



Sun™ ONE Grid Engine 5.3 관리 및 사용 설명서

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

부품 번호 : 816-7464-10
2002년 9월, 개정판 A

이 문서에 대한 의견은 다음 주소로 보내십시오: docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

복사, 배포, 변경을 제한하는 승인하에 배포됩니다. 본 제품이나 문서는 Sun과 승인자의 사전 서면 허가 없이 어떤 형태나 방법으로도 재생산될 수 없습니다. 글꼴 기술을 포함한 타사의 소프트웨어도 저작권에 의해 보호되며 Sun사의 공급업체에 의해 승인되었습니다.

이 제품의 일부는 캘리포니아 대학에서 승인된 Berkeley BSD 시스템을 토대로 합니다. UNIX는 미국 및 기타 국가에서 X/Open Company, Ltd. 사에 독점권이 부여된 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, AnswerBook2, docs.sun.com 및 Solaris는 미국 및 기타 국가에 있는 Sun Microsystems, Inc. 의 상표, 등록 상표 또는 서비스 마크입니다. 모든 SPARC 상표는 미국 및 기타 국가에서 SPARC International, Inc. 의 승인하에 사용되는 SPARC International, Inc. 의 상표 또는 등록 상표입니다. SPARC 상표가 있는 제품은 Sun Microsystems, Inc. 가 개발한 구조에 기초합니다. OPEN LOOK과 Sun™ Graphical User Interface는 Sun Microsystems, Inc. 가 사용자와 승인자를 위해 개발한 것입니다. Sun은 Xerox사의 컴퓨터 산업을 위한 비주얼 또는 그래픽 사용자 인터페이스의 개념 연구와 개발에 대한 선구적 업적을 높이 평가합니다. Sun은 Xerox사로부터 Xerox Graphical User Interface에 대한 비독점권을 부여 받았으며 이 권한은 OPEN LOOK GUI를 구현하는 Sun의 승인자에게도 해당되며 Sun의 서면 허가 계약에 기초합니다.

출판물은 “사실”만을 제공하며 본 제품의 시장성, 합목적성, 특허권 비침해에 대한 묵시적 보증을 비롯하여 모든 명시적, 묵시적 조건 제시, 책임이나 보증을 하지 않습니다. 단, 이러한 권리 포기가 법적으로 무효가 되는 경우는 예외로 합니다.



재활용
가능



Adobe PostScript

목차

머리말 xv

설명서의 구성 xv

UNIX 명령어 사용 xvi

문서 규약 xvi

셸 프롬프트 xvii

관련 문서 xvii

온라인 Sun 문서에 액세스 xvii

Sun은 당신의 의견을 환영합니다 xviii

I 부. 배경 및 정의

1. Sun Grid Engine 5.3 소개 1

그리드 컴퓨팅의 정의 1

자원 및 정책 관리를 통한 작업부하 관리 2

시스템이 동작하는 방법 2

요구에 자원을 대응 2

작업과 대기열: Sun Grid Engine 세계 3

Sun Grid Engine 5.3 구성요소 4

호스트 4

마스터 호스트 4

실행 호스트	4
관리 호스트	4
제출 호스트	5
데몬	5
sge_qmaster - 마스터 데몬	5
sge_schedd - 스케줄러 데몬	5
sge_execd - 실행 데몬	5
sge_commd - 통신 데몬	5
대기열	6
클라이언트 명령	6
QMON, Sun Grid Engine 그래픽 사용자 인터페이스	8
QMON 사용자 정의	9
Sun Grid Engine 용어의 용어집	10

II 부. 시작하기

2. 설치 15

빠른시작 설치 개요	15
빠른시작 설치 필요조건	16
설치 계정	16
설치 디렉토리	16
▼ 설치 디렉토리 생성 방법	17
통신 포트 번호	17
배포 매체 읽기	17
▼ Sun Grid Engine 5.3 배포 포장 제거	18
사용자 클러스터용 기본 Sun Grid Engine 시스템 설치	18
▼ 마스터 호스트 설치 방법	18
▼ 실행 호스트 설치 방법	19
기본 시스템 구성	20

전체 설치 개요	22
1 단계 - 계획	22
2 단계 - 소프트웨어 설치	23
3 단계 - 설치 검증	23
설치 계획	24
필수조건 작업	24
설치 디렉토리 <sgc 루트>	24
루트 디렉토리의 스폴 디렉토리	25
디렉토리 구성	25
디스크 공간 요구 사항	26
설치 계정	27
파일 액세스 사용 권한	27
네트워크 서비스	27
마스터 호스트	28
새도우 마스터 호스트	28
실행 호스트	28
관리 호스트	28
제출 호스트	29
셸	29
사용자 이름	29
대기열	29
▼ 설치 계획 방법	30
▼ 배포 매체를 읽는 방법	30
▼ 마스터 호스트 설치 방법	31
▼ 실행 호스트 설치 방법	32
▼ 관리 및 제출 호스트 설치 방법	33
향상된 보안으로 설치	33
필요한 추가 설정	34

- ▼ CSP 기반 시스템 설치 및 설정 방법 34
- ▼ 사용자를 위한 인증 및 개인용 키 생성 방법 43
- ▼ 인증 점검 방법 45
 - 인증 표시 45
 - 발행자 점검 45
 - 주제 점검 45
 - 인증의 이메일 표시 45
 - 유효성 표시 46
 - 지문 표시 46
- ▼ 설치 검증 방법 46

III 부. Sun Grid Engine 5.3 소프트웨어 사용

3. Sun Grid Engine 5.3 프로그램 사용 53

- Sun Grid Engine 사용자 유형 및 조작 54
- 대기열 및 대기열 등록정보 55
 - QMON 브라우저 55
- ▼ QMON 브라우저를 시작하는 방법 55
 - Queue Control QMON 대화 상자 56
- ▼ 대기열 목록을 표시하는 방법 56
- ▼ 대기열 등록정보를 표시하는 방법 56
 - QMON 브라우저 사용 56
 - 명령줄에서 58
 - 대기열 등록 정보 해석 58
- 호스트 기능 59
 - ▼ 마스터 호스트의 이름을 찾는 방법 59
 - ▼ 실행 호스트의 목록을 표시하는 방법 59
 - ▼ 관리 호스트의 목록을 표시하는 방법 60
 - ▼ 제출 호스트의 목록을 표시하는 방법 60

요청 가능한 속성	60
▼ 요청 가능 속성 목록을 표시하는 방법	61
사용자 액세스 사용 권한	64
관리자, 운영자 및 소유자	65
4. 작업 제출	67
간단한 작업 실행	67
▼ 명령줄에서 간단한 작업을 실행하는 방법	68
▼ 그래픽 사용자 인터페이스 QMON에서 작업을 제출하는 방법	70
일괄처리 작업 제출	73
셸 스크립트 정보	73
스크립트 파일의 예	74
QMON을 사용한 확장 및 고급 작업 제출	74
확장 예	74
고급 보기	79
자원 요구사항 정의	83
Sun Grid Engine 시스템이 자원을 할당하는 방법	85
정규 셸 스크립트에 대한 확장	86
명령 해석기가 선택되는 방법	86
출력 경로 재지정	86
활성 Sun Grid Engine 주석	87
▼ 명령줄에서 작업을 제출하는 방법	90
기본 요청	91
어레이 작업	92
▼ 명령줄에서 어레이 작업을 제출하는 방법	93
▼ QMON을 사용하여 어레이 작업을 제출하는 방법	93
대화식 작업 제출	94
QMON을 사용한 대화식 작업 제출	95

- ▼ QMON을 사용하여 대화식 작업을 제출하는 방법 95
 - qsh를 사용한 대화식 작업 제출 98
- ▼ qsh를 사용하여 대화식 작업을 제출하는 방법 98
 - qlogin을 사용한 대화식 작업 제출 98
- ▼ qlogin을 사용하여 대화식 작업을 제출하는 방법 98
 - 병렬 작업 99
 - Sun Grid Engine 작업이 스케줄되는 방법 99
 - 작업 우선순위 99
 - 동일공유스케줄링 100
 - 대기열 선택 100
- 투명한 원격 실행 100
 - qrsh를 사용한 원격 실행 101
 - qrsh 사용법 101
 - qtcsh를 사용한 투명한 작업 분배 102
 - qtcsh 사용법 103
 - qmake를 사용한 병렬 Makefile 처리 104
 - qmake 사용법 105

5. 작업 검사점, 모니터링 및 제어 107

- 작업 검사점 정보 107
 - 사용자 레벨 검사점 108
 - 커널 레벨 검사점 108
 - 검사점 작업의 이주 108
 - 검사점 작업 스크립트 구성 109
- ▼ 명령줄에서 검사점 작업을 제출, 모니터 또는 삭제하는 방법 110
- ▼ QMON을 사용하여 검사점 작업을 제출하는 방법 111
 - 파일 시스템 요구사항 111
- Sun Grid Engine 작업 모니터링 및 제어 112

- ▼ QMON을 사용하여 작업을 모니터 및 제어하는 방법 112
 - QMON 오브젝트 브라우저를 사용한 추가 정보 121
- ▼ qstat 를 사용하여 작업을 모니터하는 방법 122
- ▼ 전자 우편에 의해 작업을 모니터하는 방법 124
 - 명령줄에서 Sun Grid Engine 작업 제어 124
- ▼ 명령줄에서 작업을 제어하는 방법 125
- 작업 종속성 126
- 대기열 제어 126
 - ▼ QMON을 사용하여 대기열을 제어하는 방법 126
 - ▼ qmod를 사용하여 대기열을 제어하는 방법 130
- QMON 사용자 정의 131

IV 부. 관리

6. 호스트 및 클러스터 구성 135

- 마스터 및 새도우 마스터 구성 정보 136
- 데몬 및 호스트 정보 137
 - 호스트 구성 정보 138
- ▼ QMON을 사용한 관리 호스트 구성 방법 138
- ▼ 관리 호스트 삭제 방법 140
- ▼ 관리 호스트 추가 방법 140
- ▼ 명령줄에서 관리 호스트를 구성하는 방법 140
- ▼ QMON을 사용한 제출 호스트 구성 140
- ▼ 제출 호스트 삭제 방법 142
- ▼ 제출 호스트 추가 방법 142
- ▼ 명령줄에서 제출 호스트를 구성하는 방법 142
- ▼ QMON을 사용하여 실행 호스트를 구성하는 방법 142
- ▼ 실행 호스트를 삭제하는 방법 144
- ▼ 실행 호스트 데몬을 종료하는 방법 144

- ▼ 실행 호스트를 추가 또는 수정하는 방법 144
- ▼ 명령줄에서 실행 호스트를 구성하는 방법 148
- ▼ qhost로 실행 호스트를 모니터하는 방법 149
- ▼ 명령줄에서 데몬을 종료하는 방법 150
- ▼ 명령줄에서 데몬을 재시작하는 방법 151
- 기본 클러스터 구성 151
 - ▼ 명령줄에서 기본 클러스터 구성을 표시하는 방법 152
 - ▼ 명령줄에서 기본 클러스터 구성을 수정하는 방법 152
 - ▼ QMON으로 클러스터 구성을 표시하는 방법 152
 - ▼ QMON으로 클러스터 구성을 삭제하는 방법 153
 - ▼ QMON으로 글로벌 클러스터 구성을 표시하는 방법 153
 - ▼ 글로벌 및 호스트 구성을 수정하기 위해 QMON을 사용하는 방법 154

7. 대기열 구성 및 대기열 캘린더 157

- 대기열 구성 정보 157
 - ▼ QMON을 사용하여 대기열을 구성하는 방법 158
 - ▼ General 매개변수 구성 방법 159
 - ▼ Execution Method 매개변수 구성 방법 160
 - ▼ Checkpointing 매개변수 구성 방법 161
 - ▼ Load 및 Suspend Thresholds 구성 방법 162
 - ▼ Limits 구성 방법 163
 - ▼ User Complexes 구성 방법 165
 - ▼ 종속 대기열 구성 방법 166
 - ▼ User Access 구성 방법 167
 - ▼ Owners 구성 방법 168
 - ▼ 명령줄에서 대기열을 구성하는 방법 169
- 대기열 캘린더 정보 170
 - ▼ QMON을 사용하여 대기열 캘린더를 구성하는 방법 171

- ▼ 명령줄에서 캘린더를 구성하는 방법 173
- 8. 콤플렉스 개념 175**
 - 콤플렉스 관련 정보 175
 - ▼ 콤플렉스 구성을 추가 또는 수정하는 방법 176
 - 콤플렉스 유형 177
 - 대기열 콤플렉스 178
 - 호스트 콤플렉스 178
 - 글로벌 콤플렉스 180
 - 사용자 정의 콤플렉스 181
 - 소비 가능 자원 185
 - ▼ 소비 가능 자원을 설정하는 방법 185
 - 소비 가능 자원 설정의 예 187
 - 콤플렉스 구성 196
 - ▼ 명령줄에서 콤플렉스 구성을 수정하는 방법 196
 - qconf 명령의 예 197
 - 부하 매개변수 197
 - 기본 부하 매개변수 197
 - 사이트 고유 부하 매개변수 추가 198
 - ▼ 사용자 자신의 부하 센서를 작성하는 방법 198
 - 규칙 199
 - 스크립트의 예 199
- 9. 사용자 액세스 및 정책 관리 203**
 - 사용자 설정 203
 - 사용자 액세스 정보 205
 - ▼ QMON을 사용하여 계정을 구성하는 방법 205
 - ▼ QMON을 사용하여 관리자 계정을 구성하는 방법 206
 - ▼ 명령줄에서 관리자 계정을 구성하는 방법 207

사용 가능한 스위치	207
▼ QMON을 사용하여 운영자 계정을 구성하는 방법	207
▼ 명령줄에서 운영자 계정을 구성하는 방법	208
사용 가능한 스위치	208
대기열 소유자 계정 정보	209
사용자 액세스 권한	209
▼ QMON을 사용하여 사용자 액세스 목록을 구성하는 방법	210
▼ 명령줄에서 사용자 액세스 목록을 구성하는 방법	212
사용 가능한 옵션	212
스케줄링 정보	212
스케줄링 전략	213
대기열 정렬	213
작업 정렬	213
스케줄러 간격에서 발생하는 것	214
스케줄러 모니터링	214
기본 스케줄링	215
스케줄링 대안	215
▼ QMON을 사용하여 스케줄러 구성을 변경하는 방법	218
경로 별명 지정	221
파일 형식	222
경로 별명 지정 파일이 해석되는 방법	222
경로 별명 지정 파일의 예	223
가본 요청 구성	223
기본 요청 파일의 형식	224
기본 요청 파일의 예	224
회계 및 이용율 통계 수집	225
검사점 지원 정보	225
검사점 환경	226

- ▼ QMON을 사용하여 검사점 환경을 구성하는 방법 227
 - 구성된 검사점 환경 보기 227
 - 구성된 검사점 환경 삭제 227
 - 구성된 검사점 환경 수정 228
 - 검사점 환경 추가 230
- ▼ 명령줄에서 검사점 환경을 구성하는 방법 230
 - qconf 검사점 옵션 230

10. 병렬 환경 관리 233

- 병렬 환경 정보 233
- ▼ QMON을 사용하여 PE를 구성하는 방법 234
 - ▼ PE의 내용 표시 234
 - ▼ PE 삭제 234
 - ▼ PE 수정 235
 - ▼ PE 추가 235
- ▼ 명령줄에서 PE를 구성하는 방법 238
 - qconf PE 옵션 238
- ▼ 명령줄에서 구성된 PE 인터페이스를 표시하는 방법 239
- ▼ QMON으로 구성된 PE 인터페이스를 표시하는 방법 239
 - PE 시작 절차 241
 - PE의 종료 242
 - PE 및 Sun Grid Engine 소프트웨어의 밀접한 통합 243

11. 오류 메시지 245

- Sun Grid Engine 5.3 소프트웨어가 오류 보고서를 검색하는 방법 245
 - 여러 오류 또는 종료 코드의 결과 246
 - 디버그 모드에서 Sun Grid Engine 프로그램 실행 248

머리말

*Sun Grid Engine 5.3 관리 및 사용 설명서*는 제품에 관한 기본 정보, 설치 지침 및 제품 사용 방법에 대한 지침을 제공하는 종합 설명서입니다.

설명서의 구성

이 설명서가 Sun Grid Engine 5.3 제품 사용자 및 제품 책임이 항상 사용자와 같지는 않은 시스템 관리자 모두를 위한 것이기 때문에 이 설명서는 4 부분으로 나뉘어집니다. 각 부분에는 사용자 또는 관리자가 특별히 관심을 갖는 정보가 들어 있습니다.

각 부분에 대한 설명 및 계획된 독자들은 다음과 같습니다.

■ 제 1부 - 배경 및 정의

Sun Grid Engine 5.3 관리 및 사용 설명서 중 이 부분은 사용자 및 관리자를 위한 부분으로서 제품의 사용, 구성요소, 전문용어 등의 상세한 설명을 제공합니다.

■ 제 2부 - 시작하기

이 부분은 제품 설치자(일반적으로 관리자)를 위한 것으로 “빠른시작”, 전체 및 업그레이드 설치에 대한 상세한 설명이 들어 있습니다.

■ 제 3부 - Sun Grid Engine 5.3 소프트웨어 사용

이 부분은 사용자 및 관리자 모두를 위한 것입니다. 많은 작업과 관련되는 지침 및 기본 정보가 들어 있습니다.

■ 제 4부 - 관리

이 부분에 있는 기본 정보 및 지침은 숙련된 시스템 관리자를 위한 것입니다.

UNIX 명령어 사용

이 문서에는 시스템 종료, 시스템 시동 및 장치 구성과 같은 절차 및 기본 UNIX® 명령에 대한 정보가 없을 수도 있습니다.

이 정보에 대해서는 다음 중 하나 이상을 참조하십시오.

- *Solaris Handbook for Sun Peripherals*
- Solaris™ 운영 환경용 AnswerBook2™ 온라인 문서
- 시스템과 함께 받은 다른 소프트웨어 설명서

문서 규약

서체	의미	보기
<i>AaBbCc123</i>	명령어, 파일, 디렉토리의 이름; 화면 출력	<code>.login</code> 파일을 편집하십시오. 모든 파일을 나열하려면 <code>ls -a</code> 를 사용하십시오. % You have mail.
AaBbCc123	화면 출력에 대해 사용자가 입력하는 내용	% su Password:
<i>AaBbCc123</i>	책 제목, 새 단어나 용어, 강조하는 단어명령줄 변수를 실제 이름이나 값으로 대체하십시오.	<i>사용 설명서의 6 장을 읽어 보십시오.</i> 이러한 옵션을 <i>class</i> 옵션이라고 합니다. 이 작업을 하려면 슈퍼유저 <i>여야</i> 합니다. 파일을 삭제하려면 <code>rm 파일이름</code> 을 입력하십시오.

셸 프롬프트

셸	프롬프트
C 셸	시스템이름%
C 셸 슈퍼유저	시스템이름#
Bourne 셸 및 Korn 셸	\$
Bourne 셸 및 Korn 셸 슈퍼유저	#

관련 문서

응용 프로그램	제목	부품 번호
참조	<i>Sun Grid Engine 5. 3 및 Sun Grid Engine, Enterprise 버전 5. 3 참조 설명서</i>	816-7477-10

온라인 Sun 문서에 액세스

다음 주소에서 광범한 Sun 시스템 문서를 찾을 수 있습니다.

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

다음 주소에서 Solaris 문서 전체 세트 및 다른 많은 문서들을 구할 수 있습니다.

<http://docs.sun.com>

또한 이 사이트에서 이 안내서의 인쇄된 사본을 주문하는 방법에 대한 정보도 찾을 수 있습니다.

Sun은 당신의 의견을 환영합니다

Sun은 문서 개선을 위해 노력하고 있으며 여러분의 의견과 제안을 환영합니다. 다음 주소로 여러분의 의견을 보낼 수 있습니다.

docfeedback@sun.com

전자우편의 제목란에 문서의 부품 번호(816-7464-10)를 기입해 주십시오.

I 부 배경 및 정의

Sun Grid Engine 5.3 관리 및 사용 설명서 중 이 부분은 하나의 장으로 구성되어 있습니다.

- **제 1장** – 1 페이지, “Sun Grid Engine 5.3 소개”

장의 내용에 익숙해지면 사용자 및 관리자 모두에게 도움이 되지만, 이 장의 간결함이 독자에게 사용자 및 관리자에 대한 중요성을 간과하도록 만들 수도 있습니다. 다음 주제가 이 장에서 다루어집니다.

- 복잡한 컴퓨팅 환경에서 Sun™ Grid Engine 5.3 소프트웨어의 기본 역할의 설명
- 제품의 주요 구성요소 및 각 기능의 정의에 대한 목록
- Sun Grid Engine 5.3 환경에서 알아야 하는 중요한 용어의 용어집

Sun Grid Engine 5.3 소개

이 장은 사용자 및 관리자에게 유용한 Sun Grid Engine 5.3 시스템에 관한 기본 정보를 제공합니다. 클러스터된 컴퓨터의 혼란스런 상태를 관리하는 데 있어서 제품의 역할 설명 외에도 이 장에는 다음 주제가 포함됩니다.

- 그리드 컴퓨팅에 대한 간략한 설명
- Sun Grid Engine 5.3 그래픽 사용자 인터페이스인 QMON의 개요
- 제품의 중요한 각 구성요소의 설명
- 사용자 및 관리자가 사용할 수 있는 클라이언트 명령의 자세한 목록
- Sun Grid Engine 5.3 전문용어의 전체 용어집

그리드 컴퓨팅의 정의

개념적으로 그리드는 아주 단순합니다. 그리드는 작업을 수행하는 컴퓨팅 자원의 콜렉션입니다. 강력한 분배 자원에 대한 단일 접근점을 제공하며 사용자에게는 거대한 시스템으로 여겨집니다. 사용자들은 그리드를 단일 컴퓨터 사용 자원으로 다룹니다. Sun Grid Engine같은 자원 관리 소프트웨어는 사용자가 제출하는 작업을 허용하고 자원 관리 정책을 기반으로 그리드의 적당한 시스템에서 실행하도록 예약합니다. 사용자는 작업이 실행되는 곳에 신경쓰지 않고 글자 그대로 한 번에 수많은 작업을 제출할 수 있습니다.

두 그리드는 비슷하지 않습니다. 즉, 한 크기가 모든 상황에 맞지 않습니다. 세 개의 핵심 그리드 클래스가 있는데, 이것은 단일 시스템부터 수 천 개의 프로세서를 이용하는 슈퍼컴퓨터급의 compute farm까지의 범위를 갖습니다. 단일 프로젝트나 부서에서 사용자에게 단일 접근점을 제공하기 위해 함께 작동하는 많은 컴퓨터 사용 자원을 구성하는 클러스터 그리드는 Sun Grid Engine 5.3 시스템을 사용하여 생성 및 관리할 수 있는 것입니다.

(더 복잡한 그리드의 다른 두 가지 유형—*캠퍼스 그리드* 및 *글로벌 그리드*—은 Sun의 관련 제품인 Sun Grid Engine, Enterprise 버전에 의해 생성 및 관리됩니다.)

Sun Grid Engine 5.3 소프트웨어는 조직의 기술진 및 경영진에 의해 설정되는 엔터프라이즈 자원 정책을 기반으로 연산 능력의 전달을 지휘합니다. Sun Grid Engine 시스템은 이들 정책을 사용하여 그리드 내에서 사용할 수 있는 연산 자원을 시험하고 이들 자원을 수집한 후 그리드 전체에서 사용을 최적화하는 방법으로 자동으로 자원을 할당하고 전달합니다.

자원 및 정책 관리를 통한 작업부하 관리

Sun Grid Engine 소프트웨어는 사용자에게 연관된 작업부하의 투명한 분배를 위해 연산 작업을 많이 요구하는 작업을 Sun Grid Engine 시스템에 제출하는 방법을 제공합니다. 사용자는 Sun Grid Engine 시스템에 일괄처리 작업, 대화식 작업 및 병렬 작업을 제출할 수 있습니다.

이 제품은 또한 점검 프로그램을 지원합니다. 점검 작업은 부하 수요에 대한 사용자 간섭 없이 워크스테이션 사이에서 이주합니다.

소프트웨어는 관리자를 위한 Sun Grid Engine 작업 모니터링 및 제어를 위한 포괄적인 도구를 제공합니다.

시스템이 동작하는 방법

Sun Grid Engine 시스템은 외부 세계로부터 작업—컴퓨터 자원에 대한 사용자의 요구—을 허용하고, 작업이 실행될 수 있을 때까지 보류 영역에 놓고, 보류 영역에서 실행 장치로 작업을 송신하고, 실행 중에 작업을 관리하고 작업이 완료될 때 실행 레코드를 기록합니다.

비슷한 것으로 세계의 자본 도시 중 하나에 있는 큰 “money-center” 은행을 상상하십시오.

요구에 자원을 대응

은행의 로비에는 수많은 고객들이 서로 다른 요구사항을 갖고 서비스를 기다립니다. 어떤 고객은 계정에서 약간의 현금만 인출하려고 합니다. 바로 뒤에 도착한 또 다른 고객은 은행의 투자 전문가 중 하나와 약속이 되어 있으며 복잡한 벤처를 시작하기 전에 자문을 구하고 있습니다. 긴 줄에서 있는 그들 앞에는 큰 규모의 대출을 신청하려는 또 다른 고객이 있습니다. 또 그녀 앞에는 똑같이 대출을 받으려는 8명의 고객이 있습니다.

여러 고객들의 다양한 의도에는 다른 은행 자원 유형 및 레벨이 필요합니다. 아마도 이 특별한 날, 은행에는 단순히 자신의 계정에서 현금을 인출하는 한 고객을 다룰 수 있는 충분한 시간이 있는 직원들이 많이 있을 것입니다. 그러나 같은 날 한 두 명의 대출 직원 만 많은 대출 신청자들을 돕고 있습니다. 다른 날에는 이 상황이 반대일 수 있습니다.

물론 그 결과는 고객들이 서비스를 기다려야 한다는 것입니다. 고객의 요구사항이 즉시 식별되고 사용 가능한 자원에 일치되는 경우에만 아마 많은 고객들이 빠른 서비스를 받을 수도 있겠지만 말입니다.

Sun Grid Engine 시스템이 은행 관리자였다면 서비스를 다르게 구성했을 것입니다.

- 은행 로비에 들어설 때 고객은 자신의 이름, 소속(예: 회사 표시) 및 요구사항을 알려 줘야 합니다.
- 고객의 도착 시간이 기록됩니다.
- 고객이 로비에서 제공한 정보를 기반으로 그 요구사항이 적합하고 즉시 사용 가능한 자원과 일치하는 고객, 요구사항이 최고의 우선순위를 갖는 고객 및 가장 오랜 동안 로비에서 대기한 고객에게 서비스를 제공합니다.
- 물론 “Sun Grid Engine 은행”에서 한 은행 직원이 동시에 여러 고객을 도울 수 있을지도 모릅니다. Sun Grid Engine 시스템은 새 고객을 가장 부하가 적고 가장 적합한 은행 직원에게 할당하려 시도합니다.

작업과 대기열: Sun Grid Engine 세계

Sun Grid Engine 시스템에서 *작업*은 은행 고객에 해당하며 작업은 로비 대신 컴퓨터 보류 영역에서 대기하고, 컴퓨터 서버에 위치하는 *대기열*이 은행 직원을 대신하여 작업에 서비스를 제공합니다. 비슷하게 은행 고객의 경우와 같이, 각 작업의 요구사항(대개 사용 가능한 메모리, 실행 속도, 사용 가능한 소프트웨어 사용권 및 비슷한 필요)은 매우 다를 수 있으며 특정 대기열만이 대응하는 서비스를 제공할 수 있습니다.

유사성에 따라서 Sun Grid Engine 소프트웨어는 사용할 수 있는 자원과 작업 요구사항을 다음 형식으로 증재합니다.

- Sun Grid Engine 시스템을 통해 작업을 제출하는 사용자가 작업에 대한 요구사항 프로파일을 알려 줍니다. 또한 사용자의 신원과 *프로젝트*또 *사용자 그룹*에 대한 사용자 소속이 시스템에 의해 검색됩니다. 사용자가 작업을 제출한 시간도 저장됩니다.
- 정확히 대기열이 새 작업의 실행에 사용 가능할 수 있게 된 순간에 Sun Grid Engine 시스템이 대기열에 대한 적합한 작업을 판별하고 즉시 가장 높은 우선순위 또는 가장 긴 대기 시간을 갖는 작업을 작업 지정합니다.
- Sun Grid Engine 대기열은 많은 작업의 동시 실행을 허용할 수 있습니다. Sun Grid Engine 시스템은 가장 부하가 적고 적합한 대기열에서 새 작업을 시작하려고 합니다.

Sun Grid Engine 5.3 구성요소

그림 그림 1-1은 가장 중요한 Sun Grid Engine 구성요소 및 시스템에서 구성요소의 상호 작용을 표시합니다. 다음 절은 구성요소의 기능을 설명합니다.

호스트

4 가지 유형의 호스트가 Sun Grid Engine 5.3 시스템에 기본적입니다.

- 마스터
- 실행
- 관리
- 제출

마스터 호스트

마스터 호스트는 전체 클러스터 활동의 중심입니다. 마스터 데몬인 `sge_qmaster` 및 스케줄러 데몬 `sge_schedd`을 실행합니다. 두 데몬 모두가 대기열과 작업 같은 모든 Sun Grid Engine 구성요소를 제어하고 구성요소의 상태, 사용자 액세스 권한 등에 관한 표를 유지보수합니다.

기본적으로 마스터 호스트는 또한 관리 호스트이고 제출 호스트입니다. 이들 호스트에 대한 절을 참조하십시오.

실행 호스트

실행 호스트는 Sun Grid Engine 작업을 실행할 권한을 갖는 노드입니다. 그러므로 실행 호스트는 Sun Grid Engine 대기열을 호스트하고 Sun Grid Engine 실행 데몬인 `sge_execd`를 실행합니다.

관리 호스트

Sun Grid Engine 시스템에 대한 모든 종류의 관리 활동을 수행할 사용 권한을 호스트에 부여할 수 있습니다.

제출 호스트

제출 호스트는 *일괄처리 작업만* 제출 및 제어를 허용합니다. 특히 제출 호스트에 로그인 하는 사용자는 `qsub`를 통해 작업을 제출하고, `qstat`를 통해 작업 상태를 제어하고, Sun Grid Engine OSF/1 Motif 그래픽 사용자 인터페이스인 `QMON`을 사용할 수 있습니다. 이 인터페이스는 8 페이지, “`QMON, Sun Grid Engine` 그래픽 사용자 인터페이스” 절에 설명되어 있습니다.

참고 - 한 호스트가 위에 설명한 클래스의 둘 이상에 속할 수 있습니다.

데몬

4 개의 데몬이 Sun Grid Engine 5.3 시스템의 기능성을 제공합니다.

`sge_qmaster` - 마스터 데몬

클러스터의 관리 및 스케줄링 활동의 중심인 `sge_qmaster`가 호스트, 대기열, 작업, 시스템 부하 및 사용자 권한에 관한 표를 유지보수합니다. `sge_schedd`로부터 스케줄링 결정을 수신하고 적절한 실행 호스트의 `sge_execd`로부터 조치를 요청합니다.

`sge_schedd` - 스케줄러 데몬

스케줄링 데몬은 `sge_qmaster`의 도움을 받아서 클러스터 상태의 최신 보기를 유지보수합니다. 다음 스케줄링 결정을 수행합니다.

- 어떤 작업이 어떤 대기열에 작업 지정되는지 여부

그런 다음 이들 결정을 필요한 조치를 시작하는 `sge_qmaster`에 전달합니다.

`sge_execd` - 실행 데몬

실행 데몬은 호스트의 대기열과 이들 대기열에 있는 작업의 실행을 책임집니다. 주기적으로 호스트의 작업 상태 또는 부하 같은 정보를 `sge_qmaster`로 전송합니다.

`sge_commd` - 통신 데몬

통신 데몬은 잘 알려진 TCP 포트상에서 통신합니다. Sun Grid Engine 구성요소 사이의 모든 통신에 사용됩니다.

대기열

Sun Grid Engine 대기열은 특정 호스트에서 동시에 실행하도록 허용되는 작업 클래스에 대한 컨테이너입니다. 대기열은 특정 작업 속성, 예를 들어 이주될 수 있는지 여부를 판별합니다. 수명 동안 작업 실행은 대기열과 연관됩니다. 대기열과의 연관은 작업에 발생할 수 있는 사항의 일부에 영향을 줍니다. 예를 들어 대기열이 일시중단되면 해당 대기열과 연관된 모든 작업도 일시중단됩니다.

Sun Grid Engine 시스템에서는 작업을 대기열에 직접 제출할 필요가 없습니다. 사용자는 작업의 요구사항 프로파일(예: 메모리, 운영 체제, 사용 가능한 소프트웨어 등)을 지정하면 되고 Sun Grid Engine 소프트웨어가 자동으로 부하가 적은 호스트의 적합한 대기열에 작업을 작업 지정합니다. 작업이 특정 대기열에 제출되는 경우 작업은 이 대기열 및 그의 호스트에 구속되며, 따라서 Sun Grid Engine 데몬이 부하가 더 적거나 더 잘 맞는 장치를 선택할 수 없습니다.

클라이언트 명령

Sun Grid Engine의 명령줄 사용자 인터페이스는 사용자가 대기열을 관리하고 작업을 제출 및 삭제하고 작업 상태를 검사하고 대기열 및 작업을 일시중단/활성화할 수 있는 보조 프로그램(명령)의 세트입니다. Sun Grid Engine 시스템은 다음 보조 프로그램 세트를 사용합니다.

- **qacct** - 이 명령은 클러스터 로그파일에서 임의 계정 정보를 추출합니다.
- **qalter** - 이 명령은 제출되었지만 보류 중인 작업의 속성을 변경합니다.
- **qconf** - 이 명령은 클러스터 및 대기열 구성을 위한 사용자 인터페이스를 제공합니다.
- **qdel** - 이 명령은 사용자, 운영자 또는 관리자에게 작업 또는 서버세트에 신호를 보낼 수 있는 방법을 제공합니다.
- **qhold** - 이 명령은 제출된 명령이 실행되지 않게 합니다.
- **qhost** - 이 명령은 Sun Grid Engine 실행 호스트에 관한 상태 정보를 표시합니다.
- **qlogin** - 이 명령은 부하가 적고 적합한 호스트를 자동으로 선택하여 telnet 또는 비슷한 로그인 세션을 시작합니다.
- **qmake** - 이 명령은 표준 UNIX make 기능의 대체입니다. 적합한 시스템의 클러스터 사이에 독립된 make 단계를 분배하는 make 기능을 확장합니다.
- **qmod** - 이 명령은 소유자가 대기열을 일시중단 또는 활성화할 수 있게 합니다(이 대기열과 연관된 현재 활동하는 모든 프로세스도 신호됩니다).
- **qmon** - 이 명령은 X-windows Motif 명령 인터페이스와 모니터링 기능을 제공합니다.
- **qresub** - 이 명령은 실행 중이거나 보류 중인 작업을 복사하여 새 작업을 작성합니다.
- **qr1s** - 이 명령은 이전에 할당된, 예를 들어 qhold(위 참조)를 통해 보유에서 작업을 해제합니다.

- `qrsh` - 이 명령은 다음과 같은 다양한 목적을 위해 사용할 수 있습니다.
 - Sun Grid Engine 시스템을 통한 대화식 응용 프로그램의 원격 실행 제공—표준 UNIX 기능 `rsh`와 비교할 수 있음
 - 실행 시에 터미널 I/O(표준/오류 출력 및 표준 입력) 및 터미널 제어를 지원하는 일괄처리 작업의 제출 허용
 - 작업이 완료될 때까지 계속 활동하는 일괄처리 작업 제출 제공
 - 병렬 작업의 작업의 Sun Grid Engine 소프트웨어 제어 원격 실행 허용
- `qselect` - 이 명령은 지정된 선택 기준에 해당하는 대기열 이름의 목록을 인쇄합니다. `qselect`의 출력은 대개 선택한 대기열 세트에 조치를 적용하기 위해 다른 Sun Grid Engine 명령에 공급됩니다.
- `qsh` - 이 명령은 부하가 적은 호스트에서 대화식 셸(`xterm`에 있는)을 엽니다. 모든 종류의 대화식 작업을 이 셸에서 실행할 수 있습니다.
- `qstat` - 이 명령은 클러스터와 연관된 모든 작업 및 대기열의 상태 목록을 제공합니다.
- `qsub` - 이 명령은 Sun Grid Engine 시스템에 작업을 제출하기 위한 사용자 인터페이스입니다.
- `qtcssh` - 이 명령은 널리 알려지고 사용되는 Unix C 셸(`csh`) 변종인 `tcsh`에 대한 완전히 호환되는 대체입니다. 지정된 응용 프로그램의 실행을 Sun Grid Engine 소프트웨어를 통해 적합하고 부하가 적은 호스트에 투명하게 분배하는 확장 기능을 갖는 명령 셸을 제공합니다.

모든 프로그램은 `sge_commd`를 통해 `sge_qmaster`와 통신합니다. 이것은 그림 1-1에 묘사된 Sun Grid Engine 시스템의 구성요소 상호작용의 개요 보기에 반영됩니다.

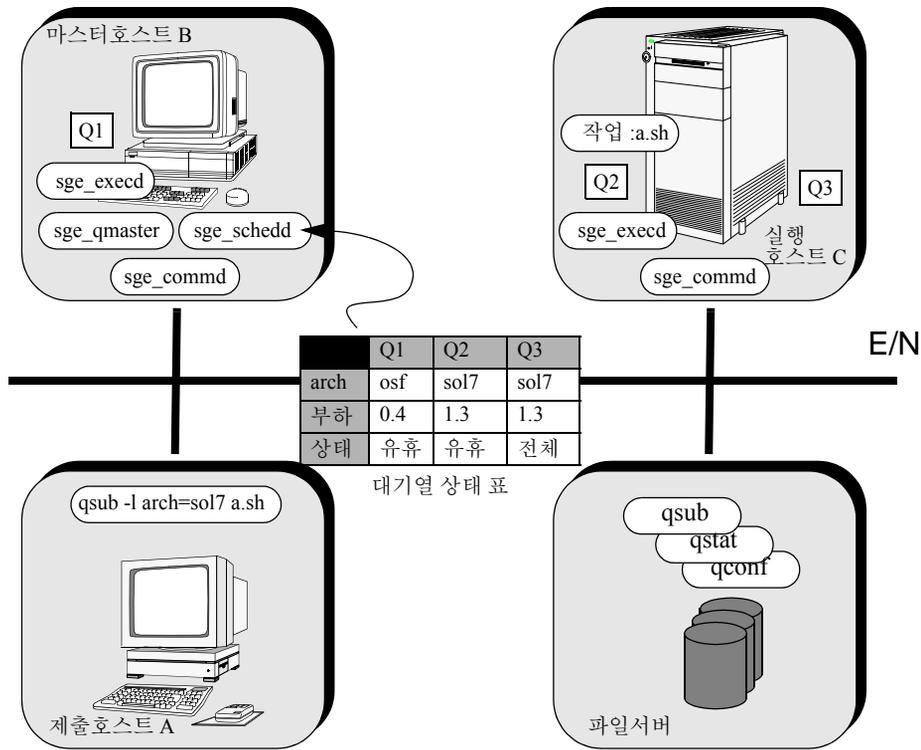


그림 1-1 Sun Grid Engine 시스템의 구성요소 상호작용

QMON, Sun Grid Engine 그래픽 사용자 인터페이스

그래픽 사용자 인터페이스(GUI) 도구인 QMON을 사용하여 대부분의—모두가 아닌 경우에—Sun Grid Engine 5.3 작업을 수행할 수 있습니다. 그림 1-2는 QMON 기본 메뉴를 표시하며, 이 메뉴가 종종 사용자 및 관리자 기능 모두에 대한 시작점입니다. 기본 메뉴의 각

아이콘은 사용자가 다양한 작업을 시작하기 위해 누르는 GUI 단추입니다. 마우스 포인터를 단추 위로 이동할 때 화면에 텍스트로 나타나는 각 단추의 이름도 단추 기능을 설명합니다.

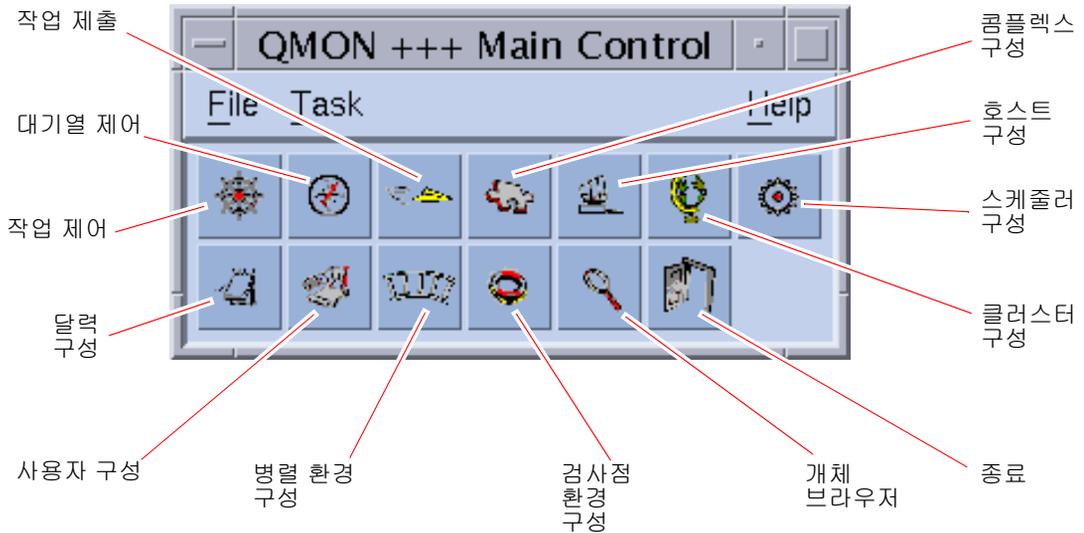


그림 1-2 QMON 기본 메뉴, 정의

QMON 사용자 정의

qmon의 모습과 느낌은 대부분 특별히 설계된 자원 파일에 의해 정의됩니다. 합리적인 기본값이 컴파일되고 샘플 자원 파일이 <sgc-루트>/qmon/Qmon에 있습니다.

클러스터 관리가 qmon 특정 자원 정의를 표준 .Xdefaults 또는 .Xresources 파일에 포함시키거나 사이트 특정 Qmon 파일을 XAPPLRESDIR 같은 표준 검색 경로에 의해 참조되는 위치에 넣어서 /usr/lib/X11/app-defaults/Qmon 같은 표준 위치에 사이트 특정 기본값을 설치할 수 있습니다. 위의 경우 중 하나가 해당하는지 여부를 관리자에게 문의하십시오.

또한 사용자가 Qmon 파일을 홈 디렉토리(또는 개인 XAPPLRESDIR 검색 경로에 의해 지시되는 다른 위치에) 복사하고 수정하거나 사용자의 개인 .Xdefaults 또는 .Xresources 파일에 필요한 자원 정의를 포함시켜서 개인 환경 설정을 구성할 수 있습니다. Qmon 개인 자원 파일은 또한 운영 중에 xrdb 명령을 통해 또는 X11 환경의 시작시에(예를 들어 .xinitrc 자원 파일에) 설치될 수 있습니다.

가능한 사용자 정의에 대한 자세한 정보는 샘플 Qmon 파일의 명령줄을 참조하십시오.

qmon을 사용자 정의하는 또 다른 방법은 그림 5-3 및 그림 5-13에 표시된 작업 제어 및 대기열 제어 사용자 정의 대화 상자에 대해 설명되었습니다. 두 대화 상자 모두에서 저장 단추를 사용하여 사용자 정의 대화 상자로 구성된 필터링 및 표시장치 정의를 사용자의 홈 디렉토리에 있는 `.qmon_preferences` 파일에 저장할 수 있습니다. 재시작할 때 qmon은 이 파일을 읽고 이전에 정의된 작동을 재활성화합니다.

Sun Grid Engine 용어의 용어집

이 용어집은 Sun Grid Engine의 문맥 및 일반적으로 자원 관리에서 자주 사용되는 용어에 대한 간략한 개요를 제공합니다. 정의된 용어 중 많은 용어가 지금까지 사용되지 않았지만 Sun Grid Engine 문서의 다른 부분에서 나타날 것입니다.

- 검사점** 작업의 실행 상태를 소위 *검사점*에 저장하여 작업이 중단되고 정보 및 이미 완료된 작업의 유실 없이 나중에 재개될 수 있게 하는 절차. 실행이 재개하기 전에 검사점이 다른 호스트로 이동되는 경우 이 프로세스를 *이주*라고 부릅니다.
- 검사점 환경** 특정 검사점 방법과 연관되는 이벤트, 인터페이스 및 조치를 정의하는 Sun Grid Engine 구성 엔티티.
- 관리자** Sun Grid Engine의 모든 측면을 조작할 수 있는 사용자. 마스터 호스트 및 관리 호스트로 선언되는 다른 모든 시스템의 슈퍼유저가 관리자 특권을 갖습니다. 관리자 특권은 루트가 아닌 사용자 계정에도 할당할 수 있습니다.
- 그룹** UNIX 그룹.
- 대기열** Sun Grid Engine 실행 호스트에서 동시에 실행하도록 허용되는 특정 클래스 및 많은 작업의 컨테이너.
- 병렬 작업** 둘 이상의 밀접하게 상관되는 작업으로 구성되는 작업. 작업은 복수 호스트 사이에 분배될 수 있습니다. 병렬 작업은 대개 공유 메모리 또는 메모리 전달(MPI, PVM) 같은 통신 도구를 사용하여 작업을 동기화하고 상관시킵니다.
- 병렬 환경** Sun Grid Engine 구성 항목으로, Sun Grid Engine이 병렬 작업을 올바르게 취급하기 위해 필요한 인터페이스를 정의합니다.
- 사용자** 최소한 하나의 제출 호스트와 실행 호스트에서 유효한 로그인을 갖는 경우 Sun Grid Engine에 작업을 제출하고 작업을 실행할 수 있습니다.
- 사용자세트(userset)** 액세스 목록(아래 참조).
- 셀** 별도의 구성과 마스터 시스템을 갖는 개별 Sun Grid Engine 클러스터. 셀을 사용하여 개별 관리 장치를 느슨하게 결합할 수 있습니다.
- 소유자** 자신이 소유하는 대기열을 일시중단/일시중단 해제 및 사용 안함/사용할 수 있는 사용자. 일반적으로 사용자는 자신의 워크스테이션에 상주하는 대기열의 소유자입니다.

소프트 자원 요구사항	작업에 필요하지만 작업이 시작되기 전에 할당될 필요는 없는 자원. 사용할 수 있는 기초로서 작업에 할당됩니다. <i>하드 자원 요구사항</i> 과 대조.
액세스 목록	대기열 같은 자원 또는 특정 호스트에 대한 액세스가 허용 또는 거부되는 사용자 및 UNIX 그룹의 목록. 사용자 및 그룹은 복수 액세스 목록에 속할 수 있으며 동일한 액세스 목록이 다양한 문맥에서 사용될 수 있습니다.
우선순위	다른 작업과 비교되는 Sun Grid Engine 작업의 상대적인 중요도 레벨.
운영자	구성을 변경할 수 없지만 조작을 유지보수하는 것으로 제안되는 점을 제외하고는 관리자와 동일한 명령을 수행할 수 있는 사용자.
이주	작업의 실행이 재개하기 전에 한 호스트에서 다른 호스트로 체크포인트를 이동하는 프로세스.
일시중단	실행 중인 작업을 보유하고 있지만 실행 시스템에 보존하는 프로세스(작업이 중단되는 검사점과 대조). 일시중단된 작업은 여전히 스왑 메모리 또는 파일 공간 같은 일부 자원을 소비합니다.
자원	실행 중인 작업에 의해 소비되거나 점유되는 연산 장치. 메모리, CPU, I/O 대역폭, 파일 공간, 소프트웨어 사용권 등이 일반적인 예입니다.
작업	일괄처리 작업은 사용자 간섭 없이 실행할 수 있는 UNIX 셸 스크립트이며 터미널에 대한 액세스가 필요 없습니다. 대화식 작업은 사용자 상호작용을 위한 <i>xterm</i> 창을 열거나 각각 원격 로그인 세션과 동등한 것을 제공하는 Sun Grid Engine 명령인 <i>qsh</i> 또는 <i>qlogin</i> 으로 시작되는 세션입니다.
작업 배열	독립적인 동일한 작업의 범위로 구성되는 작업. 각 작업은 개별 작업과 아주 유사합니다. 작업 배열 업무는 고유한 작업 ID(정수)로만 구별됩니다.
작업 클래스	어떤 의미에서 동등하며 유사하게 취급되는 작업의 집합. Sun Grid Engine에서 작업 클래스는 대응하는 작업의 동일한 요구사항과 해당 작업에 적합한 대기열의 특성에 의해 정의됩니다.
정책	Sun Grid Engine 관리자가 Sun Grid Engine의 작동을 정의하는 데 사용할 수 있는 규칙 및 구성의 집합. 정책은 Sun Grid Engine에 의해 자동으로 구현됩니다.
콤플렉스	대기열, 호스트 또는 전체 클러스터와 연관될 수 있는 속성 세트.
클러스터	Sun Grid Engine 기능이 발생하는 호스트라고 부르는 시스템의 컬렉션.
하드 자원 요구사항	작업이 시작될 수 있기 전에 반드시 할당되어야 하는 자원. <i>소프트 자원 요구사항</i> 과 대조.
호스트	Sun Grid Engine 기능이 발생하는 호스트.

II부 시작하기

*Sun Grid Engine 5.3 관리 및 사용 설명서*의 이 부분은 하나의 장으로 구성되어 있습니다.

■ **제 2장** – 15 페이지, “설치”

이 장에는 Sun Grid Engine 5.3 제품의 이전 버전을 새 릴리스로 업그레이드하기 위한 지침과 함께 제품의 최초 설치에 대한 지침이 포함되어 있습니다.

설치

이 장은 다음의 세 가지 설치 작업에 대한 상세한 지침을 설명하고 제공합니다.

- “빠른시작” 설치—모든 사이트에 대해 권장되지는 않음(15 페이지, “빠른시작 설치 개요” 참조)
- Sun Grid Engine 5.3 소프트웨어의 전체 최초 설치(22 페이지, “전체 설치 개요” 참조)
- 특수 암호화 기능이 있는 보안 설치(34 페이지, “CSP 기반 시스템 설치 및 설정 방법” 참조)

참고 - 이 장의 지침은 사용자가 Solaris™ 운영 환경을 실행 중인 컴퓨터에 소프트웨어를 설치한다고 가정합니다. Sun Grid Engine이 실행하는 다른 운영 체제 구조에 의해 작성되는 기능성의 모든 차이점이 <sgе 루트>/doc 디렉토리에 문자열로 시작하는 arc_depend_ 파일에 문서화됩니다. 파일 이름의 나머지는 파일의 설명이 적용되는 운영 체제 구조를 나타냅니다.

빠른시작 설치 개요

참고 - 이 절에서는 빠른시작 설치 절차를 사용할 수 있도록 사이트에 보편화되어야 하는 조건에 대해 설명합니다. 사용자 환경이 약속된 모든 필요조건을 만족시키지 않으면 빠른 시작 설치 절차를 사용할 수 없습니다. 이런 경우, 더 제한된 조건 아래에서 Sun Grid Engine 5.3 소프트웨어를 설치하는 방법에 대한 자세한 정보는 22 페이지, “전체 설치 개요” 절로 옮깁니다.

빠른시작 설치 필요조건

참고 – 이들 지침은 최초 Sun Grid Engine 5.3 설치만을 위한 것입니다. Sun Grid Engine 제품의 이전 버전의 기존 설치를 업그레이드하는 방법에 대한 지침은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 릴리스 노트*를 참조하십시오.

이 절에서는 빠른시작 설치 절차를 수행하기 전에 사이트에 있거나 작성되어야 하는 몇 가지 조건들에 대해 설명합니다.

- 설치 계정
- 설치 디렉토리
- 통신 포트 번호

설치 계정

빠른시작 설치 절차를 위해서는 *관리자* 계정이 존재해야 합니다. 또는 사용자가 다음 안내지침에 따라 작성해야 합니다.

- Administrator(관리자)는 sgeadmin과 같은 관리 로그인 또는 새 로그인일 수 있습니다.

이 계정은 Sun Grid Engine 설치 및 스폰링 디렉토리의 모든 파일들을 소유하게 되며 일단 설치되면 클러스터를 구성하고 관리하는 데 사용될 수 있습니다.

- 이 사용자는 루트이면 *안 됩니다*.

그러나 파일 소유권에 대해 루트를 사용하려는 경우에는 사용자 루트는 Sun Grid Engine 프로그램이 설치된 디렉토리의 모든 호스트에 대해 전체 쓰기 권한이 있어야 합니다. 대체로 공유 (NFS) 파일시스템은 사용자 루트가 쓰기 권한을 허용하도록 하는 데 사용되지 않습니다.

- 이 계정은 설치 *이전에* 존재해야 합니다.

설치 디렉토리

관리자 계정 사용할 수 있도록 준비하려면 가급적 전네트워크 공유 파일 시스템에 설치 디렉토리를 작성해야 합니다.

▼ 설치 디렉토리 생성 방법

- 다음 명령 순서를 사용하십시오.

```
% mkdir -p <설치디렉토리>  
% chown <관리사용자> <설치디렉토리>  
% chmod 755 <설치디렉토리>
```

참고 - 이것 또는 비슷한 절차로 생성된 디렉토리는 이 설명서의 나머지 부분에서 Sun Grid Engine 루트 디렉토리로 언급됩니다.

통신 포트 번호

Sun Grid Engine 시스템은 통신용 TCP 포트를 사용합니다. 클러스터의 모든 호스트는 동일한 포트 번호를 사용해야 합니다. 여러 위치 중 어느 곳에든 포트 번호를 놓을 수 있습니다. 다음 두 가지 예가 있습니다.

- NIS 서비스 또는 NIS+ 데이터베이스

서비스 데이터베이스에 다음을 추가합니다.

```
% sge_commd 535/tcp
```

- 각 시스템의 /etc/services

사용자 사이트에서 NIS를 실행하고 있지 않는 경우, 각 시스템의 /etc/services 파일에 위의 서비스들을 추가할 수 있습니다.

1024 미만이나 근처 또는 1024 이상의 포트를 묶는 응용 프로그램과의 충돌을 피하려면 600 미만의 특권 포트를 사용하는 것이 최선입니다.

배포 매체 읽기

Sun Grid Engine 5.3 소프트웨어는 인터넷 다운로드를 통해 아카이브 파일로서 배포됩니다. 배포는 매체에 직접 기록된 테이프 아카이브(tar)로 구성됩니다.

▼ Sun Grid Engine 5.3 배포 포장 제거

1. Sun Grid Engine 배포 매체를 읽으려는 호스트에 설치를 위해 선택한 계정으로 로그인합니다(16페이지, “빠른시작 설치 필요조건” 절 참조).
2. 작업중인 디렉토리를 Sun Grid Engine 루트 디렉토리로 변경합니다.
3. 배포 매체를 읽으려면 다음 명령을 사용합니다.

```
% cd sge 루트디렉토리
% tar -xvpf 배포소스
```

위 명령에서, *sge 루트디렉토리*는 Sun Grid Engine 루트 디렉토리의 경로 이름이며 *배포소스*는 하드 디스크의 테이프 아카이브 파일의 이름입니다.

사용자 클러스터용 기본 Sun Grid Engine 시스템 설치

기본 Sun Grid Engine 시스템은 *마스터 호스트* 및 *실행 호스트*의 임의 번호입니다. 실행 호스트가 마스터 호스트에 의해 그들에게 지정되어 있는 작업의 실행을 제어하는 반면 마스터 호스트는 총체적인 클러스터 활동을 제어합니다. 단일 호스트는 마스터 호스트 및 실행 호스트로 동시에 작동할 수도 있습니다.

참고 - 먼저 마스터 호스트를 설치한 후 임의의 순서로 실행 호스트를 설치합니다.

▼ 마스터 호스트 설치 방법

1. 마스터 호스트로서 시스템을 선택합니다.

마스터 호스트 시스템을 선택하는 데 다음 안내 지침을 따릅니다.

- 선택한 시스템이 다른 작업들로 과부하되어서는 안됩니다.
- 마스터 호스트는 필수 Sun Grid Engine 데몬을 시행하려면 사용 가능한 충분한 주 메모리를 공급해야 합니다.

필요한 양은 클러스터 크기 및 기대되는 시스템의 작업 수에 완전히 따릅니다. 클러스터에 대해 최고 수십 개의 호스트 및 100 개의 작업 순으로, 10 MB의 메모리 여유분은 충분해야 합니다.

아주 큰 클러스터에 대해—1,000 개 이상의 호스트 순 및 여러 10,000 개의 작업들 순으로—1 GB 메모리가 필요할 수도 있습니다.

2. 선택한 시스템에 로그인합니다.

모든 성능이 기능하는 설치를 위해서는 루트 계정(파일들은 여전히 16 페이지, “빠른시작 설치 필요조건” 절에서 생성된 관리자 계정이 소유하고 있을 수도 있습니다)을 사용하여 설치해야 합니다.

테스트 설치에 있어서, Administrator 사용자로서 설치할 수도 있지만 그 때는 Administrator만 작업을 실행할 수 있으며 Sun Grid Engine 시스템이 시스템 부하 및 시스템 제어 모니터링에 대해서는 성능을 제한하게 됩니다.

3. 디렉토리(cd)를 Sun Grid Engine 루트 디렉토리로 변경합니다.

4. 다음 명령을 사용하여 마스터 호스트 설치 절차를 실행합니다.

```
% ./install_qmaster
```

오류가 있는 경우, 설치 절차가 오류 상태에 대한 설명을 인쇄합니다

5. 설치하려는 호스트의 이름으로 설치 스크립트에 응답합니다.

설치 스크립트가 처음 설치하려는 호스트의 목록을 제공하도록 요구합니다. 이러한 호스트들은 *제출* 호스트 및 *관리* 호스트로서 추가되므로, 해당하는 모든 호스트를 나열하십시오.

실행 호스트를 설치하려면(다음 절 참조) 모든 호스트가 관리 호스트여야 합니다. 많은 호스트에 Sun Grid Engine 프로그램을 설치하려는 경우, 설치 스크립트는 해당 하나의 호스트를 사용하여 모든 호스트 이름 목록을 포함하는 파일에 대한 경로를 제공할 기회를 줍니다.

6. 남아있는 모든 설치 스크립트 프롬프트에 응답하십시오.

설치 절차에는 일부 추가 정보가 필요합니다. 대부분의 질문들은 유용한 기본값을 제공하는데, 이것은 Return을 눌러 확인할 수 있습니다.

▼ 실행 호스트 설치 방법

1. 다음 안내 지침에 따라 설치에 사용되는 계정을 선택하십시오.

- 마스터 호스트 설치에서와 같이, 모든 Sun Grid Engine 기능에 액세스하려면 루트 계정을 사용하여 실행 호스트를 설치해야 합니다.

루트로 설치하면 16 페이지, “빠른시작 설치 필요조건” 절에서 생성된 Administrator 계정에 의한 파일 소유권이 여전히 유지됩니다.

- Administrator 계정으로 설치하면 테스트 목적으로만 유용할 뿐이며 Administrator만 작업을 실행하며 사용자를 제한합니다. Sun Grid Engine 프로그램이 파일 시스템 모니터링 및 제어 기능을 제공할 수 없게 합니다.

2. 설치용으로 선택한 계정으로 마스터 호스트 설치 절차 중 지정된 실행 호스트 중 하나에 로그인합니다.
3. 디렉토리(cd)를 Sun Grid Engine 루트 디렉토리로 변경합니다.
4. 실행 호스트 설치 절차를 시작하려면 다음 명령을 입력합니다.

```
% ./install_execd
```

어떤 오류이든지 설치 절차에 의해 표시됩니다.

5. 설치 스크립트 프롬프트에 응답하십시오.

설치 절차에서 기본 대기열이 사용자 호스트용으로 구성되어야 하는지를 질문합니다. 이 경우에 정의된 대기열은 20 페이지, “기본 시스템 구성” 절에 설명된 특징을 갖게 됩니다.

6. 설치 스크립트가 실행 호스트 설치 절차를 성공적으로 마쳤음을 통지할 때, 마찬가지로 마스터 호스트 설치 중 이름을 지정한 각 실행 호스트를 진행합니다.

목록이 끝나자마자 기본 Sun Grid Engine 시스템이 사용자 클러스터에 구성되며 사용할 준비가 됩니다. 프로그램을 사용하여 연습을 하려면 68 페이지, “명령줄에서 간단한 작업을 실행하는 방법” 절을 참조하십시오.

다음 절은 설치된 기본 구성에 대한 개요입니다.

기본 시스템 구성

참고 - 다음은 빠른시작 설치 절차로 사용자 환경에 구성된 Sun Grid Engine 시스템에 대한 설명입니다. 이것은 테스트 목적의 최소 설정이며 언제라도 변경되거나 확장될 수 있습니다.

마스터 호스트 및 실행 호스트를 완전히 설치한 후 다음 기본 Sun Grid Engine 시스템이 사용자 클러스터에 구성되어 있습니다.

■ 마스터 호스트

마스터 호스트 설치 절차를 실행하는 호스트는 사용자 클러스터의 마스터 호스트가 되도록 구성됩니다. 실패할 경우 마스터 호스트의 업무를 인계하도록 구성되는 새도우 마스터 호스트는 없습니다.

■ 실행 호스트

마스터 호스트 설치 중 Sun Grid Engine 실행 에이전트를 설치하려는 시스템 목록에 대해 묻습니다. 이러한 실행 호스트 설치 중 설치 절차가 이러한 호스트에 자동으로 대기열을 생성하도록 허용할 수 있습니다. 대기열은 해당 특수 호스트에서 실행될 수 있는 작업의 프로파일(속성 및 필요조건 목록)을 설명합니다. 기본적으로 실행 호스트에 대해 구성되어 있는 대기열은 다음의 중요한 특징을 갖습니다.

- 대기열 이름: <비적격호스트이름>.q
- 슬롯 (동시 작업): <프로세서수>
- 대기열은 작업에 (메모리, CPU시간 등과 같은) 제한된 시스템 자원을 제공합니다.
- 대기열은 특수 사용자나 사용자 그룹에 대해 어떠한 액세스 제한도 강요하지 않습니다. 유효한 계정이 있는 사용자는 대기열에서 작업을 실행할 수 있습니다.
- CPU 당 1.75의 부하 임계값이 구성됩니다(즉, 각 CPU에 대한 액세스 권한을 갖기 위해 평균적으로 시도하는 1.75 프로세스).

참고 - 다른 Sun Grid Engine 구성과 같이 대기열 구성은 시스템이 작동하는 동안 나중 어느 단계에서든 진행 상태로 변경될 수 있습니다.

참고 - 또한 마스터 호스트에서 실행 호스트 설치 절차를 실행했다면 마스터 호스트는 마스터 및 실행 호스트 둘 다로서 작동합니다.

■ 관리 계정 및 호스트

마스터 호스트 및 모든 실행 호스트는 관리적 Sun Grid Engine 명령을 실행할 수 있도록 구성됩니다. Sun Grid Engine을 관리하도록 허락된 유일한 사용자들은 16 페이지, “빠른시작 설치 필요조건” 절에 설명된 Administrator 계정 및 사용자 루트입니다. 권한이 없는 사용자가 Sun Grid Engine 프로그램을 설치하는 경우, 그 사용자는 Sun Grid Engine 관리자 목록에 추가됩니다.

■ 제출 계정 및 호스트

루트 계정으로 프로그램을 설치했다면, 유효한 계정이 있는 모든 사용자는 Sun Grid Engine 작업을 제출 및 제어할 수 있습니다. Sun Grid Engine 소프트웨어를 설치한 상태에서 사용자는 다른 방법으로 권한이 허용된 유일한 사용자입니다(16 페이지, “빠른시작 설치 필요조건” 참조). 작업 제출, Sun Grid Engine 시스템 활동 제어 또는 작업 삭제 업무는 마스터 호스트나 실행 호스트 중 하나에서 실행될 수 있습니다.

■ 데몬

다음 데몬들은 다른 호스트에 시스템을 설치하는 동안 시작되거나 각각 정상 시스템 작동 중에 실행될 수도 있습니다.

- sge_qmaster는 마스터 호스트에서만 실행합니다. 이것은 중앙 클러스터 활동 제어 데몬입니다.
- sge_schedd는 또한 마스터 호스트에서만 실행됩니다. 이 데몬은 Sun Grid Engine 클러스터에서 작업부하를 분배하는 데 책임이 있습니다.

- `sge_execd`는 실행 호스트에서 작업을 실행하는 데 책임이 있으므로 모든 실행 호스트에서 실행 중입니다.
- `sge_shepherd`의 한 인스턴스는 호스트에서 실제 실행되고 있는 각 작업에 대해 실행됩니다. `sge_shepherd`는 작업 프로세스 계층을 제어하고 작업이 끝난 후 회계 데이터를 수집합니다.
- `sge_commd`는 각 실행 호스트 및 마스터 호스트에서 실행합니다. 모든 `sge_commd`의 네트워크는 Sun Grid Engine 클러스터의 네트워크 통신의 중심을 형성합니다.

전체 설치 개요

참고 – 이들 지침은 *최초* Sun Grid Engine 5.3 설치만을 위한 것입니다. Sun Grid Engine 제품의 이전 버전의 기존 설치를 업그레이드하는 방법에 대한 지침은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 릴리스 노트*를 참조하십시오.

전체 설치의 다음 광범위한 작업으로 구성됩니다.

- Sun Grid Engine 구성 및 환경 계획
- 외부 매체에서 워크스테이션으로 Sun Grid Engine 배포 파일 읽기
- Sun Grid Engine 시스템의 마스터 호스트 및 모든 실행 호스트에서 설치 스크립트 읽기
- 관리 및 제출 호스트에 관한 정보 등록
- 설치 검증

설치는 Solaris 운영 환경에 익숙한 사람이 해야 합니다. 전체 프로세스는 다음 세 단계로 수행됩니다.

1 단계 – 계획

참고 – 향상된 보안을 갖는 시스템을 설치하려는 경우, 설치를 계속하기 전에 33 페이지, “향상된 보안으로 설치” 절을 참조하십시오.

설치의 계획 단계는 다음 작업으로 구성됩니다.

- 사용자의 Sun Grid Engine 환경이 단일 클러스터인지 셀이라 부르는 서브 클러스터의 콜렉션인지 여부 결정

- Sun Grid Engine 호스트가 될 시스템 선택. 각 시스템이 어떤 종류의 호스트(마스터 호스트, 새도우 마스터 호스트, 관리 호스트, 제출 호스트, 실행 호스트 또는 조합)인지 판별
- 모든 Sun Grid Engine 사용자가 모든 제출 및 실행 호스트에서 동일한 사용자 ID를 갖는지 확인
- Sun Grid Engine 디렉토리 구성의 형태(예를 들어, 각 워크스테이션에 전체 트리로서의 디렉토리, 교차 마운트된 디렉토리 또는 일부 워크스테이션에 부분 디렉토리 트리를 구성하는 것) 및 각 Sun Grid Engine 루트 디렉토리가 위치할 장소를 결정하십시오.
- 사이트의 대기열 구조 결정
- 네트워크 서비스가 NIS 파일로서 정의되는지 아니면 /etc/services의 각 워크스테이션에 국지인지 여부 결정
- 후속 설치 단계에서 사용할 설치 워크시트(1., 30 페이지, “설치를 시작하기 전에 아래와 비슷한 표에 설치 계획을 기록하십시오.” 참조) 완성

2 단계 – 소프트웨어 설치

설치 단계는 다음 작업으로 구성됩니다.

- 설치 디렉토리 작성 및 해당 디렉토리에 배포 파일 로드
- 마스터 호스트 설치
- 모든 실행 호스트 설치
- 모든 관리 호스트 등록
- 모든 제출 호스트 등록

3 단계 – 설치 검증

검증 단계는 다음 작업으로 구성됩니다.

- 데몬이 마스터 호스트에서 실행 중인지 확인
- 데몬이 모든 실행 호스트에서 실행 중인지 확인
- Sun Grid Engine이 간단한 명령을 실행하는지 확인
- 테스트 작업 제출

설치 계획

Sun Grid Engine 5.3 소프트웨어 설치를 시작하기 전에 사용자 환경에 완벽하게 맞는 결과를 달성하는 방법을 주의깊게 계획해야 합니다. 이 절은 나머지 절차에 영향을 줄 필수 결정을 수행하는 데 도움이 됩니다.

필수조건 작업

다음 절은 생산 Sun Grid Engine 시스템을 설치하는 데 필요한 정보를 설명합니다.

설치 디렉토리 <sgе 루트>

Sun Grid Engine 배포 매체의 내용을 읽어들이 디렉토리를 준비하십시오. 이 디렉토리는 Sun Grid Engine 루트 디렉토리라고 부르며, 나중에 Sun Grid Engine 시스템이 동작하는 동안 현재 클러스터 구성 및 디스크에 스푼되어야 하는 모든 추가 데이터를 저장하는 데 사용됩니다.

모든 호스트에서 올바른 참조인 디렉토리에 대한 경로 이름을 사용하십시오. 예를 들어 파일 시스템이 자동 마운터를 사용하여 마운트되는 경우 <sgе 루트>를 /tmp_mnt/usr/SGE가 아닌 /usr/SGE로 설정하십시오. (이 책 전체에서 설치 디렉토리를 참조할 때 <sgе 루트> 환경 변수가 사용됩니다.)

<sgе 루트>는 Sun Grid Engine 디렉토리 트리의 최상위 레벨입니다. 셀(29 페이지, “셀”)의 각 Sun Grid Engine 구성요소는 시작시에 <sgе 루트>/<셀>/common에 대한 읽기 액세스가 필요합니다. 필수 사용권한의 설명에 대해서는 27 페이지, “파일 액세스 사용 권한” 절을 참조하십시오.

편하게 설치 및 관리하려면 Sun Grid Engine 설치 절차를 실행하려는 모든 호스트에서 이 디렉토리를 읽을 수 있어야 합니다. 예를 들어 네트워크 파일 시스템(NFS 같은)을 통해 사용할 수 있는 디렉토리를 선택할 수 있습니다. 호스트에 국지인 파일시스템을 선택하는 경우 특정 시스템에 대한 설치 절차를 시작하기 전에 각 호스트에 설치 디렉토리를 복사해야 합니다.

루트 디렉토리의 스폴 디렉토리

- Sun Grid Engine 마스터 호스트에서 스폴 디렉토리는 <sgc 루트>/<셀>/spool/qmaster 및 <sgc 루트>/<셀>/spool/schedd 아래에서 유지보수됩니다.
- 각 실행 호스트에서 <sgc 루트>/<셀>/spool/<exec_host>라는 스폴 디렉토리가 유지보수됩니다.

다른 시스템에 이들 디렉토리를 내보낼 필요는 없습니다. 그러나 전체 <sgc 루트> 트리를 내보내고 마스터 및 모든 실행 가능 호스트에 대해 쓰기 액세스 가능하게 하면 관리의 용이성이 향상됩니다.

디렉토리 구성

Sun Grid Engine 디렉토리 구성의 형태 및 각 Sun Grid Engine 루트 디렉토리인 <sgc 루트>가 위치할 장소를 결정하십시오.

참고 - (이전 설치의 모든 중요한 정보를 보존할 수 있지만) 설치 디렉토리 및/또는 스폴 디렉토리를 변경하려면 기본적으로 시스템을 새로 설치해야 하기 때문에 먼저 적합한 설치 디렉토리를 선택하는 데 특별한 주의를 기울여야 합니다.

기본적으로 Sun Grid Engine 설치 절차는 Sun Grid Engine 시스템, 매뉴얼, 스폴 영역 및 구성 파일을 설치 디렉토리 아래에 디렉토리 계층(그림 2-1, 26 페이지, “샘플 디렉토리 계층” 참조)으로 설치합니다. 이 기본 작동을 허용하는 경우 27 페이지, “파일 액세스 사용 권한”에 설명된 액세스 사용 권한을 허용하는 디렉토리를 설치/선택해야 합니다.

1차 설치 중에 다른 위치에 배치할 스폴 영역을 선택할 수 있습니다(지침에 대해서는 6장, 135 페이지, “호스트 및 클러스터 구성” 참조).

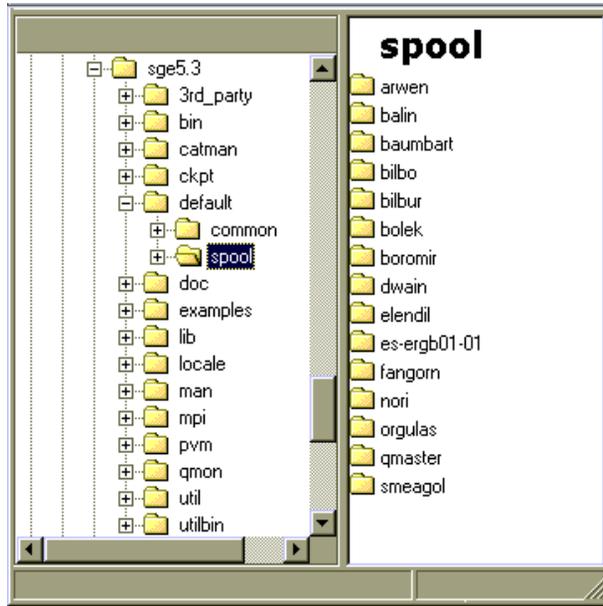


그림 2-1 스폴 디렉토리 계층

디스크 공간 요구 사항

Sun Grid Engine 디렉토리 트리는 다음과 같이 특정한 고정 디스크 공간 요구사항을 갖습니다.

- 이진을 포함하지 않는 설치 키트(문서 포함)을 위한 40MB
- 각 이진 세트에 대한 10-15MB(이진이 대략 35MB를 소비하는 아키텍처 Cray의 경우는 예외)

Sun Grid Engine 로그 파일에 대한 이상적인 디스크 공간은 다음과 같습니다.

- 클러스터의 크기에 따라서 마스터 호스트 스폴 디렉토리를 위한 30-200MB
- 각 실행 호스트를 위한 10-20MB

참고 - 마스터 호스트 및 실행 호스트를 위한 스폴 디렉토리는 구성 가능하며 <sge-루트>(기본적으로 위치하는 곳)에 상주할 필요는 없습니다. 스폴 디렉토리의 위치 변경은 1차 설치 후에 수행되어야 합니다(지침에 대해서는 6장, 135 페이지, “호스트 및 클러스터 구성” 참조).

설치 계정

루트 계정이나 특권이 없는(예: 사용자 자신의) 계정에서 Sun Grid Engine을 설치할 수 있습니다. 특권이 없는 계정에 설치하는 경우 이 설치하는 해당 특정 사용자만이 Sun Grid Engine 작업을 실행하도록 허용합니다. 다른 모든 계정에 대한 액세스는 거부됩니다. 루트 계정에 설치하면 이 제한이 해결되지만 전체 설치 절차를 위해 루트 사용 권한이 필요합니다.

파일 액세스 사용 권한

루트로서 설치하는 경우 공유 파일 시스템의 모든 호스트에 대해 루트 읽기/쓰기 액세스를 구성하는 문제가 생길 수 있으며 따라서 <sgе 루트>를 네트워크 전체의 파일 시스템에 배치하는 문제가 생길 수 있습니다. 루트가 아닌 관리 사용자 계정(예를 들어 sgeadmin이라는)을 통해 모든 Sun Grid Engine 구성요소의 전체 파일 취급을 실행하도록 Sun Grid Engine 소프트웨어에 강제할 수 있습니다. 따라서 이 특정 사용자에 대해 공유 루프 파일 시스템에 대한 읽기/쓰기 액세스만이 필요합니다. Sun Grid Engine 설치 절차는 관리 사용자 계정 아래에서 파일 취급을 원하는지 여부를 묻습니다. **Yes**를 대답하고 올바른 사용자 이름을 제공하는 경우 파일 취급이 이 사용자 이름을 통해 수행됩니다. 그렇지 않으면 설치 절차를 실행하는 사용자 이름이 사용됩니다.

모든 경우에 모든 호스트에서 파일 취급에 사용되는 계정이 Sun Grid Engine 루트 디렉토리에 대해 읽기/쓰기 액세스를 갖는지 확인해야 합니다. 또한 설치 절차는 Sun Grid Engine 배포 매체를 읽어들이 호스트가 이 디렉토리에 액세스할 수 있다고 가정합니다.

네트워크 서비스

사이트의 네트워크 서비스가 NIS 파일로서 정의되는지 아니면 /etc/services의 각 워크스테이션에 대한 로컬로 정의되는지 확인하십시오. 사이트가 NIS를 사용하는 경우 서비스 NIS 맵에 항목을 추가할 수 있도록 NIS 서버 호스트를 찾으십시오.

Sun Grid Engine 서비스는 sge_commd입니다. NIS 맵에 서비스를 추가하려면 예약된 사용되지 않는 포트 번호를 선택하십시오. 그 중에 하나가 아래의 1024입니다. 다음은 sge_commd 항목의 예입니다.

```
sgе_commd 536/tcp
```

마스터 호스트

이 호스트는 Sun Grid Engine이 제어되는 호스트입니다. 이 호스트는 마스터 데몬인 `sge_qmaster`를 실행합니다. 아주 큰 클러스터(언제나 시스템에 수백 또는 수천 개의 호스트와 수만 개의 작업을 포함하는 클러스터)의 경우 1 GB 이상의 사용되지 않는 기본 메모리가 필요할 수 있으며 두 개의 CPU가 도움이 될 수 있습니다.

- 안정한 플랫폼이어야 합니다.
- 다른 처리로 과도하게 사용 중이지 *않아야* 합니다.
- Sun Grid Engine 데몬을 실행하기 위해 최소한 20 MB의 사용되지 않는 기본 메모리가 있어야 합니다.
- *선택적으로* 네트워크 통신량을 줄이기 위해 시스템에 로컬인 Sun Grid Engine 디렉토리 `<sge 루트>`가 있어야 합니다.

새도우 마스터 호스트

이들 호스트는 마스터 호스트 또는 마스터 데몬이 실패하는 경우에 `sge_qmaster`의 기능을 백업합니다. 새도우 마스터 호스트가 되려면 시스템이 다음 특성을 가져야 합니다.

- `sge_shadowd`를 실행해야 합니다.
- `sge_qmaster`의 상태, 작업 및 디스크에 기록되는 대기열 구성 정보를 공유해야 합니다. 특히 새도우 마스터 호스트는 `sge_qmaster`의 스펴 디렉토리 및 `<sge 루트>/<스플>/common` 디렉토리에 대한 읽기/쓰기 루트 또는 관리 사용자 액세스가 필요합니다.
- `<sge 루트>/<스플>/common/shadow_masters` 파일에 새도우 마스터 호스트로서 호스트를 정의하는 행이 있어야 합니다.

새도우 마스터 호스트 기능은 이들 조건이 만족되자마자 호스트에 대해 활성화됩니다. 따라서 호스트를 새도우 호스트로 만들기 위해 Sun Grid Engine 데몬을 재시작할 필요가 없습니다.

실행 호스트

이들 호스트는 Sun Grid Engine에 제출되는 작업을 실행합니다. 각 실행 호스트에서 설치 스크립트를 실행합니다.

관리 호스트

Sun Grid Engine 운영자 및 관리자는 이들 호스트에서 대기열 재구성 또는 Sun Grid Engine 사용자 추가 같은 관리 업무를 수행합니다. 마스터 호스트 설치 스크립트가 자동으로 마스터 호스트를 관리 호스트로 만듭니다.

제출 호스트

제출 호스트에서 Sun Grid Engine 작업을 제출 및 제어할 수 있습니다. 마스터 호스트 설치 스크립트가 자동으로 마스터 호스트를 제출 호스트로 만듭니다.

셀

Sun Grid Engine를 단일 클러스터 또는 셀이라고 부르는 느슨하게 결합된 클러스터의 컬렉션으로 설정할 수 있습니다. SGE_CELL 환경 변수가 참조되는 클러스터를 표시합니다. Sun Grid Engine이 단일 클러스터로서 설치될 때 SGE_CELL이 설정되지 않으며 값 default가 셀 값에 대해 가정됩니다.

사용자 이름

Sun Grid Engine이 작업을 제출하는 사용자가 작업을 제출할 권한이 있는지 검증하고 필요한 실행 호스트를 사용하기 위해서는 사용자의 이름이 관련되는 제출 및 실행 호스트에서 동일해야 합니다. 이 요구사항은 일부 시스템에서 사용자 ID 변경이 필요할 수 있습니다.

참고 - 마스터 호스트의 사용자 ID는 사용 권한 점검과는 관련이 없으며 일치하거나 심지어 존재할 필요가 없습니다.

대기열

사용자 사이트의 수요에 맞는 대기열 구조를 계획하십시오. 이것은 어떤 대기열이 어떤 실행 호스트에 존재해야 하는지, 순차, 대화식, 병렬 및 기타 작업 유형에 대해 대기열이 필요한지 여부, 각 대기열에서 필요한 작업 슬롯의 수 및 기타 대기열 구성 결정사항 판별을 의미합니다.

또한 Sun Grid Engine 관리자가 설치 절차가 시스템에 소개되기에 적합하고 차후 조정을 위한 시작점으로 적합한 기본 대기열 구조를 작성하도록 만들 수 있습니다.

참고 - Sun Grid Engine가 설치되는 디렉토리라 해도 Sun Grid Engine 설치 절차에 의해 작성된 모든 설정은 시스템의 동작 중에 변경될 수 있습니다.

이미 Sun Grid Engine에 익숙하거나 이전에 클러스터에 부과하려는 대기열 구조를 결정한 경우 설치 절차가 사용자 대신 기본 대기열 구조를 설치하도록 허용하지 않아야 합니다. 대신, 해당 대기열 구조를 지정하는 문서를 준비하고 설치 프로세스를 완료한 직후에 7장, 157 페이지, “대기열 구성 및 대기열 쉘런더”로 가야 합니다.

▼ 설치 계획 방법

1. 설치를 시작하기 전에 아래와 비슷한 표에 설치 계획을 기록하십시오.

매개변수	값
<sgе 루트>	
관리 사용자	
관리 그룹	
sgе_commd 포트 번호	
마스터 호스트	
새도우 마스터 호스트	
실행 호스트	
관리 호스트	
제출 호스트	

2. 위에서 정의한 대로 액세스 사용 권한을 설정하여 Sun Grid Engine 배포와 스펴 및 구성 파일을 포함한 파일 시스템과 디렉토리가 적절하게 설정되는지 확인하십시오.

▼ 배포 매체를 읽는 방법

Sun Grid Engine 소프트웨어는 인터넷 다운로드를 통해 아카이브 파일로 배포됩니다. 또한 웹 배포는 결국 compress(확장자 .Z) 또는 gzip(확장자 .gz)로 압축된 tar 파일 형식으로 제공됩니다. 다음 단계를 진행하기 전에 파일 압축을 해제해야 합니다. (uncompress 또는 gunzip을 사용하십시오.)

1. 배포 매체에 대한 액세스를 제공하고 시스템(되도록이면 파일 서버에 직접 연결을 갖는 시스템)에 로그인합니다.
2. 24 페이지, “설치 디렉토리 <sgc루트>” 절에 설명된 것처럼 설치 디렉토리를 생성하여 Sun Grid Engine 설치 키트에서 읽고 설치 디렉토리에 대한 액세스 권한이 적절하게 설정되어 있는지 확인합니다.
3. 명령 프롬프트에서 다음 절차를 실행합니다.

```
% cd 설치디렉토리
% tar -xvpf 배포소스
```

여기서 *설치디렉토리*는 설치 디렉토리의 경로 이름이고 *배포소스*는 하드 디스크의 압축 해제된 테이프 아카이브 파일의 이름입니다. 이 명령은 Sun Grid Engine 설치 키트를 읽어 들입니다.

▼ 마스터 호스트 설치 방법

1. 마스터 호스트에 root로서 로그인합니다.
2. 설치 키트가 상주하는 디렉토리가 마스터 호스트에서 표시할 수 있는지 여부에 따라서 다음 중 하나를 수행합니다.
 - a. 설치 키트가 상주하는 디렉토리가 마스터 호스트에서 *표시되는* 경우 디렉토리를 설치 디렉토리로 변경(cd)한 후 3 단계로 갑니다.
 - b. 디렉토리가 표시되지 *않고* 표시할 수 없는 경우 다음을 수행하십시오.
 - i. 마스터 호스트에 로컬 설치 디렉토리를 작성합니다.
 - ii. 설치 키트를 네트워크를 통해(예를 들어 ftp 또는 rcp를 사용하여) 로컬 설치 디렉토리에 복사합니다.
 - iii. 디렉토리를 로컬 설치 디렉토리로 변경(cd)합니다.
3. 다음 명령어를 실행합니다.

참고 - 인증 보안 프로토콜 방법을 통해 설치를 수행 중인 경우 다음 명령에 `-csp` 플래그를 추가해야 합니다(34 페이지, “CSP 기반 시스템 설치 및 설정 방법”을 참조하십시오).

```
% ./install_qmaster
```

이것은 마스터 설치 절차를 시작합니다. 여러 가지 질문을 받게 되며 일부 관리 조치를 수행해야 할 수도 있습니다. 질문과 조치 항목은 자체 설명적입니다.

참고 - 관리 업무를 실행하기 위해 보조 터미널 세션을 활성화하는 것이 편리합니다.

마스터 설치 절차는 `sge_qmaster` 및 `sge_schedd`에 필요한 적당한 디렉토리 계층을 작성합니다. 절차가 마스터 호스트에서 Sun Grid Engine 구성요소인 `sge_commd`, `sge_qmaster` 및 `sge_schedd`를 시작합니다. 마스터 호스트는 또한 관리 및 제출 권한을 갖고 호스트에 등록됩니다.

이상이 있으면 언제든지 설치 절차를 중단하고 반복할 수 있습니다.

▼ 실행 호스트 설치 방법

1. 실행 호스트에 `root`로서 로그인합니다.
2. 마스터 설치의 경우와 같이 설치 키트를 로컬 설치 디렉토리에 복사하거나 네트워크 설치 디렉토리를 사용합니다.
3. 디렉토리(`cd`)를 설치 디렉토리로 변경하고 다음 명령을 실행하십시오.

참고 - 인증 보안 프로토콜 방법을 통해 설치를 수행 중인 경우 다음 명령에 `-csp` 플래그를 추가해야 합니다(34 페이지, “CSP 기반 시스템 설치 및 설정 방법”을 참조하십시오).

```
% ./install_execd
```

이것은 실행 호스트 설치 절차를 시작합니다. 실행 호스트 설치 절차의 작동과 처리는 마스터 호스트에 대한 것과 매우 비슷합니다.

4. 설치 스크립트의 프롬프트에 응답합니다.

참고 - 작업의 실행을 위해 마스터 호스트를 사용할 수도 있습니다. 단지 마스터 시스템에 대해 실행 호스트 설치를 수행하면 됩니다.

참고 - 매우 느린 시스템을 마스터 호스트로 사용하는 경우 또는 클러스터가 상당히 큰 경우 마스터 시스템을 마스터 업무에만 사용해야 합니다.

실행 호스트 설치 절차는 `sge_execd`에 필요한 적당한 디렉토리 계층을 작성합니다. 절차가 실행 호스트에서 Sun Grid Engine 구성요소인 `sge_commd` 및 `sge_execd`를 시작합니다.

▼ 관리 및 제출 호스트 설치 방법

마스터 호스트는 내재적으로 관리 업무를 실행하고 작업을 제출, 모니터 및 삭제하도록 허용됩니다. 관리 또는 제출 호스트로서의 어떤 추가 설치도 필요하지 않습니다. 이와는 반대로 순수한 관리 및 제출 호스트는 등록이 필요합니다.

- 관리 호스트(즉, 마스터 호스트)에서 및 관리 계정(즉, 슈퍼유저 계정)을 통해 다음 명령을 입력합니다.

```
% qconf -ah 관리호스트이름 [...]
% qconf -as 제출호스트이름 [...]
```

자세한 내용과 다른 호스트 유형을 구성하는 다른 방법에 대해서는 137 페이지, “데몬 및 호스트 정보” 절을 참조하십시오.

향상된 보안으로 설치

다음 지침을 사용하여 보다 안전하게 시스템을 설정할 수 있습니다. 이들 지침은 인증 보안 프로토콜(CSP) 기반 암호화를 사용하여 시스템을 설정하는 것을 도와 줍니다.

Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 제품 모두가 이 보안 설정을 활용할 수 있으며 이들 지침이 양 제품 모두에 적용됩니다. 간략함을 위해 이들 지침은 Sun Grid Engine 제품만을 인용합니다.

이 보다 안전한 시스템 내에서 메시지는 일반 텍스트로 메시지를 전송하는 대신 비밀 키를 사용하여 암호화됩니다. 비밀 키가 공용/개인용 키 프로토콜을 통해 교환됩니다. 사용자는 신원을 증명하기 위해 Sun Grid Engine 시스템을 통해 자신의 인증을 제시하고, 사용자가 올바른 시스템과 통신 중임을 보장하기 위해 Sun Grid Engine 시스템에서 인증을 수신합니다. 이 초기 발표 단계 후에 통신은 암호화된 양식으로 투명하게 계속됩니다. 세션은 특정 기간 동안만 유효하며, 그 이후에는 세션을 다시 알려야 합니다.

필요한 추가 설정

Sun Grid Engine 시스템의 인증 보안 프로토콜 향상 버전을 설정하는데 필요한 단계는 표준 설정과 매우 유사합니다. 일반적으로 30 페이지, “설치 계획 방법”, 30 페이지, “배포 매체를 읽는 방법”, 31 페이지, “마스터 호스트 설치 방법”, 32 페이지, “실행 호스트 설치 방법” 및 33 페이지, “관리 및 제출 호스트 설치 방법” 절에 있는 지침을 따릅니다.

그러나 다음의 추가 업무가 필요합니다.

- 마스터 호스트에서 인증 기관(CA) 시스템 키 및 인증의 생성
이것은 `-csp` 플래그와 함께 설치 스크립트를 호출하여 수행됩니다.
- 실행 및 제출 호스트에 시스템 키 및 인증의 배포
안전한 방법으로 이 업무를 수행하는 것은 시스템 관리자의 업무입니다. 즉 키는 `ssh`를 통하는 것처럼 보안된 방식으로 실행 호스트 및 제출 호스트에 전송되어야 합니다.
- 사용자 키 및 인증의 생성
이것은 마스터 설치 후에 시스템 관리자에 의해 자동으로 수행될 수 있습니다.
- 시스템 관리자에 의한 새 사용자의 입장 허가

▼ CSP 기반 시스템 설치 및 설정 방법

1. 다음을 예외로 하여 22 페이지, “전체 설치 개요”, 24 페이지, “설치 계획”, 31 페이지, “마스터 호스트 설치 방법”, 32 페이지, “실행 호스트 설치 방법” 및 33 페이지, “관리 및 제출 호스트 설치 방법” 절에서 개략적으로 설명한 것처럼 Sun Grid Engine 시스템을 설치합니다. 여러 설치 스크립트를 호출할 때 추가 플래그 `-csp`를 사용하십시오.

예를 들어 마스터 호스트 설치를 위한 기본 설치 지침이 `./install_qmaster`를 입력하여 스크립트를 호출하도록 지시할 때 `-csp` 플래그를 추가하여 해당 지침을 수정합니다. 그러므로 CSP 보안 시스템을 설치하려면 다음을 입력하여 마스터 호스트 설치 절차를 변경합니다.

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

2. 설치 스크립트의 프롬프트에 응답합니다.

CSP 인증 및 키를 생성하려면 다음 정보를 제공해야 합니다.

- 두 문자의 국가 코드(예를 들어, 미국의 경우 US)
- 시도
- 위치(도시명)
- 조직
- 조직 단위
- CA 이메일 주소

설치가 진행함에 따라서 인증 기관이 작성됩니다. Sun Grid Engine 특정 CA가 마스터 호스트에 작성됩니다. 보안 관련 정보를 포함하는 디렉토리는 다음과 같습니다.

- `$SGE_ROOT/{기본|$SGE_CELL}/common/sgeCA`에는 공개적으로 액세스 가능한 CA 및 데몬 인증이 저장됩니다.
- `/var/sgeCA/{sge 서비스| 포트$COMM_PORT}/{기본|$SGE_CELL}/private`에는 대응하는 개인용 키가 저장됩니다.
- `/var/sgeCA/{sge 서비스| 포트$COMM_PORT}/{기본|$SGE_CELL}/userkeys/$USER`에는 사용자 키 및 인증이 저장됩니다.

이 프로세스 동안 스크립트 출력이 코드 예 2-1의 예와 비슷하게 나타납니다.

코드 예 2-1

CSP 설치 스크립트 — 디렉토리 생성

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

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

디렉토리를 설정한 후 CA 특정 인증 및 개인용 키가 생성됩니다. Sun Grid Engine 시스템은 특수 파일 또는 사용 가능한 경우 `/dev/random`의 모조 무작위 데이터를 사용하여 모조 무작위 숫자 생성기(PRNG)를 생성합니다.(무작위 숫자에 관한 자세한 정보는 <http://www.openssl.org/support/faq.html> 및 <http://www.cosy.sbg.ac.at/~andi> 참조)

CA 하부구조의 설치 후에 관리 사용자, 의사 데몬 사용자 및 사용자 root에 대해 CA가 응용 프로그램 인증, 사용자 인증 및 개인용 키를 작성 및 서명합니다. 스크립트(그 출력은 코드 예 2-2의 예와 비슷함)가 먼저 사이트 정보를 조회합니다.

코드 예 2-2 CSP 설치 스크립트 — 정보 콜렉션

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

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

We will ask for

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

Hit Return to continue >>

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

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

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

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

사용자가 제공한 정보가 맞는지 확인한 후 설치 프로그램이 CA 하부구조 설정으로 시작하여 CA 인증 및 개인용 키 생성을 계속합니다. 스크립트 출력은 코드 예 2-3의 예와 비슷합니다.

코드 예 2-3 CSP 설치 스크립트 — CA 하부구조 생성

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

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

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

CA 하부구조의 설치 후에 CA가 응용 프로그램과 모조 데몬 사용자 및 root 사용자를 위한 사용자 인증 및 개인용 키를 작성하고 서명합니다. 스크립트 출력은 표시되는 것과 비슷합니다(다음 페이지로 계속됨). 예에 있는 행의 일부는 해당 페이지에서 각 단일 행에 맞도록 약어화됨을 유의하십시오. 약어화된 부분은 (...)로 표시됩니다.

코드 예 2-4 CSP 설치 스크립트 — 인증 및 개인용 키 작성

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

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

1513428 semi-random bytes loaded
Using configuration from /tmp/sge_ca14364.tmp
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/var/sgeCA/port6789/default/private/key.pem'
-----

Using configuration from /tmp/sge_ca14364.tmp
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName         :PRINTABLE:'Regensburg'
```

```

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

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

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

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

```

The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Bavaria'
localityName         :PRINTABLE:'Regensburg'
organizationName     :PRINTABLE:'Myorg'
organizationalUnitName:PRINTABLE:'Mydept'
uniqueIdentifier     :PRINTABLE:'root'
commonName           :PRINTABLE:'SGE install user'
emailAddress         :IA5STRING:'none'
Certificate is to be certified until Mar  5 13:50:59 2003 GMT (365 days)

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

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

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

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

```

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

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

마스터 호스트 sge_qmaster의 보안 관련 설정이 완료된 후에 스크립트가 코드 예 2-5의 예와 비슷한 나머지 설치 절차를 계속하도록 프롬프트합니다.

```
SGE startup script
-----

Your system wide SGE startup script is installed as:

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

Hit Return to continue >>
```

3. 다음 중 하나를 수행합니다.

- a. 공유 파일 시스템이 실행 데몬이 액세스할 수 있는 장소에 CSP 보안 정보를 보유하기에 충분히 안전하지 *않다*고 믿는 경우 4 단계로 갑니다.

- b. 공유 파일 시스템이 충분히 안전하다고 믿는 경우 32 페이지, “실행 호스트 설치 방법” 절에서 개략적으로 설명하는 것처럼 기본 설치 절차를 계속합니다.

실행 호스트 설치를 위해 “./inst“ 스크립트를 호출할 때 -csp 플래그를 추가해야 합니다.

모든 나머지 설치 단계를 완료한 후 43 페이지, “사용자를 위한 인증 및 개인용 키 생성 방법” 절에 있는 지침으로 가십시오.

4. (선택적) 공유 파일 시스템이 실행 데몬도 액세스할 수 있는 장소에 CSP 보안 정보를 보유하기에 충분히 안전하지 않은 경우 데몬의 개인용 키와 무작위 파일이 들어 있는 디렉토리를 실행 호스트로 이전해야 합니다.

- a. 마스터 호스트의 root로서 다음 명령을 입력하여 개인용 키를 실행 호스트로서 설정한 시스템에 복사할 준비를 하십시오.

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

- b. 각 실행 호스트의 root로서 다음 명령을 입력하여 파일을 복사합니다.

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

- c. 다음 명령을 입력하여 파일 권한을 검증합니다.

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

출력은 코드 예 2-6의 예와 비슷하게 보일 것입니다.

코드 예 2-6 파일 권한 검증

```
/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. 다음 명령을 입력하여 Sun Grid Engine 설치를 계속합니다.

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

e. 설치 스크립트를 시작하려면 `-csp` 플래그를 추가해야 하는 것을 기억하고 32 페이지, “실행 호스트 설치 방법” 절의 3 단계로 시작하는 설치 지침의 나머지 부분을 따르십시오.

모든 나머지 설치 단계를 완료한 후 43 페이지, “사용자를 위한 인증 및 개인용 키 생성 방법” 절에 있는 지침으로 가십시오.

▼ 사용자를 위한 인증 및 개인용 키 생성 방법

CSP 보안 시스템을 사용하려면 사용자가 사용자에 특정한 인증 및 개인용 키에 액세스해야 합니다. 이것을 수행하는 가장 편리한 방법은 사용자를 식별하는 텍스트 파일을 작성하는 것입니다.

1. 사용자를 식별하는 텍스트 파일을 작성하고 저장합니다.

다음 예에서 표시된 `myusers.txt` 파일의 형식을 사용하십시오. (파일의 필드는 *UNIX 사용자이름:Gecos 필드:이메일주소*입니다.)

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

2. 마스터 호스트의 root로서 다음 명령을 입력합니다.

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

3. 다음 명령을 입력하여 확인합니다.

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

이 디렉토리 목록은 다음과 유사한 출력을 생성해야 합니다.

```
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. 다음 명령을 입력하여 보안 관련 정보를 사용자의 \$HOME/.sge 디렉토리에 설치하기 위해 파일(예의 myusers.txt)에 나열했음을 각 사용자에게 알립니다.

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

사용자들이 다음 확인을 보아야 합니다(예의 사용자 eddy).

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

모든 설치의 경우에 대응하는 COMMD_PORT 번호에 대한 서브디렉토리가 설치됩니다. myusers.txt 파일을 바탕으로 하는 다음 예는 출력 앞에 있는 명령을 발행한 결과입니다.

```
% 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
```

▼ 인증 점검 방법

- 사용자가 수행하려는 사항에 따라서 다음 명령의 하나 이상을 입력합니다.

인증 표시

-in 및 ~/.sge 구성요소 사이에 하나의 공백을 갖고 다음을 한 문자열로 입력하십시오(명령이 이 설명서에서 한 행에 맞추기에는 너무 깁니다).

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

발행자 점검

-in 및 ~/.sge 구성요소 사이에 하나의 공백을 갖고 다음을 한 문자열로 입력하십시오(명령이 이 설명서에서 한 행에 맞추기에는 너무 깁니다).

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

주제 점검

-in 및 ~/.sge 구성요소 사이에 하나의 공백을 갖고 다음을 한 문자열로 입력하십시오(명령이 이 설명서에서 한 행에 맞추기에는 너무 깁니다).

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

인증의 이메일 표시

-in 및 ~/.sge 구성요소 사이에 하나의 공백을 갖고 다음을 한 문자열로 입력하십시오(명령이 이 설명서에서 한 행에 맞추기에는 너무 깁니다).

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

유효성 표시

-in 및 ~/.sge 구성요소 사이에 하나의 공백을 갖고 다음을 한 문자열로 입력하십시오(명령이 이 설명서에서 한 행에 맞추기에는 너무 길니다).

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

지문 표시

-in 및 ~/.sge 구성요소 사이에 하나의 공백을 갖고 다음을 한 문자열로 입력하십시오(명령이 이 설명서에서 한 행에 맞추기에는 너무 길니다).

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

▼ 설치 검증 방법

마스터 호스트에서

1. 마스터 호스트에 로그인합니다.
2. 실행 중인 운영 체제에 따라서 다음 명령 중 하나를 실행합니다.
 - a. BSD 기반 UNIX 시스템에서는 다음 명령을 입력합니다.

```
% ps -ax
```

- b. UNIX 시스템 5 기반 운영 체제(Solaris 운영 환경 같은)를 실행하는 시스템에서는 다음 명령을 입력합니다.

```
% ps -ef
```

3. 다음 예와 유사한 sge 문자열이 있는지 출력을 조사합니다.

BSD 기반 UNIX 시스템에서는 다음과 유사한 출력을 보게 됩니다.

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

UNIX 시스템 5 시스템에서는 다음과 유사한 출력을 보게 됩니다.

```
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
```

적당한 문자열을 *보지 못하는* 경우 마스터 호스트에서 필요한 하나 이상의 Sun Grid Engine 데몬이 이 시스템에서 실행하지 않고 있습니다(실제로 마스터 호스트에 있는지 여부와 관계없이 <sgе-루트>/<셀>/common/act_qmaster 파일을 조사할 수 있습니다). 다음 단계로 가십시오.

4. (선택적)수동으로 데몬을 재시작합니다.

계속하는 방법에 대한 지침은 137 페이지, “데몬 및 호스트 정보” 절을 참조하십시오.

실행 호스트에서

1. Sun Grid Engine 실행 호스트 설치 절차를 실행한 실행 호스트를 조사합니다.
2. 사용자 시스템에 대한 적당한 ps 명령을 판별하기 위해 마스터 호스트 절차의 2 단계를 참조하고 해당 명령을 입력합니다.
3. sge 출력에서 문자열을 찾습니다.

BSD 기반 UNIX 시스템에서는 다음과 유사한 출력을 보게 됩니다.

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

Solaris 운영 환경 같은 UNIX 시스템 5 기반 시스템의 경우에는 다음과 유사한 출력을 보게 됩니다.

```
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
```

유사한 출력을 보지 못하는 경우 실행 호스트에서 필요한 하나 이상의 데몬이 실행하지 않고 있습니다. 다음 단계로 가십시오.

4. (선택적)수동으로 데몬을 재시작합니다.

계속하는 방법에 대한 지침은 137 페이지, “데몬 및 호스트 정보” 절을 참조하십시오.

명령 시도

필요한 데몬 모두가 마스터 및 실행 호스트에서 실행하는 경우 Sun Grid Engine 시스템이 동작해야 합니다. 시험 명령을 발행하여 점검하십시오.

1. 마스터 호스트 또는 다른 관리 호스트에 로그인합니다.

반드시 표준 검색 경로에 Sun Grid Engine 이진을 설치한 경로를 포함시키십시오.

2. 명령줄에서 다음 명령을 입력합니다.

```
% qconf -sconf
```

이 qconf 명령은 현재 글로벌 클러스터 구성을 표시합니다(151 페이지, “기본 클러스터 구성” 절을 참조하십시오). 이 명령이 실패하는 경우 아마도 사용자의 SGE_ROOT 환경 변수가 잘못 설정되거나 qconf가 sge_qmaster와 연관된 sge_commd에 접속하는데 실패했을 것입니다. 다음 단계로 가십시오.

3. 스크립트 파일 <sge 루트>/<셸>/common/settings.csh 또는

<sge 루트>/<셸>/common/settings.sh가 환경 변수 COMMD_PORT를 설정하는지 여부를 확인합니다.

설정하는 경우 위의 명령을 다시 시도하기 전에 환경 변수 COMMD_PORT가 해당 특정 값으로 설정되는지 확인하십시오. COMMD_PORT 변수가 설정 파일에서 사용되지 않는 경우 명령을 실행한 시스템의 서비스 데이터베이스(예: /etc/services 또는 NIS 서비스 맵)가 sge_commd 항목을 제공해야 합니다. 이 경우에 해당하지 않는 경우 시스템의 서비스 데이터베이스에 항목을 추가하고 항목에 Sun Grid Engine 마스터 호스트에서 구성되는 것과 동일한 값을 제공한 후 다음 단계로 가십시오.

4. qconf 명령을 재시도합니다.

작업 제출 준비

Sun Grid Engine 시스템에 일괄처리 스크립트 제출을 시작하기 전에 사이트의 표준 및 사용자의 개인 셸 자원 파일(.cshrc, .profile 또는 .kshrc)이 stty 같은 명령을 포함하는지 확인하십시오(일괄처리 작업은 기본적으로 터미널 연결을 갖지 않으며, 따라서 stty에 대한 호출은 오류를 발생시킵니다).

1. 마스터 호스트에 로그인합니다.
2. 다음 명령을 입력합니다.

```
% rsh 실행호스트 date
```

실행호스트는 사용자가 사용하려는 이미 설치된 실행 호스트 중 하나를 의미합니다(로그인 또는 홈 디렉토리가 호스트에 따라 다른 경우 모든 실행 호스트에서 확인해야 합니다). rsh 명령은 마스터 호스트에서 로컬로 실행되는 date 명령과 아주 유사한 출력을 제공해야 합니다. 오류 메시지를 포함하는 추가 행이 있는 경우 일괄처리 작업을 성공적으로 실행할 수 있으려면 오류의 원인을 제거해야 합니다.

모든 명령 해석기의 경우 stty 같은 명령을 실행하기 전에 실제 터미널 연결을 점검할 수 있습니다. 다음은 이것을 수행하는 Bourne/Korn 셸 예입니다.

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

C 셸 구문도 매우 유사합니다.

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

3. <sgе 루트>/examples/jobs 디렉토리에 들어 있는 샘플 스크립트 중 하나를 제출합니다.

다음 명령을 입력합니다.

```
% qsub 스크립트경로
```

4. Sun Grid Engine qstat 명령을 사용하여 작업의 작동을 모니터합니다.

일괄처리 작업 제출 및 모니터링에 관한 자세한 정보는 73 페이지, “일괄처리 작업 제출”을 참조하십시오.

5. 작업이 실행을 완료한 후 경로 재지정된 stdout/stderr 파일인 <스크립트이름>.e<작업 ID> 및 <스크립트이름>.o<작업ID>가 있는지 홈 디렉토리를 확인합니다. <작업ID>는 각 작업에 할당된 연속적이며 고유한 정수 번호입니다.

문제점이 있는 경우 11장, “문제 해결책”을 참조하십시오.

III 부 Sun Grid Engine 5.3 소프트웨어 사용

주로 사용자, 즉 시스템 관리자(4부, 133 페이지, “관리” 참조)의 직무도 수행하지 않는 사람을 위한 것으로서 *Sun Grid Engine 5.3 관리 및 사용 설명서* 중 이 부분은 세 장으로 구성됩니다.

■ **제 3장** – 53 페이지, “Sun Grid Engine 5.3 프로그램 사용”

이 장은 몇몇 Sun Grid Engine 5.3 기본을 소개하고 다양한 자원을 나열하는 방법에 대한 지침을 포함합니다.

■ **제 4장** – 67 페이지, “작업 제출”

이 장은 Sun Grid Engine 5.3 시스템을 사용하여 작업을 제출하기 위한 완전한 지침을 제공하고, 사용자에게 프로세스를 숙지시키는 “연습” 작업 제출로 시작합니다.

■ **제 5장** – 107 페이지, “작업 검사점, 모니터링 및 제어”

이 장은 작업 제어의 개념을 설명하며 다양한 작업 제어 업무를 수행하기 위한 지침이 들어 있습니다.

제 3부의 각 장에는 Sun Grid Engine 5.3 시스템을 사용하여 무수한 업무 수행에 관한 백그라운드 정보와 상세한 지침이 모두 들어 있습니다.

Sun Grid Engine 5.3 프로그램 사용

이 장은 소프트웨어 사용을 시작할 때 도움이 되는 몇 가지 기본 Sun Grid Engine 5.3 개념과 전문용어를 소개합니다. 포괄적인 용어집을 포함하여 제품에 관한 전체 백그라운드 정보에 대해서는 1장, 1 페이지, “Sun Grid Engine 5.3 소개”를 참조하십시오.

이 장에는 또한 다음 업무를 수행하기 위한 지침이 들어 있습니다.

- 55 페이지, “QMON 브라우저를 시작하는 방법”
- 56 페이지, “대기열 목록을 표시하는 방법”
- 56 페이지, “대기열 등록정보를 표시하는 방법”
- 59 페이지, “마스터 호스트의 이름을 찾는 방법”
- 59 페이지, “실행 호스트의 목록을 표시하는 방법”
- 60 페이지, “관리 호스트의 목록을 표시하는 방법”
- 60 페이지, “제출 호스트의 목록을 표시하는 방법”
- 61 페이지, “요청 가능 속성 목록을 표시하는 방법”

Sun Grid Engine 사용자 유형 및 조작

사용자 유형은 Sun Grid Engine에서 4 범주로 나뉘어집니다.

- **관리자** - 관리자는 Sun Grid Engine을 조작할 모든 능력을 갖습니다. 기본적으로 대기열을 호스트하는 모든 시스템의 슈퍼유저는 관리자 특권을 갖습니다.
- **운영자** - 운영자는 관리자와 동일한 많은 명령을 수행할 수 있으며, 예를 들어 대기열을 추가, 삭제 또는 수정하여 구성을 변경하는 것은 예외입니다.
- **소유자** - 대기열 소유자는 소유한 대기열이나 그 안에 있는 작업을 일시중단 또는 활성화할 수 있지만 더 이상의 관리 권한은 없습니다.
- **사용자** - 사용자는 64 페이지, “사용자 액세스 사용 권한”에서 설명하는 것처럼 특정 액세스 권한을 갖지만 클러스터 또는 대기열 관리 능력은 없습니다.

표 3-1은 여러 사용자 범주에 사용할 수 있는 Sun Grid Engine 5.3 명령 기능을 보여줍니다.

표 3-1 사용자 범주 및 연관된 명령 기능

명령	관리자	운영자	소유자	사용자
qacct	전체	전체	작업만 소유	작업만 소유
qalter	전체	전체	작업만 소유	작업만 소유
qconf	전체	시스템 설정 수정 없음	구성 및 액세스 권한만 표시	구성 및 액세스 권한만 표시
qdel	전체	전체	작업만 소유	작업만 소유
qhold	전체	전체	작업만 소유	작업만 소유
qhost	전체	전체	전체	전체
qlogin	전체	전체	전체	전체
qmod	전체	전체	작업 소유 및 소유된 대기열만	작업만 소유
qmon	전체	시스템 설정 수정 없음	구성 변경 없음	구성 변경 없음
qrexec	전체	전체	전체	전체
qselect	전체	전체	전체	전체
qsh	전체	전체	전체	전체
qstat	전체	전체	전체	전체
qsub	전체	전체	전체	전체

대기열 및 대기열 등록정보

사이트에서 Sun Grid Engine 시스템을 최적으로 이용할 수 있으려면 Sun Grid Engine 시스템에 대해 구성되는 대기열 구조 및 대기열의 등록정보에 익숙해져야 합니다.

QMON 브라우저

Sun Grid Engine은 그래픽 사용자 인터페이스(GUI) 명령 도구인 QMON 브라우저가 특징입니다. QMON 브라우저는 작업 제출, 작업 제어 및 중요한 정보 수집을 포함하여 무수한 Sun Grid Engine 기능을 제공합니다.

▼ QMON 브라우저를 시작하는 방법

- 명령줄에서 다음 명령을 입력하십시오.

```
% qmon
```

메시지 창이 표시된 후 다음과 유사한 QMON 기본 제어 패널이 나타납니다(아이콘의 의미를 식별하려면 그림 1-2를 참조하십시오).



그림 3-1 QMON Main Control 메뉴

이 설명서의 많은 지침이 QMON 브라우저를 사용하기 위해 호출합니다. 기능의 설명적인 아이콘 단추의 이름이 아이콘 위로 마우스 포인터를 이동할 때 화면에 나타납니다.

(QMON 브라우저를 사용자 정의하는 방법에 대한 지침은 9 페이지, “QMON 사용자 정의”를 참조하십시오.)

Queue Control QMON 대화 상자

126 페이지, “QMON을 사용하여 대기열을 제어하는 방법” 절에 표시되고 설명되는 QMON Queue Control 대화 상자는 설치된 대기열 및 그의 현재 상태에 대한 빠른 개요를 제공합니다.

▼ 대기열 목록을 표시하는 방법

- 다음 명령을 입력합니다.

```
% qconf -sql
```

▼ 대기열 등록정보를 표시하는 방법

QMON 또는 명령줄을 사용하여 대기열 등록정보를 표시할 수 있습니다.

QMON 브라우저 사용

1. 기본 QMON 메뉴에서 **Browser** 아이콘을 누릅니다.
2. **Queue** 단추를 누릅니다.
3. **Queue Control** 대화 상자에서 마우스 포인터를 적당한 대기열에 대한 아이콘 위로 이동합니다.

그림 3-2 가 표시되는 대기열 등록 정보의 부분적인 예입니다.

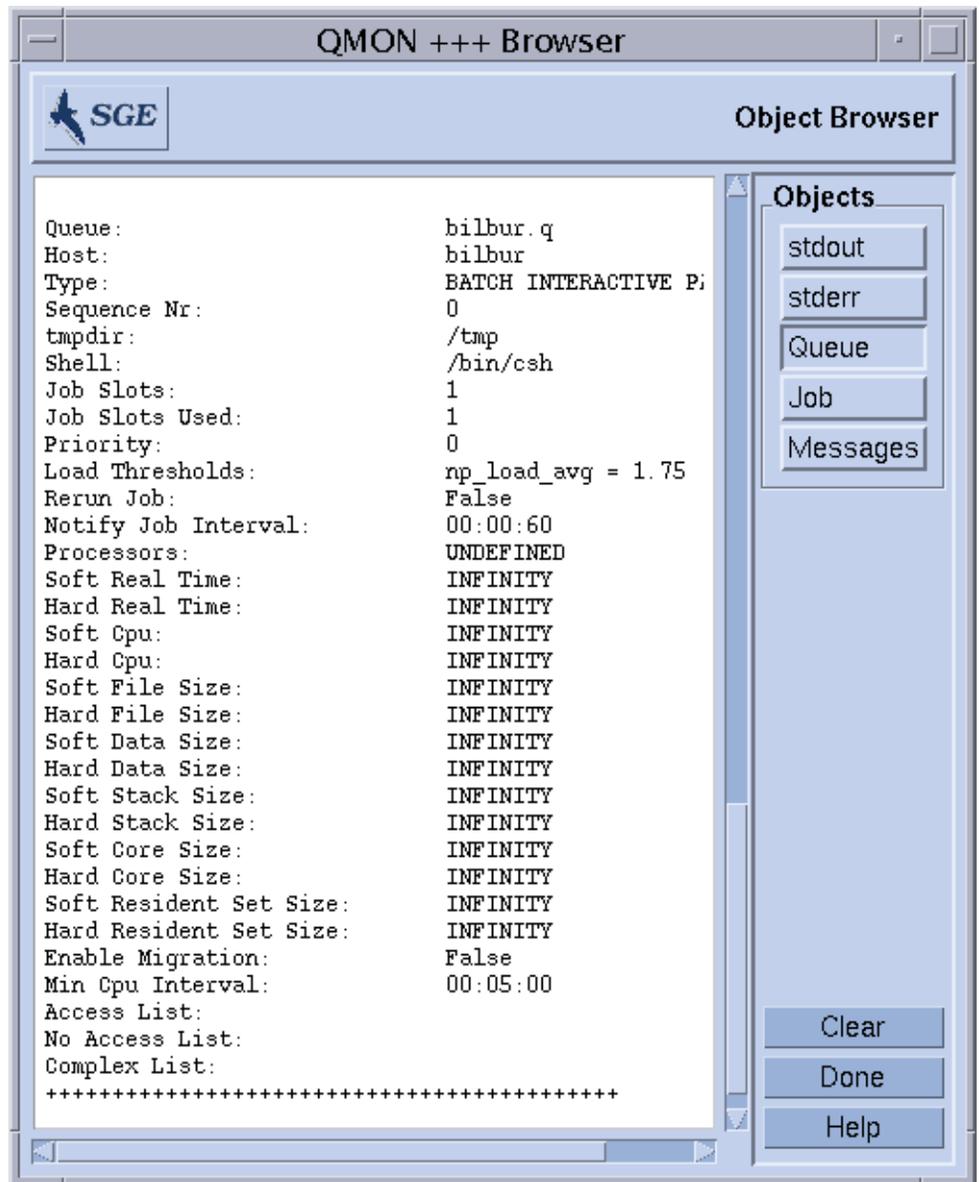


그림 3-2 대기열 등록정보의 QMON 브라우저 표시

명령줄에서

- 다음 명령을 입력합니다.

```
% qconf -sq 대기열이름
```

그림 3-2에 표시된 것과 유사한 정보가 표시됩니다.

대기열 등록 정보 해석

queue_conf 매뉴얼 페이지 및 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 queue_conf 절에서 각 대기열 등록정보의 상세한 설명을 찾을 수 있습니다.

다음은 가장 중요한 일부 매개변수의 목록입니다.

- 대기열이름 - 요청된 대기열 이름.
- 호스트이름 - 대기열의 호스트.
- 프로세서 - 대기열이 액세스하는 다중 프로세서 시스템의 프로세서.
- 대기열유형 - 이 대기열에서 실행하도록 허용되는 작업의 유형. 현재 이것은 일괄처리, 대화식, 검사점, 병렬 또는 그들의 모든 조합 또는 대안으로서 전송입니다.
- 슬롯 - 해당 대기열에서 동시에 실행될 수 있는 작업 수.
- 소유자목록 - 65 페이지, “관리자, 운영자 및 소유자” 절에서 설명한 것과 같은 대기열의 소유자.
- 사용자목록 - 이 매개변수 아래에 나열되는 사용자 액세스 목록(64 페이지, “사용자 액세스 사용 권한” 참조)에 있는 사용자 또는 그룹 ID가 대기열에 액세스할 수 있습니다.
- xuser_lists - 이 매개변수 아래에 나열되는 사용자 액세스 목록(64 페이지, “사용자 액세스 사용 권한” 참조)에 있는 사용자 또는 그룹 ID가 대기열에 액세스할 수 없습니다.
- 콤플렉스목록 - 이 매개변수 아래에 나열되는 콤플렉스가 대기열과 연관되며 이들 콤플렉스에 들어 있는 속성이 대기열에 대한 요청 가능한 속성 세트에 기여합니다 (60 페이지, “요청 가능한 속성” 참조).
- 콤플렉스값 - 특정 콤플렉스 속성에 대해 이 대기열에 제공되는 용량을 할당합니다 (60 페이지, “요청 가능한 속성” 참조).

호스트 기능

QMON Main 메뉴의 Host Configuration 단추를 누르면 Sun Grid Engine 클러스터의 호스트와 연관된 기능의 개요가 표시됩니다. 그러나 Sun Grid Engine 관리자 특권이 없으면 표시된 구성에 어떤 변경사항도 적용할 수 없습니다.

호스트 구성 대화 상자는 137 페이지, “데몬 및 호스트 정보” 절에서 설명됩니다. 다음 절은 명령줄에서 이런 종류의 정보를 검색하기 위한 명령을 제공합니다.

▼ 마스터 호스트의 이름을 찾는 방법

마스터 호스트가 언제든지 현재 마스터 호스트와 새도우 마스터 호스트 중 하나 사이에서 이주할 수 있기 때문에 마스터 호스트의 위치는 사용자에게 투명해야 합니다.

- **텍스트 편집기를 사용하여 <sgc-루트>/<셀>/common/act_qmaster 파일을 여십시오.**
현재 마스터 호스트의 이름이 이 파일에 있습니다.

▼ 실행 호스트의 목록을 표시하는 방법

클러스터에서 실행 호스트로서 구성되는 호스트 목록을 표시하려면 다음 명령을 사용하십시오.

```
% qconf -sel
% qconf -se 호스트이름
% qhost
```

첫 번째 명령은 현재 실행 호스트로 구성되는 모든 호스트의 이름 목록을 표시합니다. 두 번째 명령은 지정된 실행 호스트에 대한 상세한 정보를 표시합니다. 세 번째 명령은 실행 호스트에 관한 상태 및 부하 정보를 표시합니다. qconf를 통해 표시되는 정보에 대한 자세한 내용은 host_conf 매뉴얼 페이지를 참조하고 그의 출력 및 추가 옵션에 대한 자세한 내용은 qhost 매뉴얼 페이지를 참조하십시오.

▼ 관리 호스트의 목록을 표시하는 방법

관리 권한을 갖는 호스트 목록은 다음 명령으로 표시할 수 있습니다.

```
% qconf -sh
```

▼ 제출 호스트의 목록을 표시하는 방법

제출 호스트 목록은 다음 명령으로 표시할 수 있습니다.

```
% qconf -ss
```

요청 가능한 속성

Sun Grid Engine 작업을 요청할 때 작업의 요구사항 프로파일을 지정할 수 있습니다. 사용자가 작업이 성공적으로 실행하기 위해 필요한 호스트 또는 대기열의 속성 또는 특성을 지정할 수 있습니다. Sun Grid Engine은 이들 작업 요구사항을 Sun Grid Engine 클러스터의 호스트 및 대기열 구성에 맵핑하며 따라서 작업에 적합한 호스트를 찾습니다.

작업 요구사항을 지정하는 데 사용할 수 있는 속성은 Sun Grid Engine 클러스터(예: 네트워크 공유 디스크에서 필요한 공간), 호스트(예: 운영 체제 아키텍처) 또는 대기열(허용되는 CPU 시간)과 관련되거나 속성이 일부 호스트에만 설치된 소프트웨어의 가용성 같은 사이트 정책에서 파생됩니다.

사용 가능한 속성에는 대기열 등록정보 목록(55 페이지, “대기열 및 대기열 등록정보” 참조), 글로벌 및 호스트 관련 속성의 목록(177 페이지, “복합형 유형” 참조) 및 관리자가 정의하는 속성이 포함됩니다. 그러나 편의를 위해 Sun Grid Engine 관리자는 일반적으로 모든 사용 가능한 속성의 서브세트만을 요청 가능하도록 정의할 것을 선택합니다.

현재 요청 가능한 속성이 QMON Submit 대화 상자(작업 제출 방법에 대한 자세한 정보는 73 페이지, “일괄처리 작업 제출” 절을 참조)에 대한 Requested Resources 부속 대화 상자(그림 3-3 참조)에 표시됩니다. 해당 속성은 Available Resources 선택 목록에 나열됩니다.

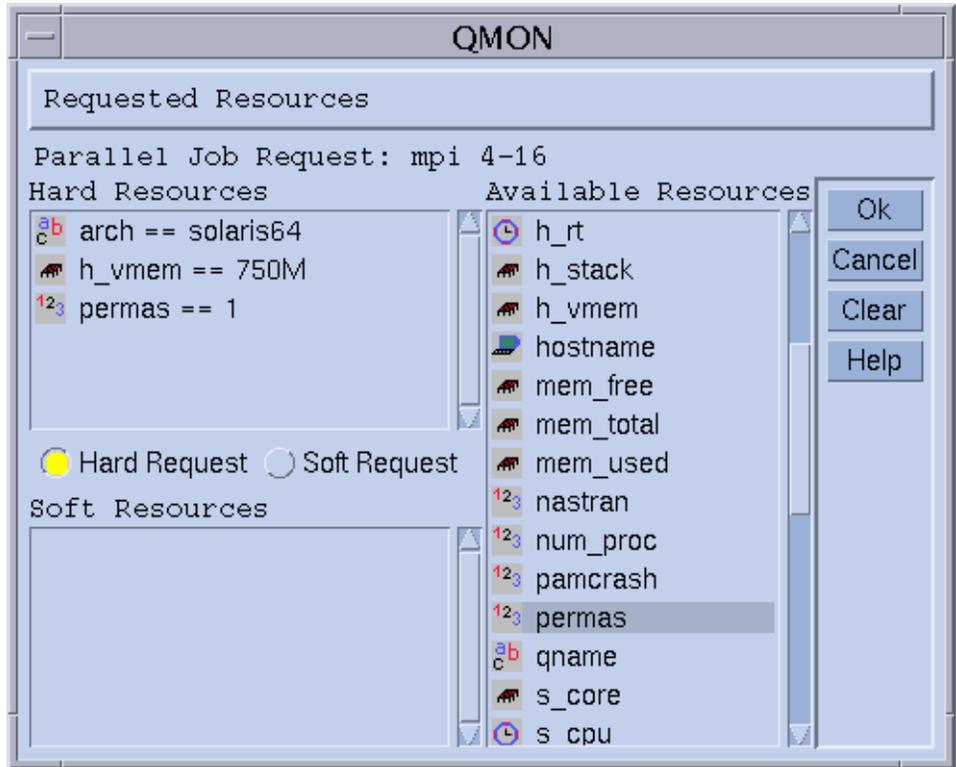


그림 3-3 Requested Resources 대화 상자

▼ 요청 가능 속성 목록을 표시하는 방법

1. 명령줄에서 다음 명령을 입력하여 구성된 콤플렉스의 목록을 표시합니다.

```
% qconf -scl
```

콤플렉스에는 속성 세트에 대한 정의가 들어 있습니다. 다음의 세 가지 표준 콤플렉스가 있습니다.

- global- (선택적) 클러스터 글로벌 속성의 경우
- host - 호스트 특정 속성의 경우
- queue - 대기열 등록정보 속성의 경우

위의 명령의 결과로서 인쇄되는 모든 추가 콤플렉스 이름은 관리자 정의 콤플렉스를 의미합니다(콤플렉스에 대한 자세한 정보는 8장, 175 페이지, “콤플렉스 개념” 또는 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*에 있는 콤플렉스 형식 설명을 참조하십시오).

2. 다음 명령을 입력하여 특수 콤플렉스의 속성을 표시합니다.

```
% qconf -sc 콤플렉스이름 [...]
```

대기열 콤플렉스에 대한 출력은 예를 들어 표 3-2에 표시된 것처럼 보일 수 있습니다.

표 3-2 표시되는 queue 콤플렉스 속성

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

`name` 열은 기본적으로 `qconf -sq` 명령에 의해 표시되는 첫 번째 열과 동일합니다. 대기열 속성은 대부분의 Sun Grid Engine 대기열 등록정보를 포함합니다. `shortcut` 열에는 첫 번째 열의 전체 이름에 대해 관리자가 정의할 수 있는 약어가 들어 있습니다. 사용자가 `qsub` 명령의 요청 옵션에 전체 이름 또는 단축키를 제공할 수 있습니다.

`requestable` 열은 대응하는 항목을 `qsub`에서 사용할 수 있는지 여부를 표시합니다. 따라서 관리자는 예를 들어 단순히 항목 `qname` 및/또는 `qhostname`을 요청이 불가능하도록 설정하여 클러스터의 사용자가 작업에 대한 특정 시스템/대기열을 직접 요청할 수 있도록 허용하지 않을 수 있습니다. 이것을 수행하는 것은 적당한 사용자 요청이 일반적으로 복수 대기열에 의해 충족될 수 있음을 내포하며, 이는 Sun Grid Engine의 부하 균형 기능을 강제합니다.

`relop` 열은 대기열이 사용자 요청을 만족시킬 수 있는지 여부를 연산하기 위해 사용되는 관계 조작을 정의합니다. 실행되는 비교는 다음과 같습니다.

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

비교의 결과가 거짓인 경우 사용자의 작업이 고려된 대기열에서 실행될 수 없습니다. 예를 들어 대기열 `q1`이 100초의 소프트 `cpu` 시간 한계(사용자 프로세스 한계의 설명은 `queue_conf` 및 `setrlimit` 매뉴얼 페이지 참조)를 갖는 반면 대기열 `q2`는 1000 초 소프트 `cpu` 시간 제한을 제공하도록 구성되게 합니다.

`consumables` 및 `default` 열은 관리자가 소위 소비 가능 자원을 선언하는 데 의미가 있습니다(185 페이지, “소비 가능 자원” 절을 참조하십시오). 사용자는 다른 모든 속성과 마찬가지로 소비 가능을 요청합니다. 그러나 자원에 대한 Sun Grid Engine 내부 회계 기입은 다릅니다.

사용자가 다음 요청을 제출한다고 가정하십시오.

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

`s_cpu=0:5:0` 요청(구문에 대한 자세한 내용은 `qsub` 매뉴얼 페이지 참조)은 최소한 5분의 소프트 한계 `cpu` 시간을 부여하는 대기열을 요청합니다. 그러므로 최소한 5분의 소프트 CPU 런타임 한계를 제공하는 대기열만이 해당 작업을 실행하도록 적절하게 설정됩니다.

참고 - Sun Grid Engine은 둘 이상의 대기열이 작업을 실행할 수 있는 경우에만 스케줄링 프로세스의 작업부하 정보를 고려합니다.

사용자 액세스 사용 권한

대기열 및 다른 Sun Grid Engine 기능(예: 병렬 환경 인터페이스, 99 페이지, “병렬 작업” 참조)에 대한 액세스가 Sun Grid Engine 관리자에 의해 특정 사용자 또는 사용자 그룹에 대해 제한될 수 있습니다.

참고 – Sun Grid Engine은 자동으로 클러스터 관리에 의해 구성되는 액세스 제한을 고려합니다. 다음 절은 개인 액세스 권한을 조회하려는 경우에만 중요합니다.

액세스 권한 제한을 위해서는 관리자가 소위 액세스 목록(또는 줄여서 *ACL*)을 작성하고 유지보수합니다. ACL에는 임의 사용자 및 UNIX 그룹 이름이 들어 있습니다. 그런 다음 ACL이 대기열 또는 병렬 환경 인터페이스 구성의 *액세스허용*- 또는 *액세스거부*-목록에 추가됩니다(각각 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서* 5절의 `queue_conf` 또는 `sgc_pe` 참조).

액세스허용목록에 나열되는 ACL에 속하는 사용자는 대기열 또는 병렬 환경 인터페이스에 액세스하는 사용 권한을 갖습니다. 액세스거부목록의 ACL의 구성원인 사용자는 관련 자원에 액세스할 수 없습니다.

QMON 기본 메뉴의 사용자 구성 아이콘 단추를 통해 열리는 사용자세트 구성 대화 상자를 사용하여 사용자세트 구성 대화 상자를 통해 액세스할 수 있는 ACL을 조회할 수 있습니다. 자세한 내용은 9장, 203 페이지, “사용자 액세스 및 정책 관리”를 참조하십시오.

명령줄에서 다음 명령에 의해 현재 구성된 ACL의 목록을 얻을 수 있습니다.

```
% qconf -sul
```

다음 명령으로 하나 또는 복수 액세스 목록의 항목이 인쇄됩니다.

```
% qconf -su ACL 이름 [...]
```

ACL은 사용자 계정 이름과 접두어인 “@” 기호에 의해 식별되는 UNIX 그룹 이름을 갖는 UNIX 그룹 이름으로 구성됩니다. 이 방법으로 사용자 계정이 속하는 ACL을 판별할 수 있습니다.

참고 – `newgrp` 명령을 사용하여 1차 UNIX 그룹을 전환할 권한이 있는 경우에는 액세스 권한이 변경될 수 있습니다.

이제 사용자가 액세스할 수 있거나 사용자의 액세스가 거부되는 대기열 또는 병렬 환경 인터페이스를 확인할 수 있습니다. 55 페이지, “대기열 및 대기열 등록정보” 및 234 페이지, “QMON을 사용하여 PE를 구성하는 방법”에 설명된 대로 대기열 또는 병렬 환경 인터페이스 구성을 조회하십시오. 액세스허용목록은 `user_lists`로 명명됩니다. 액세스 거부목록은 `xuser_lists`라는 이름을 갖습니다. 사용자 계정 또는 1차 UNIX 그룹이 액세스허용목록과 연관되는 경우 관련 자원에 액세스할 수 있습니다. 액세스거부목록과 연관되는 경우 해당 대기열 또는 병렬 환경 인터페이스에 액세스할 수 없습니다. 두 목록이 모두 비어 있는 경우 유효한 계정을 갖는 모든 사용자가 관련 자원에 액세스할 수 있습니다.

관리자, 운영자 및 소유자

다음을 사용하여 Sun Grid Engine 관리자의 목록을 얻을 수 있습니다.

```
% qconf -sm
```

또한 다음을 사용하여 운영자 목록을 얻을 수 있습니다.

```
% qconf -so
```

참고 – Sun Grid Engine 관리 호스트의 슈퍼유저는 기본적으로 관리자로 간주됩니다.

특정 대기열의 소유자인 사용자는 55 페이지, “대기열 및 대기열 등록정보” 절에서 설명하는 것처럼 대기열 구성 데이터베이스에 포함됩니다. 이 데이터베이스는 다음을 실행하여 검색할 수 있습니다.

```
% qconf -sq 대기열이름
```

관련 대기열 구성 항목을 `owners`라고 부릅니다.

작업 제출

이 장은 처리를 위해 작업을 제출하기 위한 Sun Grid Engine 5.3 사용에 대한 백그라운드 정보 및 지침을 제공합니다. 이 장은 간단한 작업을 실행하는 예로 시작한 후 더 복잡한 작업 실행에 대한 지침으로 계속합니다.

다음 업무를 수행하기 위한 지침이 이 장에 들어 있습니다.

- 68 페이지, “명령줄에서 간단한 작업을 실행하는 방법”
- 70 페이지, “그래픽 사용자 인터페이스 QMON에서 작업을 제출하는 방법”
- 90 페이지, “명령줄에서 작업을 제출하는 방법”
- 93 페이지, “명령줄에서 어레이 작업을 제출하는 방법”
- 93 페이지, “QMON을 사용하여 어레이 작업을 제출하는 방법”
- 95 페이지, “QMON을 사용하여 대화식 작업을 제출하는 방법”
- 98 페이지, “qsh를 사용하여 대화식 작업을 제출하는 방법”
- 98 페이지, “qlogin을 사용하여 대화식 작업을 제출하는 방법”

간단한 작업 실행

이 절의 정보와 지침을 사용하여 Sun Grid Engine 5.3 작업 제출과 관련된 기본 절차에 익숙해지십시오.

참고 - 특권이 없는 계정에서 Sun Grid Engine 프로그램을 설치한 경우 작업을 실행할 수 있으려면 해당 특정 사용자로서 로그인해야 합니다(자세한 내용은 24 페이지, “필수 조건 작업”을 참조하십시오).

▼ 명령줄에서 간단한 작업을 실행하는 방법

임의의 Sun Grid Engine 명령을 실행하기 전에 먼저 실행 가능한 검색 경로와 기타 환경 조건을 적절히 설정해야 합니다.

1. 사용자의 명령 해석기에 따라서 다음 명령 중 하나를 입력합니다.

a. csh 또는 tcsh를 명령 해석기로 사용 중인 경우,

```
% source sge 루트디렉토리/default/common/settings.csh
```

*sge 루트디렉토리*는 설치 절차의 시작시에 선택된 Sun Grid Engine 루트 디렉토리의 위치를 지정합니다.

b. sh, ksh 또는 bash를 명령 해석기로 사용 중인 경우,

```
# . sge 루트디렉토리/default/common/settings.sh
```

참고 - 나중에 시작할 모든 대화식 세션에 대한 적절한 Sun Grid Engine 설정을 보장하기 위해 위의 명령을 `.login`, `.cshrc` 또는 `.profile` 파일(어느 것인든지 적당한 파일)에 추가할 수 있습니다.

2. Sun Grid Engine 클러스터에 다음의 간단한 작업 스크립트를 제출합니다.

Sun Grid Engine 루트 디렉토리의 `examples/jobs/simple.sh` 파일에서 다음 작업을 찾을 수 있습니다.

```
#!/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
```

다음 명령을 입력하십시오. 이것은 `simple.sh`가 위의 스크립트가 저장되는 스크립트 파일의 이름이며 파일이 현재 작업 디렉토리에 위치한다고 가정합니다.

```
% qsub simple.sh
```

`qsub` 명령은 다음과 같이 성공적인 작업 제출을 확인해야 합니다.

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

3. 다음 명령을 입력하여 사용자 작업에 대한 상태 정보를 검색합니다.

```
% qstat
```

현재 Sun Grid Engine 시스템에 알려진 모든 작업에 대한 정보 및 각 작업에 대한 소위 *작업 ID* (제출 확인에 포함되는 고유한 번호), 작업 스크립트의 이름, 작업 소유자, 상태 정보(`r`은 실행 중을 의미함), 제출 또는 시작 시간 및 마지막으로 작업이 실행하는 대기열의 이름이 들어 있는 상태 보고서를 수신해야 합니다.

`qstat` 명령이 출력을 생성하지 않는 경우 시스템에 실제로 알려진 작업이 없습니다. 예를 들어 작업이 이미 종료했을 수 있습니다. 작업의 `stdout` 및 `stderr` 경로 재지정 파일을 점검하여 종료된 작업의 출력을 제어할 수 있습니다. 기본적으로 이들 파일은 작업을 실행한 호스트에서 작업 소유자의 홈 디렉토리에 생성됩니다. 파일의 이름은 작업 스크립트 파일 이름, `stdout` 파일에 대한 “`o`” 및 `stderr` 파일에 대한 “`e`” 뒤에 추가 점 기호, 그리고 마지막으로 고유 작업 ID로 구성됩니다. 그래서 해당 작업이 새로 설치된 Sun Grid Engine, Enterprise 버전 시스템에서 처음 실행되었다면 작업의 `stdout` 및 `stderr` 파일은 각각 `simple.sh.o1` 및 `simple.sh.e1` 이름 하에서 찾을 수 있습니다.

▼ 그래픽 사용자 인터페이스 QMON에서 작업을 제출하는 방법

Sun Grid Engine 작업 제출 및 제어와 Sun Grid Engine 시스템의 개요를 얻는 보다 편리한 방법이 그래픽 사용자 인터페이스 QMON입니다. 다른 기능 중에서 QMON은 작업 제출 메뉴와 작업 제출 및 모니터링 업무를 위한 작업 제어 대화 상자를 제공합니다.

명령줄 프롬프트에서, 다음 명령을 입력하십시오.

```
% qmon
```

시작 중에 메시지 창이 표시된 후 QMON Main 메뉴가 나타납니다.

1. Job Control 단추를 누른 후 Submit Jobs 단추를 누릅니다.



그림 4-1 QMON Main 메뉴

Job Submission 및 Job Control 대화 상자가 나타납니다(그림 4-2 및 그림 4-3을 각각 참조하십시오). 단추 위로 마우스 포인터를 이동하면 단추 이름(예: Job Control)이 표시됩니다.

먼저, 여기를 눌러서
스크립트 파일을 선택하십시오 ...

... 그리고 **Submit** 를 눌러서
작업을 제출하십시오 .

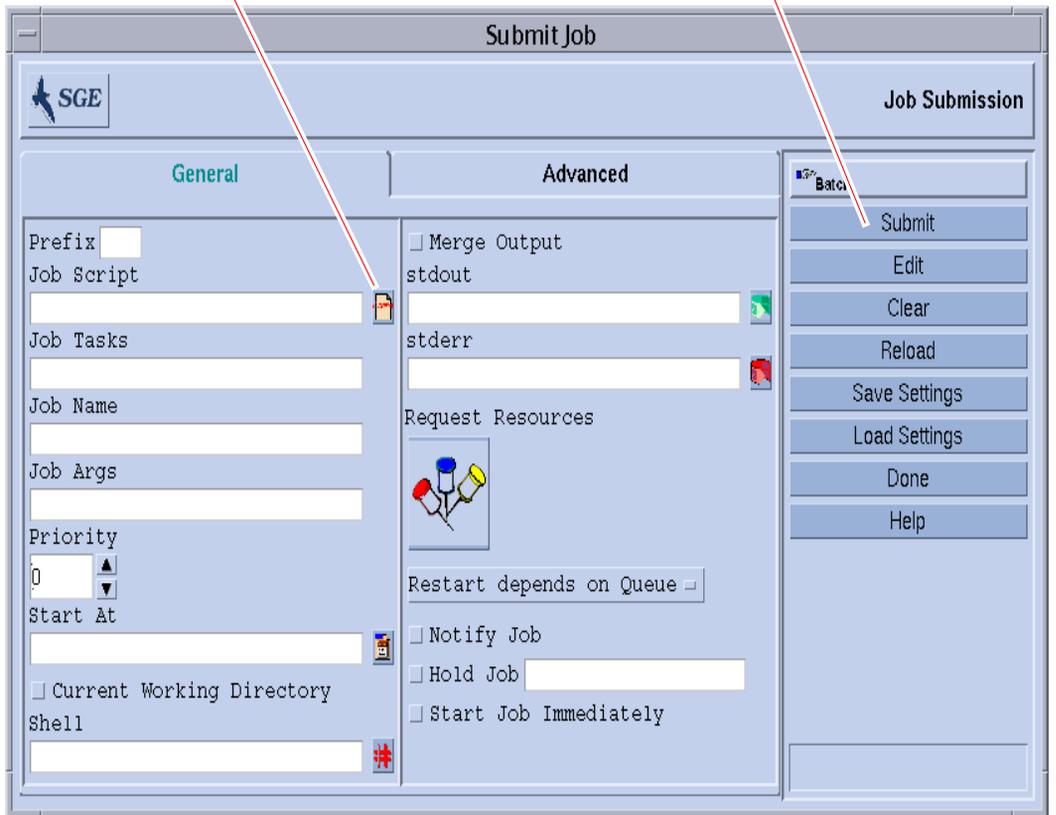


그림 4-2 QMON Job Submission 대화 상자



그림 4-3 QMON Job Control 대화 상자

2. **Job Submission** 메뉴에서 **Job Script** 파일 선택 아이콘을 눌러서 파일 선택 상자를 엽니다.
3. 적당한 파일 이름을 눌러서 스크립트 파일(예: 명령줄 예에서 *simple.sh* 파일)을 선택합니다.
4. **Job Submission** 메뉴의 맨 아래에 있는 **Submit** 단추를 누릅니다.

몇 초 후 Job Control 패널에서 작업을 모니터할 수 있어야 합니다. 먼저 Pending Jobs에 작업이 표시되고 작업이 시작되면 빨리 Running Jobs로 이동합니다.

일괄처리 작업 제출

다음 절은 Sun Grid Engine 5.3 프로그램을 통해 더 복잡한 작업을 제출하는 방법을 설명합니다.

셸 스크립트 정보

일괄처리 작업이라고도 부르는 셸 스크립트는 원칙적으로 파일에 모은 일련의 명령줄 명령어입니다. 스크립트 파일은 `chmod` 명령에 의해 실행 가능하게 됩니다. 스크립트가 호출되는 경우 적당한 명령 해석기가 시작되고(예: `csh`, `tcsh`, `sh` 또는 `ksh`) 각 명령어가 스크립트를 실행하는 사용자에게 의해 수동으로 입력된 것으로 해석됩니다. 셸 스크립트 안에서 임의의 명령, 응용 프로그램 및 다른 셸 스크립트를 호출할 수 있습니다.

이름(`csh`, `tcsh`, `sh`, `ksh`, ...)이 작업을 실행하는 특정 호스트 및 대기열에 대해 적용되는 Sun Grid Engine 구성의 `login_shells`의 값 목록에 포함되는지 여부에 따라서 적당한 명령 해석기가 `login-shell`로서 호출되거나 호출되지 않습니다.

참고 - Sun Grid Engine 구성은 클러스터에 구성되는 여러 호스트 및 대기열에 대해 다를 수 있습니다. `qconf` 명령의 `-sconf` 및 `-sq` 옵션을 통해 유효 구성을 표시할 수 있습니다(자세한 정보는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서* 참조).

명령 해석기가 `login-shell`로서 호출되는 경우 작업의 환경이 사용자가 로그인하고 스크립트를 실행한 경우와 정확히 동일합니다. 예를 들어 `csh`를 사용할 때 `.login` 및 `.cshrc`가 시스템 기본 시작 자원 파일(예: `/etc/login`과 같은 파일) 외에 실행되는 반면, `csh`가 `login-shell`로 호출되지 않는 경우 `.cshrc`만이 실행됩니다. `login-shell`로서 호출되는 것과 그렇지 않은 경우의 차이에 대한 설명은 사용자가 선택한 명령 해석기의 매뉴얼 페이지를 참조하십시오.

스크립트 파일의 예

코드 예 4-1은 간단한 셸 스크립트의 예로서, 먼저 Fortran77 소스로부터 응용 프로그램 flow를 컴파일한 후 실행합니다.

```
#!/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
```

코드 예 4-1 간단한 셸 스크립트

로컬 시스템 사용자 설명서가 셸 스크립트 빌드 및 사용자 정의에 대한 상세한 정보를 제공할 것입니다(또한 sh, ksh, csh 또는 tcsh 매뉴얼 페이지를 볼 수도 있습니다). 다음 절에서는 Sun Grid Engine을 위한 일괄처리 스크립트를 준비하기 위해 고려되어야 하는 특수성을 강조합니다.

일반적으로 터미널 연결이 필요하지 않고(자동으로 경로 재지정되는 표준 오류 및 출력 장치의 경우는 예외) 및 대화식 사용자 간섭이 필요하지 않은 한, 명령 프롬프트에서 수동으로 실행할 수 있는 모든 셸 스크립트를 Sun Grid Engine에 제출할 수 있습니다. 그러므로 코드 예 4-1은 Sun Grid Engine에 제출될 준비가 되었으며 원하는 조치를 수행할 것입니다.

QMON을 사용한 확장 및 고급 작업 제출

작업 제출의 더 복잡한 양식(확장 또는 고급)을 시도하기 전에 프로세스에 관한 몇 가지 중요한 백그라운드 정보를 이해하는 것이 유용합니다. 다음 절이 해당 정보를 제공합니다.

확장 예

Job Submission 대화 상자(그림 4-2 참조)의 표준 양식은 확장 작업에 대한 다음 매개변수를 구성하는 방법을 제공합니다.

- 스크립트 내장 Sun Grid Engine 제출 옵션에 사용되는 접두어 문자열(자세한 정보는 87 페이지, “활성 Sun Grid Engine 주석” 절 참조)

- 사용될 작업 스크립트
 - 연관된 파일 단추를 누르면 파일 선택 상자가 열립니다(그림 4-3 참조).
- 어레이 작업 제출을 위한 업무 ID 범위(92 페이지, “어레이 작업” 참조)
- 작업의 이름(작업 스크립트가 선택된 후 기본값이 설정됩니다)
- 작업 스크립트에 대한 인수
- 작업의 초기 우선순위 값
 - 관리자나 운영자의 승인이 없으면 사용자는 초기 우선순위 값을 더 낮추기만 할 수 있습니다.
- 작업이 실행에 적합한 것으로 간주되어야 하는 시간
 - 연관된 파일 단추를 누르면 올바르게 형식화된 시간을 입력할 수 있는 도움말 대화 상자가 열립니다(그림 4-4 참조).
- 작업이 현재 작업 디렉토리에서 작업이 실행될 것인지를 표시하는 플래그(제출 및 잠재적인 실행 호스트 사이의 동일한 디렉토리 계층의 경우에만)
- 작업 스크립트를 실행하는 데 사용될 명령 해석기(86 페이지, “명령 해석기가 선택되는 방법” 참조)
 - 연관된 단추를 누르면 작업의 명령 해석기 스펙을 입력할 수 있는 도움말 대화 상자가 열립니다(그림 4-5 참조).
- 작업의 표준 출력 및 표준 오류 출력이 표준 출력 스트림에 함께 병합될 것인지를 표시하는 플래그
- 사용될 표준 출력 경로 재지정(86 페이지, “출력 경로 재지정” 참조)
 - 아무 것도 지정하지 않으면 기본값이 사용됩니다. 연관된 파일 단추를 누르면 출력 경로 재지정 대안을 입력할 수 있는 헬퍼 대화 상자가 사용 가능하게 됩니다(86 페이지, “출력 경로 재지정” 참조).
- 사용할 표준 오류 출력 경로 재지정-표준 출력 경로 재지정과 매우 유사함
- 작업의 자원 요구사항
 - 작업에 대한 자원 수요를 정의하려면 대응하는 아이콘 단추를 누르십시오. 자원이 작업에 요청된 경우 아이콘 단추의 색상이 변경됩니다.
- 작업이 시스템 붕괴 또는 유사항 이벤트에 의해 중단된 후 재시작될 수 있는지 및 재시작 작동이 대기열에 의존하는지 또는 작업에 의해 요구되는지 여부를 정의하는 선택 목록 단추
- 작업이 일시중단 또는 취소될 예정인 경우 작업이 SIGUSR1 또는 SIGUSR2 신호에 의해 통지될 것인지 여부를 표시하는 플래그
- 사용자 보류 또는 작업 중속성이 작업에 할당될 것임을 표시하는 플래그
 - 작업이 임의 유형의 보류가 할당되는 동안은 실행에 적합하지 않습니다(보류에 관한 자세한 정보는 112 페이지, “Sun Grid Engine 작업 모니터링 및 제어” 절 참조). 보류 플래그에 첨부된 입력 필드를 사용하여 보류를 어레이 작업의 특정 범위의 업무에만 제한할 수 있습니다(92 페이지, “어레이 작업” 참조).

- 작업이 가능한 경우 즉시 시작되거나 거부되도록 강제하는 플래그
이 플래그가 선택되는 경우 작업이 대기열에 들어가지 않습니다.

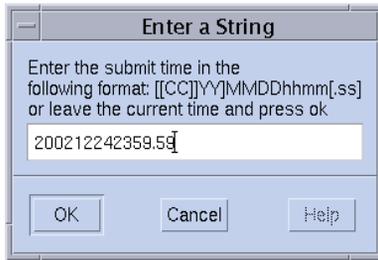


그림 4-4 시작시 입력 상자



그림 4-5 셸 선택 상자



그림 4-6 출력 경로 재지정 상자

Job Submission 화면의 오른쪽에 있는 단추로 다양한 조치를 시작할 수 있습니다.

- **Submit** – 대화 상자에 지정된 대로 작업을 제출합니다.
- **Edit** – vi 또는 \$EDITOR 환경 변수에 정의된 편집기를 사용하여 X-terminal에서 선택된 스크립트 파일을 편집합니다.
- **Clear** – 지정된 모든 자원 요청을 포함하여 Job Submission 대화 상자의 모든 설정을 지웁니다.
- **Reload** – 지정된 스크립트 파일을 재로드하고, 모든 스크립트 내장 옵션을 구문 분석하고(87 페이지, “활성 Sun Grid Engine 주석” 절 참조), 기본 설정을 구문 분석하고(91 페이지, “기본 요청” 절 참조) 이들 설정에 대한 중간 수동 변경사항을 삭제합니다. 이 조치는 이전 스크립트 파일의 후속 스펙을 갖는 Clear 조치와 동일합니다. 옵션은 스크립트 파일이 이미 선택된 경우에만 효과를 보입니다.
- **Save Settings** – 현재 설정을 파일에 저장합니다. 파일을 선택할 파일 선택 상자가 열립니다. 저장된 파일은 나중에 명시적으로 로드되거나(아래 참조) 기본 요청으로 사용될 수 있습니다(91 페이지, “기본 요청” 절 참조).
- **Load Settings** – 이전에 Save Settings 단추(위 참조)로 저장된 설정을 로드합니다. 로드된 설정이 현재 설정을 겹쳐줍니다.
- **Done** – Job Submission 대화 상자를 닫습니다.
- **Help** – 대화 상자에 특정한 도움말을 표시합니다.

그림 4-7 는 대부분의 매개변수가 설정된 Job Submission 대화 상자를 보여줍니다.

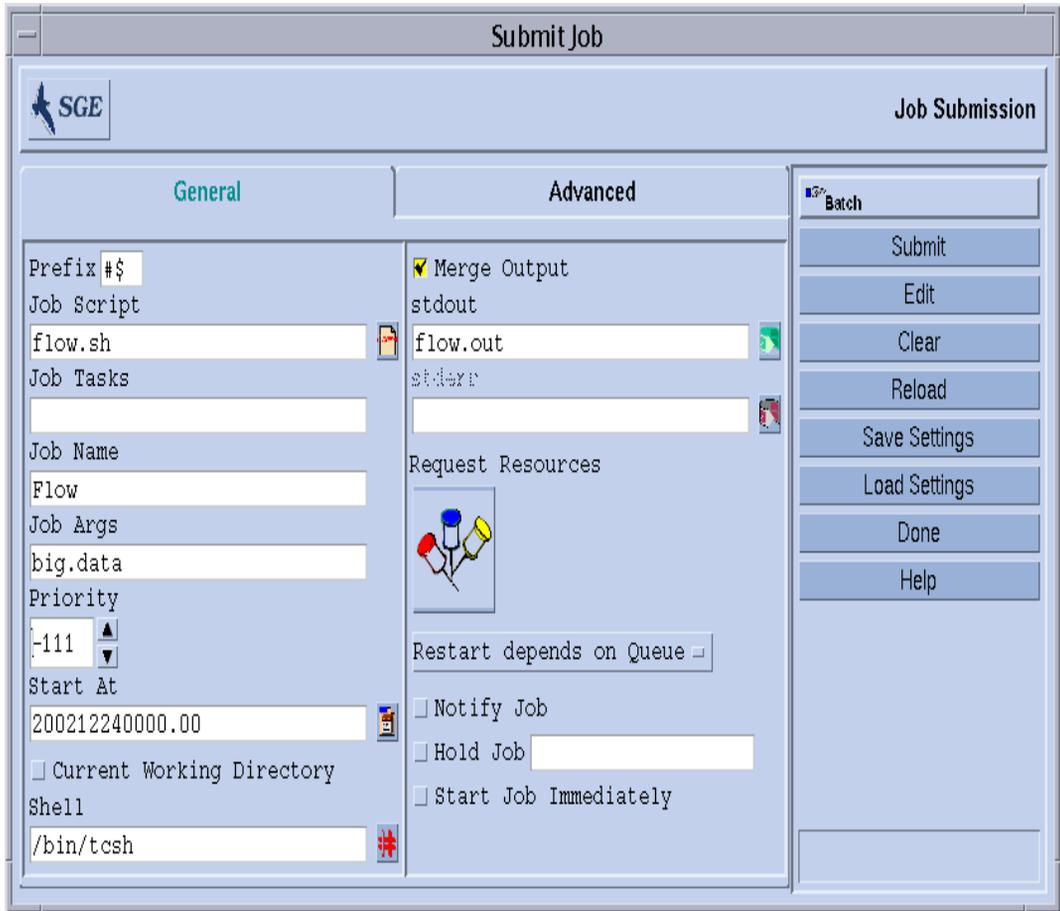


그림 4-7 확장된 작업 제출 보기

보기에서 구성된 작업은 스크립트 파일 flow.sh를 갖는데, 이 파일은 QMON의 작업 디렉토리에 상주해야 합니다. 작업은 Flow라고 부르며 스크립트 파일은 단 하나의 인수인 big.data를 갖습니다. 작업은 우선순위 -111로 시작되며 2002년 12월 24일 자정 이후에 실행될 수 있습니다. 작업은 제출 작업 디렉토리에서 실행되며 tcsh 명령 해석기를 사용합니다. 마지막으로 표준 출력 및 표준 오류 출력은 flow.out 파일에 병합되며, 이 파일도 현재 작업 디렉토리에 작성됩니다.

고급 보기

Advanced 제출 화면은 다음 추가 매개변수의 정의를 허용합니다.

- 사용할 병렬 환경 인터페이스 및 필요한 프로세스 범위(99 페이지, “병렬 작업” 절 참조)
- 작업이 실행되기 전에 작업에 대해 설정될 환경 변수 세트
연관된 아이콘 단추를 누르면 내보내길 환경 변수의 정의를 위한 헬퍼 대화 상자가 사용 가능하게 됩니다(그림 4-8 참조). 환경 변수는 QMON의 런타임 환경에서 취할 수 있거나 임의의 환경 변수를 정의할 수 있습니다.
- 문맥이라고 부르는 이름/값 쌍의 목록(그림 4-9 참조), 이것은 Sun Grid Engine 클러스터 내의 어디에서나 액세스할 수 있는 작업 관련 정보를 저장하고 통신하는 데 사용될 수 있습니다.
문맥 변수는 qsub, qrsh, qsh, qlogin, 또는 qalter에 대한 -ac/-dc/-sc 옵션을 통해 명령줄에서 수정할 수 있으며 qstat -j를 통해 검색할 수 있습니다.
- 검사점이 바람직하고 적합한 작업의 경우에 사용될 검사점 환경(107 페이지, “작업 검사점 정보” 절을 참조하십시오)
- 작업과 연관될 계정 문자열
계정 문자열은 작업에 대해 보존된 회계 레코드에 추가되며 나중에 회계 분석에 사용될 수 있습니다.
- 작업에 대한 일관성 점검 모드를 판별하는 Verify 플래그
작업 요청의 일관성을 점검하려면 Sun Grid Engine은 비어 있고 로드되지 않은 클러스터를 가정하고 작업이 실행될 수 있는 최소한 하나의 대기열을 찾습니다. 가능한 점검 모드는 다음과 같습니다.
 - **Skip** – 일관성 점검이 전혀 없습니다.
 - **Warning** – 불일치가 보고되지만 작업은 여전히 승인됩니다(클러스터 구성이 작업의 제출 후에 변경하도록 제안되는 경우에 바람직합니다).
 - **Error** – 오류가 발생하는 경우 불일치가 보고되고 작업이 거부됩니다.
 - **Just verify** – 작업이 제출되지 않지만 클러스터의 각 호스트 및 대기열에 대해 작업의 적합성에 관한 방대한 보고서가 생성됩니다.
- 사용자에게 전자 우편을 통해 통지되는 이벤트
작업의 시작/종료/중단/일시중단 이벤트가 현재 정의되어 있습니다..
- 이들 통지 메일이 전송될 전자 우편 주소의 목록
연관된 단추를 누르면 메일링 목록을 정의하는 헬퍼 대화 상자를 사용할 수 있습니다(그림 4-10 참조).

- 작업 실행에 대한 필수 선택이 되도록 요청되는 대기열 이름의 목록.
하드 대기열 목록 및 소프트 대기열 목록은 나열된 목록 항목 75 페이지, “작업의 자원 요구사항”에서 설명한 것처럼 대응하는 자원 요구사항과 동일하게 취급됩니다.
- 병렬 작업에 대한 *마스터 대기열*로 적합한 대기열 이름의 목록.
병렬 작업은 마스터 대기열에서 시작됩니다. 작업이 병렬 업무를 야기시키는 다른 모든 대기열은 *슬레이브 대기열*이라고 부릅니다.
- 제출될 작업이 시작할 수 있기 전에 성공적으로 종료되어야 하는 작업의 ID 목록
새로 작성되는 작업은 해당 작업의 성공적인 완료에 *따릅니다*.



그림 4-8 작업 환경 정의



그림 4-9 작업 문맥 정의



그림 4-10 메일 주소 지정

그림 4-11에 정의되는 작업은 74 페이지, “확장 예” 절에 있는 작업 정의와 비교할 때 다음의 추가 특성을 갖습니다.

- 작업은 병렬 환경 `mpi`의 사용이 필요합니다. 최소한 4개의 병렬 프로세스가 작성되어야 하며 사용 가능한 경우 최대 16개 프로세스를 이용할 수 있습니다.
- 작업에 대해 두 개의 환경 변수가 설정되고 내보내집니다.
- 두 문맥 변수가 설정됩니다.
- 계정 문자열 `FLOW`가 작업 회계 레코드에 추가될 것입니다.
- 작업이 시스템 붕괴의 경우에 실패하는 경우 재시작됩니다.
- 작업 요청과 클러스터 구성 사이의 불일치가 감지되는 경우 경고가 인쇄됩니다.
- 작업이 시작하고 종료하자마자 메일이 두 이메일 주소의 목록에 전송되어야 합니다.
- 가능하면 작업이 대기열 `big_q`에서 실행되어야 합니다.

그림 4-11는 고급 작업 제출의 예를 보여줍니다.

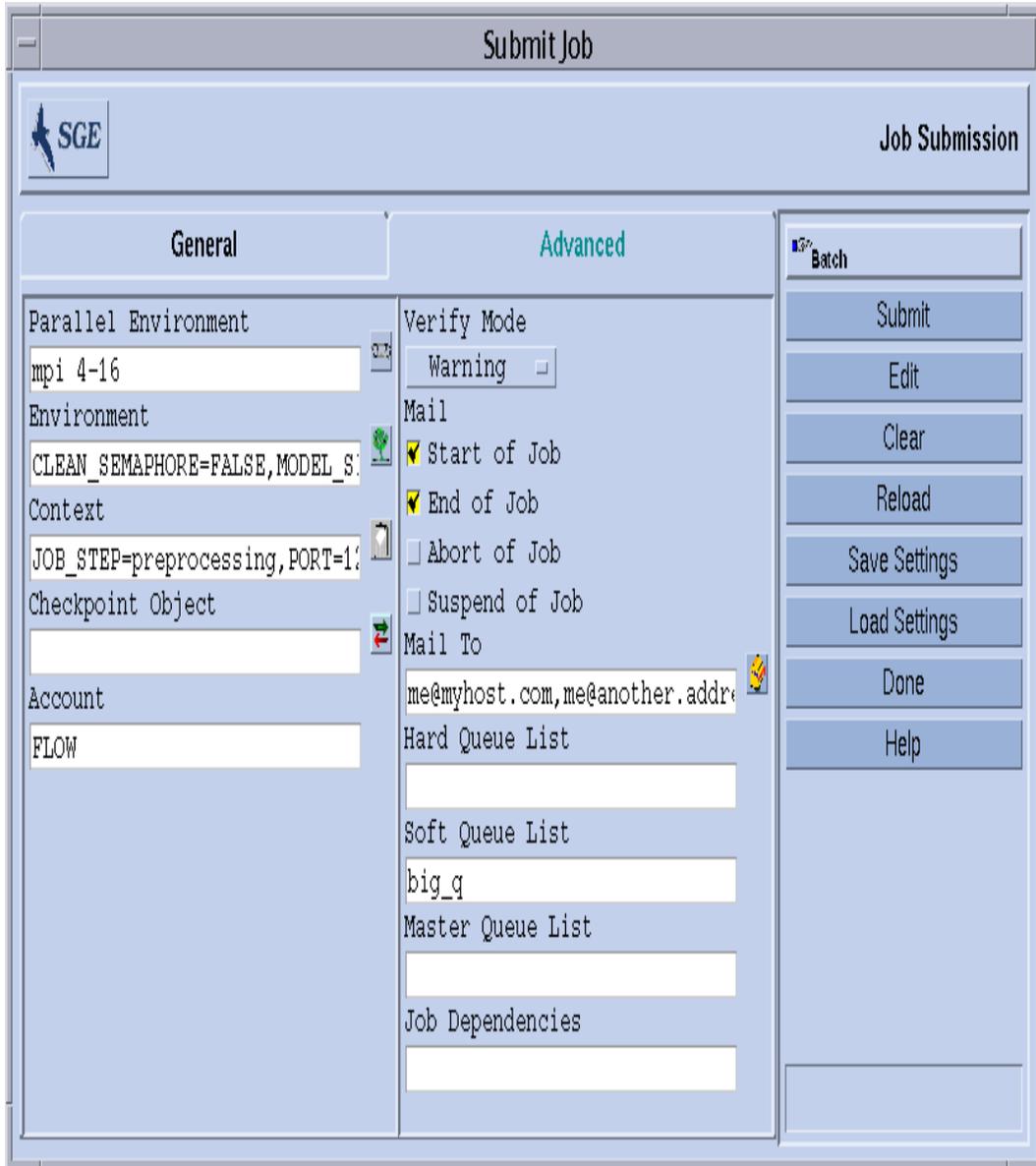


그림 4-11 고급 작업 제출 보기

자원 요구사항 정의

지금까지의 예에서 사용된 제출 옵션은 작업이 실행될 호스트에 대한 어떤 요구사항도 표현하지 않았습니다. Sun Grid Engine은 그런 작업이 어떤 호스트에서나 실행될 수 있다고 가정합니다. 그러나 실제로는 대부분의 작업이 성공적으로 완료할 수 있기 위해서 실행 호스트에서 만족되어야 하는 특정 전제조건을 요구합니다. 그런 전제조건은 충분히 사용 가능한 메모리, 설치되어야 하는 필수 소프트웨어 또는 특정 운영 체제 아키텍처입니다. 또한 클러스터 관리가 대개 클러스터의 시스템 사용에 대해 제한을 부과합니다. 예를 들어 종종 작업이 소비하도록 허용되는 CPU 시간이 제한됩니다.

Sun Grid Engine은 사용자에게 클러스터의 장비 및 그의 이용 정책의 간결한 지식 없이 적당한 호스트를 찾는 방법을 제공합니다. 사용자가 수행해야 하는 모든 것은 사용자 작업의 요구사항을 지정하고 Sun Grid Engine이 적당하고 부하가 적은 호스트를 찾는 작업을 관리하게 하는 것입니다.

자원 요구사항은 60 페이지, “요청 가능한 속성” 절에서 설명한 *요구 가능한 속성*을 통해 지정됩니다. 작업의 요구사항을 지정하는 매우 편리한 방법을 QMON이 제공합니다. 작업 제출 대화 상자의 요청된 자원 단추를 누를 때 열리는 요청된 자원 대화 상자(예를 들어 그림 4-12 참조)는 현재 적합하고 사용 가능한 자원 선택 목록에 있는 속성만을 표시합니다. 속성을 두 번 누르면 속성이 작업의 하드 또는 소프트웨어(아래 참조) 자원 목록에 추가되고(단지 True로 설정되는 BOOLEAN 유형 속성은 제외) 관련 속성에 대한 값 속성을 입력하는 것을 도와주는 헬퍼 대화 상자가 열립니다.

그림 4-12에 표시되는 요청된 자원 대화 상자 보기는 최소한 750 MB의 메모리를 제공하는 사용 가능한 `permas` 사용권을 갖는 `solaris64` 호스트가 필요한 작업에 대한 자원 프로파일을 보여줍니다. 이 스펙을 충족하는 둘 이상의 대기열이 발견되는 경우 정의된 모든 소프트 자원 요구사항이 고려됩니다(이 보기에서는 없습니다). 그러나 하드 및 소프트 요구사항을 둘 다 만족하는 대기열이 없는 경우 하드 요구사항을 제공하는 모든 대기열이 적합한 것으로 간주됩니다.

참고 - 한 작업에 대해 둘 이상의 대기열이 적합한 경우에만 부하 기준이 작업을 시작할 곳을 결정합니다.

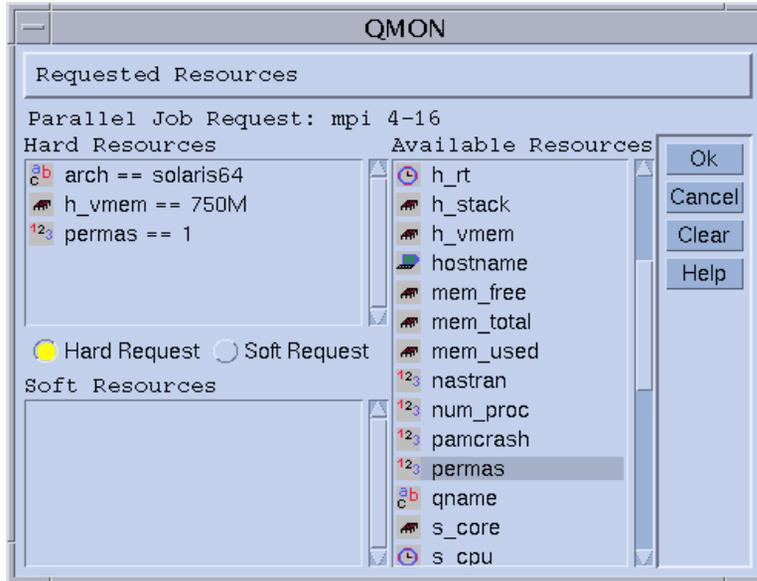


그림 4-12 Requested Resources 대화 상자

참고 - INTEGER 속성 `permas`는 "글로벌" 콤플렉스에 대한관리자 확장을 통해 소개되며, MEMORY 속성 `h_vmem`을 "대기열" 콤플렉스에서 가져오는 반면, STRING 속성 `arch`는 "호스트" 콤플렉스에서 가져옵니다.

`qsub` 명령줄로부터 동일한 자원 요구사항 프로파일을 제출할 수도 있습니다.

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

참고 - 첫 번째 `-l` 옵션 앞에 있는 내재적 `-hard` 스위치가 생략되었습니다.

750MB에 대한 표기법 `750M`은 Sun Grid Engine 수량 구분에 대한 예입니다. 메모리 소비를 요청하는 해당 속성에 대해 소위 승수가 첨부되는 정수 십진수, 부동 소수점, 정수 8진수 및 정수 16진수 중 하나를 지정할 수 있습니다.

- k - 값에 1000을 곱합니다.
- K - 값에 1024를 곱합니다.
- m - 값에 1000의 1000배를 곱합니다.
- M - 값에 1024의 1024배를 곱합니다.

8진법 상수는 선행하는 0(영)과 0-7 범위의 십진수에 의해서만 지정됩니다. 16진 상수를 지정하려면 숫자 앞에 0x를 첨부하고 0-9 범위의 십진수, a-f 및 A-F를 사용해야 합니다. 승수가 첨부되지 않으면 값은 바이트로서 계수하는 것으로 간주됩니다. 부동 소수점 십진수를 사용하는 경우 결과 값은 정수 값으로 잘라집니다.

시간 제한을 부과하는 속성의 경우 시, 분 또는 초 및 모든 조합으로 시간 값을 지정할 수 있습니다. 시, 분 및 초는 콜론으로 구분되는 십진수로 지정됩니다. 3:5:11의 시간은 11111 초로 변환됩니다. 시, 분 또는 초에 대한 지정자가 0인 경우 콜론이 남아 있으면 생략할 수 있습니다. 따라서 값 :5:는 5분으로 해석됩니다. 위의 요청된 자원 대화 상자에서 사용되는 양식은 확장으로, QMON 안에서만 유효합니다.

Sun Grid Engine 시스템이 자원을 할당하는 방법

마지막 절에 표시된 것처럼 Sun Grid Engine 소프트웨어가 자원 요청을 처리하는 방법과 자원을 할당하는 방법을 아는 것이 중요합니다. 다음은 Sun Grid Engine 소프트웨어의 자원 할당 알고리즘의 도식적인 보기를 제공합니다.

1. 모든 기본 요청 파일을 읽고 구문 분석합니다(91 페이지, “기본 요청” 절 참조).
2. 내장 옵션에 대해 스크립트 파일을 처리합니다(87 페이지, “활성 Sun Grid Engine 주석” 절 참조).
3. 작업이 제출될 때 스크립트 파일에서의 위치와 상관없이 모든 스크립트 내장 옵션을 읽습니다.
4. 명령줄에서 모든 요청을 읽고 구문분석합니다.

모든 qsub 요청이 수집되자마자 하드 및 소프트 요청이 별도로 처리됩니다(하드 우선). 요청이 다음의 우선권 순서에 대응하여 평가됩니다.

1. 스크립트/기본 요청 파일의 왼쪽에서 오른쪽으로
2. 스크립트/기본 요청 파일의 맨 위에서 맨 아래로
3. 명령줄의 왼쪽에서 오른쪽으로

다시 말하면 명령줄은 내장 플래그를 대체하는 데 사용할 수 있습니다.

하드로 요청된 자원이 할당됩니다. 요청이 올바르지 않은 경우 제출이 거부됩니다. 제출시에 하나 이상의 요청이 만족될 수 없는 경우(예를 들어, 요청된 대기열이 사용중) 작업은 스톱되고 나중에 다시 예약됩니다. 모든 하드 요청이 만족될 수 있는 경우 요청이 할당되고 작업이 실행될 수 있습니다.

소프트로 요청된 자원이 점검됩니다. 작업은 이들 요청의 일부 또는 전부가 만족될 수 없는 경우에도 실행할 수 있습니다. 복수 대기열(이미 하드 요청을 만족하는)이 소프트 자원 목록의 일부(중첩하거나 서로 다른 부분)를 제공하는 경우 Sun Grid Engine 소프트웨어는 가장 많은 소프트 요청을 제공하는 대기열을 선택합니다.

작업이 시작되고 할당된 자원을 포함합니다.

hostname 또는 date 같은 UNIX 명령을 실행하는 작은 테스트 스크립트 파일로 시험하여 인수 목록 옵션 및 내장 옵션 또는 하드 및 소프트 요청이 서로 영향을 미치는 방법에 대한 약간의 경험을 얻은 것이 유용합니다.

정규 셸 스크립트에 대한 확장

Sun Grid Engine 제어 하에서 실행 중인 경우 스크립트의 작동에 영향을 주는 정규 셸 스크립트에 대한 몇 가지 확장이 있습니다. 다음 절은 이들 확장에 대해 설명합니다.

명령 해석기가 선택되는 방법

작업 스크립트 파일을 처리하는 데 사용될 명령 해석기는 제출시에 지정할 수 있습니다(예를 들어 그림 4-7 참조). 그러나 아무 것도 지정하지 않는 경우 구성 변수 shell_start_mode가 명령 해석기가 선택되는 방법을 판별합니다.

- shell_start_mode가 unix_behavior로 설정되는 경우 스크립트 파일의 첫 번째 행(,#!“ 순서로 시작하는 경우)이 명령 해석기를 판별하기 위해 평가됩니다. 첫 번째 행에 #!“ 순서가 없으면, Bourne 셸 sh이 기본값으로 사용됩니다.
- shell_start_mode의 다른 모든 설정에 대해 작업이 시작되는 대기열에 대해 shell 매개변수로 구성되는 기본 명령 해석기가 사용됩니다(55 페이지, “대기열 및 대기열 등록정보” 절 및 queue_conf 매뉴얼 페이지 참조).

출력 경로 재지정

일괄처리 작업은 터미널 연결을 갖지 않으므로 표준 출력 및 표준 오류 출력은 파일로 경로 재지정되어야 합니다. Sun Grid Engine에서는 사용자가 출력이 경로 재지정되는 파일의 위치를 정의할 수 있지만 아무 것도 지정하지 않으면 기본값을 사용합니다.

파일에 대한 표준 위치는 작업이 실행하는 현재 작업 디렉토리에 있습니다. 기본 표준 출력 파일 이름은 <작업이름>.o<작업ID>이고, 기본 표준 오류 출력은 <작업이름>.e<작업ID>로 경로 재지정됩니다. <작업이름>은 스크립트 파일 이름으로부터 빌드되거나 사용자가 정의할 수 있습니다(예를 들어 qsub 매뉴얼 페이지의 -N 옵션 참조). <작업ID>는 Sun Grid Engine에 의해 작업에 할당되는 고유한 ID입니다.

어레이 작업 업무(92 페이지, “어레이 작업” 절 참조)의 경우 업무 ID가 도트 기호에 의해 구분되는 이들 파일 이름에 추가됩니다. 따라서 결과 표준 경로 재지정 경로는 <작업이름>.o<작업ID>.<업무ID> 및 <작업이름>.e<작업ID>.<업무ID>입니다.

표준 위치가 적합하지 않은 경우 사용자가 그림 4-11 및 그림 4-6에 표시된 것처럼 QMON을 사용하거나 -e 및 -o qsub 옵션을 사용하여 출력 방향을 지정할 수 있습니다. 표준 출력 및 표준 오류 출력이 하나의 파일에 병합될 수 있으며 경로 재지정을 실행 호스트별로 지정할 수 있습니다. 즉, 작업이 실행되는 호스트에 따라서 출력 경로 재지

정 파일의 위치가 달라지게 됩니다. 사용자가 정의하지만 고유한 경로 재지정 파일 경로를 빌드하기 위해 `qsub -e` 및 `-o` 옵션으로 함께 사용할 수 있는 모조 환경 변수를 사용할 수 있습니다. 이들 변수의 목록은 다음과 같습니다.

- `$HOME` - 실행 시스템의 홈 디렉토리
- `$USER` - 작업 소유자의 사용자 ID
- `$JOB_ID` - 현재 작업 ID
- `$JOB_NAME` - 현재 작업 이름(-N 옵션 참조)
- `$HOSTNAME` - 실행 호스트의 이름
- `$TASK_ID` - 어레이 작업 업무 색인 번호

이들 변수는 작업의 런타임 중에 실제 값으로 확장되고 경로 재지정 경로가 해당 값을 사용하여 빌드됩니다.

자세한 내용은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `qsub` 항목을 참조하십시오.

활성 Sun Grid Engine 주석

주요한 “#” 표시가 있는 행은 셸 스크립트에서 주석으로 다루어집니다. 그러나 Sun Grid Engine은 특수한 주석 행을 인식하고 특별한 방법으로 사용합니다. 그런 스크립트의 나머지는 Sun Grid Engine 제출 명령 `qsub`의 명령줄 인수 목록의 일부처럼 취급됩니다. 이들 특수 주석 행 안에 제공되는 `qsub` 옵션도 QMON Job Submission 대화 상자에 의해 해석되고 스크립트 파일이 선택될 때 대응하는 매개변수가 사전 설정됩니다.

기본값당 특수 주석 행은 “#`$`” 접두어 문자열에 의해 식별됩니다. 접두어 문자열은 `qsub -C` 옵션을 사용하여 재정의할 수 있습니다.

설명한 메커니즘을 제출 인수의 스크립트 내장이라고 부릅니다. 다음은 스크립트 내장 명령줄 옵션을 사용하는 스크립트 파일의 예입니다.

```
#!/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
```

코드 예 4-2 스크립트 내장 명령줄 옵션 사용

환경 변수

Sun Grid Engine 작업이 실행될 때 많은 변수가 아래에 나열된 것처럼 작업의 환경으로 사전 설정됩니다.

- ARC – 작업이 실행 중인 노드의 Sun Grid Engine 아키텍처 이름. 이 이름이 `sge_execd` 이진으로 컴파일됩니다.
- `COMMD_PORT` – `sge_commd(8)`가 통신 요청을 청취할 것으로 예상되는 TCP 포트를 지정합니다.
- `SGE_ROOT` – 시작 전에 `sge_execd`에 대해 설정되는 Sun Grid Engine 루트 디렉토리 또는 기본값 `/usr/SGE`
- `SGE_CELL` – 작업이 실행되는 Sun Grid Engine 셀
- `SGE_JOB_SPOOL_DIR` – `sge_shepherd(8)`가 작업 실행 중엔 작업 관련 데이터를 저장하는 데 사용하는 디렉토리

- SGE_O_HOME - 작업이 제출된 호스트에서 작업 소유자의 홈 디렉토리 경로
- SGE_O_HOST - 작업이 제출된 호스트
- SGE_O_LOGNAME - 작업이 제출된 호스트에서 작업 소유자의 로그인 이름
- SGE_O_MAIL - 작업 제출 명령의 문맥에 있는 MAIL 환경 변수의 내용
- SGE_O_PATH - 작업 제출 명령의 문맥에서 PATH 환경 변수의 내용
- SGE_O_SHELL - 작업 제출 명령의 문맥에서 SHELL 환경 변수의 내용
- SGE_O_TZ - 작업 제출 명령의 문맥에서 TZ 환경 변수의 내용
- SGE_O_WORKDIR - 작업 제출 명령의 작업 디렉토리
- SGE_CKPT_ENV - 검사점 작업이 실행되는 검사점 환경(qsub -ckpt 옵션으로 선택되는 대로)을 지정합니다.
- SGE_CKPT_DIR - 검사점 작업의 경우에만 설정됩니다. 체크포인트 인터페이스의 경로 ckpt_dir(checkpoint 매뉴얼 페이지 참조)이 들어 있습니다.
- SGE_STDERR_PATH - 작업의 표준 오류 스트림이 전환되는 파일의 경로 이름. 일반적으로 프로로그, 에필로그, 병렬 환경 시작/중지 또는 검사점 스크립트의 오류 메시지로 출력을 향상시키는 데 사용됩니다.
- SGE_STDOUT_PATH - 작업의 표준 출력 스트림이 전환되는 파일의 경로 이름. 일반적으로 프로로그, 에필로그, 병렬 환경 시작/중지 또는 검사점 스크립트의 오류 메시지로 출력을 향상시키는 데 사용됩니다.
- SGE_TASK_ID - 이 업무에 의해 표시되는 어레이 작업의 업무 ID
- ENVIRONMENT - 항상 BATCH로 설정됩니다. 이 변수는 스크립트가 일괄처리 모드로 실행됨을 표시합니다.
- HOME - passwd 파일에 있는 사용자의 홈 디렉토리 경로
- HOSTNAME - 작업이 실행 중인 노드의 호스트 이름
- JOB_ID - 작업이 제출될 때 sge_qmaster에 의해 할당되는 고유한 ID. 작업 ID는 99999까지의 범위에 있는 십진 정수입니다.
- JOB_NAME - qsub 스크립트 파일이름, 마침표 및 작업 ID의 숫자로부터 빌드되는 작업 이름. 이 기본값을 qsub -N으로 겹쳐쓸 수 있습니다.
- LOGNAME - passwd 파일에 있는 사용자의 로그인 이름
- NHOSTS - 병렬 작업에 의해 사용 중인 호스트의 수
- NQUEUES - 작업에 대해 할당되는 대기열 수(직렬 작업의 경우 항상 1임)
- NSLOTS - 병렬 작업이 사용하는 대기열 슬롯의 수
- PATH - 기본 쉘 검색 경로 /usr/local/bin:/usr/ucb:/bin:/usr/bin
- PE - 작업이 실행하는 병렬 환경(병렬 작업의 경우에만)
- PE_HOSTFILE - Sun Grid Engine에 의해 병렬 작업에 할당되는 가상 병렬 시스템의 정의가 들어 있는 파일의 경로

이 파일의 형식에 대한 자세한 내용은 sge_pe에 있는 \$pe_hostfile 매개변수의 설명을 참조하십시오. 이 환경 변수는 병렬 작업의 경우에만 사용할 수 있습니다.

- QUEUE – 작업이 실행되고 있는 대기열의 이름
- REQUEST – 작업의 요청 이름으로, 작업 스크립트 파일 이름이거나 `qsub -N` 옵션을 통해 작업에 명시적으로 지정됩니다.
- RESTARTED – 검사점 작업이 재시작되었는지 또는 (값 1로) 설정된 경우 작업이 최소한 한 번 중단되어서 다시 시작되었는지 여부를 표시합니다.
- SHELL – `passwd` 파일에 있는 사용자의 로그인 셸

참고 – 이것이 반드시 작업에 대해 사용 중인 셸은 아닙니다.

- TMPDIR – 작업의 임시 작업 디렉토리에 대한 절대 경로
- TMP – TMPDIR과 동일하며 NQS와의 호환성을 위해 제공됩니다.
- TZ – 설정된 경우 `sgc_execd`에서 가져온 시간대 변수
- USER – `passwd` 파일에 있는 사용자의 로그인 이름.

▼ 명령줄에서 작업을 제출하는 방법

- **적당한 인수와 함께 `qsub` 명령을 입력하십시오.**

예를 들어 스크립트 파일 이름 `flow.sh`(68 페이지, “명령줄에서 간단한 작업을 실행하는 방법” 절에서 설명한 것처럼)를 사용하는 간단한 작업은 다음 명령으로 제출할 수 있습니다.

```
% qsub flow.sh
```

그러나 확장된 QMON 작업 제출의 동일한 결과를 얻으려면(그림 4-7에 표시된 것처럼) 다음과 같을 것입니다.

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

더 복잡한 요청을 구성하기 위해 추가 명령줄 옵션을 추가할 수 있습니다. 예를 들어 그림 4-11에 표시된 고급 작업 요청은 다음과 비슷합니다.

```
% 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
```

기본 요청

위의 절에 있는 마지막 예는 고급 작업 요청이 상당히 복잡하고 특히 유사한 요청이 자주 제출되어야 하는 경우에 다루기 어렵게 될 수 있음을 보여줍니다. 그런 명령줄을 입력하는 성가시고 오류를 낳는 업무를 피하기 위해 스크립트 파일에 `qsub` 옵션을 내장하거나(87 페이지, “활성 Sun Grid Engine 주석” 참조) 소위 *기본 요청*을 이용할 수 있습니다.

클러스터 관리가 모든 Sun Grid Engine 사용자에게 기본 요청 파일을 설정할 수 있습니다. 한편으로 사용자가 작업 디렉토리에 위치하는 응용 프로그램 특정 기본 요청 파일뿐 아니라 사용자의 홈 디렉토리에 위치하는 개인 기본 요청 파일을 작성할 수 있습니다.

기본 요청 파일에는 단순히 단일 또는 복수 행으로 Sun Grid Engine 작업에 대해 기본적으로 적용될 `qsub` 옵션이 들어 있습니다. 클러스터 글로벌 기본 요청 파일의 위치는 `<sgc 루트>/<셀>/common/sgc_request`입니다. 개인 일반 기본 요청 파일은 `$HOME/.sgc_request`에 위치하는 반면, 응용 프로그램 특정 기본 요청 파일은 `$cwd/.sgc_request`에 위치할 것입니다.

이들 파일의 둘 이상의 사용 가능한 경우 다음 우선권 순서로 하나의 기본 요청 파일에 병합됩니다.

1. 글로벌 기본 요청 파일.
2. 일반 개인 기본 요청 파일.
3. 응용 프로그램 특정 기본 요청 파일.

참고 - 스크립트 내장 및 `qsub` 명령줄이 기본 요청 파일보다 더 높은 우선순위를 갖습니다. 따라서 스크립트 내장이 기본 요청 파일 설정을 겹쳐쓰고, `qsub` 명령줄 옵션이 이들 설정을 다시 겹쳐줄 수 있습니다.

참고 - `qsub -clear` 옵션을 언제든지 기본 요청 파일, 내장된 스크립트 명령 및 `qsub` 명령줄에서 사용하여 모든 이전 설정을 삭제할 수 있습니다.

개인 기본 요청 파일의 예가 아래에 제공됩니다.

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

접촉하지 않는 경우, 주어진 사용자의 모든 작업에 대해 계정 문자열은 *내프로젝트*이고, 작업은 현재 작업 디렉토리에서 실행하고, 메일 통지가 작업의 시작 및 종료시에 *me@myhost.com*으로 전송되고, 작업이 시스템 붕괴 후에 재시작되며, 표준 출력 및 표준 오류 출력이 병합되고 `ksh`가 명령 해석기로 사용됩니다.

어레이 작업

동일한 조각(작업 스크립트에 들어 있는) 세트의 매개변수화되고 반복되는 실행이 **Sun Grid Engine 어레이작업** 기능의 이상적인 응용 프로그램입니다. 그런 응용 프로그램의 전형적인 예가 렌더링 같은 업무에 대한 디지털 콘텐츠 작성 산업에서 발견됩니다. 이 예에서 애니메이션의 연산은 프레임으로 분할되며 동일한 렌더링 연산을 각 프레임에 대해 독립적으로 수행할 수 있습니다.

어레이 작업 기능은 그런 응용 프로그램을 제출, 모니터 및 제어하는 편리한 방법을 제공합니다. 한편, **Sun Grid Engine**은 연산을 단일 작업으로 결합되는 독립적 업무의 어레이로서 처리하는 어레이 작업의 효율적인 구현을 제공합니다. 어레이 작업의 업무는 어레이 색인 번호를 통해 참조됩니다. 모든 업무에 대한 색인은 단일 `qsub` 명령에 의해 어레이 작업의 제출 중에 정의되는 전체 어레이 작업에 대한 색인 범위에 걸쳐 있습니다.

어레이 작업은 전체적으로 또는 개별 업무나 업무의 서브세트에 의해 모니터되고 제어(예를 들어, 일시중단, 재개 또는 취소)될 수 있으며, 이 경우 대응하는 색인 번호가 업무를 참조하기 위해 작업 ID에 접미어로 붙습니다. 업무가 실행될 때(정식 작업과 아주 유사하게) 업무는 환경 변수 `$$SGE_TASK_ID`를 사용하여 자신의 업무 색인 번호를 검색하고 이 업무 ID에 대해 지정된 입력 데이터 세트에 액세스할 수 있습니다.

▼ 명령줄에서 어레이 작업을 제출하는 방법

- 적당한 인수와 함께 `qsub` 명령을 입력하십시오.

다음은 어레이 작업 제출의 한 예입니다.

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

`-t` 옵션은 업무 색인 범위를 정의합니다. 이 경우 `2-10:2`는 2가 가장 낮고 10이 가장 높은 색인 번호인 한편 모든 두 번째 색인(지정의 `:2` 부분)만이 사용되도록 지정합니다. 따라서 어레이 작업은 업무 색인 2, 4, 6, 8 및 10을 갖는 5개 업무로 구성됩니다. 각 업무는 45분의 하드 CPU 시간 제한을 요청하며(`-l` 옵션) Sun Grid Engine에 의해 작업 지정되고 시작된 후에는 작업 스크립트 `render.sh`를 실행합니다. 업무는 `$SGE_TASK_ID`를 사용하여 업무 2, 4, 6, 8 또는 10인지 여부를 확인할 수 있으며 색인 번호를 사용하여 데이터 파일 `data.in`에서 입력 데이터 레코드를 찾을 수 있습니다.

▼ QMON을 사용하여 어레이 작업을 제출하는 방법

- 다음 참고를 추가로 고려하여 70 페이지, “그래픽 사용자 인터페이스 QMON에서 작업을 제출하는 방법”의 지침을 따르십시오.

참고 - QMON에서의 어레이 작업 제출은 가상적으로 70 페이지, “그래픽 사용자 인터페이스 QMON에서 작업을 제출하는 방법”에서 설명된 방법과 동일하게 작동합니다. 유일한 차이점은 그림 4-7에 표시된 작업 업무 입력 창이 `qsub -t` 옵션의 경우와 같이 동일한 구문을 갖는 업무 범위 지정을 포함해야 합니다. 어레이 색인 구문에 대한 자세한 정보는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*에 있는 `qsub` 항목을 참조하십시오.

112 페이지, “Sun Grid Engine 작업 모니터링 및 제어” 및 124 페이지, “명령줄에서 Sun Grid Engine 작업 제어”뿐 아니라 `qstat`, `qhold`, `qrls`, `qmod` 및 `qdel`에 관한 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서* 절에는 일반적인 Sun Grid Engine 작업 모니터링과 제어 및 특히 어레이 작업에 관한 정보가 들어 있습니다.

참고 - 어레이 작업은 정규 작업에 대해 알려진 모든 Sun Grid Engine 기능에 대한 전체 액세스를 제공합니다. 특히 동시에 병렬 작업일 수 있거나 다른 작업과의 상호 종속성을 가질 수 있습니다.

대화식 작업 제출

일괄처리 작업 대신 대화식 작업 제출은 작업이 작업의 결과에 영향을 주기 위해 사용자의 직접 입력을 필요로 하는 상황에서 유용합니다. 이것은 일반적으로 정의에 의해 대화식인 X-windows 응용 프로그램 또는 당면한 결과의 해석이 추가 연산을 조정하기 위해 필요한 업무의 경우입니다.

대화식 작업을 작성하기 위해 세 가지 메소드가 Sun Grid Engine 시스템에 존재합니다.

- **qlogin** - 이것은 Sun Grid Engine 소프트웨어에 의해 선택되는 호스트에서 시작되는 telnet형 세션입니다.
- **qrsh** - 이것은 표준 UNIX rsh 기능과 동일합니다. 명령이 Sun Grid Engine 시스템에 의해 선택되는 호스트에서 원격으로 실행되거나, 실행에 대한 명령이 지정되지 않은 경우 원격 호스트에서 원격 로그인(rlogin) 세션이 시작됩니다.
- **qsh** - 이것은 사용자 스펙 또는 DISPLAY 환경 변수의 설정에 대응하여 설정된 표시 장치로 작업을 실행하는 시스템에서 시작되는 xterm입니다. DISPLAY 변수가 설정되지 않고 표시장치 목적지가 특별히 정의되지 않은 경우, Sun Grid Engine이 xterm을 대화식 작업이 제출된 호스트의 X 서버의 0.0 화면으로 지시합니다.

참고 - 올바르게 기능하려면 모든 기능이 Sun Grid Engine 클러스터 매개변수의 적합한 구성이 필요합니다. 올바른 xterm 실행 경로가 qsh에 대해 정의되어야 하며 대화식 대기열이 이 유형의 작업에 대해 사용 가능해야 합니다. 클러스터가 대화식 작업 실행에 대해 준비되었는지 여부를 시스템 관리자에게 문의하십시오.

대화식 작업이 제출시에 실행될 수 없는 경우 대기열에 들어가지 않는다는 점에서 대화식 작업의 기본 처리는 일괄처리 작업의 처리와 다릅니다. 이것은 대화식 작업이 제출된 직후에 해당 작업을 작업 지정하기 위해 충분한 적당한 자원을 사용할 수 없음을 즉각적으로 표시할 것입니다. 그런 경우 Sun Grid Engine 클러스터가 현재 너무 바쁘다고 사용자에게 통지됩니다.

이 기본 작동은 qsh, qlogin 및 qrsh에 대한 -now no 옵션을 사용하여 변경할 수 있습니다. 이 옵션이 제공되는 경우 대화식 작업은 일괄처리 작업 같이 대기열에 들어갑니다. -now yes를 사용할 때 qsub와 함께 제출되는 일괄처리 작업은 또한 대화식 작업 같이 처리될 수 있으며 실행을 위해 즉시 작업 지정되거나 거부됩니다.

참고 - 대화식 작업은 유형 INTERACTIVE의 대기열에서만 실행될 수 있습니다(자세한 내용은 157 페이지, “대기열 구성 정보”를 참조).

후속 절들은 qlogin 및 qsh 기능의 사용을 간략히 설명합니다. qrsh 명령은 100 페이지, “투명한 원격 실행” 절에서 더 넓은 문맥으로 설명됩니다.

QMON을 사용한 대화식 작업 제출

QMON에서 제출할 수 있는 대화식 작업의 유일한 유형은 Sun Grid Engine에 의해 선택된 호스트에서 xterm을 호출하는 작업입니다.

▼ QMON을 사용하여 대화식 작업을 제출하는 방법

- **Interactive** 아이콘이 표시될 때까지 **Job Submission** 대화 상자의 오른쪽에 있는 단추 열의 맨 위에 있는 아이콘을 누르십시오.

이것은 대화식 작업을 제출할 Job Submission 대화 상자를 준비합니다(그림 4-13 및 그림 4-14 참조).

대화 상자에 있는 선택 옵션의 의미와 사용법은 73 페이지, “일괄처리 작업 제출” 절에서 일괄처리 작업에 대해 설명된 것과 동일합니다. 기본 차이점은 대화식 작업에 대해서는 적용되지 않기 때문에 여러 입력 필드가 반응하지 않도록 설정된다는 점입니다.

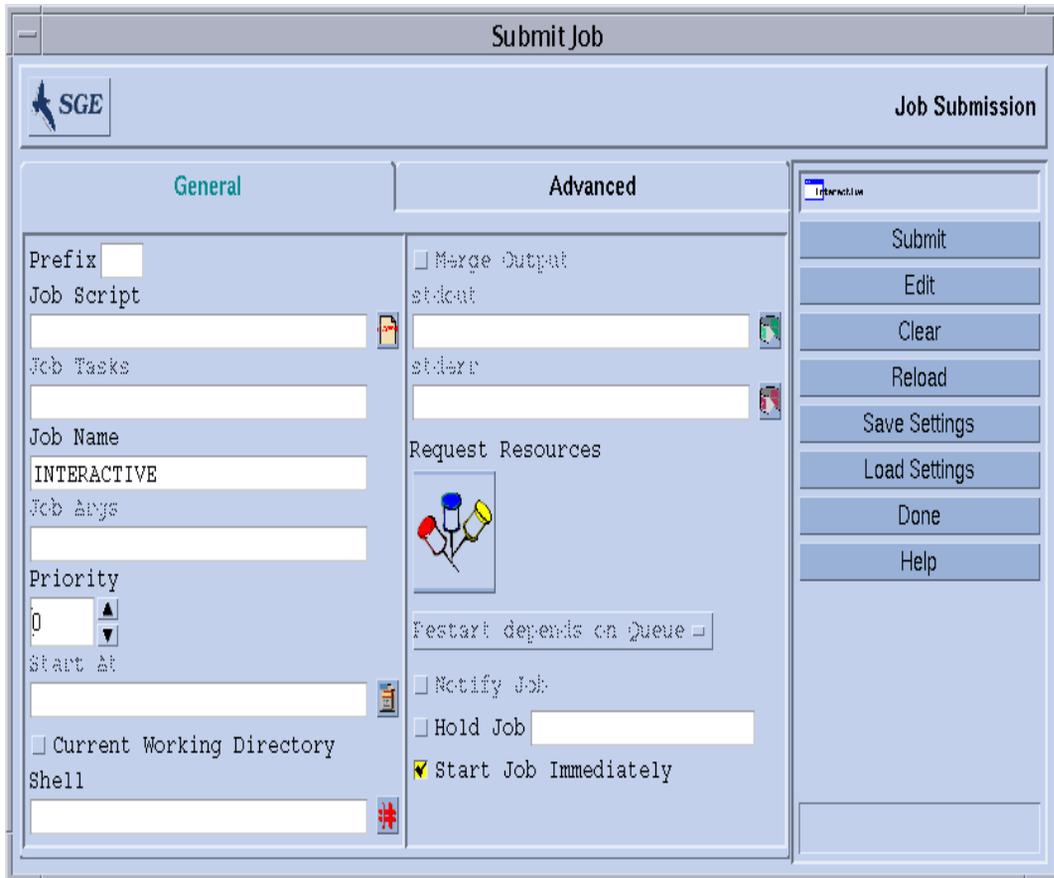


그림 4-13 대화식 작업 제출 대화 상자, General

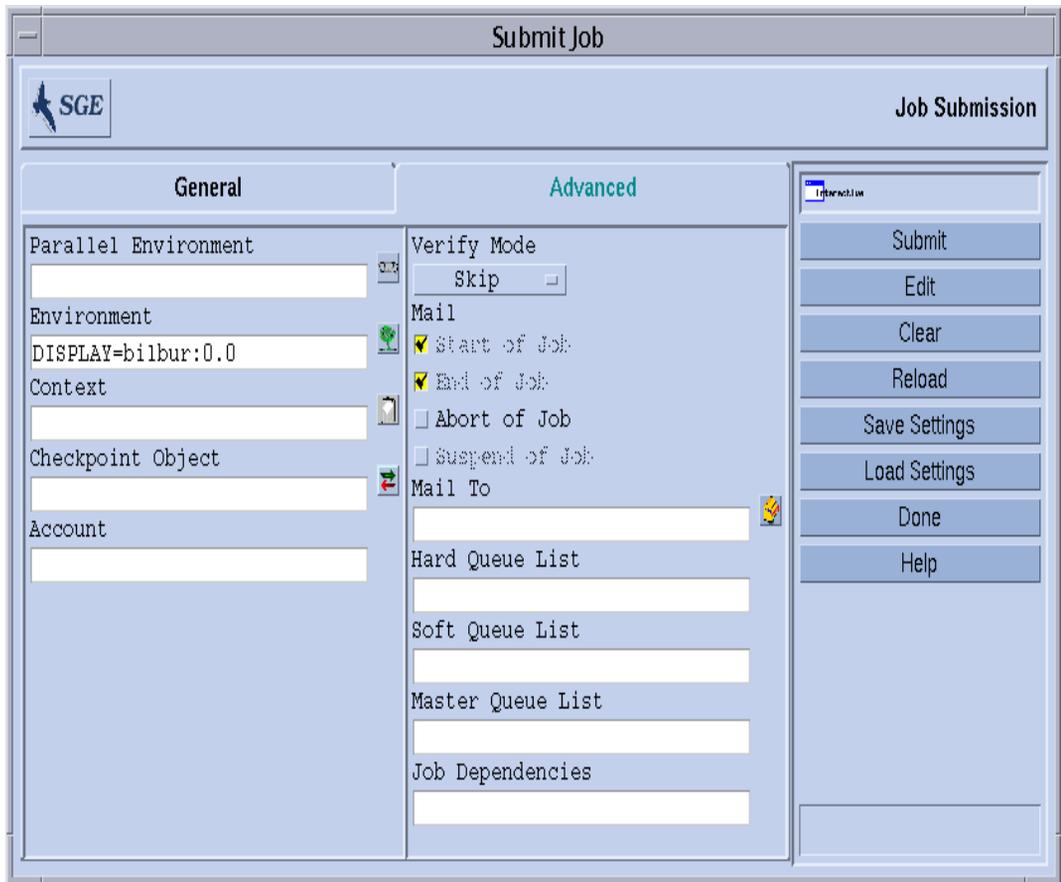


그림 4-14 대화식 작업 제출 대화 상자, Advanced

qsh를 사용한 대화식 작업 제출

Qsh는 qsub와 매우 비슷하며 여러 qsub 옵션뿐 아니라 호출될 xterm의 표시장치를 지시하는 추가 스위치 `-display`를 지원합니다(자세한 내용은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 qsh 항목 참조).

▼ qsh를 사용하여 대화식 작업을 제출하는 방법

- 다음 명령을 입력하여 모든 사용 가능한 Sun Solaris 64비트 운영 체제 호스트에서 xterm을 시작하십시오.

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

qlogin을 사용한 대화식 작업 제출

임의의 터미널 또는 터미널 에뮬레이션에서 qlogin 명령을 사용하여 Sun Grid Engine의 제어하에서 대화식 세션을 시작할 수 있습니다.

▼ qlogin을 사용하여 대화식 작업을 제출하는 방법

- 다음 명령을 입력하여 사용 가능한 Star-CD 사용권을 갖고 6 시간 하드 CPU 시간 제한을 제공하는 최소한 하나의 대기열을 갖는 부하가 적은 호스트를 찾으십시오.

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

참고 – Sun Grid Engine 시스템이 사용하도록 구성된 원격 로그인 기능에 따라서 로그인 프롬프트에 사용자 이름, 암호 또는 둘 모두를 입력해야 할 수 있습니다.

병렬 작업

Sun Grid Engine은 PVM 또는 MPI 같은 임의의 메시지 전달 환경(자세한 내용은 *PVM User's Guide* 및 *MPI User's Guide* 참조) 또는 단일 대기열 또는 복수 대기열 및 (분산 메모리 병렬 작업의 경우) 시스템 사이에 분산되는 복수 슬롯에서 공유 메모리 병렬 프로그램을 사용하여 병렬 작업을 실행하는 방법을 제공합니다. 여러 *병렬 환경* (PE) 인터페이스의 임의의 숫자가 동시에 공동으로 구성될 수 있습니다. PE에 대한 자세한 설명은 10 장, 233 페이지, “병렬 환경 관리”를 참조하십시오.

Sun Grid Engine 작업이 스케줄되는 방법

본질적으로, Sun Grid Engine 5.3 소프트웨어는 작업을 스케줄하는 두 세트의 기준을 사용합니다.

- 작업 우선순위
- 동일공유

작업 우선순위

다른 작업의 스케줄링 절차의 스케줄링 우선권 순서와 관련하여, *선입선출* (fifo) 규칙이 기본적으로 적용됩니다. 모든 *보류중* (스케줄되지 않은) 작업들은 목록에 삽입되는데, 해당 목록의 맨 위에는 첫 번째 제출된 작업, 그 밑에는 두 번째 제출된 작업 등이 있습니다. 처음 제출된 작업이 첫 번째로 스케줄됩니다. 최소한 하나의 적절한 대기열이 사용 가능한 경우, 작업이 스케줄됩니다. Sun Grid Engine 소프트웨어는 첫 번째 작업이 디스패치되었는지 여부에 상관없이 나중에 두 번째 작업을 스케줄합니다.

보류중 작업 사이의 이 우선 순위는 작업에 할당되어 있는 *우선순위 값* 을 통해 클러스터 관리가 지배할 수 있습니다. `gstat` 명령을 사용하여 표시될 수 있습니다. (우선순위 값은 `p`로 표시되는 보류중 작업의 마지막 열에 포함됩니다. 자세한 정보는 122 페이지, “`gstat` 를 사용하여 작업을 모니터링하는 방법” 절을 참조하십시오.) 제출 시간에 작업에 할당된 기본 우선순위 값은 0입니다. 우선순위 값은 양수 및 음수이며 보류중 작업 목록은 하향 우선순위 값의 순서로 대응하면서 정렬됩니다. 작업에 상대적으로 높은 우선순위 값을 할당하면 작업이 보류중인 작업 목록의 맨 위로 이동됩니다. 음수의 우선순위 값을 가진 작업은 작업이 막 제출된 후에도 삽입됩니다. 동일한 우선순위 값을 가진 여러 작업이 있는 경우, 선입선출 규칙이 해당 우선순위 값 범주 내에서 적용됩니다.

동일공유스케줄링

특히 사용자가 일련의 작업을 거의 동시에 제출하려는 경우에는(예를 들어, 하나 뒤에 또 다른 하나를 제출하는 쉘스크립트를 통해), 선입선출 규칙이 문제가 되기도 합니다. 나중에 제출되고 동일한 대기열 그룹에 지정되는 모든 작업은 너무 오래 기다려야 합니다. **동일공유스케줄링**은 우선권 목록의 끝에 실행중인 작업을 이미 소유하고 있는 사용자의 작업을 정렬하여 이 문제점을 피합니다. 정렬은 동일 우선순위 값 범주 내의 작업 사이에서만 수행됩니다. Sun Grid Engine 스케줄러 구성 입력 `user_sort` (자세한 설명은 `sched_conf` 매뉴얼 페이지 참조)이 `TRUE`로 설정된 경우 동일공유스케줄링이 사용됩니다.

대기열 선택

Sun Grid Engine 시스템은 작업이 즉시 시작될 수 없는 경우 특정하지 않은 대기열을 요청하는 작업을 작업 지정하지 않습니다. 그런 작업은 때때로 작업을 다시 스케줄하려 시도하는 `sge_qmaster`에 스푼되는 것으로 표시됩니다. 따라서 그런 작업은 사용 가능하게 되는 다음의 적합한 대기열에 작업 지정됩니다.

이와는 반대로 특정 대기열에 이름에 의해 요청되는 작업은 작업이 시작될 수 있는지 아니면 스푼되어야 하는지 여부와 상관없이 이 대기열로 직접 갑니다. 그러므로 Sun Grid Engine 대기열을 컴퓨터 과학 *일괄처리 대기열*로 보는 것은 이름에 의해 요청되는 작업의 경우에만 유효합니다. 특정하지 않은 요청을 갖고 제출되는 작업은 대기열에 들어가기 위해 `sge_qmaster`의 스푼링 메커니즘을 사용하므로 보다 추상적이고 유연한 대기열화 개념을 이용합니다.

작업이 스케줄되고 복수의 사용 가능한 대기열이 작업의 자원 요청을 만족하는 경우 작업은 대개 부하가 가장 적은 호스트에 속하는 대기열(적합한 것 중에서)에 작업 지정됩니다. Sun Grid Engine 스케줄러 구성 항목 `queue_sort_method`를 `seq_no`로 설정하여 클러스터 관리가 이 부하 중속 체계를 고정 순서 알고리즘으로 변경할 수 있습니다. 대기열 구성 항목 `seq_no`는 가장 낮은 순번을 갖는 대기열에 최상위 우선순위를 지정하여 대기열 사이에 우선권을 정의하는 데 사용됩니다.

투명한 원격 실행

Sun Grid Engine은 특정 연산 업무의 투명한 원격 실행을 지원하는 밀접하게 관련된 기능 세트를 제공합니다. 이 기능에 대한 코어 도구가 101 페이지, “`qrsh`를 사용한 원격 실행” 절에서 설명된 `qrsh` 명령입니다. `qrsh`의 위에 빌드되는 두 고급 기능(`qtcsh` 및 `qmake`)이 Sun Grid Engine을 통한 내재적 연산 업무의 투명한 분배를 허용하므로, 표준 UNIX 기능 `make` 및 `csch`를 향상시킵니다. `qtcsh`는 102 페이지, “`qtcsh`를 사용한 투명한 작업 분배” 절에서 설명되고 `qmake`는 104 페이지, “`qmake`를 사용한 병렬 Makefile 처리” 절에서 설명됩니다.

qrsh를 사용한 원격 실행

Qrsh는 표준 rsh 기능 주위에 빌드되며(rsh의 관련에 대한 자세한 내용은 <sgc 루트>/3rd_party에 제공되는 정보를 참조) 다양한 목적에 사용될 수 있습니다.

- 표준 UNIX 기능 rsh(HP-UX의 경우 remsh라고도 함)와 비교할 수 있는 Sun Grid Engine을 통한 대화식 작업의 원격 실행 제공하기 위해.
- 표준 UNIX 기능 rlogin과 유사한 Sun Grid Engine을 통해 대화식 로그인 세션 기능을 제공하기 위해(qlogin은 여전히 UNIX telnet 기능의 Sun Grid Engine 표시로서 필요함을 유의하십시오).
- 실행 시에 터미널 I/O(표준/오류 출력 및 표준 입력) 및 터미널 제어를 지원하는 일괄 처리 작업의 제출을 허용하기 위해.
- 셸 스크립트에 내장되지 않는 독립형 프로그램 제출을 위한 수단을 제공하기 위해.
- 작업이 보류 중이거나 실행 중인 동안 계속 활동 중이고 작업이 완료되거나 취소된 경우에만 종료하는 일괄처리 작업 제출 클라이언트를 제공하기 위해.
- 병렬 작업에 의해 할당되는 배포된 자원의 프레임워크 내에서 작업 업무(병렬 작업의 동시 업무 같은)의 Sun Grid Engine 시스템 제어 원격 실행을 허용하기 위해(243 페이지, “PE 및 Sun Grid Engine 소프트웨어의 밀접한 통합” 절을 참조하십시오).

이러한 기능 덕분에 qrsh가 qtcsh 및 qmake 기능의 구현뿐 아니라 MPI 또는 PVM 같은 병렬 환경과 Sun Grid Engine의 밀접한 통합을 위한 주요 활성화 하부구조입니다.

qrsh 사용법

qrsh 명령의 일반적인 형식은 다음과 같습니다.

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

qrsh는 qsub의 거의 모든 옵션을 이해하며 소수의 추가 옵션만을 제공합니다.

- -now yes|no - 이 옵션은 대개 대화식 작업에 바람직하므로(따라서 기본값임) 적당한 자원을 사용할 수 없는 경우 작업이 즉시 스케줄되고 거부되는지 여부 또는 작업이 제출시에 시작될 수 없는 경우 일괄처리 작업 같이 대기열에 들어가는지 여부를 제어합니다.
- -inherit - qrsh는 작업 업무를 시작하기 위해 Sun Grid Engine 스케줄링 프로세스를 통과하지 않지만 지정된 원격 실행 호스트에서 이미 적합한 자원을 할당한 병렬 작업의 문맥 안에 내장된다고 가정합니다. qrsh의 이 양식은 일반적으로 qmake 안에서 및 엄격한 병렬 환경 통합 내에서 사용됩니다. 기본값은 외부 작업 자원을 계승하지 않는 것입니다.
- -verbose - 이 옵션은 스케줄링 프로세스에 출력을 표시합니다. 주로 디버깅 목적을 위한 것이며 따라서 기본적으로 꺼집니다.

qtcsh를 사용한 투명한 작업 분배

qtcsh는 널리 알려지고 사용되는 UNIX C 셸(csh) 파생물인 tcsh에 대한 완전히 호환되는 대체물입니다(qmake는 tcsh 주위에 빌드됩니다. tcsh의 관계에 대한 자세한 내용은 <sgc 루트>/3rd_party에 제공되는 정보를 참조하십시오). 지정된 응용 프로그램의 실행을 Sun Grid Engine을 통해 적합하고 부하가 적은 호스트에 투명하게 분배하는 확장 기능을 갖는 명령 셸을 제공합니다. 원격으로 실행될 응용 프로그램과 실행 호스트의 선택에 적용되는 요구사항은 .qtask라는 구성 파일에 정의됩니다.

사용자에게 투명하므로 그런 응용 프로그램은 qrsh 기능을 통해 Sun Grid Engine에 실행을 위해 제출됩니다. qrsh가 원격으로 실행하는 응용 프로그램에 대한 터미널 제어 연결뿐 아니라 표준 출력, 오류 출력 및 표준 입력 처리를 제공하므로 그런 응용 프로그램을 셸과 동일한 호스트에서 실행하는 것에 비해 원격으로 실행하는 것 사이에는 단 세 가지의 주목할 만한 차이점이 있습니다.

- 원격 호스트가 응용 프로그램의 실행을 전혀 허용하지 않을 수 있는 로컬 호스트보다 훨씬 더 적합할 수 있습니다(더 강력하고, 부하가 낮으며, 하드/소프트웨어 자원이 설치되어야 합니다). 물론 이것이 원하는 차이점입니다.
- 작업의 원격 시작 및 Sun Grid Engine을 통한 처리에 의해 유발되는 약간의 지연이 있습니다.
- 관리자가 대화식 작업(qrsh) 및 따라서 qtcsh를 통해 자원의 사용을 제한할 수 있습니다. qrsh 기능을 통해 시작될 응용 프로그램에 대해 충분한 적합한 자원이 사용 불가능하거나 모든 적합한 시스템이 과부하되는 경우 내재적 qrsh 제출이 실패하고 대응하는 오류 메시지가 리턴됩니다(Not enough resources ... try later).

표준 사용 외에, qtcsh는 타사 코드 및 도구 통합에 적합한 플랫폼입니다. 통합 환경 안에서 qtcsh를 단일 응용 프로그램 실행 양식인 qtcsh -c *응용프로그램이름*으로 사용하면 거의 변경되지 않아야 하는 지속적 인터페이스가 표시됩니다. 모든 필수 응용 프로그램, 도구, 통합, 사이트 및 심지어 사용자 특정 구성이 적절하게 정의된 .qtask 파일에 들어 있습니다. 추가 장점은 이 인터페이스를 임의 유형의 C 프로그램 및 심지어 Java 응용 프로그램의 셸 스크립트 안에서 사용할 수 있다는 점입니다.

qtcsh 사용법

qtcsh의 호출은 tcsh의 경우와 정확하게 같습니다. Qtcsh는 .qtask 파일에 대한 지원을 제공하고 특수한 셸 내장 모드 세트를 제공하여 tcsh를 확장합니다.

.qtask 파일은 다음과 같이 정의됩니다. 파일의 각 행은 다음 형식을 갖습니다.

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

선택적인 선행 감탄 부호(!)는 클러스터 글로벌 .qtask 파일 및 qtcsh 사용자의 개인 .qtask 파일에 있는 상충하는 정의 사이의 우선순위를 정의합니다. 클러스터 글로벌 파일에 감탄 부호가 누락되는 경우 결국 사용자 파일의 상충하는 정의가 지배합니다. 클러스터 글로벌 파일에 감탄 부호가 있는 경우 해당하는 정의를 겹쳐줄 수 없습니다.

행의 나머지는 qtcsh의 명령줄에 입력될 때 원격 실행을 위해 Sun Grid Engine에 제출될 응용 프로그램의 이름과 응용 프로그램에 대한 자원 요구사항을 정의하고 사용될 qrsh 기능에 대한 옵션을 지정합니다.

참고 - 응용 프로그램 이름은 .qtask 파일에 정의되는 것처럼 정확하게 명령줄에 나타나야 합니다. 절대 또는 상대 디렉토리 지정이 접두어로 붙는 경우 로컬 이진이 주소 지정되고 원격 실행이 의도되지 않는다고 가정합니다.

참고 - 그러나 csh 별명이 응용 프로그램 이름과의 비교가 수행되기 전에 확장됩니다. 원격 실행을 위한 응용 프로그램은 또한 qtcsh 명령줄의 어디에나, 특히 표준 I/O 경로 재지정 앞 또는 뒤에 나타날 수 있습니다.

따라서 다음 예는 유효하고 의미있는 구문입니다.

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

이 .qtask 파일이 주어질 때 다음 qtcsh 명령줄은

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

내재적으로 다음 결과를 갖습니다.

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

qtcsch는 각각을 켜거나 끌 수 있는 스위치에 의해 영향을 받는 여러 모드에서 동작할 수 있습니다.

- 명령의 로컬 또는 원격 실행(원격이 기본값임)
- 즉시 또는 일괄처리 원격 실행(즉시가 기본값임)
- Verbose 또는 비 verbose 출력(비 verbose가 기본값임)

이들 모드의 설정은 시작시에 qtcsch의 옵션 인수를 사용하여 또는 실행시에 쉘 내장 명령 qrshmode를 사용하여 변경할 수 있습니다. 자세한 내용은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 qtcsch 항목을 참조하십시오.

qmake를 사용한 병렬 Makefile 처리

qmake는 표준 UNIX make 기능의 대체입니다. 독립적 make 단계를 적합한 시스템의 클러스터 사이에 분배하는 기능에 의해 make를 확장합니다. qmake는 대중적인 GNU-make 기능인 gmake로부터 빌드됩니다. gmake의 관련에 대한 자세한 내용은 <sge 루트>/3rd_party에 제공되는 정보를 참조하십시오.

복잡한 배포 make 프로세스가 완료할 때까지 실행할 수 있도록 보장하기 위해 qmake는 먼저 병렬 작업과 비슷한 형태로 필수 자원을 할당합니다. Qmake는 그런 다음 Sun Grid Engine 스케줄링과의 추가 상호작용 없이 이 자원 세트를 관리합니다. 자원이 -inherit 옵션이 사용되는 qrsh 기능을 통해 사용 가능하거나 가능하게 될 때 make 단계를 분배합니다.

qrsh가 원격으로 실행하는 make 단계에 대한 터미널 제어 연결뿐 아니라 표준 출력, 오류 출력 및 표준 입력 처리를 제공하므로, make 절차를 로컬로 실행하는 것과 qmake 사용 사이에는 세 가지의 주목할 만한 차이점만이 있습니다.

- 개별 make 단계가 특정 존속 기간을 갖고 처리될 독립적 make 단계가 충분한 경우 make 프로세스의 병렬화가 상당히 가속화됩니다. 물론, 이것은 원하는 차이점입니다.
- 원격으로 시작될 make 단계에서 qrsh 및 원격 실행 자체에 의해 유발되는 내재적인 작은 오버헤드가 있습니다.
- qmake의 make 단계 분배를 활용하려면 사용자가 최소한 병렬화 정도, 즉 동시에 실행 가능한 make 단계의 수를 지정해야 합니다. 또한 사용자는 사용 가능한 소프트웨어 사용권, 시스템 아키텍처, 메모리 또는 CPU 시간 요구사항 같이 make 단계에서 필요한 자원 특성을 지정할 수 있습니다.

일반적으로 make의 가장 일반적인 사용은 분명히 복잡한 소프트웨어 패키지의 컴파일입니다. 그러나 이것이 qmake에 대한 주된 응용이 아닐 수 있습니다. 프로그램 파일은 종종 아주 작으며(좋은 프로그래밍 관례의 문제로서) 따라서 단일 make 단계인 단일 프로그램의 컴파일은 보통 수 초밖에 소요되지 않습니다. 게다가 파일 서버가 모든 파일 액세스를 효과적으로 직렬화하는 병목이 될 수 있기 때문에 컴파일은 대개 복수 make 단계에 대해 병렬로 수행되는 경우 가속되지 않을 수 있는 많은 파일 액세스(내포된 포함 파일)를 암시합니다. 따라서 컴파일 프로세스의 만족할 만한 가속화를 기대할 수 없습니다.

qmake의 다른 잠재적인 응용이 더 적합합니다. 한 예는 make-file을 통한 복잡한 분석 업무의 상호종속성 및 워크플로우의 조정입니다. 이것은 EDA 같은 일부 영역에서 일반적이며 그런 환경의 각 make 단계는 대개 무시할 수 없는 자원과 연산 시간 요구사항을 갖는 시뮬레이션 또는 데이터 분석 조작입니다. 그런 경우에 상당한 가속화를 달성할 수 있습니다.

qmake 사용법

qmake의 명령줄 구문은 qrsh의 구문과 아주 유사하게 보입니다.

```
% qmake [-pe PE 이름 PE 범위][further options] \  
-- [gnu-make-options][대상]
```

참고 - -inherit 옵션도 이 절에서 나중에 설명하는 것처럼 qmake에 의해 지원됩니다.

-pe 옵션 사용 및 그의 gmake -j 옵션에 대한 관계에 특별히 주의해야 합니다. 두 옵션 모두가 달성될 병렬화의 양을 표현하는 데 사용할 수 있습니다. 차이점은 gmake가 -j를 사용하여 사용할 병렬 환경 같은 사항을 지정하는 가능성을 제공하지 않는다는 점입니다. 따라서 qmake는 make라고 부르는 병렬 make의 기본 환경이 구성된다고 가정합니다. 게다가 gmake의 -j는 범위의 지정을 허용하지 않으며 단일 숫자만을 허용합니다. Qmake는 -j로 주어지는 숫자를 1-<주어진숫자>의 범위로 해석합니다. 이와는 반대로 -pe는 이러한 모든 매개변수의 상세한 지정을 허용합니다. 결국, 다음 명령줄 예는 동일합니다..

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

반면에 다음 명령줄은 -j 옵션을 통해 표현될 수 없습니다.

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

구문과는 별도로 qmake는 두 호출 모드, 즉명령줄에서 대화식으로(-inherit 없이) 또는 일괄처리 작업 내에서(-inherit 포함)를 지원합니다. 이들 두 모드는 서로 다른 조치 순서를 시작합니다.

- **대화식** - qmake가 명령줄에서 호출될 때 make 프로세스가 프로세스로서 qmake 명령줄에 지정되는 자원 요구사항을 고려하는 qrsh를 통해 내재적으로 Sun Grid Engine에 제출됩니다. 그러면 Sun Grid Engine이 병렬 make 작업과 연관된 병렬 작업의 실행을 위한 *마스터 시스템*을 선택하고 그곳에서 make 절차를 시작합니다. 이것은 make 프로세스가 아키텍처에 종속될 수 있고 필수 아키텍처가 qmake 명령줄에 지정되기 때문에 필요합니다. 그런 다음 마스터 시스템의 qmake 프로세스가 개별 make 작업의 실행을 작업에 대해 Sun Grid Engine에 의해 할당되었고 병렬 환경 호스트 파일을 통해 qmake에 전달되는 다른 호스트에 위임합니다.
- **일괄처리** - 이 경우에 qmake는 -inherit 옵션을 갖는 일괄처리 스크립트 안에 나타냅니다(-inherit 옵션이 없는 경우 위의 첫 번째 경우에 대해 설명한 것처럼 새 작업이 파생됩니다). 이것은 qmake가 qmake가 내장되는 작업에 이미 할당된 자원을 사용하게 합니다. qrsh -inherit를 직접 사용하여 make 단계를 시작합니다. 일괄처리 모드로 qmake를 호출할 때 자원 요구사항 또는 -pe 및 -j 옵션의 지정은 무시됩니다.

참고 - 또한, 단일 CPU 작업이 병렬 환경(qmake -pe make 1 --)을 요청해야 합니다. 병렬 실행이 필요하지 않은 경우 gmake 명령줄 구문(Sun Grid Engine 옵션 및 "--" 없음)과 함께 qmake를 호출하면 gmake처럼 작동할 것입니다.

qmake에 대한 자세한 내용은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 qmake 항목을 참조하십시오.

작업 검사점, 모니터링 및 제어

Sun Grid Engine 5.3 시스템을 사용하여 작업을 제출한 후 작업을 모니터링하고 제어할 수 있어야 합니다. 이 장은 이들 업무 수행에 대한 백그라운드 정보 및 지침을 둘다 제공합니다.

이 장에는 다음 특정 업무에 대한 지침이 들어 있습니다.

- 110 페이지, “명령줄에서 검사점 작업을 제출, 모니터 또는 삭제하는 방법”
- 111 페이지, “QMON을 사용하여 검사점 작업을 제출하는 방법”
- 112 페이지, “QMON을 사용하여 작업을 모니터 및 제어하는 방법”
- 122 페이지, “qstat 를 사용하여 작업을 모니터링하는 방법”
- 124 페이지, “전자 우편에 의해 작업을 모니터링하는 방법”
- 125 페이지, “명령줄에서 작업을 제어하는 방법”
- 126 페이지, “QMON을 사용하여 대기열을 제어하는 방법”
- 130 페이지, “qmod를 사용하여 대기열을 제어하는 방법”

작업 검사점 정보

이 절은 작업 검사점의 두 가지 유형에 대해 알아봅니다.

- 사용자 레벨
- 커널 레벨

사용자 레벨 검사점

많은 응용 프로그램, 특히 정상적으로 상당한 CPU 시간을 소비하는 응용 프로그램은 결합 허용한계를 증가시키기 위해 검사점 및 재시작 메커니즘을 구현했습니다. 상태 정보 및 처리된 데이터의 중요한 부분이 알고리즘의 특정 단계에서 하나 이상의 파일에 반복적으로 기록됩니다. 이들 파일(재시작 파일이라고 부름)은 응용 프로그램이 중단되고 나중에 재시작되는 경우에 처리될 수 있으며, 체크포인트 직전의 상황과 비교할 수 있는 일관성있는 상태에 도달할 수 있습니다. 사용자가 주로 재시작 파일을 적합한 위치로 이동하기 위해 해당 파일을 다루어야 하기 때문에 이런 종류의 검사점을 *사용자 레벨* 검사점이라고 합니다.

통합된(사용자 레벨) 검사점을 갖지 않는 응용 프로그램의 경우 공용 도메인(예를 들어 위스콘신 대학교의 *Condor* 프로젝트를 참조하십시오) 또는 일부 하드웨어 공급자가 제공할 수 있는 소위 *검사점 라이브러리*를 사용하는 것이 대안일 수 있습니다. 응용 프로그램을 그런 라이브러리와 다시 링크하면 소스 코드 변경이 필요없이 응용 프로그램에 검사점 메커니즘이 설치됩니다.

커널 레벨 검사점

일부 운영 체제는 운영 체제 커널 안에서 검사점 지원을 제공합니다. 이 경우에는 응용 프로그램에서의 준비와 응용 프로그램의 재 연결이 필요없습니다. 커널 레벨 검사점은 대개 전체 프로세스 계층뿐 아니라 단일 프로세스에도 적용할 수 있습니다. 즉, 상호종속적 프로세스의 계층이 언제든지 체크포인트되고 재시작될 수 있습니다. 대개 체크포인트를 시작하기 위해 사용자 명령과 C 라이브러리 인터페이스 둘다 사용할 수 있습니다.

Sun Grid Engine은 사용 가능한 경우 운영 체제 검사점을 지원합니다. 현재 지원되는 커널 레벨 검사점 기능에 대한 정보는 Sun Grid Engine 릴리스 노트를 참조하십시오.

검사점 작업의 이주

검사점 작업은 그의 재시작 기능이 이미 수행된 소수의 작업만이 반복되어야 하도록 보장하므로 언제든지 인터럽트할 수 있습니다. 이 기능이 Sun Grid Engine의 이주 및 동적 부하 균형 메커니즘을 구축하는 데 사용됩니다. 요청되는 경우 검사점 Sun Grid Engine 작업이 요구에 따라 중단되고 Sun Grid Engine 풀의 다른 시스템으로 이주되므로 동적으로 클러스터의 부하를 균일하게 합니다. 검사점 작업은 다음 이유 때문에 중단되고 이주됩니다.

- 실행 중인 대기열 또는 작업이 `qmod` 또는 `qmon` 명령에 의해 명시적으로 일시중단됩니다.

- 대기열에 대한 일시중단 임계값이 초과되었고(162 페이지, “Load 및 Suspend Thresholds 구성 방법” 절 참조) 작업에 대한 체크포인트 근거 지정이 일시중단 경우를 포함하기(110 페이지, “명령줄에서 검사점 작업을 제출, 모니터 또는 삭제하는 방법” 절 참조) 때문에 실행 중인 대기열 또는 작업이 자동으로 일시중단됩니다.

qstat 출력에서 이주에 대한 상태 m을 사용하여 이주하려는 작업을 식별할 수 있습니다. 이주 작업이 다시 sge_qmaster로 이동하고 그 뒤에 하나가 사용 가능한 경우 다른 적합한 대기열에 작업 지정됩니다.

검사점 작업 스크립트 구성

커널 레벨 검사점에 대한 셸 스크립트는 정규 셸 스크립트와 차이를 보이지 않습니다.

사용자 레벨 검사점에 대한 셸 스크립트는 재시작되는 경우를 적절하게 처리하는 능력에 있어서만 정규 Sun Grid Engine 일괄처리 스크립트와 다릅니다. 환경 변수인 RESTARTED가 재시작되는 검사점 작업에 대해 설정됩니다. 이 환경 변수는 초기 호출 중에만 실행되어야 하는 작업 스크립트의 섹션을 건너뛰는 데 사용될 수 있습니다.

따라서 투명하게 검사점하는 작업 스크립트는 코드 예 5-1과 유사하게 보일 수 있습니다.

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

# Test if restarted/migrated
if [ $RESTARTED = 0 ]; then
    # 0 = not restarted
    # Parts to be executed only during the first
    # start go in here
    set_up_grid
fi

# Start the checkpointing executable
fem
#End of scriptfile
```

코드 예 5-1 검사점 작업 스크립트의 예

사용자 레벨 검사점 작업이 이주되는 경우 작업 스크립트가 처음부터 재시작되는 것에 주의해야 합니다. 사용자는 셸 스크립트의 프로그램 흐름을 작업이 인터럽트된 위치로 보낸 후 두 번 이상 실행되는 것이 치명적인 스크립트의 해당 행을 건너뛰는 책임을 갖습니다.

참고 - 커널 레벨 검사점 작업은 언제든지 인터럽트할 수 있고 또한 포함하는 쉘 스크립트가 정확하게 최종 체크포인트가 발생한 위치에서 재시작됩니다. 그러므로 RESTARTED 환경 변수는 커널 레벨 검사점 작업과는 관련이 없습니다.

▼ 명령줄에서 검사점 작업을 제출, 모니터 또는 삭제하는 방법

적당한 스위치를 갖는 다음 명령을 입력하십시오.

```
#qsub 옵션 인수
```

검사점 작업 제출은 정규 일괄처리 스크립트의 경우와 동일한 방법을 작업하며, 검사점 메커니즘을 요청하고 체크포인트가 작업에 대해 생성되어야 하는 근거를 정의하는 `qsub -ckpt` 및 `-c` 스위치는 예외입니다. `-ckpt` 옵션은 사용될 검사점 환경(225 페이지, “검사점 지원 정보”)의 이름인 하나의 인수를 갖습니다. `-c` 옵션은 필수가 아니며 역시 하나의 인수를 갖습니다. 이 인수는 검사점 환경 구성의 `when` 매개변수의 정의를 겹쳐쓰는 데 사용될 수 있습니다(자세한 내용은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `checkpoint` 항목을 참조하십시오).

`-c` 옵션에 대한 인수는 다음의 1 문자 선택(또는 그의 모든 조합) 또는 대안으로서 시간 값 중 하나일 수 있습니다.

- **n** - 검사점이 수행되지 않았습니니다. 이것이 가장 높은 우선순위를 갖습니다.
- **s** - 작업 호스트의 `sge_execd`가 종료되는 경우에만 검사점이 생성됩니다.
- **m** - 대응하는 대기열 구성에서 정의되는 최소 CPU 간격으로 체크포인트를 생성합니다(`queue_conf` 매뉴얼 페이지의 `min_cpu_interval` 매개변수를 참조하십시오).
- **x** - 작업이 일시중단되는 경우에 체크포인트가 생성됩니다.
- **interval - min_cpu_interval**(위 참조)에 의해 정의되는 것보다 자주는 아니지만 주어진 간격으로 체크포인트를 생성합니다. 시간 값이 `hh:mm:ss`(콜론으로 구분되는 두 자리 숫자의 시, 분 및 초)으로 지정되어야 합니다.

검사점 작업의 *모니터링*은 이들 작업이 때때로 이주할 수 있으며(`qstat`,의 출력에서 이주에 대한 상태 `m`에 의해 지정됨. 위를 참조) 따라서 단일 대기열에 구속되지 않는다는 사실에 의해 정규 작업과는 다릅니다. 그러나 고유한 작업 ID 숫자는 작업 이름과 마찬가지로 동일하게 유지됩니다.

검사점 작업 삭제는 124 페이지, “명령줄에서 Sun Grid Engine 작업 제어” 절에서 설명된 것과 동일한 방법으로 작동합니다.

▼ QMON을 사용하여 검사점 작업을 제출하는 방법

- 다음의 추가 정보를 주의하면서 79 페이지, “고급 보기”의 지침을 따르십시오.

QMON을 통한 검사점 작업의 제출은 적당한 검사점 환경을 지정하는 것을 제외하고는 정규 일괄처리 작업의 제출과 동일합니다. 79 페이지, “고급 보기” 절차에서 설명한 것처럼 Job Submission 대화 상자는 작업과 연관된 검사점 환경을 위한 입력 창을 제공합니다. 입력 창 옆에는 그림 5-1에 표시되는 Selection 대화 상자를 여는 아이콘 단추가 있습니다. 이 아이콘 단추를 사용하여 사용 가능한 환경 목록에서 적합한 검사점 환경을 선택할 수 있습니다. 사용자 사이트에 설치된 검사점 환경의 등록정보에 대한 정보에 대해 시스템 관리자에게 문의하거나 225 페이지, “검사점 지원 정보”를 참조하십시오.



그림 5-1 검사점 오브젝트 선택

파일 시스템 요구사항

검사점 라이브러리 기반 사용자 레벨 또는 커널 레벨 검사점이 기록될 때 체크포인트될 프로세스 또는 작업이 포함하는 가상 메모리의 전체 이미지가 덤프되어야 합니다. 이 목적을 위해 충분한 디스크 공간을 사용할 수 있어야 합니다. 검사점 환경 구성 매개변수 `ckpt_dir`이 설정되는 경우 검사점 정보가 `ckpt_dir`의 작업 개인 위치에 덤프됩니다. `Ckpt_dir`이 `NONE`으로 설정되는 경우 검사점 작업이 시작된 디렉토리가 사용됩니다. 검사점 환경 구성에 관한 자세한 정보는 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 checkpoint 항목을 참조하십시오.

참고 - ckpt_dir이 NONE으로 설정되는 경우 qsub -cwd 스크립트로 검사점 작업을 시작해야 합니다.

파일 시스템이 구성되는 방법에 관한 추가 요구사항은 작업을 성공적으로 이주하고 재시작하기 위해 검사점 파일과 재시작 파일이 모든 시스템에서 표시되어야 한다는 사실에 의해 유발됩니다. 따라서 NFS 또는 비슷한 파일 시스템이 필수입니다. 이 요구사항이 사용자 사이트에 대해 만족되는 경우 클러스터 관리에게 문의하십시오.

사용자 사이트가 NFS를 실행하지 않거나 어떤 이유로 NFS를 사용하는 것이 바람직하지 않은 경우 사용자 레벨 검사점 작업의 경우에 셸 스크립트의 시작시에 재시작 파일을 명시적으로 전송(예를 들어 rcp 또는 ftp를 통해)할 수 있어야 합니다.

Sun Grid Engine 작업 모니터링 및 제어

원리상 제출된 작업을 모니터링하는 세 가지 방법이 있습니다.

- Sun Grid Engine 그래픽 사용자 인터페이스인 QMON을 사용하여
- qstat 명령을 사용하여 명령줄에서
- 전자 우편에 의해

▼ QMON을 사용하여 작업을 모니터 및 제어하는 방법

Sun Grid Engine 그래픽 사용자 인터페이스인 QMON은 작업 제어를 위해 특별히 설계된 대화 상자를 제공합니다.

- QMON 기본 메뉴에서 **Job Control** 단추를 누른 후 다음 절에서 자세히 설명하는 추가 정보에 따라 계속하십시오.

이 대화 상자의 일반 목적은 시스템 또는 그의 부분에 알려진 모든 실행 중, 보류 중 및 종료된 작업의 구성 가능한 수를 모니터링하는 방법을 제공하는 것입니다. 대화 상자는 또한 작업을 조작, 즉 우선순위 변경, 일시중단, 재개 및 취소하는 데 사용할 수 있습니다. 세 개의 목록 환경이 표시되는데, 하나는 실행 중인 작업에 대한 것이고, 다른 하나는 적당한 자원에 작업 지정되기를 기다리는 보류 중 작업에 대한 것이며 세 번째는 최근에 종료된 작업에 대한 것입니다. 화면의 맨 위에 있는 대응하는 탭 레이블을 눌러서 세 목록 환경 중에서 선택할 수 있습니다.

기본 양식(그림 5-2 참조)에서는 각 실행 중 및 보류 중 작업에 대한 JobId, Priority, JobName 및 Queue 열을 표시합니다.

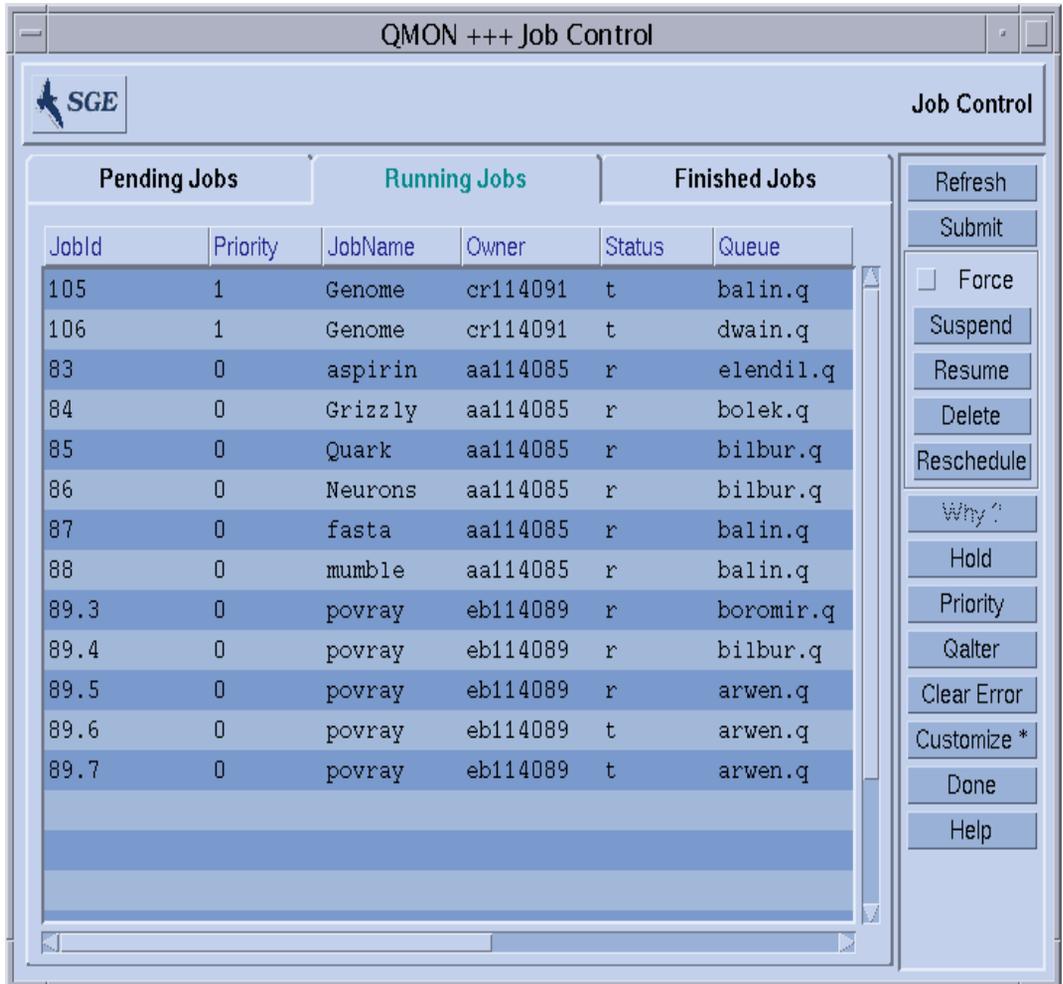


그림 5-2 Job Control 대화 상자—표준 양식

Job Control 대화 상자의 Customize 단추를 누를 때 열리는 Customization 대화 상자(그림 5-3 참조)와 함께 표시되는 정보 세트를 구성할 수 있습니다.

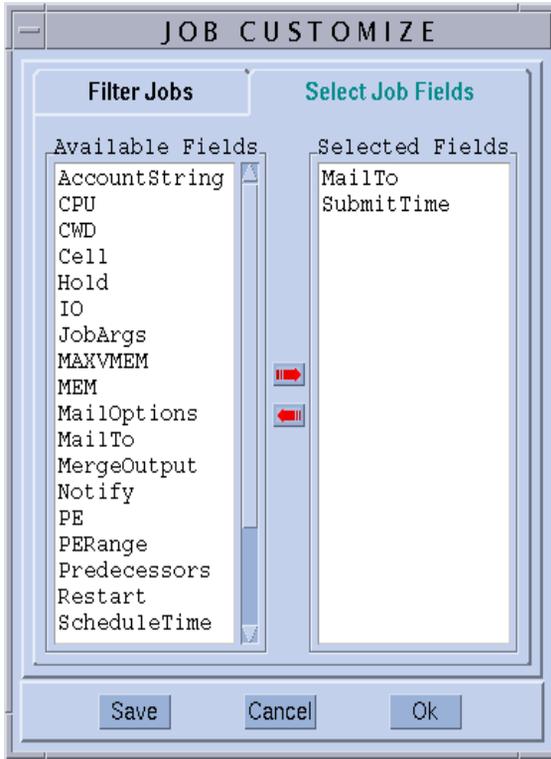


그림 5-3 Job Control Customization 대화 상자

Customization 대화 상자를 사용하여 표시되고 관심있는 작업을 필터링하는 Sun Grid Engine 작업 오브젝트의 추가 항목을 선택할 수 있습니다. 그림 5-3의 예는 추가 필드로 MailTo 및 Submit Time을 선택합니다.

그림 5-4에 표시되는 Job Control 대화 상자는 Finished Jobs 목록의 경우에 사용자 정의가 적용된 후의 향상된 모습을 묘사합니다.



그림 5-4 Job Control 대화 상자 Finished Jobs—항상

그림 5-5의 필터링 기능의 예는 solaris64 구조에 적합하거나 실행하는 ferst1가 소유한 해당 작업들만 선택합니다.

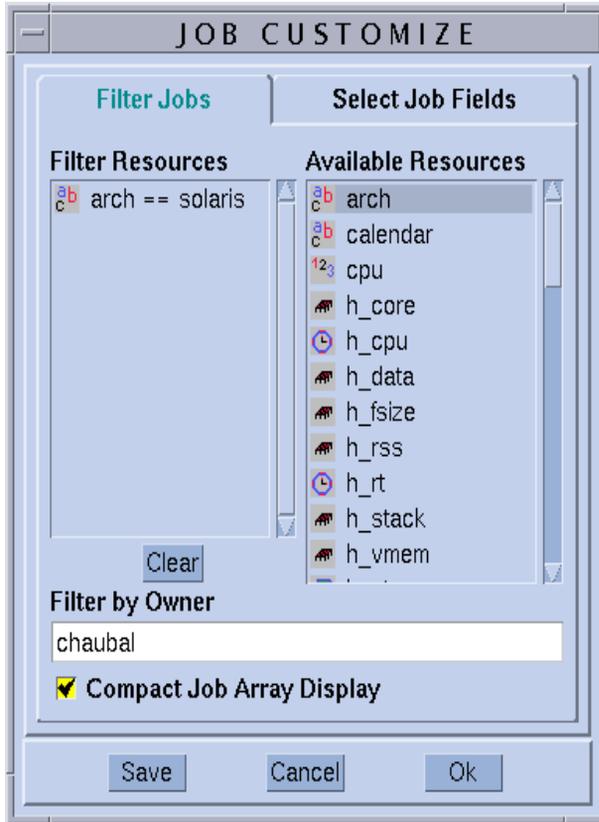


그림 5-5 작업 제어 필터링

Running Jobs을 표시하는 결과 Job Control 대화 상자가 그림 5-6에 표시됩니다.

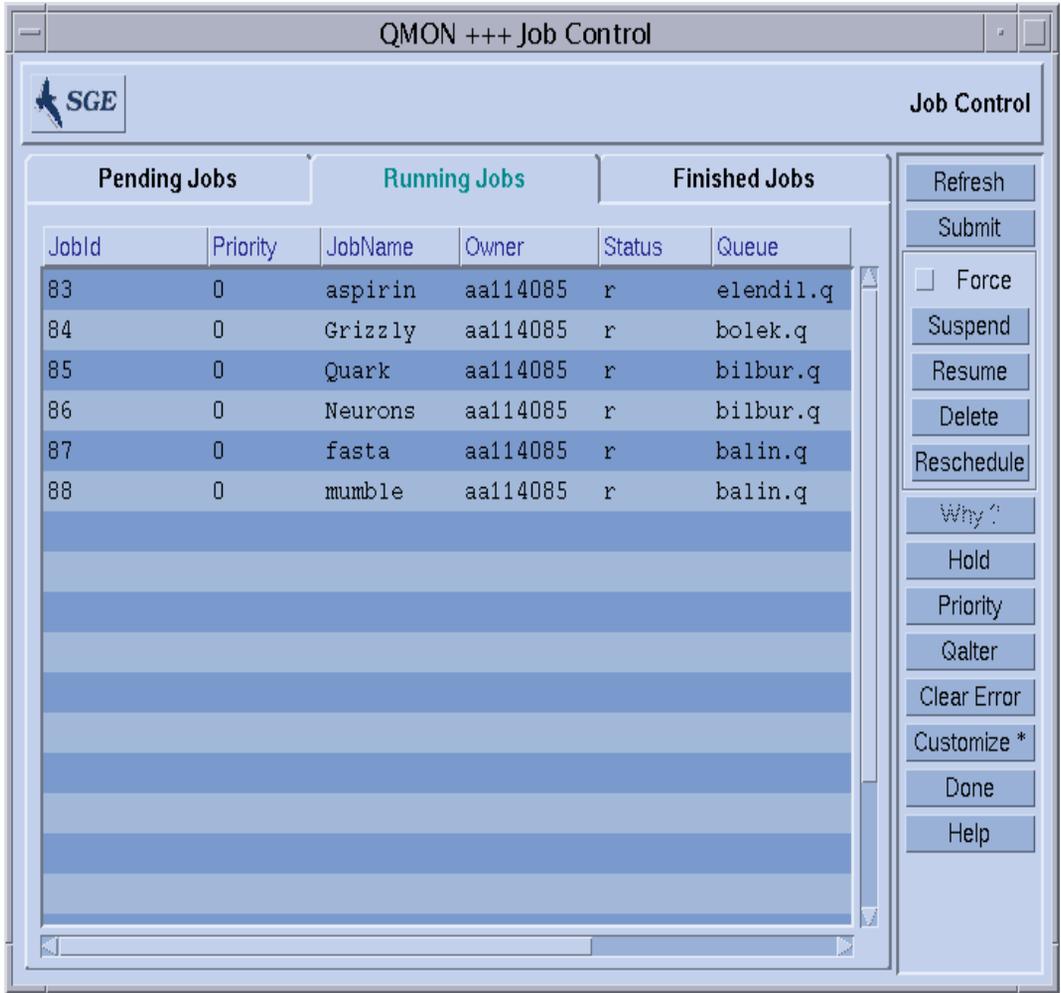


그림 5-6 Job Control 대화 상자—필터링 후

참고 - 예를 들어 그림 5-3의 Customization 대화 상자에 표시되는 Save 단추는 사용자 정의를 사용자의 홈 디렉토리에 있는 `.qmon_preferences` 파일에 저장하며 따라서 Job Control 대화 상자의 기본 외양을 재정의합니다.

그림 5-6의 Job Control 대화 상자도 어레이 작업이 QMON에 표시되는 방법의 예입니다.

다음의 마우스/키 조합을 사용하여 작업을 선택할(나중의 조작을 위해) 수 있습니다.

- Control 키를 누른 상태에서 왼쪽 마우스 단추로 작업을 누르면 복수 작업의 선택이 시작됩니다.
- Shift 키를 누른 상태에서 왼쪽 마우스 단추로 다른 작업을 누르면 선택 시작에 있는 작업과 현재 작업을 포함하여 그 사이의 모든 작업이 선택됩니다.
- Control 키를 누른 상태에서 왼쪽 마우스 단추로 작업을 누르면 단일 작업의 선택 상태가 전환됩니다.

선택된 작업은 화면의 오른쪽에 있는 대응하는 단추를 통해 일시중단, 재개(일시중단 해제), 삭제, 다시 보류(및 해제), 재 우선순위화 및 수정(Alter)될 수 있습니다.

일시중단, 일시중단 해제, 삭제, 보류, 우선순위 수정 및 작업 수정 조치는 작업 소유자 또는 Sun Grid Engine 관리자 및 운영자에 의해서만 작업에 적용될 수 있습니다(65 페이지, “관리자, 운영자 및 소유자” 참조). 실행 중인 작업만을 일시중단/재개할 수 있으며 보류 중 작업만을 보류 및 수정(다른 속성에서뿐 아니라 우선순위에서)할 수 있습니다.

작업 일시중단은 UNIX kill 명령을 사용하여 SIGSTOP 신호를 작업의 프로세스 그룹으로 전송하는 것과 동등한 것을 의미하며, 이것은 작업을 정지시키고 더 이상 CPU 시간을 소비하지 않습니다. 작업 일시중단 해제는 SIGCONT 신호를 전송하므로 작업을 재개합니다(신호 전달 프로세스에 대한 자세한 정보는 시스템에 대한 kill 매뉴얼 페이지를 참조하십시오).

참고 - 일시중단, 일시중단 해제 및 삭제는 강제 집행할 수 있습니다. 즉, 대응하는 sge_execd가 예를 들어 네트워크 문제점으로 인해 도달 불가능한 경우 sge_execd 제어 작업의 통지 없이 sge_qmaster에 등록됩니다. 이 목적을 위해서는 Force 플래그를 사용하십시오.

선택된 보류 중 작업에서 Hold 단추를 사용하는 경우 Set Hold 부속 대화 상자가 열립니다(그림 5-7 참조).



그림 5-7 작업 제어 보류

Set Hold 부속 대화 상자는 사용자, 시스템 및 운영자 보류 설정 및 재설정을 가능케 합니다. 사용자 보류는 Sun Grid Engine 운영자 및 관리자뿐 아니라 작업 소유자가 설정 또는 재설정할 수 있습니다. 운영자 보류는 관리자 및 운영자가 설정 또는 재설정할 수 있고, 시스템 보류는 관리자만이 설정 또는 재설정할 수 있습니다. 모든 보류가 작업에 지정되는 동안은 작업이 실행될 수 없습니다. 보류를 설정 또는 재설정하는 대체 방법은 `qalter`, `qhold` 및 `qrls` 명령입니다(*Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서의 해당 항목을 참조하십시오*).

Priority 단추를 누르면 다른 부속 대화 상자가 열리는데(그림 5-8), 이 대화 상자에서 선택된 보류 중 작업의 새 우선순위를 입력할 수 있습니다. Sun Grid Engine에서 우선순위는 Job Control가 표시하는 보류 중 작업 순서 및 보류중 작업 목록에서의 작업 순서를 결정합니다. 사용자는 0과 1024 사이의 범위에서만 우선순위를 선택할 수 있습니다. Sun Grid Engine 운영자 및 관리자 또한 최고 1023까지 우선순위를 증가시킬 수 있습니다(작업 우선순위에 대한 자세한 설명은 99 페이지, “작업 우선순위” 절을 참조).

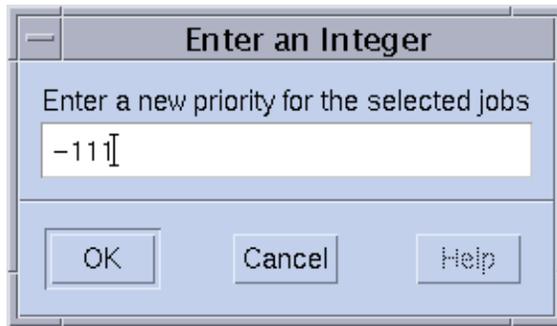


그림 5-8 작업 제어 우선순위 정의

Qalter 단추는 보류 중 작업에 대해 누를 때 작업 제출 중에 정의된 작업의 속성에 대응하여 설정된 대화 상자의 모든 항목을 갖는 70 페이지, “그래픽 사용자 인터페이스 QMON에서 작업을 제출하는 방법”에 설명된 작업 제출 화면을 엽니다. 변경할 수 없는 항목들은 응답하지 않도록 설정됩니다. 다른 것은 편집할 수 있으며 변경사항은 Job Submission 대화 상자의 Qalter 단추(Submit 단추의 대체)를 눌러서 Sun Grid Engine에 등록됩니다.

작업 제출 화면의 Verify 플래그는 Qalter 모드에서 사용될 때 특별한 의미를 갖습니다. 보류 중인 작업의 일관성을 점검하고 아직 스케줄되지 않은 이유를 조사할 수 있습니다. 단지 Verify 플래그에 대해 원하는 일관성 점검 모드를 선택하고 Qalter 단추를 누르면 됩니다. 시스템은 선택한 점검 모드에 따라서 불일치에 대한 경고를 표시합니다. 추가 정보는 79 페이지, “고급 보기” 절과 `qalter` 매뉴얼 페이지의 `-w` 옵션을 참조하십시오.

작업이 여전히 보류중인 이유를 확인하기 위한 또 다른 방법은 작업을 선택하여 Job Control 대화 상자의 why? 단추를 누르는 것입니다. 그러면 Object Browser 대화 상자가 열리고 Sun Grid Engine 스케줄러가 가장 최근 통과에서 작업을 작업 지정하지 못하게 한 이유의 목록이 표시됩니다. 그런 메시지를 표시하는 예제 브라우저 화면이 그림 5-9에 표시됩니다.

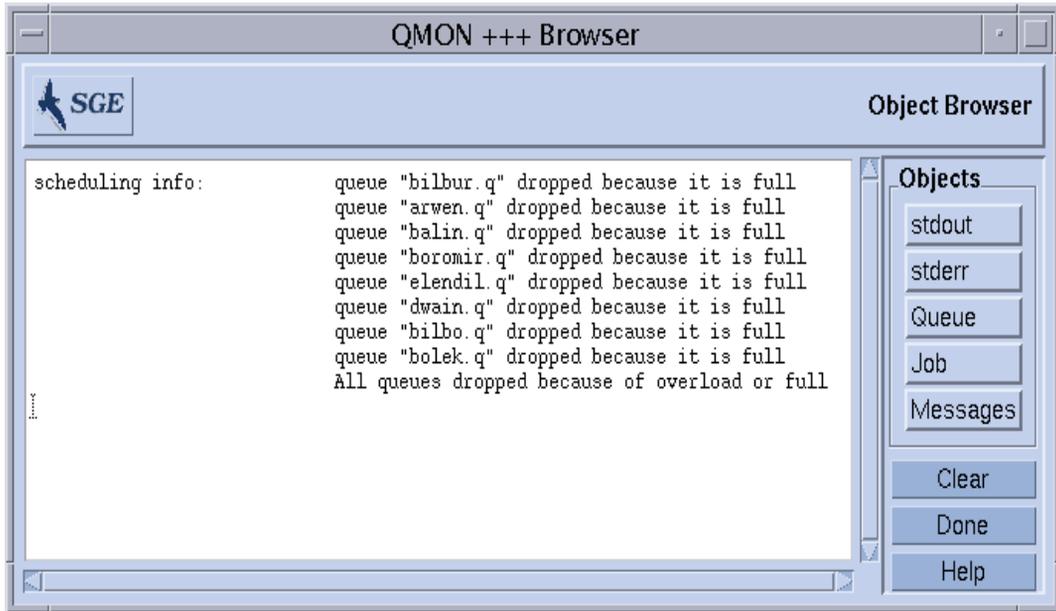


그림 5-9 스케줄링 정보를 표시하는 브라우저

참고 - why? 단추는 스케줄러 구성 매개변수 schedd_job_info가 true로 설정되는 경우에만 의미있는 출력을 제공합니다(*Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 sched_conf를 참조하십시오). 표시되는 스케줄러 정보는 최종 스케줄링 간격과 관련됩니다. 작업이 스케줄되지 않은 이유를 조사하는 시점에서는 더 이상 정확하지 않을 수 있습니다.

Clear Error 단추를 사용하여 선택된 보류 중 작업에서 이전 시도에서 시작되었지만 작업 관련 문제점(예: 지정된 작업 출력 파일에 쓰기 위한 권한 부족)으로 인해 실패한 오류 상태를 제거할 수 있습니다.

참고 - 오류 상태는 보류 중 작업 목록에서 빨간색 글꼴을 사용하여 표시되며 오류 조건을 정정(예를 들어 qalter를 통해)한 후에만 제거됩니다. 작업이 중단되는 경우에 작업이 이메일을 전송하도록 요청하는 경우(예를 들어 qsub -m a 옵션을 통해) 그런 오류 조건은 자동으로 전자 우편을 통해 보고됩니다.

정보가 계속 최신 상태로 표시되도록 하기 위해 QMON은 폴링 체계를 사용하여 sge_qmaster에서 작업의 상태를 검색합니다. refresh 단추를 눌러서 강제로 갱신할 수 있습니다.

마지막으로 이 단추는 QMON Job Submission 대화 상자에 대한 링크를 제공합니다(예를 들어 그림 5-10을 참조하십시오).

QMON 오브젝트 브라우저를 사용한 추가 정보

QMON Object Browser를 사용하여 112 페이지, “QMON을 사용하여 작업을 모니터 및 제어하는 방법” 절에서 설명한 것처럼 Job Control 대화 상자를 사용자 정의할 필요 없이 Sun Grid Engine 작업에 대한 추가 정보를 빨리 검색할 수 있습니다.

QMON 기본 메뉴의 Browser 아이콘 단추를 누를 때 Object Browser가 열립니다. 브라우저에서 Job 단추를 선택하는 경우와 마우스 포인터가 Job Control 대화 상자에서 작업의 행위로 이동하는 경우 브라우저는 Sun Grid Engine 작업에 관한 정보를 표시합니다(예에 대해서는 그림 5-2를 참조하십시오).

그림 5-10의 브라우저 화면은 그런 상황에서 표시되는 정보의 예를 제공합니다.

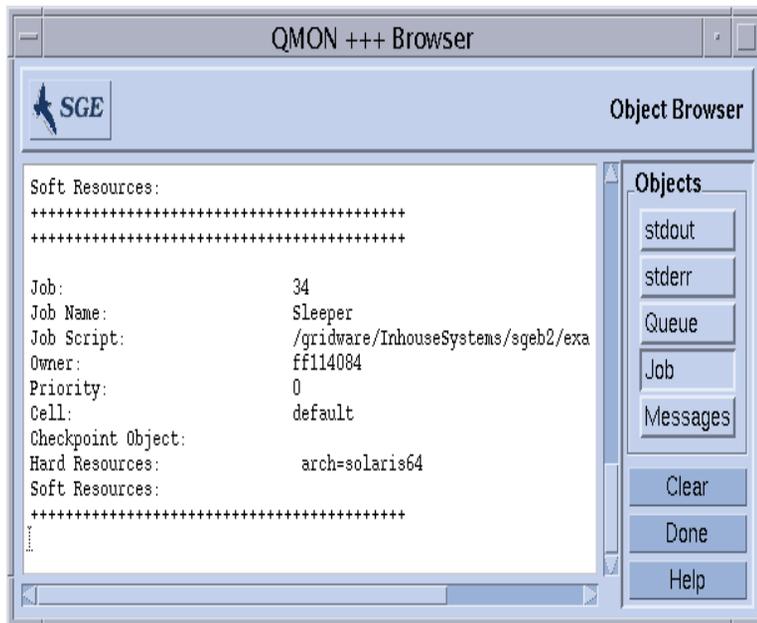


그림 5-10 Object Browser—Job

▼ qstat 를 사용하여 작업을 모니터링하는 방법

- 다음 절에서 상세히 설명하는 정보를 바탕으로 명령줄에서 다음 명령 중 하나를 사용하십시오.

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

첫 번째 양식은 제출된 작업의 개요만을 제공합니다(표 5-1을 참조하십시오). 두 번째 양식은 현재 구성된 대기열에 대한 정보를 추가로 포함합니다(표 5-2를 참조하십시오).

첫 번째 양식에서 헤더 행은 열의 의미를 표시합니다. 대부분의 열의 목적이 자체 설명됩니다. 그러나 state 열에는 다음 의미를 갖는 단일 문자 코드가 들어 있습니다. r은 실행 중, s는 일시중단됨, q는 대기열에 들어감 및 w는 대기 중을 의미합니다(qstat 출력 형식에 대한 자세한 설명은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 qstat 항목을 참조하십시오).

두 번째 양식은 두 섹션으로 나뉘어지는데, 첫 번째 섹션은 모든 사용 가능한 대기열의 상태를 표시하고 두 번째 섹션(- PENDING JOBS - ... 구분문자로 제목이 지정됨)은 sge_qmaster 작업 스푼 영역의 상태를 표시합니다. 대기열 섹션의 첫 행은 나열된 대기열에 관한 열의 의미를 정의합니다. 대기열은 수평 괄선에 의해 구분됩니다. 작업이 대기열에서 실행하는 경우 첫 번째 양식에 qstat 명령에서와 같은 형식으로 연관된 대기열 아래에 인쇄됩니다. 두 번째 출력 섹션의 보류 중 작업도 qstat의 첫 번째 양식에 인쇄됩니다.

대기열 설명의 다음 열은 약간의 추가 설명이 필요합니다.

- qtype - 이것은 대기열 유형입니다.
즉, B(일괄처리), I(대화식), P(병렬) 및 C(검사점) 또는 그들의 조합입니다.
- used/free - 대기열에 있는 사용/사용 가능 작업 슬롯의 계수입니다.
- states - 대기열의 상태입니다.
즉, u(알 수 없음), a(경보), s(일시중단), d(사용 안함), E(오류) 또는 그의 조합 중 하나입니다.

다시, qstat 매뉴얼 페이지에 qstat 출력 형식의 더 상세한 설명이 들어 있습니다.

qstat 명령에 대한 다양한 추가 옵션들이 양 버전의 기능성을 향상시킵니다. -r 옵션을 사용하여 제출된 작업의 자원 요구사항을 표시할 수 있습니다. 게다가 출력이 특정 사용자 특정 대기열로 제한될 수 있으며 -l 옵션을 사용하여 qsub 명령에 대해 83 페이지, “자원 요구사항 정의” 절에서 설명한 것처럼 자원 요구사항을 지정할 수 있습니다. 자원 요구사항이 사용되는 경우 qstat 명령줄의 자원 요구사항 스푼과 일치하는 대기열(및 이들 대기열에서 실행 중인 작업)만이 표시됩니다.

표 5-1 qstat 출력의 예

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

표 5-2 qstat -f 출력의 예

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

▼ 전자 우편에 의해 작업을 모니터하는 방법

- 다음 절에서 상세히 설명하는 정보를 바탕으로 명령줄에서 적당한 인수를 갖는 다음 명령을 입력하십시오.

```
% qsub 인수
```

qsub -m 스위치는 특정 이벤트가 발생하는 경우 작업을 제출하는 사용자나 -M 플래그에 의해 지정되는 이메일 주소로 전자 우편이 전송되도록 요청합니다(플래그의 설명에 대해서는 qsub 매뉴얼 페이지를 참조). -m 옵션에 대한 인수가 이벤트를 지정합니다. 다음 선택을 사용할 수 있습니다.

- b - 작업을 시작할 때 메일이 전송됩니다.
- e - 작업을 종료할 때 메일이 전송됩니다.
- a - 작업이 중단될 때(예를 들어, qdel 명령에 의해) 메일이 전송됩니다.
- s - 작업이 일시중단될 때 메일이 전송됩니다.
- n - 메일이 전송되지 않습니다(기본값).

선택으로 구분된 목록으로 단일 -m 옵션과 함께 이들 옵션의 복수를 선택할 수 있습니다.

QMON Job Submission 대화 상자의 도움을 받아서 동일한 메일 이벤트를 구성할 수 있습니다. 79 페이지, “고급 보기” 절을 참조하십시오.

명령줄에서 Sun Grid Engine 작업 제어

112 페이지, “QMON을 사용하여 작업을 모니터 및 제어하는 방법” 절은 Sun Grid Engine 그래픽 사용자 인터페이스인 QMON을 사용하여 Sun Grid Engine 작업을 삭제, 일시중단 및 재개할 수 있는 방법을 설명합니다.

이 절에서 설명하는 것처럼 동등한 기능을 명령줄에서도 사용할 수 있습니다.

▼ 명령줄에서 작업을 제어하는 방법

- 다음 절에서 상세히 설명하는 정보를 바탕으로 명령줄에서 다음 명령 중 하나와 적당한 인수를 입력하십시오.

```
% qdel 인수  
% qmod 인수
```

qdel 명령을 사용하여 Sun Grid Engine 작업을 실행 중인지 아니면 스푼되는지 여부와 상관없이 취소합니다. qmod 명령은 이미 실행 중인 작업을 일시중단 및 일시중단 해제(해제)하는 수단을 제공합니다.

두 명령 모두에 대해 작업 ID 번호를 알아야 하는데, 성공적인 qsub 명령에 대한 응답으로 표시됩니다. 번호를 잊은 경우 qstat를 통해 검색할 수 있습니다(122 페이지, “qstat 를 사용하여 작업을 모니터링하는 방법” 절 참조).

아래에 두 명령에 대한 여러 예가 있습니다.

```
% qdel 작업ID  
% qdel -f 작업ID1, 작업ID2  
% qmod -s 작업ID  
% qmod -us -f 작업ID1, 작업ID2  
% qmod -s 작업ID. 업무ID 범위
```

작업을 삭제, 일시중단 또는 일시중단 해제하려면 작업의 소유자, Sun Grid Engine 관리자 또는 운영자 중 하나여야 합니다(65 페이지, “관리자, 운영자 및 소유자”를 참조하십시오).

두 명령 모두에 대해 -f 강제 옵션을 사용하여 sge_execd에 도달할 수 없는 경우(예를 들면 네트워크 문제점으로 인해) sge_execd에 접속하지 않고 sge_qmaster에서 작업에 대한 상태 변경을 등록할 수 있습니다. -f 옵션은 관리자가 사용하기 위한 것입니다. 그러나 qdel의 경우 클러스터 구성 qmaster_params 항목에 플래그 ENABLE_FORCED_QDEL이 설정되는 경우 사용자가 자신의 작업의 삭제를 강제할 수 있습니다(자세한 정보는 Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서의 sge_conf 매뉴얼 페이지를 참조하십시오).

작업 종속성

복잡한 업무를 자주 빌드하는 가장 편리한 방법은 업무를 부속 업무로 분할하는 것입니다. 이들 경우에 부속 업무는 시작될 수 있기 전에 다른 부속 업무의 성공적인 완료에 종속됩니다. 전임 업무가 후속 업무에서 읽고 처리해야 하는 출력 파일을 생성하는 것이 한 예입니다.

Sun Grid Engine은 작업 종속성 기능으로 상호종속적 업무를 지원합니다. 하나 또는 복수의 다른 작업의 성공적인 완료에 의존하도록 작업을 구성할 수 있습니다. 이 기능은 `qsub -hold_jid` 옵션에 의해 강제 실행됩니다. 제출된 작업이 의존하는 작업 목록을 지정할 수 있습니다. 또한 작업 목록이 어레이 작업의 서브세트를 포함할 수도 있습니다. 제출된 작업은 종속성 목록의 모든 작업이 성공적으로 완료되어야 실행될 수 있습니다.

대기열 제어

55 페이지, “대기열 및 대기열 등록정보” 절에서 설명한 것처럼 대기열의 소유자는 대기열을 일시중단/일시중단 해제 또는 사용 안함/사용하는 권한을 갖습니다. 이것은 이들 사용자가 중요한 작업을 위해 수시로 특정 시스템이 필요한 경우 및 백그라운드에서 실행 중인 Sun Grid Engine 작업에 의해 크게 영향을 받는 경우에 바람직합니다.

대기열을 일시중단 또는 사용 가능하게 하는 두 가지 방법이 있습니다.

- QMON Queue Control 대화 상자
- `qmod` 명령

▼ QMON을 사용하여 대기열을 제어하는 방법

- QMON 기본 메뉴에서 Queue Control 단추를 누르십시오.

그림 5-11에 표시된 것과 비슷한 Queue Control 대화 상자가 표시됩니다.

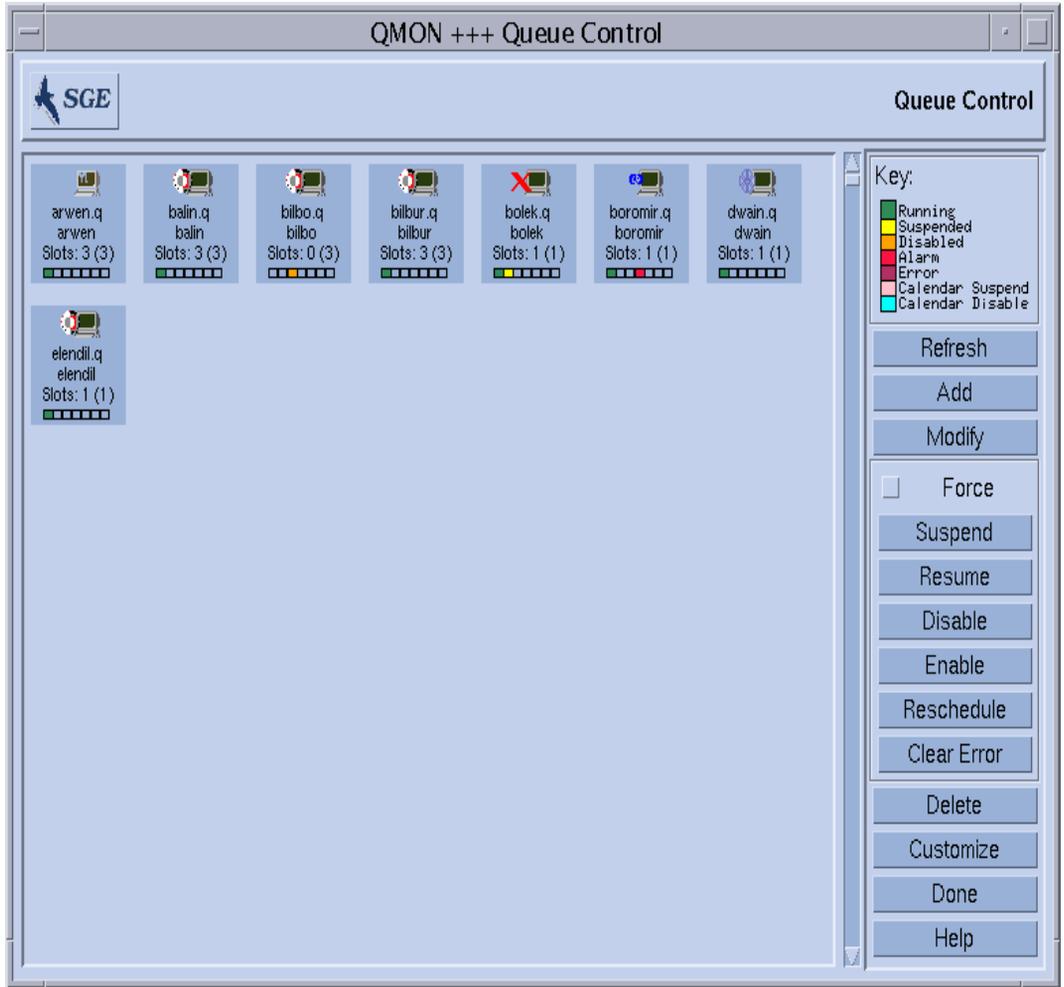


그림 5-11 Queue Control 대화 상자

Queue Control 대화 상자의 목적은 사용 가능한 자원과 클러스터의 활동에 대한 빠른 개요를 제공하는 것입니다. 또한 대기열을 구성할 뿐 아니라 대기열을 일시중단/일시중단 해제 및 사용 안함/사용하는 수단을 제공합니다. 표시되는 각 아이콘은 대기열을 표시합니다. 주 표시 영역이 빈 경우 구성된 대기열이 없습니다. 각 대기열 아이콘은 대기열 이름, 대기열이 상주하는 호스트의 이름 및 점유 중인 작업 슬롯의 수로 레이블됩니다. sge_execd가 대기열 호스트에서 실행 중이고 sge_qmaster에 이미 등록된 경우 대기열 아이콘의 그림은 대기열 호스트의 운영 체제 아키텍처를 표시하고 아이콘의 맨 아래에 있는 컬러 막대는 대기열의 상태를 알려줍니다. Queue Control 대화 상자의 오른쪽에 있는 범례가 컬러의 의미를 표시합니다.

해당 대기열에 대해 사용자는 키보드의 **Shift** 키를 누른 상태에서 왼쪽 마우스 단추로 대기열 아이콘을 눌러서 대기열 및 내재적으로 대기열을 호스트하는 시스템에 대한 현재 속성, 부하 및 자원 소비 정보를 검색할 수 있습니다. 이 조치는 그림 5-12에 표시된 것과 비슷한 정보 화면을 팝업합니다.

대기열은 왼쪽 마우스로 해당 단추 또는 대기열 아이콘 단추를 둘러싸는 직사각형 영역 안을 눌러서 선택됩니다. **Delete**, **Suspend/Unsuspend** 또는 **Disable/Enable** 단추를 사용하여 선택된 대기열에 대해 해당하는 조작을 실행할 수 있습니다. 일시중단/일시중단 해제 및 사용 안함/사용 조작은 대응하는 `sge_execd`의 통지가 필요합니다. 이것이 불가능한 경우(예를 들어 호스트가 정지했기 때문에) `sge_qmaster` 내부 상태 변경은 **Force toggle** 단추가 켜지는 경우에 강제 실행될 수 있습니다.

대기열이 일시중단되는 경우 대기열은 추가 작업에 대해 닫히고 대기열에서 이미 실행 중인 작업들은 112 페이지, “**QMON**을 사용하여 작업을 모니터 및 제어하는 방법” 절에서 설명한 것처럼 일시중단됩니다. 대기열 및 그의 작업은 대기열이 일시중단 해제되자마자 재개됩니다.

참고 - 일시중단된 대기열의 작업이 추가로 명시적으로 일시중단된 경우 대기열이 일시중단 해제되는 경우에 해당 작업은 재개되지 않습니다. 명시적으로 다시 작업을 일시중단 해제해야 합니다.

사용하지 않는 대기열은 닫히지만, 해당 대기열에서 실행 중인 작업은 계속될 수 있습니다. 대기열 사용 안함은 일반적으로 대기열을 “배출“하는 데 사용됩니다. 대기열이 활성화된 후 다시 작업 실행이 가능합니다. 여전히 실행 중인 작업에는 어떤 조치도 수행되지 않습니다.

일시중단/일시중단 해제 및 사용 안함/사용 조작은 대기열 소유자 또는 **Sun Grid Engine** 관리자 또는 운영자 권한이 필요합니다(65 페이지, “관리자, 운영자 및 소유자” 절을 참조하십시오).

Queue Control 대화 상자에 표시되는 정보는 주기적으로 갱신됩니다. **refresh** 단추를 눌러서 강제로 갱신할 수 있습니다. **Done** 단추는 대화 상자를 닫습니다.

Customize 단추를 사용하여 필터 조작을 통해 표시될 대기열을 선택할 수 있습니다. 그림 5-13의 샘플 화면은 아키텍처 `osf4`(즉, Compaq UNIX 버전 4)에 속하는 호스트에서 실행하는 대기열만의 선택을 보여줍니다. **Customization** 대화 상자의 **Save** 단추를 사용하여 **QMON**의 차후 호출시 표준 재활성화를 위해 사용자의 홈 디렉토리의 `.qmon_preferences` 파일에 설정을 저장할 수 있습니다.

대기열 구성을 위해 Queue Control 화면의 오른쪽에 있는 Add 또는 Modify 단추를 누를 때 부속 대화 상자가 열립니다(자세한 내용은 158 페이지, “QMON을 사용하여 대기열을 구성하는 방법” 절을 참조하십시오).

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

그림 5-12 대기열 속성 화면

대기열에 첨부되는 모든 속성(호스트 또는 클러스터로부터 상속되는 속성 포함)이 속성 열에 나열됩니다. Slot-Limits/Fixed Attributes 열은 대기열 슬롯 한계당 또는 고정 콤프렉스 속성당 정의되는 속성에 대한 값을 표시합니다. Load(scaled)/Consumable 열은 보고된(그리고 조정으로 구성되는 경우) 부하 매개변수(197 페이지, “부하 매개변수” 참조) 및 Sun Grid Engine 소비 가능 자원 기능을 기반으로 사용할 수 있는 자원 용량에 대해 알려줍니다(185 페이지, “소비 가능 자원” 절을 참조하십시오).

참고 – 부하 보고서 및 소비 가능한 용량은 부하 속성이 소비 가능한 자원으로 구성되는 경우 서로 겹쳐줄 수 있습니다. 작업 지정 알고리즘에서 사용되는 두 속성 모두의 최소값이 표시됩니다.

참고 - 표시되는 부하 및 소비 가능한 값은 28 페이지, “실행 호스트” 절에서 설명하는 것처럼 현재 부하 조정 보정을 고려하지 않습니다.

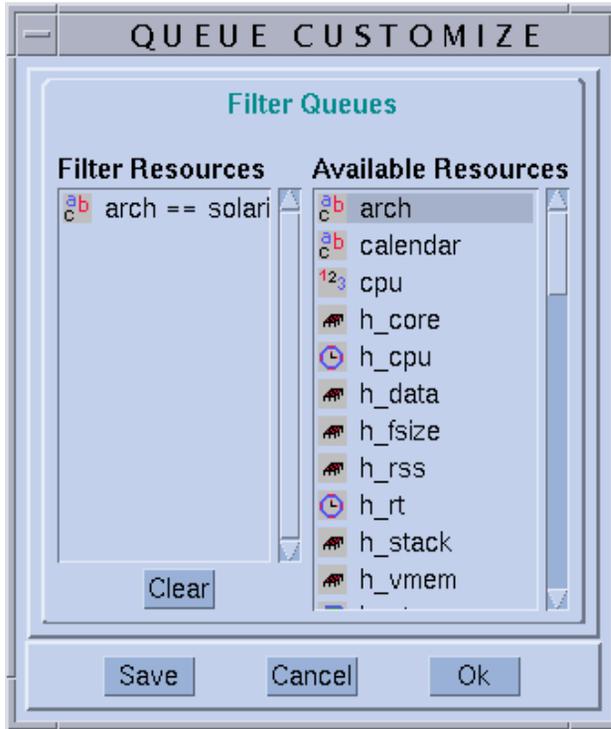


그림 5-13 대기열 제어 사용자 정의

▼ qmod를 사용하여 대기열을 제어하는 방법

125 페이지, “명령줄에서 작업을 제어하는 방법” 절에서 Sun Grid Engine 명령인 qmod 를 사용하여 Sun Grid Engine 작업을 일시중단/일시중단 해제하는 방법을 설명했습니다. 그러나 qmod 명령은 추가로 사용자에게 대기열을 일시중단/일시중단 해제 또는 사용 안함/사용하는 방법을 제공합니다.

- 다음 절에서 자세히 설명하는 정보를 바탕으로 적당한 인수를 갖는 다음 명령을 입력하십시오.

```
% qmod 인수
```

다음 명령은 이 목적을 위해 `qmod`를 사용하는 방법의 예입니다.

```
% qmod -s 대기열이름
% qmod -us -f 대기열이름 1, 대기열이름 2
% qmod -d 대기열이름
% qmod -e 대기열이름 1, 대기열이름 2, 대기열이름 3
```

처음 두 명령은 대기열을 일시중단 또는 일시중단 해제하는 반면, 세 번째와 네 번째 명령은 대기열을 사용 안함 및 사용합니다. 두 번째 명령은 `qmod -f` 옵션을 추가로 사용하여 대응하는 `sge_execd`가 도달 불가능한 경우(예를 들어, 네트워크 문제점으로 인해) `sge_qmaster`에 상태 변경의 등록을 강제합니다.

참고 - 대기열 사용 안함/사용뿐 아니라 일시중단/일시중단 해제는 대기열 소유자, Sun Grid Engine 관리자 또는 운영자 권한이 필요합니다(65 페이지, “관리자, 운영자 및 소유자” 절을 참조하십시오).

참고 - `crontab` 또는 `at` 작업과 함께 `qmod` 명령을 사용할 수 있습니다.

QMON 사용자 정의

QMON의 모습과 느낌은 대부분 특별히 설계된 자원 파일에 의해 정의됩니다. 합리적인 기본값이 컴파일되고 샘플 자원 파일이 `<sge 루트>/qmon/Qmon`에 있습니다.

클러스터 관리가 QMON 특정 자원 정의를 표준 `.Xdefaults` 또는 `.Xresources` 파일에 포함시키거나 사이트 특정 Qmon 파일을 `XAPPLRESDIR` 같은 표준 검색 경로에 의해 참조되는 위치에 넣어서 `/usr/lib/X11/app-defaults/Qmon` 같은 표준 위치에 사이트 특정 기본값을 설치할 수 있습니다. 위의 경우 중 하나가 해당하는지 여부를 관리자에게 문의하십시오.

또한 사용자가 Qmon 파일을 홈 디렉토리(또는 개인 `XAPPLRESDIR` 검색 경로에 의해 지시되는 다른 위치에) 복사하고 수정하거나 사용자의 개인 `.Xdefaults` 또는 `.Xresources` 파일에 필요한 자원 정의를 포함시켜서 개인 환경 설정을 구성할 수 있습니다. qmon 개인 자원 파일은 또한 운영 중에 `xrdb` 명령을 통해 또는 X11 환경의 시작시에(예를 들어 `.xinitrc` 자원 파일에) 설치될 수 있습니다.

가능한 사용자 정의에 대한 자세한 정보는 샘플 Qmon 파일의 명령줄을 참조하십시오.

QMON을 사용자 정의하는 또 다른 방법이 그림 5-2 및 그림 5-13에 표시된 Job Control 및 Queue Control Customization 대화 상자에 대해 설명되었습니다. 두 대화 상자 모두에서 Save 단추를 사용하여 사용자 정의 대화 상자로 구성된 필터링 및 표시장치 정의를 사용자의 홈 디렉토리에 있는 `.qmon_preferences` 파일에 저장할 수 있습니다. 재시작할 때 QMON은 이 파일을 읽고 이전에 정의된 작동을 재활성화합니다.

IV 부 관리

*Sun Grid Engine 5.3 관리 및 사용 설명서*의 이 부분은 관리자를 위한 것으로 6개의 장을 포함합니다.

■ **제 6장** – 135 페이지, “호스트 및 클러스터 구성”

이 장에서는 Sun Grid Engine 5.3 호스트 및 클러스터 구성에 대한 일반 백그라운드 및 상세한 지침을 제공합니다.

■ **제 7장** – 157 페이지, “대기열 구성 및 대기열 캘린더”

이 장에는 Sun Grid Engine 5.3 작업의 여러 범주에 대한 “컨테이너”로서 기능하는 *대기열*이라는 중요한 개념의 설명이 들어 있습니다. 이 장에는 대기열 구성에 대한 전체 지침이 포함됩니다.

■ **제 8장** – 175 페이지, “컴플렉스 개념”

이 장에서는 Sun Grid Engine 5.3 시스템이 *컴플렉스*를 사용하여 사용자가 작업에 대해 요청할 수 있는 자원 속성에 관한 모든 관련 정보를 정의하는 방법에 대해 설명합니다. 관리자는 환경의 요구사항과 일치할 다양한 *컴플렉스*를 구성하며, 이 장은 그를 위한 상세한 지침을 제공합니다.

■ **제 9장** – 203 페이지, “사용자 액세스 및 정책 관리”

이 장은 Sun Grid Engine 5.3 시스템을 통해 사용할 수 있는 사용자 정책의 유형에 관한 전체 백그라운드 정보를 제공하고 연산 환경에 이들 정책을 일치시키는 방법에 대한 지침을 제공합니다.

■ **제 10장** – 233 페이지, “병렬 환경 관리”

Sun Grid Engine 5.3 시스템이 *병렬 환경*과 조화되는 방법을 설명하는 것 외에도, 이 장은 병렬 환경을 다루는 전체 구성 지침을 제공합니다.

■ **제 11장** – 245 페이지, “오류 메시지”

이 장에서는 오류 메시지 검색을 위한 Sun Grid Engine 5.3 절차를 설명하고 디버그 모드에서 소프트웨어를 실행하는 방법에 대해 설명합니다.

호스트 및 클러스터 구성

이 장에서는 Sun Grid Engine 5.3 시스템의 여러 측면의 구성에 대한 백그라운드 정보와 지침을 제공합니다. 이 장에서 다음 업무에 대한 지침을 찾을 수 있습니다.

- 138 페이지, “QMON을 사용한 관리 호스트 구성 방법”
- 140 페이지, “관리 호스트 삭제 방법”
- 140 페이지, “관리 호스트 추가 방법”
- 140 페이지, “명령줄에서 관리 호스트를 구성하는 방법”
- 140 페이지, “QMON을 사용한 제출 호스트 구성”
- 142 페이지, “제출 호스트 삭제 방법”
- 142 페이지, “제출 호스트 추가 방법”
- 142 페이지, “명령줄에서 제출 호스트를 구성하는 방법”
- 142 페이지, “QMON을 사용하여 실행 호스트를 구성하는 방법”
- 144 페이지, “실행 호스트를 삭제하는 방법”
- 144 페이지, “실행 호스트 데몬을 종료하는 방법”
- 144 페이지, “실행 호스트를 추가 또는 수정하는 방법”
- 148 페이지, “명령줄에서 실행 호스트를 구성하는 방법”
- 149 페이지, “ghost로 실행 호스트를 모니터링하는 방법”
- 150 페이지, “명령줄에서 데몬을 종료하는 방법”
- 151 페이지, “명령줄에서 데몬을 재시작하는 방법”
- 152 페이지, “명령줄에서 기본 클러스터 구성을 표시하는 방법”
- 152 페이지, “명령줄에서 기본 클러스터 구성을 수정하는 방법”
- 152 페이지, “QMON으로 클러스터 구성을 표시하는 방법”
- 153 페이지, “QMON으로 클러스터 구성을 삭제하는 방법”
- 153 페이지, “QMON으로 글로벌 클러스터 구성을 표시하는 방법”
- 154 페이지, “글로벌 및 호스트 구성을 수정하기 위해 QMON을 사용하는 방법”

마스터 및 새도우 마스터 구성 정보

새도우 마스터 호스트 이름 파일인 `<sgerule>/<셀>/common/shadow_masters`에는 1차 마스터 호스트(Sun Grid Engine 마스터 데몬 `sgemaster`가 초기에 실행 중인 시스템) 및 *새도우 마스터* 호스트의 이름이 들어 있습니다. 마스터 호스트 이름 파일의 형식은 다음과 같습니다.

- 파일의 첫 행은 1차 마스터 호스트를 정의합니다.
- 그 뒤의 행들은 행마다 하나씩 새도우 마스터 호스트를 지정합니다.

(새도우)마스터 호스트의 나열 순서가 중요합니다. 1차 마스터 호스트(파일의 첫 번째 행)가 계속하는 데 실패하는 경우 두 번째 행에 정의된 새도우 마스터가 인계합니다. 이 마스터도 실패하는 경우 세 번째 행에 정의되는 호스트가 임무를 인계하며, 같은 방식으로 진행합니다.

호스트를 Sun Grid Engine 새도우 마스터로 준비하려면 다음 요구사항이 만족되어야 합니다.

- 새도우 마스터 호스트는 `sgeshadowd`를 실행해야 합니다.
- 새도우 마스터 호스트는 `sgemaster`의 상태 정보, 디스크에 기록된 작업 및 대기열 구성을 공유해야 합니다. 특히 (새도우)마스터 호스트는 마스터의 스펙 디렉토리 및 디렉토리 `<sgerule>/<셀>/common`에 대한 읽기/쓰기 루트 액세스가 필요합니다.
- 새도우 마스터 호스트이름 파일은 호스트를 새도우 마스터 호스트로 정의하는 행을 포함해야 합니다.

이들 요구사항이 만족되자마자, 새도우 마스터 호스트 기능이 이 호스트에 대해 활성화됩니다. 기능을 활성화하기 위해 Sun Grid Engine 데몬을 재시작할 필요는 없습니다.

새도우 마스터 호스트에서 `sgemaster`의 자동 장애 조치는 약간의 시간(1분 정도)이 소요됩니다. 한편 사용자는 Sun Grid Engine 명령이 실행될 때마다 대응하는 오류 메시지를 수신합니다.

참고 - `<sgerule>/<셀>/common/act_qmaster` 파일에는 실제로 `sgemaster` 데몬을 실행 중인 호스트의 이름이 들어 있습니다.

새도우 `sgemaster`를 시작하려면 반드시 Sun Grid Engine의 *이전* `sgemaster`가 이미 종료됐거나, 방금 시작된 새도우 `sgemaster`를 방해하는 조치 없이 종료되어야 합니다. 아주 가끔은 이것이 불가능합니다. 이 경우 대응하는 오류 메시지가 새도우 마스터 호스트의 `sgeshadowd`의 메시지 로그 파일에 기록되고(11장, 245 페이지, “오류 메시지” 참조) `sgemaster` 데몬에 `tcp`를 연결하려는 모든 시도가 완전히 실패할 것입니다. 이것이 발생하는 경우 실행 중인 마스터 데몬이 없는지 확인하고 어떤 새도우 마스터 시스템에서든지 `sgemaster`를 수동으로 재시작하십시오(150 페이지, “명령 줄에서 데몬을 종료하는 방법” 절 참조).

데몬 및 호스트 정보

Sun Grid Engine 호스트는 시스템에서 실행 중인 데몬과 호스트가 다음에 등록되는 방법에 따라서 4 그룹으로 분류됩니다: `sge_qmaster`

- **마스터 호스트** - 마스터 호스트는 전체 클러스터 활동에 대한 중심입니다. 이 호스트는 마스터 데몬 `sge_qmaster`를 실행합니다. `sge_qmaster`는 대기열 및 작업과 같은 모든 Sun Grid Engine 구성요소를 제어하며 구성요소의 상태, 사용자 액세스 권한 등에 관한 표를 유지보수합니다. 31 페이지, “마스터 호스트 설치 방법” 절이 마스터 호스트를 초기에 설정하는 방법을 설명하고 136 페이지, “마스터 및 새도우 마스터 구성 정보” 절은 동적 마스터 호스트 변경사항을 구성할 수 있는 방법을 표시합니다. 마스터 호스트는 대개 Sun Grid Engine 스케줄러 `sge_schedd`를 실행합니다. 마스터 호스트는 설치 절차에 의해 수행되는 것 외에는 추가 구성이 필요 없습니다.
- **실행 호스트** - 실행 호스트는 Sun Grid Engine 작업을 실행할 권한을 갖는 노드입니다. 그러므로 이 호스트는 Sun Grid Engine 대기열을 호스트하고 있으며 Sun Grid Engine 실행 데몬 `sge_execd`를 실행합니다. 실행 호스트는 초기에 32 페이지, “실행 호스트 설치 방법” 절에서 설명하는 것처럼 실행 호스트 설치 절차에 의해 설정됩니다.
- **관리 호스트** - 마스터 호스트 이외의 다른 호스트에 Sun Grid Engine에서 모든 종류의 관리 활동을 수행할 권한을 부여할 수 있습니다. 관리 호스트는 다음 명령으로 설정됩니다.

```
qconf -ah 호스트이름
```

자세한 내용은 `qconf` 매뉴얼 페이지를 참조하십시오.

- **제출 호스트** - 제출 호스트는 일괄처리 작업의 제출 및 제어만을 허용합니다. 특히 제출 호스트에 로그인하는 사용자는 `qsub`를 통해 작업을 제출하거나, `qstat`를 통해 작업 상태를 제어하거나, Sun Grid Engine의 OSF/1 Motif 그래픽 사용자 인터페이스인 `QMON`을 사용할 수 있습니다. 제출 호스트는 다음 명령으로 설정됩니다.

```
qconf -as 호스트이름
```

자세한 내용은 `qconf` 매뉴얼 페이지를 참조하십시오.

참고 - 한 호스트가 위에 설명한 클래스의 둘 이상에 속할 수 있습니다. 마스터 호스트는 기본적으로 관리 및 제출 호스트입니다.

호스트 구성 정보

Sun Grid Engine은 마스터 호스트를 제외한 모든 유형의 호스트에 대한 오브젝트 목록을 유지보수합니다. 관리 및 제출 호스트의 경우에 이들 목록은 단순히 호스트가 관리 또는 제출 권한을 갖는지 여부에 대한 정보를 제공합니다. 실행 호스트 오브젝트의 경우에는 Sun Grid Engine 관리자가 제공해야 하는 부하 매개변수 조정 인자뿐 아니라 호스트에서 실행 중인 `sge_execd`가 보고하는 부하 정보 같은 추가 정보가 목록에 저장됩니다.

다음 절은 Sun Grid Engine 그래픽 사용자 인터페이스인 QMON을 사용하여 및 명령줄에서 다른 호스트 오브젝트를 구성하는 방법을 설명합니다.

GUI 관리는 QMON 기본 메뉴의 Host Config 아이콘 단추를 눌러서 호출되는 호스트 구성 대화 상자 세트에 의해 제공됩니다. 사용할 수 있는 대화 상자는 Administration Host Configuration(그림 6-1 참조), Submit Host Configuration(그림 6-2 참조) 및 Execution Host Configuration(그림 6-3 참조)입니다. 화면의 맨 위에 있는 선택 목록 단추를 사용하여 대화 상자를 전환할 수 있습니다.

`qconf` 명령은 호스트 오브젝트 관리를 위한 명령줄 인터페이스를 제공합니다.

▼ QMON을 사용한 관리 호스트 구성 방법

1. QMON 기본 메뉴의 맨 위에 있는 **Administration Host** 탭을 누릅니다.

다음 그림과 유사한 Administration Host Configuration 대화 상자가 열립니다.

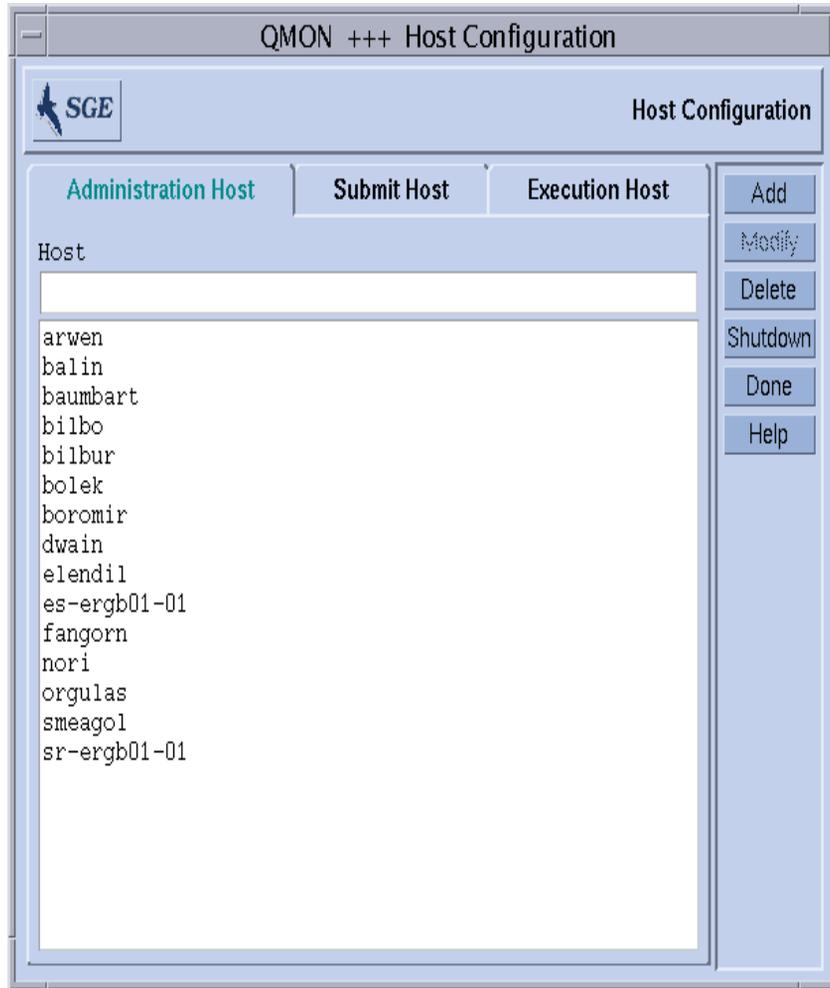


그림 6-1 Administration Host Configuration 대화 상자

참고 - Administration Host Configuration 대화 상자는 Host Config 단추를 처음으로 누를 때 기본적으로 열립니다.

2. 호스트를 구성하려는 방법에 따라서 다음 절에 있는 지침에 따라 진행합니다.

이 대화 상자를 사용하여 관리 Sun Grid Engine 명령이 허용되는 호스트를 구성할 수 있습니다. 화면의 중앙에 있는 선택 목록은 관리 권한을 제공하기 위해 이미 선언된 호스트를 표시합니다.

▼ 관리 호스트 삭제 방법

- 왼쪽 마우스 단추로 호스트의 이름을 누른 후 대화 상자의 맨 아래에 있는 **Delete** 단추를 눌러서 이 목록에서 기존 호스트를 삭제하십시오.

▼ 관리 호스트 추가 방법

- **Hostname** 입력 창에 이름을 입력한 후 **Add** 단추나 **Return** 키를 눌러서 새 호스트를 입력하십시오.

▼ 명령줄에서 관리 호스트를 구성하는 방법

- 호스트를 구성하려는 방법에 따라서 적당한 인수와 함께 다음 명령을 입력하십시오.

```
% qconf 인수
```

qconf 명령에 대한 인수와 그의 결과는 다음과 같습니다.

- **qconf -ah 호스트이름**

관리 호스트 추가—관리 호스트의 목록에 지정된 호스트를 추가합니다.

- **qconf -dh 호스트이름**

관리 호스트 삭제—관리 호스트 목록에서 지정된 호스트를 삭제합니다.

- **qconf -sh**

관리 호스트 표시—모든 현재 구성된 관리 호스트의 목록을 표시합니다.

▼ QMON을 사용한 제출 호스트 구성

1. QMON 기본 메뉴의 맨 위에 있는 **Submit Host** 탭을 누릅니다.

다음 그림과 유사한 Submit Host Configuration 대화 상자가 열립니다.

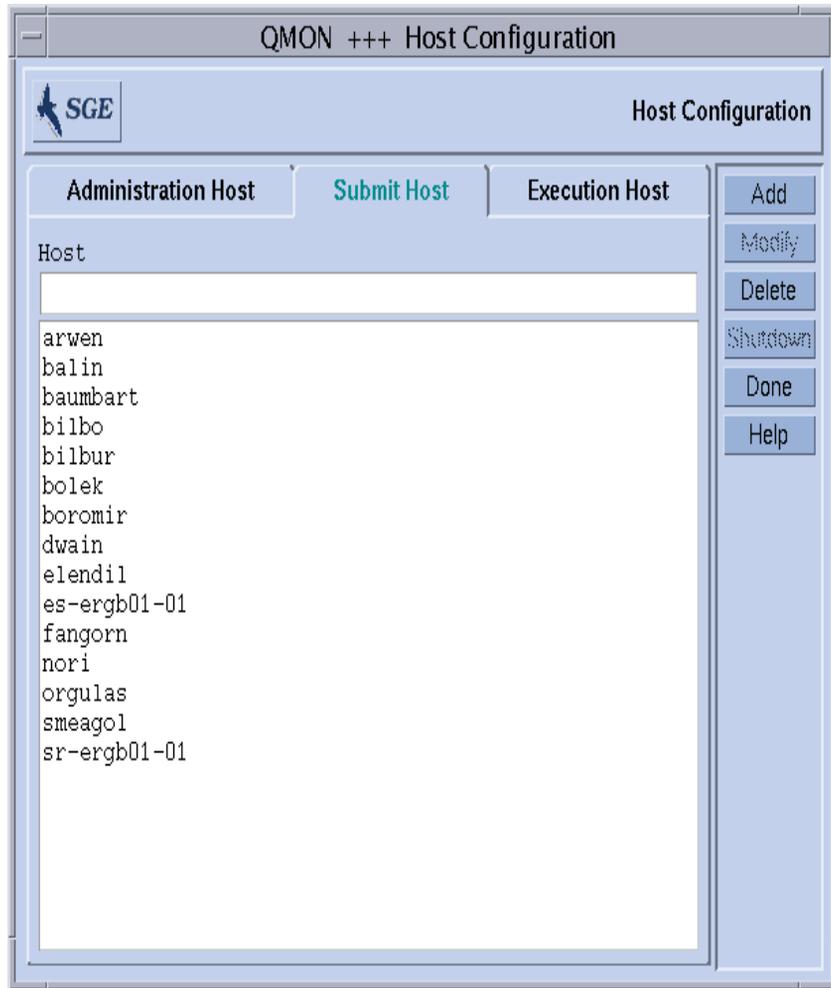


그림 6-2 제출 호스트 구성

2. 호스트를 구성하려는 방법에 따라서 다음 절에 있는 지침에 따라 진행합니다.

이 대화 상자를 사용하여 작업을 제출, 모니터 및 제어할 수 있는 호스트를 선언할 수 있습니다. 이들 호스트가 또한 관리 호스트라고 선언되지 않으면 이들 호스트에서 관리 Sun Grid Engine 명령은 허용되지 않습니다(138 페이지, “QMON을 사용한 관리 호스트 구성 방법” 참조). 화면의 중앙에 있는 선택 목록은 제출 권한을 제공하기 위해 이미 선언된 호스트를 표시합니다.

▼ 제출 호스트 삭제 방법

- **Submit Host** 대화 상자에서 이름을 누른 후 대화 상자의 맨 아래에 있는 **Delete** 단추를 눌러서 기존 호스트를 삭제하십시오.

▼ 제출 호스트 추가 방법

- **Submit Host** 대화 상자의 **Hostname** 입력 창에 이름을 입력한 후 **Add** 단추나 **Return** 키를 눌러서 호스트를 추가하십시오.

▼ 명령줄에서 제출 호스트를 구성하는 방법

- 호스트를 구성하려는 방법에 따라서 적당한 인수와 함께 다음 명령을 입력하십시오.

```
% qconf 인수
```

qconf 명령에 대한 인수와 그의 결과는 다음과 같습니다.

- **qconf -as 호스트이름**

제출 호스트 추가—제출 호스트의 목록에 지정된 호스트를 추가합니다.

- **qconf -ds 호스트이름**

제출 호스트 삭제—제출 호스트 목록에서 지정된 호스트를 삭제합니다.

- **qconf -ss**

제출 호스트 표시—제출 권한을 제공하도록 현재 구성된 모든 호스트의 이름 목록을 표시합니다.

▼ QMON을 사용하여 실행 호스트를 구성하는 방법

1. QMON 기본 메뉴의 맨 위에 있는 **Execution Host** 탭을 누릅니다.

다음 그림과 유사한 Execution Host Configuration 대화 상자가 열립니다.

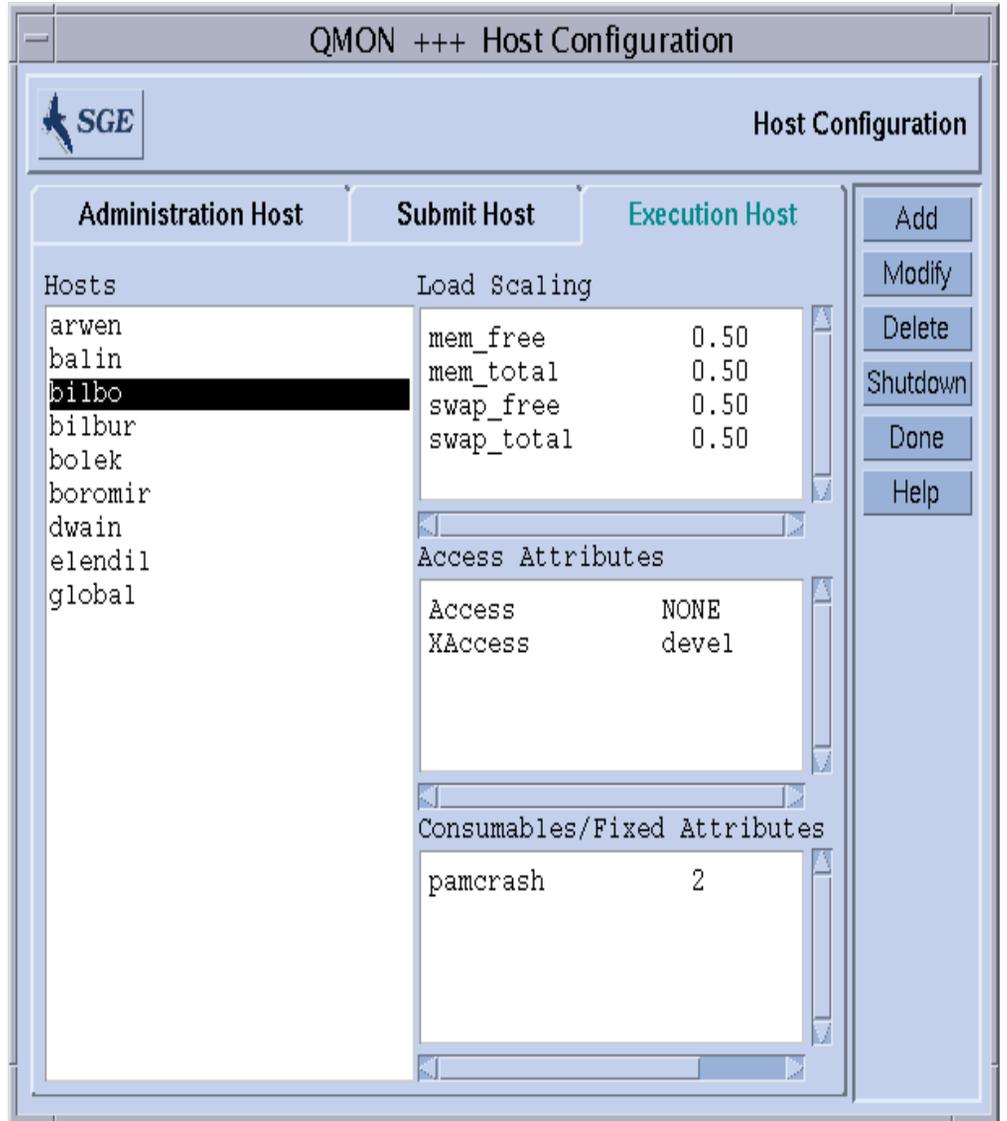


그림 6-3 실행 호스트 구성

2. 호스트를 구성하려는 방법에 따라서 다음 절에 있는 지침에 따라 진행합니다.

이 대화 상자에서 Sun Grid Engine 실행 호스트를 구성할 수 있습니다. 이들 호스트가 또한 관리 또는 제출 호스트라고 선언되지 않으면 이들 호스트에서 관리 또는 제출 명령은 자동으로 허용되지 않습니다(138 페이지, “QMON을 사용한 관리 호스트 구성 방법” 및 140 페이지, “QMON을 사용한 제출 호스트 구성” 참조).

호스트 선택 목록이 이미 정의된 실행 호스트를 표시합니다. 현재 구성된 부하 조정 인자, 호스트와 연관된 소비 가능 및 고정 콤플렉스 속성에 대한 액세스 권한 및 자원 가용성이 선택된 실행 호스트에 대한 Load Scaling, Access Attributes 및 Consumable/Fixed Attributes 표시 창에 표시됩니다. 콤플렉스 속성, 사용자 액세스 권한 및 부하 매개변수에 대한 자세한 내용은 175 페이지, “콤플렉스 관련 정보”, 64 페이지, “사용자 액세스 사용 권한” 및 197 페이지, “부하 매개변수” 절을 참조하십시오.

▼ 실행 호스트를 삭제하는 방법

- Execution Host 대화 상자에서 삭제할 실행 호스트의 이름을 누른 후 대화 상자의 오른쪽에 있는 맨 아래 열에서 Delete 단추를 누르십시오.

▼ 실행 호스트 데몬을 종료하는 방법

- 임의의 선택된 호스트에 대해 Execution Host 대화 상자의 Shutdown 단추를 누르십시오.

▼ 실행 호스트를 추가 또는 수정하는 방법

1. Execution Host 대화 상자의 맨 아래 열에 있는 Add 또는 Modify 단추를 누릅니다.

그림 6-4에 표시된 것과 유사한 대화 상자가 나타납니다.

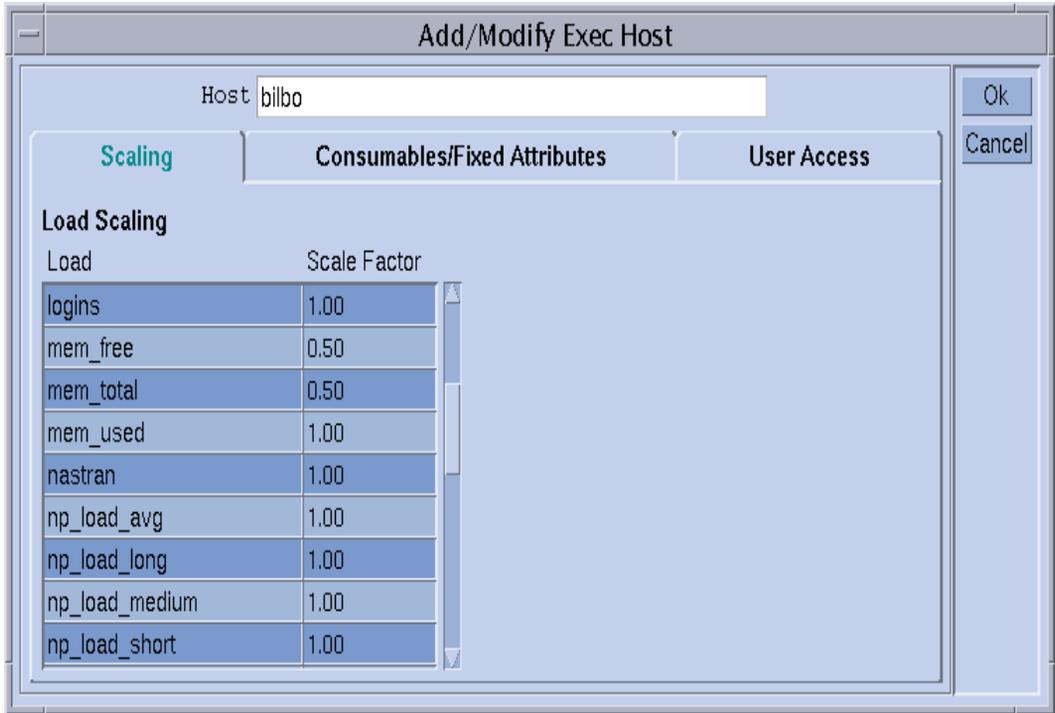


그림 6-4 부하 조정 수정

2. 호스트를 수정하려는 방법에 따라서 다음 절에 있는 지침에 따라 진행합니다.

새 실행 호스트를 추가하거나 기존 호스트의 구성을 수정하기 위한 대화 상자는 호스트와 연관된 모든 속성의 수정을 허용합니다. 실행 호스트의 이름이 표시되거나 Host 입력창에 추가할 수 있습니다. 대화 상자의 Scaling 탭을 선택하여 조정 인자를 정의할 수 있습니다(그림 6-4 참조).

모든 사용 가능한 부하 매개변수가 Load Scaling 표의 Load 열에 표시되고 Scale Factor 열에서 조정의 대응하는 정의를 찾을 수 있습니다. Scale Factor 열은 편집할 수 있습니다. 유효한 조정 인자는 고정점 또는 공학용 표기법으로 된 양의 부동 소수점 숫자입니다.

Consumables/Fixed Attributes를 탭 도구에서 선택하는 경우 호스트와 연관된 콤플렉스 속성을 정의할 수 있습니다(그림 6-5 참조). 호스트와 연관된 콤플렉스(175 페이지, “콤플렉스 관련 정보” 절 참조)는 글로블이며 대화 상자의 왼쪽 맨 아래에 있는 Complex Selection 영역을 통해 호스트에 추가되는 호스트 콤플렉스 또는 관리자 정의 콤플렉스입니다. 사용 가능한 관리자 정의 콤플렉스가 왼쪽에 표시되고 빨간색 화살표를 통해 추가 또는 제거할 수 있습니다. Complex Configuration 아이콘 단추는 현재 콤플렉스 구성에 대한 추가 정보가 필요한 경우 또는 그를 수정하려는 경우에 최상위 Complex Configuration 대화 상자를 엽니다.

대화 상자의 오른쪽 맨 아래 영역에 있는 Consumable/Fixed Attributes 탭은 값이 현재 정의되는 모든 콤플렉스 속성을 나열합니다. 맨 위에 있는 Name 또는 Value 단추를 눌러서 목록을 향상시킬 수 있습니다. 그러면 호스트에 추가된 모든 속성(즉, 글로벌로 구성된 모든 속성의 집합, 위에서 설명한 것처럼 이 호스트에 추가된 호스트 및 관리자 정의 콤플렉스)을 갖는 선택 목록이 열립니다. Attribute Selection 대화 상자는 그림 6-6에 표시되어 있습니다. 속성 중 하나를 선택하고 Ok 단추로 선택을 확정하면 해당 속성이 Consumable/Fixed Attributes 표의 Name 열에 추가되고 포인터가 대응하는 Value 필드에 놓입니다. Value 필드를 왼쪽 마우스 단추로 두 번 눌러서 기존 값을 수정할 수 있습니다. 속성 삭제는 먼저 왼쪽 마우스 단추로 대응하는 표 행을 선택하여 수행됩니다. CTRL-D를 입력하거나 오른쪽 마우스 단추를 눌러서 삭제 상자를 열고 삭제를 확정하여 선택한 목록 항목을 삭제할 수 있습니다.

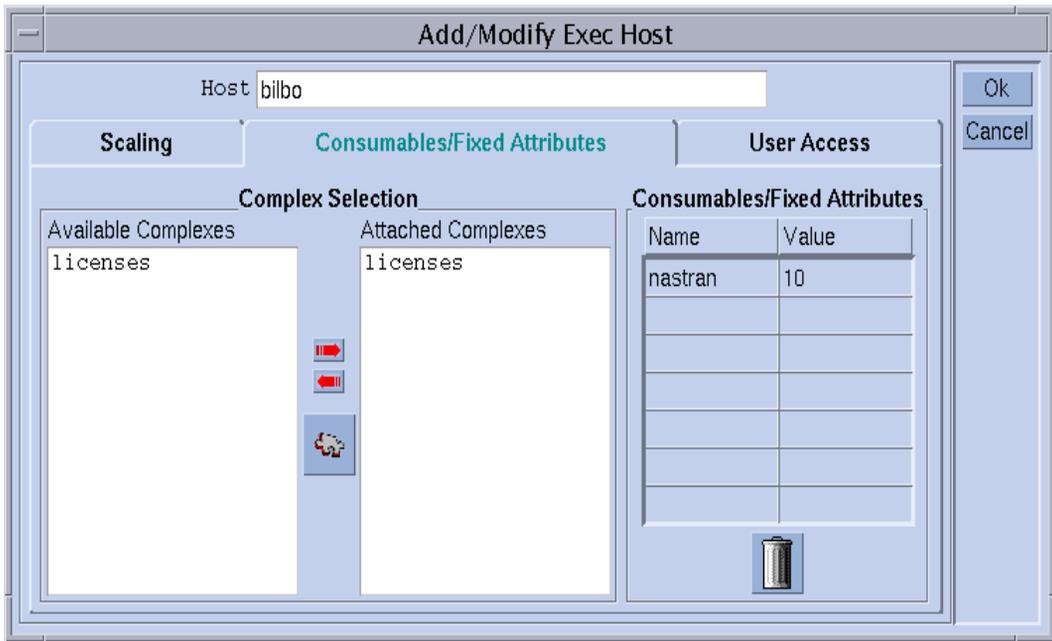


그림 6-5 Consumable/Fixed Attributes 수정

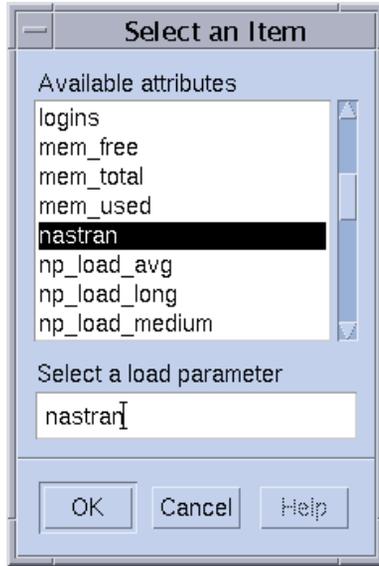


그림 6-6 사용 가능한 콤플렉스 속성

User Access 탭(그림 6-7)을 선택하여 이전에 구성된 사용자 액세스 목록을 바탕으로 실행 호스트에 대한 액세스 권한을 정의할 수 있습니다.

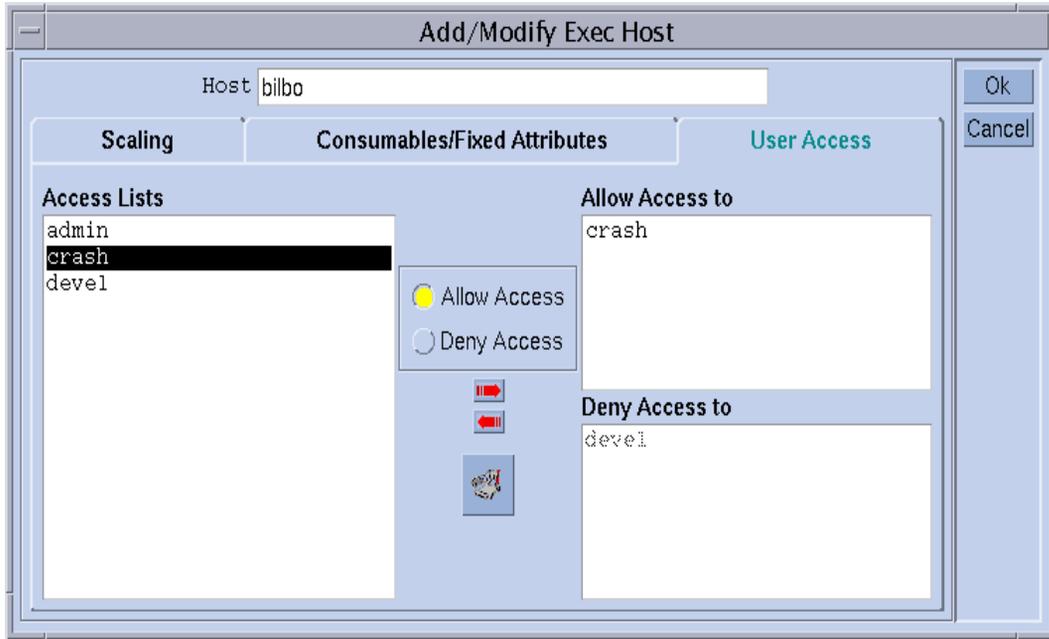


그림 6-7 User Access 수정

▼ 명령줄에서 실행 호스트를 구성하는 방법

- 호스트를 구성하려는 방법에 따라서 적당한 인수와 함께 다음 명령을 입력하십시오.

```
% qconf 인수
```

실행 호스트 목록 유지보수를 위한 명령줄 인터페이스는 qconf 명령에 대한 다음 옵션에 의해 제공됩니다.

■ qconf -ae [실행호스트템플리트]

실행 호스트 추가—실행 호스트 구성 템플리트를 갖는 편집기(기본 vi 또는 \$EDITOR 환경 변수에 따라서)를 표시합니다. 선택적 매개변수 실행호스트템플리트(이미 구성된 실행 호스트의 이름)가 존재하는 경우 이 실행 호스트의 구성이 템플리트로 사용됩니다. 실행 호스트는 템플리트를 변경하고 디스크에 저장하여 구성됩니다. 변경할 템플리트 항목의 자세한 설명은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 host_conf 항목을 참조하십시오.

■ `qconf -de 호스트이름`

실행 호스트 삭제—실행 호스트 목록에서 지정된 호스트를 삭제합니다. 실행 호스트 구성의 모든 항목이 유실됩니다.

■ `qconf -me 호스트이름`

실행 호스트 수정—템플릿로서 지정된 실행 호스트의 구성을 갖는 편집기(`vi` 또는 `$EDITOR` 환경 변수에 따라서)를 표시합니다. 실행 호스트 구성은 템플릿을 변경하고 디스크에 저장하여 수정됩니다. 변경할 템플릿 항목의 자세한 설명은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `host_conf` 매뉴얼 페이지를 참조하십시오.

■ `qconf -Me 파일이름`

실행 호스트 수정—`파일이름`의 내용을 실행 호스트 구성 템플릿으로 사용합니다. 지정된 파일의 구성이 기존 실행 호스트를 참조해야 합니다. 이 실행 호스트의 구성이 파일 내용에 의해 대체됩니다. 이 `qconf` 옵션은 수동 상호작용이 필요 없기 때문에 오프라인 실행 호스트 변경, 예를 들면 `cron` 작업에 유용합니다.

■ `qconf -se 호스트이름`

실행 호스트 표시—`host_conf`에 정의된 대로 지정된 실행 호스트의 구성을 표시합니다.

■ `qconf -sel`

실행 호스트 목록 표시—실행 호스트가 되도록 구성되는 호스트 이름의 목록을 표시합니다.

▼ qghost로 실행 호스트를 모니터링하는 방법

`qghost` 명령은 실행 호스트 상태의 빠른 개요를 검색하는 편리한 방법을 제공합니다.

- 다음 명령을 입력합니다.

```
% qghost
```

명령이 다음과 유사한 출력을 생성합니다.

표 6-1 샘플 qhost 출력

HOSTNAME	ARCH	NPROC	LOAD	MEMTOT	MEMUSE	SWAPTO	SWAPUS
global	-	-	-	-	-	-	-
BALROG.genias.de	solaris6	2	0.38	1.0G	994M	900M	891M
BILBUR.genias.de	solaris	1	0.18	96M	70M	164M	9.0M
DWAIN.genias.de	irix6	1	1.13	149M	55M	40M	0.0
GLOIN.genias.de	osf4	2	0.05	768M	701M	1.9G	13.5M
SPEEDY.genias.de	alinux	1	0.08	248M	60.6M	125M	232.0K
SARUMAN.genias.de	solaris	1	0.11	96M	77M	192M	9.0M
FANGORN.genias.de	linux	1	2.01	124.8M	49.9M	127.7M	4.3M

출력 형식의 설명과 추가 옵션에 대해서는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 qhost 항목을 참조하십시오.

▼ 명령줄에서 데몬을 종료하는 방법

- 다음 명령 중 하나를 사용하십시오. 이들 조작을 위해서는 Sun Grid Engine 관리자 또는 운영자 특권이 필요함을 주의하십시오(9장, 203 페이지, “사용자 액세스 및 정책 관리” 참조).

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

- 첫 번째 명령은 현재 활동하는 모든 작업을 종료하고 모든 Sun Grid Engine 실행 데몬을 정지시킵니다.

참고 - 해당 명령을 qconf -ke에 의해 대체하는 경우 Sun Grid Engine 실행 데몬이 중단되지만 활동 작업은 취소되지 않습니다. 해당 시스템에 실행 중인 sge_execd가 없는 동안 종료하는 작업은 sge_execd가 다시 재시작될 때까지 sge_qmaster에 보고되지 않습니다. 그러나 작업 보고서는 유실되지 않습니다.

- 두 번째 명령은 Sun Grid Engine 스케줄러 sge_schedd를 종료합니다.
- 세 번째 명령은 sge_qmaster 프로세스를 강제로 종료합니다.

실행 중인 작업이 있고 현재 활동 작업이 종료될 때까지 Sun Grid Engine의 종료 절차가 기다리기 원하는 경우 위에서 설명한 qconf 순서를 실행하기 전에 각 대기열에 대해 아래 명령을 사용할 수 있습니다.

```
% qmod -d 대기열이름
```

qmod 사용 안함 명령은 새 작업이 사용되지 않는 대기열에 스케줄되지 않게 합니다. 그런 다음 대기열에서 더 이상 작업이 실행하지 않을 때까지 데몬의 종료를 기다려야 합니다.

▼ 명령줄에서 데몬을 재시작하는 방법

1. Sun Grid Engine 5.3 데몬이 재시작되기 원하는 시스템에 root로서 로그인합니다.
2. 다음 스크립트를 실행합니다.

```
% <sgе 루트>/< 셸>/common/rcsgе
```

이 스크립트는 이 호스트에서 정상적으로 실행 중인 데몬을 찾은 후 대응하는 데몬을 시작합니다.

기본 클러스터 구성

기본 Sun Grid Engine 클러스터 구성은 mail 또는 xterm 같은 프로그램에 대한 유효한 경로와 같은 사이트 종속성을 반영하고 Sun Grid Engine 작동에 영향을 주도록 구성되는 정보의 세트입니다. Sun Grid Engine 풀의 모든 호스트뿐 아니라 Sun Grid Engine 마스터 호스트에 대해 제공되는 글로벌 구성이 있습니다. 또한 Sun Grid Engine 시스템은 글로벌 구성의 특정 항목을 대체하기 위해 모든 호스트에 로컬인 구성을 사용하도록 구성될 수 있습니다.

Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서의 sge_conf 항목에 구성 항목의 자세한 설명이 들어 있습니다. Sun Grid Engine 클러스터 관리자는 설치 후에 사이트의 필요에 글로벌 및 로컬 구성을 직접 적응시키고 그 이후에는 구성을 최신으로 유지해야 합니다.

▼ 명령줄에서 기본 클러스터 구성을 표시하는 방법

현재 구성을 표시하는 Sun Grid Engine 명령은 `qconf` 프로그램의 구성 표시 옵션입니다. 다음은 몇 가지 예입니다(자세한 설명은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서* 참조).

- 다음 명령 중 하나를 입력하십시오.

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

처음 두 명령은 동등하며 글로벌 구성을 표시합니다. 세 번째 명령은 호스트의 로컬 구성을 표시합니다.

▼ 명령줄에서 기본 클러스터 구성을 수정하는 방법

참고 - 클러스터 구성을 변경하는 Sun Grid Engine 명령(`qconf`)은 Sun Grid Engine 관리자만이 사용할 수 있습니다.

- 다음 명령 중 하나를 입력하십시오.

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

- 첫 번째 명령 예는 글로벌 구성을 수정합니다.
- 두 번째 예는 지정된 실행 또는 마스터 호스트의 로컬 구성에 대해 동작합니다.

위 두 명령은 사용할 수 있는 많은 `qconf` 명령의 예입니다. 다른 명령에 대해서는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*를 참조하십시오.

▼ QMON으로 클러스터 구성을 표시하는 방법

1. QMON Main 메뉴에서 **Cluster Configuration** 단추를 누릅니다.

그림 6-8에 표시된 것과 비슷한 Cluster Configuration 대화 상자가 표시됩니다.

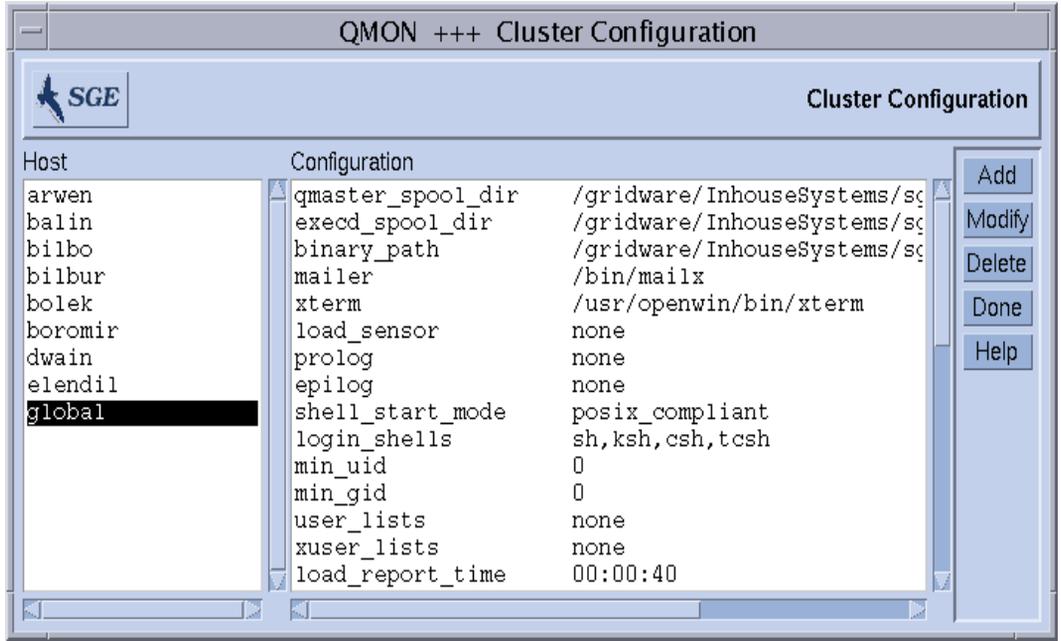


그림 6-8 Cluster Configuration 대화 상자

- 화면의 왼쪽에 있는 Host 선택 목록에서 해당 호스트에 대한 현재 구성을 표시할 호스트의 이름을 선택합니다.

▼ QMON으로 클러스터 구성을 삭제하는 방법

- QMON Main 메뉴에서 Cluster Configuration 단추를 누릅니다.
- 화면의 왼쪽에 있는 Host 선택 목록에서 구성을 삭제하려는 호스트의 이름을 선택합니다.
- Delete 단추를 누릅니다.

▼ QMON으로 글로벌 클러스터 구성을 표시하는 방법

- Host 선택 목록에서 이름 global을 선택합니다.

sgc_conf 매뉴얼 페이지에서 설명하는 형식으로 구성이 표시됩니다. 선택한 글로벌 또는 호스트 로컬 구성을 수정하려면 **Modify** 단추를 사용하십시오. 특정 호스트에 대한 새 구성을 추가하려면 **Add** 단추를 사용하십시오.

▼ 글로벌 및 호스트 구성을 수정하기 위해 QMON을 사용하는 방법

1. **Cluster Configuration** 대화 상자(152 페이지, “QMON으로 클러스터 구성을 표시하는 방법” 절에서 설명됨)에서 **Add** 단추 또는 **Modify** 단추를 누릅니다.

그림 6-9의 보기와 유사한 Cluster Settings 대화 상자가 열립니다.

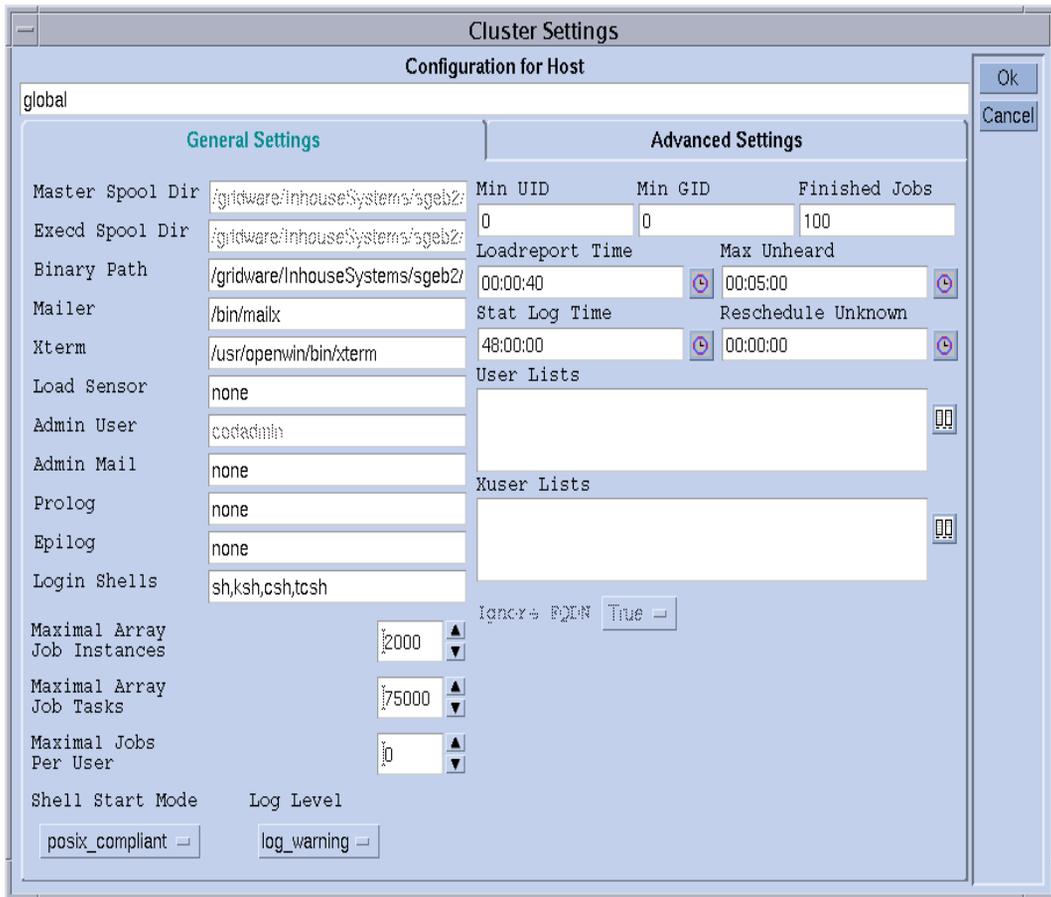


그림 6-9 Cluster Settings 대화 상자—General Settings

2. 다음 절에서 자세히 설명되는 정보를 바탕으로 원하는 변경을 수행합니다.

Cluster Settings 대화 상자는 글로벌 또는 호스트 로컬 구성의 모든 매개변수를 변경하는 수단을 제공합니다. 모든 항목 필드는 글로벌 구성이 변경되는 경우, 즉 사용자가 호스트 global을 선택했고 Modify를 누른 경우에만 액세스할 수 있습니다. 정규 호스트가 수정되는 경우 그의 실제 구성이 대화 상자에 반영되고 호스트 로컬 변경에 적합한 매개변수만을 수정할 수 있습니다. 새 호스트 로컬 구성이 추가되는 경우 대화 상자 항목은 빈 필드입니다.

Advanced Settings 탭(그림 6-10)은 글로벌, 호스트 로컬 또는 새 구성이 변경되는지 여부에 따라서 대응하는 작동을 표시합니다. 이 탭은 훨씬 드물게 사용되는 클러스터 구성 매개변수에 대한 액세스를 제공합니다.

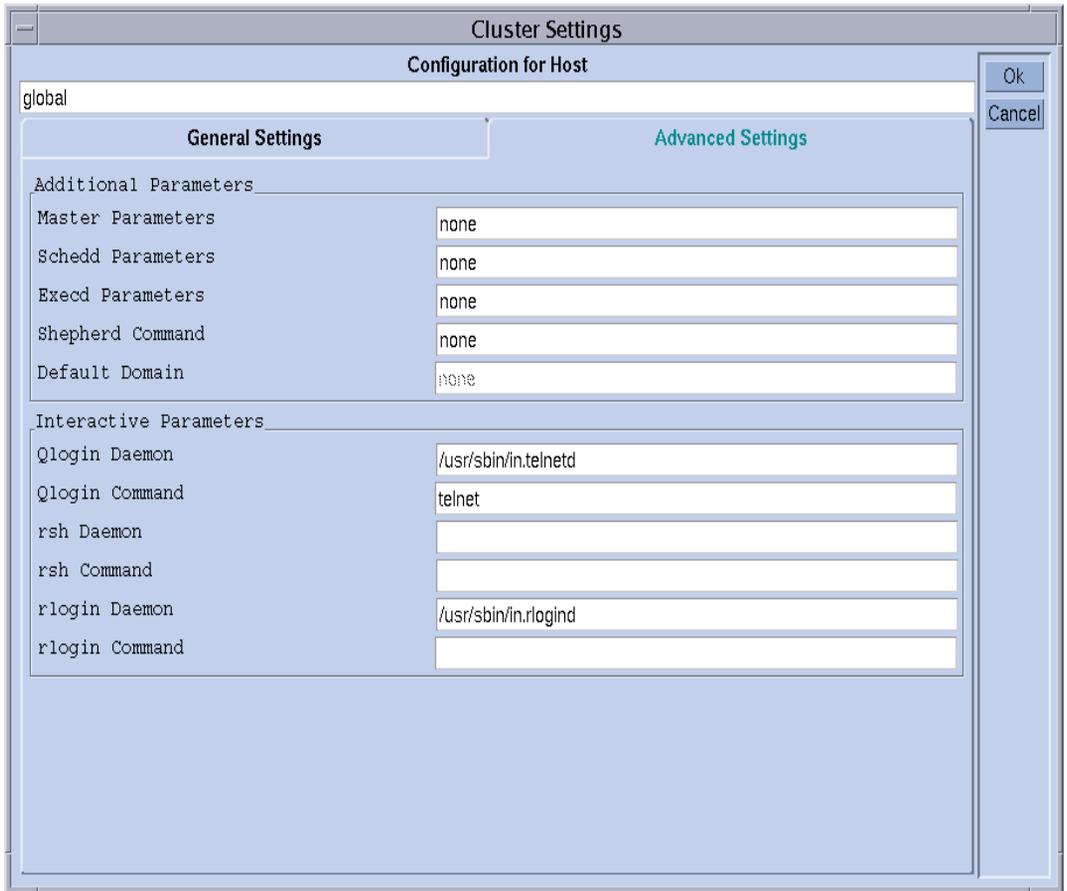


그림 6-10 Cluster Settings 대화 상자—Advanced Settings

수정을 마친 후 오른쪽 상단 모서리의 Ok 단추가 수정된 구성을 등록합니다. Cancel를 누르면 모든 변경사항이 삭제됩니다. 두 경우 모두에 대화 상자가 닫힙니다.

모든 클러스터 구성 매개변수의 완전한 설명은 `sge_conf` 매뉴얼 페이지를 참조하십시오.

대기열 구성 및 대기열 캘린더

이 장에서는 Sun Grid Engine 5.3 대기열에 대한 백그라운드 정보 및 지침을 제공합니다.

다음은 이 장에 지침이 들어 있는 특정 업무의 목록입니다.

- 158 페이지, “QMON을 사용하여 대기열을 구성하는 방법”
- 159 페이지, “General 매개변수 구성 방법”
- 160 페이지, “Execution Method 매개변수 구성 방법”
- 161 페이지, “Checkpointing 매개변수 구성 방법”
- 162 페이지, “Load 및 Suspend Thresholds 구성 방법”
- 163 페이지, “Limits 구성 방법”
- 165 페이지, “User Complexes 구성 방법”
- 166 페이지, “중속 대기열 구성 방법”
- 167 페이지, “User Access 구성 방법”
- 168 페이지, “Owners 구성 방법”
- 169 페이지, “명령줄에서 대기열을 구성하는 방법”
- 171 페이지, “QMON을 사용하여 대기열 캘린더를 구성하는 방법”
- 173 페이지, “명령줄에서 캘린더를 구성하는 방법”

대기열 구성 정보

Sun Grid Engine 대기열은 작업의 여러 범주에 대한 컨테이너이며 동일한 범주에 속하는 복수 작업의 동시 실행을 위한 대응하는 자원을 제공합니다. 작업은 Sun Grid Engine 대기열에서 대기하지 않지만 작업 지정되자마자 즉시 실행을 시작합니다. Sun Grid Engine 스케줄러의 작업 보류 목록이 Sun Grid Engine 작업에 대한 유일한 대기 영역입니다.

Sun Grid Engine 대기열을 구성하면 대기열 속성이 `sge_qmaster`에 등록됩니다. 대기열은 구성되자마자 전체 클러스터 및 Sun Grid Engine 풀에 속하는 모든 호스트의 모든 Sun Grid Engine 사용자에게 즉시 보이게 됩니다.

▼ QMON을 사용하여 대기열을 구성하는 방법

1. QMON Main 메뉴에서 Queue Control 단추를 누릅니다.
2. Queue Control 대화 상자에서 Add 또는 Modify 단추를 누릅니다.

Queue Configuration 대화 상자가 열립니다. Queue Control 대화 상자 및 대기열 상태를 모니터하고 조작하는 그의 기능은 126 페이지, “QMON을 사용하여 대기열을 제어하는 방법” 절에 설명되어 있습니다. Queue Configuration 대화 상자가 처음으로 열리는 경우 일반적인 매개변수 양식을 표시합니다. (159 페이지, “General 매개변수 구성 방법” 참조)

3. 다음 절에 자세히 설명되는 정보는 바탕으로 구성을 결정합니다.

원하는 조작에 의해 영향을 받을 대기열이 화면 상단 영역에 있는 Queue 및 Hostname 창에 표시 또는 정의됩니다. 대기열이 수정될 경우 기존 대기열은 Queue Configuration 대화 상자가 열리기 전에 Queue Control 대화 상자에서 선택되어야 합니다. 새 대기열이 추가될 경우 대기열 이름 및 대기열이 상주하는 호스트가 정의되어야 합니다.

Queue Configuration 대화 상자의 사용 편의성을 증진하기 위해 Hostname 창의 바로 아래 세 개의 단추가 사용 가능합니다. 대기열 선택 목록을 통해 기존 대기열의 모든 매개변수 가져오기를 허용하는 Clone 단추, 템플릿 대기열의 구성을 로드하는 Reset 단추, Queue Configuration 대화 상자가 열린 동안 수정된 다른 오브젝트의 구성을 로드하는 Refresh 단추가 그것입니다(Refresh 단추에 관한 추가 세부사항에 대해서는 165 페이지, “User Complexes 구성 방법” 및 167 페이지, “User Access 구성 방법” 절 참조).

Queue Configuration 대화 상자의 오른쪽 상단 모서리의 Ok 단추는 sge_qmaster에 변경사항을 등록하는 반면, 아래의 Cancel 단추는 모든 변경사항을 삭제합니다. 두 단추 모두가 대화 상자를 닫습니다.

9개의 매개변수 세트가 대기열을 정의하는 데 사용 가능합니다.

- General (159 페이지, “General 매개변수 구성 방법” 참조)
- Execution Method (160 페이지, “Execution Method 매개변수 구성 방법” 참조)
- Checkpointing (161 페이지, “Checkpointing 매개변수 구성 방법” 참조)
- Load/Suspend Thresholds (162 페이지, “Load 및 Suspend Thresholds 구성 방법” 참조)
- Limits (163 페이지, “Limits 구성 방법” 참조)
- Complexes (165 페이지, “User Complexes 구성 방법” 참조)
- Subordinates (166 페이지, “종속 대기열 구성 방법” 참조)
- User Access (167 페이지, “User Access 구성 방법” 참조)
- Owners (168 페이지, “Owners 구성 방법” 참조)

Queue Parameter 탭을 통해 원하는 매개변수를 선택합니다.

▼ General 매개변수 구성 방법

- General 매개변수 세트를 선택합니다.

그림 7-1의 보기와 유사한 화면이 표시됩니다.

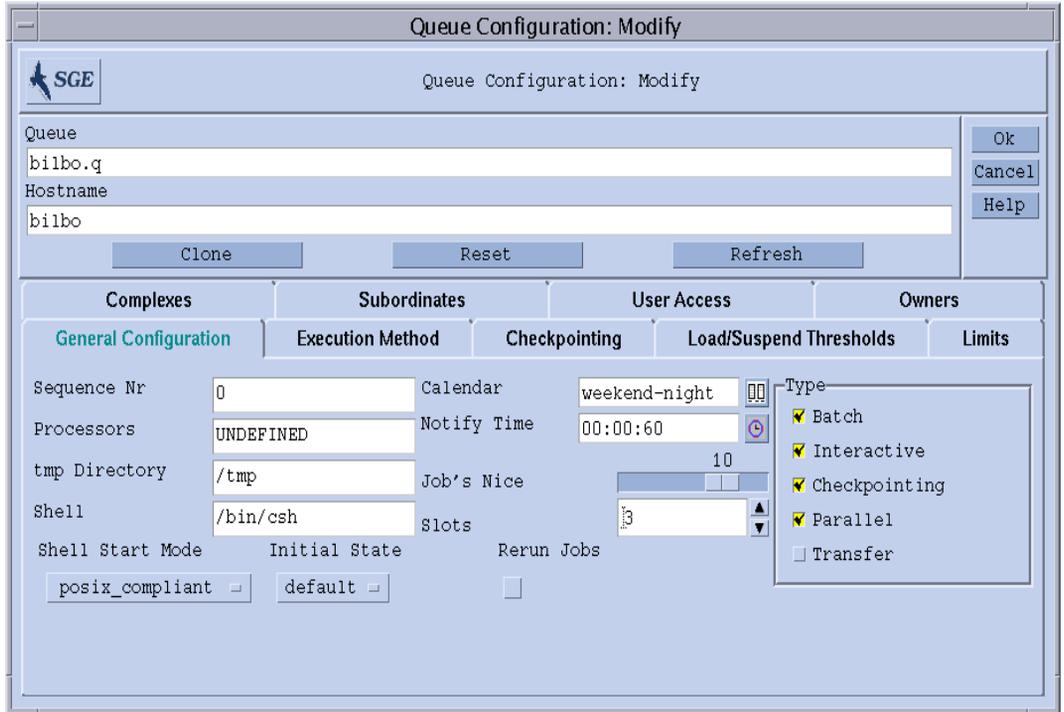


그림 7-1 Queue Configuration—General 매개변수

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 대기열의 순번.
- 프로세서—해당 대기열에서 실행 중인 작업이 사용할 프로세서에 대한 지정자. 일부 운영 체제 아키텍처의 경우 이것은 프로세서 세트의 범위(1-4,8,10 같은) 또는 정수 ID일 수 있습니다. 자세한 정보는 Sun Grid Engine 분배의 doc 디렉토리에 있는 arc_depend_*.asc 파일을 참조하십시오.
- 임시 디렉토리 경로.
- 작업 스크립트를 실행하는 데 사용될 기본 명령 해석기(Shell).
- 대기열에 대한 *당번* 및 *비번* 시간을 정의하기 위해 대기열에 첨부되는 캘린더.
- SIGUSR1/SIGUSR2 통지 신호 및 일시중단/종료 신호(Notify)의 전달 사이에 대기할 시간.
- 이 대기열에서 작업을 시작할 nice 값(0은 시스템 기본값 사용을 의미합니다).

- 대기열에서 동시에 실행하도록 허용될 작업 수(작업 슬롯).
- 대기열 및 이 대기열에서 실행하도록 허용되는 작업의 유형. 복수 선택이 가능합니다.
- Shell Start Mode, 즉 작업 스크립트를 시작할 모드.
- 새로 추가된 대기열이 접근하거나 대기열 호스트에서 실행 중인 sge_execd가 재시작되는 Initial State(초기 상태).
- 중단된(예를 들어 시스템 붕괴로 인해) 작업에서 강제될 대기열의 기본 rerun 정책. 사용자가 qsub -r 옵션이나 Job Submission 대화 상자에 의해 이 정책을 겹쳐쓸 수 있습니다(그림 4-7 참조).

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

▼ Execution Method 매개변수 구성 방법

- Execution Method 매개변수 세트를 선택합니다.

그림 7-2의 보기와 유사한 화면이 표시됩니다.

The screenshot shows a window titled "Queue Configuration: Modify". At the top left is the SGE logo. The main area contains a form with the following fields:

- Queue: bilbo.q
- Hostname: bilbo

Buttons for "Clone", "Reset", and "Refresh" are located below the form. On the right side, there are "Ok", "Cancel", and "Help" buttons. Below the form is a tabbed interface with the following tabs:

- Complexes
- Subordinates
- User Access
- Owners

Under the "Execution Method" tab, there are several sub-tabs: "General Configuration", "Execution Method", "Checkpointing", "Load/Suspend Thresholds", and "Limits". The "Execution Method" sub-tab is active, showing input fields for:

- Prolog
- Epilog
- Starter Method
- Suspend Method
- Resume Method
- Terminate Method

그림 7-2 Queue Configuration—Execution Method 매개변수

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 각각 작업 스크립트가 시작되기 전 및 종료된 후에 작업과 동일한 환경을 갖고 실행된 대기열 특정 프로로그 및 에필로그 스크립트.
- 시작/일시중단/재개/종료 메소드가 이들 조치를 작업에 적용하기 위해 Sun Grid Engine의 기본 메소드를 겹쳐쓰는 시작/일시중단/재개/종료 방법.

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

▼ Checkpointing 매개변수 구성 방법

- Checkpointing 매개변수 세트를 선택합니다.

그림 7-3의 보기와 유사한 화면이 표시됩니다.

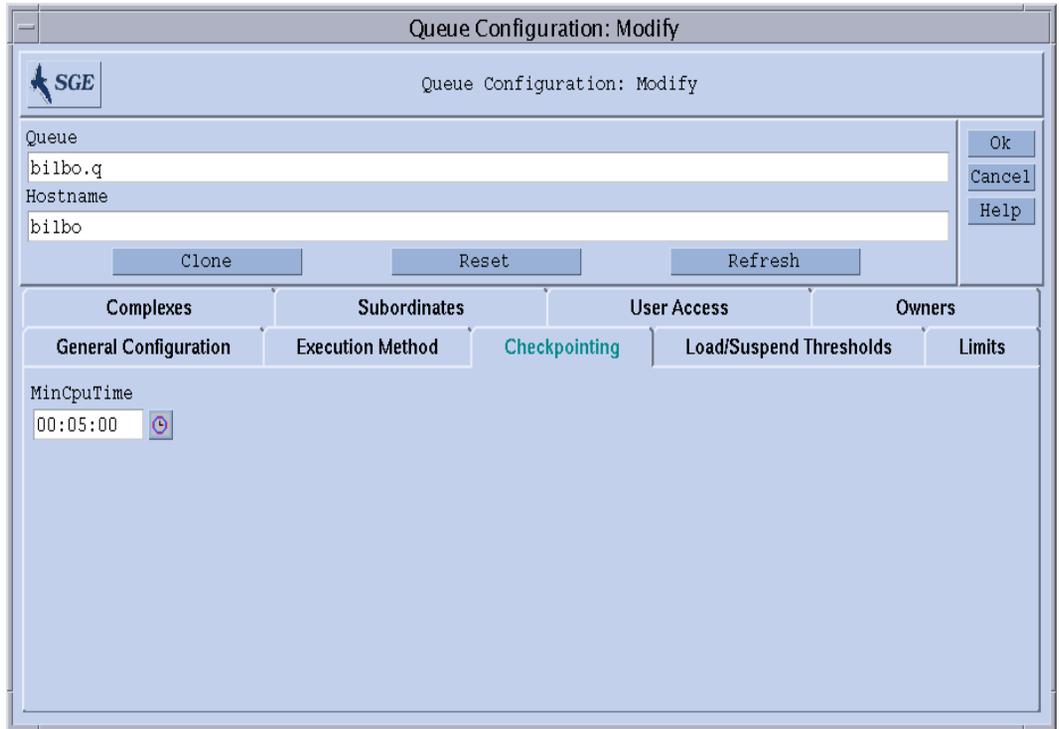


그림 7-3 Queue Configuration—Checkpointing 매개변수

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 주기적 검사점 간격(MinCpuTime)

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

▼ Load 및 Suspend Thresholds 구성 방법

- Load/Suspend Thresholds 매개변수 세트를 선택합니다.

그림 7-4의 보기와 유사한 화면이 표시됩니다.

그림 7-4 Queue Configuration—Load Thresholds

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 부하 매개변수 및 소비 가능 콤플렉스 속성에 대한 과부하 임계값을 정의하는 Load Thresholds 및 Suspend Thresholds 표(175 페이지, “콤플렉스 관련 정보”를 참조하십시오).

부하 임계값의 경우에 과부하는 대기열이 Sun Grid Engine에 의해 추가 작업을 받지 못하게 합니다. 하나 이상의 일시중단 임계값을 초과하면 부하를 줄이기 위해 대기열에 있는 작업의 일시중단을 유발합니다. 현재 구성된 임계값이 표에 표시됩니다. 대응하는 Value 필드에 왼쪽 마우스 단추로 두 번 눌러서 기존 임계값을 선택하고 변경할 수 있습니다. 새 임계값을 추가하려면 맨 위에 있는 Name 또는 Value 단추를 누르십시오. 그러면 대기열에 첨부된 모든 유효한 임계값을 갖는 선택 목록이 열립니다. Attributes Selection 대화 상자가 그림 6-6에 표시됩니다. 속성 중 하나를 선택하고 Ok 단추로 선택을 확정하면 해당 속성이 대응하는 임계값 표의 Name 열에 추가되고 포인터가 그의 Value 필드에 놓입니다. CTRL-D를 입력하거나 오른쪽 마우스 단추를 눌러서 삭제 상자를 열고 삭제를 확정하여 선택한 목록 항목을 삭제할 수 있습니다.

- 호스트가 대기열을 구성한 시스템의 부하를 줄이기 위해 시간 간격당 일시중단되는 작업의 수.
- 일시중단 임계값이 여전히 초과되는 경우에 추가 작업의 일시중단 사이의 시간 간격.

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

▼ Limits 구성 방법

- Limits 매개변수 세트를 선택합니다.

그림 7-5의 보기와 유사한 화면이 표시됩니다.

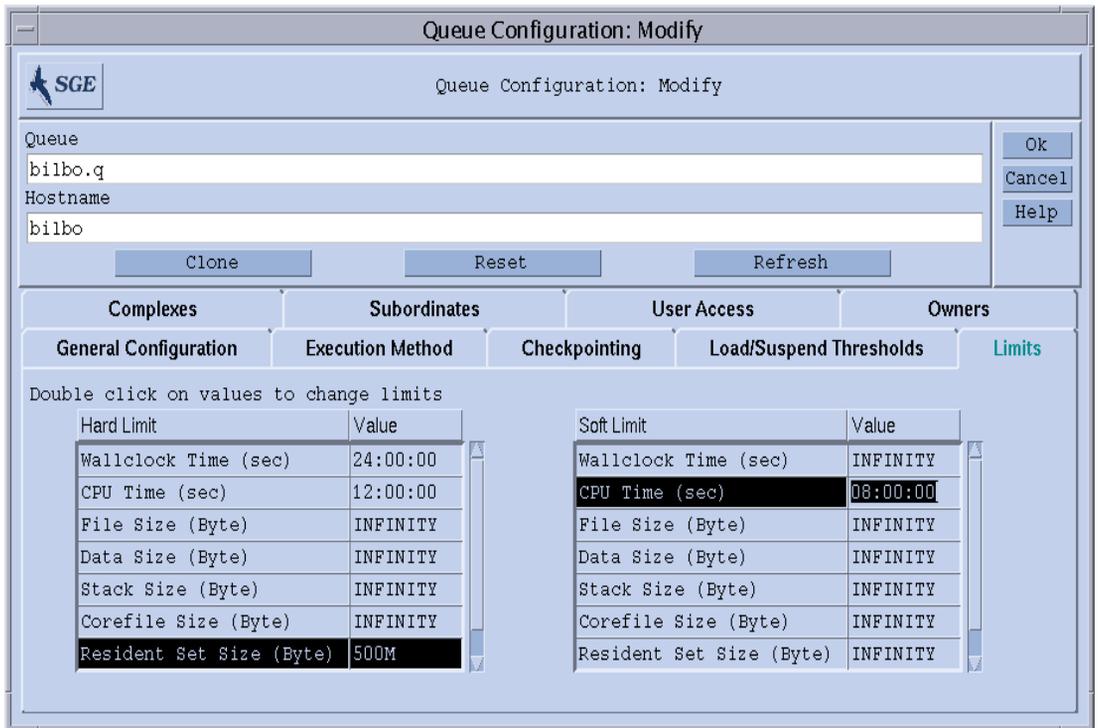


그림 7-5 Queue Configuration—Limits

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 대기열에서 실행 중인 작업에 부과될 하드 및 소프트 한계.

한계의 값을 변경하려면 한계 항목의 Value 필드를 두 번 누르십시오. Value 필드를 두 번째로 두 번 누르면 Memory 또는 Time 한계 값에 대한 편리한 입력 대화 상자가 열립니다(그림 7-6 및 그림 7-7 참조).

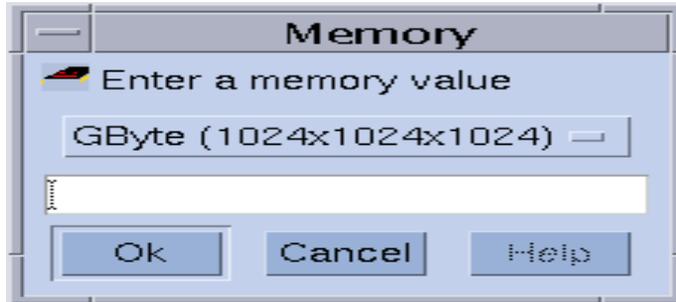


그림 7-6 Memory 입력 대화 상자

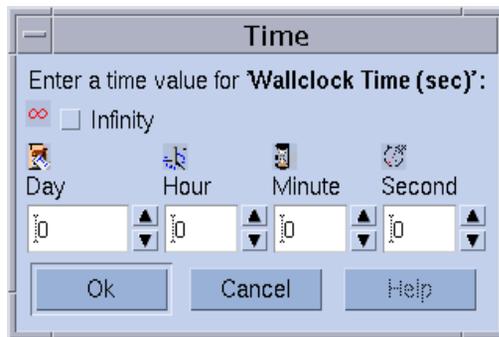


그림 7-7 Time 입력 대화 상자

여러 운영 체제 아키텍처에 대한 개별 한계 매개변수 및 그의 해석에 대한 자세한 정보는 `queue_conf` 및 `setrlimit` 매뉴얼 페이지를 참조하십시오.

▼ User Complexes 구성 방법

- User Complexes 매개변수 세트를 선택합니다.

그림 7-8의 보기와 유사한 화면이 표시됩니다.

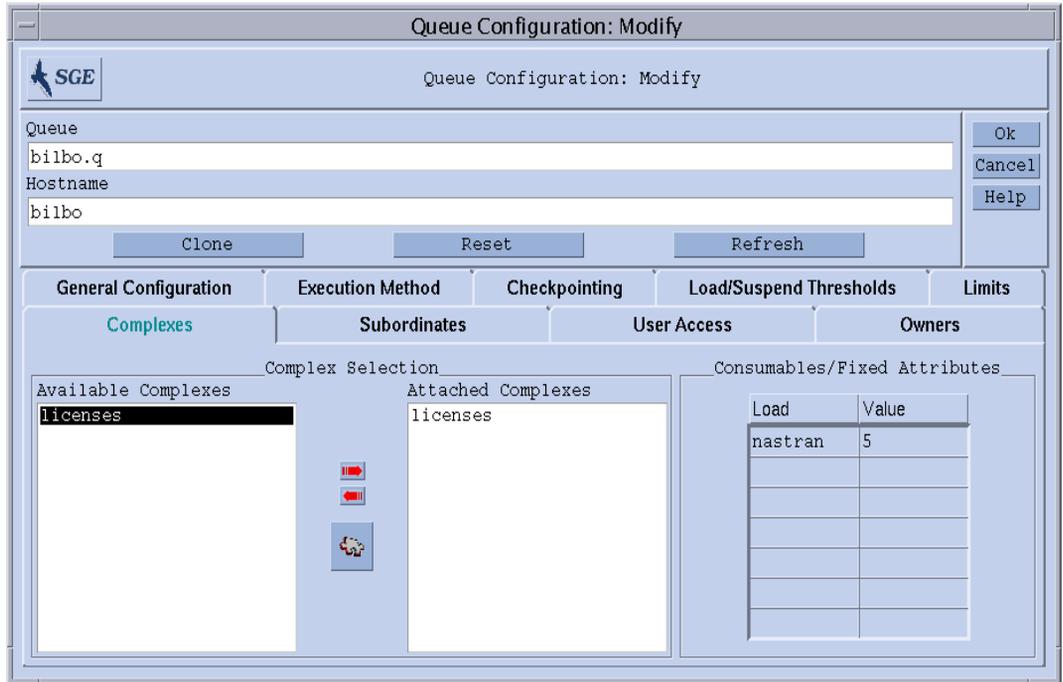


그림 7-8 Queue Configuration—User Complexes

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 대기열에 첨부되는 사용자 정의 콤플렉스의 세트(181 페이지, “사용자 정의 콤플렉스” 참조)

Complex Selection 상자의 중앙에 있는 빨간색 화살표를 사용하여 대기열에/에서 사용자 정의 콤플렉스를 추가 및 삭제할 수 있습니다.

- 이 대기열에 사용 가능한 콤플렉스 매개변수 세트에서 선택한 속성에 대한 값 정의 사용 가능한 콤플렉스 매개변수는 글로벌 콤플렉스, 호스트 콤플렉스 및 추가된 사용자 정의 콤플렉스로부터 기본값당 어셈블됩니다. 속성은 소비 가능 또는 고정 매개변수입니다. 대기열 값의 정의는 소비 가능한 속성의 경우에 대기열에 의해 관리되는 용량 또는 고정 속성의 경우에 간단하게 고정된 대기열 특정 값을 정의합니다. (자세한 내용은 175 페이지, “콤플렉스 관련 정보”를 참조하십시오.) 그에 대한 값이 명시적으로 정의되는 속성이 Consumable/Fixed Attributes 표에 표시됩니다. 대응하는 Value 필드를 두 번 눌러서 기존 속성을 선택하고 변경할 수 있습니다. 새 속성 정의를 추가하려면 맨 위에 있는 Name 또는 Value 단추를 누르십시오. 그러면 대기열에 첨부된 모든 유효한 임계값을 갖는 선택 목록이 열립니다. Attribute Selection 대화 상

자는 그림 6-6에 표시되어 있습니다. 속성 중 하나를 선택하고 Ok 단추로 선택을 확정하면 해당 속성이 속성 표의 Name 열에 추가되고 포인터가 그의 Value 필드에 놓입니다. CTRL-D를 입력하거나 오른쪽 마우스 단추를 눌러서 삭제 상자를 열고 삭제를 확정하여 선택한 목록 항목을 삭제할 수 있습니다.

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

Complex Config 아이콘 단추를 누를 때 Complex Configuration 대화 상자(예를 들어 8장, 175 페이지, “복플렉스 개념”의 그림 8-5를 참조하십시오)가 열립니다. 사용자 정의 콤플렉스가 대기열에 추가 또는 삭제되기 전에 현재 콤플렉스 구성을 확인 또는 수정할 수 있습니다.

▼ 종속 대기열 구성 방법

- **Subordinates** 매개변수 세트를 선택합니다.

그림 7-9의 보기와 유사한 화면이 표시됩니다.

Queue Configuration: Modify

Queue Configuration: Modify

Queue: bilbo.q

Hostname: bilbo

Clone Reset Refresh

Ok Cancel Help

General Configuration Execution Method Checkpointing Load/Suspend Thresholds Limits

Complexes Subordinates User Access Owners

Queue	Max Slots
arwen.q	2

If 'Max Slots' are filled in the current queue, the queues in column one are suspended.

그림 7-9 Queue Configuration—Subordinates

제공되는 필드는 다음 매개변수 설정을 허용합니다.

■ 구성된 대기열에 종속되는 대기열

종속되는 대기열은 구성된 대기열이 사용 중이 되는 경우에 일시중단되고 구성된 대기열이 더 이상 사용 중이 아닌 경우 일시중단 해제됩니다. 모든 종속된 대기열에 대해 일시중단을 트리거하기 위해 구성된 대기열에서 최소한 점유되어야 하는 작업 슬롯의 수를 구성할 수 있습니다. 작업 슬롯 값이 지정되지 않는 경우 대응하는 대기열의 일시중단을 트리거하려면 모든 슬롯이 채워져야 합니다.

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

독립형 대기열뿐 아니라 높은 우선순위 및 낮은 우선순위 대기열을 구현하려면 종속 대기열 기능을 사용하십시오.

▼ User Access 구성 방법

● User Access 매개변수 세트를 선택합니다.

그림 7-10의 보기와 유사한 화면이 표시됩니다.

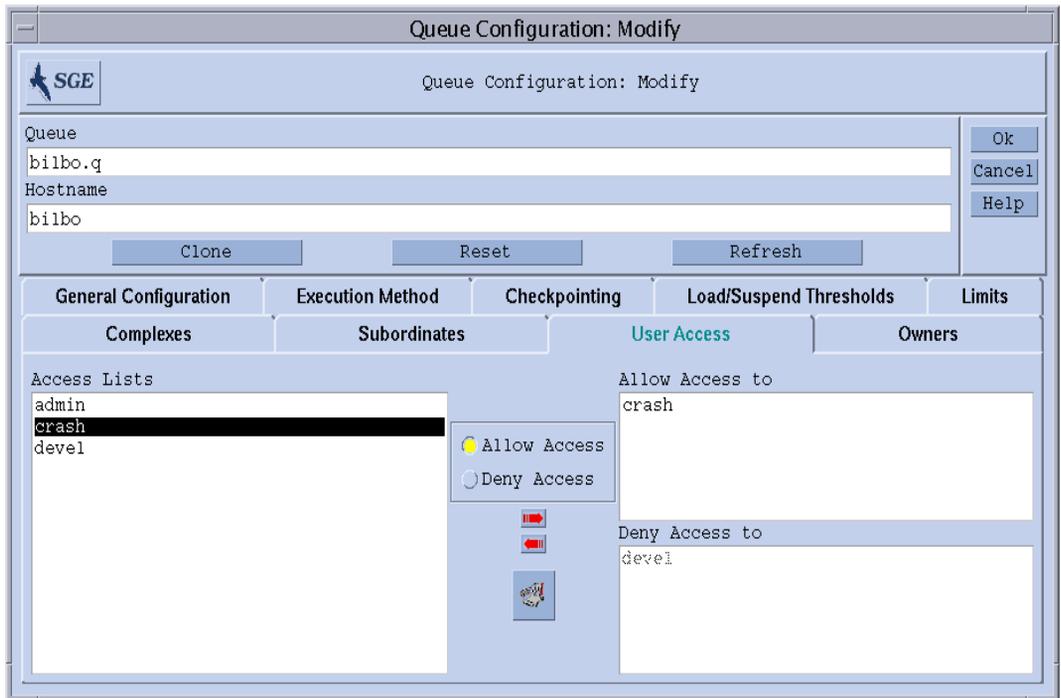


그림 7-10 Queue Configuration—User Access

제공되는 필드는 다음 매개변수 설정을 허용합니다.

- 대기열의 허용 또는 거부 목록에 추가될 사용자 액세스 목록

허용 목록에 포함되는 액세스 목록에 속하는 사용자 또는 사용자 그룹은 대기열에 대한 액세스를 갖습니다. 거부 목록과 연관되는 사용자는 대기열에 액세스할 수 없습니다. 허용 목록이 빈 경우 거부 목록에서 명시적으로 언급되지 않는 한 액세스는 제한되지 않습니다.

이들 매개변수에 대한 자세한 정보는 queue_conf 매뉴얼 페이지를 참조하십시오.

화면의 중간 맨 아래에 있는 단추를 눌러서 Access List Configuration 대화 상자(64 페이지, “사용자 액세스 사용 권한” 참조)를 여십시오.

▼ Owners 구성 방법

- Owners 매개변수 세트를 선택합니다.

그림 7-11의 보기와 유사한 화면이 표시됩니다.

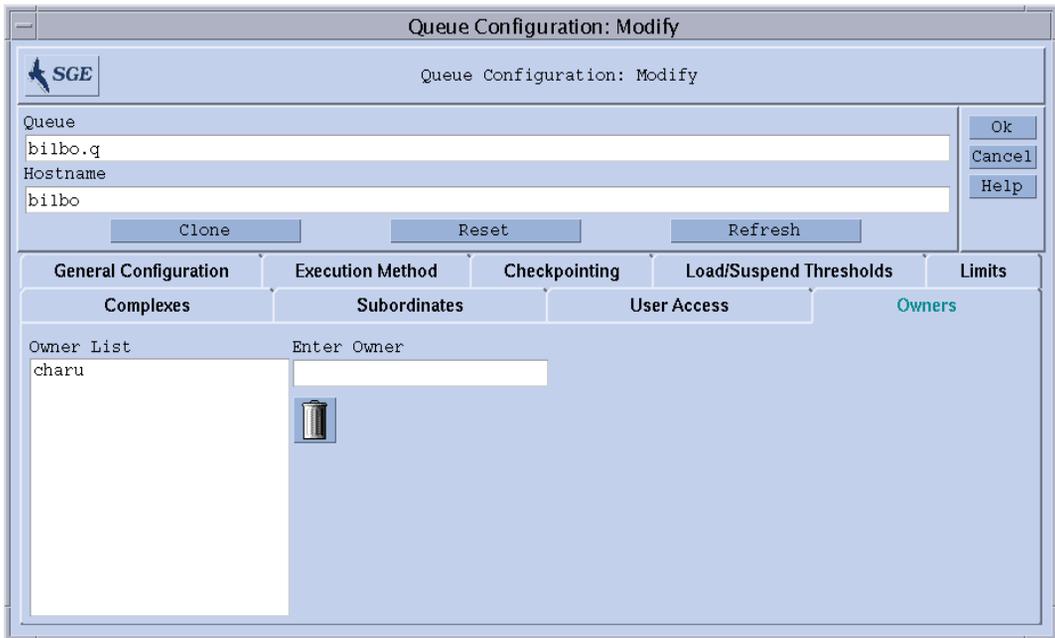


그림 7-11 Queue Configuration—Owners

제공되는 필드는 다음 매개변수 설정을 허용합니다.

■ 대기열 소유자의 목록

대기열의 소유자는 대기열을 일시중단/일시중단 해제 또는 사용 안함/사용하는 권한이 부여됩니다. 모든 적합한 사용자 계정이 대기열 소유자 목록에 추가될 유효한 값입니다. 대기열 소유자 목록에서 사용자 계정을 삭제하려면 **Owner List** 창에서 계정을 선택하고 대화 상자의 오른쪽 하단 모서리에 있는 휴지통 아이콘을 누르십시오.

이들 매개변수에 대한 자세한 정보는 `queue_conf` 매뉴얼 페이지를 참조하십시오.

▼ 명령줄에서 대기열을 구성하는 방법

- 대기열을 구성하려는 방법에 따라서 적당한 인수와 함께 다음 명령을 입력하십시오.

```
# qconf 옵션
```

`qconf` 명령은 다음 옵션을 갖습니다.

■ `qconf -aq [대기열이름]`

대기열 추가—대기 구성 템플릿을 갖는 편집기(기본 `vi` 또는 `$EDITOR` 환경 변수에 따라서)를 표시합니다. 선택적 매개변수인 *대기열이름*이 제공되면 이 대기열의 구성이 템플릿으로 사용됩니다. 대기열은 템플릿을 변경하고 디스크에 저장하여 구성됩니다. 변경할 템플릿 항목의 자세한 설명은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `queue_conf` 항목을 참조하십시오.

■ `qconf -Aq 파일이름`

대기열 추가—*파일이름* 파일을 사용하여 대기열을 정의합니다. 정의 파일은 `qconf -sq 대기열이름`(아래 참조)에 의해 생성되었을 수 있습니다.

■ `qconf -cq 대기열이름[...]`

대기열 지우기—지정된 대기열의 상태를 유휴가 되도록 지우고 실행 중인 작업에서 사용 가능하게 합니다. 상태는 현재 상태와 상관없이 재설정됩니다. 이 옵션은 오류 조건을 제거하기에 유용하지만 정상 동작 모드에서는 사용하지 않아야 합니다.

■ `qconf -dq 대기열이름[...]`

대기열 삭제—사용 가능한 대기열의 목록에서 인수 목록에 지정되는 대기열을 삭제합니다.

■ `qconf -mq 대기열이름`

대기열 수정—지정된 대기열을 수정합니다. 변경될 대기열의 구성을 갖는 편집기(기본 `vi` 또는 `$EDITOR` 환경 변수에 따라서)를 표시합니다. 대기열은 구성을 변경하고 디스크에 저장하여 수정됩니다.

■ `qconf -Mq 파일이름`

대기열 수정—`파일이름` 파일을 사용하여 수정된 대기열 구성을 정의합니다. 정의 파일은 `qconf -sq 대기열이름`(아래 참조) 및 후속 수정에 의해 생성되었을 수 있습니다.

■ `qconf -sq [대기열이름,...]`

대기열 표시—기본 템플릿 대기열 구성(인수가 없는 경우) 또는 쉘표로 구분된 인수 목록에 나열되는 대기열의 현재 구성을 표시합니다.

■ `qconf -sql`

대기열 목록 표시—현재 구성된 모든 대기열의 목록을 표시합니다.

대기열 캘린더 정보

대기열 캘린더는 1년 중 날짜, 요일 및/또는 하루 중 시간에 종속되는 Sun Grid Engine 대기열의 가용성을 정의합니다. 임의의 시점에서 상태를 변경하도록 대기열을 구성할 수 있습니다. 대기열 상태는 사용 안함, 사용, 일시중단 및 재개(일시중단 해제)로 변경할 수 있습니다.

Sun Grid Engine은 각각이 임의의 상태 변경 및 상태 변경이 발생하는 시간 이벤트가 들어 있는 사이트에 고유한 캘린더 세트를 정의하는 기능을 제공합니다. 이들 캘린더를 대기열이 참조할 수 있습니다. 즉, 각 대기열이 단일 캘린더를 추가할 수 있으므로(또는 없으므로) 연결된 캘린더에 정의되는 가용성 프로파일을 채택할 수 있습니다.

캘린더 형식의 구문은 매뉴얼 페이지 `calendar_conf`에 자세히 설명되어 있습니다. 아래에 대응하는 관리 기능의 설명과 함께 몇 가지 예가 제공됩니다.

▼ QMON을 사용하여 대기열 캘린더를 구성하는 방법

1. QMON Main 메뉴에서 **Calendar Config**를 누릅니다.

그림 7-12과 유사한 Queue Calendar Configuration 대화 상자가 표시됩니다.

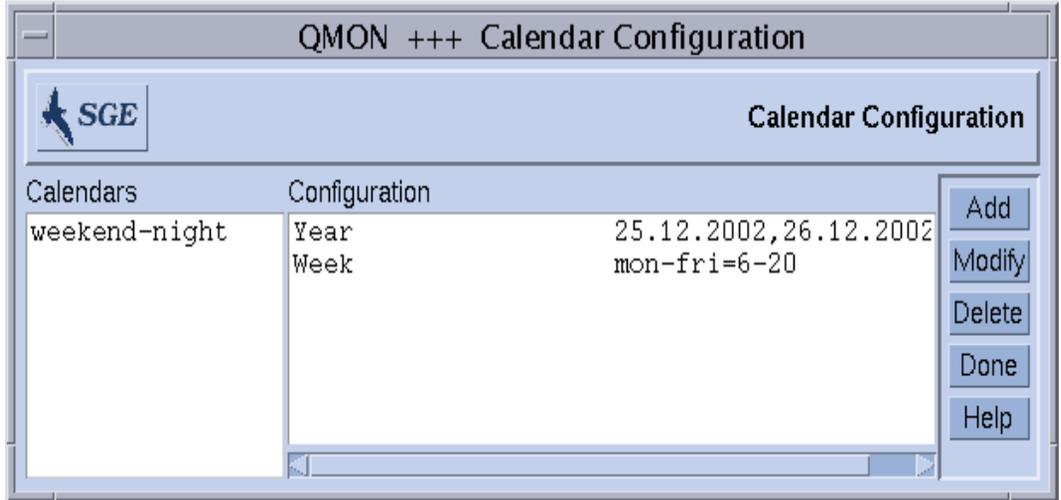


그림 7-12 Calendar Configuration

사용 가능한 액세스 목록이 화면의 왼쪽에 있는 Calendars 선택 목록에 표시됩니다.

2. **Calendars** 선택 목록에서 수정 또는 삭제하려는 캘린더 구성을 누릅니다.
3. 구성을 변경하려는 방법에 따라서 다음 중 하나를 수행합니다.
 - a. 화면의 오른쪽에 있는 **Delete** 단추를 눌러서 선택된 캘린더를 삭제합니다.
 - b. **Modify** 단추를 눌러서 선택된 캘린더를 수정합니다.
 - c. **Add** 단추를 눌러서 액세스 목록을 추가합니다.

모든 경우에 그림 7-13의 보기와 유사한 Calendar Definition 대화 상자가 열리고 삭제, 수정 또는 추가하는 수단을 제공합니다.

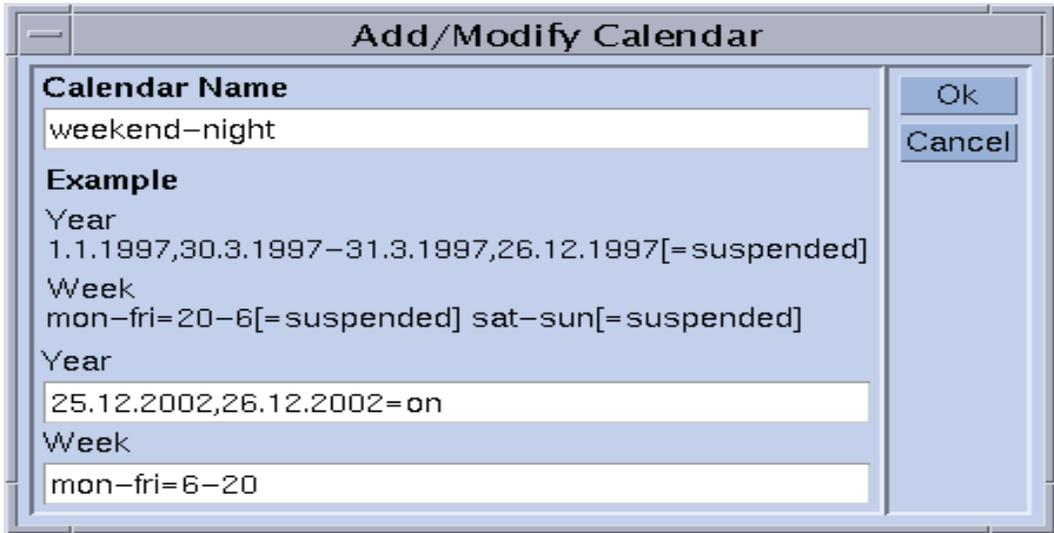


그림 7-13 캘린더 추가, 삭제 또는 수정

4. 다음 절의 지침에 따라서 계속합니다.

수정 조작의 경우에 Calendar Name 입력 창이 선택된 캘린더의 이름을 표시하거나 이 창을 사용하여 선언된 캘린더의 이름을 입력할 수 있습니다. Year 및 Week 입력 필드에서 calendar_conf 매뉴얼 페이지에 설명되는 구문을 사용하여 캘린더 이벤트를 정의할 수 있습니다.

위의 캘린더 구성의 예는 외근 시간과 주말에 사용 가능해야 하는 대기열에 적합합니다. 또한 크리스마스 휴일은 주말처럼 처리되도록 정의되었습니다.

구문의 자세한 설명과 추가 예에 대해서는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 calendar_conf 항목을 참조하십시오.

대기열에 대한 캘린더 구성을 접속하여 캘린더에 의해 정의된 가용성 프로파일이 대기열에 대해 설정됩니다. 캘린더는 그림 7-14에 표시되는 대로 일반 매개변수 대기열 구성에서 접속됩니다. Calendar 입력 필드에는 접속될 캘린더 이름이 들어 있고 입력 필드 옆의 아이콘 단추는 현재 구성된 캘린더의 목록을 갖는 선택 대화 상자를 엽니다. 대기열 구성에 대한 더 자세한 내용은 157 페이지, “대기열 구성 정보” 절을 참조하십시오.

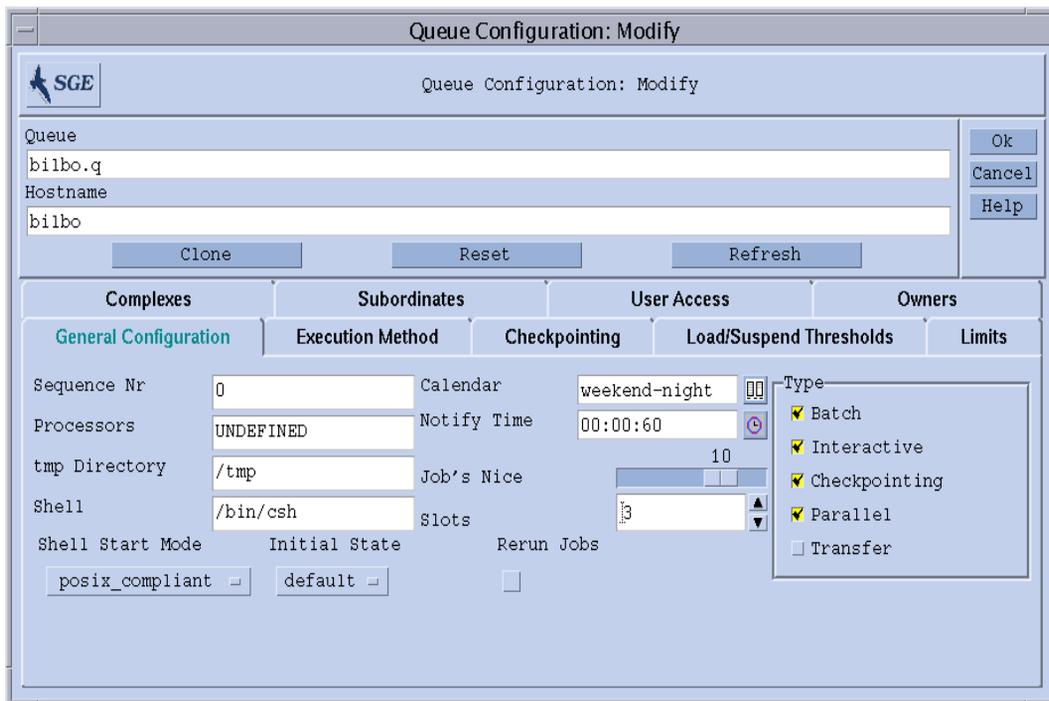


그림 7-14 General 매개변수 Queue Configuration 메뉴의 Calendar Configuration

▼ 명령줄에서 캘린더를 구성하는 방법

- 적당한 스위치와 함께 다음 명령을 입력하십시오.

```
% qconf 스위치
```

4개의 사용 가능한 스위치는 다음과 같습니다.

- `qconf -Acal, -acal`

Add calendar – 이 명령은 새 캘린더 구성을 Sun Grid Engine 클러스터에 추가합니다. 추가될 캘린더가 파일에서 읽히거나 템플릿 구성을 갖는 편집기가 열려서 캘린더를 입력할 수 있습니다.

- `qconf -dcal`

Delete Calendar.

- `qconf -Mcal, -mcal`

Modify calendar - 이 명령은 기존 캘린더 구성을 수정합니다. 수정될 캘린더가 파일에서 읽어지거나 이전 구성을 갖는 편집기가 열려서 새 정의를 입력할 수 있습니다

- `qconf -scal, -scall`

Show calendar - 이 명령은 기존 캘린더 구성(-scal)을 표시하거나 구성된 모든 캘린더 목록(-scall)을 인쇄합니다.

컴플렉스 개념

이 장에서는 *컴플렉스*로 알려진 중요한 Sun Grid Engine 5.3 개념에 대해 설명합니다. 컴플렉스 및 연관된 개념과 관련된 백그라운드 정보 외에 이 장은 다음 업무를 수행하는 방법에 대한 자세한 지침을 제공합니다.

- 176 페이지, “컴플렉스 구성을 추가 또는 수정하는 방법”
- 185 페이지, “소비 가능 자원을 설정하는 방법”
- 196 페이지, “명령줄에서 컴플렉스 구성을 수정하는 방법”
- 198 페이지, “사용자 자신의 부하 센서를 작성하는 방법”

컴플렉스 관련 정보

컴플렉스의 정의는 사용자가 `qsub` 또는 `qalter -l` 옵션을 통한 Sun Grid Engine 작업 및 Sun Grid Engine 시스템 내에서 이들 매개변수의 해석에 대해 요청할 수 있는 자원 속성에 관한 모든 관련 정보를 제공합니다.

컴플렉스는 또한 연관된 용량을 갖는 자원을 식별하는 클러스터 글로벌, 호스트 특정 또는 대기열 관련 속성의 정의를 허용하는 기능인 Sun Grid Engine 시스템의 *소비 가능 자원* 기능을 위한 프레임워크를 구축합니다. Sun Grid Engine 작업의 요구사항과 결합되는 자원의 가용성이 스케줄링 프로세스 중에 고려됩니다. Sun Grid Engine은 또한 소비 가능 자원의 과다 신청을 막기 위해 필요한 부기 및 용량 계획을 수행합니다. 전형적인 소비 가능 속성에 대한 예는 사용 가능한 메모리, 소프트웨어 패키지의 점유되지 않은 사용권, 여유 디스크 공간 또는 네트워크 연결의 사용 가능한 대역폭입니다.

보다 일반적인 의미에서 Sun Grid Engine 컴플렉스는 대기열, 호스트 및 클러스터 속성의 의도된 해석을 설명하는 수단으로 사용됩니다. 설명에는 속성 이름, 그를 참조하는 데 사용할 수 있는 단축키, 속성의 값 유형(예: STRING 또는 TIME), 컴플렉스 속성에 지정될 사전 정의된 값, Sun Grid Engine 스케줄러 `sge_schedd`가 사용하는 관계 연산자, 속성이 사용자에 의해 작업에 대해 요청될 수 있는지 여부를 판별하는 요청 가능 플래그, 설정되는 경우 속성을 소비 가능 속성으로 식별하는 소비 가능 플래그 및 작업이 소비 가능 속성에 대해 요청을 명시적으로 지정하지 않는 경우 그런 속성에 대해 고려되는 기본 요청 값이 포함됩니다.

그림 8-1에 표시된 QMON Complex Configuration 대화 상자가 콤플렉스 속성을 정의할 수 있는 방법을 보여줍니다.

▼ 콤플렉스 구성을 추가 또는 수정하는 방법

1. QMON Main 메뉴에서 Complex Configuration 단추를 누릅니다.

그림 8-1에 표시된 것과 비슷한 Complex Configuration 대화 상자가 표시됩니다.

2. 다음 절에서 자세히 설명되는 정보를 바탕으로 콤플렉스 구성을 추가 또는 수정합니다.

- 178 페이지, “대기열 콤플렉스”
- 178 페이지, “호스트 콤플렉스”
- 180 페이지, “글로벌 콤플렉스”
- 181 페이지, “사용자 정의 콤플렉스”

Complex Configuration 대화 상자는 기존 콤플렉스의 정의를 변경하고 새 사용자 콤플렉스를 정의하기 위한 수단을 제공합니다.

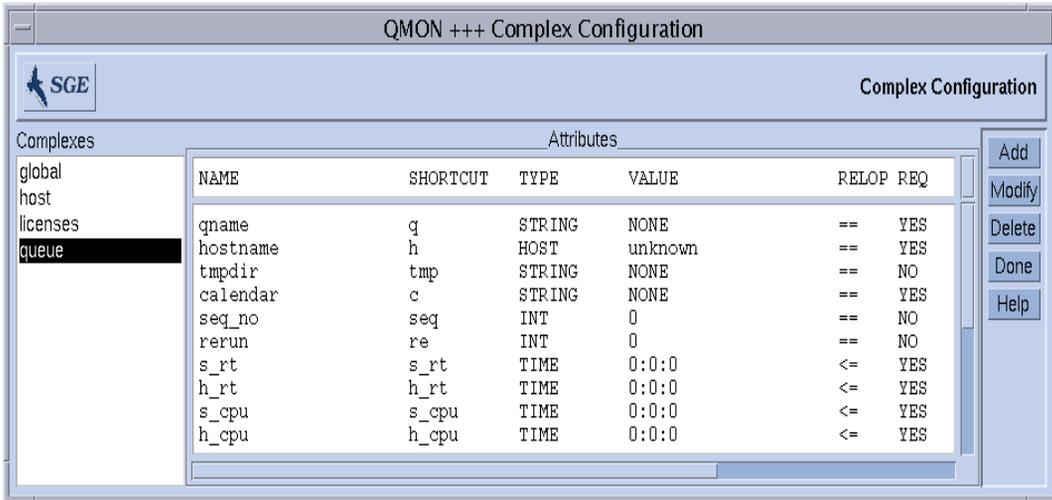


그림 8-1 Complex Configuration 대화 상자—Queue

화면의 왼쪽에 시스템에 알려진 모든 콤플렉스에 대한 선택 목록이 표시됩니다. 콤플렉스를 수정 또는 삭제해야 하는 경우에 사용할 수 있습니다. 화면의 오른쪽에 있는 대응하는 단추로 원하는 조작(Add, Modify 또는 Delete)을 선택할 수 있습니다. 새 콤플렉스가 작성되거나 기존 콤플렉스가 수정되는 경우 그림 8-2의 보기와 유사한 대화 상자가 열립니다.

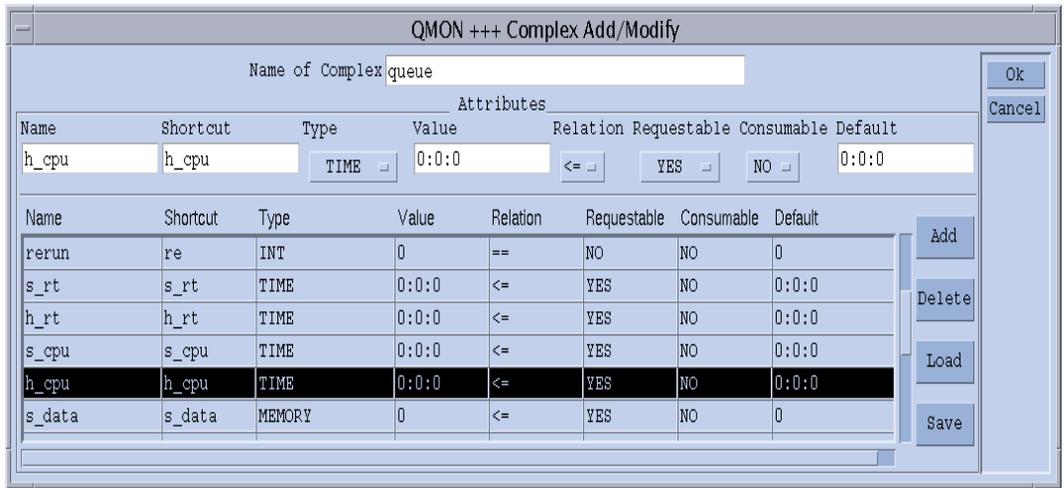


그림 8-2 Complex Add/Modify 대화 상자

컴플렉스의 이름을 입력하거나 맨 위에 있는 Name of Complex 입력 창에 이름이 표시되는 경우 선택해야 합니다. 왼쪽 마우스 단추로 행을 선택하여 Complex Definition 표에 컴플렉스 속성을 수정할 수 있습니다. 선택한 항목이 정의 창과 Attributes 상자의 맨 위에 있는 선택기에 표시됩니다. 정의를 변경하고 Add 단추를 누르면 정의 표에서 변경사항이 갱신됩니다.

정의 창을 기입하고 선택기를 사용한 후 Add 단추를 눌러서 새 항목을 추가할 수 있습니다. 새 항목을 추가할 때는 속성 표에서 행을 선택하지 않아야 합니다.

Load 및 Save 단추를 사용하여 일반 파일에서 컴플렉스 구성을 로드하고 일반 파일에 저장할 수 있습니다. 파일을 선택할 파일 선택 상자가 열립니다. Delete 단추를 사용하여 컴플렉스 구성의 선택한 행을 삭제할 수 있습니다.

표에 있는 행 및 열의 의미에 대해서는 컴플렉스 매뉴얼 페이지를 참조하십시오. 화면의 우측 상단 모서리의 Ok 단추는 마지막으로 신규/변경된 컴플렉스를 sge_qmaster에 등록합니다.

컴플렉스 유형

Sun Grid Engine 컴플렉스 오브젝트는 4 가지 다른 유형의 컴플렉스를 통합합니다.

- 대기열 컴플렉스
- 호스트 컴플렉스
- 글로벌 컴플렉스
- 사용자 정의 컴플렉스

다음 절은 각 유형을 세부적으로 설명합니다.

대기열 콤플렉스

대기열 콤플렉스는 `queue`라는 특수 이름으로 참조됩니다.

기본 양식에서 `queue_conf`에 정의되는 대기열 구성의 매개변수 선택이 들어 있습니다. 대기열 콤플렉스의 주 목적은 이들 매개변수가 해석되는 방법을 정의하고 모든 대기열에 사용 가능할 추가 속성에 대한 컨테이너를 제공하는 것입니다. 따라서 대기열 콤플렉스는 사용자 정의 속성에 의해 확장될 수 있습니다.

대기열 콤플렉스가 특정 대기열과의 컨텍스트에서 참조되는 경우 대기열의 대응하는 구성 값이 대기열 콤플렉스의 속성 값을 대체합니다(`value` 열을 *겹쳐줍니다*).

예를 들어 대기열 콤플렉스가 *big*이라는 대기열에 대한 설정인 경우, 기본값 `unknown` (그림 8-1 참조)을 전달하는 대기열 콤플렉스 속성 `qname`에 대한 값 열이 *big*으로 설정됩니다.

대기열 구성에서 `complex_values` 매개변수를 사용하여 이 내재적 값 설정을 겹쳐쓸 수 있습니다(157 페이지, “대기열 구성 정보”를 참조하십시오). 이것은 대개 *소비 가능 자원*에 대해 수행됩니다(185 페이지, “소비 가능 자원” 절 참조). 예를 들어 가상 메모리 크기 한계의 경우 대기열 구성 값 `h_vmem`이 작업당 총 점유 메모리의 양을 제한하는데 사용되는 반면, `complex_values` 목록의 대응하는 항목은 호스트에 있거나 대기열에 할당된 가상 메모리의 총 사용 가능한 양을 정의합니다.

관리자가 대기열 콤플렉스에 속성을 추가하는 경우 특정 대기열과 연관되는 속성의 값은 해당 대기열의 `complex_values` 매개변수를 통해 정의되거나 대기열 콤플렉스 구성의 `value` 열이 기본적으로 사용됩니다.

호스트 콤플렉스

호스트 콤플렉스는 특수 이름 `host`에 의해 참조되며 호스트 기반으로 관리될 모든 속성의 특성 정의가 들어 있습니다(그림 8-3 참조). 호스트 관련 속성의 표준 세트는 두 범주로 구성되지만 위에서 설명한 대기열 콤플렉스와 같이 향상될 수 있습니다. 첫 번째 범주는 특히 호스트 기반으로 관리하기에 적합한 여러 대기열 구성 속성에 의해 구성됩니다. 이들 속성은 다음과 같습니다.

- 슬롯
- `seven`
- `h_vmem`
- `s_fsize`
- `h_fsize`

(자세한 내용은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `queue_conf` 항목 참조).

참고 - 호스트 콤플렉스에 이들 속성을 정의하는 것은 대기열 속성에서도 갖는 것과 상충하지 않습니다. 이것은 대응하는 자원을 호스트 레벨에서 및 동시에 대기열 레벨에서 유지보수하는 것을 허용합니다. 예를 들어 총 가상 여유 메모리(h_vmem)를 호스트에 대해 관리할 수 있으며 총 메모리의 서브셋을 해당 호스트의 대기열과 연관시킬 수 있습니다.

표준 호스트 콤플렉스에서 두 번째 속성 범주는 기본 부하 값입니다. 모든 sge_execd 는 주기적으로 sge_qmaster에 부하를 보고합니다. 보고되는 부하 값은 CPU 부하 평균 같은 표준 Sun Grid Engine 부하 값 또는 Sun Grid Engine 관리에 의해 정의되는 부하 값입니다(197 페이지, “부하 매개변수” 절 참조). 표준 부하 값에 대한 특성 정의는 기본 호스트 콤플렉스의 일부인 반면, 관리자 정의의 부하 값은 호스트 콤플렉스의 확장이 필요합니다.

호스트 콤플렉스는 보통 비 표준 부하 매개변수를 포함하고 또한 호스트에 할당되는 소프트웨어 사용권의 수나 호스트 로컬 파일 시스템의 사용 가능한 디스크 공간 같은 호스트 관련 자원을 관리하도록 확장됩니다.

호스트 콤플렉스가 호스트 또는 해당 호스트의 대기열과 연관되는 경우 특정 호스트 콤플렉스 속성에 대한 구체적인 값은 다음 중 하나에 의해 판별됩니다.

- 대기열 구성 파생 속성의 경우에 대기열 구성
- 보고된 부하 값
- 대응하는 호스트 구성의 complex_values 항목에 있는 값의 명시적 정의(138 페이지, “호스트 구성 정보” 절 참조)

위의 어느 것도 사용할 수 없는 경우(예를 들어 값이 부하 매개변수인 것으로 가정되지만 sge_execd가 그에 대한 부하 값을 보고하지 않는 경우) 호스트 콤플렉스 구성의 value 필드가 사용됩니다.

예를 들어 총 여유 가상 메모리 속성 h_vmem이 대기열 구성에 한계로서 정의되고 또한 표준 부하 매개변수로서 보고됩니다. 호스트 및 해당 호스트의 대기열에 첨부된 가상 메모리의 총 사용 가능 용량은 해당 호스트 및 해당 대기열 구성의 complex_values 목록에 정의될 수 있습니다. h_vmem을 소비 가능 자원(185 페이지, “소비 가능 자원” 참조)으로 정의하는 것과 함께, 이것은 종종 스왑핑에 의해 유발되는 축소된 시스템 성능을 가져오는 메모리 과다신청의 위험 없이 시스템의 메모리를 효율적으로 이용할 수 있게 합니다.

참고 - 시스템 기본 부하 속성에 대해 Shortcut, Value, Relation, Requestable, Consumable 및 Default 열만을 변경할 수 있습니다. 기본 속성을 삭제해서는 안 됩니다.

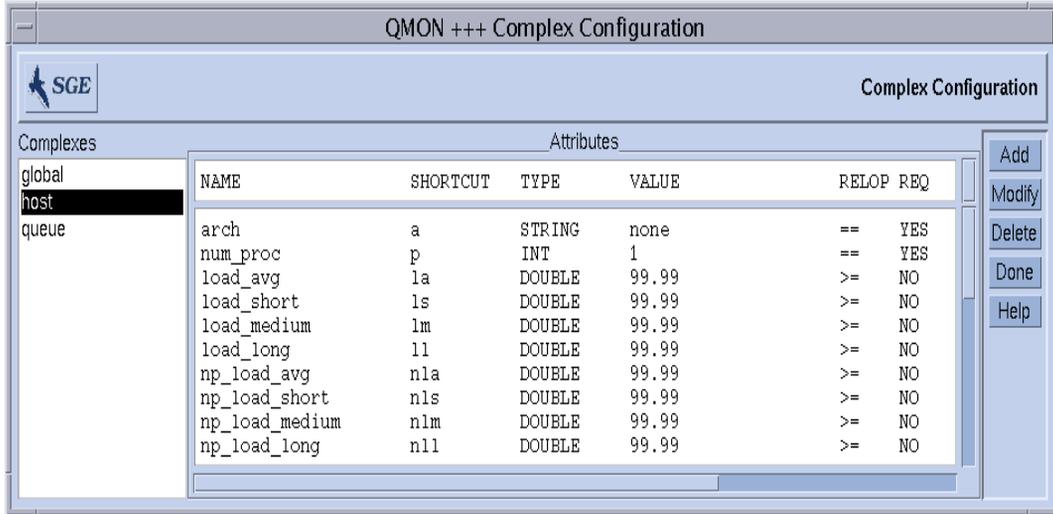


그림 8-3 Complex Configuration 대화 상자—Host

글로벌 콤플렉스

글로벌 콤플렉스는 global라는 특수 이름으로 참조됩니다.

글로벌 콤플렉스에 구성되는 항목은 파일 서버의 사용 가능한 네트워크 대역폭 또는 네트워크측 사용 가능 파일 시스템의 여유 디스크 공간 같은 클러스터 전체의 자원 속성을 참조합니다(그림 8-4 참조). 글로벌 자원 속성은 또한 대응하는 부하 보고서에 GLOBAL ID가 들어 있는 경우, 부하 보고서와 연관될 수도 있습니다(197 페이지, “부하 매개변수” 절 참조). 글로벌 부하값은 클러스터의 모든 호스트로부터 보고될 수 있습니다. 기본적으로 Sun Grid Engine이 보고하는 글로벌 부하 값은 없으며 따라서 기본 글로벌 콤플렉스 구성이 없습니다.

글로벌 콤플렉스 속성에 대한 구체적인 값은 글로벌 부하 보고서에 의해서 또는 global 호스트 구성(138 페이지, “호스트 구성 정보” 절 참조)의 complex_values 매개변수에 있는 명시적인 정의에 의해서 또는 특정 호스트 또는 대기열 및 대응하는

complex_values 목록의 명시적인 정의와 연관되어 판별됩니다. 위의 어느 것도 해당되지 않는 경우(예를 들어 부하 값이 아직 보고되지 않은 경우) 글로벌 콤플렉스 구성의 value 필드가 사용됩니다.

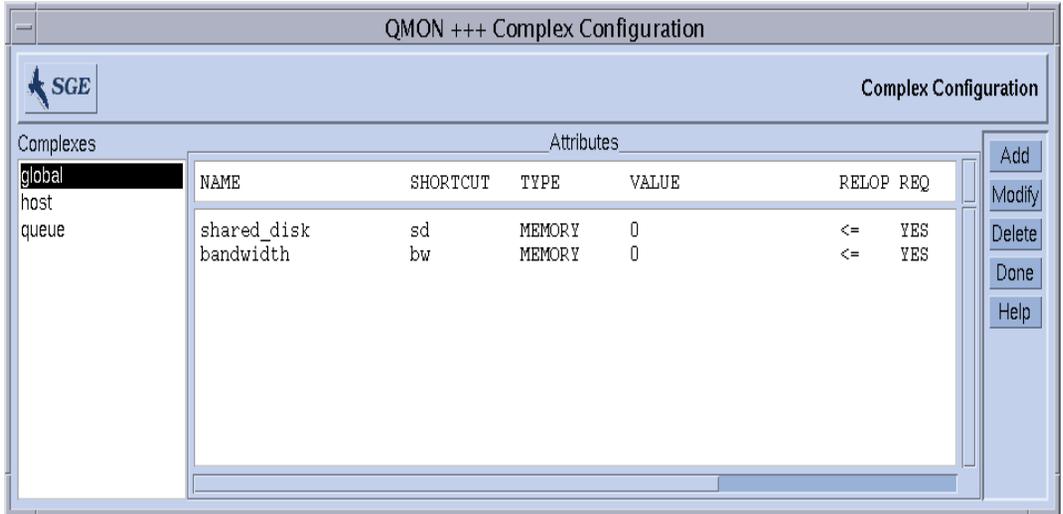


그림 8-4 Complex Configuration 대화 상자—Global

사용자 정의 콤플렉스

사용자 정의 콤플렉스를 설정함으로써 Sun Grid Engine 관리는 특정 대기열 및/또는 호스트에 대한 해당 속성의 영향을 제한하는 한편 Sun Grid Engine에 의해 관리되는 속성 세트를 확장하는 기능을 갖습니다. 사용자 콤플렉스는 단순히 속성 및 이들 속성이 Sun Grid Engine에 의해 처리될 방법에 대한 대응하는 정의의 이름 지정된 콜렉션입니다. 이들 사용자 정의 콤플렉스의 하나 이상을 complex_list 대기열 및 호스트 구성 매개변수를 통해 대기열 및/또는 호스트에 첨부할 수 있습니다 (157 페이지, “대기열 구성 정보” 및 138 페이지, “호스트 구성 정보” 절 참조). 모든 할당된 콤플렉스에서 정의되는 속성이 기본 콤플렉스 속성에 추가하여 대기열 및 호스트에 각각 사용 가능하게 됩니다.

대기열 및 호스트와 연관되는 사용자 정의 콤플렉스의 구체적인 값은 대기열 및 호스트 구성의 complex_values 매개변수에 의해 설정되어야 하며 그렇지 않으면 사용자 콤플렉스 구성의 value 필드가 사용됩니다.

예를 들어 다음 사용자 정의 콤플렉스 licenses를 정의합니다.

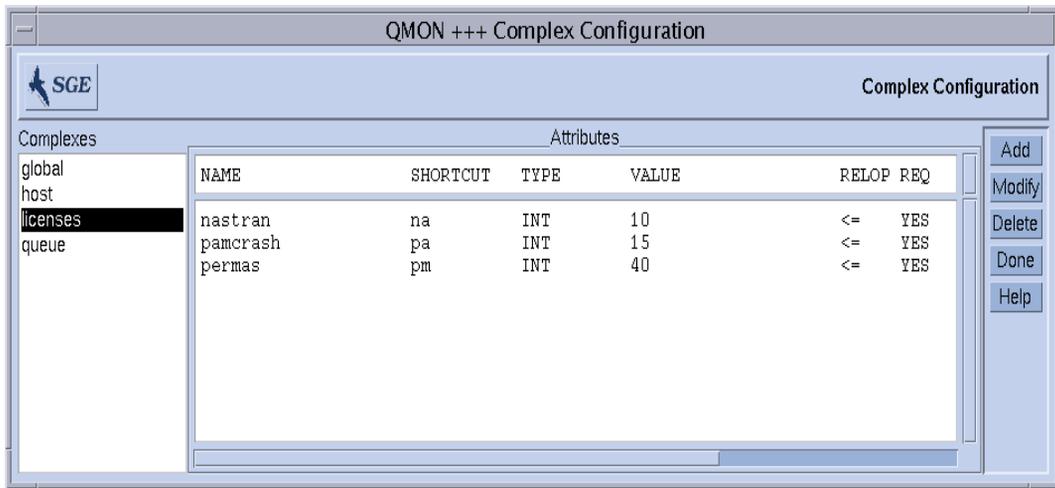


그림 8-5 Complex Configuration 대화 상자—Licenses

그리고 최소한 하나 이상의 대기열에 대해 licenses 콤플렉스를 그림 8-6에 표시되는 대기열 구성 User Complexes 부속 대화 상자에 표시된 것처럼 연관된 사용자 정의 콤플렉스의 목록에 추가합니다(대기열 구성 방법에 대한 자세한 내용은 157 페이지, “대기열 구성 정보” 및 그의 관련된 절 참조).

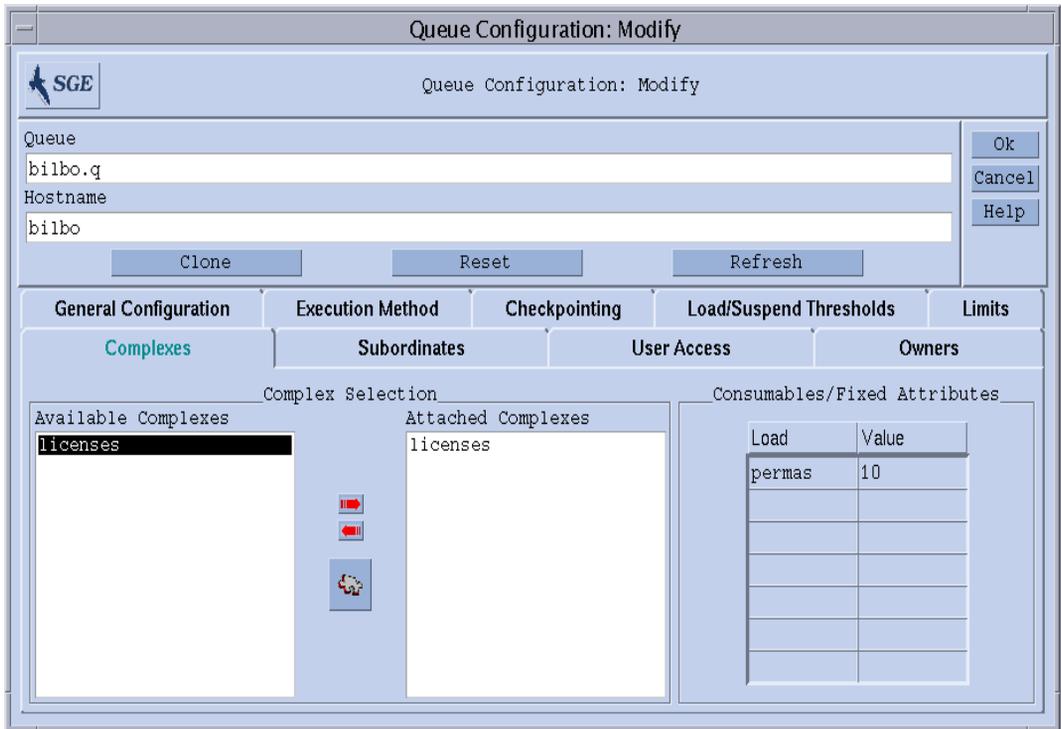


그림 8-6 사용자 정의 콤플렉스 대기열 구성

그러면 표시되는 대기열이 소프트웨어 패키지 `permas`의 최고 10개 사용권을 관리하도록 구성됩니다. 게다가 `licenses` 콤플렉스 속성 `permas`가 그림 8-7에 표시된 Submit 대화 상자의 Requested Resources 부속 대화 상자에서 Available Resources 목록에 표현되는 것처럼 Sun Grid Engine 작업에 대해 요청 가능하게 됩니다(작업 제출 방법에 대한 자세한 내용은 4장, 67 페이지, “작업 제출” 절 참조).

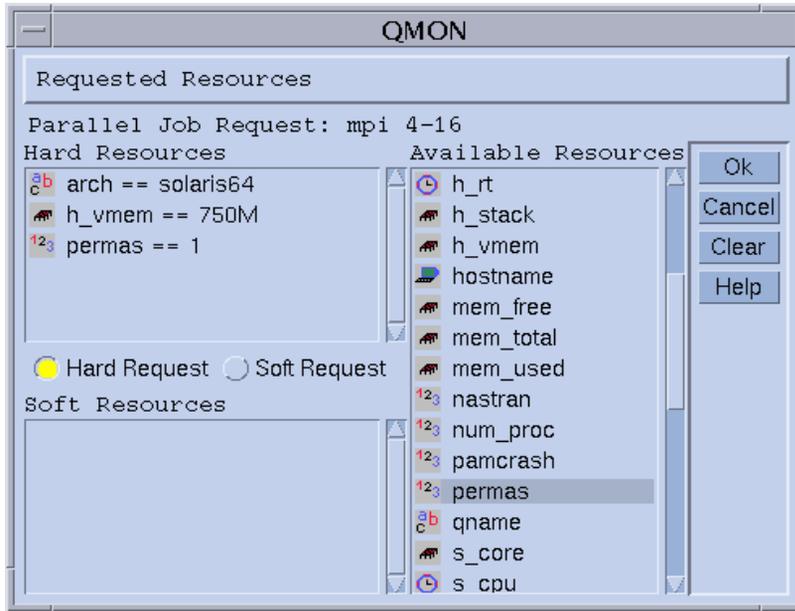


그림 8-7 Requested Resources 제출 부속 대화 상자

다른 방법으로는 사용자가 다음과 같이 명령줄에서 작업을 제출하고 licenses 속성을 요청할 수 있습니다.

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

참고 - 전체 속성 이름 permas 대신 pm 단축명을 사용할 수 있습니다.

그런 구성 및 유사한 작업 요청의 결과로서, 이들 작업에 적합한 유일한 대기열은 permas 사용권이 구성되고 사용 가능하게 하는 사용자 정의 licenses 콤플렉스와 연관되는 대기열입니다.

유효하지 않은 사용자 정의 콤플렉스 이름

다음은 예약되기 때문에 사용자 정의 콤플렉스 이름으로 지정될 수 없는 콤플렉스 이름의 목록입니다.

- global
- host
- queue

소비 가능 자원

소비 가능이라고도 부르는 소비 가능 자원은 사용 가능한 메모리, 파일 시스템의 여유 공간, 네트워크 대역폭 또는 부동 소프트웨어 사용권 같은 제한된 자원을 관리하기 위한 효율적인 수단입니다. 소비 가능의 총 사용 가능한 용량은 Sun Grid Engine 관리자가 정의하며 대응하는 자원의 소비는 Sun Grid Engine 내부 부기에 의해 모니터링됩니다. Sun Grid Engine는 모든 실행 중인 작업에 대한 이 자원의 소비를 고려하고 Sun Grid Engine 내부 부기가 충분히 사용 가능한 소비 가능 자원을 표시하는 경우에만 작업이 작업 지정되도록 보장합니다.

소비 가능은 기본 또는 사용자 정의 부하 매개변수(197 페이지, “부하 매개변수” 참조)와 결합될 수 있습니다. 즉, 부하 값이 소비 가능 속성에 대해 보고되거나 역으로 소비 가능 플래그가 부하 속성에 설정될 수 있습니다. Sun Grid Engine 소비 가능 자원 관리는 이 경우에 부하(자원의 가용성 측정) 및 내부 부기를 모두 고려하고 둘 중 어느 것도 주어진 한계를 초과하지 않게 합니다.

소비 가능 자원 관리를 사용하려면 자원의 총 용량을 정의해야 합니다. 이것은 클러스터 글로벌, 호스트당 및 대기열별로 수행될 수 있지만 이들 범주는 주어진 순서대로 서로를 대체할 수 있습니다(즉, 호스트가 클러스터 자원의 가용성을 제한할 수 있으며 대기열이 호스트 및 클러스터 자원을 제한할 수 있습니다). 자원 용량의 정의는 대기열 및 호스트 구성의 `complex_values` 항목으로 수행됩니다(157 페이지, “대기열 구성 정보” 및 138 페이지, “호스트 구성 정보”뿐 아니라 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `host_conf` 및 `queue_conf` 항목 참조). `global` 호스트의 `complex_values` 정의는 클러스터 글로벌 소비 가능 설정을 지정합니다. `Complex_values` 목록의 각 소비 가능 콤플렉스 속성에 해당 속성에 대한 최대 사용 가능 양을 표시하는 값이 지정됩니다. 내부 부기는 이 총량에서 작업의 자원 요청을 통해 표현되는 모든 실행 중인 작업에 의한 예상 자원 소비를 뺍니다.

▼ 소비 가능 자원을 설정하는 방법

숫자 콤플렉스 속성(유형 INT, MEMORY 및 TIME을 갖는 속성)만이 소비 가능으로 구성될 수 있습니다.

1. QMON Main 메뉴에서 **Complex Configuration** 단추를 누릅니다.

그림 8-1에 표시된 것과 비슷한 Complex Configuration 대화 상자가 표시됩니다.

2. 속성에 대해 Sun Grid Engine 소비 가능 관리를 켜려면 콤플렉스 구성에 있는 해당 속성에 대한 CONSUMABLE 플래그를 예를 들어 `virtual_free` 메모리 자원에 대해 그림 8-8에 묘사된 것처럼 설정합니다.

3. 다음 절에서 자세히 설명하는 보기를 바탕으로 다른 소비 가능 자원을 설정합니다.

- 187 페이지, “예 1:부동 소프트웨어 사용권 관리”
- 191 페이지, “예 2:가상 메모리를 위한 공간 공유”
- 194 페이지, “예 3:사용 가능한 디스크 공간 관리”

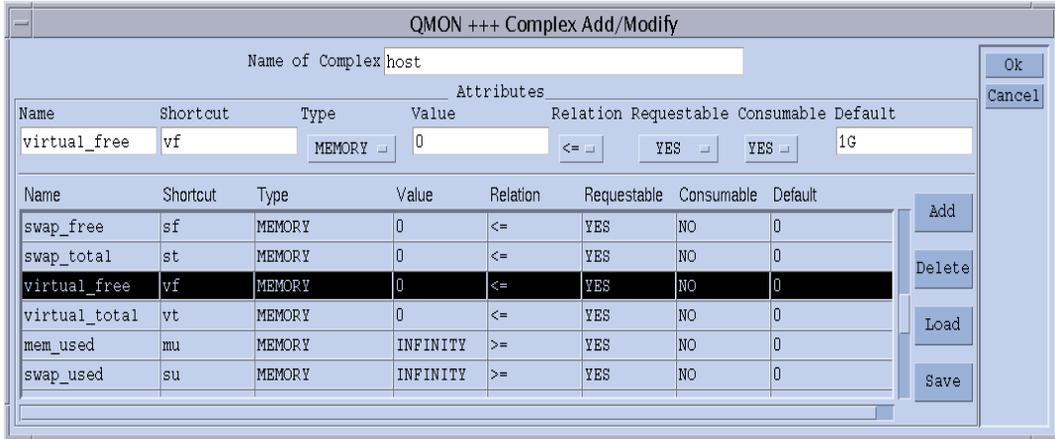


그림 8-8 Complex Configuration 대화 상자—virtual_free

그런 다음 Sun Grid Engine이 필요한 용량 계획을 수행하기 원하는 각 대기열 또는 각 호스트에 대해 complex_values 목록에 용량을 정의해야 합니다. 1GB의 가상 메모리가 현재 호스트의 용량 값으로 정의되는 예가 그림 8-9에 표시되어 있습니다.

해당 호스트에서(모든 대기열에 있는) 동시에 실행 중인 모든 작업의 가상 메모리 요구 사항이 누적되고 1GB의 용량에서 빠져 사용 가능한 가상 메모리를 판별합니다. virtual_free에 대한 작업 요청이 사용 가능한 양을 초과하는 경우 작업은 해당 호스트의 대기열에 작업 지정되지 않습니다.

참고 - 작업은 자원을 요청하고 Requestable 매개변수(그림 8-8 참조)의 강제 값을 통해 예상 소비를 지정하도록 강제될 수 있습니다.

참고 - 작업에 의해 명시적으로 요청되지 않는 소비 가능 속성에 대해 관리자가 기본 자원 소비 값을 사전 정의할 수 있습니다. (그림 8-8을 참조하십시오. 200MB가 기본값으로 설정됩니다.) 이것은 위에서 설명한 것처럼 자원 요청이 강제되지 않는 경우에만 의미가 있습니다.

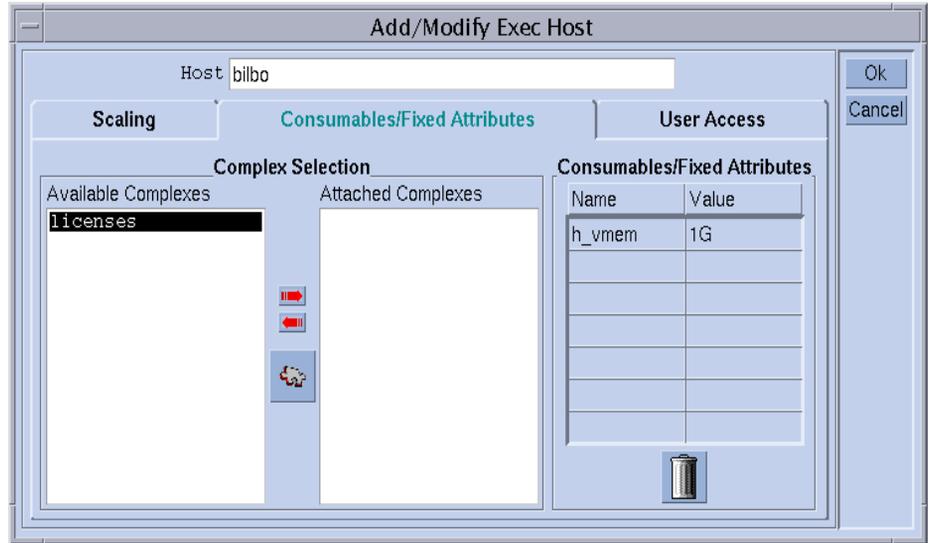


그림 8-9 실행 호스트 구성—virtual_free

소비 가능 자원 설정의 예

사용자 사이트에 대한 소비 가능 자원 설정을 익히려면 다음 예를 사용하십시오.

예 1: 부동 소프트웨어 사용권 관리

클러스터에서 소프트웨어 패키지 pam-crash를 사용 중이며 10개의 부동 사용권에 액세스할 수 있다고 가정하십시오. 즉, 소프트웨어의 총 활동 호출이 숫자 10을 초과하지 않는 한 모든 시스템에서 pam-crash를 사용할 수 있습니다. 목표는 모든 10개의 사용권이 다른 실행 중인 pam-crash 작업에 의해 점유되는 동안은 pam-crash 작업 스케줄링을 막는 방식으로 Sun Grid Engine을 구성하는 것입니다.

Sun Grid Engine 소비 가능 자원을 사용하면 이것을 쉽게 달성할 수 있습니다. 먼저, 그림 8-10에 표시된 것처럼 사용 가능한 pam-crash 사용권의 수를 Global 콤플렉스 구성에 소비 가능 자원으로 추가해야 합니다.

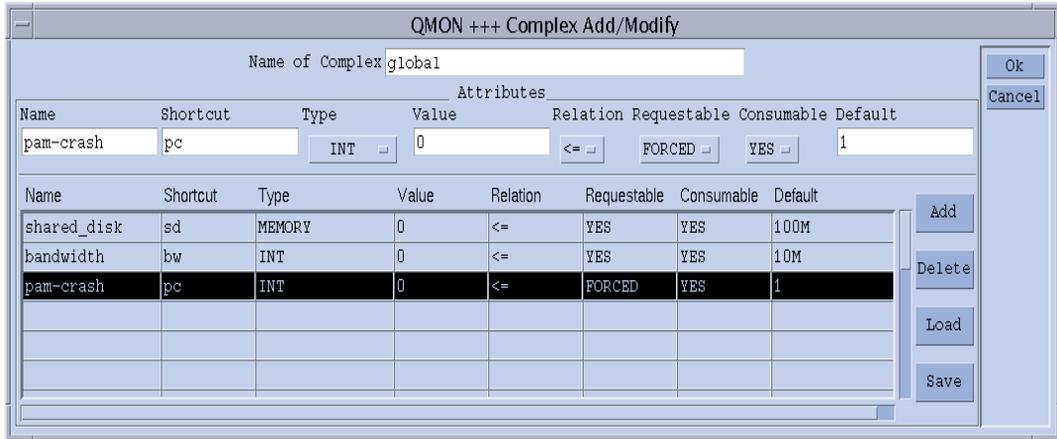


그림 8-10 Complex Configuration 대화 상자—pam-crash

소비 가능 속성의 이름이 pam-crash로 설정되고 pc를 qalter, qselect, qsh, qstat 또는 qsub -l 옵션에서 대신 단축명으로 사용할 수 있습니다. 속성 유형은 정수 카운터로서 정의됩니다. Value 필드의 설정은 complex_values 목록을 통해 글로벌, 호스트 또는 대기열 구성으로부터 값을 받기 때문에 소비 가능 자원의 경우에는 관계 없습니다(아래를 참조하십시오). Requestable 플래그는 사용자가 작업을 제출할 때 작업이 얼마나 많은 pam-crash 사용권을 점유할 것인지를 요청해야 함을 표시하기 위해 FORCED로 설정됩니다. 마지막으로 Consumable 플래그는 속성이 소비 가능 자원이도록 정의하는 반면 Default의 설정은 Requestable이 FORCED로 설정되기 때문에 관계 없고 따라서 요청 값은 임의의 작업에서 이 속성에 대해 수신됩니다.

이 속성 및 클러스터에 대해 자원 계획을 활성화하려면 그림 8-11에 표시되는 것처럼 사용 가능한 pam-crash 사용권의 수가 글로벌 호스트 구성에 정의되어야 합니다. 속성 pam-crash에 대한 값이 10개 부동 사용권에 따라서 10으로 설정됩니다.

참고 - Consumable/Fixed Attributes 표는 호스트 구성 파일 형식 host_conf에서 설명되는 complex_values 항목에 대응합니다.

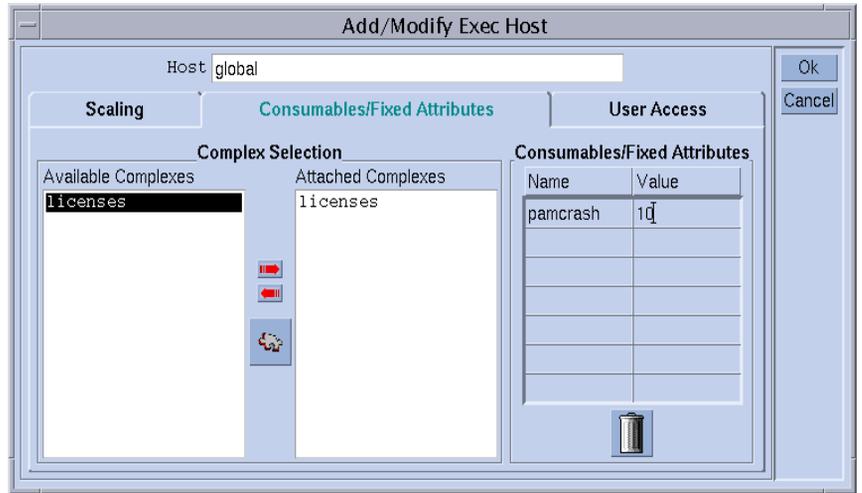


그림 8-11 Global Host Configuration—pam-crash

사용자가 다음 작업을 제출한다고 가정하십시오.

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

작업은 10개 미만의 pam-crash 사용권이 현재 점유되는 경우에만 시작됩니다. 그러나 작업은 클러스터의 어디에서나 실행할 수 있으며 실행하는 동안은 스스로가 하나의 pam-crash 사용권을 점유합니다.

클러스터에 있는 호스트 중 하나가 부동 사용권에 포함될 수 없는 경우(예를 들어 그에 대한 pam-crash 이진이 없기 때문에) 소비 가능 속성 pam-crash에 대해 이 호스트와 관련된 용량을 0으로 설정하여 pam-crash 사용권 관리에서 해당 호스트를 제외시킬 수 있습니다. 이것은 그림 8-12의 호스트 bilbo에 대해 표시될 것처럼 Execution Host Configuration 대화 상자에서 수행되어야 합니다.

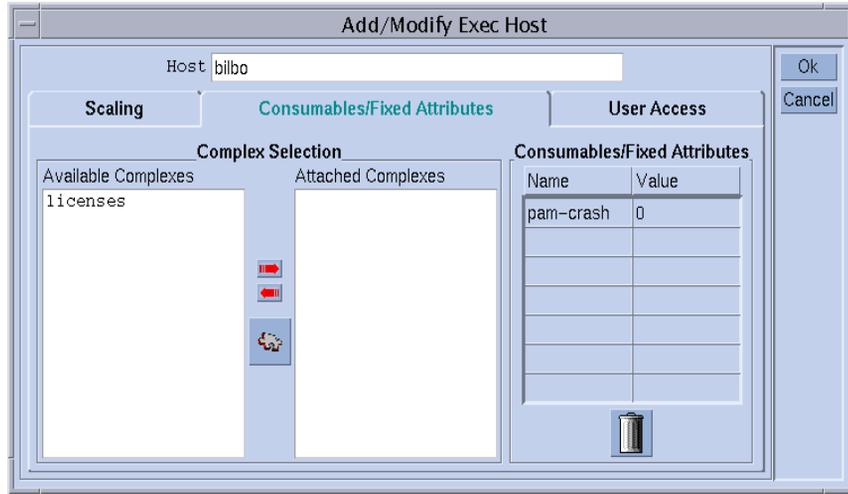


그림 8-12 Execution Host Configuration—pam-crash

참고 - global 콤플렉스의 속성이 모든 실행 호스트에 상속되기 때문에 pam-crash 속성은 내재적으로 실행 호스트에 사용 가능합니다. 마찬가지로, 용량을 0으로 설정하며 클러스터의 모든 사용권의 일부로서 특정 호스트에 의해 관리될 사용권 수를 0이 아닌 값(예: 2)으로 제한할 수도 있습니다. 이 경우 최대 2개의 pam-crash 작업이 해당 호스트에 공존할 수 있습니다.

유사하게 예를 들어 pam-crash에 적합하지 않은 메모리 및 CPU 시간 한계를 갖는 고속 대기열이기 때문에 특정 대기열이 pam-crash 작업을 실행하지 못하게 막기를 원할 수 있습니다. 이 경우 그림 8-13에 표시된 것처럼 대기열 구성에서 대응하는 용량을 0으로 설정하면 됩니다.

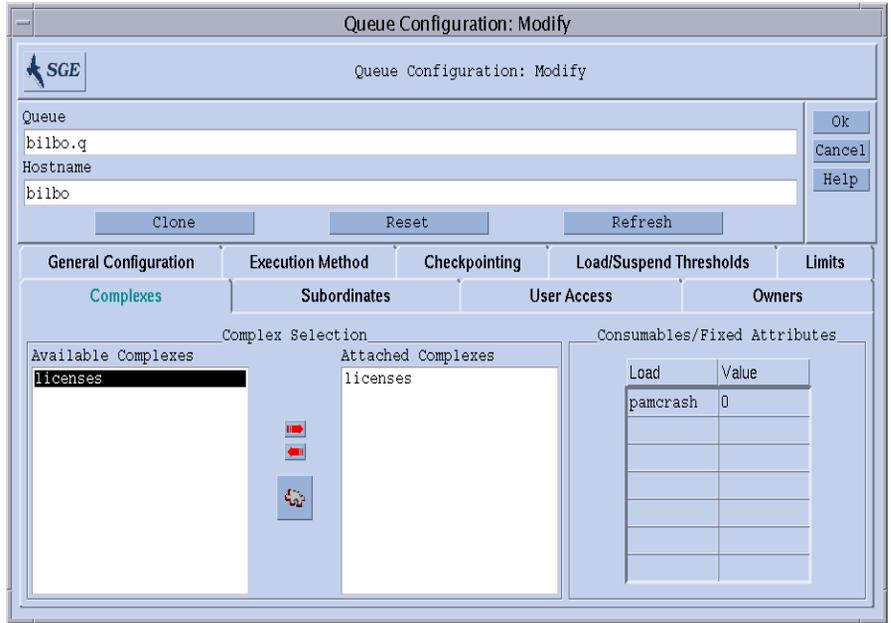


그림 8-13 Queue Configuration—pam-crash

참고 - global 콤플렉스의 속성이 모든 대기열에 상속되기 때문에 pam-crash 속성은 내재적으로 대기열에 사용 가능합니다.

예 2: 가상 메모리를 위한 공간 공유

시스템 관리자에 대한 공통적인 업무는 메모리 과다신청 및 그 결과인 시스템의 스왑핑에 의해 유발되는 성능 저하를 피하는 방식으로 시스템을 조정하는 것입니다. Sun Grid Engine 소프트웨어는 소비 가능 자원 기능을 통해 이 업무를 지원할 수 있습니다.

표준 부하 매개변수인 `virtual_free`는 사용 가능한 여유 가상 메모리, 즉 사용 가능한 스왑 공간과 사용 가능한 실제 메모리의 조합을 보고합니다. 스왑핑을 피하려면 스왑 공간의 사용을 최소화해야 합니다. 이상적인 경우에 호스트에서 실행하는 모든 프로세스에 필요한 모든 메모리가 실제 메모리와 맞아야 합니다.

Sun Grid Engine 소프트웨어는 다음 가정과 구성이 주어질 때 그를 통해 시작되는 모든 작업에 대해 이것을 보장할 수 있습니다.

- `virtual_free`가 소비 가능 자원으로 구성되고 각 호스트에서의 용량이 사용 가능한 실제 메모리(또는 그 이하)로 설정됩니다.
- 작업이 예상 메모리 사용을 요청하고 요청되는 값이 실행 중에 초과되지 않습니다.

가능한 호스트 콤플렉스 구성의 예가 그림 8-8에 표시되며 1GB의 주 메모리를 갖는 호스트에 대한 대응하는 실행 호스트 구성이 그림 8-9에 표시됩니다.

참고 - Requestable 플래그는 글로벌 콤플렉스 구성의 이전 예에 있는 FORCED와는 반대로 YES로 설정됩니다. 이것은 사용자가 작업의 메모리 요구사항을 표시할 필요는 없지만 명시적인 메모리 요청이 누락되는 경우 Default 필드의 값이 사용됨을 의미합니다. 이 경우에 기본 요청으로서의 값 1GB는 요청이 없는 작업이 모든 사용 가능한 실제 메모리를 점유하는 것으로 가정됨을 의미합니다.

참고 - `virtual_free`는 Sun Grid Engine의 표준 부하 매개변수 중 하나입니다. 가상 메모리 용량 계획에서 Sun Grid Engine이 자동으로 최근 메모리 통계의 추가 가용성을 고려합니다. 여유 가상 메모리에 대한 부하 보고서가 Sun Grid Engine 내부 부기에 의해 얻어지는 값 아래로 떨어지는 경우 메모리 과다신청을 피하기 위해 부하 값이 사용됩니다. 보고된 부하 값과 Sun Grid Engine 내부 부기 사이의 차이는 작업이 Sun Grid Engine을 사용하지 않고 시작되는 경우에 쉽게 발생할 수 있습니다.

단일 시스템에서 전형적인 서로 다른 메모리 요구사항을 갖는 여러 작업 클래스의 혼합을 실행하는 경우 이들 작업 클래스를 통한 사용을 위해 시스템의 메모리를 분할하기 원할 수 있습니다. 이 기능은 종종 *공간 공유*라고 부르며 각 작업 클래스에 대해 하나의 대기열을 구성하고 해당 호스트의 총 메모리의 일부를 대기열에 할당하여 수행할 수 있습니다.

예에서 그림 8-14에 표시된 대기열 구성은 bilbo 호스트에 사용 가능한 총 메모리의 절반(500 MB)을 `bilbo.q` 대기열에 첨부합니다. 따라서 `bilbo.q` 대기열에서 실행 중인 모든 작업의 누적 메모리 소비는 500MB를 초과할 수 없습니다. 다른 대기열의 작업은 고려되지 않지만 bilbo 호스트에서 실행 중인 모든 작업의 총 메모리 소비는 여전히 1GB를 초과할 수 없습니다.

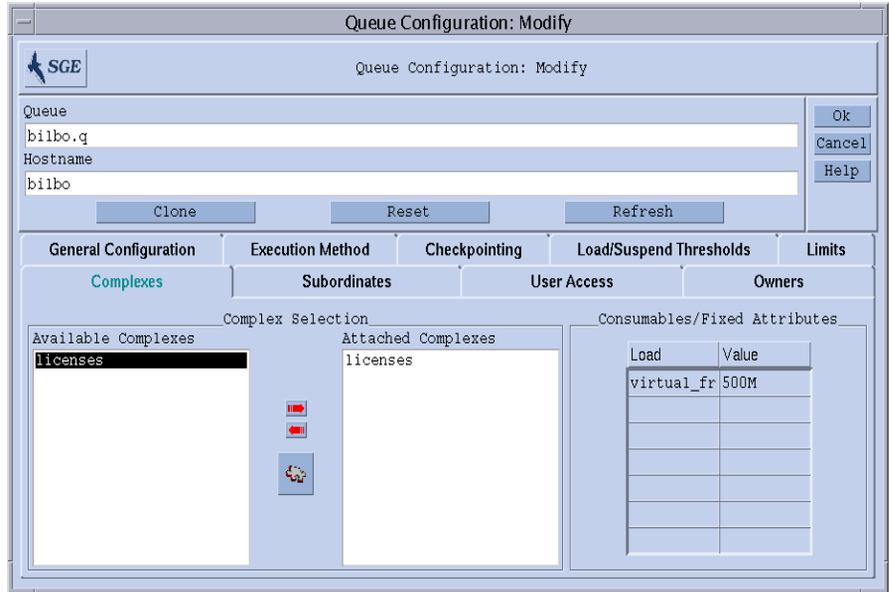


그림 8-14 Queue Configuration—virtual_free

참고 - virtual_free 속성은 호스트 콤플렉스로부터의 상속을 통해 모든 대기열에 사용 가능합니다.

사용자는 다음 양식 중 하나로 예제 경우와 유사하게 구성되는 시스템에 작업을 제출할 수 있습니다.

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

첫 번째 명령에 의해 제출되는 작업은 최소한 100MB의 메모리가 사용 가능할 때 바로 시작될 수 있으며 이 양은 virtual_free 소비 가능 자원에 대한 용량 계획에 반영됩니다. 두 번째 작업은 내재적으로 모든 사용 가능한 메모리를 요청하기 때문에 시스템에 작업이 없는 경우에만 실행합니다. 또한 대기열의 메모리 용량을 초과하기 때문에 bilbo.q 대기열에서는 실행할 수 없습니다.

예 3: 사용 가능한 디스크 공간 관리

일부 응용 프로그램은 파일에 저장되는 거대한 데이터 세트를 조작해야 하며 따라서 실행하는 동안 충분한 디스크 공간의 가용성에 종속됩니다. 이 요구사항은 이전 예에서 논의한 것처럼 사용 가능한 메모리의 공간 공유와 유사합니다. 주된 차이점은 Sun Grid Engine이 표준 부하 매개변수의 하나로서 여유 디스크 공간을 제공하지 않는다는 점입니다. 이것은 디스크가 대개 사이트에 고유한 방법으로 파일 시스템으로 파티션되며 이것은 관심있는 파일 시스템을 자동으로 식별할 수 없다는 사실에 기인합니다.

그럼에도 불구하고 소비 가능 자원 기능을 통해 Sun Grid Engine이 사용 가능한 디스크 공간을 효율적으로 관리할 수 있습니다. 이 절의 뒤에서 설명하는 이유 때문에 이 목적을 위해 호스트 콤플렉스 속성인 `h_fsize`를 사용하는 것이 바람직합니다. 첫 번째로, 이 속성이 예를 들어 그림 8-15에 표시된 것처럼 소비 가능 자원으로 구성되어야 합니다.

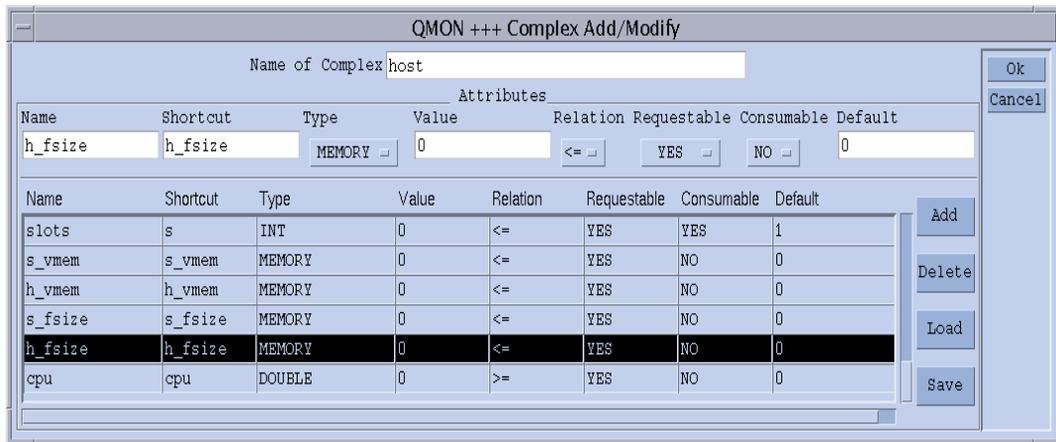


그림 8-15 Complex Configuration—`h_fsize`

호스트 로컬 파일 시스템을 가정할 때 그림 8-16에 표시된 것처럼 디스크 공간에 대한 용량 정의를 호스트 구성에 소비 가능으로 두는 것이 타당합니다.

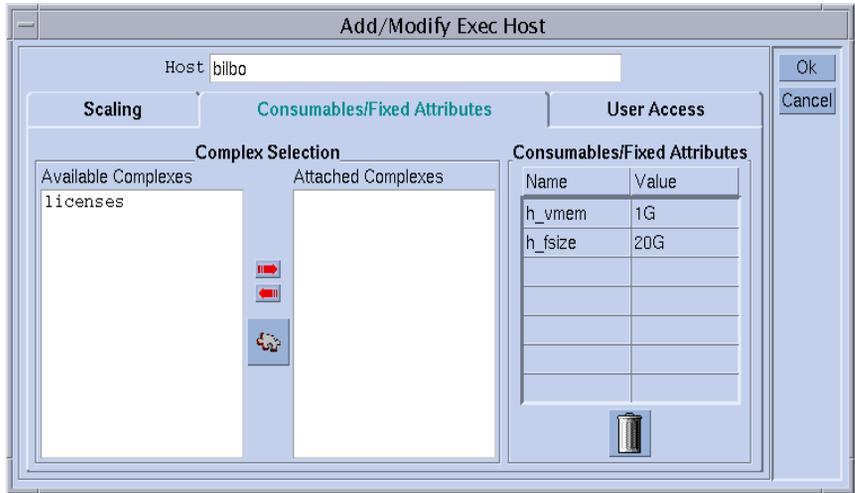


그림 8-16 Execution Host Configuration—h_fsize

그런 방법으로 구성된 Sun Grid Engine 시스템에 작업을 제출하면 이전 예와 유사하게 작동합니다.

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

이 예에서 h_fsize 속성이 권장된 이유는 h_fsize가 대기열 구성에서 *하드 파일 크기 한계*로도 사용된다는 사실에 있습니다. 파일 크기 한계는 작업 제출 중에 지정되는 값(위의 예에서는 20GB) 또는 작업이 해당 속성을 요청하지 않는 경우 대기열 구성의 대응하는 값보다 큰 파일을 작성하는 작업의 능력을 제한하는 데 사용됩니다. h_fsize에 대한 Requestable 플래그가 예에서 FORCED로 설정되었으므로, 요청이 항상 존재합니다.

대기열 한계를 소비 가능 자원으로 사용함으로써 자동적으로 작업 스크립트에 의한 실제 자원 소비와 대조하여 사용자가 지정하는 대로 요청에 대한 제어를 얻습니다. 한계의 모든 위반이 용인되고 작업은 결국 중단됩니다(자세한 내용은 queue_conf 및 setrlimit 매뉴얼 페이지를 참조하십시오). 이 방법으로 Sun Grid Engine 내부 용량 계획이 기반으로 하는 자원 요청이 신뢰할 수 있음을 보장할 수 있습니다.

참고 - 일부 운영 체제는 프로세스당 파일 크기 한계만을 제공합니다. 이 경우 작업은 최대 한계까지의 크기를 갖는 복수 파일을 작성할 수 있습니다. 작업당 파일 크기 한계를 지원하는 시스템에서는 Sun Grid Engine이 이 기능을 h_fsize 속성과 함께 사용합니다(자세한 내용은 queue_conf 매뉴얼 페이지를 참조하십시오).

응용 프로그램이 동시에 디스크 공간을 점유하기 위해 Sun Grid Engine에 제출되지 않을 것으로 예상하는 경우 Sun Grid Engine 내부 부기가 디스크 공간 부족으로 인한 응용 프로그램 실패를 막기에 충분하지 않을 수 있습니다. 이 문제점을 피하기 위해 주기적으로 디스크 공간 사용 통계를 받는 것이 도움이 될 수 있는데, 이 통계는 Sun Grid Engine 외부에서 발생하는 것을 포함하여 총 디스크 공간 소비를 표시합니다.

Sun Grid Engine 부하 센서 인터페이스(198 페이지, “사이트 고유 부하 매개변수 추가” 참조)를 사용하여 특정 파일 시스템의 사용 가능한 디스크 공간 같은 사이트 고유 정보로 표준 Sun Grid Engine 부하 매개변수 세트를 향상시킬 수 있습니다.

적당한 부하 센서를 추가하고 `h_fsize`에 대한 여유 디스크 공간을 보고하여 소비 가능 자원 관리와 자원 가용성 통계를 결합시킬 수 있습니다. Sun Grid Engine은 디스크 공간에 대한 작업 요구사항을 Sun Grid Engine 내부 자원 계획에서 파생되는 사용 가능한 용량 및 가장 최근에 보고된 부하 값과 비교합니다. 양 기준이 모두 만족되는 경우에만 작업이 호스트에 작업 지정됩니다.

컴플렉스 구성

Sun Grid Engine 콤플렉스는 176 페이지, “컴플렉스 구성을 추가 또는 수정하는 방법” 절 및 그 이후에 표시되고 설명되는 QMON Complex Configuration 대화 상자를 통해 정의 및 유지보수하거나 명령줄에서 수행할 수 있습니다.

▼ 명령줄에서 콤플렉스 구성을 수정하는 방법

다음 명령과 적당한 옵션을 입력하십시오.

```
% qconf 옵션
```

`qconf` 명령 형식과 유효한 값 필드 구문에 대한 자세한 정의에 대해서는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 콤플렉스 항목이나 콤플렉스 매뉴얼 페이지를 참조하십시오.

유용한 옵션에는 다음이 포함됩니다.

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

`qconf -Ac` 및 `-Mc` 옵션이 콤플렉스 구성 파일을 인수로 취하는 반면, `-ac` 및 `-mc` 옵션은 템플릿 콤플렉스 구성 또는 수정할 기본 콤플렉스의 구성으로 채워진 편집기를 엽니다.

옵션의 의미는 다음과 같습니다.

■ `qconf -Ac, -ac`

사용 가능한 콤플렉스 목록에 새 콤플렉스를 추가합니다.

■ `qconf -Mc, -mc`

기존 콤플렉스를 수정합니다.

qconf 명령의 예

다음 명령

```
% qconf -sc licenses
```

은 `complex (5)` 매뉴얼 페이지에 정의되는 파일 형식으로 `nastran` 콤플렉스(그림 8-5에 정의되는)를 표준 출력 스트림에 인쇄합니다. `licenses` 콤플렉스에 대한 샘플 출력이 표 8-1에 표시되어 있습니다.

#name	shortcut	type	value	relop	requestable	consumable	default
nastran	na	INT	10	<=	YES	NO	0
pam-crash	pc	INT	15	<=	YES	YES	1
permas	pm	INT	40	<=	FORCED	YES	1

#---- # start a comment but comments are not saved across edits -----

표 8-1 qconf -sc 샘플 출력

부하 매개변수

이 절은 부하 매개변수의 Sun Grid Engine 5.3 개념을 설명하며, 사용자 자신의 부하 센서 작성에 대한 지침을 포함합니다.

기본 부하 매개변수

기본적으로 `sge_execd`는 주기적으로 여러 부하 매개변수 및 대응하는 값을 `sge_qmaster`에 보고합니다. 이들은 `sge_qmaster` 내부 호스트 오브젝트에 저장됩니다(137 페이지, “데몬 및 호스트 정보” 절 참조). 그러나 대응하는 이름을 갖는 콤플렉스

속성이 정의되는 경우에만 내부적으로 사용됩니다. 그런 콤플렉스 속성에는 부하 값을 해석하는 방법에 대한 정의가 들어 있습니다(자세한 내용은 177 페이지, “콤플렉스 유형” 절 참조).

1차 설치 후에 부하 매개변수의 표준 세트가 보고됩니다. 표준 부하 매개변수에 대해 필요한 모든 속성은 호스트 콤플렉스에 정의됩니다. Sun Grid Engine의 후속 릴리스는 기본 부하 매개변수의 확장된 세트를 제공할 수 있습니다. 그러므로 기본적으로 보고되는 부하 매개변수의 세트가 <sgeroot>/doc/load_parameters.asc 파일에 문서화됩니다.

참고 - 부하 속성이 정의되는 콤플렉스가 속성의 액세스 가능성을 결정합니다. 글로벌 콤플렉스에 부하 매개변수를 정의하면 전체 클러스터 및 모든 호스트에 대해 사용 가능하게 됩니다. 호스트 콤플렉스에 정의하면 모든 호스트에 대해 속성이 제공되지만 클러스터에 글로벌하게 제공되지는 않습니다. 사용자 정의 콤플렉스에 정의하면 사용자 콤플렉스를 호스트에 추가 또는 제거하여 부하 매개변수의 표시를 제어할 수 있습니다.

참고 - 대기열 콤플렉스가 어떤 호스트나 클러스터에도 사용 가능하지 않기 때문에 부하 속성을 대기열 콤플렉스에 정의해서는 안됩니다.

사이트 고유 부하 매개변수 추가

기본 부하 매개변수 세트가 클러스터의 경우에, 특히 사이트 고유 정책, 응용 프로그램 및 구성의 부하 상황을 완벽하게 기술하기에 충분하지 않을 수 있습니다. 그러므로 Sun Grid Engine 소프트웨어는 임의의 방법으로 부하 매개변수 세트를 확장하는 방법을 제공합니다. 이를 위해 `sgexecd`는 `sgexecd`에 현재 부하 값과 함께 부하 매개변수를 공급하는 인터페이스를 제공합니다. 그 이후에 이들 매개변수는 정확히 기본 부하 매개변수처럼 취급됩니다. 기본 부하 매개변수의 경우와 마찬가지로(197 페이지, “기본 부하 매개변수” 절 참조) 부하 매개변수가 효력을 갖기 위해서는 대응하는 속성이 부하 콤플렉스에 정의되어야 합니다.

▼ 사용자 자신의 부하 센서를 작성하는 방법

추가 부하 정보를 `sgexecd`에 제공하려면 *부하 센서*를 공급해야 합니다. 부하 센서는 스크립트 또는 이진 실행 파일일 수 있습니다. 어느 경우나 부하 센서의 표준 입력 및 출력 스트림의 처리 및 제어 흐름이 다음 규칙을 준수해야 합니다.

부하 센서는 특정 시점에서 STDIN으로부터의 입력을 기다리는 무한 루프로서 작성되어야 합니다. 문자열 `quit`가 STDIN에서 입력되는 경우 부하 센서는 종료하는 것으로 가정됩니다. STDIN에서 행 끝을 읽자마자 부하 데이터 검색 주기가 시작하는 것으로 가정됩니다. 그런 다음 부하 센서는 원하는 부하 수치를 연산하기 위해 필요한 모든 조작을 수행합니다. 주기의 끝에서 부하 센서는 그 결과를 `stdout`에 기록합니다.

규칙

형식은 다음과 같습니다.

- 부하 값 보고서는 단어 `begin`만을 포함하는 행으로 시작합니다.
- 개별 부하 값은 새 행으로 구분됩니다.
- 각 부하 값 정보는 콜론(:)으로 구분되는 세 부분으로 구성되며 공백을 포함하지 않습니다.
- 부하 값 정보의 첫 번째 부분은 부하가 보고되는 호스트의 이름 또는 특수 이름인 `global` 중 하나입니다.
- 두 번째 부분은 호스트 또는 글로벌 콤플렉스 목록에 정의되는 것과 같은 부하 값의 기호 이름입니다(자세한 내용은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `complex(5)` 항목 참조). 부하 값이 호스트 또는 글로벌 콤플렉스 목록에 존재하지 않는 항목에 대해 보고되는 경우 보고된 부하 값은 사용되지 않습니다.
- 세 번째 부분은 측정된 부하 값입니다.
- 부하 값 보고서는 단어 `end`를 갖는 행으로 끝납니다.

스크립트의 예

코드 예 8-1은 Bourne 셸 스크립트 부하 센서의 예입니다.

```
#!/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
```

코드 예 8-1

Bourne 셸 스크립트 부하 센서

이 예가 load.sh 파일에 저장되고 실행 가능 권한이 chmod를 통해 할당되는 경우 load.sh를 호출하고 키보드의 Return 키를 반복적으로 눌러서 명령줄에서 대화식으로 스크립트를 테스트할 수 있습니다.

절차가 작동하자마자 부하 센서의 경로를 클러스터, 글로벌 또는 실행 호스트 고유의 구성에 대한 load_sensor 매개변수로 구성하여 임의의 실행 호스트에 대해 설치할 수 있습니다(151 페이지, “기본 클러스터 구성” 절이나 sge_conf 매뉴얼 페이지를 참조하십시오).

대응하는 QMON 화면은 그림 8-17의 보기처럼 보일 수 있습니다.

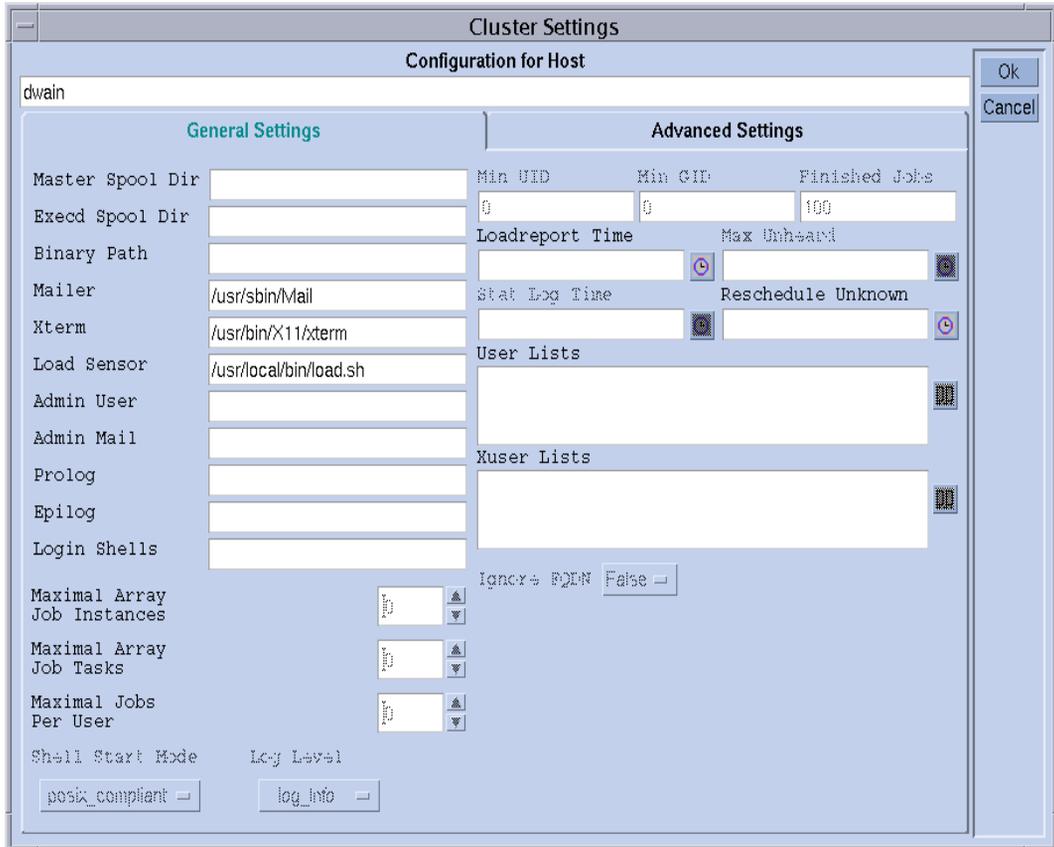


그림 8-17 부하 센서를 갖는 로컬 구성

대응하는 속성이 호스트 콤플렉스에 추가되자마자 보고된 부하 매개변수인 logins가 사용 가능합니다. 필수 정의는 QMON Complex Configuration 화면의 예인 그림 8-18에 있는 마지막 표 항목과 비슷하게 보일 수 있습니다.

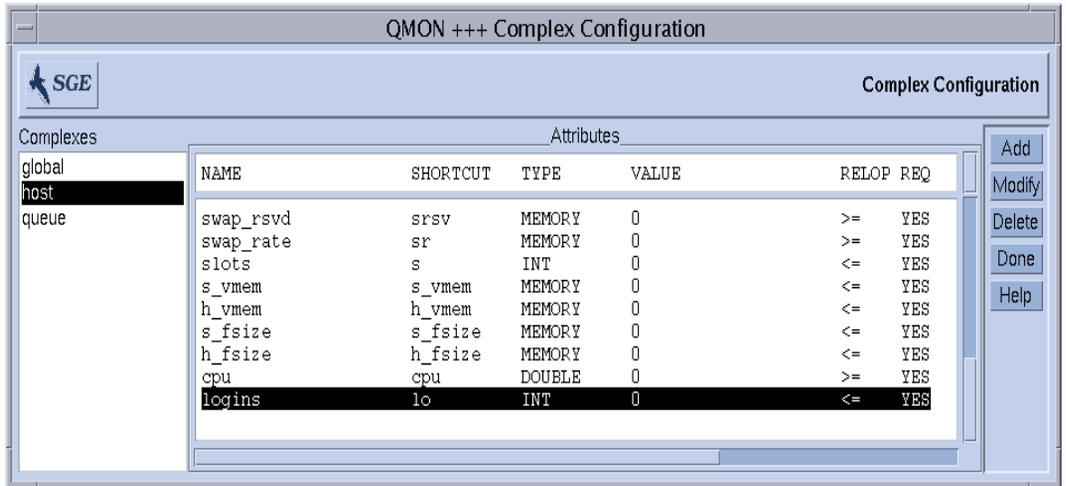


그림 8-18 Complex Configuration 대화 상자—logins

사용자 액세스 및 정책 관리

이 장에는 Sun Grid Engine 시스템에서 사용자 및 관련된 계정과 정책의 관리와 관계되는 중요한 정보가 들어 있습니다. 이 장의 주제에는 사용자 액세스, 스케줄링, 경로 별명 지정, 기본 요청, 회계 및 이용 통계 및 검사점에 대한 지원이 포함됩니다.

이 장에는 백그라운드 정보 외에 다음 업무를 수행하는 방법에 대한 자세한 지침이 들어 있습니다.

- 205 페이지, “QMON을 사용하여 계정을 구성하는 방법”
- 206 페이지, “QMON을 사용하여 관리자 계정을 구성하는 방법”
- 207 페이지, “명령줄에서 관리자 계정을 구성하는 방법”
- 207 페이지, “QMON을 사용하여 운영자 계정을 구성하는 방법”
- 208 페이지, “명령줄에서 운영자 계정을 구성하는 방법”
- 210 페이지, “QMON을 사용하여 사용자 액세스 목록을 구성하는 방법”
- 212 페이지, “명령줄에서 사용자 액세스 목록을 구성하는 방법”
- 218 페이지, “QMON을 사용하여 스케줄러 구성을 변경하는 방법”
- 227 페이지, “QMON을 사용하여 검사점 환경을 구성하는 방법”
- 230 페이지, “명령줄에서 검사점 환경을 구성하는 방법”

사용자 설정

다음 목록은 Sun Grid Engine에 대한 사용자를 설정하기 위해 필요한/사용 가능한 업무를 설명합니다.

■ 필수 로그인

A 호스트에서 *B* 호스트에서 실행하기 위한 작업을 제출하려면 사용자는 *A* 및 *B* 호스트에서 동일한 계정(즉, 동일한 사용자 이름)을 가져야 합니다. `sge_qmaster`가 실행하는 시스템에서의 로그인은 필요없습니다.

■ Sun Grid Engine 액세스 권한 설정

Sun Grid Engine 소프트웨어는 사용자 액세스를 전체 클러스터, 대기열 및 병렬 환경으로 제한하는 기능을 제공합니다. 자세한 설명은 209 페이지, “사용자 액세스 권한” 절을 참조하십시오.

또한 Sun Grid Engine 시스템 사용자는 특정 대기열을 일시중단 또는 활성화하는 권한을 얻을 수 있습니다(자세한 정보는 168 페이지, “Owners 구성 방법” 참조).

■ 파일 액세스 제한

Sun Grid Engine 사용자는 <sgе 루트>/cell/common 디렉토리에 대한 읽기 액세스가 있어야 합니다.

Sun Grid Engine 작업이 시작되기 전에 Sun Grid Engine 실행 데몬(*root*로서 실행)이 작업에 대한 임시 작업 디렉토리를 작성하고 해당 디렉토리의 소유권을 작업 소유자로 변경합니다(임시 디렉토리는 작업이 끝나자마자 제거됩니다). 임시 작업 디렉토리는 대기열 구성 매개변수 `tmpdir`에 의해 정의되는 경로에 작성됩니다(자세한 정보는 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `queue_conf` 항목 참조).

임시 디렉토리가 `tmpdir` 위치에 작성될 수 있고 Sun Grid Engine 사용자 소유권으로 설정될 수 있으며 사용자가 그 이후에 임시 디렉토리에 쓸 수 있는지 확인하십시오.

■ 사이트 종속성

정의에 의해 일괄처리 작업은 터미널 연결을 갖지 않습니다. 따라서 명령 해석기 시작 자원 파일(예: `csh`의 경우 `.cshrc`)의 `stty` 같은 UNIX 명령은 오류를 유발할 수 있습니다. 46 페이지, “설치 검증 방법”에서 설명한 것처럼 발생을 확인하고 그런 명령을 피하십시오.

Sun Grid Engine 일괄처리 작업이 대개 오프라인으로 실행되기 때문에 오류 이벤트 및 비슷한 이벤트에 대해 작업 소유자에게 통지하는 두 가지 방법만이 있습니다. 한 가지 방법은 파일에 오류 메시지를 기록하는 것이고 다른 하나는 전자 우편(이메일)을 전송하는 것입니다. 아주 가끔은(예를 들어 오류 로그 파일을 열 수 없는 경우), 이메일이 사용자에게 직접 통지할 수 있는 유일한 방법입니다(이와 같은 오류 메시지는 Sun Grid Engine 시스템 로그파일에 어떤 방식으로든 기록되지만 대개는 사용자가 시스템 로그파일을 조사하지 않습니다). 그러므로 전자 우편 시스템이 Sun Grid Engine 사용자에게 적절하게 설치되는 경우는 유익합니다.

■ Sun Grid Engine 정의 파일

Sun Grid Engine 사용자에게 대해 다음 정의 파일을 설정할 수 있습니다.

- `qmon` (Sun Grid Engine GUI에 대한 자원 파일은 9 페이지, “QMON 사용자 정의” 절을 참조하십시오)
- `sgе_aliases` (현재 작업 디렉토리 경로 별명, 221 페이지, “경로 별명 지정” 절 참조)
- `sgе_request` (기본 요청 정의 파일, 223 페이지, “가본 요청 구성” 절 참조).

사용자 액세스 정보

세 가지 사용자 범주가 Sun Grid Engine 5.3 시스템에 존재합니다.

- **관리자** - 관리자는 Sun Grid Engine을 조작할 모든 능력을 갖습니다. 기본적으로 마스터 호스트 및 대기열을 호스트하는 모든 시스템의 슈퍼유저는 관리자 특권을 갖습니다.
- **운영자** - 운영자는 대기열을 추가, 삭제 또는 수정할 수 있는 점을 제외하고는 많은 관리자와 동일한 명령을 수행할 수 있습니다.
- **소유자** - 대기열 소유자는 소유한 대기열의 일시중단/일시중단 해제 또는 사용 안함/사용으로 제한됩니다. 이들 특권은 qidle의 성공적인 사용을 위해 필요합니다. 사용자는 일반적으로 자신의 데스크탑 워크스테이션에 상주하는 대기열의 소유자인 것으로 선언됩니다.

각 범주는 후속 절에 더 자세히 설명되어 있습니다.

▼ QMON을 사용하여 계정을 구성하는 방법

1. QMON 주 메뉴에서 **User Configuration** 단추를 누릅니다.
2. 사용자가 수행하려는 작업에 따라서 다음 탭 선택기 중 하나를 누릅니다.
 - Manager Account Configuration (그림 9-1 참조)
 - Operator Account Configuration (그림 9-2 참조)
 - User Access List Configuration (그림 9-3 참조)
3. 다음 절의 지침에 따라서 계속합니다.

참고 - User Configuration 단추를 처음으로 누를 때 기본적으로 Manager Account Configuration 대화 상자가 열립니다.

▼ QMON을 사용하여 관리자 계정을 구성하는 방법

Manager 탭을 선택할 때 Manager Configuration 대화 상자(그림 9-1 참조)가 표시되고 그 곳에서 어떤 계정이 모든 관리 Sun Grid Engine 명령을 실행하도록 허용되는지를 선언할 수 있습니다. 화면의 하단부에 있는 선택 목록은 관리 권한을 갖는 것으로 이미 선언된 계정을 표시합니다.

- **삭제** - 계정 이름을 누른 후 대화 상자의 오른쪽에 있는 Delete 단추를 눌러서 이 목록에서 기존 관리자 계정을 삭제하십시오.
- **추가** - 선택 목록 위에 있는 입력 창에 계정 이름을 입력한 후 Add 단추를 누르거나 키보드의 Return 키를 눌러서 새 관리자 계정을 추가하십시오.

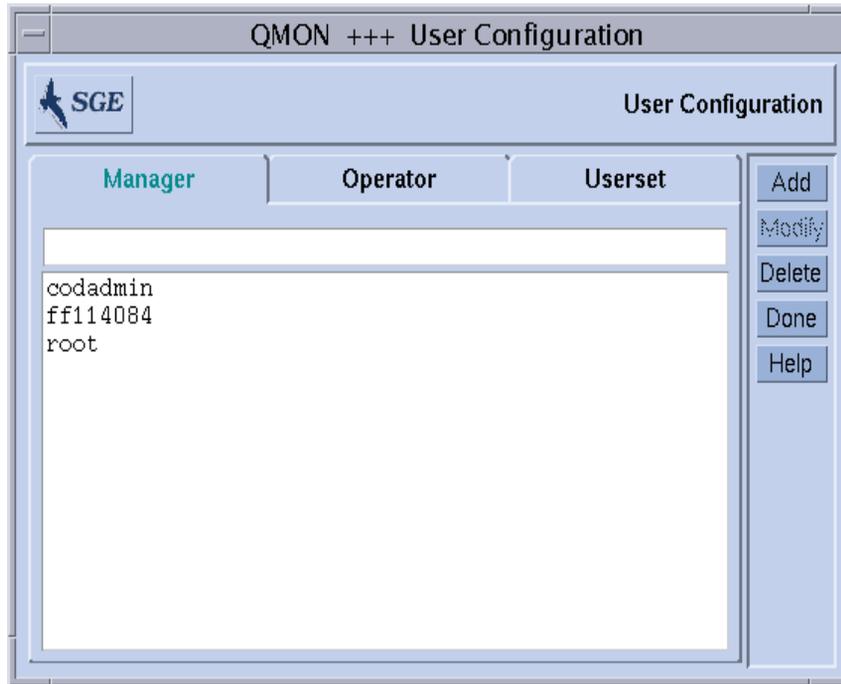


그림 9-1 Manager Configuration 대화 상자

▼ 명령줄에서 관리자 계정을 구성하는 방법

- 적당한 스위치와 함께 다음 명령을 입력하십시오.

```
# qconf 스위치
```

사용 가능한 스위치

- `qconf -am 사용자이름[...]`

관리자 추가 - 이 명령은 Sun Grid Engine 관리자 목록에 하나 또는 복수의 사용자를 추가합니다. 기본적으로 모든 Sun Grid Engine 신뢰 호스트(137 페이지, “데몬 및 호스트 정보” 절 참조)의 루트 계정은 Sun Grid Engine 관리자입니다.

- `qconf -dm 사용자이름[...]`

관리자 삭제 - 이 명령으로 Sun Grid Engine 관리자 목록에서 지정된 사용자를 삭제합니다.

- `qconf -sm`

관리자 표시 - 이 명령은 모든 Sun Grid Engine 관리자의 목록을 표시합니다.

▼ QMON을 사용하여 운영자 계정을 구성하는 방법

Operator 탭을 선택할 때 Operator Configuration 대화 상자가 표시되고(그림 9-2 참조) 그곳에서 어떤 계정이 제한된 관리 Sun Grid Engine 명령 권한을 갖도록 허용되는지를 선언할 수 있습니다(해당 계정이 관리자 계정인 것으로도 선언되지 않는 경우, 206 페이지, “QMON을 사용하여 관리자 계정을 구성하는 방법” 참조). 화면의 하단부에 있는 선택 목록은 운영자 권한을 갖는 것으로 이미 선언된 계정을 표시합니다.

- **삭제** - 계정 이름을 누른 후 대화 상자의 오른쪽에 있는 Delete 단추를 눌러서 이 목록에서 기존 운영자 계정을 삭제하십시오.
- **추가** - 선택 목록 위에 있는 입력 창에 계정 이름을 입력한 후 Add 단추를 누르거나 키보드의 Return 키를 눌러서 새 운영자 계정을 추가하십시오.

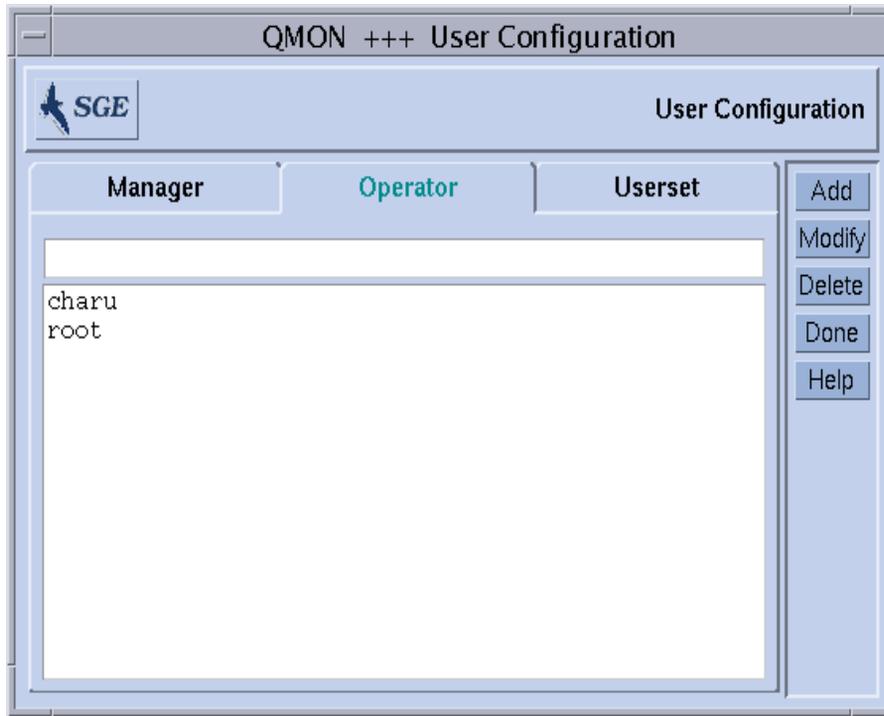


그림 9-2 Operator Configuration 대화 상자

▼ 명령줄에서 운영자 계정을 구성하는 방법

- 적당한 스위치와 함께 다음 명령을 입력하십시오.

```
# qconf 스위치
```

사용 가능한 스위치

- `qconf -ao 사용자이름[...]`

운영자 추가 - 이 명령은 Sun Grid Engine 운영자 목록에 하나 또는 복수 사용자를 추가합니다

- `qconf -do 사용자이름[...]`

운영자 삭제 - 이 명령은 Sun Grid Engine 운영자 목록에서 지정된 사용자를 삭제합니다

■ qconf -so

운영자 표시 - 이 명령은 모든 Sun Grid Engine 운영자의 목록을 표시합니다

대기열 소유자 계정 정보

대기열 소유자는 Sun Grid Engine 대기열의 구성 또는 수정 중에 정의됩니다. 158 페이지, “QMON을 사용하여 대기열을 구성하는 방법” 및 169 페이지, “명령줄에서 대기열을 구성하는 방법” 절을 참조하십시오. 대기열의 소유자는 다음을 수행할 수 있습니다.

- **일시중단**—대기열에서 실행 중인 모든 작업의 실행을 중지하고 대기열을 닫습니다.
- **일시중단 해제**—대기열에서 실행을 재개하고 대기열을 엽니다.
- **사용 안함**—대기열을 닫지만 실행 중인 작업에는 영향을 주지 않습니다
- **사용**—대기열을 엽니다.

참고 - 대기열이 일시중단된 동안 명시적으로 일시중단된 작업은 대기열이 일시중단 해제될 때 실행을 재개하지 않습니다. 해당 작업은 명시적으로 일시중단 해제되어야 합니다.

일반적으로 사용자가 중요한 작업을 위해 수시로 특정 시스템이 필요한 경우 및 백그라운드에서 실행 중인 Sun Grid Engine 작업에 의해 크게 영향을 받는 경우 특정 대기열의 소유자인 것으로 설정됩니다.

사용자 액세스 권한

최소한 하나의 제출 호스트 및 실행 호스트에서 유효한 로그인을 갖는 모든 사용자는 Sun Grid Engine 시스템을 사용할 수 있습니다. 그러나 Sun Grid Engine 관리자가 특정 또는 모든 대기열에 대한 특정 사용자의 액세스를 금지할 수 있습니다. 게다가 특정 병렬 환경(233 페이지, “병렬 환경 정보” 절 참조)과 같은 기능의 사용도 제한할 수 있습니다.

액세스 권한 정의를 위해서는 사용자의 이름 지정된 임의의 중첩 또는 비 중첩 세트로 구성되는 *사용자 액세스 목록*이 정의되어야 합니다. 사용자 이름과 UNIX 그룹 이름을 사용하여 사용자 액세스 목록을 정의할 수 있습니다. 그런 다음 사용자 액세스 목록을 클러스터 구성(151 페이지, “기본 클러스터 구성” 절 참조), 대기열 구성(166 페이지, “중속 대기열 구성 방법” 절 참조) 또는 병렬 환경 인터페이스 구성의 프로세스에서(234 페이지, “QMON을 사용하여 PE를 구성하는 방법” 절 참조) 사용하여 특정 자원에 대한 액세스를 거부 또는 허용합니다.

▼ QMON을 사용하여 사용자 액세스 목록을 구성하는 방법

Userset 탭을 선택할 때 그림 9-3의 보기와 유사한 Userset Configuration 대화 상자가 표시됩니다.

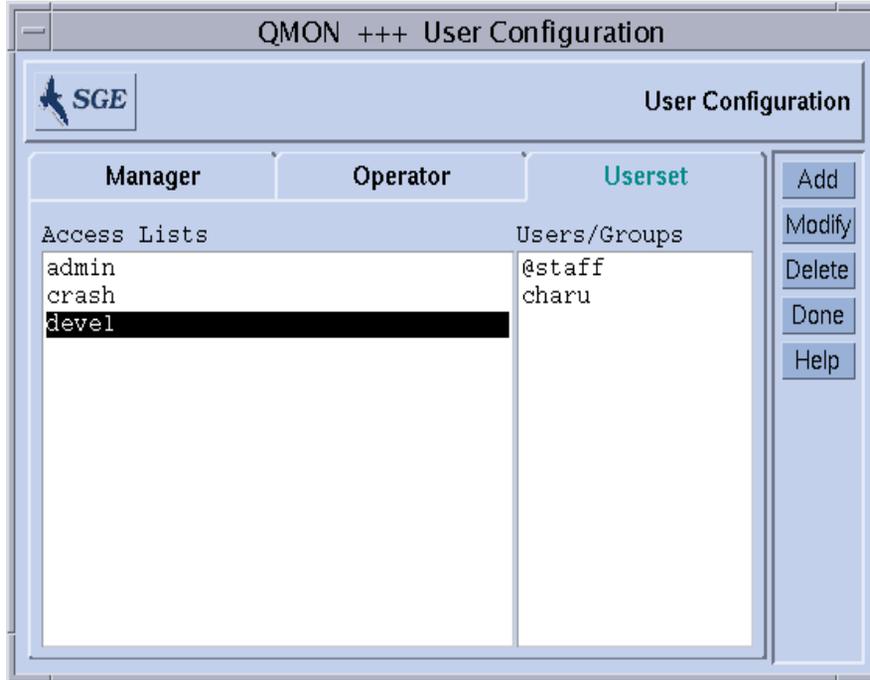


그림 9-3 Userset 구성 대화 상자

사용 가능한 액세스 목록이 화면의 왼쪽에 있는 Usersets 선택 목록에 표시됩니다. Users/Groups 표시 영역에 액세스 목록의 내용을 표시하려면 Access Lists 선택 목록에서 해당 목록을 누르십시오.

참고 - 그룹은 접두어로 붙은 @ 기호에 의해 사용자와 구별됩니다.

Userset Configuration 대화 상자를 사용하여 다음 업무를 수행할 수 있습니다.

- **삭제** - 목록의 이름을 누른 후 대화 상자의 오른쪽에 있는 Delete 단추를 눌러서 Userset 선택 목록에서 기존 액세스 목록을 삭제하십시오
- **추가** - 선택 목록 위에 있는 입력 창에 계정 이름을 입력한 후 Add 단추를 누르거나 키보드의 Return 키를 눌러서 새 액세스 목록을 추가하십시오
- **수정** - Modify 단추를 눌러서 선택한 액세스 목록을 수정하십시오.

모든 경우에서 그림 9-4에 표시된 것과 유사한 Access List Definition 대화 상자가 열리고 대응하는 방법을 제공합니다.

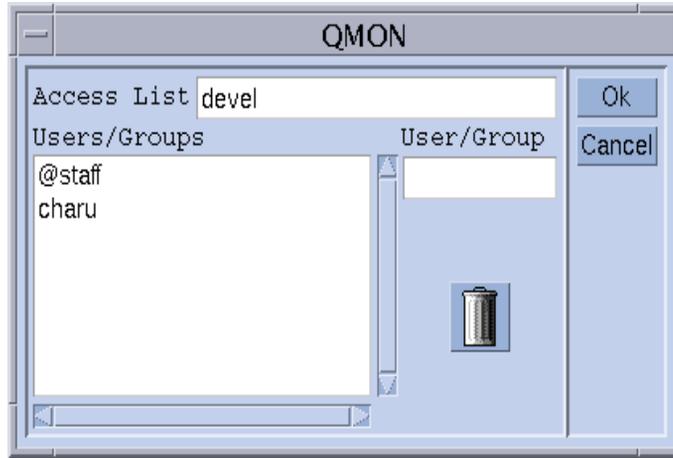


그림 9-4 Access List Definition 대화 상자

Access List Definition 대화 상자 창의 설명

- **Userset Name 입력 창** - Modify 조작의 경우에 선택한 액세스 목록의 이름을 표시하거나 이 창을 사용하여 선언될 액세스 목록의 이름을 입력할 수 있습니다.
- **Users/Groups 표시 영역** - 지금까지 정의된 액세스 목록 항목이 들어 있습니다.
- **User/Group 입력 창** - 액세스 목록에 새 항목을 추가하는 데 사용해야 합니다.

입력된 사용자 또는 그룹 이름(그룹은 @ 기호가 앞에 붙음)이 키보드의 Return 키를 누른 후에 Users/Groups 표시 영역에 첨부됩니다. 표시 영역에서 항목을 선택한 후 쓰레기통 아이콘 단추를 눌러서 항목을 삭제할 수 있습니다.

수정되거나 새로 정의된 액세스 목록은 Ok 단추를 누르자마자 등록되거나, Cancel 단추를 누르는 경우 삭제됩니다. 어느 경우에도 Access List Definition 대화 상자가 닫힙니다.

▼ 명령줄에서 사용자 액세스 목록을 구성하는 방법

- 적당한 옵션과 함께 다음 명령을 입력하십시오.

```
# qconf 스위치
```

사용 가능한 옵션

- `qconf -au 사용자이름[...] 액세스목록이름[...]`
사용자 추가—이 명령은 지정된 액세스 목록(들)에 하나 이상의 사용자를 추가합니다.
- `qconf -du 사용자이름[...] 액세스목록이름[...]`
사용자 삭제—이 명령은 지정된 액세스 목록(들)에서 하나 이상의 사용자를 삭제합니다.
- `qconf -su 액세스목록이름[...]`
사용자 액세스 목록 표시—이 명령은 지정된 액세스 목록을 표시합니다.
- `qconf -sul`
사용자 액세스 목록 표시—이 명령은 현재 정의된 모든 액세스 목록의 목록을 인쇄합니다.

스케줄링 정보

Sun Grid Engine 시스템의 작업 스케줄링 활동은 다음으로 구성됩니다.

- 처리 준비 결정—이들은 대기열이 가득 찼거나 과부하되었고 스폴링 작업이 현재 대기 영역에서의 실행에 적합하지 않기 때문에 실행 대기열 삭제 같은 활동입니다.
- 처리—이들 활동은 다른 모든 보류 중 및 실행 중인 작업과 관련하여 작업의 중요성 결정, 클러스터에 있는 모든 시스템의 부하 감지 및 구성된 선택 기준에 따라서 선택된 시스템의 실행 대기열에 작업 전송을 포함합니다.

Sun Grid Engine 소프트웨어는 다음을 바탕으로 컴퓨터의 이중 클러스터 사이에 작업을 스케줄합니다.

- 클러스터의 현재 부하
- 작업의 자원 요구사항(예: CPU, 메모리 및 I/O 대역폭)

스케줄링 결정은 사이트에 대한 전략과 클러스터의 각 컴퓨터의 순간적인 부하 특성을 바탕으로 합니다. 사이트의 스케줄링 전략은 Sun Grid Engine 시스템의 구성 매개변수를 통해 표현됩니다. 부하 특성은 시스템이 실행할 때 성능 데이터를 수집하여 확인됩니다.

스케줄링 전략

관리자는 다음 Sun Grid Engine 스케줄링 업무와 관련된 전략을 설정할 수 있습니다.

- **대기열 정렬**—소프트웨어는 대기열이 채워져야 하는 순서에 따라서 클러스터의 대기열을 정렬시킵니다.
- **작업 정렬**—이것은 Sun Grid Engine 시스템이 작업을 스케줄하려 시도하는 순서를 판별합니다.

대기열 정렬

다음 방법은 Sun Grid Engine이 대기열을 채우려 시도하는 순서를 판별하기 위해 제공됩니다.

- **부하 보고**—Sun Grid Engine 관리자는 호스트 및 그의 대기열의 부하 상태를 비교하는 데 사용되는 부하 매개변수를 선택할 수 있습니다. 사용 가능한 광범위한 표준 부하 매개변수 및 이 세트를 사이트 고유 부하 센서로 확장하기 위한 인터페이스가 197 페이지, “부하 매개변수” 절에 설명되어 있습니다.
- **부하 크기 조정**—다른 호스트의 부하 보고서가 비교 가능한 상황을 반영하기 위해 정규화될 수 있습니다(142 페이지, “QMON을 사용하여 실행 호스트를 구성하는 방법” 절 참조).
- **부하 조정**—Sun Grid Engine 소프트웨어는 작업이 호스트에 작업 지정될 때 마지막 보고된 부하를 자동으로 정정하도록 구성할 수 있습니다. 정정된 부하는 최근에 시작된 작업에 의해 유발되는 부하 상황의 예상 증가를 표시합니다. 이 부하의 인위적인 증가는 이들 작업의 부하 영향이 효과를 보일 때 자동으로 축소될 수 있습니다.
- **순서 번호**—정확한 순서에 따라서 대기열을 정렬할 수 있습니다.

작업 정렬

Sun Grid Engine이 작업 지정을 시작하기 전에 작업이 먼저 가장 높은 우선순위의 순서가 됩니다. 그러면 Sun Grid Engine이 우선순위 순서대로 작업에 대한 적당한 자원을 찾으려 시도합니다. 관리자의 어떤 영향도 없을 때 순서는 선입선출(FIFO)입니다. 관리자는 작업 순서에 대한 다음 제어 수단을 갖습니다.

- **사용자 정렬** — 이 스케줄링 대체가 사용되면, 다른 사용자의 작업들이 삽입됩니다. 즉, 모든 사용자가 제출한 첫 번째 작업이 동등하게 취급되며, 다음에는 두 번째 작업이, 그리고 그 다음 순으로 이루어집니다.
- **작업 우선순위** — 관리자는 작업에 정렬 순서를 직접 결정하는 우선순위 번호를 할당할 수 있습니다. 사용자는 그에게 작업에 할당된 우선순위를 더 낮출 수 있습니다.
- **사용자/그룹 작업의 최대 수** — 사용자 또는 UNIX 사용자 그룹이 Sun Grid Engine 시스템에서 동시에 실행할 수 있는 작업의 최대 수를 제한할 수 있습니다. 이것은 한계를 초과하지 않는 사용자의 작업에 우선권이 제공되기 때문에 보류 중 작업 목록 정렬 순서에 영향을 줍니다.

스케줄러 간격에서 발생하는 것

스케줄러는 간격을 두고 작업을 스케줄합니다. 스케줄링 조치 사이에 Sun Grid Engine은 작업 제출, 작업 완료, 작업 취소, 클러스터 구성의 갱신 또는 클러스터에 새 시스템의 등록 같은 중요한 이벤트에 관한 정보를 보존합니다. 스케줄링이 발생할 때 스케줄러는 다음을 수행합니다.

- 모든 중요한 이벤트를 고려합니다.
- 관리자 스펙에 따라서 작업 및 대기열을 정렬합니다.
- 모든 작업의 자원 요구사항을 고려합니다.

그런 다음 Sun Grid Engine 시스템은 필요할 때 다음을 수행합니다.

- 새 작업을 작업 지정합니다.
- 실행 중인 작업을 일시중단합니다.
- 상대 quo를 유지보수합니다.

스케줄러 모니터링

작업이 시작되지 않고 이유가 확실하지 않은 경우 작업에 대해 `-w v` 옵션을 갖는 `qalter`를 실행할 수 있습니다. Sun Grid Engine 소프트웨어는 빈 클러스터를 가정하고 작업에 적합한 임의의 대기열이 사용 가능한지 여부를 점검합니다.

`qstat -j 작업ID`를 실행하여 추가 정보를 얻을 수 있습니다. 이 명령은 작업이 마지막 스케줄링 실행에서 스케줄되지 않은 이유도 포함하는 작업의 요청 프로파일의 요약을 인쇄합니다. 작업 ID 없이 `qstat -j`를 실행하면 마지막 스케줄링 간격에서 스케줄되지 않은 모든 작업에 대한 이유가 요약됩니다.

참고 - 스케줄링 이유 정보의 컬렉션은 스케줄링 구성 `sched_conf`에서 커져야 합니다. 해당하는 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `schedd_job_info` 매개변수나 218 페이지, “QMON을 사용하여 스케줄러 구성을 변경하는 방법” 절을 참조하십시오.

Sun Grid Engine 스케줄러 `sge_schedd`의 결정에 관한 추가 세부사항을 검색하기 위해 `qconf` 명령의 `-tsm` 옵션을 사용할 수 있습니다. 이 명령은 `sge_schedd`가 추적 결과를 파일에 기록하도록 합니다.

기본 스케줄링

기본 Sun Grid Engine 스케줄링은 **선입선출** 정책입니다. 즉, 제출된 첫 번째 작업이 스케줄러가 대기열에 작업 지정하기 위해 시험하는 첫 번째 작업입니다. 보류 중 작업의 목록에 있는 첫 번째 작업이 적당하고 유효한 대기열을 찾는 경우 스케줄러 실행에서 첫 번째로 시작됩니다. 첫 번째 작업이 적합하고 사용 가능한 자원을 찾지 못하는 경우에만 두 번째 작업 또는 그 다음에 랭크되는 작업이 보류 중 작업 목록의 첫 번째 작업 이전에 시작될 수 있습니다.

작업에 대한 대기열 선택이 관련되는 한, 기본 Sun Grid Engine 전략은 작업의 자원 요구 사항에 대해 적합한 서비스를 전달하고 가장 부하가 적은 호스트의 대기열을 선택하는 것입니다. 복수의 적합한 대기열이 동일한 부하를 공유하는 경우 선택되는 대기열은 예측할 수 없습니다.

스케줄링 대안

작업 스케줄링 및 대기열 선택 전략을 수정하는 다양한 방법이 있습니다.

- 스케줄링 알고리즘 변경
- 시스템 부하 크기 조정
- 순서 번호에 의한 대기열 선택
- 지분에 의한 대기열 선택
- 사용자당 또는 그룹당 작업 수 제한

다음 절은 이들 대안을 자세히 알아봅니다.

스케줄링 알고리즘 변경

스케줄러 구성 매개변수 `algorithm`(자세한 정보는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `sched_conf` 항목 참조)이 사용 중인 스케줄링 알고리즘에 대한 선택을 제공하도록 고안됩니다. 현재는 `default`가 유일하게 허용되는 설정입니다.

작업 우선순위

Sun Grid Engine 관리자는 보류 중 작업 목록에 스폴된 작업에 작업 우선순위라고 하는 정수를 할당할 수 있습니다. 작업 우선순위는 보류 중 작업 목록에 작업의 위치를 정의합니다. 가장 높은 우선순위 번호를 가진 작업은 스케줄러가 첫 번째로 검사하게 됩니다. 작업 우선순위에 대한 값의 범위는 -1024 ~ 1023 사이가 되며 바로 제출된 새 작업의 우선순위는 0 입니다. 작업에 음수의 우선순위 값이 할당되는 경우, 작업은 바로 제출된 새 작업 뒤에 정렬되기도 합니다. 동일한 우선순위 번호를 가진 여러 작업이 존재하는 경우, 기본 선입선출 규칙이 이 우선순위 클래스에 적용됩니다.

작업 우선순위는 다음 명령을 통해 작업에 할당됩니다.

```
%qalter -p 우선순위 작업ID ...
```

위 명령에서, `prio`는 길게 이어지는 공백 구분 작업 ID 목록에 지정된 것처럼 작업의 목록에 할당될 우선순위를 지정합니다.

참고 - 작업 우선순위라는 용어는 특수 대기열에서 실행된 모든 작업에 대해 설정된 `nice` 값을 정의하는 우선순위 대기열 구성 매개변수(*Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `queue_conf` 매뉴얼 페이지 참조)와 혼동해서는 안됩니다. 또한 `qstat` 출력의 두 번째 열이 제출된 작업에 현재 할당된 우선순위를 보여준다는 것에 유의하십시오.

동일 공유 정렬

위에서 설명한 기본 선입선출 스케줄링 정책은 사용자가 짧은 시간에 잇따라 일련의 작업들을 제출하는 경우(예를 들어, 셸 스크립트 절차를 사용하여) 오히려 불공평한 결과를 초래하는 것으로 잘 알려져 있습니다. 이 사용자의 작업들이 나머지 사용자들에게 이들 대기열을 할당할 기회를 주지 않으면서 아주 오랜 시간 동안 적절한 자원들을 감춥니다.

이런 경우, 클러스터 관리자는 스케줄링 정책을 소위 **동일 공유 정렬**로 변경할 수 있습니다. 이 스케줄링 대체가 사용되고 사용자가 이미 시스템에서 실행 중인 작업이 있는 경우, 그의 다른 모든 작업들은 동일한 우선순위 클래스에 있는 다른 사용자들의 작업 뒤에 정렬됩니다(우선순위 클래스에 대한 자세한 설명은 이전 절을 참조).

동일 공유 정렬은 스케줄러 구성 매개변수 `user_sort`가 TRUE로 설정되는 경우에 작동됩니다(*Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `sched_conf` 매뉴얼 페이지 참조).

시스템 부하 크기 조정

Sun Grid Engine 시스템은 대기열을 호스트하는 시스템의 시스템 부하 정보를 사용하여 작업에 대한 실행 대기열을 선택합니다. 이 대기열 선택 체계는 부하 균형 상황을 구축하므로 클러스터에 있는 사용 가능한 자원의 더 좋은 이용을 보증합니다.

그러나 시스템 부하가 항상 진실을 말할 수는 없습니다. 예를 들어 다중 CPU 시스템이 단일 CPU 시스템에 비교되는 경우 다중프로세서 시스템이 더 많은 프로세스를 실행할 것이기 때문에 대개 더 높은 부하 수치를 보고하며 시스템 부하는 CPU 액세스를 얻으려 시도하는 프로세스의 수에 크게 영향을 받는 측정값입니다. 그러나 다중 CPU 시스템은 단일 CPU 시스템보다 훨씬 더 높은 부하를 만족시킬 수 있습니다. 이 문제는 `sge_execd`에 의해 기본적으로 보고되는 부하 값의 프로세서 수로 조정된 세트에 의해 다루어집니다(자세한 내용은 197 페이지, “부하 매개변수” 절 및 `<sge 루트>/doc/load_parameters.asc` 파일 참조). 위에서 설명한 문제점을 피하기 위해 원시 부하 값 대신 이들 부하 매개변수를 사용하십시오.

부하 값의 잠재적으로 부적절한 해석에 대한 다른 예는 모두가 동일한 부하 값이 임의의 호스트가 작업을 실행하도록 선택될 수 있음을 의미하지 않는 성능 잠재력 또는 가격 성능 비율의 큰 차이를 갖는 시스템입니다. 이런 종류의 상황에서는 Sun Grid Engine 관리자가 관련 실행 호스트 및 부하 매개변수에 대한 부하 크기 조정 요소를 정의해야 합니다(142 페이지, “QMON을 사용하여 실행 호스트를 구성하는 방법” 및 관련 절 참조).

참고 - 크기 조정된 부하 매개변수는 또한 부하 임계값 목록 *부하임계값* 및 *이주부하임계값*에 대해 비교하는 데 사용됩니다(자세한 내용은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `queue_conf` 항목 참조).

부하 매개변수와 연관된 추가 문제점은 값과 그의 상대적 중요성의 응용 프로그램 및 사이트 종속 해석의 필요성입니다. CPU 부하가 특정 사이트에서 공통적인 응용 프로그램의 특정 유형에 우세한 반면, 다른 사이트 및 사이트의 연산 클러스터가 일반적으로 전용되는 응용 프로그램 프로파일의 경우 메모리 부하가 훨씬 더 중요할 수 있습니다. 이 문제점을 다루기 위해 Sun Grid Engine은 관리자가 클러스터 구성 파일 `sched_conf`에 소위 *부하 공식*을 지정할 수 있게 합니다(자세한 내용은 대응하는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서* 절 참조). 부하 공식에 사이트 정의 부하 매개변수(198 페이지, “사이트 고유 부하 매개변수 추가” 절 참조) 및 소비 가능 자원(185 페이지, “소비 가능 자원” 절 참조)을 사용하여 자원 이용률 및 용량 계획에 대한 사이트 고유 정보를 고려할 수 있습니다.

마지막으로 부하 매개변수의 시간 종속성을 고려해야 합니다. 시스템에서 실행 중인 Sun Grid Engine 작업에 의해 부과되는 부하는 시간에 따라 달라지며 종종(예를 들어 CPU 부하의 경우) 운영 체제에 의해 적당한 수량으로 보고되기 위해 다소의 시간이 필요합니다. 결국 작업이 아주 최근에 시작된 경우 보고되는 부하는 작업에 의해 해당 호스트에 이미 부과되는 부하의 충분한 표지를 제공하지 않을 수 있습니다. 보고되는 부하는 시간에 따른 실제 부하를 채택하지만, 보고되는 부하가 너무 낮은 시간 기간이 이미 해당 호스트의 과다신청을 유발할 수 있습니다. Sun Grid Engine은 관리자가 Sun Grid Engine 스케줄러에서 이 문제를 보상하기 위해 사용되는 *부하 조정* 요소를 지정할

수 있게 합니다. 이들 부하 조정 요소를 설정하는 방법에 대한 자세한 정보는 스케줄러 구성 파일 sched_conf를 다루는 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*를 참조하십시오.

순서 번호에 의한 대기열 선택

기본 대기열 선택 체계를 변경하는 또 다른 방법은 글로벌 Sun Grid Engine 클러스터 구성 매개변수인 queue_sort_method를 기본값 load 대신 seq_no로 설정하는 것입니다(*Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 sched_conf 항목 참조). 이 경우에 시스템 부하는 더 이상 대기열을 선택하는 1차 방법으로 사용되지 않습니다. 대신 대기열 구성 매개변수 seq_no(*Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 queue_conf 항목 참조)에 의해 대기열에 지정되는 순서 번호가 대기열 사이에서 대기열이 선택되는(대기열이 고려되는 작업에 적합하고 사용 가능한 경우) 고정 순서를 정의하는데 사용됩니다.

이 대기열 선택 정책은 사용자 사이트에서 일괄처리 서비스를 제공하는 시스템이 단조로운 작업당 가격 순서로 순위가 매겨지는 경우에 유용할 수 있습니다. 예를 들어, 시스템 A에서 실행하는 작업이 1 화폐 단위의 비용이 드는 반면 시스템 B에서는 10 단위, 시스템 C에서는 100 단위의 비용이 듭니다. 따라서 선호되는 스케줄링 정책은 먼저 호스트 A를 채운 후 호스트 B를 채우고 대안이 없는 경우에만 호스트 C를 사용하는 것입니다.

참고 - 대기열 선택 방법을 seq_no로 바꾸었고 고려된 모든 대기열이 동일한 연속 번호를 공유하면 대기열은 기본값 load에 의해 선택됩니다.

사용자 또는 그룹당 작업 수 제한

Sun Grid Engine 관리자는 임의의 시점에서 모든 사용자 또는 모든 UNIX 그룹에 의해 실행되도록 허용되는 작업 수에 대한 상한을 지정할 수 있습니다. 이 기능을 집행하려면 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 sched_conf 절에서 설명하는 것처럼 maxujobs 및/또는 maxgjobs를 설정하십시오.

▼ QMON을 사용하여 스케줄러 구성을 변경하는 방법

1. QMON Main 메뉴에서 Scheduler Configuration을 누릅니다.

Scheduler Configuration 대화 상자가 표시됩니다. 대화 상자는 General Parameters 섹션과 Load Adjustment 섹션으로 분리됩니다. 수행하려는 작업에 따라서 어느 하나를 선택합니다.

- a. 일반 스케줄링 매개변수를 변경하려면 **General Parameters** 탭을 누릅니다.
General Parameters 대화 상자는 그림 9-5의 보기와 유사합니다.

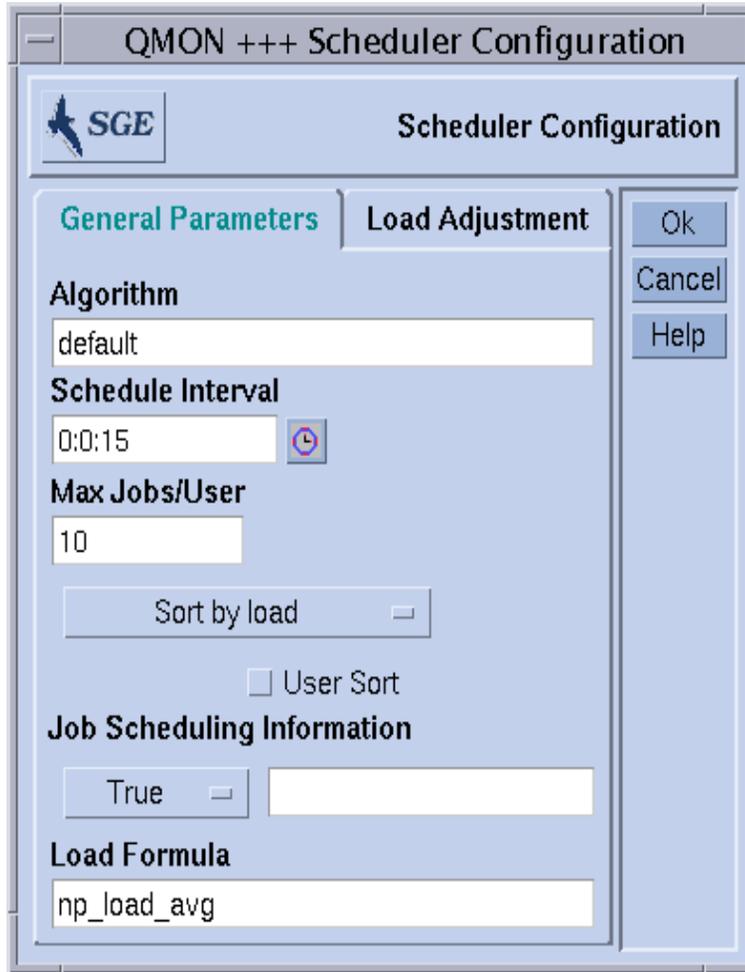


그림 9-5 Scheduler Configuration 대화 상자—General Parameters

General Parameters 대화 상자에서 다음 매개변수를 설정할 수 있습니다.

- 스케줄링 알고리즘(215 페이지, “스케줄링 알고리즘 변경” 참조)
- 스케줄러 실행 사이의 정규 시간 간격
- 사용자당 및 UNIX 그룹당 동시에 실행하도록 허용되는 작업의 최대 수(218 페이지, “사용자 또는 그룹당 작업 수 제한” 참조).
- 대기열 정렬 체계—부하에 의한 정렬 또는 순서 번호에 의한 정렬(218 페이지, “순서 번호에 의한 대기열 선택” 참조).

- 동일 공유 정렬(User Sort 플래그)의 사용 여부(216 페이지, “동일 공유 정렬” 절 참조).
 - 작업 스케줄링 정보가 `qstat -j` 를 통해 액세스 가능한지 여부 또는 이 정보가 첨부된 입력 필드에 지정되는 작업 ID의 범위에 대해서만 수집되어야 하는지 여부. 보류 중인 작업의 극히 높은 수가 있는 경우에만 일시적으로 작업 스케줄링 정보의 일반 콜렉션을 켜도록 권장합니다.
 - 호스트 및 대기열을 정렬하는 데 사용될 부하 공식
- b. 부하 조정 매개변수를 변경하려면 **Load Adjustment** 탭을 선택합니다.

Load Adjustment 매개변수 대화 상자는 그림 9-6의 보기와 유사합니다.

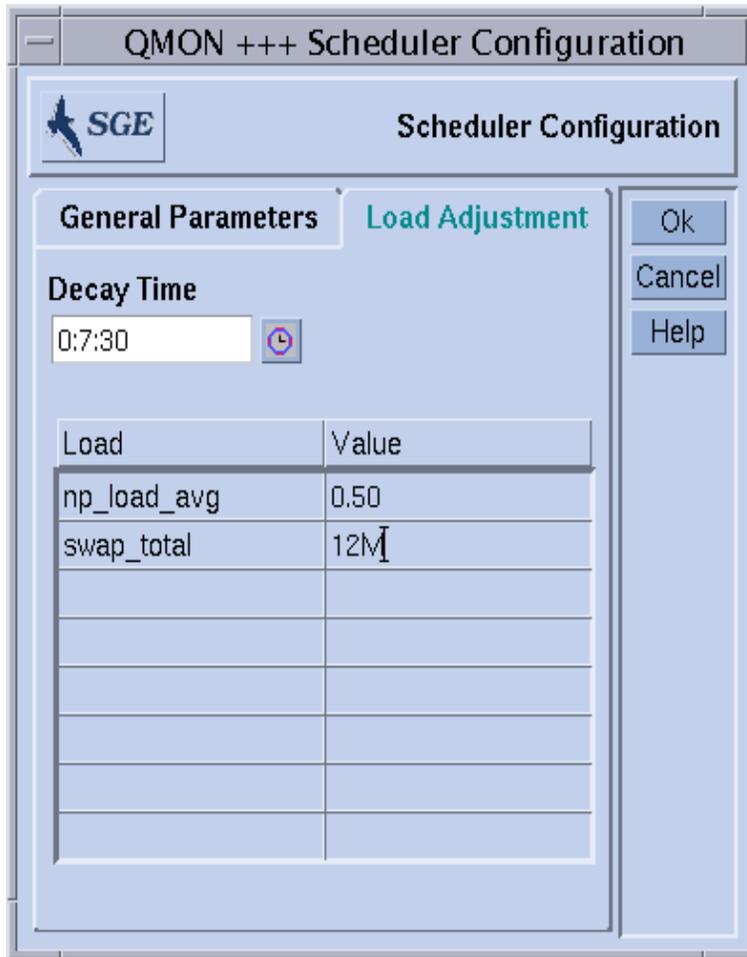


그림 9-6 Scheduler Configuration 대화 상자—Load Adjustment

Load Adjustment 대화 상자를 사용하면 다음 매개변수를 정의할 수 있습니다.

- 부하 조정 소멸 시간
- 조정 값이 현재 정의되는 모든 부하 및 소비 가능 속성을 나열하는 대화 상자의 하단 부의 부하 조정 값의 표. 맨 위에 있는 Load 또는 Value 단추를 눌러서 목록을 항상 시킬 수 있습니다. 그러면 호스트에 추가된 모든 속성(즉, 글로벌, 호스트 및 관리자 정의 콤플렉스에 구성된 모든 속성의 집합)을 갖는 선택 목록이 열립니다. Attribute Selection 대화 상자는 그림 6-6에 표시되어 있습니다. 속성 중 하나를 선택하고 OK 단추로 선택을 확인하면 해당 속성이 Consumable/Fixed Attributes 표의 Load 열에 추가되고 대응하는 Value 필드에 대한 포인터가 추가됩니다. Value 필드를 두 번 눌러서 기존 값을 수정할 수 있습니다. 해당하는 표 행을 선택한 후 CTRL-D를 입력하거나 오른쪽 마우스 단추를 눌러서 삭제 상자를 열고 삭제를 확인하여 속성을 삭제하십시오.

백그라운드 정보에 대해서는 217 페이지, “시스템 부하 크기 조정”을 참조하십시오. 스케줄러 구성에 대한 자세한 내용은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 sched_conf 매뉴얼 페이지를 참조하십시오.

경로 별명 지정

Solaris 및 기타 네트워크화된 UNIX 환경에서 사용자는 NFS를 통해 액세스 가능하게 된 경우 매우 자주 다른 시스템에 동일한 홈 디렉토리(또는 그의 경로)를 갖습니다. 그러나 때로는 홈 디렉토리 경로가 모든 시스템에서 정확하게 동일하지 않습니다.

예를 들어 NFS 및 자동 마운터를 통해 사용 가능한 사용자 홈 디렉토리를 고려하십시오. 사용자가 NFS 서버에 홈 디렉토리 /home/foo를 갖는 경우 자동 마운터를 실행하는 적절하게 설치된 모든 NFS 클라이언트에서 이 경로 아래에서 홈 디렉토리에 액세스할 수 있습니다. 그러나 클라이언트의 /home/foo가 바로 자동 마운터가 실제로 디렉토리를 마운트하는 NFS 서버의 실제 위치인 /tmp_mnt/home/foo에 대한 기호 링크임을 유의하는 것이 중요합니다.

그런 상황에서 사용자가 홈 디렉토리 트리 안에 어딘가에서 qsub -cwd 플래그(현재 작업 디렉토리에서 작업 실행)와 함께 클라이언트에 작업을 제출하는 경우 Sun Grid Engine 시스템이 실행 호스트(해당 호스트가 NFS 서버인 경우)의 현재 작업 디렉토리를 찾으려 시도하는 문제점을 갖고 표시됩니다. 이것은 qsub 명령이 제출 호스트의 현재 작업 디렉토리에 도달하고 /tmp_mnt/home/foo/를 얻기 때문입니다—이것이 제출 호스트의 실제 위치이기 때문입니다. 이 경로가 실행 호스트에 전달되며 실행 호스트가 /home/foo의 실제 홈 디렉토리 경로를 갖는 NFS 서버인 경우 해석될 수 없습니다.

대개 비슷한 문제점을 유발하는 다른 경우는 다른 시스템에 다른 마운트 지점 경로를 갖는 고정된(자동화되지 않은) NFS 마운트(예를 들어, 한 호스트에서는 /usr/people 및 다른 호스트에서는 /usr/users 아래에 홈 디렉토리 마운팅) 또는 외부에서 네트워크 가능 파일 시스템으로의 기호 링크입니다.

그런 문제점을 막기 위해 Sun Grid Engine 소프트웨어는 관리자와 사용자 모두가 *경로 별명 지정 파일*을 구성할 수 있게 합니다. 그런 두 파일의 위치는 다음과 같습니다.

- `<sges루트>/<셀>/common/sg_aliases`—이것은 클러스터 글로벌 경로 별명 지정 파일입니다.
- `$HOME/.sg_aliases`—이것은 사용자 고유의 경로 별명 지정 파일입니다.

참고 - 규정된 관리자만이 클러스터 글로벌 파일을 수정할 수 있습니다.

파일 형식

두 파일이 모두 동일한 형식을 공유합니다.

- 공백 행과 첫 번째 열에 # 기호를 갖는 행은 건너뜁니다.
- 각 행(공백 행 또는 #이 앞에 오는 행 이외)이 임의 숫자의 공백이나 탭에 의해 구분되는 4개의 문자열을 포함해야 합니다.
첫 번째 문자열은 소스 경로, 두 번째는 제출 호스트, 세 번째는 실행 호스트 및 네 번째는 소스 경로 대체를 지정합니다.
- 제출 및 실행 호스트 항목 둘다 어떤 호스트와도 일치하는 * 기호으로 구성될 수 있습니다.

경로 별명 지정 파일이 해석되는 방법

파일은 다음과 같이 해석됩니다.

- `qsub`가 실제 현재 작업 디렉토리 경로를 검색한 후 클러스터 글로벌 경로 별명 지정 파일이 읽어집니다(있는 경우). 사용자 경로 별명 지정 파일이 글로벌 파일에 첨부된 것처럼 그 후에 읽어집니다.
- 건너뛰지 않는 행은 파일의 맨 위부터 한 행씩 읽어드는 동안 해당 행에 의해 지정되는 변환이 필요한 경우 저장됩니다.
- 변환은 제출 호스트 항목이 `qsub` 명령이 실행되는 호스트와 일치하는 경우 및 소스 경로가 현재 작업 디렉토리 또는 이미 저장된 소스 경로 대체의 시작 부분을 형성하는 경우에만 저장됩니다.
- 양 파일이 읽어드리자마자, 저장된 경로 별명 지정 정보가 제출된 작업과 함께 전달됩니다.
- 실행 호스트에서 별명 지정 정보가 평가됩니다. 경로 별명의 실행 호스트 항목이 실행 호스트와 일치하는 경우 현재 작업 디렉토리의 선두 부분이 소스 경로 대체에 의해 대체됩니다. 이 경우에 현재 작업 디렉토리 문자열이 변경되고 후속 경로 별명이 적용될 대체된 작업 디렉토리 경로와 일치해야 함을 주의하십시오.

경로 별명 지정 파일의 예

은 위에서 설명한 NFS/자동 마운터 문제점이 별명 파일 항목으로 해결될 수 있는 방법의 예입니다.

```
# cluster global path aliases file
# src-path      subm-host  exec-host      dest-path
/tmp_mnt/      *              *              /
```

경로 별명 지정 파일의 예

기본 요청 구성

일괄처리 작업은 일반적으로 특정 작업에 대해 사용자에게 의해 정의되는 요청 프로파일에 대해서 Sun Grid Engine 시스템에 의해 대기열에 할당됩니다. 사용자는 작업을 성공적으로 실행하기 위해 만족되어야 하는 요청 세트를 어셈블하고 Sun Grid Engine 스케줄러는 이 작업에 대한 요청 세트를 만족하는 대기열만을 고려합니다.

사용자가 작업에 대해 어떤 요청도 지정하지 않는 경우 스케줄러는 사용자가 추가 제한 없이 액세스하는 모든 대기열을 고려합니다. 그러나 Sun Grid Engine 소프트웨어는 사용자가 명시적으로 지정하지 않은 경우에도 작업에 대한 자원 요구사항을 정의할 수 있는 기본 요청의 구성을 허용합니다.

기본 요청은 모든 사용자에게 대해 개인적으로 뿐만 아니라 Sun Grid Engine 클러스터의 모든 사용자에게 대해 글로벌로 구성될 수 있습니다. 기본 요청 구성은 기본 요청 파일로 표현됩니다. 글로벌 요청 파일은 <sgc 루트>/<셀>/common/sgc_request 아래에 위치하는 반면, .sgc_request라고 부르는 사용자 고유 요청 파일은 사용자의 홈 디렉토리나 qsub 명령이 실행되는 현재 작업 디렉토리에 위치할 수 있습니다.

이들 파일이 존재하는 경우 모든 작업에 대해 평가됩니다. 평가의 순서는 다음과 같습니다.

1. 글로벌 기본 요청 파일
2. 사용자의 홈 디렉토리에 있는 사용자 기본 요청 파일
3. 현재 작업 디렉토리에 있는 사용자 기본 요청 파일

참고 - 작업 스크립트에 지정되거나 qsub 명령줄과 함께 제공되는 요청은 기본 요청 파일에 있는 요청으로 더 높은 우선권을 갖습니다(작업에 대해 명시적으로 자원을 요청하는 방법에 대한 자세한 내용은 4장 참조).

참고 - 기본 요청 파일의 의도하지 않은 영향은 모든 이전 요구사항 스펙을 삭제하는 `qsub -clear` 옵션의 사용에 의해 금지될 수 있습니다.

기본 요청 파일의 형식

로컬 및 글로벌 기본 요청 파일 모두의 형식이 다음 목록에 설명되어 있습니다.

- 기본 요청 파일은 임의 숫자의 행을 포함할 수 있습니다. 공백 행과 첫 번째 열에 # 기호를 갖는 행은 건너뛴니다.
- 건너뛰지 않는 각 행은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*에 설명되는 것처럼 모든 `qsub` 옵션을 포함할 수 있습니다. 행당 둘 이상의 옵션이 허용됩니다. 일괄처리 스크립트 파일과 일괄처리 스크립트에 대한 인수 옵션은 `qsub` 옵션으로 간주되지 않고 따라서 기본 요청 파일에 허용되지 않습니다.
- `qsub -clear` 옵션은 현재 평가되는 요청 파일 또는 이전에 처리된 요청 파일에 있는 모든 이전 요구사항 스펙을 삭제합니다.

기본 요청 파일의 예

예를 들어 사용자의 로컬 기본 요청 파일이 코드 예 9-1에 있는 스크립트인 `test.sh`와 동일하게 구성된다고 가정하십시오.

```
# 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
```

코드 예 9-1 기본 요청 파일의 예

스크립트를 실행하려면 사용자가 다음 명령을 입력합니다.

```
% qsub test.sh
```

`test.sh` 스크립트 실행의 효과는 사용자가 다음과 같이 명령줄에서 직접 모든 `qsub` 옵션을 지정한 경우와 동일합니다.

```
% qsub -l arch=solaris64,s_cpu=5:0:0 -cwd test.sh
```

참고 - qsub를 통해 제출되는 일괄처리 작업과 마찬가지로, qsh를 통해 제출되는 대화식 작업도 기본 요청 파일을 고려합니다. QMON을 통해 제출되는 대화식 또는 일괄처리 작업도 이들 요청 파일을 고려합니다.

회계 및 이용률 통계 수집

Sun Grid Engine 명령인 qacct를 사용하여 영숫자 회계 통계를 생성할 수 있습니다. 스위치 없이 호출되는 경우에는 qacct는 종료했고 클러스터 구성 파일, <sgc 루트>/common/accounting에 포함되는 모든 작업에 의해 생성되는 Sun Grid Engine 클러스터의 모든 시스템에 대한 누적 이용률을 표시합니다. 이 경우 qacct는 단지 초 단위로 다음 세 시간을 보고합니다.

- REAL—이것은 벽시계 시간, 즉 작업의 시작 및 종료 사이의 시간입니다.
- USER—이것은 사용자 프로세스에서 소비된 CPU 시간입니다.
- SYSTEM—이것은 시스템 호출에서 소비된 CPU 시간입니다.

특정 또는 모든 대기열, 특정 또는 모든 사용자 등에 관한 회계 정보를 보고하기 위해 여러 스위치를 사용할 수 있습니다. 특히 완료했고 작업을 제출하기 위해 qsub 명령과 함께 사용된 동일한 -1 구문으로 표현되는 자원 요구사항 스펙과 일치하는 모든 작업에 관한 정보를 요청하는 것이 가능합니다. 자세한 정보는 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 qacct 항목을 참조하십시오.

qacct 옵션(-j [작업ID|작업이름])은 getrusage 시스템 호출에 의해 제공되는 정보를 포함하여 Sun Grid Engine 시스템에 의해 저장되는 전체 자원 사용 정보에 대한 직접 액세스를 제공합니다(*Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 해당하는 항목 참조).

이 옵션은 작업 ID [작업ID]를 갖는 작업 또는 작업 이름 [작업이름]을 갖는 작업에 대한 자원 사용 항목을 각각 보고합니다. 인수가 주어지지 않는 경우 참조된 회계 파일에 들어 있는 모든 작업이 표시됩니다. 작업 ID가 선택되는 경우 및 둘 이상의 항목이 표시되는 경우 작업 ID 번호가 감싸졌거나(작업 ID의 범위는 1 - 999999) 이주된 검사점 작업이 표시됩니다.

검사점 지원 정보

검사점은 실행 중인 작업 또는 응용 프로그램의 상태를 동결시키고, 이 상태(소위 검사점)를 디스크에 저장하고 작업이나 응용 프로그램이 완료하는 데 실패한 경우(예를 들어, 시스템 종료로 인해) 나중에 해당 검사점에서 재시작하는 기능입니다. 검사점이 한

호스트에서 다른 호스트로 이동될 수 있는 경우 검사점을 사용하여 연산 자원의 상당한 유실 없이 클러스터에서 응용 프로그램 또는 작업을 이주할 수 있습니다. 따라서 검사점 기능의 도움으로 동적 부하 균형을 제공할 수 있습니다.

Sun Grid Engine 시스템은 두 레벨의 검사점을 지원합니다.

■ 사용자 레벨 검사점

이 레벨에서는 검사점 생성 메커니즘을 제공하는 것은 전적으로 사용자 또는 응용 프로그램의 책임입니다. 사용자 레벨 검사점의 예에는 다음이 포함됩니다.

- 응용 프로그램의 재시작 시에 이들 파일의 적합한 처리와 결합되는, 중요한 알고리즘 단계에서 응용 프로그램에 코드화되는 재시작 파일의 주기적 기록
- 응용 프로그램과 링크되어야 하고 그것에 의하여 검사점 메커니즘을 설치하는 검사점 라이브러리의 사용.

참고 - 다양한 타사 응용 프로그램이 재시작 파일의 기록을 바탕으로 통합 검사점 기능을 제공합니다. 검사점 라이브러리는 공용 도메인(예를 들어 위스콘신 대학교의 *Condor* 프로젝트를 참조하십시오) 또는 하드웨어 공급자에서 구할 수 있습니다.

■ 커널 레벨 투명 검사점

이 검사점 레벨은 잠재적으로 임의의 작업에 적용될 수 있는 운영 체제(및 그의 향상)에 의해 제공되어야 합니다. 커널 레벨 검사점을 사용하기 위해 응용 프로그램의 소스 코드 변경이나 재링크를 제공할 필요는 없습니다.

커널 레벨 검사점은 전체 작업(즉, 작업에 의해 작성되는 프로세스 계층)에 적용될 수 있는 반면, 사용자 레벨 검사점은 대개 단일 프로그램으로 제한됩니다. 따라서 그런 프로그램이 내장되는 작업은 전체 작업이 재시작되는 경우를 적절하게 처리해야 합니다.

검사점의 시점에서 작업이나 응용 프로그램에 의해 사용 중인 전체 가상 주소 공간이 디스크에 덤프되어야 하기 때문에 검사점 라이브러리를 바탕으로 하는 검사점뿐 아니라 커널 레벨 검사점은 매우 자원 소비적일 수 있습니다. 이와는 반대로, 재시작 파일을 기반으로 하는 사용자 레벨 검사점은 검사점에 기록되는 데이터를 중요한 정보만으로 제한할 수 있습니다.

검사점 환경

검사점 방법의 여러 유형과 여러 운영 체제 아키텍처에서 이들 방법의 잠재적인 다양한 변종을 반영하기 위해 Sun Grid Engine은 사용 중인 각 검사점에 대한 구성 가능한 속성 설명을 제공합니다.

이 속성 설명을 *검사점 환경*이라고 부릅니다. 기본 검사점 환경은 Sun Grid Engine 분배와 함께 제공되며 사이트의 수요에 따라서 수정될 수 있습니다.

새 검사점 방법이 원칙적으로 통합될 수 있지만, 이것은 간단하지 않은 작업이 될 수 있으며 경험이 있는 서비스 요원이나 Sun Grid Engine 지원팀에 의해서만 수행되어야 합니다.

▼ QMON을 사용하여 검사점 환경을 구성하는 방법

1. QMON 주 메뉴에서 **Checkpointing Configuration** 아이콘을 누릅니다.

그림 9-7에 표시되는 보기와 유사한 Checkpointing Configuration 대화 상자가 표시됩니다.

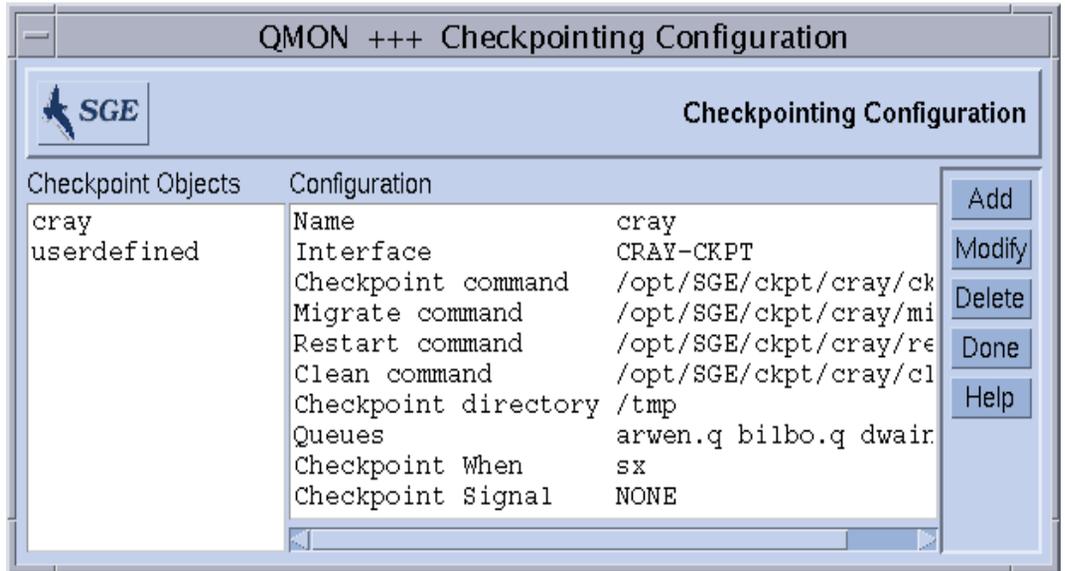


그림 9-7 Checkpointing Configuration 대화 상자

2. **Checkpointing Configuration** 대화 상자에서, 수행하려는 것에 따라서 다음 중 하나를 수행합니다.

구성된 검사점 환경 보기

- 이전에 구성된 검사점 환경을 보려면 **Checkpoint Objects** 열에 나열되는 검사점 환경 이름 중 하나를 선택하십시오.

대응하는 구성이 **Configuration** 열에 표시됩니다.

구성된 검사점 환경 삭제

- 구성된 검사점 환경을 삭제하려면 **Checkpoint Objects** 열에서 그의 이름을 강조표시하고 **Delete**를 누르십시오.

구성된 검사점 환경 수정

1. **Checkpoint Objects** 열에서 수정하려는 구성된 검사점 환경의 이름을 강조표시한 후 **Modify**를 누릅니다.

그림 9-8의 보기와 유사한 Change Checkpoint Object 대화 상자가 선택된 검사점 환경의 현재 구성과 함께 표시됩니다.

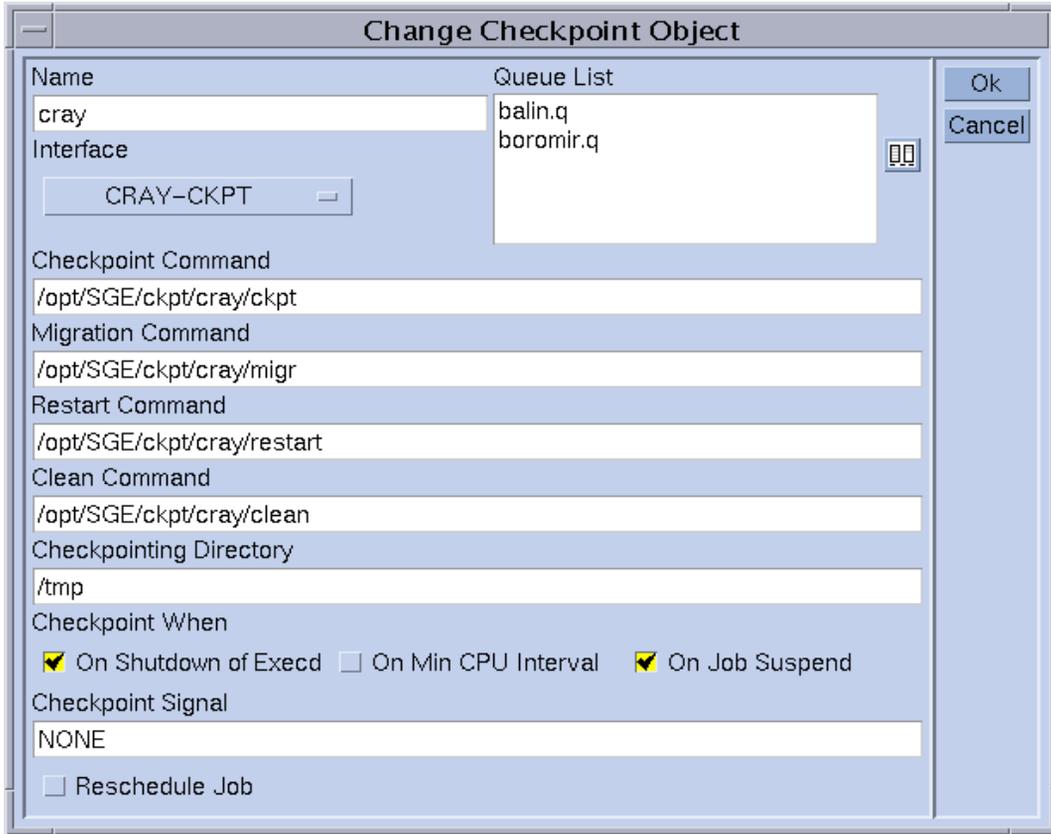


그림 9-8 Change Checkpoint Object 대화 상자

2. 다음 지침에 따라서 선택된 검사점 환경을 수정합니다.

Change Checkpoint Object 대화 상자에서 다음을 변경할 수 있습니다.

- 이름
- Checkpoint, migrate, restart, clean 명령 문자열
- 검사점 파일이 저장되는 디렉토리
- 검사점이 시작되어야 하는 시기
- 검사점이 시작될 때 작업이나 응용 프로그램에 전송될 신호

참고 - 이들 매개변수에 대한 자세한 내용은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 checkpoint 항목을 참조하십시오. 또한 사용될 인터페이스(검사점 방법라고도 부름)를 정의해야 합니다. 대응하는 선택 목록에 제공되는 인터페이스 중 하나를 선택하고 여러 인터페이스의 의미에 대한 자세한 내용은 checkpoint 항목을 참조하십시오.

3. 중요 - Sun Grid Engine 분배와 함께 제공되는 검사점 환경의 경우 Name, Checkpointing Directory 및 Queue List 매개변수만을 변경하십시오.

Queue List 매개변수를 변경하려면 “a 단계”로 가십시오. 그렇지 않으면 “a 단계”를 건너 뛰어 4 단계로 가십시오.

a. Queue List 창의 오른쪽에 있는 아이콘을 누르십시오(그림 9-8 참조).

그림 9-9의 보기와 유사한 Select Queues 대화 상자가 표시됩니다.

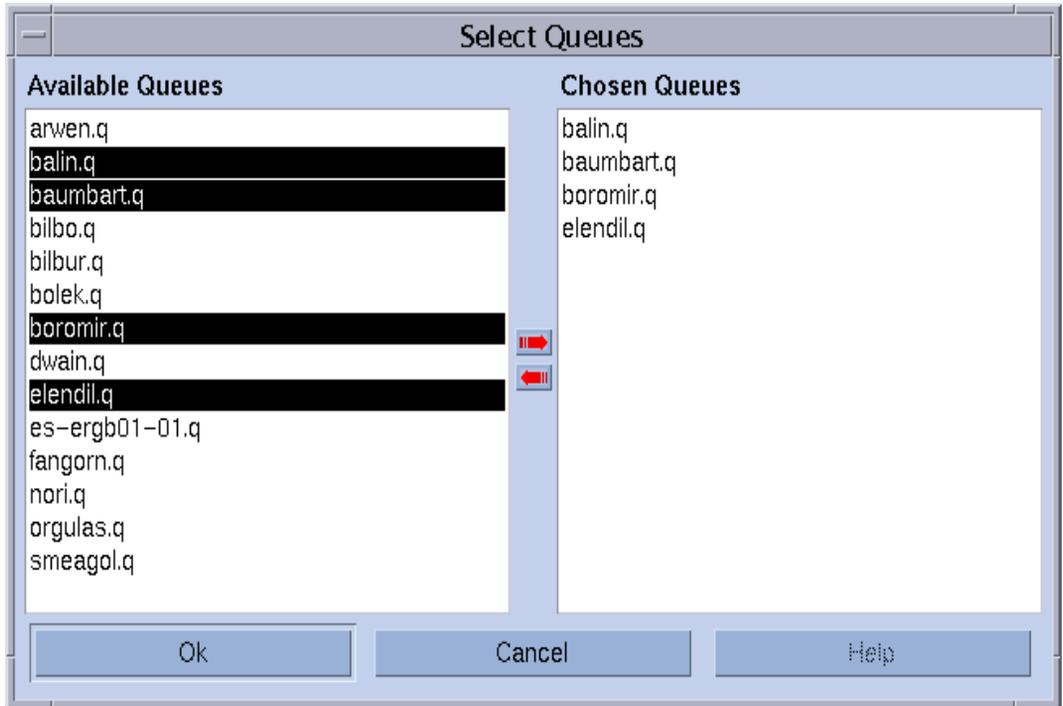


그림 9-9 검사점 대기열 선택 대화 상자

b. Available Queues 목록에서 검사점 환경에 포함시키고자 하는 대기열을 선택하고 Chosen Queues 목록에 추가하십시오.

c. OK를 누르십시오.

OK를 누르면 이들 대기열이 Change Checkpoint Object 대화 상자의 Queue List 창에 들어갑니다.

4. OK를 눌러서 변경사항을 sge_qmaster에 등록하거나 Cancel을 눌러 변경사항을 삭제합니다.

검사점 환경 추가

1. Checkpointing Configuration 대화 상자에서 Add를 누릅니다.

그림 9-8에 표시된 보기와 유사한 Change Checkpoint Object 대화 상자가 사용자가 편집할 수 있는 템플릿 구성과 함께 표시됩니다.

2. 요청된 정보로 템플릿을 채웁니다.

3. OK를 눌러서 변경사항을 sge_qmaster에 등록하거나 Cancel을 눌러 변경사항을 삭제합니다.

▼ 명령줄에서 검사점 환경을 구성하는 방법

- 다음 절에서 설명하는 적당한 옵션을 갖는 qconf 명령을 입력하십시오.

qconf 검사점 옵션

■ qconf -ackpt *검사점이름*

검사점환경 추가—이 명령은 체크포인팅 환경 구성 템플릿을 갖는 편집기(기본 vi 또는 \$EDITOR 환경 변수에 따라서)를 표시합니다. 매개변수 *검사점이름*은 검사점 환경의 이름을 지정하며 이미 템플릿의 대응하는 필드에 채워집니다. 템플릿을 변경하고 디스크에 저장하여 검사점 환경을 구성하십시오. 변경될 템플릿 항목의 자세한 설명은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 checkpoint 항목을 참조하십시오.

■ qconf -dckpt *검사점이름*

검사점 환경 삭제—이 명령은 지정된 검사점 환경을 삭제합니다.

■ qconf -mckpt *검사점이름*

검사점 환경 수정—이 명령은 구성 템플릿으로서 지정된 체크포인팅 환경을 갖는 편집기(기본 vi 또는 \$EDITOR 환경 변수에 따라서)를 표시합니다. 템플릿을 변경하고 디스크에 저장하여 검사점 환경을 수정하십시오. 변경될 템플릿 항목의 자세한 설명은 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 checkpoint 항목을 참조하십시오.

■ `qconf -sckpt` 검사점이름

검사점 환경 표시—이 명령은 지정된 검사점 환경의 구성을 표준 출력에 인쇄합니다.

■ `qconf -sckptl`

검사점 환경 목록 표시—이 명령은 현재 구성된 모든 검사점 환경의 이름의 목록을 표시합니다.

병렬 환경 관리

이 장에는 병렬 환경의 관리 및 운영과 관련된 정보가 들어 있습니다.

이들 주제에 대한 백그라운드 정보 외에, 이 장에는 다음 업무의 수행을 위한 자세한 지침이 들어 있습니다.

- 234 페이지, “QMON을 사용하여 PE를 구성하는 방법”
 - 234 페이지, “PE의 내용 표시”
 - 234 페이지, “PE 삭제”
 - 235 페이지, “PE 수정”
 - 235 페이지, “PE 추가”
- 238 페이지, “명령줄에서 PE를 구성하는 방법”
- 239 페이지, “명령줄에서 구성된 PE 인터페이스를 표시하는 방법”
- 239 페이지, “QMON으로 구성된 PE 인터페이스를 표시하는 방법”

병렬 환경 정보

*병렬 환경(PE)*은 네트워크된 환경이나 병렬 플랫폼에서 동시 연산을 위해 설계된 소프트웨어 패키지입니다. 다양한 시스템이 지난 몇 년 동안 다양한 하드웨어 플랫폼에서의 분산 및 병렬 처리를 위한 실행 가능한 기술로 발전했습니다. 오늘날의 가장 일반적인 메시지 전달 환경의 둘에 대한 예가 PVM(Parallel Virtual Machine, 미국 오크리지 국립연구소)과 MPI(Message Passing Interface, Message Passing Interface 포럼)입니다. 하드웨어 공급자가 제공하는 구현뿐 아니라 공용 도메인이 두 도구 모두에 대해 존재합니다.

이러한 모든 시스템은 서로 다른 특성을 보이며 차별화된 요구사항을 갖습니다. 그런 시스템 위에서 실행하는 임의의 병렬 작업을 처리할 수 있기 위해서 Sun Grid Engine 시스템은 다양한 요구를 만족하는 유연하고 강력한 인터페이스를 제공합니다.

241 페이지, “PE 시작 절차” 절과 242 페이지, “PE의 종료” 절에서 각각 설명하는 것처럼 적합한 시작 및 중지 절차가 제공되는 한 임의 PE를 Sun Grid Engine이 인터페이스할 수 있습니다.

▼ QMON을 사용하여 PE를 구성하는 방법

1. QMON Main 메뉴에서 PE Configuration 단추를 누릅니다.

그림 10-1의 보기와 유사한 Parallel Environment Configuration 대화 상자가 표시됩니다.

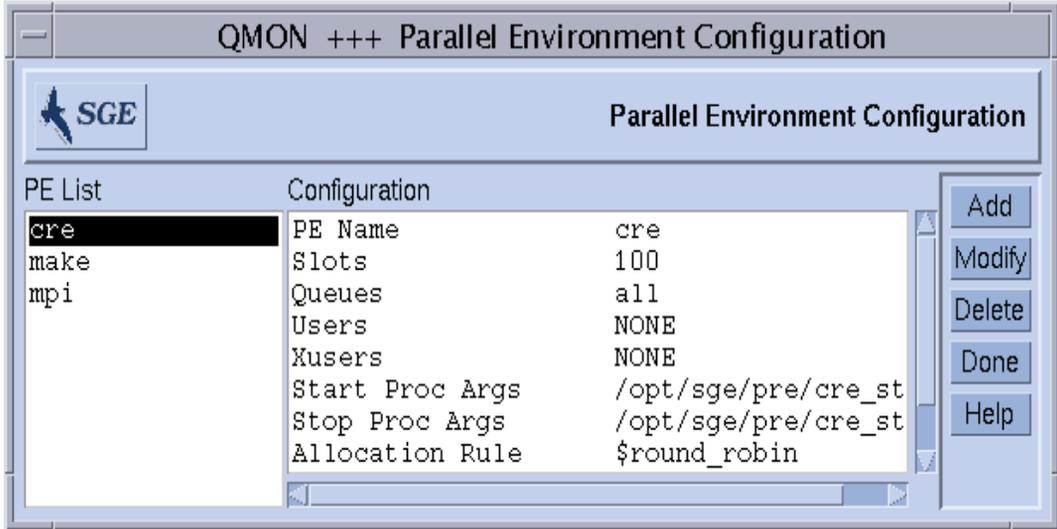


그림 10-1 Parallel Environment Configuration 대화 상자

이미 구성된 PE들이 화면의 왼쪽에 있는 PE List 선택 목록에 표시됩니다.

2. Parallel Environment Configuration 대화 상자에서, 수행하려는 것에 따라서 다음 중 하나를 수행합니다.

▼ PE의 내용 표시

- PE의 내용을 표시하려면 PE List 선택 목록에서 그의 이름을 누르십시오.
PE 구성의 내용이 Configuration 표시 영역에 표시됩니다.

▼ PE 삭제

- 선택한 PE를 삭제하려면 PE List 선택 목록에서 이름을 강조표시한 후 Delete(창의 오른쪽에 있는)를 누르십시오.

▼ PE 수정

1. 선택한 PE를 수정하려면 **Modify** 단추를 누르십시오.

그림 10-2에 표시된 보기와 유사한 PE Definition 대화 상자가 표시됩니다.

2. 236 페이지, “Parallel Environment 정의 매개변수의 설명” 절의 지침에 따라서 PE 정의를 수정합니다.

3. 변경사항을 저장하려면 **OK**, 변경사항을 삭제하려면 **Cancel**을 누릅니다.

OK 또는 Cancel을 누르면 대화 상자가 사라집니다.

▼ PE 추가

1. 새 PE를 추가하려면 **Add** 단추를 누릅니다.

그림 10-2에 표시된 보기와 유사한 PE Definition 대화 상자가 표시됩니다.

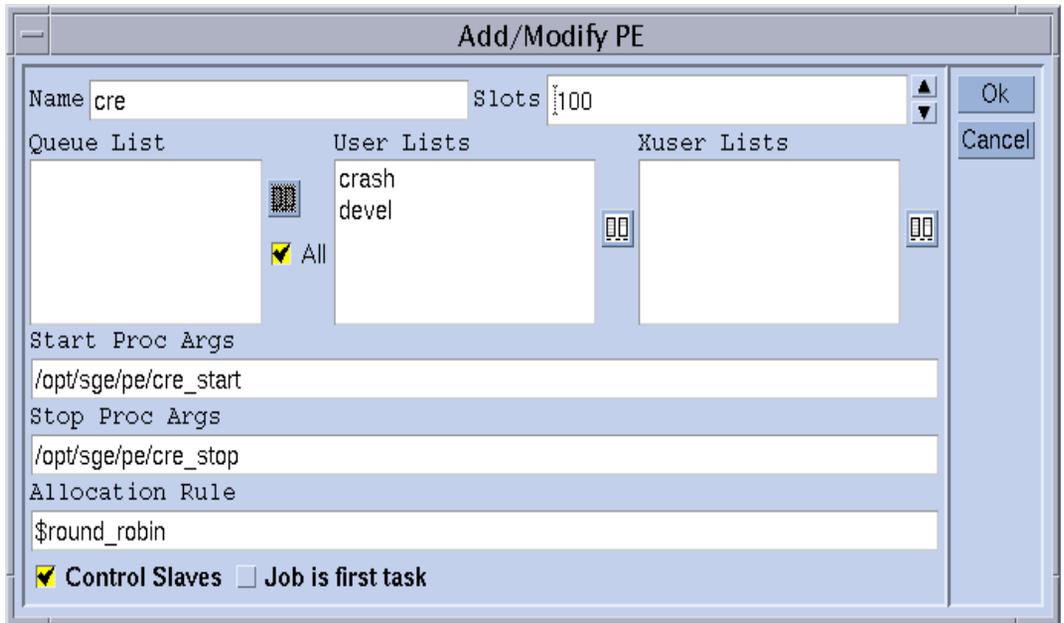


그림 10-2 Parallel Environment 정의 대화 상자

2. 236 페이지, “Parallel Environment 정의 매개변수의 설명” 절의 지침에 따라서 PE 정의를 추가합니다.

3. 변경사항을 저장하려면 **OK**, 변경사항을 삭제하려면 **Cancel**을 누릅니다.

OK 또는 Cancel을 누르면 대화 상자가 사라집니다.

Parallel Environment 정의 매개변수의 설명

- *Name* 입력 창은 수정 조작의 경우에 선택된 PE의 이름을 표시하거나 선언될 PE의 이름을 입력하는 데 사용할 수 있습니다.
- *Slots* 스핀 상자는 동시에 실행하는 모든 PE 작업에 의해 점유될 수 있는 작업 슬롯의 총 수를 입력하는 데 사용해야 합니다.
- *Queue List* 표시 영역은 PE가 사용할 수 있는 대기열을 표시합니다. *Queue List* 표시 영역의 오른쪽에 있는 아이콘 단추를 누르면 PE 대기열 목록을 수정할 수 있는 그림 10-3의 보기와 유사한 *Select Queues* 대화 상자가 표시됩니다.

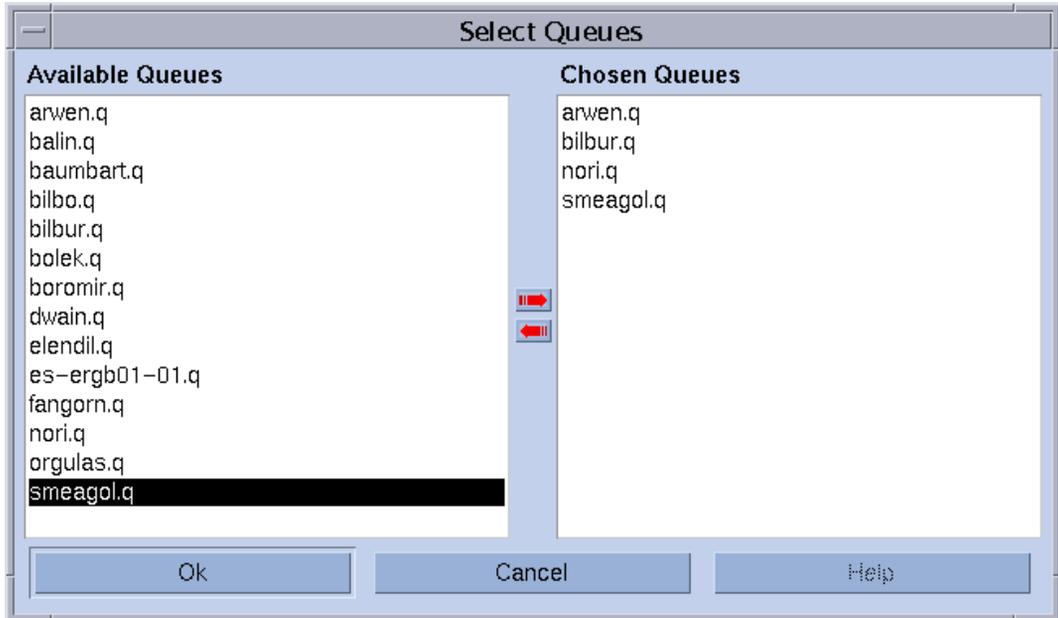


그림 10-3 Select Queues 대화 상자

- *User Lists* 표시 영역에는 PE에 액세스하도록 허용되는 사용자 액세스 목록이 들어 있습니다(209 페이지, “사용자 액세스 권한” 절 참조).
- *Xuser Lists* 표시 영역은 액세스가 거부되는 액세스 목록을 표시합니다.

양 표시 영역과 연관된 아이콘 단추를 누르면 그림 10-4의 보기와 유사한 Select Access Lists 대화 상자가 표시됩니다. 이들 대화 상자를 사용하여 양 액세스 목록 표시 영역의 내용을 수정할 수 있습니다.

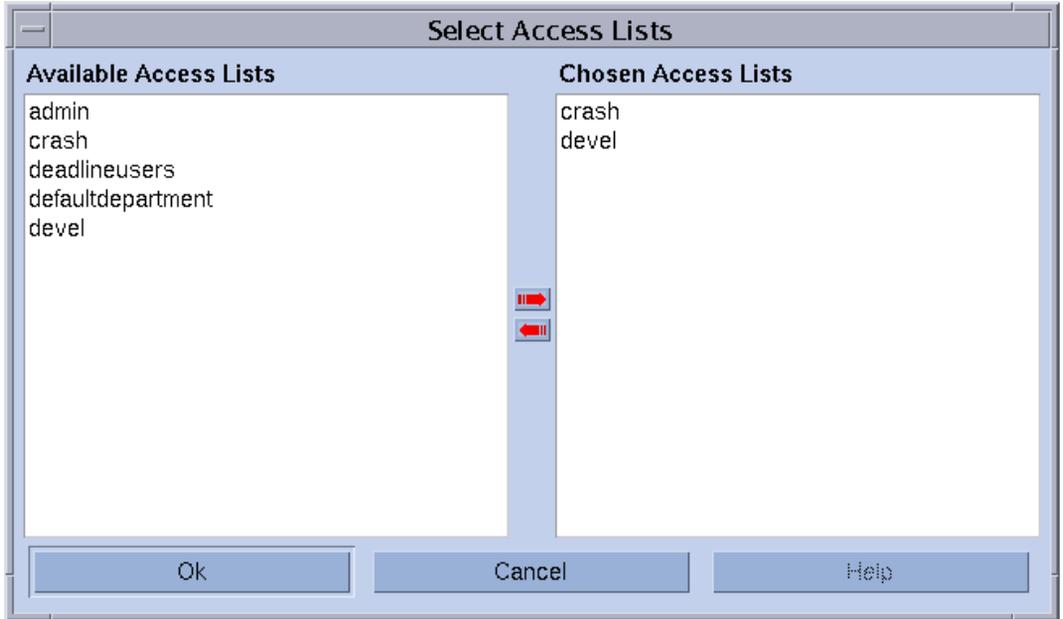


그림 10-4 Select Access Lists 대화 상자

- *Start Proc Args* 및 *Stop Proc Args* 입력 창은 PE 시작 및 중지 절차의 정확한 호출 순서를 입력하기 위해 제공됩니다(241 페이지, “PE 시작 절차” 또는 242 페이지, “PE의 종료” 절을 각각 참조하십시오). 첫 번째 인수는 대개 시작 또는 중지 절차 자체입니다. 나머지 매개변수는 절차에 대한 명령줄 인수입니다.

절차에 Sun Grid Engine 내부 런타임 정보를 전달하기 위해 다양한 특수 ID(\$ 접두어로 시작)를 사용할 수 있습니다. *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 *sge_pe* 항목에 모든 사용 가능한 매개변수의 목록이 들어 있습니다.

- *Allocation Rule* 입력 창은 PE가 사용하는 각 시스템에서 할당될 병렬 프로세스의 수를 정의합니다. 현재 양수 및 특수 값 *\$pe_slots*만 지원됩니다. *\$pe_slots*는 생성된 모든 프로세스들이 단일 호스트에 있어야 한다는 것을 의미합니다.
- *Control Slaves* 토글 단추는 병렬 업무가 Sun Grid Engine을 통해(즉, *sge_execd* 및 *sge_shepherd*를 통해) 생성되는지 여부 또는 대응하는 PE가 자체의 프로세스 작성을 수행하는지 여부를 선언합니다. Sun Grid Engine 시스템이 슬레이브 업무에 대한 모든 제어를 갖는 경우에는 이익이지만, 이 기능은 Sun Grid Engine에 대해 특별히 사용자 정의되는 PE 인터페이스에만 사용할 수 있습니다. 자세한 내용은 243 페이지, “PE 및 Sun Grid Engine 소프트웨어의 밀접한 통합” 절을 참조하십시오.

- *Job is first task* 토글 단추는 Control Slaves가 켜진 경우에만 의미가 있습니다. 이것은 작업 스크립트 또는 그의 하위 프로세스 중 하나가 병렬 응용 프로그램(이것은 대개 예를 들어 PVM의 경우입니다)의 병렬 업무 중 하나로 작용함을 표시합니다. 꺼지는 경우 작업 스크립트가 병렬 응용 프로그램을 시작하지만 참여하지는 않습니다(예를 들어, MPI의 경우에 mpirun을 사용할 때).

▼ 명령줄에서 PE를 구성하는 방법

다음 절에서 설명하는 적당한 옵션을 갖는 qconf 명령을 입력하십시오.

qconf PE 옵션

- `qconf -ap PE이름`

병렬 환경 추가—이 명령은 PE 구성 템플릿을 갖는 편집기(기본 vi 또는 \$EDITOR 환경 변수에 따라서)를 표시합니다. 매개변수 *PE이름*은 PE의 이름을 지정하며 이미 템플릿의 대응하는 필드에 채워집니다. 템플릿을 변경하고 디스크에 저장하여 PE를 구성하십시오. 변경할 템플릿 항목의 자세한 설명은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `sge_pe` 항목을 참조하십시오.

- `qconf -dp PE이름`

병렬 환경 삭제—이 명령은 지정된 PE를 삭제합니다.

- `qconf -mp PE이름`

병렬 환경 수정—이 명령은 구성 템플릿으로서 지정된 PE를 갖는 편집기(기본 vi 또는 \$EDITOR 환경 변수에 따라서)를 표시합니다. 템플릿을 변경하고 디스크에 저장하여 PE를 수정하십시오. 변경할 템플릿 항목의 자세한 설명은 *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `sge_pe` 항목을 참조하십시오.

- `qconf -sp PE이름`

병렬 환경 표시—이 명령은 지정된 PE의 구성을 표준 출력에 인쇄합니다.

- `qconf -spl`

병렬 환경 목록 표시—이 명령은 현재 구성된 모든 병렬 환경의 이름 목록을 표시합니다.

▼ 명령줄에서 구성된 PE 인터페이스를 표시하는 방법

- 다음 명령을 입력하십시오.

```
% qconf -spl
% qconf -sp PE 이름
```

첫 번째 명령은 현재 사용 가능한 PE 인터페이스의 이름 목록을 인쇄합니다. 두 번째 명령은 특정 PE 인터페이스의 구성을 표시합니다. PE 구성에 대한 자세한 정보는 sge_pe 매뉴얼 페이지를 참조하십시오.

▼ QMON으로 구성된 PE 인터페이스를 표시하는 방법

- QMON Main 메뉴에서 PE Config 단추를 누르십시오.

Parallel Environment Configuration 대화 상자가 표시됩니다(234 페이지, “QMON을 사용하여 PE를 구성하는 방법” 절 참조).

79 페이지, “고급 보기” 절의 예가 이미 최소한 4개, 최고(그리고 되도록이면) 16개 프로세스와 함께 사용될 PE 인터페이스 mpi(*message passing interface*를 의미)를 요청하는 병렬 작업을 정의합니다. Parallel Environment(PE) Specification 창의 오른쪽에 있는 단추를 사용하여 사용 가능한 PE 목록에서 원하는 병렬 환경을 선택하는 대화 상자를 팝업할 수 있습니다(그림 10-5 참조). Advanced Submission 화면의 PE Specification 창에 있는 PE 이름 뒤에 작업이 시작하는 병렬 업무의 수에 대한 요청된 범위를 추가할 수 있습니다.

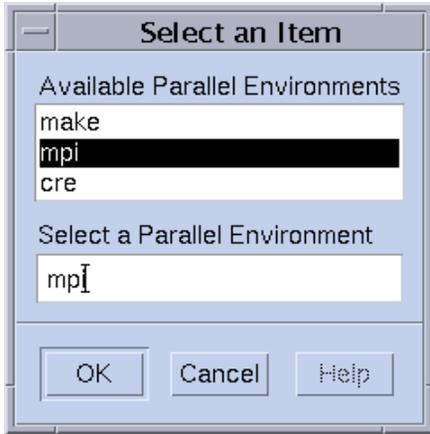


그림 10-5 PE 선택

위에서 설명한 병렬 작업 스펙에 대응하는 명령줄 제출 명령은 90 페이지, “명령줄에서 작업을 제출하는 방법” 절에 제공되며 동등한 요청을 공식화하기 위해 `qsub -pe` 옵션을 사용해야 하는 방법을 보여줍니다. *Sun Grid Engine 5.3 및 Sun Grid Engine, Enterprise 버전 5.3 참조 설명서의 qsub 항목이 -pe 구문에 대한 추가 정보를 제공합니다.*

병렬 작업에 대한 적합한 PE 인터페이스를 선택하는 것이 중요합니다. PE 인터페이스는 메시지 전달 시스템을 이용하지 않거나 다른 메시지 전달 시스템을 이용할 수 있고, 단일 또는 복수 호스트에 프로세스를 할당할 수 있으며, PE에 대한 액세스가 특정 사용자에게 거부될 수 있고, 대기열의 특정 세트만이 PE 인터페이스에 의해 사용될 수 있고 대기열 슬롯의 특정 수만이 임의의 시점에서 PE 인터페이스에 의해 점유될 수 있습니다. 그러므로 병렬 작업의 사용자 유형에 가장 적합한 사용 가능한 PE 인터페이스에 대해 Sun Grid Engine 관리에 질문해야 합니다.

PE 요청과 함께 83 페이지, “자원 요구사항 정의” 절에서 설명한 자원 요구사항을 지정할 수 있습니다. 이것은 PE 인터페이스에 적합한 대기열 세트를 사용자가 지정한 자원 요구사항 정의에도 맞는 대기열로 더욱 축소합니다. 예를 들어 다음 명령을 제출했다고 가정하십시오.

```
% qsub -pe mpi 1,2,4,8 -l nastran,arch=osf nastran.par
```

이 작업에 적합한 대기열은 PE 구성에 의해 PE 인터페이스 `mpi`와 연관되고 또한 `qsub -l` 옵션에 의해 지정되는 자원 요구사항 스펙을 만족하는 대기열입니다.

참고 - Sun Grid Engine PE 인터페이스 기능은 고도로 구성할 수 있습니다. 특히 Sun Grid Engine 관리는 사이트 고유의 수요를 지원하기 위한 PE 시작 및 중지 절차(sge_pe 매뉴얼 페이지 참조)를 구성할 수 있습니다. 환경 변수를 내보내는 `qsub -v` 및 `-V` 옵션을 사용하여 작업을 제출하는 사용자로부터 PE 시작 및 중지 절차로 정보를 전달할 수 있습니다. 특정 환경 변수를 내보내야 하는 경우 Sun Grid Engine 관리자에게 문의하십시오.

PE 시작 절차

Sun Grid Engine 시스템은 `exec` 시스템 호출을 통해 시작 절차를 호출하여 PE를 시작합니다. 시작 실행 파일의 이름과 이 실행 파일에 전달되는 매개변수는 Sun Grid Engine 시스템 안에서 구성할 수 있습니다. PVM 환경에 대한 그런 시작 절차의 예가 Sun Grid Engine 분배 트리에 들어 있습니다. 절차는 셸 스크립트와 셸 스크립트가 호출하는 C 프로그램으로 구성됩니다. 셸 스크립트는 C 프로그램을 사용하여 PVM을 정확히 시작합니다. 필요한 다른 모든 조작은 셸 스크립트에 의해 처리됩니다.

이 셸 스크립트는 `<sge 루트>/pvm/startpvm.sh` 아래에 있습니다. C 프로그램 파일은 `<sge 루트>/pvm/src/start_pvm.c`에서 찾을 수 있습니다.

참고 - 시작 절차는 단일 C 프로그램에 의해서도 감당될 수 있었습니다. 셸 스크립트는 샘플 시작 절차의 더 쉬운 사용자 정의를 허용하기 위해 사용됩니다.

예제 스크립트인 `startpvm.sh`는 다음 세 인수가 필요합니다.

- PVM이 시작될 호스트의 이름이 들어있는, Sun Grid Engine 소프트웨어에 의해 생성되는 호스트 파일의 경로
- `startpvm.sh` 절차가 호출된 호스트
- PVM 루트 디렉토리의 경로(대개 `PVM_ROOT` 환경 변수에 들어 있음)

이들 매개변수는 234 페이지, “QMON을 사용하여 PE를 구성하는 방법”에서 설명되는 방법을 통해 시작 스크립트에 전달될 수 있습니다. 매개변수는 런타임 중에 Sun Grid Engine에 의해 PE 시작 및 중지 스크립트에 제공된 매개변수에 섞여 있습니다. 예를 들어 필수 호스트 파일은 Sun Grid Engine에 의해 생성되고 파일의 이름이 특수 매개변수 이름 `$sge_hostfile`에 의해 PE 구성의 시작 절차에 전달될 수 있습니다. 모든 사용 가능한 매개변수의 설명이 *Sun Grid Engine 5.3* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*의 `sge_pe` 항목에 제공됩니다.

호스트 파일은 다음 형식을 갖습니다.

- 파일의 각 행은 병렬 프로세스가 실행될 호스트를 참조합니다.
- 각 행의 첫 입력은 호스트 이름을 표시합니다.
- 두 번째 항목은 호스트에서 실행될 병렬 프로세스의 수를 지정합니다.

■ 세 번째 항목은 다중프로세서 시스템의 경우에 사용될 프로세스 범위를 표시합니다. 이 파일 형식은 Sun Grid Engine에 의해 생성되고 수정됩니다. 다른 파일 형식이 필요한 (예를 들어 PVM) PE는 시작 프로세듀어 안에서 형식을 변환해야 합니다 (startpvm.sh 파일 참조).

PE 시작 절차가 Sun Grid Engine 시스템에 의해 시작되었을 때 즉시 PE를 실행합니다. 시작 절차는 0의 종료 상태를 갖고 종료해야 합니다. 시작 절차의 종료 상태가 0이 아닌 경우 Sun Grid Engine 소프트웨어가 오류를 보고하고 병렬 작업을 시작하지 않습니다.

참고 - 절차가 Sun Grid Engine 프레임워크에 통합되는 경우 추적하기 힘들 수 있는 모든 오류를 제거하기 위해 먼저 명령줄에서 (Sun Grid Engine 없이) 모든 시작 절차를 테스트하는 것이 좋습니다.

PE의 종료

병렬 작업이 끝나거나 중단되는 경우(qdel을 통해) 병렬 환경을 중단하는 절차가 호출됩니다. 이 절차의 정의와 의미론은 시작 프로그램에 대해 기술되는 것과 매우 유사합니다. 중지 절차는 또한 PE 구성에 정의될 수도 있습니다(예를 들어 234 페이지, “QMON을 사용하여 PE를 구성하는 방법” 참조).

중지 절차의 목적은 PE를 종료하고 모든 연관된 프로세스를 획득하는 것입니다.

참고 - 중지 절차가 PE 프로세스를 정리하지 못하는 경우 Sun Grid Engine 시스템이 PE 제어하에서 실행 중인 프로세스에 대한 정보를 갖지 않을 수 있으며 따라서 정리할 수 없습니다. 물론 Sun Grid Engine 소프트웨어는 실행한 작업 스크립트와 직접 연관된 프로세스를 정리합니다.

Sun Grid Engine 분배 트리에 PVM PE에 대한 중지 절차의 예도 들어 있습니다. 이 예제는 <sge루트>/pvm/stoppvm.sh 아래에 상주합니다. 이 절차는 다음 두 인수를 취합니다.

- Sun Grid Engine 시스템에 의해 생성되는 호스트 파일에 대한 경로
- 중지 절차가 시작되는 호스트의 이름

시작 절차와 유사하게, 중지 절차는 성공시에 종료 상태 0을 리턴하고 실패시에 0이 아닌 종료 상태를 리턴할 것으로 예상됩니다.

참고 - 절차가 Sun Grid Engine 프레임워크에 통합되는 경우 추적하기 힘들 수 있는 모든 오류를 제거하기 위해 먼저 명령줄에서 (Sun Grid Engine 없이) 모든 중지 절차를 테스트하는 것이 좋습니다.

PE 및 Sun Grid Engine 소프트웨어의 밀접한 통합

234 페이지, “QMON을 사용하여 PE를 구성하는 방법” 절에서 Control Slaves 매개변수의 설명이 병렬 업무의 작업이 Sun Grid Engine 구성요소 `sge_execd` 및 `sge_shepherd`에 의해 수행되는 PE는 자체 프로세스 작성을 수행하는 PE에 비해 이점을 제공한다고 언급합니다. 이것은 UNIX 운영 체제가 프로세스 계층의 작성자에 대해서만 신뢰할 수 있는 자원 제어를 허용한다는 사실 때문입니다. 병렬 응용 프로그램에 대한 올바른 회계, 자원 한계 및 프로세스 제어 같은 기능은 모든 병렬 업무의 작성자에 의해서만 향상될 수 있습니다.

대부분의 PE는 이들 기능을 구현하지 않으며 따라서 Sun Grid Engine 같은 자원 관리 시스템과의 통합을 위한 충분한 인터페이스를 제공하지 않습니다. 이 문제점을 극복하기 위해 Sun Grid Engine 시스템은 PE와의 밀접한 통합을 위한 고급 PE 인터페이스를 제공하는데, 이 인터페이스는 업무 작성의 책임을 PE에서 Sun Grid Engine 소프트웨어로 이전합니다.

Sun Grid Engine 분배에 PVM 공용 도메인 버전 또는 미국 아르곤 국립 연구소의 MPICH MPI 구현에 대한 그런 밀접한 통합의 두 예제가 들어 있습니다. 예제는 각각 `<sge 루트>/pvm` 및 `<sge 루트>/mpi` 디렉토리에 들어 있습니다. 디렉토리는 사용법 및 모든 현재 제한사항을 설명하는 README 파일 뿐 아니라 추가로 느슨하게 통합된 비교용 인터페이스의 변형을 포함합니다. 자세한 내용은 해당 README 파일을 참조하십시오.

참고 – PE와의 밀접한 통합 수행은 고급 업무이며 PE 및 Sun Grid Engine PE 인터페이스의 전문적인 지식이 필요할 수 있습니다. Sun 지원 대표 분배자에게 도움을 요청할 수 있습니다.

오류 메시지

이 장에서는 Sun Grid Engine 5.3 오류 메시지 절차에 대해 설명합니다. 이 정보를 이용하면 소프트웨어 사용시 발생할 수 있는 모든 문제점을 해결하는 데 도움이 될 수 있습니다.

Sun Grid Engine 5.3 소프트웨어가 오류 보고서를 검색하는 방법

Sun Grid Engine 소프트웨어는 특정 파일에 메시지를 기록하여 및/또는 전자 우편(이메일)에 의해 오류 또는 경고를 보고합니다. 사용되는 로그파일은 다음과 같습니다.

■ 메시지 파일:

`sgc_qmaster`, `sgc_schedd` 및 `sgc_execd`에 대한 별도의 메시지 파일이 있습니다. 파일은 동일한 파일 이름은 `messages`를 갖습니다. `sgc_qmaster` 로그파일은 마스터 스펴 디렉토리에 상주하고, `sgc_schedd` 메시지 파일은 스케줄러 스펴 디렉토리에 상주하고 실행 데몬의 로그파일은 실행 데몬의 스펴 디렉토리에 상주합니다 (스푼 디렉토리에 대한 자세한 정보는 25 페이지, “루트 디렉토리의 스펴 디렉토리” 절 참조).

메시지 파일은 다음 형식을 갖습니다.

- 각 메시지는 단일 행을 차지합니다.
- 메시지는 수직선 기호(`|`)에 의해 분리되는 5개 구성요소로 나뉘어집니다.
- 첫 번째 구성요소는 메시지에 대한 시간 기록입니다.
- 두 번째는 메시지를 생성하는 Sun Grid Engine 데몬을 지정합니다.
- 세 번째는 데몬이 실행하는 호스트이름입니다.

- 네 번째는 주의의 경우 N, 정보의 경우 I(둘 다 정보 목적만을 위해), 경고의 경우 W(잘못된 것이 있을 수 있음), 오류의 경우 E(오류 조건이 감지되었음) 또는 위험의 경우 C(프로그램 중단을 가져올 수 있음) 중 하나인 메시지 유형입니다.
- 5번째는 메시지 텍스트입니다.

참고 - 어떤 이유로 오류 로그파일이 액세스 불가능한 경우 Sun Grid Engine은 대응하는 호스트의 /tmp/sge_qmaster_messages, /tmp/sge_schedd_messages 또는 /tmp/sge_execd_messages에 오류 메시지를 기록하려 시도합니다.

- 작업 STDERR 출력:

작업이 시작되자마자, 작업 스크립트의 표준 오류(STDERR) 출력이 파일로 경로 재지정됩니다. 파일 이름과 위치는 기본값을 따르거나 특정 qsub 명령줄 스위치에 의해 지정될 수 있습니다. 자세한 정보는 *Sun Grid Engine 5.3 관리 및 사용 설명서* 및 *Sun Grid Engine, Enterprise 버전 5.3 참조 설명서*를 참조하십시오.

일부 상황에서는 Sun Grid Engine이 사용자 및/또는 관리자에게 이메일을 통해 오류 이벤트에 대해 통지합니다. Sun Grid Engine이 전송하는 메일 메시지에는 메시지 본문이 없습니다. 메시지 텍스트는 완전히 메일 주제 필드에 들어 있습니다.

여러 오류 또는 종료 코드의 결과

표 11-1은 여러 작업 관련 오류 또는 종료 코드의 결과를 나열합니다. 이들 코드는 Sun Grid Engine 작업의 모든 유형에 유효합니다.

표 11-1 작업 관련 오류 또는 종료 코드

스크립트/방법	종료 또는 오류 코드	결과
작업 스크립트	0	성공
	99	다시 대기열에 들어감
	나머지	성공:회계 파일의 종료 코드
프롤로그/에필로그	0	성공
	99	다시 대기열에 들어감
	나머지	대기열 오류 상태: 작업이 다시 대기열에 들어감

표 11-2는 병렬 환경(PE) 구성과 관련된 작업의 오류 또는 종료 코드의 결과를 나열합니다.

표 11-2 PE 관련 오류 또는 종료 코드

스크립트/방법	종료 또는 오류 코드	결과
pe_start	0	성공
	나머지	대기열이 오류 상태로 설정됨, 작업이 다시 대기열에 들어감
pe_stop	0	성공
	나머지	대기열이 오류 상태로 설정됨, 작업이 다시 대기열에 들어가지 않음

표 11-3은 대기열 구성과 관련된 작업의 오류 또는 종료 코드의 결과를 나열합니다. 이들은 대응하는 방법이 겹쳐져진 경우에만 유효합니다.

표 11-3 대기열 관련 오류 또는 종료 코드

스크립트/방법	종료 또는 오류 코드	결과
작업 시작기	0	성공
	나머지	성공, 기타 특수 의미 없음
일시중단	0	성공
	나머지	성공, 기타 특수 의미 없음
재개	0	성공
	나머지	성공, 기타 특수 의미 없음
종료	0	성공
	나머지	성공, 기타 특수 의미 없음

표 11-4는 체크포인팅과 관련된 작업의 오류 또는 종료 코드의 결과를 나열합니다.

표 11-4 체크포인팅 관련 오류 또는 종료 코드

스크립트/방법	종료 또는 오류 코드	결과
검사점	0	성공
	나머지	성공—그러나 커널 검사점의 경우 특수한 의미: 검사점 실패; 발생하지 않았음.
이주	0	성공
	나머지	성공—그러나 커널 검사점의 경우 특수한 의미: 검사점 실패; 발생하지 않았음. 이주가 발생함.
재시작	0	성공
	나머지	성공, 기타 특수 의미 없음
정리	0	성공
	나머지	성공, 기타 특수 의미 없음

디버그 모드에서 Sun Grid Engine 프로그램 실행

일부 심각한 오류 조건의 경우 오류 기록 메커니즘이 문제점을 식별하기에 충분한 정보를 생성하지 않을 수 있습니다. 그러므로 Sun Grid Engine은 디버그 모드에서 거의 모든 보조 프로그램과 데몬을 실행하는 능력을 제공합니다. 제공되는 정보의 범위와 깊이에서 변하는 여러 가지 디버그 레벨이 있습니다. 디버그 레벨은 0부터 10까지의 범위이며 10은 가장 상세한 정보를 전달하는 레벨이고 0은 디버깅을 끕니다.

디버그 레벨을 설정하기 위해 `.cshrc` 또는 `.profile` 자원 파일에 대한 확장이 Sun Grid Engine 분배와 함께 제공됩니다. `csh` 또는 `tcsh` 사용자의 경우 `<sge 루트>/<util>/dl.csh` 파일이 포함됩니다. `sh` 또는 `ksh` 사용자의 경우 대응하는 파일은 `<sge 루트>/util/dl.sh`로 명명됩니다. 파일은 표준 자원 파일로 “sourced” 되어야 합니다. `csh` 또는 `tcsh` 사용자의 경우 다음 행을

```
source <sge 루트>/util/dl.csh
```

`.cshrc` 파일에 포함시키십시오. `sh` 또는 `ksh` 사용자의 경우 행을

```
. <sge 루트>/util/dl.sh
```

`.profile` 파일에 추가하는 것은 동등합니다. 사용자가 이제 로그아웃하고 다시 로그인할 때 다음 명령을 사용하여 디버그 레벨을 `level`로 설정할 수 있습니다.

```
% dl level
```

`level`이 0보다 큰 경우 그 이후부터 Sun Grid Engine 명령을 시작하면 명령이 추적 출력을 `STDOUT`에 기록하도록 강제됩니다. 추적 출력은 강제되는 디버그 레벨에 따라서 소스 코드 행 번호 정보(오류 보고에 도움이 됨)와 함께 내부적으로 호출되는 프로그램 모듈의 이름뿐 아니라 경고, 상태 및 오류 메시지를 포함할 수 있습니다.

참고 - 상당한 화면이동 행 버퍼(예: 1000행)를 갖는 창에서 디버그 추적을 감시하는 것이 유용할 수 있습니다.

참고 - 창이 `xterm`인 경우 `xterm` 로깅 메커니즘을 사용하여 나중에 추적 출력을 시험하기 원할 수 있습니다.

디버그 모드에서 Sun Grid Engine 데몬 중 하나를 실행하는 것은 데몬이 추적 출력을 기록하기 위해 터미널 연결을 유지하는 결과를 갖습니다. 사용자가 사용하는 터미널 에뮬레이션의 인터럽트 문자를 입력하여 중지할 수 있습니다(예를 들어, `Control-C`).

참고 - 디버그 모드를 끄려면 디버그 레벨을 다시 0으로 설정하십시오.
