台主机上的容量均设置为可用物理内存量（或更低）。
- 作业请求其预期的内存用量，并且在运行时间内不会超出该请求值。

一个可能的主机属性组配置的例子如图 8-8 所示，并且相应的 1 GB 主内存的执行主机配置如图 8-9 所述。

注意 – 与前面全局属性组配置示例中的强制相反，主机配置示例中的可否请求标志设置为是。这意味着用户无须指明其作业的内存需求，而是使用缺省值字段中的值，如果没有明确的内存请求的话。这种情况下，缺省请求值为 1 GB 意味着没有请求值的作业假定为占用所有可用物理内存。

注意 – `virtual_free` 是 Sun Grid Engine 的标准负荷参数之一。Sun Grid Engine 在规划虚拟内存容量时，将自动考虑最近内存统计数据的附加可用内存。若空闲虚拟内存的负荷报告低于 Sun Grid Engine 内部簿记获取的值，则将使用负荷值以避免内存过度预订。若不是用 Sun Grid Engine 来启动作业，则报告的负荷值和 Sun Grid Engine 内部簿记很容易出现差异。

若您在一台计算机上运行多个类别的作业（其内存需求各不相同），则您可能想要将这台计算机的内存进行分区，以用于这些作业类别。这项功能（通常称作*空间共享*）可通过为每一作业类别配置一个队列，并将该主机上一定比例的总内存量指定给它来实现。

在本例中，图 8-14 所示的队列配置将主机 `bilbo` 的一半可用内存总量 (500 MB) 赋予队列 `bilbo.q`。因此，队列 `bilbo.q` 中执行的所有作业的累积内存使用量不能超过 500 MB。其它队列中的作业并不考虑在内，但主机 `bilbo` 上所有运行作业的内存使用总量仍然不能超过 1 GB。

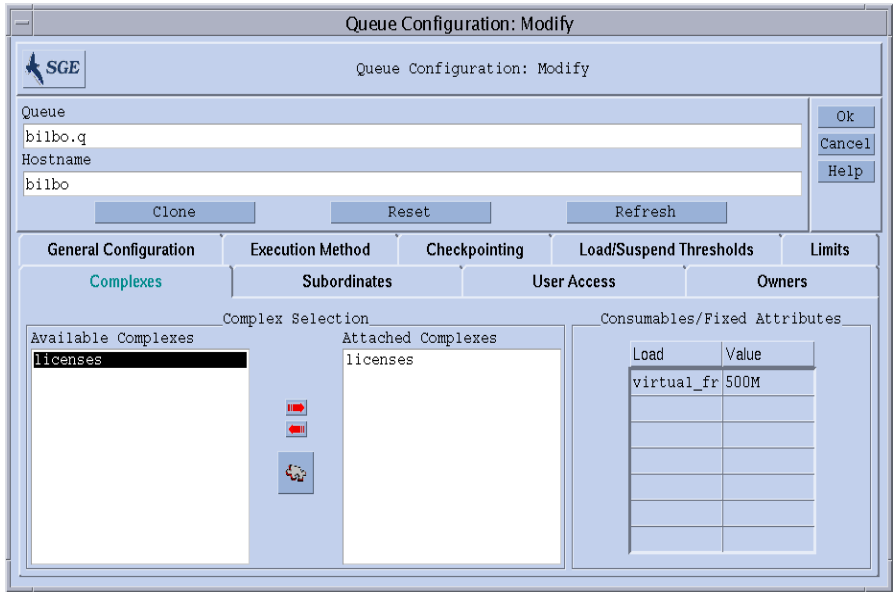


图 8-14 队列配置 — virtual_free

注意 – 属性 virtual_free 可通过从主机属性组继承而用于所有队列。

用户可通过以下任一格式将作业提交到与示例中的配置类似的系统：

```
% qsub -l vf=100M honest.sh
% qsub dont_care.sh
```

一旦有不少于 100 MB 的可用内存，即可启动由第一条命令提交的作业，且此内存容量将计算在 virtual_free 可使用资源的容量规划内。仅当系统上无其它作业运行时才运行第二个作业，因为其默认请求所有可用内存。此外，该作业将不能在队列 bilbo.q 中运行，因为它超过了队列的内存容量。

示例 3: 管理可用磁盘空间

某些应用程序需要操作存储在文件中的大型数据集，且因此在其运行时间内始终依赖充足磁盘空间的可用性。此需求类似于前面示例中讨论的可用内存的空间共享。主要区别在于 Sun Grid Engine 并未将空闲磁盘空间作为其标准负荷参数之一来提供。这是由于磁盘通常以站点特有的方式分区成为文件系统，无法自动识别所关心的文件系统。

不过，可用磁盘空间可由 Sun Grid Engine 通过可使用资源功能进行有效管理。推荐使用主机属性组属性 `h_fsize` 来达到此目的，其原因将在后面的章节中说明。首先，该属性必须配置为可使用资源，例如，如图 8-15 所示。

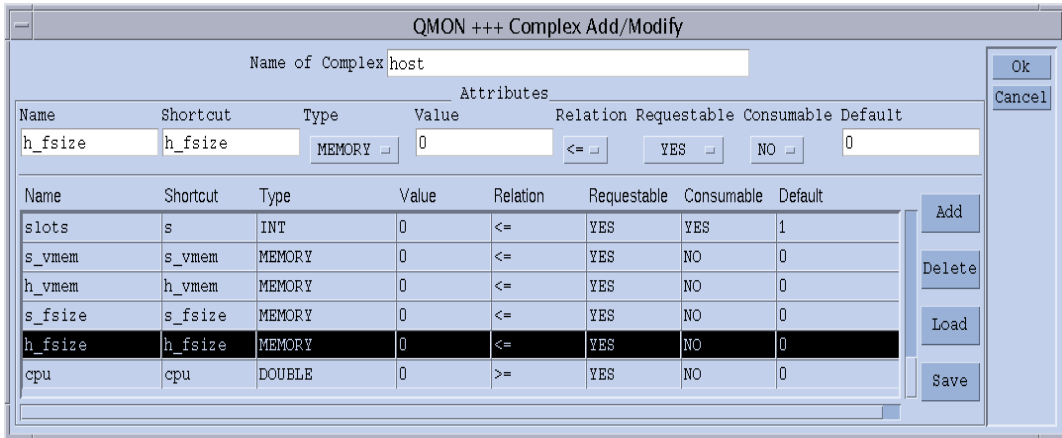


图 8-15 属性组配置 — `h_fsize`

对于主机本地文件系统而言，如图 8-16 所示将磁盘空间可使用资源的容量定义置于主机配置中是合理的。

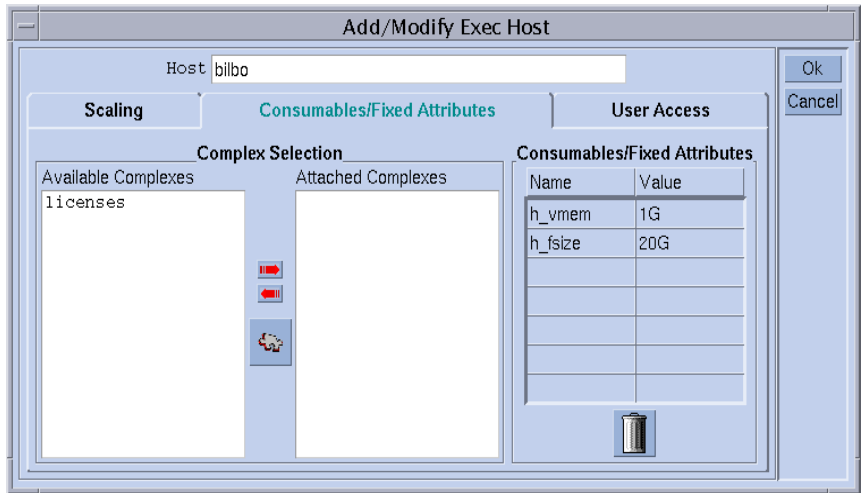


图 8-16 执行主机配置配置 — h_fsize

将作业提交到以这种方式配置的 Sun Grid Engine 系统中与前面的示例方式类似：

```
% qsub -l hf=5G big_sort.sh
```

本例中推荐 h_fsize 属性的原因是 h_fsize 也用作队列配置中的 *硬性文件大小限制*。文件大小限制用于在作业提交过程中限制作业创建大于指定大小的文件（上例中为 20 GB），或若作业未请求该属性时限制队列配置中的相应数值。本例中，h_fsize 的可否请求标志已设置为强制，因此总是提出请求。

通过将队列限制用作可使用资源，我们自动获得用户指定的请求（相对于作业脚本所用实际资源）的控制。违背该限制将受到制裁，且作业最终将中止（请参见 queue_conf 和 setrlimit 手册页以获得细节）。这种方式可确保基于 Sun Grid Engine 内部容量规划的资源请求是可靠的。

注意 – 某些操作系统只提供基于进程的文件大小限制。这种情况下，一项作业可能创建多个大小达到上限的文件。但是，在支持基于作业的文件大小限制的系统上，Sun Grid Engine 将此功能与 h_fsize 属性结合使用（请参见 queue_conf 手册页，以获得进一步细节）。

若您期望不同时将应用程序提交到 Sun Grid Engine 以占用磁盘空间，则 Sun Grid Engine 内部簿记可能不足以阻止由于缺少磁盘空间导致的应用程序故障。定期接收磁盘空间用量统计信息有助于避免此问题，该统计信息将指明磁盘空间使用总量，其中包括出现在 Sun Grid Engine 以外的那些。

Sun Grid Engine 负荷传感器接口（请参见第 190 页的“添加特定于站点的负荷参数”）允许您用特定于站点的信息来改进标准 Sun Grid Engine 负荷参数的设置，例如特定文件系统上的可用磁盘空间。

通过添加适当的负荷传感器和报告 `h_fsize` 的空闲磁盘空间，您可以将可使用资源管理与资源可用性统计信息结合起来。Sun Grid Engine 将把作业的磁盘空间需求与来自 Sun Grid Engine 内部资源规划的可用容量及最近报告的负荷值进行比较。仅当两项标准均符合时才将作业分派给主机。

配置属性组

Sun Grid Engine 属性组可通过 QMON 属性组配置对话框以图形方式进行定义和维护（如第 168 页的“如何添加或修改属性组配置”一节中的图示和说明），也可通过命令行执行。

▼ 如何从命令行修改属性组配置

输入以下命令及其适当的选项。

```
% qconf 选项
```

请参考《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中的属性组项，或属性组手册页，以获得 `qconf` 命令格式和有效值字段语法的详细定义。

有用的选项如下。

- `-ac`
- `-mc`
- `-Ac`
- `-Mc`

`qconf -Ac` 和 `-Mc` 选项将属性组配置作为自变量，而 `-ac` 和 `-mc` 选项启动一个编辑器，其中显示模板属性组配置或现有属性组配置以供修改。

选项的含义如下。

- `qconf -Ac, -ac`
将新属性组添加到可用属性组列表。
- `qconf -Mc, -mc`
修改现有属性组。

qconf 命令示例

以下命令：

```
% qconf -sc licenses
```

将 `nastran` 属性组（如图 8-5 中所定义）以 `complex (5)` 手册页中所定义的文件格式打印到标准输出流。`licenses` 属性组的输出示例如表 8-1 所示。

| # 名称 | 缩写名 | 类型 | 值 | 关系 | 可否请求 | 可否使用 | 缺省值 |
|-----------|-----|-----|----|----|------|------|-----|
| nastran | na | INT | 10 | <= | 是 | 否 | 0 |
| pam-crash | pc | INT | 15 | <= | 是 | 是 | 1 |
| permas | pm | INT | 40 | <= | 强制 | 是 | 1 |

#---- # 是注释行的起始符，但注释行在编辑时并不保存 -----

表 8-1 qconf -sc 输出示例

负荷参数

本节说明 Sun Grid Engine 5.3 负荷参数的概念，包括有关如何写您自己的负荷传感器的指导。

缺省负荷参数

缺省情况下，`sgc_execd` 定期向 `sgc_qmaster` 报告几个负荷参数和相应的值。它们存储于 `sgc_qmaster` 内部主机对象中（请参见第 131 页的“关于守护程序和主机”一节）。不过，它们仅当定义了相应名称的属性组属性后才在内部使用。这样的属性组属性包含诸如如何解释负荷参数的定义（请参见第 169 页的“属性组类型”一节，以获得细节）。

完成主要安装后，会报告一组标准负荷参数。标准负荷参数的所有所需属性均在主机属性组中定义。Sun Grid Engine 的后续版本可能会提供一组扩展的缺省负荷参数。因此，缺省情况下报告的这组负荷参数记录在文件 `<sgc 根目录>/doc/load_parameters.asc` 中。

注意 – 定义了负荷参数的属性组决定这些参数的访问权限。在全局属性组中定义负荷参数，可使其在整个群集和所有主机中均可用。在主机属性组中定义它们，则将这些属性提供给所有主机但并非群集全局。在用户定义的属性组中定义它们，则可控制负荷参数的可见性，方法是将用户属性组附加到主机或从中分离。

注意 – 负荷属性不应在队列属性组中定义，因为它们既不能用于任何主机，又不能用于群集。

添加特定于站点的负荷参数

缺省的负荷参数组可能不足以全面描述群集中的负荷情况，尤其是涉及到站点特有的策略、应用程序和配置时。因此，Sun Grid Engine 软件提供了以任意形式扩展这组负荷参数的方法。为此，`sge_execd` 提供了一个接口以将负荷参数及当前负荷值提供给 `sge_execd`。然后，这些参数即被当作缺省负荷参数。与缺省负荷参数一样（请参见第 189 页的“缺省负荷参数”一节），相应的属性需要在负荷参数的符合属性组中定义才能生效。

▼ 如何写您自己的负荷传感器

要向 `sge_execd` 提供附加的负荷信息，您必须提供一个 *负荷传感器*。负荷传感器可以是一个脚本或二进制可执行程序。在任一情况下，它对标准输入和输出流及其控制流的处理均必须遵循以下规则：

负荷传感器必须写成无限循环，在某一时刻等待来自 STDIN 的输入。若从 STDIN 读到字符串 `quit`，则负荷传感器应该退出。一旦从 STDIN 读取到一个行尾符，就应启动一个负荷数据检索循环。接下来，负荷传感器执行必要的操作以计算所需负荷数字。循环结束时，负荷传感器将结果写到 `stdout`。

规则

格式如下：

- 负荷值报告的起始行只有一个单词 `begin`。
- 各负荷值以新行分隔。
- 每一负荷值信息包含由冒号 (`:`) 分隔的三个组成部分，而且不包含空格。
- 负荷值信息的第一部分要么是为其报告负荷的主机名，要么是专用名 `global`。

- 第二部分为负荷值的符号名称，它是在主机或全局属性组列表中定义的（请参见《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册》中的 complex(5) 项以获得细节）。若所报告的负荷值在主机或全局属性组列表中不存在对应项，则不使用报告的负荷值。
- 第三部分是检测到的负荷值。
- 负荷值报告以只含单词 end 的行作为结尾。

脚本示例

代码示例 8-1 是 Bourne shell 脚本负荷传感器的一个示例。

```
#!/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
```

代码示例 8-1 Bourne Shell 脚本负荷传感器

若已将本例保存到文件 load.sh 中，并且用 chmod 赋予了其可执行权限，则可以从命令行交互测试它，测试方法是：调用 load.sh 并反复按键盘上的回车键。

一旦此过程起作用，您就可以为任何执行主机安装它，方法是：将负荷传感器的路径配置为群集全局或执行主机专用配置的 load_sensor 参数（请参见第 144 页的“基本群集配置”一节或 sge_conf 手册页）。

相应的 QMON 屏幕可能看起来与图 8-17 中的示例类似。

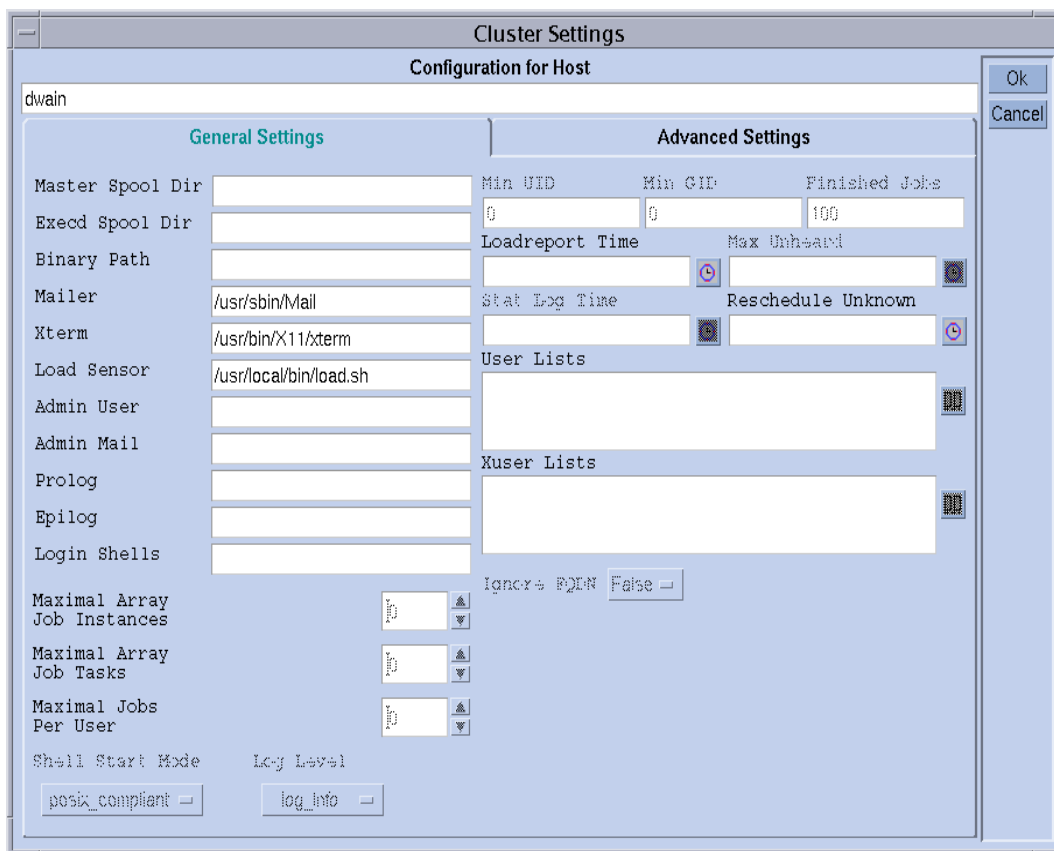


图 8-17 带负荷传感器的本地配置

一旦相应的属性添加到主机属性组中，所报告的负荷参数 `logins` 就变为可用。所需的定义可能与图 8-18 (QMON 属性组配置屏幕的一个示例) 中的最后一个表项类似。

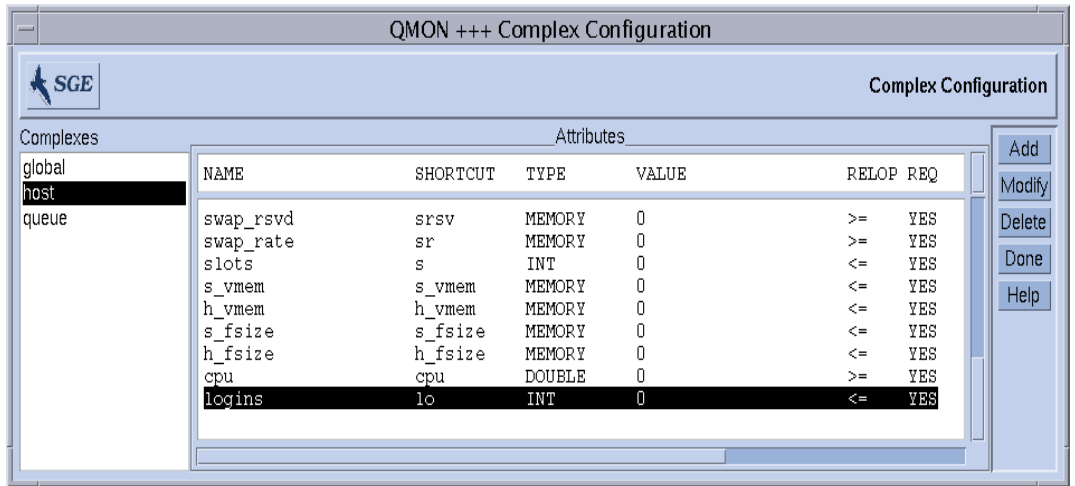


图 8-18 “属性组配置”对话框 — logins

管理用户访问权限和策略

本章包括 Sun Grid Engine 系统中与管理用户、相关帐户管理和策略有关的重要信息。本章的主题包括用户访问权限、调度、路径别名、缺省请求、帐户和用量统计信息以及对点检查的支持。

除背景信息外，本章还包括有关如何完成以下任务的详细指导。

- 第 197 页的 “如何用 QMON 配置帐户”
- 第 197 页的 “如何用 QMON 配置管理人员帐户”
- 第 198 页的 “如何从命令行配置管理人员帐户”
- 第 199 页的 “如何用 QMON 配置操作人员帐户”
- 第 200 页的 “如何从命令行配置操作人员帐户”
- 第 201 页的 “如何用 QMON 配置用户访问列表”
- 第 203 页的 “如何从命令行配置用户访问列表”
- 第 209 页的 “如何用 QMON 更改调度程序配置”
- 第 217 页的 “如何用 QMON 配置点检查环境”
- 第 221 页的 “如何从命令行配置点检查环境”

关于设置用户

下面的列表说明设置 Sun Grid Engine 用户的必要 / 可用任务：

- 登录要求
为了从主机 *A* 提交作业以在主机 *B* 上执行，用户在主机 *A* 和 *B* 上必须有相同的帐户（即相同的用户名）。不必登录到运行 `sgemaster` 的主机。
- 设置 Sun Grid Engine 访问权限
Sun Grid Engine 软件能够限制用户对整个群集、队列及并行环境的访问权限。请参见第 201 页的 “关于用户访问权限” 一节，以获得详细说明。

此外，Sun Grid Engine 系统用户可获得暂停或启用某些队列的权限（请参见第 160 页的“如何配置“拥有者””，以获得更多信息）。

■ 文件访问限制

Sun Grid Engine 用户需要对 `<sgc 根目录>/cell/common` 目录具有读取访问权限。

Sun Grid Engine 作业启动之前，Sun Grid Engine 执行守护程序（以 `root` 用户身份运行）为该作业创建一个临时工作目录，并将该目录的拥有权更改为作业拥有者（一旦作业完成就删除该临时目录）。该临时工作目录创建于由队列配置参数 `tmpdir` 定义的路径下（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `queue_conf` 项，以获得更多信息）。

请确保临时目录创建于 `tmpdir` 位置下，设置为 Sun Grid Engine 用户拥有并且用户以后可以对该临时目录执行写操作。

■ 站点从属性

依据定义，批处理作业没有终端连接。因此，命令解释程序启动资源文件（例如用于 `csh` 的 `.cshrc`）中的 UNIX 命令（如 `stty`）可能会导致出错。请如第 42 页的“如何验证安装”所述，检查是否有类似情形并避免使用这样的命令。

由于 Sun Grid Engine 批处理作业通常脱机执行，所以只有两种方法可以向作业拥有者通知错误事件及类似情况。一种方法是将错误消息记录到文件，另一种方法是发送电子邮件（e-mail）。在某些极罕见的情况下（例如无法打开错误日志文件），则电子邮件成为直接通知用户的唯一方式（无论如何，这类的错误消息仍记录到 Sun Grid Engine 系统日志文件，但通常用户不会去查看系统日志文件）。因此，若 Sun Grid Engine 用户正确安装电子邮件系统会很有裨益。

■ Sun Grid Engine 定义文件

您可以为 Sun Grid Engine 用户设置以下定义文件。

- `qmon`（Sun Grid Engine GUI 的资源文件；请参见第 8 页的“自定义 QMON”一节）
- `sgc_aliases`（当前工作目录路径别名；请参见第 212 页的“关于路径别名”一节）
- `sgc_request`（缺省请求定义文件；请参见第 214 页的“关于配置缺省请求”一节）。

关于用户访问权限

Sun Grid Engine 5.3 系统中存在三类用户。

- **管理人员** – 管理人员可以对 Sun Grid Engine 进行全面操控。缺省情况下，主控主机及队列所在的任何计算机的超级用户均有管理人员权限。
- **操作人员** – 操作人员可执行许多与管理人员相同的命令，但不能添加、删除或修改队列。
- **拥有者** – 队列拥有者只限于暂停/取消暂停或禁用/启用其所拥有的队列。这些权限对 `qidle` 的成功使用是必要的。用户通常声明为位于其桌面工作站上的队列的拥有者。

每一种类都将在后续章节中详尽讲述。

▼ 如何用 QMON 配置帐户

1. 在 QMON 主菜单中，按下“用户配置”按钮。
2. 根据您想要执行的操作，按下以下选项卡选择器之一。
 - 管理人员账户配置（请参见图 9-1）
 - 操作人员账户配置（请参见图 9-2）
 - 用户访问权限列表配置（请参见图 9-3）
3. 根据以下各节的指导继续进行。

注意 – 缺省情况下，第一次按下“用户配置”按钮时，会打开“管理人员帐户配置”对话框。

▼ 如何用 QMON 配置管理人员帐户

当选择“管理人员”选项卡时，会显示“管理人员配置”对话框（请参见图 9-1），可以在此声明哪些帐户允许执行所有 Sun Grid Engine 管理命令。屏幕下半部分的选择列表显示已声明为有管理权限的帐户。

- **删除** – 从该列表中删除现有管理人员帐户的方法是：单击其名称然后按下对话框右边的“删除”按钮。

- 添加 – 添加新管理人员帐户的方法是：在选择列表之上的输入窗口输入其名称，然后按下“添加”按钮或按下键盘上的回车键。

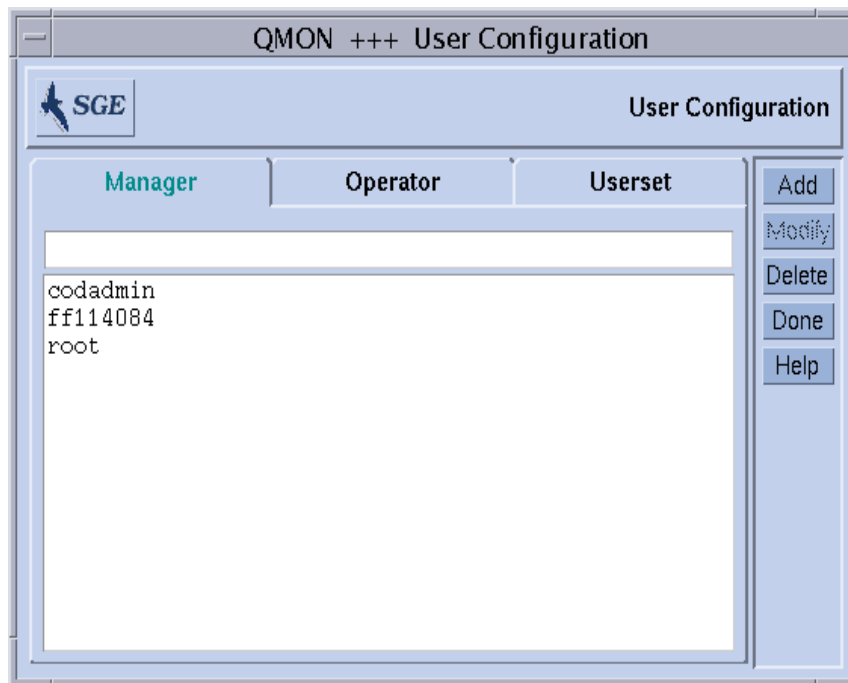


图 9-1 管理人员配置对话框

▼ 如何从命令行配置管理人员帐户

- 请输入以下命令及其适当开关选项。

```
# qconf 开关选项
```

可用开关选项

- `qconf -am 用户名 [...]`

添加管理人员 – 此命令将一位或多位用户添加到 Sun Grid Engine 管理人员列表。缺省情况下，所有 Sun Grid Engine 受托主机的 root 账户（请参见第 131 页的“关于守护程序和主机”一节）均为 Sun Grid Engine 管理人员。

- `qconf -dm 用户名 [...]`

删除管理人员 – 此命令从 Sun Grid Engine 管理人员列表删除指定用户。

- `qconf -sm`

显示管理人员 – 此命令显示所有 Sun Grid Engine 管理人员的列表。

▼ 如何用 QMON 配置操作人员帐户

当选择“操作人员”选项卡时，会显示“操作人员配置”对话框（请参见图 9-2），可以在此声明哪些帐户允许有受限的 Sun Grid Engine 管理命令权限的帐户（除非也声明为管理人员帐户。请参见第 197 页的“如何用 QMON 配置管理人员帐户”）。屏幕下半部分的选择列表显示已声明为提供操作人员权限的帐户。

- 删除 – 从该列表中删除现有操作人员帐户的方法是：单击其名称，然后按下对话框右边的“删除”按钮。
- 添加 – 添加新操作人员帐户的方法是：在选择列表之上的输入窗口输入其名称，然后按下“添加”按钮或按下键盘上的回车键。

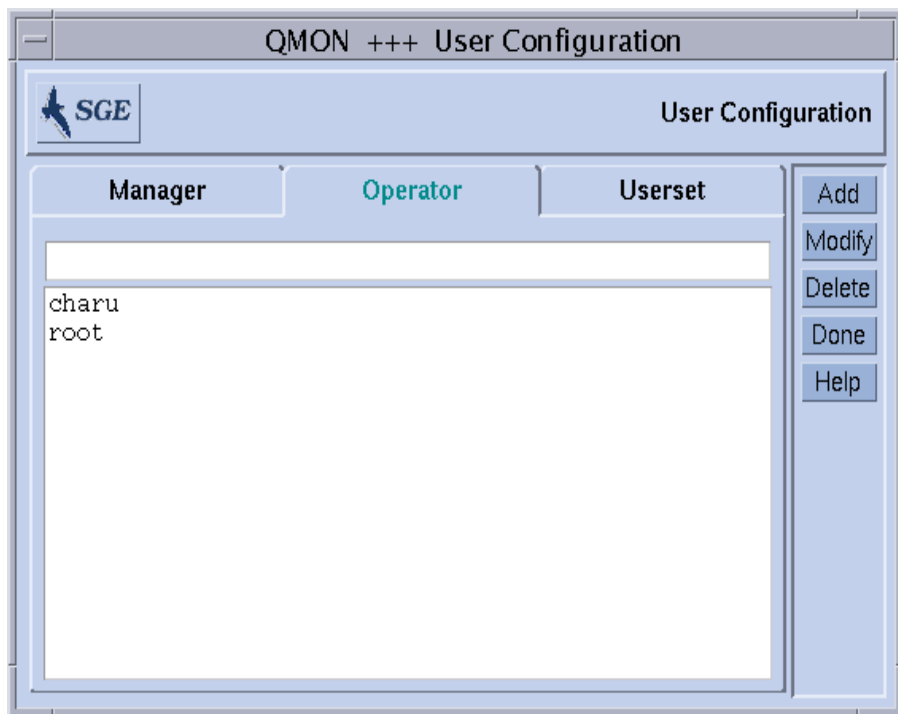


图 9-2 操作人员配置对话框

▼ 如何从命令行配置操作人员帐户

- 请输入以下命令及其适当开关选项。

```
# qconf 开关选项
```

可用开关选项

- `qconf -ao 用户名 [...]`
添加操作人员 – 此命令将一位或多位用户添加到 Sun Grid Engine 操作人员列表。
- `qconf -do 用户名 [...]`
删除操作人员 – 此命令从 Sun Grid Engine 操作人员列表删除指定用户。
- `qconf -so`
显示操作人员 – 此命令显示所有 Sun Grid Engine 操作人员的列表。

关于队列拥有者帐户

队列拥有者是在配置或修改 Sun Grid Engine 队列的过程中定义的。请参考第 150 页的“如何用 QMON 配置队列”和第 161 页的“如何从命令行配置队列”这两节。队列的拥有者可执行以下操作。

- **暂停** – 停止队列中所有正运行作业的执行并关闭队列。
- **取消暂停** – 恢复队列中作业的执行并打开队列。
- **禁用** – 关闭队列，但不影响正运行的作业。
- **启用** – 打开队列。

注意 – 队列暂停时被明令暂停的作业在队列取消暂停时将不会继续执行。它们需要明令取消暂停。

通常情况下，若用户不时需要某些计算机执行重要工作，并且他们深受运行于后台的 Sun Grid Engine 作业影响，则这些用户会被设置为某些队列的拥有者。

关于用户访问权限

任何在至少一台提交主机和执行主机上具有有效登录身份的用户均可使用 Sun Grid Engine 系统。不过，Sun Grid Engine 管理人员可以限制某些用户对某些或所有队列的访问权限。除此以外，还可以限制诸如特定并行环境之类工具的使用（请参见第 223 页的“关于并行环境”一节）。

为了定义访问权限，必须定义*用户访问列表*（由指定的任意重叠或非重叠用户组组成）。用户名和 UNIX 组名可用于定义那些用户访问列表。于是，在群集配置中（请参见第 144 页的“基本群集配置”一节）、队列配置中（请参见第 158 页的“如何配置“从属队列””一节）或配置并行环境接口的处理中（请参见第 224 页的“如何用 QMON 配置 PE”一节），就可以使用用户访问列表来拒绝或允许对特定资源的访问。

▼ 如何用 QMON 配置用户访问列表

选择“用户组”选项卡时，会显示与图 9-3 中示例类似的“用户组配置”对话框。

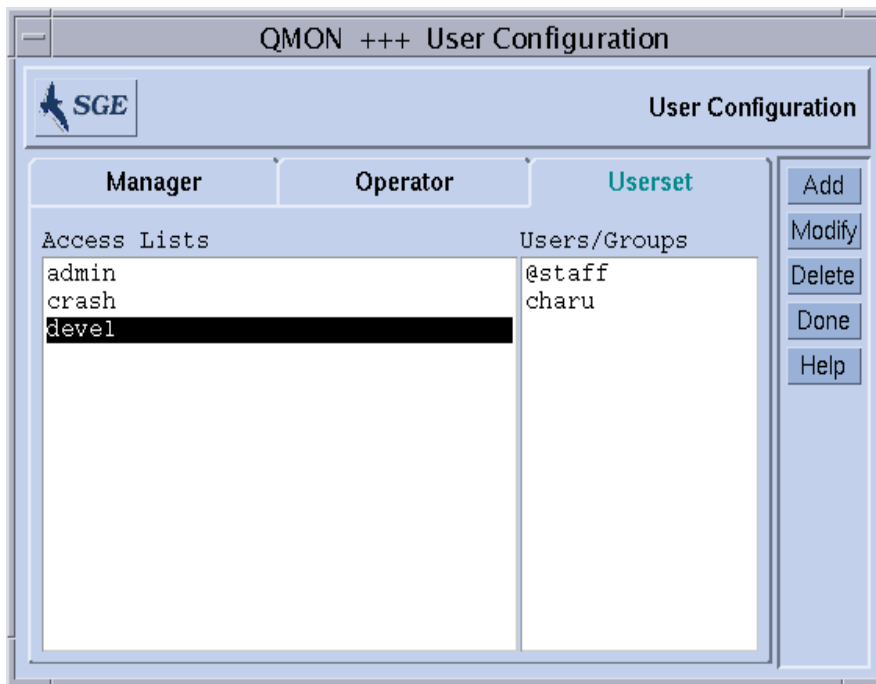


图 9-3 用户组配置对话框

屏幕左边的“用户组”选择列表中显示可用的访问列表。要在“用户/组”显示区域显示访问列表的内容，请在“访问列表”选择列表中单击它。

注意 – 组名的前缀 @ 符号可区分组与用户。

可使用“用户组配置”对话框执行以下任务。

- **删除** – 从“用户组”选择列表中删除现有访问列表的方法是：单击其名称然后按下对话框右边的“删除”按钮。
- **添加** – 添加一个新的访问列表，方法是：在选择列表之上的输入窗口输入其名称，然后按下“添加”按钮或按下键盘上的回车键。
- **修改** – 按“修改”按钮即可修改选定的访问列表。

在所有情况下，均打开“访问列表定义”对话框（与图 9-4 中所示类似）并提供相应的方式。

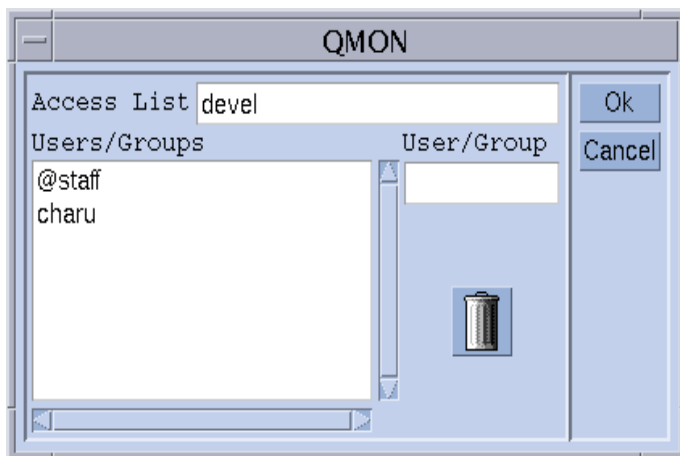


图 9-4 “访问列表定义”对话框

访问列表对话框窗口说明

- “用户组名”输入窗口 – 若执行的是修改操作，会显示所选访问列表的名称，或者，您可以用它来输入要声明的访问列表的名称。
- “用户/组”显示区域 – 包含目前为止定义的访问列表项。
- “用户/组”输入窗口 – 必须用此窗口将新项添加到访问列表。

所输入的用户名或组名（组名带有 @ 符号作为前缀）会在按下键盘上的回车键后，追加到“用户/组”显示区域。您可以通过选择相应项并按下垃圾桶图标按钮来删除它们。

一旦按下确定按钮，修改过的或新定义的访问列表就被注册，或者，若您按下“取消”按钮，则会放弃它们。这两种情况下，“访问列表定义”对话框均会关闭。

▼ 如何从命令行配置用户访问列表

- 请输入以下命令及其适当选项。

```
# qconf 开关选项
```

可用选项

- `qconf -au 用户名 [...]` 访问列表名 [...]
添加用户 – 此命令将一个或多个用户添加到指定的访问列表。
- `qconf -du 用户名 [...]` 访问列表名 [...]
删除用户 – 此命令从指定的访问列表中删除一个或多个用户。
- `qconf -su 访问列表名 [...]`
显示用户访问列表 – 此命令显示指定的访问列表。
- `qconf -sul`
显示用户访问列表 – 此命令显示当前已定义的所有访问列表清单。

关于调度

Sun Grid Engine 系统的作业调度活动包括以下几项。

- 预分配决策 – 这些活动涉及诸如排除执行队列（因其已满或超负荷）和将等待区域中当前无法执行的作业假脱机等等。
- 分配 – 这些活动涉及：确定一项作业的重要性（相对于所有其它暂挂和运行作业而言）、判断群集中所有计算机上的负荷、以及将作业发送到所选主机上的执行队列（主机是根据配置选择标准来选定的）。

Sun Grid Engine 软件基于以下各项在整个混合群集的计算机中调度作业。

- 群集的当前负荷
- 作业的资源需求（例如，CPU、内存和 I/O 带宽）

调度决策基于本站点的策略以及群集中每台计算机的即时负荷特征。站点的调度策略通过 Sun Grid Engine 系统的配置参数表达。负荷特征通过收集系统运行时的性能数据来确定。

调度策略

管理员可设置与以下 Sun Grid Engine 调度任务相关的策略。

- **队列排序** – 软件根据队列应填充的顺序排列群集中的队列。
- **作业排序** – 决定 Sun Grid Engine 系统尝试调度作业的顺序。

队列排序

可用以下方法确定 Sun Grid Engine 试图填充队列的顺序。

- **负荷报告** – Sun Grid Engine 管理员可选择用哪些负荷参数比较主机及其队列的负荷状态。各种可用的标准负荷参数和用站点专用负荷传感器来扩展此设置的接口均在第 189 页的“负荷参数”一节作了描述。
- **负荷调节** – 可规范来自不同主机的负荷报告以反映可比较的情况（请参见第 136 页的“如何用 QMON 配置执行主机”一节）。
- **负荷调整** – Sun Grid Engine 软件可配置为当作业分配到主机时，自动更正上次报告的负荷。更正后的负荷将体现由最近启动的作业所引起的预期负荷增长情况。当这些作业造成的负荷开始产生影响时，负荷的模拟增长可自动缩减。
- **序列号** – 队列可遵循严格顺序进行排序。

作业排序

Sun Grid Engine 开始分配之前，作业首先按最高优先级排序。接下来，Sun Grid Engine 将试图按优先级顺序为作业查找适当的资源。在没有管理员干预的情况下，顺序为先进先出 (FIFO)。管理员可通过以下方式控制作业顺序。

- **用户排序** – 若此调度选项生效，则不同用户的作业可交叉存取。即，所有用户提交的第一项作业一视同仁，然后是第二项，等等。
- **作业优先级** – 管理员可分配给作业一个优先级编号，这样就可以直接确定排列顺序。用户可以降低分配给他或她拥有的作业的优先级。
- **最大用户/组作业数** – 可限制用户或 UNIX 用户组拥有的在 Sun Grid Engine 系统上同时运行的作业的最大数目。这将影响到暂挂作业列表的排序顺序，因为将优先选择未超过其限制的用户的作业。

发生于调度间隔内的操作

调度程序按时间间隔进行调度。在两次调度操作之间，Sun Grid Engine 保留有关重大事件的信息，例如作业提交、作业完成、作业取消、群集配置的更新或群集中新计算机的注册。进行调度时，调度程序执行以下操作。

- 考虑所有重大事件。
- 根据管理员的规定对作业和队列排序。
- 考虑所有作业的资源需求。

然后，按照需要，Sun Grid Engine 系统执行以下操作。

- 分配新作业。
- 暂停正在执行的作业。
- 维持现状。

调度程序监视

若一项作业并未启动且原因不明，则您可对该作业执行带 `-w v` 选项的 `qalter` 命令。Sun Grid Engine 软件采用一个空群集，并检查是否有适合该作业的可用队列。

通过执行 `qstat -j 作业 ID` 可获得进一步信息。它将显示作业请求概况的摘要，其中还包括上次调度运行时未调度该作业的原因。不带作业 ID 执行 `qstat -j`，将总结上次调度时间间隔所有未调度作业的原因。

注意 – 调度原因信息的收集必须在调度程序配置 `sched_conf` 中开启。请参考相应的《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中的 `schedd_job_info` 参数或第 209 页的“如何用 QMON 更改调度程序配置”一节。

要检索 Sun Grid Engine 调度程序 `sgeschedd` 决策的更进一步细节，可使用 `qconf` 命令的 `-tsm` 选项。该命令将强制 `sgeschedd` 把追踪记录输出写入文件。

缺省调度

缺省的 Sun Grid Engine 调度为 *先进先出* 策略，即调度程序首先检查最先提交的作业，以便其分配到队列。若暂挂作业列表中的第一项作业找到合适的闲置队列，则在运行调度程序时最先启动该作业。仅当第一项作业未找到合适的空闲资源时，第二项作业或排列在其后的作业才可能在暂挂作业列表的第一项之前启动。

只要涉及到作业的队列选择，缺省 Sun Grid Engine 策略总是选择负荷最少的主机上的队列，只要其能针对作业资源需求交付适当的服务。若多个合适队列具有相同的负荷，则无法预测选中的队列。

调度方案

修改作业调度和队列选择策略的方式有多种。

- 更改调度算法
- 调节系统负荷
- 按序列号选择队列
- 按份额选择队列
- 限制每位用户或每组的作业数

以下各节探究这些方案。

更改调度算法

调度程序配置参数 `algorithm`（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `sched_conf` 项，以获得进一步信息）是为选择所用调度算法而设计的。`default` 是当前唯一允许的设置。

作业优先级

Sun Grid Engine 管理者可分配给在暂挂作业列表中假脱机的作业一个整数，称为作业优先级。作业优先级定义作业在暂挂作业列表中的位置。调度程序将首先检查最高优先级数字的作业。作业优先级的取值范围在 `-1024` 和 `1023` 之间，其中 `0` 为刚提交的新作业的优先级。若分配给某项作业一个负的优先级值，则该作业甚至排列在刚提交的新作业后面。若存在优先级相同的多项作业，则在此优先级类别内应用缺省的先进先出规则。

作业优先级通过以下命令分配给作业。

```
% qalter -p 优先级 作业 ID ...
```

在上述命令中，优先级指定分配到作业列表的优先级，它位于以空格分隔的作业 ID 列表尾部。

注意 – 应注意不要将术语“作业优先级”与优先级队列配置参数混淆（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `queue_conf` 手册页），后者定义某个特定队列中所有执行作业设置的系统调度优先级值。还请注意，`qstat` 输出中的第二栏显示当前分配给提交作业的优先级。

等份额排序

众所周知，上述的缺省先进先出调度策略会产生非常不公平的结果，比如当某一用户在短时间内接二连三地提交一系列作业时（例如通过使用脚本程序）。该用户的作业将在很长的时间内占用合适的资源，以至其他用户没有机会分配到这些队列。

在这种情况下，群集管理员可将调度策略更改为所谓的等份额排序。若此调度选项生效，并且用户在系统中已有一项运行的作业，则其所有其它作业均排列在同一优先级别的其它用户的作业之后（请参见上节以获得有关优先级别的细节）。

若调度程序配置参数 `user_sort` 设置为 TRUE（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `sched_conf` 手册页），则使用等份额排序。

调节系统负荷

Sun Grid Engine 系统使用队列所在计算机上的系统负荷信息来为作业选择执行队列。这种队列选择方案构建了一个负荷平衡的情况，因此保证了群集中可用资源的更好利用。

不过，系统负荷并非总是反映真实情况。例如，若一个多 CPU 的计算机与单 CPU 系统进行比较，多处理器系统通常会报告较高的数字，因为其很可能运行更多的进程，并且系统负荷的度量受试图访问 CPU 的进程数目的影响极大。不过，多 CPU 系统比单 CPU 计算机能够满足更高的负荷。使用以处理器数调整过的负荷值（缺省情况下由 `sge_execd` 报告）可解决此问题（请参见第 189 页的“负荷参数”一节和 `<sge 根目录>/doc/load_parameters.asc` 文件以获得细节）。使用这些负荷参数代替原始负荷值以避免上述问题。

负荷值可能解释不正确的另一个例子是，那些性能潜力或性价比差异很大的系统，对于它们来说，相同的负荷值并不意味着可选择任意主机来执行作业。这种情况下，Sun Grid Engine 管理员应定义相关执行主机及负荷参数的负荷调节系数（请参见第 136 页的“如何用 QMON 配置执行主机”及相关章节）。

注意 – 已调节的负荷参数还可与负荷阈值列表 *负荷阈值* 和 *迁移负荷阈值* 进行比较（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `queue_conf` 项以获得细节）。

另一个与负荷参数相关的问题是，需要对所有数值及其相对重要性进行解释，这种解释与应用程序和站点紧密相关。CPU 负荷可能对某一站点上常用的某种应用程序非常重要，而内存负荷则可能对另一个站点以及应用程序配置文件（该站点的运算群集所专注的）更为重要。要解决此问题，Sun Grid Engine 允许管理员在调度程序配置文件 (`sched_conf`) 中指定所谓的 *负荷公式*（请参考相应的《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》章节以获得更多细节）。

可将有关资源利用和容量规划的站点专用信息考虑在内，方法是：在负荷公式中使用站点定义的负荷参数（请参见第 190 页的“添加特定于站点的负荷参数”一节）和可使用资源（请参见第 177 页的“可使用的资源”一节）。

最后，还需要考虑负荷参数的时间依赖性。由系统上正运行的 Sun Grid Engine 作业所施加的负荷随时间而变化，并且经常需要一定的时间才能通过操作系统报告正确的数量（例如，对于 CPU 负荷）。因此，若一项作业是最近启动的，则报告的负荷可能不足以代表由该作业施加给主机的负荷。报告的负荷会随着时间的推移不断进行修改以反映真实的负荷，但是，如果在某段时间内报告的负荷过低，就可能已导致过度预订该主机。Sun Grid Engine 允许管理员指定 *负荷调整系数*（用在 Sun Grid Engine 调度程序中）以弥补此问题。请参考《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中有关调度程序配置文件 sched_conf 的处理，以获得有关如何设置这些负荷调整系数的详细信息。

按序列号选择队列

更改缺省队列选择方案的另一种方式是，将全局 Sun Grid Engine 群集配置参数 queue_sort_method 设置为 seq_no 以代替缺省的 load（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 sched_conf 项）。这种情况下，系统负荷不再是选择队列的主要方式。替代地，由队列配置参数 seq_no（请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 queue_conf 项）指定的序列号被用来定义一个固定的队列次序，即它们被选择的顺序（若其适用于所考虑的作业并且正空闲）。

若您的站点上提供批处理服务的计算机是按每项作业的固定价格排序的，则这种队列选择策略可能很有用处：例如，一项作业运行于计算机 A 上要花费 1 个单位的成本，而运行在计算机 B 上要花费 10 个单位的成本，并且运行在计算机 C 上要花费 100 个单位的成本。这样首选调度策略将是首先填满主机 A，然后是主机 B，仅当无其它选择时才使用主机 C。

注意 – 若已将队列选择方式改为 seq_no，并且所考虑的所有队列共享同一个序列号，则按缺省的 load 选择队列。

限制每位用户或每组的作业数

Sun Grid Engine 管理员可指定任何用户或 UNIX 组在任一时刻允许运行的最大作业数上限。为了强制执行此功能，可如《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 sched_conf 一节所述，设置 maxujobs 和 / 或 maxgjobs。

▼ 如何用 QMON 更改调度程序配置

1. 从 QMON 主菜单，单击“调度程序配置”图标。

显示“调度程序配置”对话框。该对话框分“常规参数”部分和“负荷调整”部分。根据您想要完成的任务选择其一。

- a. 要更改常规调度参数，请单击“常规参数”选项卡。

“常规参数”对话框与图 9-5 中的示例类似。

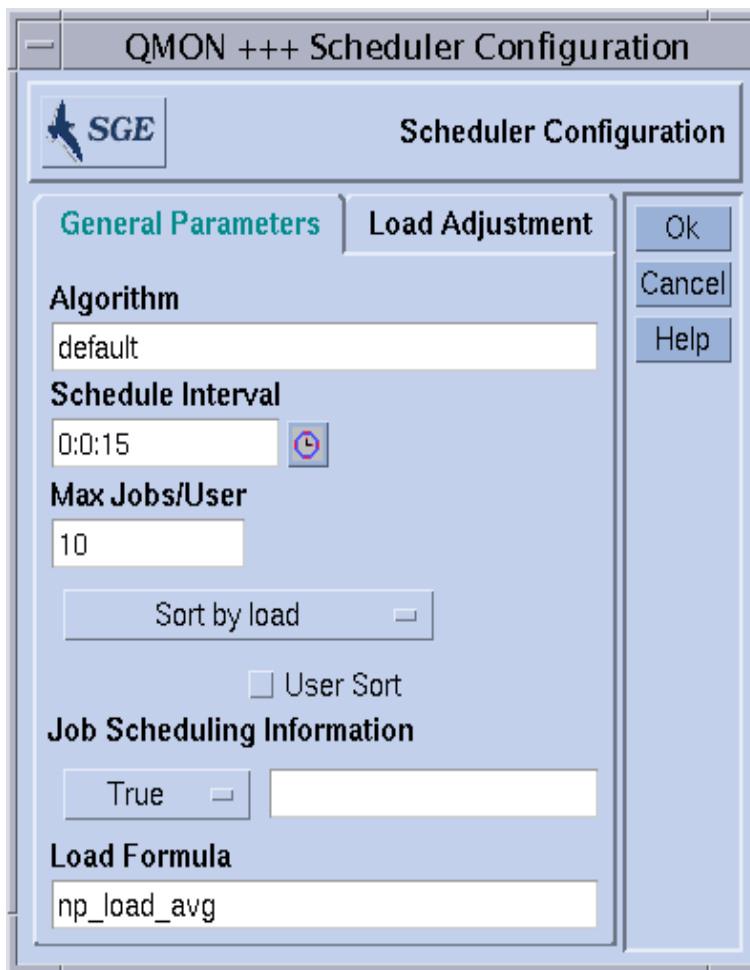


图 9-5 “调度程序配置”对话框 — 常规参数

您可以通过“常规参数”对话框设置以下参数。

- 调度算法（请参见第 206 页的“更改调度算法”）
 - 调度程序两次运行之间的常规时间间隔
 - 每位用户或每个 UNIX 组允许同时运行的最大作业数（请参见第 208 页的“限制每位用户或每组的作业数”）。
 - 队列排序方案 — 按负荷排序、或按序列号排序（请参见第 208 页的“按序列号选择队列”）。
 - 是否激活等份额排序（用户排序标志）（请参见第 207 页的“等份额排序”章节）。
 - 作业调度信息是否可通过 `qstat -j` 访问，或是否只应为附加的输入字段中所指定范围的作业 ID 收集此信息。建议：仅在暂挂作业数非常高的情况下，才临时开启作业调度信息的常规收集。
 - 用于对主机和队列进行排序的负荷公式
- b. 要更改负荷调整参数，请选择“负荷调整”选项卡。

“负荷调整参数”对话框与图 9-6 中的示例类似。

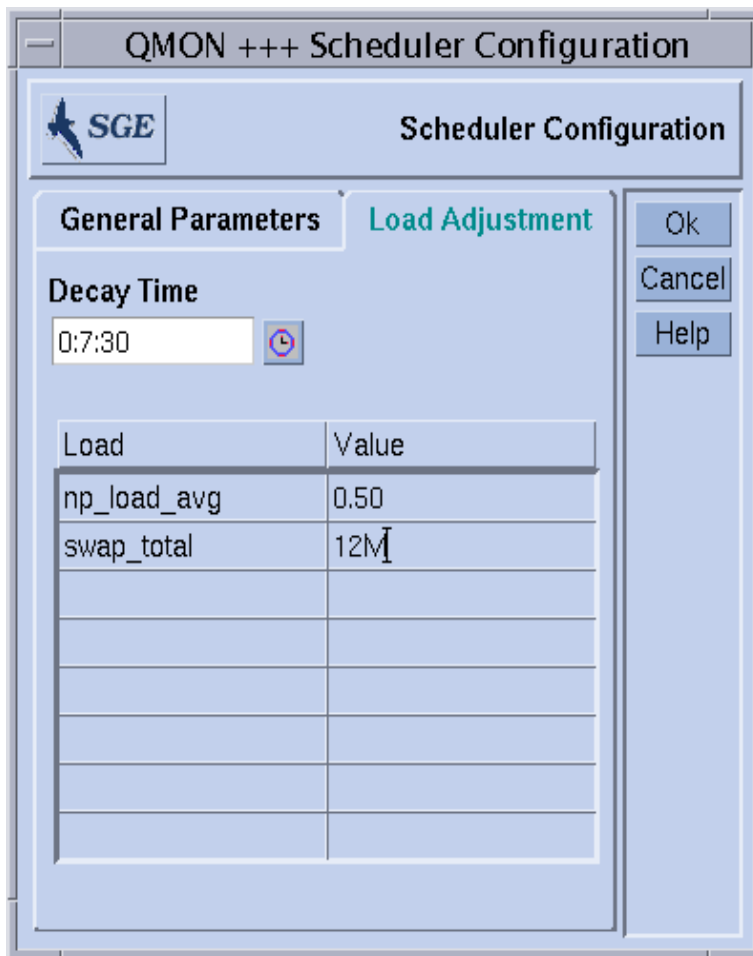


图 9-6 “调度程序配置”对话框 — 负荷调整

“负荷调整”对话框允许您定义以下参数。

- 负荷调整衰减时间
- 一个位于对话框下半部分的负荷调整值表格，它列出了当前已为其定义了调整值的所有负荷和可使用属性。该列表可通过单击顶端的“负荷”或“值”按钮进行改进。这将打开一个选择列表，其中列出所有附加到主机的属性（即在全局属性组、主机属性组和管理员定义的属性组中配置的所有属性）。“属性选择”对话框如图 6-6 所示。选择某个属性并按下确定按钮确认所作的选择，即可将该

属性添加到“可使用/固定属性”表格的“负荷”栏，并将光标置于相应的“值”字段。双击“值”字段即可修改现有值。选择相应的表格行，然后键入 CTRL-D，或单击鼠标右键打开一个删除对话框，即可删除一个属性。

请参见第 207 页的“调节系统负荷”，以获得背景信息。请参考《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》中的 sched_conf 手册页，以获得有关调度程序配置的进一步细节。

关于路径别名

在 Solaris 和其它联网的 UNIX 环境下，若用户可通过 NFS 访问，则其经常在不同的计算机上有相同的主目录（或其一部分）。不过，有时主目录路径并非在所有计算机上都完全相同。

例如，考虑通过 NFS 和自动装入程序均可用的用户主目录。若用户在 NFS 服务器上有主目录 /home/foo，他就可以在所有已正确安装 NFS 且运行自动装入程序的客户机上访问此路径下的主目录。不过，务必要注意客户机上的 /home/foo 只是一个到 /tmp_mnt/home/foo（即 NFS 服务器上自动装入程序物理安装目录的实际位置）的符号链接。

在这种情况下，若用户在客户机上从主目录树内某处提交作业，并使用 qsub -cwd 标志（在当前工作目录下执行作业），则 Sun Grid Engine 系统在试图从执行主机（若该主机为 NFS 服务器）上查找当前工作目录时可能出现这个问题。这是因为 qsub 命令将到达提交主机的当前工作目录并获取 /tmp_mnt/home/foo/（因为这是提交主机上的物理位置）。此路径将传递给执行主机，若执行主机是物理主目录路径为 /home/foo 的 NFS 服务器，则无法解析。

其它通常导致类似问题的情况是，在不同计算机上具有不同装入点路径的固定（非自动装入的）NFS 装入（例如，一台主机上的装入主目录位于 /usr/people 下，而在另一台上位于 /usr/users 之下）或从外部到网络可用文件系统的符号链接。

为了避免这些问题，Sun Grid Engine 软件允许管理和用户配置路径别名文件。这两个文件的位置如下。

- <sgc 根目录>/<单元>/common/sgc_aliases — 群集全局路径别名文件。
- \$HOME/.sgc_aliases — 特定于用户的路径别名文件。

注意 – 群集全局文件只应由合格的管理员进行修改。

文件格式

两个文件格式相同。

- 忽略空行和以 # 符号开头的行。
- 除了空行和以 # 开头的行之外，每行必须包含四个字符串，字符串之间以任何数目的空格或制表符隔开。
第一个字符串指定源路径，第二个为提交主机，第三个为执行主机，而第四个为源路径的替代路径。
- 提交和执行主机项均可只包含一个 * 符号，它与任何主机相匹配。

如何解释路径别名文件

文件解释如下。

- 在 qsub 检索到当前工作目录的物理路径之后，会读取群集全局路径别名文件（如果有的话）。然后会读取用户路径别名文件，就好像它是追加到全局文件上的一样。
- 从文件顶端开始逐行读取不应忽略的行，而那些行所指定的转换信息会存储起来（如有必要）。
- 仅当提交主机项与执行 qsub 命令的主机相匹配，以及当源路径组成当前工作目录或已存储的源路径替代路径的开头部分时，才存储转换信息。
- 一旦读取了这两个文件，存储的路径别名信息就随提交的作业一起传递。
- 在执行主机上，别名信息将得到评估。当路径别名的执行主机项与执行主机相匹配时，当前工作目录的开头部分将由源路径的替代路径所取代。请注意，这种情况下当前工作目录字符串将改变，并且其后的路径别名必须匹配要应用的已更改工作目录路径。

路径别名文件示例

以下是通过别名文件项解决上述 NFS/ 自动装入程序问题的示例。

```
# cluster global path aliases file
# src-path      subm-host  exec-host      dest-path
/tmp_mnt/      *              *              /
```

路径别名文件示例

关于配置缺省请求

批处理作业通常由 Sun Grid Engine 系统根据用户为该作业定义的请求概况来分配给队列。用户将成功运行作业所需满足的一组请求汇集起来，而 Sun Grid Engine 调度程序只考虑满足此作业的该组请求的队列。

若用户并未对作业指定任何请求，则调度程序将考虑用户有权访问的所有队列，而不另加限制。不过，Sun Grid Engine 软件允许配置缺省请求，它可为作业定义资源需求，即使用户并未明确指定请求。

既可为 Sun Grid Engine 群集的所有用户全局配置缺省请求，也可专为任何用户配置缺省请求。缺省请求配置存放在缺省请求文件中。全局请求文件位于 `<sge 根目录>/<单元>/common/sge_request` 下，而用户专用请求文件（称为 `.sge_request`）可位于该用户的主目录下或 `qsub` 命令执行的当前工作目录下。

若有这些文件，则每项作业均用其评估。评估顺序如下：

1. 全局缺省请求文件
2. 用户主目录下的用户缺省请求文件
3. 当前工作目录下的用户缺省请求文件

注意 – 作业脚本中指定的或随 `qsub` 命令行提供的请求比缺省请求文件中的请求优先级高（请参见第四章，以获得有关如何明确请求作业资源的细节）。

注意 – 可通过使用 `qsub -clear` 选项防止缺省请求文件的无意识影响，该选项放弃所有以前的需求说明。

缺省请求文件的格式

本地和全局缺省请求文件的格式如以下列表所述。

- 缺省请求文件可包含任意数目的行。忽略空行和以 `#` 符号开头的行。
- 不忽略的每一行均可包含任意 `qsub` 选项，如《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中所述。允许每行有多个选项。批处理脚本文件和批处理脚本的自变量选项不视作 `qsub` 选项，因此在缺省请求文件中不允许使用。

- `qsub -clear` 选项放弃当前评估的请求文件或以前处理的请求文件中所有以前指定的需求。

缺省请求文件的示例

例如，假定用户的本地缺省请求文件配置与代码示例 9-1 中的脚本 `test.sh` 相同。

```
# 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
```

代码示例 9-1 缺省请求文件的示例

要执行该脚本，用户将输入以下命令。

```
% qsub test.sh
```

执行 `test.sh` 脚本的效果与用户在命令行中如下所示指定所有 `qsub` 选项相同。

```
% qsub -l arch=solaris64,s_cpu=5:0:0 -cwd test.sh
```

注意 – 与通过 `qsub` 提交的批处理作业类似，通过 `qsh` 提交的交互式作业也会考虑缺省请求文件。通过 `QMON` 提交的交互式或批处理作业也会考虑这些请求文件。

关于收集帐户信息和利用统计信息

Sun Grid Engine 命令 `qacct` 可用于生成文字和数字混合的账户统计信息。若不带开关选项调用，则 `qacct` 显示 Sun Grid Engine 群集中所有主机的累计使用情况，它由所有已完成并包含在群集帐户文件 (`<sgc 根目录>/<单元>/common/accounting`) 中的作业生成。这种情况下，`qacct` 只报告以下三个时间（单位为秒）：

- 真实 — 计时时间，即从作业开始到作业结束之间的时间。

- 用户 — 此为用户进程花费的 CPU 时间。
- 系统 — 此为系统调用花费的 CPU 时间。

有几个开关选项可用于报告有关所有队列或某些队列、所有用户或某些用户等等诸如此类的统计信息。尤其是，可请求有关所有已完成并匹配资源需求说明的作业的信息，资源需求说明是通过 `-l` 语法来表达的，该语法与使用 `qsub` 命令提交作业相同。请参考《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中的 `qacct` 项，以获得更多信息。

`qacct` 选项 (`-j [作业ID|作业名称]`) 可用于直接访问由 Sun Grid Engine 系统存储的详尽资源用量信息，包括 `getrusage` 系统调用提供的信息（请参考《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中相应的项）。

此选项报告作业 ID 为 `[作业ID]` 或作业名称为 `[作业名称]` 的作业的资源用量项。若未给出自变量，则显示所有包含在所提及帐户文件中的作业。若选择了一个作业 ID，且显示了多项，则表明此 ID 号代表一组作业 ID（作业 ID 范围从 1 到 999999），或所显示的是已迁移的点检查作业。

关于点检查支持

点检查是一种功能，可冻结某一执行作业或应用程序的状态，将该状态（所谓的检查点）保存到磁盘，之后若作业或应用程序因发生其它故障（例如，由于系统关闭）无法完成时，可从该检查点重新启动。若检查点可从一台主机移动到另一台，则点检查可用于在群集环境中迁移应用程序或作业，而无需考虑计算资源的损失。因此，可借助点检查功能来实现动态负荷平衡。

Sun Grid Engine 系统支持两种级别的点检查。

■ 用户级别点检查

在此级别，提供点检查生成机制完全是用户或应用程序的责任。用户级别的点检查示例包括：

- 在应用程序中编码的重新启动文件以卓越的算法步骤定期写入，并且在应用程序重新启动时正确处理这些文件
- 点检查库的使用，它需要与应用程序链接从而安装点检查机制。

注意 – 许多第三方应用程序提供基于重新启动文件的写入而集成式检查点功能。检查点库可从公用域（例如，请参考 University of Wisconsin 的 *Condor* 项目）或从硬件供应商处获得。

■ 内核级别透明点检查

此级别的点检查必须由操作系统（或其扩充工具）提供，它潜在地可应用于任意作业。使用内核级别点检查无需更改源代码或重新链接应用程序。

内核级别点检查可应用于整个作业（即由作业创建的进程分层结构），而用户级别的点检查通常限于单个程序。这样一来，这种程序所嵌入的作业就需要适当地处理整个作业重新启动的情况。

内核级别点检查（与基于点检查库的点检查一样）要使用大量资源，因为在进行点检查时作业或应用程序所占用的全部虚拟地址空间需要转储到磁盘上。与此相反，基于重新启动文件的用户级别点检查可将写入检查点的数据仅限于重要信息。

点检查环境

为了反映不同类型的点检查方法，以及这些方法在不同操作系统体系结构中的潜在派生种类，Sun Grid Engine 为所用的每个点检查方法提供了配置属性说明。

此属性说明称为点检查环境。缺省的点检查环境随 Sun Grid Engine 发行软件一起提供，并可根据站点的需要进行修改。

新的点检查方法原则上可集成，但这可能是一项颇具挑战性的任务，并且只应由有经验的工作人员或 Sun Grid Engine 支持小组来执行。

▼ 如何用 QMON 配置点检查环境

1. 从 QMON 主菜单，单击“点检查配置”图标。

显示“点检查配置”对话框，与图 9-7 中的示例类似。

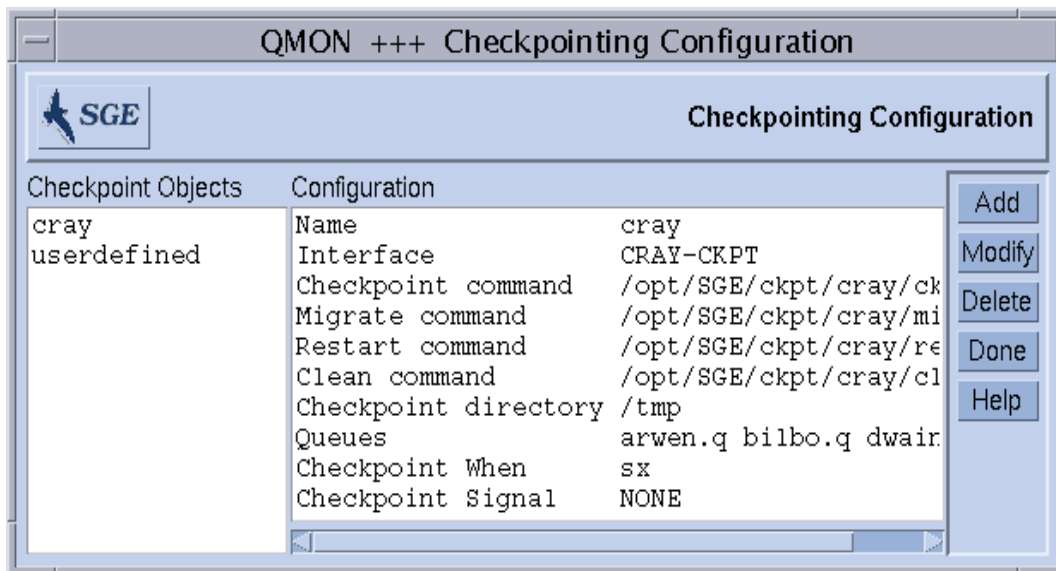


图 9-7 “点检查配置”对话框

2. 根据您想要完成的任务，从“点检查配置”对话框执行以下操作之一。

查看已配置的点检查环境

- 要查看以前配置的点检查环境，请选择列于“点检查对象”栏中的点检查环境名之一。

相应的配置将显示于“配置”栏中。

删除已配置的点检查环境

- 要删除已配置的点检查环境，从“点检查对象”栏高亮显示其名称并按“删除”。

修改已配置的点检查环境

1. 在“点检查对象”栏中，高亮显示您要修改的点检查环境名并按下“修改”。
出现“添加 / 修改点检查对象”对话框以及所选点检查环境的当前配置，与图 9-8 中示例类似。

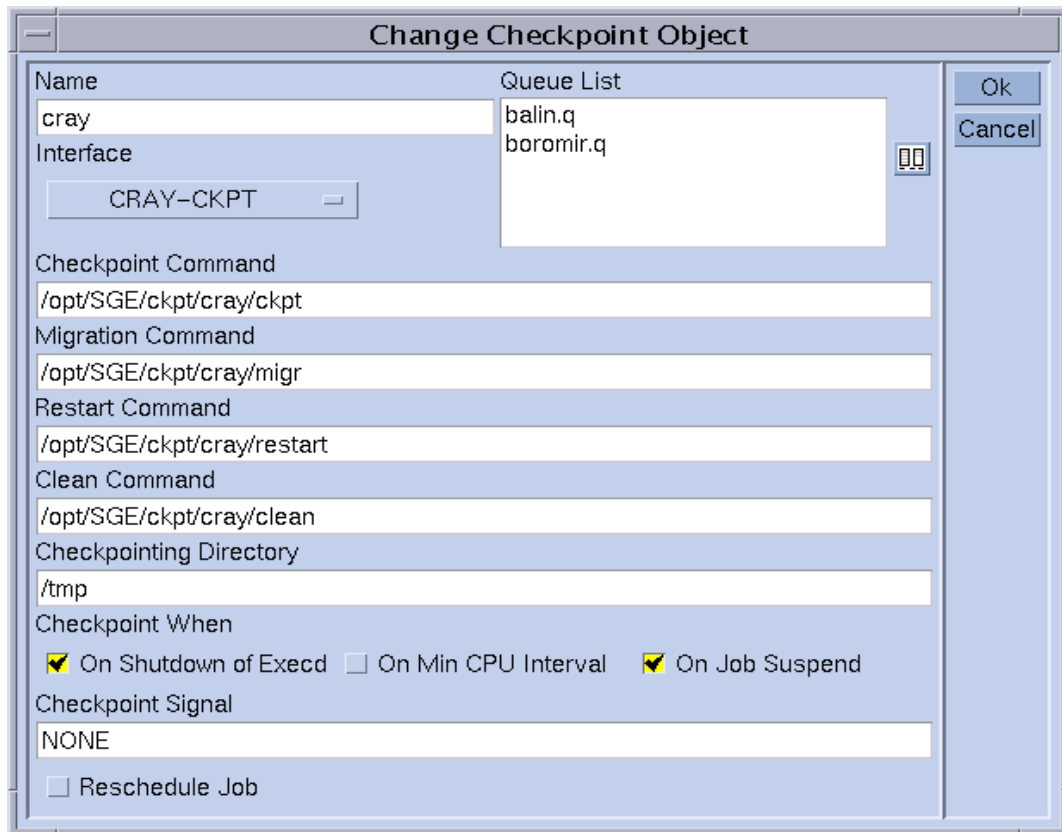


图 9-8 “添加 / 修改点检查对象”对话框

2. 根据下列指导修改选定的点检查环境。
“添加 / 修改点检查对象”对话框可用来更改下列各项。
 - 名称
 - 点检查、迁移、重新启动、清除命令行字符串
 - 存放点检查文件的目录
 - 必须启动点检查的情况
 - 启动点检查时发送给作业或应用程序的信号

注意 – 请参考 《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》中的 checkpoint 项，以获得有关这些参数的细节。此外，您必须定义要使用的接口（也称作点检查方法）。选择相应选择列表中的一项，并请参考 checkpoint 项，以获得有关不同接口的含义的细节。

3. **重要** – 对于随 Sun Grid Engine 发行软件一起提供的点检查环境，只能更改名称、点检查目录和队列列表参数。

要更改队列列表参数，请转到“步骤 a”。否则，跳过“步骤 a”并转到“步骤 4”。

a. 单击“队列列表”窗口右边的图标（请参见图 9-8）。

显示“选择队列”对话框，与图 9-9 中的示例类似。

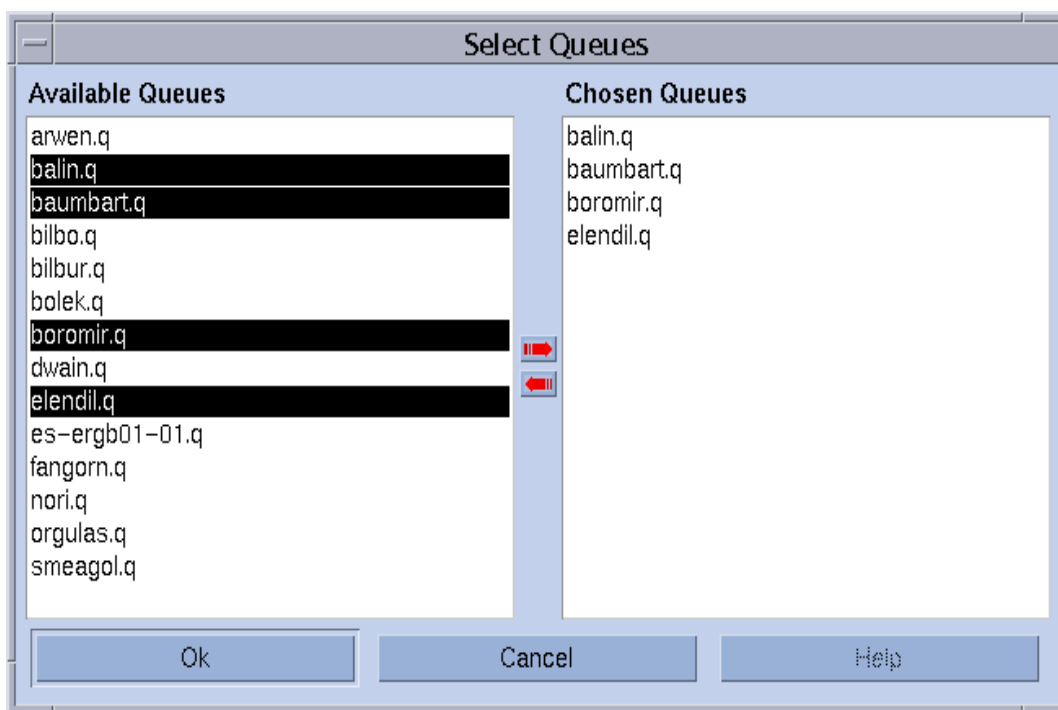


图 9-9 “点检查队列选择”对话框

b. 从“可用队列”列表中选择您想要包含在点检查环境中的队列，并将其添加到“选定的队列”列表中。

c. 按下“确定”。

按下“确定”会将这些队列输入到“添加 / 修改点检查对象”对话框的“队列列表”窗口。

4. 按下“确定”向 `sgc_qmaster` 注册所作的更改，或按下“取消”放弃所作的更改。

添加点检查环境

1. 在“点检查配置”对话框中，单击“添加”。
显示与图 9-8 中所示类似的“添加 / 修改点检查对象”对话框，以及一个您可以编辑的模板配置。
2. 将所要求的信息填入模板。
3. 按下“确定”向 `sgc_qmaster` 注册所作的更改，或按下“取消”放弃所作的更改。

▼ 如何从命令行配置点检查环境

- 按照以下各节的指导，输入 `qconf` 命令及其适当的选项。

qconf 点检查选项

- `qconf -ackpt` 点检查名称

添加点检查环境 — 此命令启动一个带点检查环境配置模板的编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量所对应的编辑器）。参数 *点检查名称* 指定点检查环境的名称，并已填入模板的相应字段。更改模板并将其保存到磁盘，即可配置点检查环境。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `checkpoint` 项，以获得要更改模板项的详细说明。

- `qconf -dckpt` 点检查名称

删除点检查环境 — 此命令删除指定的点检查环境。

- `qconf -mckpt` 点检查名称

修改点检查环境 — 此命令启动一个以指定点检查环境作为配置模板的编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量所对应的编辑器）。更改模板并将其保存到磁盘，即可修改点检查环境。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `checkpoint` 项，以获得要更改模板项的详细说明。

- `qconf -sckpt` 点检查名称

显示点检查环境 — 此命令将指定点检查环境的配置显示到标准输出。

- `qconf -sckptl`

显示点检查环境列表 — 此命令显示所有当前已配置的点检查环境的名称列表。

管理并行环境

本章包括有关管理和控制并行环境的信息。

除了有关这些主题的背景信息外，本章还包括完成以下任务的详细说明。

- 第 224 页的“如何用 QMON 配置 PE”
 - 第 224 页的“显示 PE 内容”
 - 第 224 页的“删除 PE”
 - 第 224 页的“修改 PE”
 - 第 225 页的“添加 PE”
 - 第 228 页的“如何从命令行配置 PE”
 - 第 228 页的“如何从命令行显示已配置的 PE 接口”
 - 第 229 页的“如何用 QMON 显示已配置的 PE 接口”
-

关于并行环境

并行环境(PE) 是一个软件包，它是专为在联网环境或并行平台上进行并行运算而设计的。在过去的几年中，许多系统已发展成为能够采用可行的技术，在不同硬件平台上进行分布式和并行处理。当今最常用的两个消息传递环境的例子是 PVM（并行虚拟机，Oak Ridge National Laboratories）和 MPI（消息传递接口，Message Passing Interface Forum）。这两个工具既有公用的域，也有供应商提供的硬件设备。

所有这些系统显示不同特性并有其各自的要求。为了可以处理运行在这种系统上的任意并行作业，Sun Grid Engine 系统提供了一个灵活而强大的接口以满足各种需要。

任意 PE 均可由 Sun Grid Engine 接合，只要分别如第 230 页的“PE 启动过程”和第 231 页的“终止 PE”中所述执行了适当的启动和停止步骤。

▼ 如何用 QMON 配置 PE

1. 从 QMON 主菜单，单击“PE 配置”按钮。

出现“并行环境配置”对话框，与图 10-1 中的示例类似。

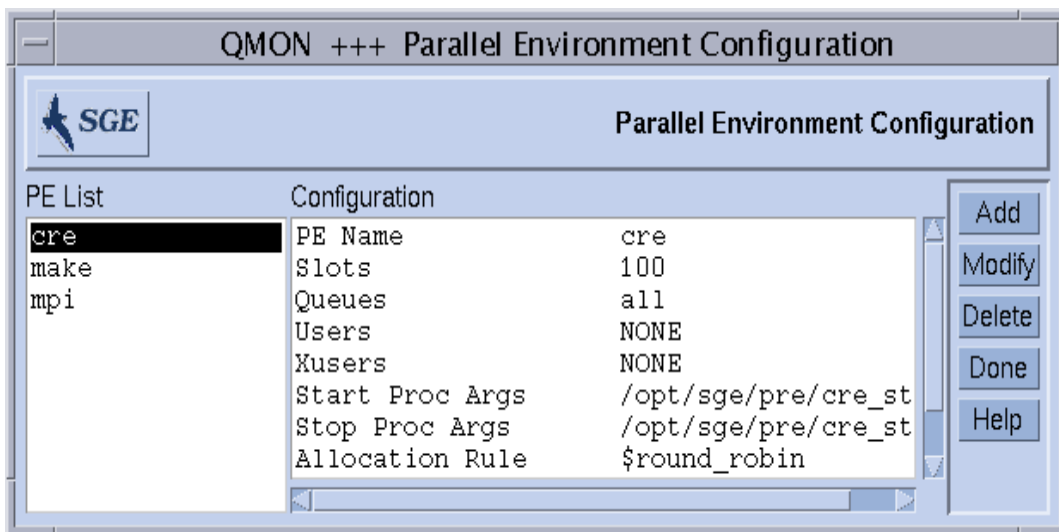


图 10-1 “并行环境配置”对话框

已配置的 PE 显示在屏幕左边的“PE 列表”选择列表中。

2. 根据您想要完成的任务，从“并行环境配置”对话框执行以下操作之一。

▼ 显示 PE 内容

- 要显示 PE 内容，在“PE 列表”选择列表中单击其名称。
PE 配置的内容会显示在“配置”显示区域。

▼ 删除 PE

- 要删除选定的 PE，在“PE 列表”选择列表中高亮显示其名称，然后按“删除”按钮（位于窗口的右边）。

▼ 修改 PE

1. 要修改选定的 PE，请按“修改”按钮。
出现“PE 定义”对话框，与图 10-2 中所示的示例类似。
2. 根据第 225 页的“并行环境定义参数说明”一节中的指导，修改 PE 定义。

- 按“确定”以保存更改，或按“取消”放弃更改。
无论是按“确定”还是“取消”均会关闭该对话框。

▼ 添加 PE

- 要添加新的 PE，请按“添加”按钮。
出现“PE 定义”对话框，与图 10-2 中所示的示例类似。

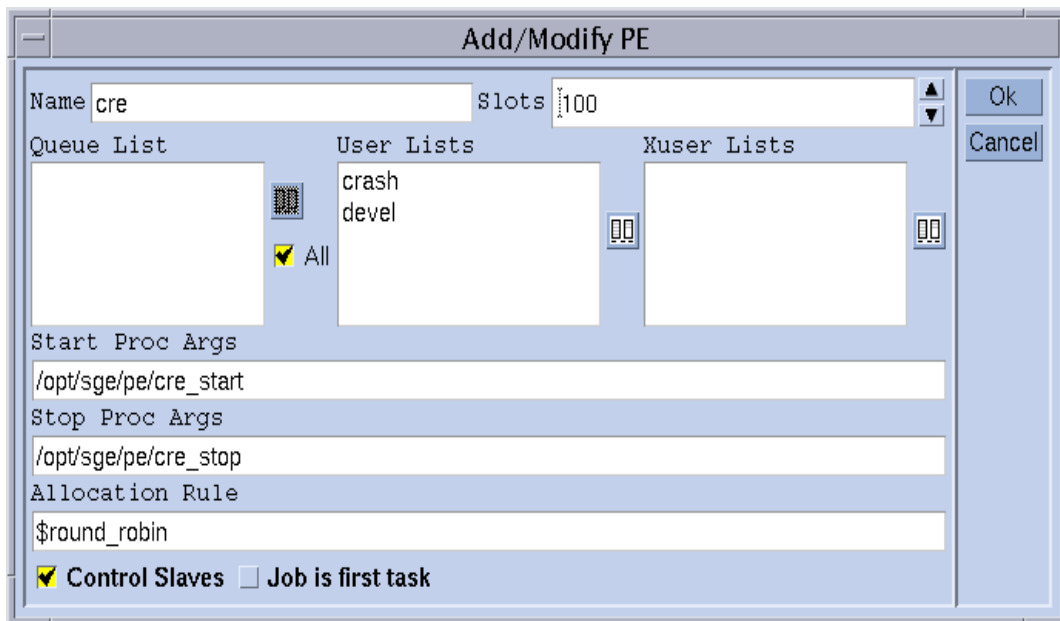


图 10-2 “并行环境定义”对话框

- 根据第 225 页的“并行环境定义参数说明”一节中的指导，添加 PE 定义。
- 按“确定”以保存更改，或按“取消”放弃更改。
无论是按“确定”还是“取消”均会关闭该对话框。

并行环境定义参数说明

- 名称输入窗口显示选定的 PE 名称（如果是在执行修改操作），或者可用于输入要声明的 PE 名称。
- 位置数 数字调节框用于输入可由所有并行运行的 PE 作业占用的总作业位置数。

- **队列列表** 显示区域显示 PE 可用的队列。单击“队列列表”显示区域右边的图标按钮，会出现一个“选择队列”对话框（与图 10-3 中的示例类似），可供您修改 PE 队列列表。

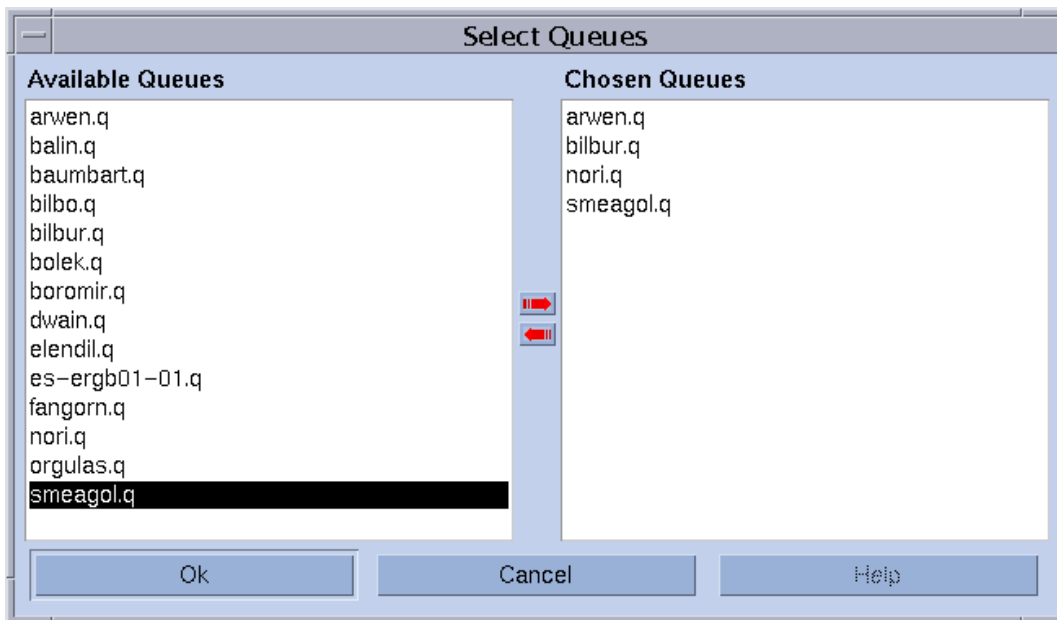


图 10-3 “选择队列”对话框

- **有权限用户列表** 显示区域包含有权访问 PE 的用户访问列表（请参见第 201 页的“关于用户访问权限”一节）。
- **无权限用户列表** 显示区域显示那些无权访问的访问列表。

单击与这两个显示区域相关联的图标按钮，会出现“选择访问列表”对话框，它与图 10-4 中的示例类似。使用这些对话框可修改这两个访问列表显示区域的内容。

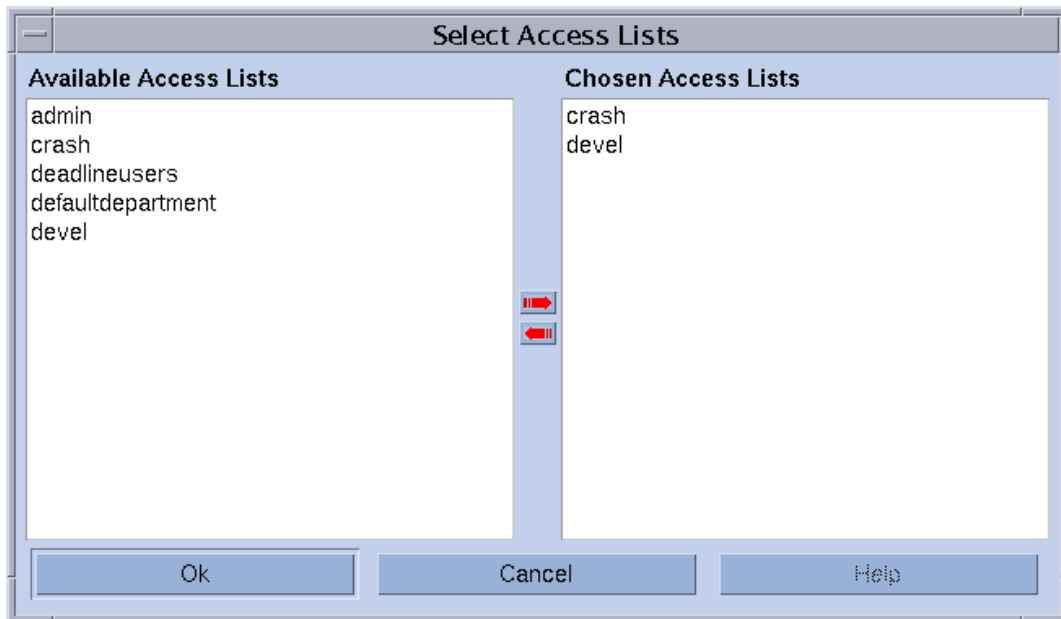


图 10-4 “选择访问列表”对话框

- **启动过程自变量**和**停止过程自变量**输入窗口可用于输入 PE 启动和停止过程的精确调用序列（请分别参见第 230 页的“PE 启动过程”和第 231 页的“终止 PE”这两节）。第一个自变量通常为启动或停止过程本身。其余的参数为这些过程的命令行自变量。

可使用一些特殊标识符（以 \$ 前缀开头）将 Sun Grid Engine 内部运行时间信息传递给这些过程。《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `sgc_pe` 项包含了所有可用参数的列表。

- **分配规则**输入窗口定义分配给每台 PE 所使用的计算机的并行进程数目。当前只支持正整数和特殊值 `$pe_slots`。`$pe_slots` 表示所有创建的进程必须位于同一主机上。
- **控制从属任务**切换按钮声明是否通过 Sun Grid Engine（即 `sgc_execd` 和 `sgc_shepherd`）生成并行任务，或相应的 PE 是否创建其自身的进程。若 Sun Grid Engine 系统能完全控制从属任务（正确统计和资源控制）则很有裨益，但此功能仅适用于专为 Sun Grid Engine 定制的 PE 接口。请参考第 232 页的“PE 和 Sun Grid Engine 软件的紧密集成”一节，以获得进一步细节。

- 作业首先作为任务切换按钮仅当“控制从属任务”已开启时才有意义。它表示作业脚本或其子进程之一作为并行应用程序的一个并行任务（例如，通常 PVM 就属此情况）。若其关闭，则作业脚本启动并行应用程序但不参与（例如，在 MPI 中使用 mpirun 时的情况）。

▼ 如何从命令行配置 PE

按照以下各节的指导，输入带适当选项的 `qconf` 命令。

qconf PE 选项

- `qconf -ap 并行环境名`

添加并行环境 — 此命令启动一个带 PE 配置模板的编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量对应的编辑器）。参数 *并行环境名* 指定 PE 的名称，并已填入模板的相应字段。更改此模板并将其保存到磁盘，即可配置 PE。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `sge_pe` 项，以获得要更改的模板项的详细说明。

- `qconf -dp 并行环境名`

删除并行环境 — 此命令删除指定的 PE。

- `qconf -mp 并行环境名`

修改并行环境 — 此命令启动一个编辑器（缺省情况下为 `vi` 或 `$EDITOR` 环境变量对应的编辑器），其中显示的配置模板即为指定的 PE。更改模板并将其保存到磁盘，即可修改 PE。请参见《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》中的 `sge_pe` 项，以获得要更改的模板项的详细说明。

- `qconf -sp 并行环境名`

显示并行环境 — 此命令将指定 PE 的配置显示到标准输出。

- `qconf -spl`

显示并行环境列表 — 此命令显示所有当前已配置并行环境的名称列表。

▼ 如何从命令行显示已配置的 PE 接口

- 请输入以下命令。

```
% qconf -spl
% qconf -sp 并行环境名
```


第一行命令显示当前可用 PE 接口的名称列表。第二行命令显示特定 PE 接口的配置。请参考 `sgc_pe` 手册页，以获得有关 PE 配置的细节。

▼ 如何用 QMON 显示已配置的 PE 接口

- 在 QMON 主菜单中，按下“PE 配置”按钮。

会显示“并行环境配置”对话框（请参见第 224 页的“如何用 QMON 配置 PE”一节）。

第 74 页的“高级示例”一节中的示例定义了一个并行作业，它请求使用至少 4 个、最多（推荐使用）16 个进程的 PE 接口 `mpi`（即消息传递接口）。“并行环境 (PE) 规范”窗口右边的按钮可用于弹出一个对话框，可从此窗口的可用 PE 列表中选出所需的并行环境（请参见图 10-5）。由作业启动的并行任务数的所需范围可添加到“高级提交”屏幕的“PE 规范”窗口中 PE 名之后。

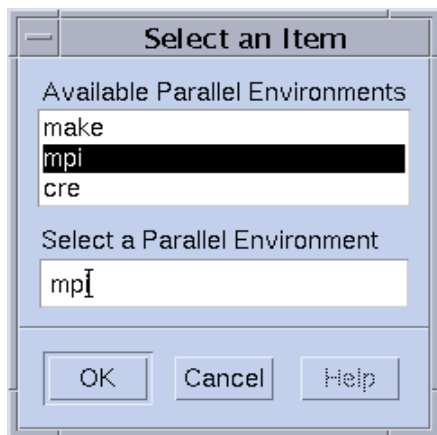


图 10-5 PE 选择

在第 84 页的“如何从命令行提交作业”一节中，给出了与上述并行作业指定相对应的命令行提交命令，并指明了如何用 `qsub -pe` 选项来表达同一请求。《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册*》中的 `qsub` 项提供了有关 `-pe` 语法的更多细节。

为并行作业选择适当的 PE 接口至关重要。PE 接口可能不使用或使用不同的消息传递系统；它们可将进程分配到单台或多台主机；可能拒绝某些用户访问 PE；PE 接口可能只使用一组专用的队列，并且 PE 接口在任一时间点可能只占据一定数目的队列位置。因此，您应该询问 Sun Grid Engine 管理者，以获知最适合您的并行作业类型的可用 PE 接口。

您可以如第 78 页的“资源需求定义”一节中所述，指定资源要求及 PE 请求。这将进一步缩小适合于 PE 接口的合格队列的范围，只包含那些同时也满足所指定的资源要求的队列。例如，假设您已提交了如下命令：

```
% qsub -pe mpi 1,2,4,8 -l nastran,arch=of nastran.par
```

适合于此项作业的队列是：那些通过 PE 配置与 PE 接口 mpi 相关联并满足由 qsub -l 选项指定的资源要求的队列。

注意 – Sun Grid Engine PE 接口程序具有高度可配置性。尤其是，Sun Grid Engine 管理者可配置 PE 启动和停止过程（请参见 `sge_pe` 手册页）以支持站点的特定需求。导出环境变量的 `qsub -v` 和 `-V` 选项可用于将信息从提交作业的用户传递到 PE 启动和停止过程。若需要导出某些环境变量，请咨询 Sun Grid Engine 管理员。

PE 启动过程

Sun Grid Engine 系统通过调用启动过程（通过 `exec` 系统调用）来启动 PE。该启动过程的可执行文件的名称以及传递给此可执行文件的参数，均可在 Sun Grid Engine 系统内配置。Sun Grid Engine 发行软件中包含了一个 PVM 环境的此类启动过程的示例。它由一个 shell 脚本和一个由 shell 脚本调用的 C 程序组成。该 shell 脚本使用 C 程序利索地启动 PVM。所有其它所需操作均由该 shell 脚本处理。

shell 脚本位于 `<sge 根目录>/pvm/startpvm.sh` 之下。C 程序文件可在 `<sge 根目录>/pvm/src/start_pvm.c` 之下找到。

注意 – 启动过程也可由单个 C 程序完成。shell 脚本用于方便地自定义启动过程示例。

示例脚本（`startpvm.sh`）需要以下这三个自变量。

- Sun Grid Engine 软件生成的主机文件的路径，其中包括将要启动 PVM 的主机名
- 调用 `startpvm.sh` 过程的主机
- PVM 根目录的路径（通常包含在 `PVM_ROOT` 环境变量中）

这些参数可通过如第 224 页的“如何用 QMON 配置 PE”中所述的方式传递给启动脚本。这些参数包含在由 Sun Grid Engine 在运行时间提供给 PE 启动和停止脚本的参数中。例如，所需的主机文件由 Sun Grid Engine 生成，并且文件名可通过特殊参数名（`$sge_hostfile`）传递给 PE 配置中的启动过程。《*Sun Grid Engine 5.3 和 Sun Grid Engine 5.3（企业版）参考手册*》的 `sge_pe` 项中给出了所有可用参数的说明。

主机文件的格式如下。

- 文件的每行均指一个将要运行并行进程的主机。
- 每行的第一项表示主机名。
- 第二项表示在该主机上运行的并行进程数。
- 第三项表示所用的处理器范围（针对多处理器计算机）。

此文件格式由 Sun Grid Engine 产生并且为固定格式。需要不同文件格式（例如，对于 PVM）的 PE 要在启动过程中进行转换（请参见 `startpvm.sh` 文件）。

一旦通过 Sun Grid Engine 系统启动了 PE 启动过程，它就会启动 PE。启动过程应以零退出状态退出。若启动过程的退出状态并非为零，则 Sun Grid Engine 软件报告出一条错误并且不启动并行作业。

注意 – 最好先从命令行测试所有启动过程（不使用 Sun Grid Engine），以清除所有错误。若该过程集成到 Sun Grid Engine 框架中，可能难以跟踪这些错误。

终止 PE

当一项并行作业完成或中止时（通过 `qdel`），将调用一个过程以停止并行环境。此过程的定义和语法与启动程序所述非常相似。停止过程也可在 PE 配置中定义（例如，请参见第 224 页的“如何用 QMON 配置 PE”）。

停止过程的用途是关闭 PE 并结束所有相关联的进程。

注意 – 若停止过程无法清除 PE 进程，则 Sun Grid Engine 系统可能没有任何有关运行在 PE 控制下的进程的信息，因此无法清除。当然，Sun Grid Engine 软件可清除与其启动的作业脚本直接相关联的进程。

Sun Grid Engine 分布树中还包含一个 PVM PE 的停止过程的示例。它位于 `<sge 根目录>/pvm/stoppvm.sh` 之下。它采用以下两个自变量。

- 由 Sun Grid Engine 系统生成的主机文件的路径
- 启动该停止过程的主机名

类似于启动过程，停止过程预期在成功时返回退出状态零，而失败时返回非零退出状态。

注意 – 最好先从命令行测试所有停止过程（不使用 Sun Grid Engine），以清除所有错误。若该过程集成到 Sun Grid Engine 框架中，可能难以跟踪这些错误。

PE 和 Sun Grid Engine 软件的紧密集成

在第 224 页的“如何用 QMON 配置 PE”一节中的对“控制从属任务”参数的说明中已提到，通过 Sun Grid Engine 组件 `sge_execd` 和 `sge_shepherd` 为其创建并行任务的 PE，比创建自身进程的 PE 更有益处。这是因为 UNIX 操作系统只允许进程分层结构的创建者进行可靠资源控制。诸如并行应用程序的正确统计、资源限制和进程控制等功能，仅可由所有并行任务的创建者来强制执行。

大多数 PE 不提供这些功能，并且因此不提供与资源管理系统（如 Sun Grid Engine）集成的合适接口。为了克服这个问题，Sun Grid Engine 系统提供了一个高级 PE 接口以便于与 PE 紧密集成，该接口将创建任务的职责从 PE 转交到 Sun Grid Engine 软件。

Sun Grid Engine 发行软件包含了两个这种紧密集成的示例，一个针对 PVM 公用域版本，而另一个针对来自 Argonne National Laboratories 的 MPICH MPI 工具。这些示例分别包含在 `<sge 根目录>/pvm` 和 `<sge 根目录>/mpi` 目录中。该目录包含用于进行比较的松散集成的接口的变体，以及说明用法和所有当前限制的 README 文件。请参考那些 README 文件，以获得进一步细节。

注意 – 执行与 PE 的紧密集成是一项高级任务，并且需要 PE 和 Sun Grid Engine PE 接口的专门知识。您可能需要与 Sun 技术支持代表联系以获得帮助。

错误消息

本章讲述 Sun Grid Engine 5.3 错误消息发送过程。使用此信息可帮助您对使用此软件时可能遇到的任何问题进行错误诊断。

Sun Grid Engine 5.3 软件如何检索错误报告

Sun Grid Engine 软件通过将消息记录到某些文件和 / 或通过发送电子邮件 (e-mail) 来报告错误或警告。使用的日志文件包括：

- 消息文件：

sgc_qmaster、sgc_schedd 和 sgc_execd 各有其单独的消息文件。这些文件有相同的文件名 messages。sgc_qmaster 日志文件位于主控假脱机目录；sgc_schedd 消息文件位于调度程序的假脱机目录；而执行守护程序的日志文件位于执行守护程序的假脱机目录（请参见第 21 页的“根目录下的假脱机目录”一节，以获得有关假脱机目录的更多信息）。

这些消息文件有以下格式：

- 每条消息占一行。
- 消息细分为 5 个部分，以竖线符号 (|) 分隔。
- 第一部分为消息的时间戳。
- 第二部分指出生成此消息的 Sun Grid Engine 守护程序。
- 第三部分为运行该守护程序的主机名。
- 第四部分为消息类型，其中，N 表示注意、I 表示信息（前两者只是作为一般信息而已）、W 表示警告（某些可能出错的情况）、E 表示错误（已检测到错误情况）或 C 表示紧急（可导致程序中止）。

- 第五部分为消息文本。

注意 – 若由于某种原因无法访问错误日志文件，则 Sun Grid Engine 将试图将错误消息记录到相应主机上的以下文件中：`/tmp/sge_qmaster_messages`、`/tmp/sge_schedd_messages` 或 `/tmp/sge_execd_messages`。

- 作业 STDERR 输出：

一旦启动作业，作业脚本的标准错误 (STDERR) 输出就被重定向到一个文件。文件名和位置或遵循缺省值，或由某些 `qsub` 命令行开关选项指定。请参考 *Sun Grid Engine 5.3 管理和用户指南* 和 *《Sun Grid Engine 5.3 和 Sun Grid Engine 5.3 (企业版) 参考手册》*，以获得细节信息。

在某些情况下，Sun Grid Engine 通过电子邮件通知用户和 / 或管理员有关错误。Sun Grid Engine 发送的邮件消息不包含消息正文。消息文本完全包含在邮件主题字段中。

不同错误或退出代码的后果

表 11-1 列出了与作业有关的不同错误或退出代码的后果。这些代码对各种类型的 Sun Grid Engine 作业均有效。

表 11-1 与作业有关的错误或退出代码

| 脚本 / 方法 | 退出或错误代码 | 后果 |
|-------------|---------|---------------|
| 作业脚本 | 0 | 成功 |
| | 99 | 重新排队 |
| | 其它 | 成功：帐户文件中的退出代码 |
| 前导脚本 / 收尾脚本 | 0 | 成功 |
| | 99 | 重新排队 |
| | 其它 | 队列错误状态；作业重新排队 |

表 11-2 列出了与并行环境 (PE) 配置有关的作业错误或退出代码的后果。

表 11-2 与 PE 有关的错误或退出代码

| 脚本 / 方法 | 退出或错误代码 | 后果 |
|----------|---------|-------------------|
| pe_start | 0 | 成功 |
| | 其它 | 队列设置为错误状态，作业重新排队 |
| pe_stop | 0 | 成功 |
| | 其它 | 队列设置为错误状态，作业不重新排队 |

表 11-3 列出了与队列配置有关的作业错误或退出代码的后果。这些仅当相应的方法被覆盖时才有效。

表 11-3 与队列有关的错误或退出代码

| 脚本 / 方法 | 退出或错误代码 | 后果 |
|---------|---------|------------|
| 作业启动程序 | 0 | 成功 |
| | 其它 | 成功，无其它特殊含义 |
| 暂停 | 0 | 成功 |
| | 其它 | 成功，无其它特殊含义 |
| 恢复 | 0 | 成功 |
| | 其它 | 成功，无其它特殊含义 |
| 终止 | 0 | 成功 |
| | 其它 | 成功，无其它特殊含义 |

表 11-4 列出了与点检查有关的作业错误或退出代码的后果。

表 11-4 与点检查有关的错误或退出代码

| 脚本 / 方法 | 退出或错误代码 | 后果 |
|---------|---------|---|
| 点检查 | 0 | 成功 |
| | 其它 | 成功 — 但对于核心点检查有特殊含义：点检查未成功；未执行点检查。 |
| 迁移 | 0 | 成功 |
| | 其它 | 成功 — 但对于核心点检查有特殊含义：点检查未成功；未执行点检查。将进行迁移。 |
| 重新启动 | 0 | 成功 |
| | 其它 | 成功，无其它特殊含义 |
| 清除 | 0 | 成功 |
| | 其它 | 成功，无其它特殊含义 |

在调试模式下运行 Sun Grid Engine 程序

对于某些严重的错误情况，错误记录机制可能无法产生足够信息以识别问题。因此，Sun Grid Engine 允许在 *调试* 模式下运行几乎所有辅助程序和守护程序。有多种调试级别，各调试级别随所提供信息的广度和深度不同而各异。调试级别范围从 0 到 10，其中 10 为发送最详细信息的级别，而 0 则关闭调试。

为了设置调试级别，Sun Grid Engine 发行软件中还提供了 `.cshrc` 或 `.profile` 资源文件的扩展文件。对于 `csh` 或 `tcsch` 用户，提供了文件 `<sgc 根目录>/<util>/dl.csh`。对于 `sh` 或 `ksh` 用户，相应的文件名为 `<sgc 根目录>/util/dl.sh`。该文件需要“提供来源”至标准资源文件。作为 `csh` 或 `tcsch` 用户，请将以下行：

```
source <sgc 根目录>/util/dl.csh
```


添加至 `cshrc` 文件。作为 `sh` 或 `ksh` 用户，将以下行：

```
. <sgc 根目录>/util/dl.sh
```

添加至 `.profile` 文件，其效果相同。若此刻注销并再次登录，您就可以使用以下命令设置调试级别：

```
% dl level
```

若级别大于 0，此后启动 `Sun Grid Engine` 命令时，将强制该命令将跟踪记录输出写入 `STDOUT`。根据强制执行的调试级别的不同，跟踪记录输出可能包含警告、状态和错误消息，以及内部调用的程序模块名，连同源代码行号信息（这有助于报告错误）。

注意 – 拥有大量滚动行缓冲（例如 1000 行）的窗口可能有助于查看调试跟踪记录。

注意 – 若窗口为 `xterm`，您可能想要以后使用 `xterm` 记录机制检查跟踪记录输出。

在调试模式下运行一个 `Sun Grid Engine` 守护程序的结果是，该守护程序保持其终端连接以写入追踪记录输出。可通过键入您所用终端仿真的中止符来中止该操作（例如，`Control-C`）。

注意 – 要关闭调试模式，将调试级别设回 0。
