Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Departement Elektrotechnik
Institut für Elektronik

Resource Management for Superclusters

Student Thesis SA-2001
Winter Term 2000/2001

Guillaume Thomassin

Tutor: Josef Němeček
Supervisor: Prof. Dr. G. Tröster

Abstract

Clusters of workstations are now commonly found in high performance computing centers and tend to replace mainframe type supercomputers. They steadily grow in size reaching today several thousand nodes. Their complexity is also increasing: more nodes, complex network topologies, heterogeneous resources across the cluster, ...

The efficient use and administration of those big clusters, or superclusters, is not possible with currently available softwares: they lack important functionality and most of the time reliability.

This project exposes the different functionality requirements such software should meet regarding *resource management*: support of wide type of jobs, customisable and efficient scheduling and resource allocation, extensive monitoring, scalability, reliability, security. A priority based backfill algorithm fulfilling those requirements is proposed along with an user interface (both shell commands and graphical user interface) and some internal structures.

Contents

List of Figures

List of Tables

Listings

Introduction

The world of high performance computing has mutated in the last years. The big mainframes are now challenged by clusters of workstation. Clusters are more and more preferred to supercomputers, because they are made of off-the-shelf components and can achieve the same computing power for a fraction of the cost of supercomputers.

At the beginning the clusters were simple workstations interconnected with a simple network. They suffered a lot from the high latencies and limited bandwidth they could have. But in the past years, high bandwidth and low latency networks like Myrinet, Giganet or the T-Net have been developped to make the clusters more scalable. Some of current clusters are composed of over thousand workstations and clusters with tens of thousand or even hundred of thousand nodes are planned. These big clusters are called *superclusters*, to express the same difference in computing power as between *computer* and *supercomputer*. These superclusters need special software to be efficiently managed and used.

Such a software is needed as the size of cluster keeps growing and the administrator needs tools to configure and manage thousand of workstations. The superclusters are not homogeneous anymore complicating the task of allocating the jobs to the right nodes without spoiling computation power. The high number of workstations leads too frequent failure of a workstation: with a 10'000 nodes cluster and a mean time between failure of 20'000 hours for the harddisk it means statistically a harddisk must be changed each day. The change in the resources implied by those events must be dealt automatically. And none of the currently available software fulfill those requirements.

This project is part of the COSMOS project. COSMOS stands for *Commodity Supercluster Management Operating System*. It is a project common to the Electronic Laboratory of the Swiss Federal Institute in Zurich and Supercomputing Systems AG. The aim of COSMOS is to group all the management and monitoring tools for the different aspects of the cluster (nodes, network, storage, scheduling, accounting, ...) in one software. It will be a set of modules interacting around a framework allowing for the addition of other modules or site specific modules.

Goal of this projects is to establish the specifications of the resource management part of COSMOS so that it can be used on the future superclusters.

In chapter 1, the different needs of the users and administrators are analized. The exisiting softwares are quickly described.

In chapter 2, the functionnality requirements of COSMOS are exposed. This includes the specifications for the job types, the submission procedure, the job control, the scheduling, the resource allocation, the monitoring and accounting and the user management.

Chapter 3 is a comparison of the features of the leading existing software and COSMOS.

In chapter 4, different aspect of the scheduling are described. Several algorithm are analized and an algorithm is proposed.

Chapter 5 and 6, describe the user interface of COSMOS; a quick description of the shell commands in chapter 5 and the GUI in chapter 6. Some internal structures like the data structures used for resource management and the communication between resource management and application management are detailed.

Chapter 1

The role of resource management

1.1 User needs

Users of a cluster are the people submitting jobs to the cluster. They need tools to submit jobs. These must include shell commands and preferably also a GUI. Through the resource management system the user should be able to submit several type of jobs: serial jobs (1 program, 1 slot), parallel homogeneous jobs (1 program, n slots), parallel heterogeneous jobs (m programs, n slots). These jobs being either batch or interactive jobs[1]. The user should have the possibility to tell the system to run scripts before and after the execution of the job, as well as copying files on each node before the job execution[2] (this is also know as file staging) and copying back files on the server after the job execution. Another important aspect for the user is the possibility to have dependencies between several jobs. A user could submit three jobs (A, B and C) at the same time where C depends on the output of A and B, and therefore the scheduler would schedule C only when A and B are finished. And if A or B finish abnormally, C is not scheduled.

Once submitted the user should be able to modify (including the possibility to kill) and monitor its batch jobs. The monitoring should be possible through shell commands and through the GUI. The informations given to the user depends from the state of the job. If the job is still in waiting in the queue interesting information for the user might be:

[1] A batch job is a job which is run outside an interactive login session as opposed to an interactive job where the user interacts with the job as it is being executed

[2] The job is not started before all the files have been copied. The goal of these local copies is to improve the speed of file access during execution.

- job configuration,
- time elapsed since job submission,
- number of jobs waiting,
- number of jobs waiting with a higher priority,
- estimated waiting time,
- priority,
- history of all modifications to the job,
- depending on permissions some informations about other jobs.

While the job is running:

- job configuration,
- time left before the job is killed,
- CPU/Network/Storage/Memory usage (for the whole job or on each node),
- time of the last checkpoint,
- standard output and standard error,
- nodes on which the job is running.

The different informations are also given in the job output at the end of the job execution. Beside this, the user can see his overall use of the cluster over a certain period of time, his jobs history.

The user can also be notified at different stages of the job life (job submission received, job queued or rejected, job scheduled, job finished or killed). This notification could be made to the user standard input, through email, SMS on his mobile phone, ...

1.2 Administrator needs

The Administrator is responsible for the good functioning and good usage of the cluster. His roles are quite large.

He must administer and configure the different components of the cluster (the nodes, the network switches, the storage, ...). Besides the administration of the different components of the cluster, the administrator must also be able to monitor all the components (CPU temperature, hard disk status, network latency, ...) to see if there is any problem, or if any problem could occur (a rising CPU temperature might mean a fan is out of order).

Another aspect of cluster administration is the user and group administration. The administrator has to define which users have access to the cluster with which limitations, with the possibility to disable users or groups for a certain amount of time.

The last aspect is the resource management. The administrator must plan the supercluster usage by defining applications queues and configuring the scheduling algorithm. This includes defining the "share" of the cluster different users or groups may use (with soft and hard limits), the disponibility of the cluster based on size of the job, user, load of the cluster. He must also be able to control the job queue: forcing a job to run, holding jobs, overriding user choices, ... He must be able to run scripts on the hosts before and after each job (eventually to have different scripts depending on the user/group/node).

The resource management should also react "smartly" to hardware problems (automatically restart a job if one of the nodes crashed, move an application to a different node because a fault trap occurred, ...), without the need of human intervention in most cases.

And naturally, the administrator needs monitoring tools to control the good functioning of the cluster. He must be able to monitor CPU usage, memory and storage utilization for the whole cluster, for each node, for each job, for each user or group. Those information are also important for planning upgrades and system down time, adapting the scheduling algorithm. He also has access to the same job monitoring possibilities as the user with improved options: aggregating the information of several users, groups.

The same notifications might be used for the administrator as for the user.

1.3 Overall needs

1.3.1 Integrated management

The supercluster management software should be a one-stop program for the administrator. He should be able to control all the aspects of the cluster from a single program. This is the best solution to have a uniformized user interface for the different aspects of the management (user, system, LAN, node, application, resource, storage and high-speed network management). Integrated management is also the best way to have the differents modules interact efficiently, specially for resource and application management.

1.3.2 Scalability

The supercluster management software can be used for small systems (several nodes) as well as for large systems (many thousands of nodes). The requirements of the management systems must grow at most in linear relationship with the system size. The management systems must not have a maximal number of nodes or network switches.

1.3.3 Remote manageability

All management operations must be possible remotely (through telnet/ssh with shell commands, client-server, ...).

1.3.4 Reliability & Availability

The cluster should not have one or several single-point of failure. This is also the case for the management part. Therefore the management center should be run on a high availability cluster. There must be fail-over mechanism reconfiguring the cluster and the management software, and if necessary restarting affected applications and services. This reconfiguration should be transparent to the user and as automated as possible (in most cases it should not require an administrator intervention).

The management system should also be immune to job crash: a buggy job if it crashes should not affect other jobs.

1.3.5 Security

The management system should provide a high level of security, with the possibility to disable some features to match site-specific laws and regulations. The management system must have a proper authentification mechanism, and a security enforcement mechanism for file or device access (cluster users and groups might be different than UNIX users and groups). Communication should be done encrypted for confidentiality and authenticity[3] reasons. The different components must naturally be exempt from local or remote exploit and resistant to attacks (denial of service or other).

[3] Data integrity and user authentification

1.3.6 Low overhead

The management should not spoil the available resources. The management software must be both effective and efficient. The amount of additional hardware components must be as small as possible. The maximum additional cost should not exceed 5% of the total. There should also be low overhead for the administrator.

1.3.7 Modular expandability

The system management must be modular to support different HSN (High-speed network), node architectures, parallel environments (such as different MPI implementations) and site-specific extensions.

1.3.8 Platform interoperability

The management system software supports major UNIX platforms and LINUX. Heterogeneous systems must be supported to allow growing superclusters and superclusters farms over a longer time period.

1.4 Existing software

1.4.1 LSF

LSF (Load Sharing Facility) is developed by *Platform Computing Corporation* [1]. It is commercial closed-source software. LSF is a suite of software to transform a distributed network of workstation into a single shared computing resource. The base software (LSF 4.0 Standard Edition) offers load sharing and batch scheduling. The Professional edition adds capacity planning, accounting and improved reporting. Several other tools are also available: LSF Analyzer (workload analyzer, charge-back accounting reports), LSF Multicluster (resource sharing among multiple clusters), LSF Make (distributed, parallel Make program), LSF Parallel (manages parallel job execution) and LSF Job Scheduler (fault-tolerant, scalable calendar- and event-driven scheduling).

LSF works with all major UNIX platforms (Sun Solaris, HP-UX, IBM AIX, Compaq Tru64 UNIX, SGI IRIX, Linux for Intel and Alpha) and is compatible with Windows NT and 2000 workstations, and Cray supercomputers.

LSF was designed to run parallel tasks on a NOW (network of workstations) using idle or under-utilized workstations. The NOW is not necessarily a dedicated NOW for high-performance computing.

1.4.2 Codine/GRD

Codine and GRD (Global Resource Director) are policy and load management software for clustered systems. They were developed by *Gridware*, which was taken over by *SUN Microsystems*. The product name is now Sun Grid Engine [4]. Codine is a load management software, GRD is an extended version respecting policies for the scheduling.

It works on most UNIX platforms (Sun Solaris, HP-UX, IBM AIX, Compaq Tru64 UNIX, SGI IRIX, Linux for Intel and Alpha) and on Cray supercomputers. The recent takeover of *Gridware* by *SUN* may focus the development of future versions on Solaris, even if Sun is committed to maintaining and expanding cross-platform support [5]. The source code should be released under an industry-accepted open source license (no date has yet been announced).

1.4.3 PBS

The Portable Batch System (PBS) is a flexible batch queuing and workload management system originally developed by *Veridian Systems* for *NASA*. It comes in two flavors: *OpenPBS* [2] which is a free version and a commercial version *PBS Pro* [3]. PBS has a flexible job scheduler with the possibily to write its own scheduler.

PBS works on most UNIX platforms (Sun Solaris, HP-UX, IBM AIX, Compaq Tru64 UNIX, SGI IRIX, Linux for Intel and Alpha, Free/NetBSD), on parallel supercomputers (Cray T3D and T3E, IBM SP2, SGI Origin2000) and on some vector supercomputers.

1.4.4 Other software

A lot of resource management software exists. Most of them have been developped by universities for their own cluster. They vary greatly in quality. The two best known are NQS and Condor.

NQS

NQS [6] is the ancestor of the batch queuing systems. It exists since 1985 and was developed by the University of Sheffield (UK). It was and still is used on supercomputers (either vector or parallel). It is released under the GNU General Public License.

NQS was first written when high performance computing was done on supercomputers. It is therefore not really adapted to clusters. PBS, which is based on NQS, is more suited for clusters. NQS is capable to check the memory requirements so the global use of memory from all running jobs does not exceed the supercomputer's total memory.

Condor

Condor [7] is a software system that runs on a cluster of workstations to harness wasted CPU cycles. It is developed by the University of Wisconsin.

Chapter 2

COSMOS functionality specifications

This chapter will list functionalities that need to be or should be integrated in COSMOS.

2.1 Overall requirements

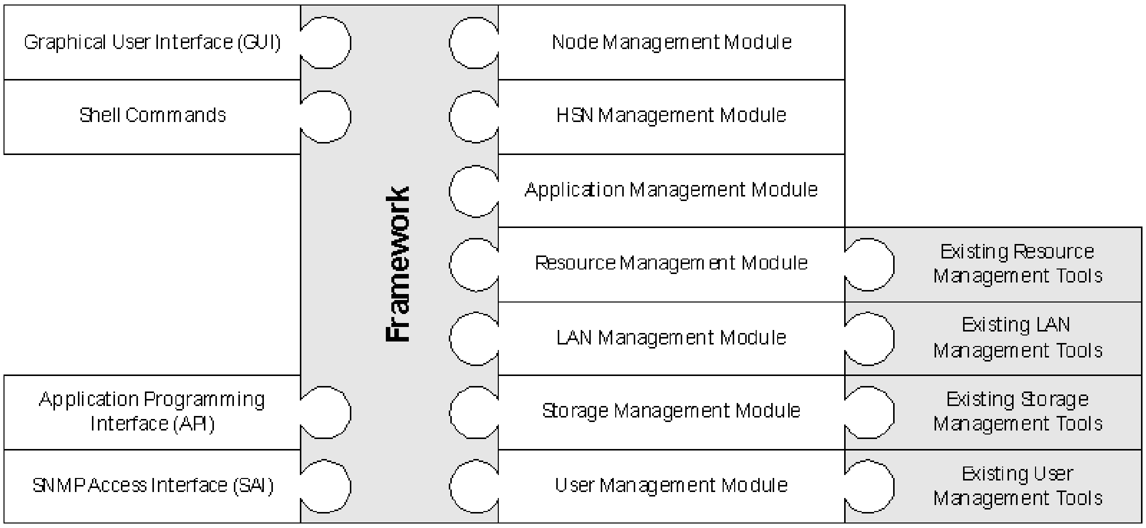

Figure 2.1: COSMOS Framework design

COSMOS will satisfy the requirements described in 1.3 . It will consist of a framework to interconnect the different modules needed for the cluster management, as shown in figure 2.1. The framework holds the central data structures and maintains the connections between the distributed modules. The interfaces between the modules are defined to support different platforms and implementations of the management modules. Thus the *integrated management* and *modular expandability* requirements are satisfied.

The *remote manageability* can be accomplished through telnet/ssh and shell scripts.

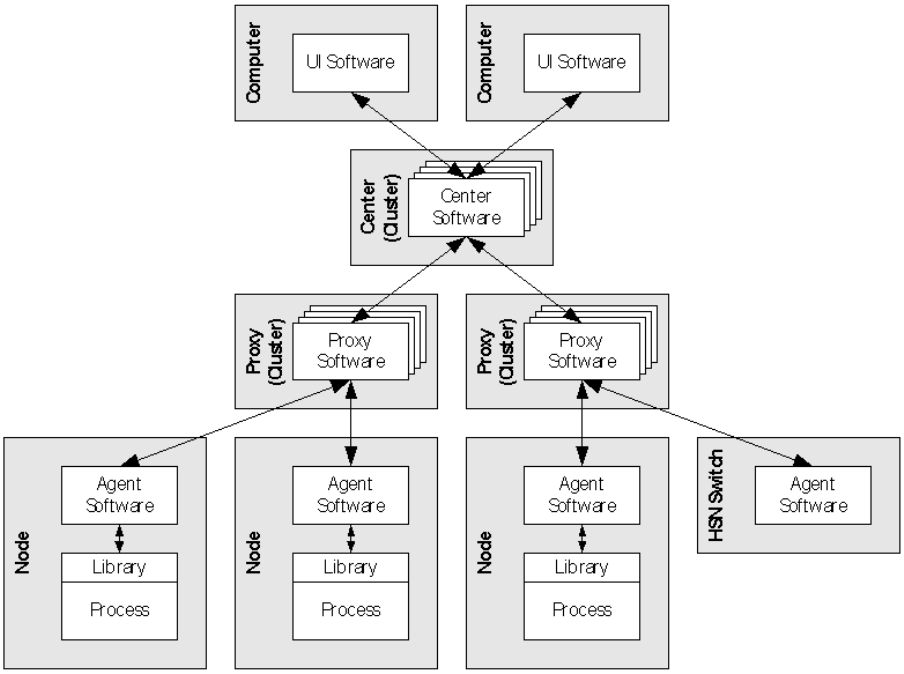

Figure 2.2: Software component distribution

To maximize the availability and scalability, the management systems is distributed, as show in figure 2.2. The management center is a high availability cluster. This way the management center can be tolerant to hardware failure and is more scalable (if the load on the servers is chronically too high, new servers can be included in the cluster). For a good scalability, proxy servers between the center and the nodes are necessary[1], otherwise a bottleneck appears between the center and the nodes (bandwidth bottleneck and limited number of connections on the center side). For the management part it is as in figure 2.3.

[1]for large clusters only

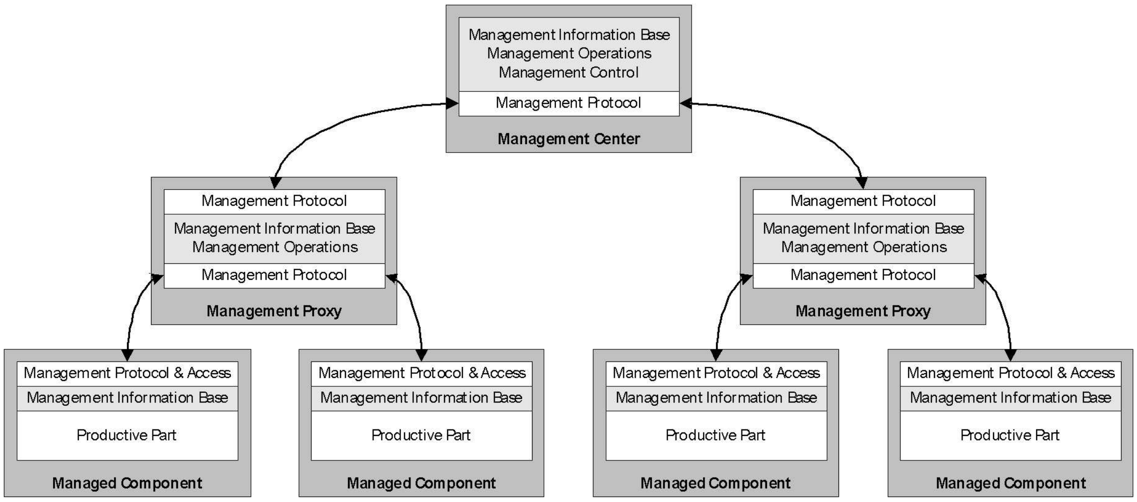

Figure 2.3: Management components

2.2 Mandatory requirements for resource management

2.2.1 Job types

COSMOS will allow interactive and batch jobs. The jobs can be homogeneous (one executable) or heterogeneous (several executables depending on the platform on which it runs). Heterogeneous jobs allow better utilization of the resources in a heterogeneous cluster. COSMOS will also support *variable size jobs* (i.e. jobs were the number of nodes it needs is not fixed, for example 16 to 32 nodes). This can improve the cluster utilization[2] and the waiting time[3] because these jobs can adapt themselves to the available nodes of the cluster.

2.2.2 Job control

The user will be able to submit jobs through shell commands or a GUI, for both interactive and batch jobs. COSMOS will provide the possibility for the user to save the parameters in a job script, for reuse at a later time. Some parameters may also be defined in a configuration file (`.cosmos`) or through environment variables. The parameters for job submission are:

[2]see 4.1.4
[3]see 4.1.2

- job script or executable
- job name
- job arguments
- priority (users may only lower the priority)
- scheduling information (for more detail see 2.2.4)
- output redirection
- pre-exec shell script for user and administrator (also known as prologue)
- post-exec shell script for user and administrator (also known as epilogue)
- file staging
- hard resource requirements
- soft resource requirements
- queue to be used
- number of nodes
- job duration
- job restarting in case of failure
- notification (for more detail see 2.2.7)
- job dependencies

The user can modify all these parameters as long as the job has not been started. He can also hold the job. If the job is still waiting it won't be started, and if it is running he will be checkpointed and wait until it is released. Jobs can also be deleted or stopped and rescheduled. In this case the resource management and the application management will kill all the child processes of the job on all nodes and clean temporary files. Naturally, a user can only act on his own jobs.

The administrator has the same possibilities for holding, deleting or modifying jobs for all users. He may rise the priority of a job, extend its duration while it is running.

2.2.3 Queue control

The administrator can create, modify, delete, suspend and resume queues. The parameters relevant for each queue are:

- queue
- queue type (executing queue, routing queue)
- type of jobs accepted
- calendar (when the queue is enabled and disabled)
- available resources: number of slots, architecture, ...
- pre-exec shell script
- post-exec shell script
- load limits
- resources limits: maximum CPU time, maximum memory/network/storage usage, ...
- user/group access

2.2.4 Scheduling[4].

COSMOS will include complex scheduling possibilities: space and time sharing, gang scheduling[5]. The scheduler will respect fair share between user and groups, as well as priorities. COSMOS will allow deadline jobs and calendar scheduling.

2.2.5 Resource allocation

COSMOS will try to allocate the resources as efficiently and smartly as possible, specially regarding the nodes positions within the supercluster topology. This is important in clusters having a high speed network, to reduce latency in the message passing, by having the different nodes directly connected to each other if possible. It should also try to put jobs with important I/O needs on nodes connected directly to the SAN (Storage Area Network). The scheduler will also try to satisfy as much of the soft resource requirements as possible.

[4]for more detailed information see chapter 4 on page 23

[5]see 4.2.4

2.2.6 User and group management

COSMOS will include a user management module. The administrator will be able to define users and groups. Users may belong to several groups. For each user or group the administrator will be able to define the share of the cluster they are allowed to use. Users or group may be disabled or enabled, to ban a user or a group from using the cluster for a certain amount of time.

2.2.7 Monitoring and accounting

The resource management will provide monitoring and notification for the users and the administrator. The user will be able to monitor its waiting or running jobs, as well as have access to accounting informations about his previous jobs. The administrator will be able to monitor jobs, queues, resource usage either for a single entity or aggregating information from several entities (for example monitoring all the resources usage from all running of jobs of a specific group).

All resource usage of all jobs must also be logged for accounting. These informations are the one reported by the application management when a job is finished or killed. They may be used for accounting and charging the different users according to their usage of the cluster.

2.2.8 Security

There are two aspects regarding security. The first is forbidding unallowed access or use of resources. For this aim, COSMOS will include an authentification mechanism which will check file and devices access of the jobs. The second is the confidentiality of data used by or produced by the jobs. This is solved by encrypting the communications over the network. This second aspect is not always necessary depending on the usage of the cluster.

2.2.9 Interface to the framework and other modules

COSMOS will also be able to be used as a “proxy” for other resource management system. It will be able to submit jobs to other clusters using a different management system. This way COSMOS can be an interface to several clusters hiding this fact to the user (he always submit his jobs to the COSMOS management center, but his jobs might be executed on a remote cluster using a different management system).

2.3 Optional requirements for resource management

2.3.1 Web based monitoring and control tools

Beside the shell commands and the GUI, web based tools facilitate the access from remote locations, from computers behind firewalls and from machines lacking a telnet/ssh for example. These tools would consist of a set of forms to accomplish the most used commands.

2.3.2 Global */proc*

Each node has its own `/proc` [6]. By creating a global */proc* on the management center, the state of the different nodes could be easily checked, either through a telnet/ssh session or with shell scripts, wihtout the need to log on each node.

[6]/proc is a pseudo-filesystem which is used as an interface to kernel data structures rather than reading and interpreting /dev/kmem.

Chapter 3

Comparison between COSMOS, LSF, GRD and PBS

Feature	Resource Management Software			
	COSMOS	LSF	GRD	PBS
Scheduling				
Scheduling respects:	★★★★	★★	★★	★★
• Current load on nodes	√	√	√	√
• Required resources on nodes	√	√	√	√
• Node position within supercluster topology	√			
Supercluster sharing:	★★★★	★★	★	★★
• Space sharing	√	√	√	√
• Time sharing	√	√		√
• Gang scheduling	√			
Scheduling policies:	★★★★	★★★★	★★★★	★★★★
• Fair share (ticket)	√	√	√	√
• Deadline	√	√	√	√
• Time frame, run window or calendar	√	√	√	√
• Dependencies	√	√	√	√
Re-submit to other cluster	√	?	√	?

Table 3.1: Resource management feature comparison

Features	COSMOS	LSF	GRD	PBS
Submission				
Job types:	★★★★	★★	★★	★★
• Serial job (1 executable on 1 node)	√	√	√	√
• Homogeneous parallel job (same executable)	√	√	√	√
• Heterogeneous parallel job (different executables)	√			
• Batch job	√	√	√	√
• Interactive job	√	√	√	√
• variable size	√			
Submission and control of jobs through:	★★★★	★★★★	★★★★	★★★★
• GUI	√	√	√	√
• shell commands	√	√	√	√
Remote submission and administration	√	√	√	√
Scripts:	★★★★			
• User-defined job-script	√	√	√	?
• User-defined prologue and epilogue scripts	√	?	?	?
• Administrator-defined prologue and epilogue scripts	√	?	?	?
Monitoring				
Resource monitoring and history	★★★★	★★★	★★★	★★
• per process/application	√	√	√	
• per user	√	√	√	√
• per group	√			√
Resource reporting and accounting	★★★★	★★★	★★★	★★
• per process/application	√	√	√	
• per user	√	√	√	√
• per group	√	√	√	√
Notification:	★★★★			
• through e-Mail	√	?	?	?
• through site specific scripts (for SMS, pager, ...)	√			
Resource consumption excess checks and traps	√	√		√

Table 3.1: Resource management feature comparison

Features	COSMOS	LSF	GRD	PBS
Application management				
Redirection of output:	★★★★	★★★★	★★★★	★★★★
• standard output	√	√	√	√
• standard error	√	√	√	√
Redirection of output to:	★★★★	★★★★	★★★	★★
• defined standard output	√	√	√	?
• file	√	√	√	√
• e-Mail	√	√		
Redirection of input to any node	√	√	√	
Reliable cleanup of applications (complete deletion of jobs)	√			
Re-schedule job in case of failure	√	√	√	?
Security				
Encryption:	★★★★			
• between the management center and the nodes	√	?	?	?
• between nodes	√	?	?	?
• between clients and the management center	√	(√)[2]	(√)[2]	(√)[2]
Hide jobs from other users	√			
Inter-application security:	★★★★	?	?	?
• applications can't interfere	√	?	?	?
• applications not harmed by another job crash	√	[3]	[3]	[3]
General aspects				
No single point of failure	★★★★	[4]	★★[5]	[6]
Platform support:	★★★	★★★★	★★★	★★★
• major UNIX and LINUX platforms	√	√	√	√
• Windows NT, 2000		√		
Scalability:	★★★★	★★	★★	★★

Table 3.1: Resource management feature comparison[1]

[1] √ = is supported, ? = not known, ★★★★= requirement is fully satisfied, ★★★= requirements is mostly satisfied, ★★= requirement is partly satisfied, ★= requirement is slightly satisfied, when the cell is empty it is not satisfied

[2] with an ssh connection to the management center and using the shell commands.

[3] The absence of a reliable cleanup of applications leads to resource spoliage by running processes of crashed jobs.

[4] The license is tied to the MAC address of a network card on the management server.

[5] If the main server is overloaded everything is transfered to the backup server. The backup server is not more powerful than the main server and therefore it will also be overloaded. The main and backup server solution only solves the hardware failure problem.

[6] If the PBS server crashes, the cluster doesn't work anymore.

Chapter 4

Scheduling

Scheduling is only needed when the demand is bigger than the offer. In the opposite case, the jobs are allocated to the slots as they come. But when there are more jobs than slots, a decision has to be taken such as which jobs will be run and which will have to wait. There are different ways to make this choice. This choice is scheduling.

4.1 Important aspects for scheduling

4.1.1 Starvation

Starvation occurs when a job is constantly delayed and therefore never runs. This can be a serious problem. In a good scheduler, there shouldn't be any possibility of starvation.

4.1.2 Average waiting time

This is the average time a job has to wait before being started. It should be as low as possible. This factor is especially important for the user, it measures the responsiveness of the cluster. It may be quite critical for small jobs, if the average waiting time is several times longer than the size of the job, the user may think he does not benefit from the power of the cluster: he might not see a speed increase.

The negative effect of the waiting time can be partially diminished by telling the user, when he submits a job, a maximum limit of the waiting time. This is not always possible and depends on the scheduling algorithm.

4.1.3 Fair scheduler

The scheduler should be fair to all users. It should not be possible to "hijack" the cluster by getting all the cluster power to the detriment of the other users, if the user is not allowed to do this by the administrator. The administrator can define in the user management a share of the cluster a user may use. If the user hasn't yet used his share, his job can have a priority boost to have more chances to be scheduled. To the contrary, if he already has used his share, his jobs priorities will be lowered to let other users have more chance having their jobs scheduled.

A fair scheduler might also give some advantage to jobs which already waited for a long time, so the waiting time is not too big.

4.1.4 Cluster usage

It represents the percentage of the computing slots used (i.e. how much of the cluster is used). It should ideally be 100%, when the scheduling queue is not empty. But due to the different sizes of jobs this is not always possible: for example in a 64 nodes cluster if a job, currently running is using 40 nodes and the only other job waiting is a 32 nodes jobs it will have to wait the end of the first job to be scheduled, and the cluster utilization will be 62.5%.

4.1.5 Which aspect is more important?

Some of the aspects described are antagonistic. For example a fair scheduler might not always maximize the cluster usage, when there are 30 free nodes and two jobs, one 20 nodes big waiting from a user which hasn't yet used his share and one 30 nodes big from a user having already exceeded his share, the fair share mechanism might schedule the smaller job first and thus the cluster will have 10 idle nodes.

The aspect which should be optimized depends on the point of view taken. The user point of view is to minimize the average waiting time, the administrator point of view is to maximize the utilization of the cluster.

Several metrics and benchmarking methods have been defined in [8]. The *cluster usage* is one of them. It can be measured with generic workload. This workload can be traces of a real workload or generated ones. Several real workloads and workload models are available in [9]. But the utilization is largely determined by the arrival process, and therefore is not a reliable metric for measuring a scheduler performance in a general case. But on a specific cluster, if the workloads tend to respect a certain model, it can be a good metric. The *average response time*, which is the time between the submission and the end of the job, is an accepted metric. But it places greater importance on long jobs. One solution is to normalize the values by using *slowdown* rather than response time. The *slowdown* is the runtime on a loaded system divided by runtime on a dedicated system (i.e. the same cluster but without any other job running on it). A value of one represents a good performance and higher values the degree of degradation. To avoid a perturbating effect caused by very short jobs, for example a 1 minute job which finished 10 hours after submission has a slowdown of 600, *bounded slowdown* is used, where a lower bound is applied to job runtimes:

$$BoundedSlowdown = \frac{FinishTime - SubmitTime}{Max(BoundTime, ActualRunningTime)}.$$

With a *lower bound* of 10 minutes, the *bounded slowdown* would only be 60 in the case of the previous example.

4.2 Scheduling algorithms

There are a lot of scheduling algorithms ranging from very simple, for example FCFS (First Come First Serve), to very complex algorithms.

4.2.1 Space and time sharing

When executing jobs on multiple processor computer or cluster two approaches are possible. Either dedicate a part of the cluster or the processors to one job exclusively, this is called *space-sharing*[1], or run several jobs parallely on each node, this is called *time-sharing*[2]. *Time-sharing* is the mechanism used on uniprocessor machines.

With *space-sharing*, some CPU cycles are "lost" when the process has to wait, for file I/O for example. But the processes belonging to the same job run approximately all at the same speed, so at a point in the program where the different processes have to wait for all the others they won't have to wait too long.

With *time-sharing*, when a process is waiting, another can be scheduled[3], thus only the cycles corresponding to the context switch are "lost". But the context switch can be extremely long if the memory is not sufficient and part of the memory must be swapped. And the speed of the different processes of a same job might run with different speeds. The *multiprogramming level*[4] characterizes between how many jobs each node is shared.

[1] the jobs share the different nodes between them
[2] the jobs on a same node share the CPU time between each other
[3] by the scheduler of the operating system running on the node
[4] the number of concurrent jobs running on a node

For both mechanism the allocation of jobs to the nodes is represented with the *Ousterhout matrix* [10] as in table 4.1 for space-sharing and table 4.2 for time-sharing. There is one cell for each processing unit P_i for space-sharing, and J_i^k represents task k of Job J_i. For time-sharing, "virtual processors" are created by adding rows (the number of rows is equal to the multiprogramming level).

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
J_1^0	J_1^1	J_1^2	J_1^3	J_2^0	J_2^1	J_3^0	J_3^1

Table 4.1: Ousterhout matrix for space-sharing

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
time-slice 0	J_1^0	J_1^1	J_1^2	J_1^3	J_1^4	J_1^5	J_1^6	J_1^7
time-slice 1	J_2^0	J_2^1	J_2^2	J_2^3	J_2^4	J_2^5	J_2^6	J_2^7
time-slice 2	J_3^0	J_3^0	J_3^1	J_3^1	J_4^0	J_4^1	J_5^0	J_5^1
time-slice 3	J_6^0	J_6^1	J_6^2	J_6^3	J_4^0	J_4^1	J_5^0	J_5^1

Table 4.2: Ousterhout matrix for time-sharing (with a multilevel programming of 4)

The matrices represent the state of the cluster at the time t. It can be extended to represent also the waiting jobs and the time at which they will be scheduled (as expected at the time t with the job duration given by the users). Such a representation is shown in figure 4.1.

4.2.2 Job terminology

Such representations have lead to qualitative job description with adjectives. Several "types" of jobs have been defined in function of the shape they have.

Small and big jobs

Big and small describe the size of the job on the processing units axis.Big jobs are jobs requiring a big share of all the nodes. Job 9 is a big job. They are also referred as wide jobs. As opposed to big jobs, small or narrow jobs use only very few nodes, like job 7 for example.

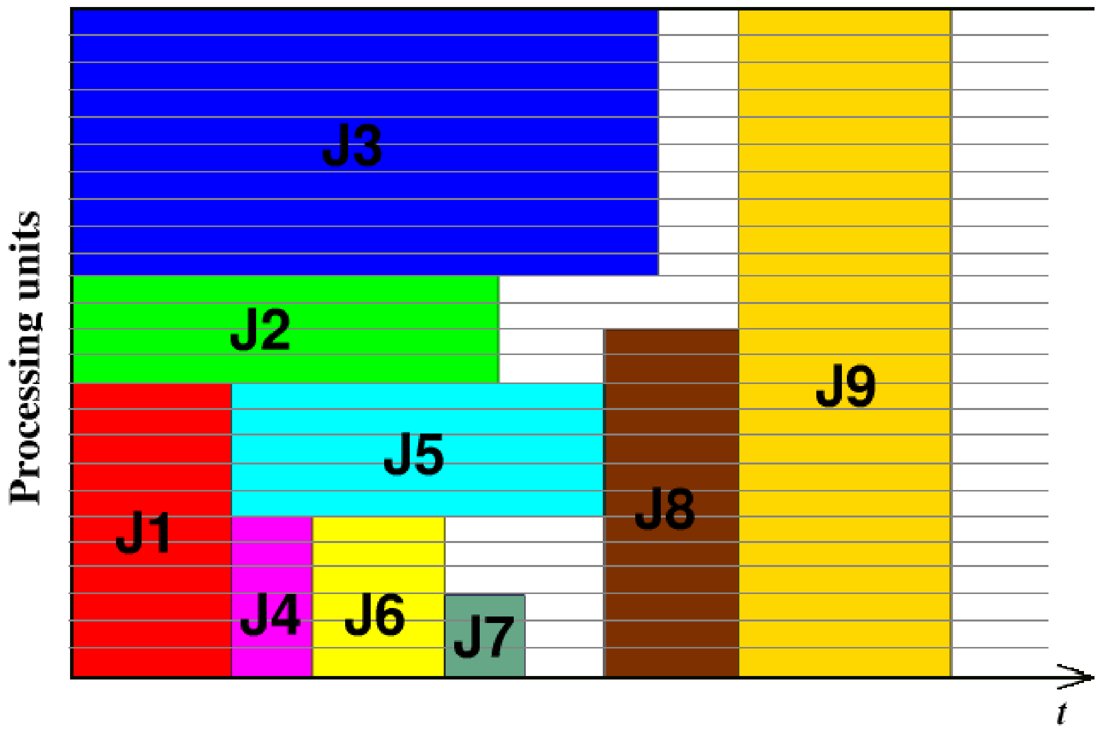

Figure 4.1: Representation of a waiting queue

Short and long jobs

Short and long describe the duration of the jobs. Job 3 is a long job and jobs 4 or 7 are short jobs.

The different types

Some job type have special characteristics. *Long narrow jobs* have the characteristic of blocking the execution of big jobs for a long period, which can lead to very bad utilization. *Big jobs* block the cluster for their duration, not allowing other jobs or only *narrow jobs*.

4.2.3 Backfilling

Backfilling is a simple technique which gives very good results [12]. It consists in filling holes at the beginning of the waiting queue with jobs which are further away in the queue. For example for the case represented in figure 4.1, lets assume that the jobs 1 to 6 have been submitted in this order and then the jobs 8, 9 and 7 (in this order). With an FCFS algorithm, the job 7 would have to wait for the job 9 to finish. The backfilling algorithm puts the job 7 before job 8 and 9 because there is a free hole in the waiting queue, and therefore running this job before the two other doesn't delay those jobs.

Backfilling improves utilization by filling the holes in the queue. It reduces the fragmentation happening with FCFS. It relies on the user estimation of the execution time of the job. These estimates are quite bad but this can be beneficial because backfilling works better if it is allowed some flexibility [12].

There are different ways to "backfill" and some of them can be combined together.

Conservative backfilling

In *conservative backfilling* [12], a job is backfilled only if in its new position in the queue it does not increase the waiting time of other jobs. This algorithm has two advantages: first, because the jobs can't be brought to wait longer due to backfilling, the user can have a limit of the waiting time for his job when he submits his job; second, the waiting time being bounded there is no starvation.

EASY backfilling

EASY backfilling [11][12] is more aggressive than conservative backfilling. A job is backfilled if it does not delay the job at the head of the queue. With this algorithm there is no execution guarantee and job starvation may occur, and it is impossible to give a job waiting time at submission. Depending on the workload, *EASY* may work better than conservative backfilling.

EASY backfilling requires less calculation than *conservative backfilling* because each backfilling candidate has to be tested only with the first job of the queue and not all.

The difference between EASY and conservative backfilling is shown in firgures 4.2 and 4.3. In this example, there are running jobs, 2 waiting jobs *J1* and *J2* and a job *J3* is submitted. In the case of EASY, the job *J3* is backfilled because it does not delay *J1* and *J2* is delayed. If after *J3*, a job of the size and shape of *J2* was submitted, EASY would lead to a better utilization. But if it was a job of the size and shape of *J3*, then conservative would give a better result.

Slack-based backfilling

In *slack-based backfilling* [13], the scheduler assigns to each waiting job some *slack*, which measures the maximal amount of time a job may be delayed

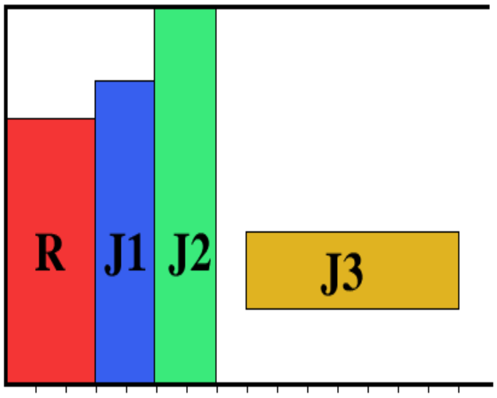

Figure 4.2: Backfilling 1: R running jobs, J1 and J2 waiting jobs, J3 newly submitted job

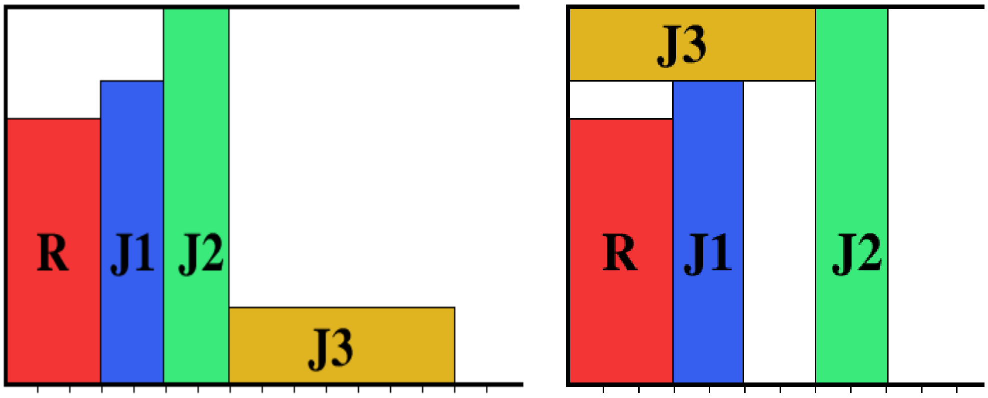

Figure 4.3: Backfilling 2: conservative (left), EASY (right)

beyond its initially assigned start time. And the scheduler allows backfilling a job if it doesn't delay any job in the queue by more than that job's slack. This way the scheduler has more flexibility than in *conservative backfilling*[5], with a guarantee that a job will be executed, because there is an upper bound for the waiting time of each job (the initial waiting time plus the slack).

Another advantage of this method is that it is possible to use priorities, which will determine the value of the slack.

[5]Conservative back-filling is a special case of slack-based backfilling where each job has a slack equal to 0.

Biggest fit first backfilling

With *biggest fit first*, the algorithm tries to backfill the big jobs first. The idea is that chances are high that small jobs will be available to fill the remaining holes after the biggest have been backfilled. According to [14], it can achieve impressive results with a very good cluster utilization (over 95%).

Shortest first backfilling

In this case, the waiting jobs are ordered by duration, and the algorithm tries to backfill the shortest jobs first. This has the advantage of decreasing the bounded slowdown, as shown in [15][16].

First fit backfilling

With *first fit* or *FCFS*, the first job, among the backfilling candidates, which fits in the free slots, is backfilled. And the process is repeated until there is no more possibilities for backfilling. One advantage of this method, is that the order of the jobs is more or less respected. The only jobs which can be overtaken are the jobs which are blocking and leaving free holes.

Best fit backfilling

In this case the algorithm tries to optimize utilization, searching for the best way to fill the holes. Other metrics may be used for the optimization like the average bounded slowdown of the whole queue. The algorithm is more complicated and demands more calculation time in case of big queues.

Speculative backfilling

Other propositions [16] have been made to improve the backfilling. They try to "correct" the duration estimation of jobs made by users. As those estimations are quite often overestimated, some jobs may not be backfilled because of this bad estimation. For the example of figures 4.2 and 4.3, if *J3* had an actual running time of 5 time units instead of the estimated 7, it could have been backfilled in a *conservative* algorithm.

Therefore after the normal backfilling phase, a speculative backfilling phase takes place: the algorithm tries to backfills the remaining jobs by giving them a *speculated running time*, for example 80% of the user estimation. When a job is backfilled its speculated time is increased to fit the hole. If a speculatevely run job runs longer than its speculated time, it is killed and returned to the queue for future scheduling. That job would be eligible for future speculative execution with speculated time equal to the average of its last speculated time and its estimated time. Resources may be spoiled for example if a new job is submitted after the speculative job is started and which would have fit in the hole. Another option is to use speculative test runs, long jobs (several hours) are run for a short period (less than 15 minutes for example), if the job finishes in this short period, the algorithm has improved slowdown and response time and in the other case it has wasted some capacity.

The improvements can be quite important: over 40% improvement for the speculative backfilling with test runs with a speculation percentage of 80%.

Conservative, *EASY* and *slack-based* backfilling are algorithm to decide if a job can be backfilled or not. *First fit*, *biggest fit*, *shortest first* and *best fit* decide which jobs the algorithm should try to backfill. *Speculative backfilling* is independent of the other methods and is only used after a first "conventional" backfilling phase has taken place.

4.2.4 Gang scheduling

Gang scheduling [17] is a special case of time-sharing. Several jobs are running parallely on the nodes sharing CPU time, but all the applications from a job are scheduled more or less at the same time. The context switch on the different nodes are coordinated. This way the job may not be delayed by one process on one node which is not scheduled for a long time, because on this node the load is abnormally high. The size of the time slots can be defined by the administrator.

With short time slots (<1s), the cluster can have a great responsiveness. A new submitted job will very quickly be scheduled. This way if the job crashes quickly after the start, the user won't have to wait too long to know it. It is also easier for interactive jobs, because interactive jobs need to be scheduled right after the submission. Gang scheduling with short time slots is very responsive with short jobs: if a long job is running for several hours and a short job (for example 15 minutes) is submitted, the short job will be finished approximately 30 minutes after submission whereas in the case of pure space sharing it would have to wait several hours for the long job to finish. But short time slots become inefficient if the context switch is too long (due to swapping).

To solve this problem two solutions are possible: the first is to increase the size of the time slots (up to several minutes), so the cost of the context switch is less important[6], but it has the disadvantage that if the time slot is

[6] if the context switch lasts 2 seconds, and the time slot is 100 seconds big, only 2% of

too big, resources can be spoiled by IO waits; the second is to allocate jobs on nodes only if the total memory usage of all the jobs is less than the size of the node memory, this way the context switches are much faster, but if all nodes are occupied, a job will have to wait before being scheduled that another jobs has finished, and the cluster lose some of its responsiveness.

With big time slots (several hours, or days) it is possible to run big very long or never ending jobs without blocking the cluster. The big job is stopped for some time, shorter jobs are scheduled, and then the big long job is resumed.

4.2.5 Calendar

With *calendar* scheduling, jobs can or cannot be scheduled in function of the time of the day or the week, for example long or big jobs can only run during night or during the week end to diminish the slowdown of small or short jobs during working hours.

4.3 The COSMOS Scheduler

This section will describe an algorithm proposition for the COSMOS scheduler. It is a priority based backfill scheduling algorithm with preemption possibilities. It's aims are to have a good utilization, to respect priorities, to avoid blocking situations with long big jobs, to prevent job starvation. The COSMOS scheduler is a space-sharing scheduler with a possibility of "gang-scheduling" with big time slices (several hours).

4.3.1 Queues

The users see one or more queues in which they may submit jobs. There are two types of queues: routing or scheduling queues. *Routing queues* redirect the jobs to other queues in function of their properties. For example such a queue might redirect jobs up to 2 hours to *queue 1*, from 2 to 10 hours to *queue 2* and from 10 hours to *queue 3*. Those queues may also redirect a job to another resource management system. *Scheduling queues* are queues were the job wait until they are scheduled. One important aspect of those queues is their *priority* which will determine a part of the job priority. Another aspect is the calendars which can be associated with those queues indicating when the queues are enabled or disabled in function of the time. These two type

---

the resources are spoiled by the context switch of queues are the one which can be created with the shell commands or the GUI.

The scheduling algorithm has internally two queues: a *schedulable queue* and a *waiting queue* which are described in 4.3.6. The scheduling takes place with the *schedulable queue* which is a global queue: the jobs of the different *scheduling queues* are at a certain time put into the *schedulable queue*

4.3.2 Job priorities

The job priority proposed here inspired by the *Maui* scheduler [18] for PBS. The priority of a job is the sum of different priorities. It is calculated as follows:

```
JobPriority = JobInitialPriority + WaitingPriorityBoost +
              QueuePriority + FairShareFactor +
              AdministratorPriorityDelta
```

where the *QueuePriority* is the priority value of the queue in which the job is, the *AdministratorPriorityDelta* is a priority modification made by the administrator and the *JobInitialPriority* is:

```
JobInitialPriority = JobInitialPriorityWeight *
                     (UserPriority + GroupPriority + JobPriorityDelta)
```

where the *UserPriority* and *GroupPriority* are defined in the User Management, and the *JobPriorityDelta* is the priority change decided by the user when submitting his job. The sum `(UserPriority + GroupPriority + JobPriorityDelta)` must be strictly positive to avoid job starvation. The *JobInitialPriorityWeight* gives the importance of the job initial priority in the total priority.

```
WaitingPriorityBoost = JobInitialPriority *
           (WaitingPriorityBoostWeight) ^ (JobWaitingTime)
```

The *WaitingPriorityBoost* reflects how much the job waited. It is the “interests” the job gets while waiting. $(WaitingPriorityBoostWeight - 1)$ is the interest rate per unit of time.

```
FairShareFactor = FareShareWeight *
                 ((UserFairShareWeight * UserFairShareDelta) +
                  (GroupFairShareWeight * GroupFairShareDelta))

UserFairShareDelta = UserShare - UserActualUsage
GroupFairShareDelta = GroupShare - GroupActualUsage
```

The *FairShareFactor* reflects the penalty or bonus to use the cluster in function of the previous usage. The *UserShare* and *GroupShare* are recalculated each time a job of the user or of the group has been scheduled. It represents the share of the cluster used over the last days/weeks/months as defined by the administrator.

4.3.3 Calendar

The idea behind calendar is to run or give a higher priority to the different type of jobs at different periods. This is done by associating calendars with queues. This calendar will determine when a hold is placed on the queue or what the queue priority value is. For example the administrator could define 3 queues: one for jobs needing less than 50% of the nodes, one for jobs bigger than 50% of the nodes lasting less than an hour and one for all the other jobs. The third queue could be held back during daytime and released during the night, and the second one could have the same priority as the third one during night, but a much lower value during day. This way long big jobs could not be scheduled during the day, short big jobs could be scheduled during day but would be penalized by the low queue priority, and could be scheduled normally during night.

4.3.4 Deadline jobs

The algorithm supports the submission of deadline jobs because they cannot be delayed past their deadline. They don't add too much overhead for the scheduling because the number of deadline jobs must be kept low[7]. The way they are treated is like if they received an infinite priority boost when they reach their deadline. For the rest they are treated as normal batch jobs and may be scheduled before the deadline.

4.3.5 Interactive jobs

Interactive jobs are a special case of deadline jobs: the deadline for start is the submission time. They can cause a big drop in the cluster utilization. First, because for them to be started immediately a bigger job may need to be held back and second because being interactive, they have to wait a certain moments for the user to input data and during this time the processors of the nodes are idle. But because the job must respond almost immediately to

[7]The deadline jobs are blocking the backfilling mechanism and lead to lower cluster utilization

user input, if another job is running on the same processor, the time slices must be small (in all cases smaller than half a second).

One solution to the problem is to have several nodes dedicated to interactive jobs during daytime. Another is to let interactive jobs stop other jobs if this is the only solution for them to be scheduled (without delaying deadline jobs over their deadline). The choice should be left to the administrator as the impact it may have depends heavily on the way the cluster is used.

4.3.6 The scheduling process

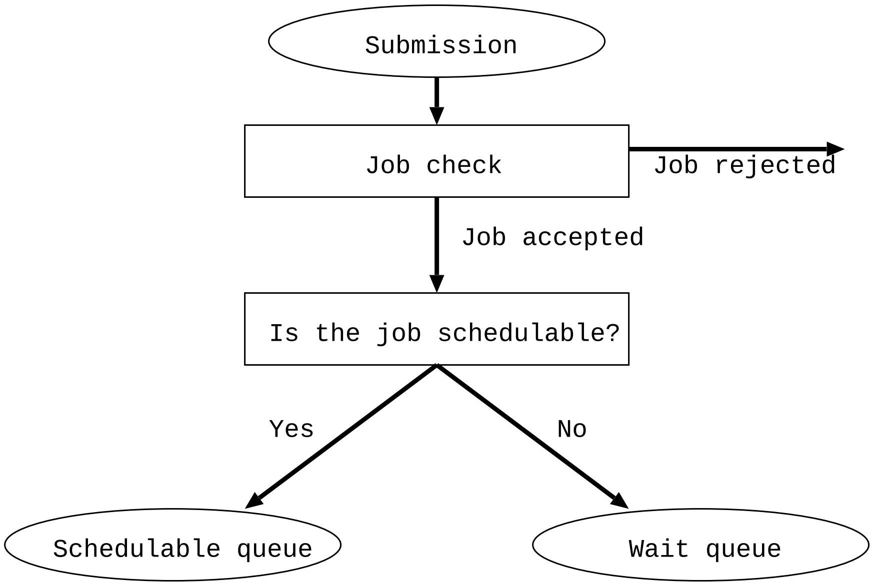

Figure 4.4: Job submission process

Before the scheduling process, the job submission process takes place. It is represented in figure 4.4.

When a job is submitted it is first checked to see if the job is accepted or rejected. A job will be rejected if the user or the group he belongs to is not allowed to submit jobs, if the resource requirements may never be met[8] or if a resource requirement exceeds the maximal value allowed by the queue in which it is submitted, or in case the submitted job is a deadline job if it is incompatible with the deadline jobs already submitted.

Then the job is either put in a wait queue or in a schedulable queue. A schedulable queue, is a queue containing jobs which are ready to be run.

[8]for example like submitting a 256 nodes job on a 128 nodes cluster.

The wait queue contains the jobs which cannot be run. If the job depends on other jobs it is put in the wait queue until the dependencies are met. If the user has specified in the submission that the job cannot be run before a certain date, it will also be put in the wait queue.

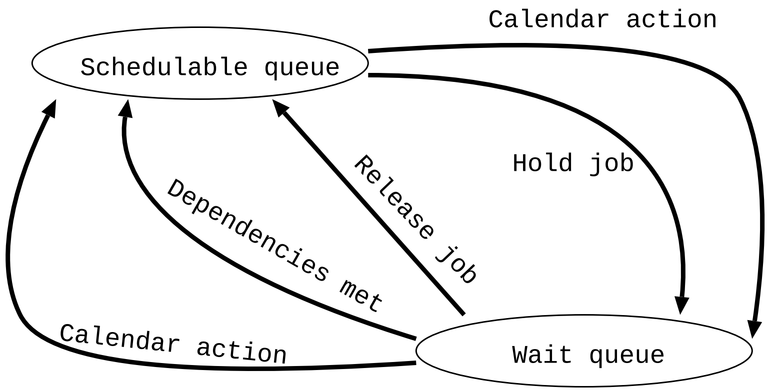

Figure 4.5: Schedulable and wait queue

The jobs can move from one queue to the other as shown in figure 4.5, on certain events. If a job is held, by the user or the administrator or if the queue in which it is is held, the job will move in the wait queue. To get back in the schedulable queue it must be released, by the user or the administrator in case of a user hold or by the administrator only in case of an administrator hold. If the job is in the wait queue because it depends on other jobs, it will be moved in the schedulable queue when these jobs have finished successfully. If the job has been submitted for an execution from a certain date, it will become schedulable at this time. The jobs can also come out of those queues if the job is deleted or if the job has dependencies in case of a crash or abnormal exit of a previous job. If the job is scheduled it is removed from the schedulable queue.

Each time resources are freed, the scheduling process shown in figure 4.6 takes place. Resources are freed when a job finishes (normally, abnormally or is killed), is held, when a queue is held or when nodes are enabled. First the priority of each job is calculated, and the schedulable queue is ordered decreasingly. Then the jobs are allocated until the algorithm has passed through all jobs and there isn't any free resource left. The backfilling is made in an *EASY* way, only jobs not delaying the first job of the queue and the deadline jobs over their deadline may be backfilled.

To avoid that long jobs block other jobs for a long period, there are several possibilities. One is to use queue for the different type of jobs in association

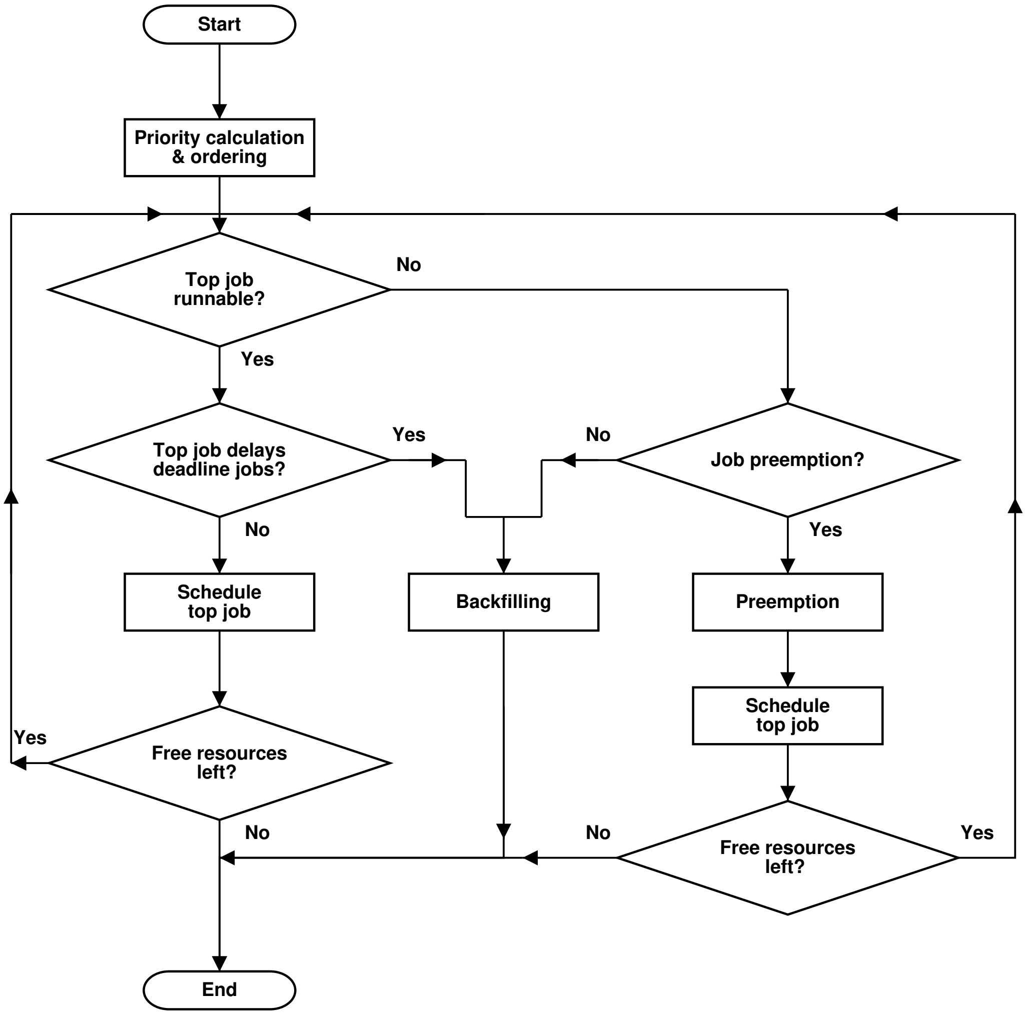

Figure 4.6: Scheduling process

with calendars (see 4.3.3). The other possibility is to let the scheduler stop temporarely jobs if their priority, at the time they were scheduled, is several times smaller[9] than the priority of the first job in the queue. The stopped jobs would get back in their schedulable queue and have to wait until their priority is high enough to be scheduled again.

In the case interactive jobs are allowed to stop other jobs, the interactive jobs bypass the normal scheduling algorithm, and the scheduler tries to stop as few jobs as possible[10]. The stopped jobs should receive a priority boost to be the first one rescheduled when their resource requirements are again met. But the priority boost should not be too big to prevent that these jobs stop other jobs due too their priority boost.

The source code for the algorithm of figure 4.6 is given in listing 4.1.

Listing 4.1: Scheduling

```
/*
FreeResources : structure representing the free resources of the cluster
schedulableJob: linked list of schedulable jobs (List<WaitingJob>)
deadlineJob: linked list of deadline jobs (List<WaitingJob>)
runningJob: linked list of running jobs (List<RunningJob>)

Member functions of the job type:
isDeadlineJob()              returns TRUE if the job is a deadline job
isRunnable(FreeResources)    returns TRUE if the job is runnable
run()                        runs the job

Member functions of the Scheduler class:
delayTopJob(Job, FreeResources)
      function returning TRUE if scheduling Job delays the top job

delayDeadlineJob(Job, FreeResources, DeadlineJobs)
      function returning TRUE if scheduling Job delays one DeadlineJob
 */

void Scheduler::Schedule()
{
  WaitingJob *CurrentWJob, *TopJob;
  RunningJob *CurrentRJob;
  bool       preemptionTest;

  /* Priority calculation */
  for (CurrentWJob = schedulableJob.begin(); CurrentWJob == schedulableJob. ¶
      end(); CurrentWJob = CurrentWJob->next())
  {
```

[9]the value can be configured by the administrator

[10]The interactive jobs should not be allowed to use too much nodes so they do not cause too much trouble.

```
    CurrentWJob->calculatePriority();
  }
  schedulableJob.sort(byPriority);

  while (TRUE)
  {
    for (CurrentWJob = schedulableJob.begin(); CurrentWJob == schedulableJob. ¶
         end(); CurrentWJob = CurrentWJob->next())
    {
      preemptionTest = FALSE;
      if (CurrentJob.isRunnable(FreeResources) == TRUE)
      {
        if (delayDeadlineJob(CurrentWJob, FreeResources) == FALSE)
        {
          SchedulableQueue.pop_front();
          if (CurrentJob.isDeadlineJob() == TRUE) deadlineJob.erase(CurrentJob ¶
               );
          if (CurrentJob.isMoldable() == TRUE) CurrentJob.expand( ¶
               FreeResources);
          CurrentJob.run();
          if (FreeResources->freeNodes() != TRUE) return;
        } else {
           TopJob = CurrentJob; /* This is the first job wich can't
                                   be scheduled immediately       */
           break;
        }
      } else {
        preemptionTest == TRUE;
        TopJob = CurrentJob;
        break;
      }
    }

    if ((preemptionTest == TRUE) && (COSMOS_Preemption == TRUE))
    {
      preemptionCandidates = PreemptionSelection(TopJob);
      if ( preemptionCandidates != NULL )
      {
        Preemption(TopJob, preemptionCandidates);
        if (FreeResources.freeNodes() != TRUE) return;
        continue;
      }
    }

    Backfilling(TopJob);
  }
}
```

The first *for* loop, in the *while(TRUE)* loop, schedules the top job until it is not possible anymore. Then the scheduler looks if it is possible to preempt a job to run the top job, and then the backfilling takes place.

4.3.7 Backfilling

The backfilling is an EASY backfilling with the addition of the check for the deadline jobs. The algorithm is represented in figure 4.7. If the jobs in the schedulable queue are in a linked list type structure and the deadline jobs also, the algorithm would be as follows (for a more detailed description of the data structures see 6.3):

If a job can be run (resource requirements met, does not delay deadline jobs or the top job) it is removed from the schedulable queue and run. The process stops when all jobs have been taken in consideration or if there aren't any free resources.

4.3.8 The preemption mechanism

The preemption mechanism is simple. The scheduler should check on a regular basis, defined by the administrator, the priority of the first job in the ordered schedulable queue and compare it to the priority value of the jobs currently running. If the ratio between the running job and the waiting job is too high, the running job is preempted. It is possible to avoid preempting jobs which execution time remainder is less than the top job execution time (or a factor of it) or less than a given time, for example less than 15 minutes because a checkpointing takes 5 minutes. The job is checkpointed and returns in the schedulable queue. It will have to wait until its priority is high enough to be scheduled. To avoid that it is immediately rescheduled, the job could receive a priority penalty or be held back for a certain time.

The listing 4.3 shows the *PreemptionSelection()* function which returns a *NULL* pointer if preemption is not possible or a linked list of the running jobs which could be preempted.

If preemption is possible, the scheduler will try to preempt the minimum of jobs. this is done in the *Preemption(WaitingJob *, List<RunningJob>)* shown in listing 4.4: the scheduler goes trough a size ordered list of jobs (growing size) until it finds a job which if stoped would let the top job run. If no running job satisfies this requirement, the biggest job is preempted and the process is repeated with the remaining preemption candidates.

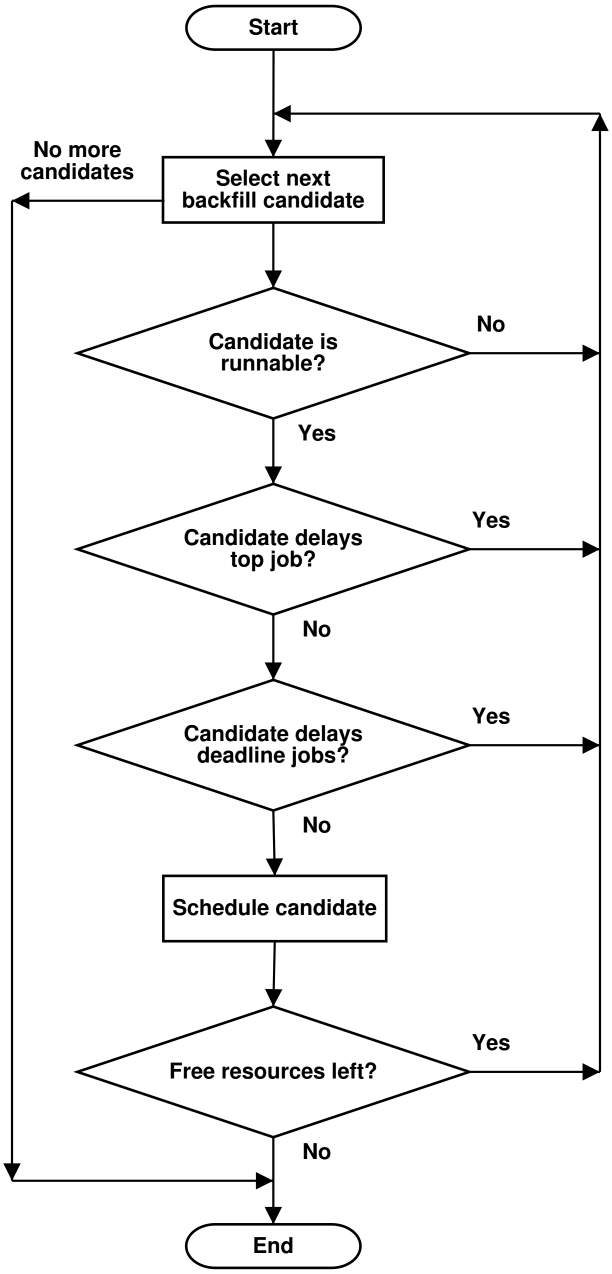

Figure 4.7: Backfilling process

Listing 4.2: Backfilling

```
void Scheduler:: Backfilling (WaitingJob *TopJob)
{
  WaitingJob *CurrentJob;
  List<WaitingJob> backfillingJob;

  backfillingJob = schedulableJob;
  backfillingJob.pop_front();
  backfillingJob.sort(sortingFunction);
  /*
    the sortingFunction is a pointer to the function used to sort
    the backfilling candidates (shortest first , biggest first , ...)
  */

  if (TopJob->next != NULL)
  {
    for( CurrentJob = backfillingJob.begin() ; CurrentJob->next !=    ¶
        backfilling.end() ; CurrentJob = CurrentJob->next() )
    {
      if (CurrentJob.isRunnable(FreeResources) != TRUE) continue;
      if (delayTopJob(CurrentJob, FreeResources) == TRUE) continue;
      if (delayDeadlineJob(Job, FreeResources, DeadlineJobs) == TRUE)    ¶
          continue;

      backfillingJob.erase(CurrentJob);
      if (CurrentJob.isMoldable() == TRUE) CurrentJob.expand(    ¶
          FreeResources);
      CurrentJob.run();
      if (FreeResources.freeNodes() != TRUE) return;
        /* No more free nodes the backfilling process is finished */
    }
}
```

Listing 4.3: Preemption selection

```
List<RuningJob> Scheduler::PreemptionSelection(WaitingJob *TopJob)
{
  double       highestPriority ;
  long         ExecutionTime, RemainderExecutionTime;
  RunningJob   *CurrentJob;
  WaitingJob   *TopJob;
  List<RunningJob> preemptionCandidates;

  possibleFreeResources = FreeResources;

  highestPriority =  TopJob->getPriority();
  ExecutionTime = TopJob->getExecutionTime();

  /*
   Selection of the preemption candidates
   */
  for (CurrentJob = runningJob.begin(); CurrentJob == runningJob.end();   ¶
       CurrentJob = CurrentJob->next())
  {
    if (CurrentJob->isDeadline() == TRUE) continue;
        // Deadline jobs are not preempted
    RemainderExecutionTime = CurrentJob->getRemainderExecTime();
    if ( highestPriority > (preemptionFactor * CurrentJob->getPriority())
        && (RemainderExecTime > (preemptionTimeFactor * ExecutionTime)
        && (RemainderExecTime > preemptionMinimalTime))
    {
      if ( TopJob->compatibleResources(CurrentJob->getUsedResources) ==   ¶
          TRUE)
      {
        possibleFreeResources += CurrentJob->getUsedResources();
        preemptionCandidates.push_back(CurrentJob);
      }
    }
  }

  if ( TopJob->isRunnable(possibleFreeResources + FreeResources) )
  {
    return(preemptionCandidates); // preemption is possible
  } else {
    return(NULL); // preemption is not possible
  }
}
```

Listing 4.4: Job Preemption

```
void Scheduler::Preemption(WaitingJob * TopJob,
                           List<RunningJob> preemptionCandidates)
{
  RunningJob *CurrentJob

  /*
   The preemption candidates are ordered by growing size (in slots)
  */
  preemptionCandidates.sort(bySize());
  preemptionCandidates.reverse();

  while (TRUE)
  {
    for( CurrentJob = preemptionCandidates.begin(); CurrentJob ==    ¶
        preemptionCandidates.end(); CurrentJob = CurrentJob->next())
    {
      if (TopJob.begin()->isRunnable(FreeResources + CurrentJob->    ¶
          getUsedResources()) == TRUE)
      {
        CurrentJob.preempt();
        if (TopJob.isMoldable() == TRUE) TopJob.expand(FreeResources)    ¶
            ;
        TopJob.run();
        return;
      }
    }
    CurrentJob.preempt();
    preemptionCandidates.pop_back();
  }
}
```

4.3.9 Starvation

No job may starve if the *WaitingPriorityBoostWeight* is greater than one. The *WaitingPriorityBoost* is the only unbounded part of the priority and grows exponentialy with the time. The priority is of the form $P(t) = P_i * r^t + c(t)$ where P_i is the initial priority and is bounded, r the interest rate and $c(t)$ is the rest of the priority and is bounded[11] and t is the time at which the job was started. There is no starvation if a job can have a higher priority than a any job started after a certain time. A minimal bound for the priority of a job is $P_{min}(t) = min(P_i) * r^t + min(c)$ and a maximal bound is $P_{max}(t) = max(P_i) * r^t + max(c)$. The conditions are:

$$\exists t_1 \in \mathbb{R}, \forall t > t_1 : P_{min}(t) > P_{max}(t - t_1) \tag{4.1}$$

$$\begin{aligned} P_{min}(t) - P_{max}(t - t_1) &= min(P_i) * r^t + min(c) - (max(P_i) * r^{t-t_1} + max(c)) \\ &= r^t * (min(P_i) - r^{-t_1} * max P_i) + min(c) - max(c) \end{aligned}$$

Lets take s_1 so that $min(P_i) - r^{-s_1} * max P_i > \frac{min(P_i)}{2}$ and s_2 so that $r^{s2} * \frac{min(P_i)}{2} > max(c) - min(c)$. Both exist because of the properties of the exponential function. If we take $t_1 = max(s_1, s_2)$ the equation 4.1 is true.

So after waiting a certain time t_1, a job is assured to have forever a higher priority than any new submitted job. And because the total execution time of the jobs submitted during the t_1 period is finite, the job is assured to come to the first place of the schedulable queue, and then cannot be delayed any further. Therefore the job won't starve.

4.3.10 Resource allocation

It has been shown in [19] that the ressource allocation of a job in a parallel heterogeneous system (wich is also know as the *Assignment Problem*) comes from the *Travelling Salesman Problem*, which is NP-complete [20]. Therefore finding an optimal allocation is complicated and require a lot of computation. Most algorithm used for solving this problem use heuristic or genetic approaches. In these methods a trade-off between solving time and solution quality has to be made. An algorithm has been proposed in [21], using the *Hungarian method* in an *Objective Oriented Algorithm*, the complexity is decreased to $O(n(E + nlog(n)))$, where $n < E < n^2$.

[11] $c(t)$ is a sum of constants, of the queue priority which different values are bounded and from the fair share which is also bounded.

Such algorithms are the only one possible in a strongly heterogeneous cluster. If the cluster is not too heterogeneous, some predefined algorithm can lead to not too bad solutions in a quick time. This is the case for allocating job efficiently on a regular network (mesh, ring, torus, hypercube, k-torus, fat-tree). In a 2-D mesh, for example, for a job of n^2 nodes, the optimal allocation, is where the nodes are in an square of n^2 nodes. The algorithm would therefore look if it can found a square of free nodes which matches the resources requirements of the job. So COSMOS, should have such algorithm for the most widely used network topologies (1-D, 2-D, 3-D mesh, ring, torus, hypercube, fat-tree and k-torus), and provide the administrator the possibility to use its own algorithm adapted to its topology.

Chapter 5

COSMOS shell commands

5.1 Job commands

5.1.1 *jacct*

Report accounting informations for a job.

`jacct [options] job_id`

The *jacct* command reports the accounting informations for the job or jobs specified by *job_id*. This command only works with jobs which are finished. These informations are the job submit options, the resource usage as reported by the application management at the end of the job and the cost of the job.

5.1.2 *jalter*

Alter a job.

`jalter [attribute modification options] job_id`

The *jalter* command modifies the attributes of the job or jobs specified by *job_id*. Only the attributes listed as options on the command will be modified. The modification can be relative to the previous value or absolute. If any of the modifications cannot be made, none will be made and an error will be returned.

5.1.3 *jdel*

Delete a job.

```
jdel [-W delay] job_id
```

The *jdel* command deletes jobs in the order in which their job identifiers are presented to the command. If the job has not been scheduled it is removed from the queue in which it is. If the job is running, the processes are killed. The server sends them a **SIGTERM** signal followed by a **SIGKILL** signal. A job may be deleted by its owner or the administrator.
The -*W* option overrides the default delay between the two signals. By specifying a delay of 0, the job won't be checkpointed before being killed.

5.1.4 *jhide*

Hide informations about a job.

```
jhide [user] job_id
```

The command *jhide* hides the job infomation for the *user* or a *user list* specified on the command line. The users might also be operators or administrators.

5.1.5 *jhold*

Hold a job.

```
jhold [hold_type] job_id
```

The *jhold* command requests that the management system place a hold on a job. A held job is not eligible for scheduling and execution. If the job is running, it is checkpointed, then the processes are killed and the job is put in the wait queue. Three type of hold exists: user, administrator and queue. A user may only place a user hold on his jobs. An administrator may place any type of hold on any job.

5.1.6 *jkill*

Kill a job or a process.

```
jkill [-W delay] job_id[@node_id]
```

The *jkill* command kills the processes of the job precised by *job_id* running on the node precised by *node_id*. The way *jkill* works is analog to *jdel*. If *@node_id* is not specified the command acts as *jdel*.

5.1.7 *jmove*

Move a job in another queue or on another node.

```
jmove [options] new_position job_id
```

The *jmove* command moves a job in another queue or on another node. A job can be moved in another queue if it respects the limits of that queue. The job or the processe of a job on a given node may be moved on another node. This is usefull for the administrator if a job is not allocated efficiently or blocks a node which could lead to a good allocation for another job.

5.1.8 *jmsg*

Append a message in a job output.

```
jmsg [options] message job_id
```

The *jmsg* command writes messages into the files of jobs. It is possible to specify if the message should be written to the standard output, standard error or the job accounting information.

5.1.9 *jredirect*

Redirect standard error and/or standard output.

```
jredirect [options] new_output job_id
```

The *jredirect* changes the output redirection of a job (standard output or error on a given node, mail, file).

5.1.10 *jrerun*

Rerun a job.

```
jrerun [options] job_id
```

The *jrerun* commands kills a job and return it to the schedulable queue.

5.1.11 *jrls*

Release a hold on a job.

```
jrls [hold_type] job_id
```

The *jrls* command removes holds which exist on a job. A user may only remove a user hold. With this command the administrator can release the queue hold of a specific job without releasing the hold on the queue.

5.1.12 *jrun*

Run a job.

```
jrun [options] job_id
```

The *jrun* command is reserved for the administrator. It can force a job to run even if its priority is not high enough.

5.1.13 *jselect*

Select job(s) matching certain criterias.

```
jselect [options] job_id
```

The *jselect* command provides a method to list the job identifier of those jobs which meet a list of selection criteria. When it succesfully completes, it will output a list of zero or more jobs which meet the criteria specified. A user can only list its jobs.

5.1.14 *jsig*

Send a signal to a job.

```
jsig [signal] job_id
```

The *jsig* comman send a signal to the job specified on the command line. The signal sent is a UNIX signal (SIGTERM, SIGKILL, SIGUSR1, SIGUSR2, ...). This command only works on running jobs.

5.1.15 *jstat*

Job status.

`jstat [options] job_executable/job_script`

The *jstat* command is used to request the status of a job or a list of jobs. The status is reported on the standard output. The options determine the information that is outputted: job submission paramters, job priority, waiting time, resource usage, ...

5.1.16 *jsub*

Submit a job.

`jsub [options] job_id`

The *jsub* command is used to submit a job to the resource management system. The options let determine the job options, the possibility to save the job options in a script file. This command can use default values for some options, that can be defined in a user defined configuration file (`.cosmos`) or through environment variables. Along with the job options, this command also sends the origin of each option (command line, configuration file, environment variable).

5.2 Queue commands

The queue commands are only accessible to the administrator, except for *qsub*.

5.2.1 *qacct*

Report accounting informations for a queue.

`qacct [options] queue_id`

The *qacct* command reports the accounting informations for the queue or all the jobs in the queue specified on the command line. When the accounting informations of all the jobs of a queue are requested, only the informations of finished jobs belonging to this queue are returned.

5.2.2 *qalter*

Alter a queue configuration.

`qalter [options] queue_id`

The *qalter* command modifies the attributes of the queue. Only the attributes listed as options on the command will be modified. The modification can be relative to the previous value or absolute. If any of the modifications cannot be made, none will be made and an error will be returned.

5.2.3 *qconf*

Create and configure a queue.

`qconf [options] queue_id`

The *qconf* command creates and configures a queue. If not specified on the command line, the queue is not enabled after creation, to prevent jobs from being submitted in this queue as it may not yet be fully configured. This command lets create either *scheduling queues* or *redirection or routing queues*. The later ones are queues where the job will be moved to a *scheduling queue* in function of its options.

5.2.4 *qdisable*

Disable a queue.

`qdisable [options] queue_id`

The *qdisable* command disables a queue. No job may be submitted in a disabled queue. It may be specified on the command line if the jobs already in the queue must be deleted, held back or stay in their current state allowing them to be scheduled as if the queue was enabled.

5.2.5 *qenable*

Enable a queue.

`qenable [options] queue_id`

The *qenable* command enables a queue. It may be specified on the command line if the jobs belonging to this queue must be released or not.

5.2.6 *qhold*

Hold a queue.

`qhold [options] queue_id`

The *qhold* command puts a queue hold on all the jobs in the queue. It may be specified on the command line if this also concerns the running jobs of the queue and therefore if those need to be checkpointed and stopped.

5.2.7 *qmon*

Queue monitor.

`qmon [options] [queue_id]`

The queue monitor returns the current state of a queue or of all queues if no queue is specified on the command line. The information returned include, among others, the status (enabled, disabled, hold), the number of jobs in the queues, the number of running jobs in the queue.

5.2.8 *qrls*

Release a queue.

`qrls [options] queue_id`

The *qrls* command removes hold which exist on a queue. This only removes the queue hold and not the user and administrator hold.

5.2.9 *qselect*

Select queue(s) matching certain criterias.

`qselect [options] queue_id`

The *qselect* command provides a method to list the queue identifier of those queues which meet a list of selection criteria. When it succesfully completes, it will output a list of zero or more queues which meet the criteria specified.

5.2.10 *qstat*

Queue status.

```
qstat [job options] queue_id
```

The *qstat* command is used to request the status of a queue or a list of queues. The status is reported on the standard output. The options determine the information which is returned: queue configuration, queue state, number of jobs, list of jobs in the queue with their priorities, calendar information related to the queue, ...

5.2.11 *qsub*

Submit a job to a queue.

```
qsub [options] queue_id
```

The *qsub* command is an alias for the *jsub* command, provided for historical reasons because it is the command commonly used by other resource management systems.

5.3 Node commands

5.3.1 *nalter*

Alter node configuration.

```
nalter [options] node_id
```

The *nalter* command modifies the attributes of the node. Only the attributes listed as options on the command will be modified. The modification can be relative to the previous value or absolute. If any of the modifications cannot be made, none will be made and an error will be returned.

5.3.2 *nconf*

Create and configure a node.

```
nconf [options] resource_list node_ip
```

The *nconf* command creates and configures a node in the resource management system. If not specified on the command line, the node is not enabled after creation, to prevent jobs from being scheduled on this node as it may not yet be fully configured. Mandatory options on the command line are: a list of resources provided by the node (architecture, number of slots, processor speed, memory size, scratch disk size, ...) special devices and the node IP address.

5.3.3 *ndisable*

Disable node.

```
ndisable [options] node_id
```

The *ndisable* command disables a queue. No job may be scheduled on a disabled node. It may be specified on the command line if the jobs already running on the node must be suspended and returned to the schedulable queue, locally checkpointed (i.e. the checkpoint is stored on the local disks and the job is resumed when the node is reenabled), killed or if the node is disabled when the jobs on the node are finished.

5.3.4 *nenable*

Enable node.

```
nenable [options] node_id
```

The *nenable* command enables a node.

5.3.5 *nreboot*

Reboot node.

```
nreboot [options] node_id
```

The *nreboot* command reboots a node.

5.3.6 *nselect*

Select node(s) matching certain criteria.

```
nselect [options]
```

The *nselect* command provides a method to list the node identifier of those nodes which meet a list of selection criteria. When it succesfully completes, it will output a list of zero or more nodes which meet the criteria specified.

5.3.7 *nshutdown*

Shutdown a node.

```
nshutdown [options] node_id
```

The *nshutdown* command shuts a node down.

5.3.8 *nstatus*

Node status.

```
nstatus [options] node_id
```

The *nstat* command is used to request the status of a node or list of nodes. The status is reported on the standard output. The options determine the information which is returned: node configuration, load, running jobs, available resources, temperature of processors, harddisks, ...

5.4 Switch commands

5.4.1 *salter*

Alter switch configuration.

```
salter [options] switch_id
```

The *salter* command modifies the attributes of the switch. Only the attributes listed as options on the command will be modified. The modification can be relative to the previous value or absolute. If any of the modifications cannot be made, none will be made and an error will be returned.

5.4.2 *sconf*

Create and configure a switch.

```
sconf [options] switch_network_address
```

The *sconf* command creates and configures a switch in the resource management system.

5.4.3 *sreboot*

Reboot switch.

```
sreboot [options] switch_id
```

The *sreboot* command reboots a switch.

5.4.4 *sselect*

Select switch(es) matching certain criteria.

```
sselect [options]
```

The *sselect* command provides a method to list the switch identifier of those switches which meet a list of selection criteria. When it succesfully completes, it will output a list of zero or more switches which meet the criteria specified.

5.4.5 *sshutdown*

Shutdown a switch.

```
sshutdown [options] switch_id
```

The *sshutdown* command shuts a switch down.

5.4.6 *sstatus*

Switch status.

```
sstatus [options] switch_id
```

The *sstat* command is used to request the status of a switch or list of switches. The status is reported on the standard output. The options determine the information which is returned: switch configuration (firmware, routing tables, . . .), resource usage (number of connections, bandwidth used, . . .), . . .

5.5 Other commands

5.5.1 *cosmos_acct*

Report accounting informations.

```
cosmos_acct [options]
```

The *cosmos_acct* command reports global accounting information about the cluster, like average utilization, average waiting time, total cost of the jobs, job workload dump, ...

5.5.2 *cosmos_useradmin*

COSMOS user administration.

```
cosmos_useradmin [options]
```

The *cosmos_useradmin* command enables the creation, deletion, modification of user and group accounts in COSMOS. The configuration information of a user or a group include, among others, its priority, its access rights, the share of the cluster he may use.

5.5.3 *cosmos_conf*

Configure COSMOS.

```
cosmos_conf
```

The *cosmos_conf* command regroups the different configuration options of COSMOS, the different management modules and the COSMOS daemons.

5.5.4 *cosmos_status*

Status of the cluster.

```
cosmos_status
```

The *cosmos_status* command reports the current status of the cluster and the different management modules.

5.5.5 *cosmos_gui*

COSMOS GUI.

```
cosmos_gui
```

The *cosmos_gui* command starts the graphical user interface for the administration of the cluster, or the job submission forms.

5.5.6 *cosmos_start*

Start COSMOS deamons.

```
cosmos_start
```

The *cosmos_start* command starts the different COSMOS daemons of the different management modules.

5.5.7 *cosmos_stop*

Stop COSMOS daemons.

```
cosmos_stop
```

The *cosmos_stop* command stops the different COSMOS daemons of the different management modules.

5.5.8 *cosmos_schedconf*

Configure COSMOS scheduler.

```
cosmos_schedconf
```

The *cosmos_schedconf* command configures the scheduler: weight of the different parts of the priority calculation, preemption factor, ...

Chapter 6

COSMOS

6.1 GUI

COSMOS will have two GUI: one for the user with job submission and monitoring options, and one for the administrator with all the options. The user GUI will therefore be a subset of the administrator GUI.

The GUI prototype has been made with Glade [22]. It uses the open source widgets GTK+. The Glade source file is included in appendix A.1.

The main window of the GUI is the *control center* shown in figure 6.1. It gives access to the different modules of COSMOS and to the configuration windows. Some key monitoring data is also displayed on the control center. The different submission, modification and creation windos would be like the *Job submission* window shown in figure 6.2. The monitoring windows would look like the *Queue monitoring* window with some overall numerical data, selectable lists with the different elements and some graphics representing the evolution of some key features over the time. A window with a representation of the global queue as the one of figure 4.1 could also be implemented.

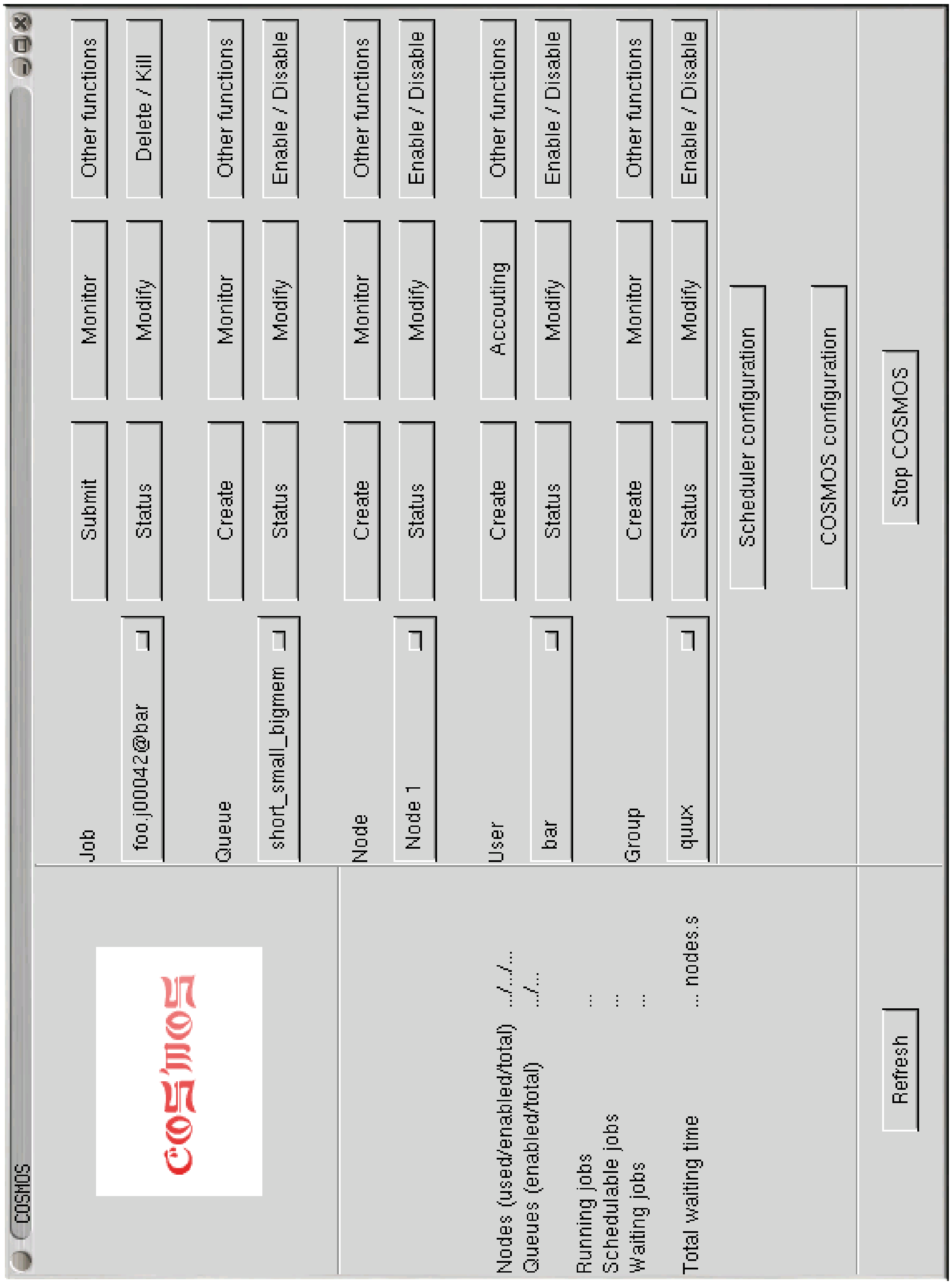

Figure 6.1: GUI: the control center

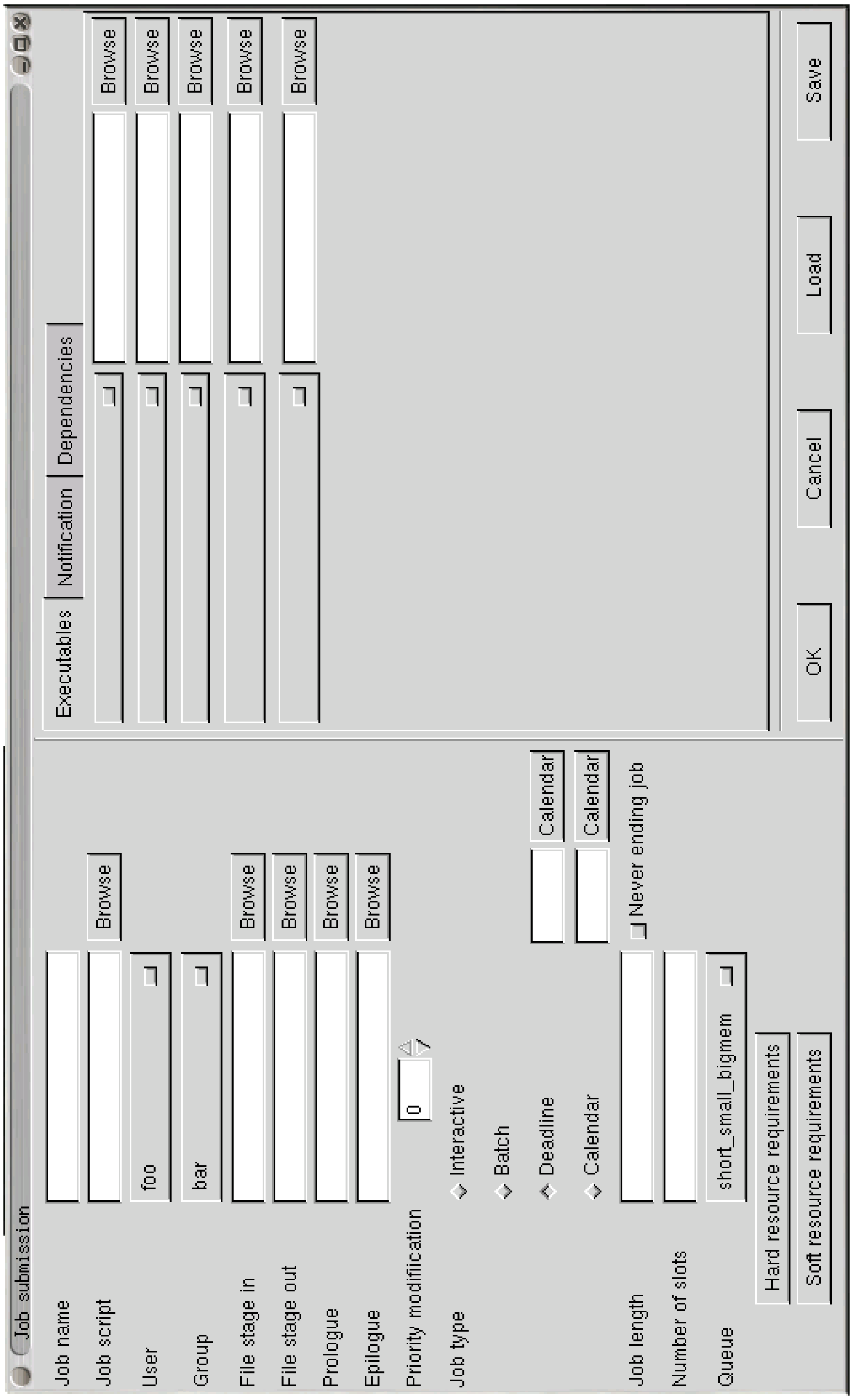

Figure 6.2: GUI: job submission

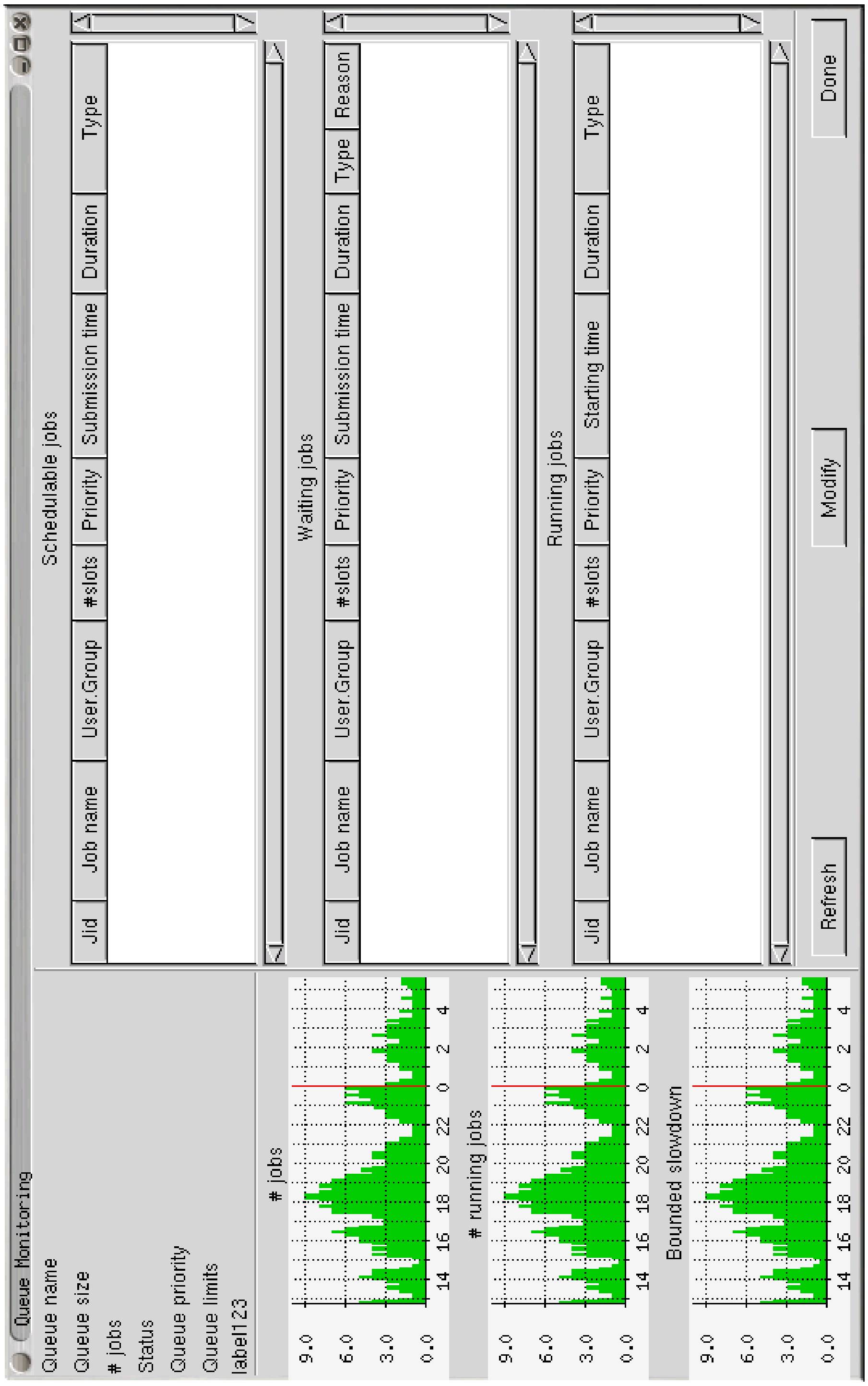

Figure 6.3: GUI: Queue monitoring

6.2 Communication between *Resource Management* and *Application Management*

The *Resource Management* and *Application Management* are two components of COSMOS which interact a lot. Almost each time, when something happen to a job, a communication takes places between these two modules. The different messages they exchange are shown in figure 6.4. When a job is

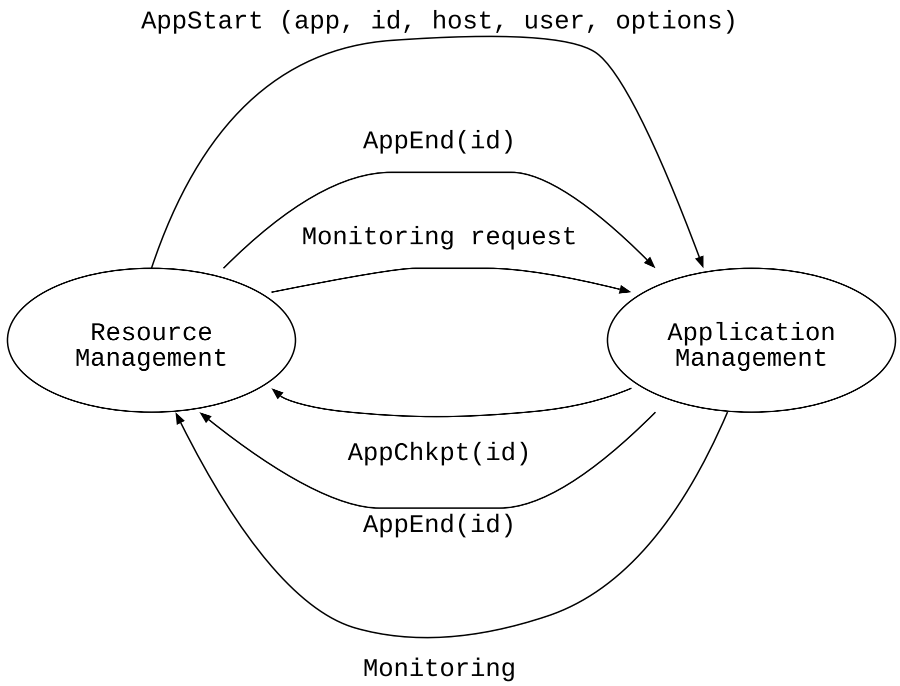

Figure 6.4: Communication between *Resource Management* and *Application Management*

started, the resource management sends a message to the application management. The message contains the script to run[1], the id of the job, the user, the host list on which the process must be spawned, the resource limits the job must respect. When the process has been started, the application management informs the resource management if it went well. If it is not the case, the errors are reported in a linked list.

[1]or the script list in case of a parallel heterogeneous job

Listing 6.1: Messages relatives to an apllication start

```
struct Executable {
  char[512]  name;      // executable with full path
  char[64]   platform;  // platform on which to run
}

struct RessourceLimit {
  char[512]  resource_name;
  double     limit;
}

struct AppStartRequest {
  ExecList   run; // list of Executable
  char[64]   jid;
  long       uid;
  char*      node_list;
  ResList     res_limits_list ; // list of RessourceLimit
  FileStage  stage_in ; // list of file names (with full path)
  FileStage  stage_out; // list of file names (with full path)
  Executable user_prolog;
  Executable admin_prolog;
  Executable user_epilog;
  Executable admin_epilog;
}

struct AppStartReply{
  int        global_status; // 0 = OK, 1 = Error
  ErrorList  error_list ;     // list of error messages (char*)
}
```

When a signal is sent to a job or a job is killed, held back, the resource management sends a message with the job id, the signal and if a checkpoint must be made.

Listing 6.2: Kill request

```
struct KillRequest {
  char[64]  jid;
  int       signal;
  int       checkpoint;
}
```

When an application ends (normally, abnormally, through an explicit user kill, because a limit was not respected), the application management reports the job end, along with the end reason, the resource usage.

Listing 6.3: Job end

```
struct JobEndReport {
  char[64]  jid;
  int       exit_code; // exit code of the job
  int       end_reason; // 0 = job finished, 1 = limit excess , ...
  ErrorList error_list ; //
  char*     std_out;
  char*     std_err;
  ResList   res_usage; // list of resource usage
}
```

The monitoring request issued by the ressource management include a job id, and a list of resources. The reply contains the job id and the monitored resources requested (list of (resource name, resource value) pairs).

When an application is checkpointed, the application management reports the time of the checkpoint along with the job id.

6.3 Data structures for the resource management

6.3.1 The resource model

Listing 6.4: The *Resource* class

```
class Resource {
  private:
    char[64] name;
    real     value;
    String   a_value;
    real     modificator; // for different processor speed for example
    bool     consumable;
    real     min_value;
    real     cost;

  public:
    constructors ...
    destructors ...
    get and set methods ..
}
```

6.3.2 The job model

The *Job* class described in listing 6.5 is an abstract class. The scheduler uses two derived classes: *WaitingJob* (see listing 6.6) and *RunningJob* (see listing 6.7). Those two derived classes differ only in their member functions and can therefore be cast in the other type.

Listing 6.5: The *Job* class

```
class Job {
  private:
    ExecList            run; // linked listof  Executable
    char[64]            jid ; // job id
    long                uid; // user id
    long                qid; // queue id
    String              node_list ;
    List<Resource>      hard; // linked list of hard resource requirements
    List<Resource>      soft ; // linked list of soft resource requirements
    List<Resource>      used; // linked list of used resources
    List<String>        stage_in ; // linked list of file names
    List<String>        stage_out ; // linked list of file names
    Executable          prolog;
    Executable          epilog;
    bool                deadline;
    bool                moldable;
    int                 status ; // run, wait, hold , ...

  public:
    constructors ...
    destructors ...
    get and set methods ...

    bool        isDeadline() ;
    bool        isMoldable();
}
```

Listing 6.6: The *WaitingJob* class

```
class WaitingJob : public Job {
  public
    double      calculatePriority () ;
    bool        isRunnable(FreeResources);
       // returns TRUE if with the resources passed in argument the
       // job could be run
    void        run();
```

```
        // send the AppStartRequest to the application management
        // changes the status of the job to running
    bool        compatibleResources(List<Resource>);
}
```

Listing 6.7: The *RunningJob* class

```
class RunningJob : public Job {
  public
    long        getRemainderExecTime();
        // returns the time the job needs to finish
    void        preempt();
}
```

6.3.3 The queue model

Listing 6.8: The *Queue* class

```
class Queue {
  private:
    int         qid;
    bool        schedulingQueue;
    int         status;  // enabled, disabled, hold
    int         size;
    int         numberOfJobs;
    List<Ressource> resourceLimits;

  public:
    constructors ...
    destructors ...
    get and set methods ..

    bool        acceptJob(Job);
         // is Job accepted. Depending on type of the queue the job is
         // submitted to another queue or the numberOfJobs is increased by ¶
              one
    void        holdQueue(bool running_jobs, bool checkpoint);
    void        releaseQueue();
    void        enable(bool release);
    void        disable(bool delete);
}
```

6.3.4 The scheduler

Listing 6.9: The *Scheduler* class

```
class Scheduler {
  private
    Resources          FreeResources;
    List<WaitingJob> schedulableJob;
    List<WaitingJob> waitingJob;
    List<RunningJob> runningJob;
    List<WaitingJob> deadlineJob;

  public
    constructors ...
    destructors ...
    get and set methods ..

    void               Schedule();
    void               Backfill (WaitingJob*);
    bool               delayDeadlineJob(WaitingJob*, Resources);
    bool               delayTopJob(WaitingJob*, Resources);
    List<RunningJob> PreemptionSelection(WaitingJob *);

}
```

Conclusion

As it has been showed in the first and thrid chapter, the user and administrator needs for supercluster are absolutely not met by the existing software. They lack specially a good security and reliability, which is not acceptable for superclusters costing several millions of dollars.

The resource management module of COSMOS will solve this problem and enable scalable heterogeneous supercluster. Its scheduler will enable a good cluster utilization with the possibilities to prioritize the jobs depending on the user, group, on the share of the cluster used by the user or group. A suggestion for the user interface has also be made.

The scheduling algorithm should now be fully implemented and tested with real and synthetic workloads to verify its quality.

Bibliography

[1] LSF, Platform Computing.
http://www.platform.com/platform/platform.nsf/webpage/LSF?OpenDocument

[2] OpenPBS, Veridian Systems. http://www.openpbs.org/

[3] PBS Pro, Veridian Systems. http://www.pbspro.com/

[4] Sun Grid Engine, Sun Microsystems. http://www.sun.com/gridware/

[5] http://www.sun.com/software/gridware/thirdparty.html

[6] Generic NQS. http://www.gnqs.org/editorial.htm

[7] Condor Project Homepage. http://www.cs.wisc.edu/condor/

[8] D. G. Feitelson and L. Rudolph, **Metrics and Benchmarking for Parallel Job Scheduling**, In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Lecture Notes Computer Science, Vol. 1459, Springer-Verlag, 1998

[9] http://www.cs.huji.ac.il/labs/parallel/workload/

[10] J. K. Ousterhout, **Scheduling Techniques for Concurrent Systems**, In *Third International Conference on Distributed Computing Systems*, 1982

[11] D. Lifka, **The ANL/IBM SP scheduling system**, In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Lecture Notes Computer Science, Vol. 949, 1995

[12] A. W. Mu'alem and D. G. Feitelson, **Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling**, In Technical Report 2000-33, 2000

[13] D. Talby and D. G. Feitelson, **Supporting priorities and improving utilization of the IBM SP2 scheduler using slack-based backfilling**, In *13th International Parallel Processing Symposium*, 1999

[14] Discussion with Jean-Michel Chenais, administrator of the SWISS-T1 and Origin2000 at the EPF Lausanne

[15] D. Zotkin and P. Keleher, **Job-Length Estimation and PErformance in Schedulers**, In *The 8^{th} High Performance Distributed Computing*, 1999

[16] D. Perković and P. Keleher, **Randomization, Speculation, and Adaptation in Batch Schedulers**, 2000

[17] D. G. Feitelson and L. Rudolph, **Parallel job scheduling: issues and approaches**, In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Lecture Notes Computer Science, Vol. 949, 1995

[18] The Maui Scheduler homepage. `http://mauischeduler.sourceforge.net/`

[19] A. Kaufmann, **Introduction à la combinatorique en vue des aplications**, Dunod, 1968

[20] M. R. Garey and D. S. Johnson, **Computer and Intractability - A guide to the Theory of NP-completeness**, Freeman NewYork, 1979

[21] Pham Hong Hanh and Valery Simonenko, **Objective-Oriented Algorithm for Job Scheduling in Parallel Heterogeneous Systems**, In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Lecture Notes Computer Science, Vol. 1291, 1997

[22] Glade, A GTK+ User Interface Builder, `http://glade.pn.org`

Appendix A

GUI source files

A.1 Glade source file

```
\HUGE{COSMOS.glade.tex != COSMOS.glade}
\\
\\

\twocolumn

<?xml version="1.0"?>
<GTK-Interface>

<project>
  <name>Appendix</name>
  <program_name>COSMOS</program_name>
  <directory></directory>
  <source_directory>src</source_directory>
  <pixmaps_directory>pixmaps</pixmaps_directory>
  <language>C</language>
  <gnome_support>False</gnome_support>
  <gettext_support>False</gettext_support>
</project>

<widget>
  <class>GtkWindow</class>
  <name>COSMOS_main</name>
  <can_focus>True</can_focus>
  <has_focus>True</has_focus>
  <events>GDK_STRUCTURE_MASK | GDK_PROPERTY_CHANGE_MASK | GDK_VISIBILITY_NOTIFY_MASK ¶
       </events>
  <signal>
    <name>destroy_event</name>
    <handler>on_COSMOS_main_destroy_event</handler>
    <last_modification_time>Wed, 03 Jan 2001 21:15:56 GMT</last_modification_time>
  </signal>
  <signal>
    <name>destroy</name>
    <handler>on_COSMOS_main_destroy</handler>
    <last_modification_time>Wed, 03 Jan 2001 21:16:41 GMT</last_modification_time>
  </signal>
  <title>COSMOS</title>
  <type>GTK_WINDOW_TOPLEVEL</type>
  <position>GTK_WIN_POS_NONE</position>
  <modal>False</modal>
  <default_width>760</default_width>
  <default_height>500</default_height>
  <allow_shrink>False</allow_shrink>
  <allow_grow>False</allow_grow>
  <auto_shrink>False</auto_shrink>

  <widget>
    <class>GtkHBox</class>
    <name>hbox1</name>
    <width>760</width>
    <homogeneous>False</homogeneous>
    <spacing>0</spacing>
```

```
<widget>
  <class>GtkHBox</class>
  <name>hbox2</name>
  <homogeneous>False</homogeneous>
  <spacing>0</spacing>
  <child>
    <padding>0</padding>
    <expand>True</expand>
    <fill>True</fill>
  </child>

  <widget>
    <class>GtkVBox</class>
    <name>vbox2</name>
    <width>245</width>
    <homogeneous>False</homogeneous>
    <spacing>0</spacing>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
      <class>GtkPixmap</class>
      <name>pixmap1</name>
      <width>260</width>
      <filename>logo.xpm</filename>
      <xalign>0.5</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <build_insensitive>True</build_insensitive>
      <child>
        <padding>0</padding>
        <expand>True</expand>
        <fill>True</fill>
      </child>
    </widget>

    <widget>
      <class>GtkHSeparator</class>
      <name>hseparator1</name>
      <child>
        <padding>0</padding>
        <expand>False</expand>
        <fill>True</fill>
      </child>
    </widget>

    <widget>
      <class>GtkTable</class>
      <name>table1</name>
      <rows>8</rows>
      <columns>2</columns>
      <homogeneous>False</homogeneous>
      <row_spacing>0</row_spacing>
      <column_spacing>0</column_spacing>
      <child>
        <padding>0</padding>
        <expand>True</expand>
        <fill>False</fill>
      </child>

      <widget>
        <class>GtkLabel</class>
        <name>label1</name>
        <width>160</width>
        <label>Nodes (used/enabled/total)</label>
        <justify>GTK_JUSTIFY_LEFT</justify>
        <wrap>False</wrap>
        <xalign>0</xalign>
        <yalign>0.5</yalign>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <child>
          <left_attach>0</left_attach>
          <right_attach>1</right_attach>
          <top_attach>0</top_attach>
          <bottom_attach>1</bottom_attach>
          <xpad>0</xpad>
          <ypad>0</ypad>
          <xexpand>False</xexpand>
```

```
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label2</name>
    <label>Queues (enabled/total)</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label3</name>
    <label>Running jobs</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>3</top_attach>
      <bottom_attach>4</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label4</name>
    <label>Schedulable jobs</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>4</top_attach>
      <bottom_attach>5</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
```

```
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label5</name>
  <label>Waiting jobs</label>
  <justify>GTK_JUSTIFY_RIGHT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>5</top_attach>
    <bottom_attach>6</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label6</name>
  <label>Total waiting time</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>7</top_attach>
    <bottom_attach>8</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label7</name>
  <label></label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>6</top_attach>
    <bottom_attach>7</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label8</name>
  <label></label>
```

```
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>2</top_attach>
      <bottom_attach>3</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label24</name>
    <width>85</width>
    <label>.../.../...</label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label25</name>
    <label>.../...</label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label26</name>
    <label>...</label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
```

```
  <ypad>0</ypad>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>3</top_attach>
    <bottom_attach>4</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label27</name>
  <label>...</label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>4</top_attach>
    <bottom_attach>5</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label28</name>
  <label>...</label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>5</top_attach>
    <bottom_attach>6</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label29</name>
  <label>... nodes.s</label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>7</top_attach>
    <bottom_attach>8</bottom_attach>
```

```
          <xpad>0</xpad>
          <ypad>0</ypad>
          <xexpand>False</xexpand>
          <yexpand>False</yexpand>
          <xshrink>False</xshrink>
          <yshrink>False</yshrink>
          <xfill>True</xfill>
          <yfill>False</yfill>
        </child>
      </widget>
    </widget>

    <widget>
      <class>GtkHSeparator</class>
      <name>hseparator5</name>
      <child>
        <padding>0</padding>
        <expand>False</expand>
        <fill>True</fill>
      </child>
    </widget>

    <widget>
      <class>GtkHButtonBox</class>
      <name>hbuttonbox11</name>
      <height>50</height>
      <layout_style>GTK_BUTTONBOX_DEFAULT_STYLE</layout_style>
      <spacing>30</spacing>
      <child_min_width>85</child_min_width>
      <child_min_height>27</child_min_height>
      <child_ipad_x>7</child_ipad_x>
      <child_ipad_y>0</child_ipad_y>
      <child>
        <padding>0</padding>
        <expand>False</expand>
        <fill>False</fill>
      </child>

      <widget>
        <class>GtkButton</class>
        <name>refresh</name>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Refresh</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>
    </widget>
  </widget>

  <widget>
    <class>GtkVSeparator</class>
    <name>vseparator1</name>
    <width>2</width>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>
  </widget>
</widget>

<widget>
  <class>GtkVBox</class>
  <name>vbox1</name>
  <homogeneous>False</homogeneous>
  <spacing>0</spacing>
  <child>
    <padding>0</padding>
    <expand>True</expand>
    <fill>True</fill>
  </child>

  <widget>
    <class>GtkVBox</class>
    <name>vbox3</name>
    <homogeneous>False</homogeneous>
    <spacing>0</spacing>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
```

```
    <class>GtkLabel</class>
    <name>label9</name>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0.5</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <padding>0</padding>
      <expand>False</expand>
      <fill>False</fill>
    </child>
  </widget>

  <widget>
    <class>GtkTable</class>
    <name>table2</name>
    <rows>2</rows>
    <columns>3</columns>
    <homogeneous>False</homogeneous>
    <row_spacing>0</row_spacing>
    <column_spacing>0</column_spacing>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
      <class>GtkLabel</class>
      <name>label19</name>
      <label>Job</label>
      <justify>GTK_JUSTIFY_LEFT</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>0</left_attach>
        <right_attach>1</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkHButtonBox</class>
      <name>hbuttonbox1</name>
      <layout_style>GTK_BUTTONBOX_START</layout_style>
      <spacing>2</spacing>
      <child_min_width>85</child_min_width>
      <child_min_height>27</child_min_height>
      <child_ipad_x>7</child_ipad_x>
      <child_ipad_y>0</child_ipad_y>
      <child>
        <left_attach>2</left_attach>
        <right_attach>3</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>True</yfill>
      </child>

      <widget>
        <class>GtkButton</class>
        <name>jsub</name>
        <width>80</width>
```

```
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <signal>
        <name>clicked</name>
        <handler>on_jsub_clicked</handler>
        <last_modification_time>Wed, 03 Jan 2001 21:32:43 GMT</last_modification_time>
      </signal>
      <label>Submit</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>jmon</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Monitor</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>jother</name>
      <width>105</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Other functions</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox2</name>
    <width>335</width>
    <layout_style>GTK_BUTTONBOX_START</layout_style>
    <spacing>2</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>True</yfill>
    </child>

    <widget>
      <class>GtkButton</class>
      <name>jstat</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label> Status </label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>jalter</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label> Modify </label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>jkill</name>
      <width>105</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label> Delete / Kill </label>
```

```
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>
    </widget>

    <widget>
      <class>GtkOptionMenu</class>
      <name>jmenu</name>
      <width>150</width>
      <can_focus>True</can_focus>
      <items>foo.j00042@bar
</items>
      <initial_choice>0</initial_choice>
      <child>
        <left_attach>0</left_attach>
        <right_attach>1</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label30</name>
      <width>2</width>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>1</left_attach>
        <right_attach>2</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label31</name>
      <width>2</width>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>1</left_attach>
        <right_attach>2</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>
  </widget>

  <widget>
```

```
    <class>GtkLabel</class>
    <name>label10</name>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0.5</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <padding>0</padding>
      <expand>False</expand>
      <fill>False</fill>
    </child>
  </widget>

  <widget>
    <class>GtkTable</class>
    <name>table3</name>
    <rows>2</rows>
    <columns>3</columns>
    <homogeneous>False</homogeneous>
    <row_spacing>0</row_spacing>
    <column_spacing>0</column_spacing>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
      <class>GtkLabel</class>
      <name>label20</name>
      <label>Queue</label>
      <justify>GTK_JUSTIFY_LEFT</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>0</left_attach>
        <right_attach>1</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkHButtonBox</class>
      <name>hbuttonbox3</name>
      <layout_style>GTK_BUTTONBOX_START</layout_style>
      <spacing>2</spacing>
      <child_min_width>85</child_min_width>
      <child_min_height>27</child_min_height>
      <child_ipad_x>7</child_ipad_x>
      <child_ipad_y>0</child_ipad_y>
      <child>
        <left_attach>2</left_attach>
        <right_attach>3</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>True</yfill>
      </child>

      <widget>
        <class>GtkButton</class>
        <name>qconf</name>
        <width>80</width>
```

```
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Create</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>qmon</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Monitor</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>qother</name>
      <width>105</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Other functions</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox4</name>
    <width>335</width>
    <layout_style>GTK_BUTTONBOX_START</layout_style>
    <spacing>2</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>True</yfill>
    </child>

    <widget>
      <class>GtkButton</class>
      <name>qstat</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label> Status </label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>qalter</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label> Modify </label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>qenable</name>
      <width>105</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Enable / Disable</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
  </widget>

  <widget>
```

```
        <class>GtkOptionMenu</class>
        <name>qmenu</name>
        <width>150</width>
        <can_focus>True</can_focus>
        <items>short_small_bigmem
</items>
        <initial_choice>0</initial_choice>
        <child>
          <left_attach>0</left_attach>
          <right_attach>1</right_attach>
          <top_attach>1</top_attach>
          <bottom_attach>2</bottom_attach>
          <xpad>0</xpad>
          <ypad>0</ypad>
          <xexpand>True</xexpand>
          <yexpand>False</yexpand>
          <xshrink>False</xshrink>
          <yshrink>False</yshrink>
          <xfill>True</xfill>
          <yfill>False</yfill>
        </child>
      </widget>

      <widget>
        <class>GtkLabel</class>
        <name>label32</name>
        <width>2</width>
        <label></label>
        <justify>GTK_JUSTIFY_CENTER</justify>
        <wrap>False</wrap>
        <xalign>0</xalign>
        <yalign>0.5</yalign>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <child>
          <left_attach>1</left_attach>
          <right_attach>2</right_attach>
          <top_attach>0</top_attach>
          <bottom_attach>1</bottom_attach>
          <xpad>0</xpad>
          <ypad>0</ypad>
          <xexpand>False</xexpand>
          <yexpand>False</yexpand>
          <xshrink>False</xshrink>
          <yshrink>False</yshrink>
          <xfill>True</xfill>
          <yfill>False</yfill>
        </child>
      </widget>

      <widget>
        <class>GtkLabel</class>
        <name>label33</name>
        <width>2</width>
        <label></label>
        <justify>GTK_JUSTIFY_CENTER</justify>
        <wrap>False</wrap>
        <xalign>0</xalign>
        <yalign>0.5</yalign>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <child>
          <left_attach>1</left_attach>
          <right_attach>2</right_attach>
          <top_attach>1</top_attach>
          <bottom_attach>2</bottom_attach>
          <xpad>0</xpad>
          <ypad>0</ypad>
          <xexpand>False</xexpand>
          <yexpand>False</yexpand>
          <xshrink>False</xshrink>
          <yshrink>False</yshrink>
          <xfill>True</xfill>
          <yfill>False</yfill>
        </child>
      </widget>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label11</name>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
```

```
  <xalign>0.5</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <padding>0</padding>
    <expand>False</expand>
    <fill>False</fill>
  </child>
</widget>

<widget>
  <class>GtkTable</class>
  <name>table4</name>
  <rows>2</rows>
  <columns>3</columns>
  <homogeneous>False</homogeneous>
  <row_spacing>0</row_spacing>
  <column_spacing>0</column_spacing>
  <child>
    <padding>0</padding>
    <expand>True</expand>
    <fill>True</fill>
  </child>

  <widget>
    <class>GtkLabel</class>
    <name>label21</name>
    <label>Node</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox5</name>
    <layout_style>GTK_BUTTONBOX_START</layout_style>
    <spacing>2</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>True</yfill>
    </child>

    <widget>
      <class>GtkButton</class>
      <name>nconf</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Create</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
```

```
      <widget>
        <class>GtkButton</class>
        <name>nmon</name>
        <width>80</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Monitor</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>

      <widget>
        <class>GtkButton</class>
        <name>nother</name>
        <width>105</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Other functions</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>
    </widget>

    <widget>
      <class>GtkHButtonBox</class>
      <name>hbuttonbox6</name>
      <layout_style>GTK_BUTTONBOX_START</layout_style>
      <spacing>2</spacing>
      <child_min_width>85</child_min_width>
      <child_min_height>27</child_min_height>
      <child_ipad_x>7</child_ipad_x>
      <child_ipad_y>0</child_ipad_y>
      <child>
        <left_attach>2</left_attach>
        <right_attach>3</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>True</yfill>
      </child>

      <widget>
        <class>GtkButton</class>
        <name>nstat</name>
        <width>80</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label> Status </label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>

      <widget>
        <class>GtkButton</class>
        <name>nalter</name>
        <width>80</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label> Modify </label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>

      <widget>
        <class>GtkButton</class>
        <name>nenable</name>
        <width>105</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Enable / Disable</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>
    </widget>

    <widget>
      <class>GtkOptionMenu</class>
      <name>nmenu</name>
      <width>150</width>
      <can_focus>True</can_focus>
      <items>Node 1
</items>
```

```
      <initial_choice>0</initial_choice>
      <child>
        <left_attach>0</left_attach>
        <right_attach>1</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label34</name>
      <width>2</width>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>1</left_attach>
        <right_attach>2</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label35</name>
      <width>2</width>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>1</left_attach>
        <right_attach>2</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label12</name>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0.5</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <padding>0</padding>
```

```
      <expand>False</expand>
      <fill>False</fill>
    </child>
  </widget>

  <widget>
    <class>GtkTable</class>
    <name>table5</name>
    <rows>2</rows>
    <columns>3</columns>
    <homogeneous>False</homogeneous>
    <row_spacing>0</row_spacing>
    <column_spacing>0</column_spacing>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
      <class>GtkLabel</class>
      <name>label22</name>
      <label>User</label>
      <justify>GTK_JUSTIFY_LEFT</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>0</left_attach>
        <right_attach>1</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkHButtonBox</class>
      <name>hbuttonbox7</name>
      <layout_style>GTK_BUTTONBOX_START</layout_style>
      <spacing>2</spacing>
      <child_min_width>85</child_min_width>
      <child_min_height>27</child_min_height>
      <child_ipad_x>7</child_ipad_x>
      <child_ipad_y>0</child_ipad_y>
      <child>
        <left_attach>2</left_attach>
        <right_attach>3</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>True</yfill>
      </child>

      <widget>
        <class>GtkButton</class>
        <name>uconf</name>
        <width>80</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Create</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>

      <widget>
        <class>GtkButton</class>
        <name>uacct</name>
        <width>80</width>
        <can_default>True</can_default>
```

```
          <can_focus>True</can_focus>
          <label>Accouting</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>

        <widget>
          <class>GtkButton</class>
          <name>button27</name>
          <width>105</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label>Other functions</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>
      </widget>

      <widget>
        <class>GtkHButtonBox</class>
        <name>hbuttonbox8</name>
        <layout_style>GTK_BUTTONBOX_START</layout_style>
        <spacing>2</spacing>
        <child_min_width>85</child_min_width>
        <child_min_height>27</child_min_height>
        <child_ipad_x>7</child_ipad_x>
        <child_ipad_y>0</child_ipad_y>
        <child>
          <left_attach>2</left_attach>
          <right_attach>3</right_attach>
          <top_attach>1</top_attach>
          <bottom_attach>2</bottom_attach>
          <xpad>0</xpad>
          <ypad>0</ypad>
          <xexpand>False</xexpand>
          <yexpand>False</yexpand>
          <xshrink>False</xshrink>
          <yshrink>False</yshrink>
          <xfill>True</xfill>
          <yfill>True</yfill>
        </child>

        <widget>
          <class>GtkButton</class>
          <name>ustat</name>
          <width>80</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label> Status </label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>

        <widget>
          <class>GtkButton</class>
          <name>ualter</name>
          <width>80</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label> Modify </label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>

        <widget>
          <class>GtkButton</class>
          <name>uenable</name>
          <width>105</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label>Enable / Disable</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>
      </widget>

      <widget>
        <class>GtkOptionMenu</class>
        <name>umenu</name>
        <width>150</width>
        <can_focus>True</can_focus>
        <items>bar
quux
</items>
        <initial_choice>0</initial_choice>
        <child>
          <left_attach>0</left_attach>
          <right_attach>1</right_attach>
          <top_attach>1</top_attach>
```

```
      <bottom_attach>2</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label36</name>
    <width>2</width>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label37</name>
    <width>2</width>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label13</name>
  <label></label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0.5</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <padding>0</padding>
    <expand>False</expand>
    <fill>False</fill>
  </child>
</widget>
```

```
<widget>
  <class>GtkTable</class>
  <name>table6</name>
  <rows>2</rows>
  <columns>3</columns>
  <homogeneous>False</homogeneous>
  <row_spacing>0</row_spacing>
  <column_spacing>0</column_spacing>
  <child>
    <padding>0</padding>
    <expand>True</expand>
    <fill>True</fill>
  </child>

  <widget>
    <class>GtkLabel</class>
    <name>label23</name>
    <label>Group</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox9</name>
    <layout_style>GTK_BUTTONBOX_START</layout_style>
    <spacing>2</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>True</yfill>
    </child>

    <widget>
      <class>GtkButton</class>
      <name>gconf</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Create</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>gmon</name>
      <width>80</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Monitor</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
```

```
      <widget>
        <class>GtkButton</class>
        <name>gother</name>
        <width>105</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Other functions</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>
    </widget>

    <widget>
      <class>GtkHButtonBox</class>
      <name>hbuttonbox10</name>
      <layout_style>GTK_BUTTONBOX_START</layout_style>
      <spacing>2</spacing>
      <child_min_width>85</child_min_width>
      <child_min_height>27</child_min_height>
      <child_ipad_x>7</child_ipad_x>
      <child_ipad_y>0</child_ipad_y>
      <child>
        <left_attach>2</left_attach>
        <right_attach>3</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>True</yfill>
      </child>

      <widget>
        <class>GtkButton</class>
        <name>gstat</name>
        <width>80</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label> Status </label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>

      <widget>
        <class>GtkButton</class>
        <name>galter</name>
        <width>80</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label> Modify </label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>

      <widget>
        <class>GtkButton</class>
        <name>genable</name>
        <width>105</width>
        <can_default>True</can_default>
        <can_focus>True</can_focus>
        <label>Enable / Disable</label>
        <relief>GTK_RELIEF_NORMAL</relief>
      </widget>
    </widget>

    <widget>
      <class>GtkOptionMenu</class>
      <name>gmenu</name>
      <width>150</width>
      <can_focus>True</can_focus>
      <items>quux
</items>
      <initial_choice>0</initial_choice>
      <child>
        <left_attach>0</left_attach>
        <right_attach>1</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
```

```
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label38</name>
      <width>2</width>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>1</left_attach>
        <right_attach>2</right_attach>
        <top_attach>0</top_attach>
        <bottom_attach>1</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkLabel</class>
      <name>label39</name>
      <width>2</width>
      <label></label>
      <justify>GTK_JUSTIFY_CENTER</justify>
      <wrap>False</wrap>
      <xalign>0</xalign>
      <yalign>0.5</yalign>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <child>
        <left_attach>1</left_attach>
        <right_attach>2</right_attach>
        <top_attach>1</top_attach>
        <bottom_attach>2</bottom_attach>
        <xpad>0</xpad>
        <ypad>0</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>True</xfill>
        <yfill>False</yfill>
      </child>
    </widget>
  </widget>

  <widget>
    <class>GtkHSeparator</class>
    <name>hseparator3</name>
    <child>
      <padding>0</padding>
      <expand>False</expand>
      <fill>False</fill>
    </child>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox12</name>
    <layout_style>GTK_BUTTONBOX_DEFAULT_STYLE</layout_style>
    <spacing>30</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
```

```
    </child>

    <widget>
      <class>GtkButton</class>
      <name>sched_conf</name>
      <width>180</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>Scheduler configuration</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label15</name>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0.5</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <padding>0</padding>
      <expand>False</expand>
      <fill>False</fill>
    </child>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox13</name>
    <layout_style>GTK_BUTTONBOX_DEFAULT_STYLE</layout_style>
    <spacing>30</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <padding>0</padding>
      <expand>False</expand>
      <fill>False</fill>
    </child>

    <widget>
      <class>GtkButton</class>
      <name>COSMOS_conf</name>
      <width>180</width>
      <can_default>True</can_default>
      <can_focus>True</can_focus>
      <label>COSMOS configuration</label>
      <relief>GTK_RELIEF_NORMAL</relief>
    </widget>
  </widget>

  <widget>
    <class>GtkHSeparator</class>
    <name>hseparator4</name>
    <child>
      <padding>0</padding>
      <expand>False</expand>
      <fill>False</fill>
    </child>
  </widget>

  <widget>
    <class>GtkHButtonBox</class>
    <name>hbuttonbox14</name>
    <height>50</height>
    <layout_style>GTK_BUTTONBOX_DEFAULT_STYLE</layout_style>
    <spacing>30</spacing>
    <child_min_width>85</child_min_width>
    <child_min_height>27</child_min_height>
    <child_ipad_x>7</child_ipad_x>
    <child_ipad_y>0</child_ipad_y>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
      <class>GtkButton</class>
```

```
            <name>stop</name>
            <can_default>True</can_default>
            <can_focus>True</can_focus>
            <label>Stop COSMOS</label>
            <relief>GTK_RELIEF_NORMAL</relief>
          </widget>
        </widget>
      </widget>
    </widget>
  </widget>
</widget>

<widget>
  <class>GtkWindow</class>
  <name>Job_submission</name>
  <title>Job submission</title>
  <type>GTK_WINDOW_TOPLEVEL</type>
  <position>GTK_WIN_POS_NONE</position>
  <modal>True</modal>
  <allow_shrink>False</allow_shrink>
  <allow_grow>True</allow_grow>
  <auto_shrink>False</auto_shrink>

  <widget>
    <class>GtkHBox</class>
    <name>hbox3</name>
    <homogeneous>False</homogeneous>
    <spacing>0</spacing>

    <widget>
      <class>GtkAlignment</class>
      <name>alignment2</name>
      <xalign>0.5</xalign>
      <yalign>0.5</yalign>
      <xscale>1</xscale>
      <yscale>1</yscale>
      <child>
        <padding>0</padding>
        <expand>True</expand>
        <fill>True</fill>
      </child>

      <widget>
        <class>GtkTable</class>
        <name>table7</name>
        <rows>20</rows>
        <columns>4</columns>
        <homogeneous>False</homogeneous>
        <row_spacing>0</row_spacing>
        <column_spacing>0</column_spacing>

        <widget>
          <class>GtkLabel</class>
          <name>label49</name>
          <label>Job type</label>
          <justify>GTK_JUSTIFY_LEFT</justify>
          <wrap>False</wrap>
          <xalign>0</xalign>
          <yalign>0.5</yalign>
          <xpad>0</xpad>
          <ypad>0</ypad>
          <child>
            <left_attach>0</left_attach>
            <right_attach>1</right_attach>
            <top_attach>11</top_attach>
            <bottom_attach>12</bottom_attach>
            <xpad>0</xpad>
            <ypad>0</ypad>
            <xexpand>False</xexpand>
            <yexpand>False</yexpand>
            <xshrink>False</xshrink>
            <yshrink>False</yshrink>
            <xfill>True</xfill>
            <yfill>False</yfill>
          </child>
        </widget>

        <widget>
          <class>GtkAlignment</class>
          <name>alignment1</name>
          <xalign>0.5</xalign>
          <yalign>0.5</yalign>
          <xscale>1</xscale>
          <yscale>1</yscale>
```

```
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>13</top_attach>
      <bottom_attach>14</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>

    <widget>
      <class>GtkRadioButton</class>
      <name>jtype_dea</name>
      <border_width>2</border_width>
      <can_focus>True</can_focus>
      <label>Deadline</label>
      <active>False</active>
      <draw_indicator>True</draw_indicator>
      <group>jtype</group>
    </widget>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jepilogue</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>8</top_attach>
      <bottom_attach>9</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jprologue</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>7</top_attach>
      <bottom_attach>8</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jstageout</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
```

```
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>6</top_attach>
      <bottom_attach>7</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jstagein</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>5</top_attach>
      <bottom_attach>6</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jscript</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>2</top_attach>
      <bottom_attach>3</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jname</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>25</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
```

```
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label48</name>
  <label>Priority modifiication</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>9</top_attach>
    <bottom_attach>10</bottom_attach>
    <xpad>1</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label47</name>
  <label>Epilogue</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>8</top_attach>
    <bottom_attach>9</bottom_attach>
    <xpad>1</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label46</name>
  <label>Prologue</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>7</top_attach>
    <bottom_attach>8</bottom_attach>
    <xpad>1</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label45</name>
```

```
  <label>File stage out</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>6</top_attach>
    <bottom_attach>7</bottom_attach>
    <xpad>1</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label44</name>
  <label>File stage in</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>5</top_attach>
    <bottom_attach>6</bottom_attach>
    <xpad>1</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label43</name>
  <label>Group</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>4</top_attach>
    <bottom_attach>5</bottom_attach>
    <xpad>1</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label42</name>
  <label>User</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
```

```
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>3</top_attach>
      <bottom_attach>4</bottom_attach>
      <xpad>1</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label41</name>
    <label>Job script</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>2</top_attach>
      <bottom_attach>3</bottom_attach>
      <xpad>1</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label40</name>
    <label>Job name</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>1</top_attach>
      <bottom_attach>2</bottom_attach>
      <xpad>1</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label51</name>
    <label>Job length</label>
    <justify>GTK_JUSTIFY_LEFT</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>15</top_attach>
      <bottom_attach>16</bottom_attach>
```

```
    <xpad>1</xpad>
    <ypad>2</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label50</name>
  <label>Number of slots</label>
  <justify>GTK_JUSTIFY_LEFT</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>0</left_attach>
    <right_attach>1</right_attach>
    <top_attach>16</top_attach>
    <bottom_attach>17</bottom_attach>
    <xpad>1</xpad>
    <ypad>2</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkButton</class>
  <name>jepilogue_browse</name>
  <width>55</width>
  <can_focus>True</can_focus>
  <label>Browse</label>
  <relief>GTK_RELIEF_NORMAL</relief>
  <child>
    <left_attach>2</left_attach>
    <right_attach>3</right_attach>
    <top_attach>8</top_attach>
    <bottom_attach>9</bottom_attach>
    <xpad>2</xpad>
    <ypad>2</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>False</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkButton</class>
  <name>jprologue_browse</name>
  <width>55</width>
  <can_focus>True</can_focus>
  <label>Browse</label>
  <relief>GTK_RELIEF_NORMAL</relief>
  <child>
    <left_attach>2</left_attach>
    <right_attach>3</right_attach>
    <top_attach>7</top_attach>
    <bottom_attach>8</bottom_attach>
    <xpad>2</xpad>
    <ypad>2</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>False</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
```

```
    <class>GtkButton</class>
    <name>jstageout_browse</name>
    <width>55</width>
    <can_focus>True</can_focus>
    <label>Browse</label>
    <relief>GTK_RELIEF_NORMAL</relief>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>6</top_attach>
      <bottom_attach>7</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkButton</class>
    <name>jstagein_browse</name>
    <width>55</width>
    <can_focus>True</can_focus>
    <label>Browse</label>
    <relief>GTK_RELIEF_NORMAL</relief>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>5</top_attach>
      <bottom_attach>6</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkButton</class>
    <name>jscript_browse</name>
    <width>55</width>
    <can_focus>True</can_focus>
    <label>Browse</label>
    <relief>GTK_RELIEF_NORMAL</relief>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>2</top_attach>
      <bottom_attach>3</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkButton</class>
    <name>cal_dead</name>
    <can_focus>True</can_focus>
    <label>Calendar</label>
    <relief>GTK_RELIEF_NORMAL</relief>
    <child>
      <left_attach>3</left_attach>
      <right_attach>4</right_attach>
      <top_attach>13</top_attach>
      <bottom_attach>14</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
```

```
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkButton</class>
    <name>cal_cal</name>
    <can_focus>True</can_focus>
    <label>Calendar</label>
    <relief>GTK_RELIEF_NORMAL</relief>
    <child>
      <left_attach>3</left_attach>
      <right_attach>4</right_attach>
      <top_attach>14</top_attach>
      <bottom_attach>15</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jlength</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>15</top_attach>
      <bottom_attach>16</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jslot</name>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>0</text_max_length>
    <text></text>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>16</top_attach>
      <bottom_attach>17</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkEntry</class>
    <name>jdeadline</name>
    <width>60</width>
    <can_focus>True</can_focus>
    <editable>True</editable>
    <text_visible>True</text_visible>
    <text_max_length>12</text_max_length>
```

```
    <text></text>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>13</top_attach>
      <bottom_attach>14</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label52</name>
    <height>5</height>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>19</top_attach>
      <bottom_attach>20</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label53</name>
    <height>5</height>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>0</left_attach>
      <right_attach>1</right_attach>
      <top_attach>0</top_attach>
      <bottom_attach>1</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label54</name>
    <height>5</height>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>1</left_attach>
```

```
    <right_attach>2</right_attach>
    <top_attach>0</top_attach>
    <bottom_attach>1</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label55</name>
  <height>5</height>
  <label></label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>2</left_attach>
    <right_attach>3</right_attach>
    <top_attach>0</top_attach>
    <bottom_attach>1</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label56</name>
  <height>5</height>
  <label></label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>3</left_attach>
    <right_attach>4</right_attach>
    <top_attach>0</top_attach>
    <bottom_attach>1</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>False</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkLabel</class>
  <name>label57</name>
  <height>5</height>
  <label></label>
  <justify>GTK_JUSTIFY_CENTER</justify>
  <wrap>False</wrap>
  <xalign>0</xalign>
  <yalign>0.5</yalign>
  <xpad>0</xpad>
  <ypad>0</ypad>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>19</top_attach>
    <bottom_attach>20</bottom_attach>
```

```
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label58</name>
    <height>5</height>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>2</left_attach>
      <right_attach>3</right_attach>
      <top_attach>19</top_attach>
      <bottom_attach>20</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkLabel</class>
    <name>label59</name>
    <height>5</height>
    <label></label>
    <justify>GTK_JUSTIFY_CENTER</justify>
    <wrap>False</wrap>
    <xalign>0</xalign>
    <yalign>0.5</yalign>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <child>
      <left_attach>3</left_attach>
      <right_attach>4</right_attach>
      <top_attach>19</top_attach>
      <bottom_attach>20</bottom_attach>
      <xpad>0</xpad>
      <ypad>0</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkCheckButton</class>
    <name>checkbutton1</name>
    <can_focus>True</can_focus>
    <signal>
      <name>clicked</name>
      <handler>gtk_true</handler>
      <last_modification_time>Wed, 03 Jan 2001 23:32:20 GMT</last_modification_time>
    </signal>
    <label>Never ending job</label>
    <active>False</active>
    <draw_indicator>True</draw_indicator>
    <child>
      <left_attach>2</left_attach>
      <right_attach>4</right_attach>
      <top_attach>15</top_attach>
      <bottom_attach>16</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
```

```
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkOptionMenu</class>
    <name>umenu</name>
    <can_focus>True</can_focus>
    <items></items>
    <initial_choice>0</initial_choice>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>3</top_attach>
      <bottom_attach>4</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkOptionMenu</class>
    <name>gmenu</name>
    <can_focus>True</can_focus>
    <items></items>
    <initial_choice>0</initial_choice>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>4</top_attach>
      <bottom_attach>5</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>True</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>True</xfill>
      <yfill>False</yfill>
    </child>
  </widget>

  <widget>
    <class>GtkSpinButton</class>
    <name>jpriority</name>
    <width>55</width>
    <can_focus>True</can_focus>
    <climb_rate>1</climb_rate>
    <digits>0</digits>
    <numeric>True</numeric>
    <update_policy>GTK_UPDATE_ALWAYS</update_policy>
    <snap>True</snap>
    <wrap>False</wrap>
    <value>0</value>
    <lower>-10</lower>
    <upper>0</upper>
    <step>1</step>
    <page>10</page>
    <page_size>10</page_size>
    <child>
      <left_attach>1</left_attach>
      <right_attach>2</right_attach>
      <top_attach>9</top_attach>
      <bottom_attach>10</bottom_attach>
      <xpad>2</xpad>
      <ypad>2</ypad>
      <xexpand>False</xexpand>
      <yexpand>False</yexpand>
      <xshrink>False</xshrink>
      <yshrink>False</yshrink>
      <xfill>False</xfill>
      <yfill>False</yfill>
    </child>
```

```
</widget>

<widget>
  <class>GtkRadioButton</class>
  <name>jtype_int</name>
  <border_width>2</border_width>
  <can_focus>True</can_focus>
  <label>Interactive</label>
  <active>False</active>
  <draw_indicator>True</draw_indicator>
  <group>jtype</group>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>11</top_attach>
    <bottom_attach>12</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>True</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkRadioButton</class>
  <name>jtype_bat</name>
  <border_width>2</border_width>
  <can_focus>True</can_focus>
  <label>Batch</label>
  <active>False</active>
  <draw_indicator>True</draw_indicator>
  <group>jtype</group>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>12</top_attach>
    <bottom_attach>13</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>True</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkRadioButton</class>
  <name>jtype_cal</name>
  <border_width>2</border_width>
  <can_focus>True</can_focus>
  <label>Calendar</label>
  <active>False</active>
  <draw_indicator>True</draw_indicator>
  <group>jtype</group>
  <child>
    <left_attach>1</left_attach>
    <right_attach>2</right_attach>
    <top_attach>14</top_attach>
    <bottom_attach>15</bottom_attach>
    <xpad>0</xpad>
    <ypad>0</ypad>
    <xexpand>True</xexpand>
    <yexpand>False</yexpand>
    <xshrink>False</xshrink>
    <yshrink>False</yshrink>
    <xfill>True</xfill>
    <yfill>False</yfill>
  </child>
</widget>

<widget>
  <class>GtkEntry</class>
  <name>jcalendar</name>
  <width>60</width>
  <can_focus>True</can_focus>
  <editable>True</editable>
  <text_visible>True</text_visible>
```

```
      <text_max_length>12</text_max_length>
      <text></text>
      <child>
        <left_attach>2</left_attach>
        <right_attach>3</right_attach>
        <top_attach>14</top_attach>
        <bottom_attach>15</bottom_attach>
        <xpad>2</xpad>
        <ypad>2</ypad>
        <xexpand>False</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>False</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>jhard</name>
      <width>170</width>
      <can_focus>True</can_focus>
      <signal>
        <name>clicked</name>
        <handler>on_jhard_clicked</handler>
        <last_modification_time>Wed, 03 Jan 2001 23:38:42 GMT</last_modification_time>
      </signal>
      <label>Hard resource requirements</label>
      <relief>GTK_RELIEF_NORMAL</relief>
      <child>
        <left_attach>0</left_attach>
        <right_attach>2</right_attach>
        <top_attach>17</top_attach>
        <bottom_attach>18</bottom_attach>
        <xpad>2</xpad>
        <ypad>2</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>False</xfill>
        <yfill>False</yfill>
      </child>
    </widget>

    <widget>
      <class>GtkButton</class>
      <name>jsoft</name>
      <width>170</width>
      <can_focus>True</can_focus>
      <signal>
        <name>clicked</name>
        <handler>on_jsoft_clicked</handler>
        <last_modification_time>Wed, 03 Jan 2001 23:39:29 GMT</last_modification_time>
      </signal>
      <label>Soft resource requirements</label>
      <relief>GTK_RELIEF_NORMAL</relief>
      <child>
        <left_attach>0</left_attach>
        <right_attach>2</right_attach>
        <top_attach>18</top_attach>
        <bottom_attach>19</bottom_attach>
        <xpad>2</xpad>
        <ypad>2</ypad>
        <xexpand>True</xexpand>
        <yexpand>False</yexpand>
        <xshrink>False</xshrink>
        <yshrink>False</yshrink>
        <xfill>False</xfill>
        <yfill>False</yfill>
      </child>
    </widget>
  </widget>
</widget>

<widget>
  <class>GtkVSeparator</class>
  <name>vseparator2</name>
  <width>10</width>
  <child>
    <padding>0</padding>
    <expand>False</expand>
    <fill>True</fill>
```

```
    </child>
  </widget>

  <widget>
    <class>GtkVBox</class>
    <name>vbox4</name>
    <homogeneous>False</homogeneous>
    <spacing>0</spacing>
    <child>
      <padding>0</padding>
      <expand>True</expand>
      <fill>True</fill>
    </child>

    <widget>
      <class>GtkNotebook</class>
      <name>notebook1</name>
      <can_focus>True</can_focus>
      <show_tabs>True</show_tabs>
      <show_border>True</show_border>
      <tab_pos>GTK_POS_TOP</tab_pos>
      <scrollable>True</scrollable>
      <tab_hborder>2</tab_hborder>
      <tab_vborder>2</tab_vborder>
      <popup_enable>False</popup_enable>
      <child>
        <padding>5</padding>
        <expand>True</expand>
        <fill>True</fill>
      </child>

      <widget>
        <class>Placeholder</class>
      </widget>

      <widget>
        <class>GtkLabel</class>
        <child_name>Notebook:tab</child_name>
        <name>label62</name>
        <label>Executables</label>
        <justify>GTK_JUSTIFY_CENTER</justify>
        <wrap>False</wrap>
        <xalign>0.5</xalign>
        <yalign>0.5</yalign>
        <xpad>2</xpad>
        <ypad>0</ypad>
      </widget>

      <widget>
        <class>Placeholder</class>
      </widget>

      <widget>
        <class>GtkLabel</class>
        <child_name>Notebook:tab</child_name>
        <name>label60</name>
        <label>Notification</label>
        <justify>GTK_JUSTIFY_LEFT</justify>
        <wrap>False</wrap>
        <xalign>0.5</xalign>
        <yalign>0.5</yalign>
        <xpad>2</xpad>
        <ypad>0</ypad>
      </widget>

      <widget>
        <class>Placeholder</class>
      </widget>

      <widget>
        <class>GtkLabel</class>
        <child_name>Notebook:tab</child_name>
        <name>label61</name>
        <label>Dependencies</label>
        <justify>GTK_JUSTIFY_LEFT</justify>
        <wrap>False</wrap>
        <xalign>0.5</xalign>
        <yalign>0.5</yalign>
        <xpad>2</xpad>
        <ypad>0</ypad>
      </widget>
    </widget>

    <widget>
```

```
        <class>GtkHSeparator</class>
        <name>hseparator6</name>
        <height>4</height>
        <child>
          <padding>0</padding>
          <expand>False</expand>
          <fill>True</fill>
        </child>
      </widget>

      <widget>
        <class>GtkHButtonBox</class>
        <name>hbuttonbox15</name>
        <layout_style>GTK_BUTTONBOX_DEFAULT_STYLE</layout_style>
        <spacing>5</spacing>
        <child_min_width>85</child_min_width>
        <child_min_height>27</child_min_height>
        <child_ipad_x>10</child_ipad_x>
        <child_ipad_y>0</child_ipad_y>
        <child>
          <padding>2</padding>
          <expand>False</expand>
          <fill>False</fill>
        </child>

        <widget>
          <class>GtkButton</class>
          <name>jsub</name>
          <width>60</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label>OK</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>

        <widget>
          <class>GtkButton</class>
          <name>jcancel</name>
          <width>60</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label>Cancel</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>

        <widget>
          <class>GtkButton</class>
          <name>jload</name>
          <width>60</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label>Load</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>

        <widget>
          <class>GtkButton</class>
          <name>jsave</name>
          <width>60</width>
          <can_default>True</can_default>
          <can_focus>True</can_focus>
          <label>Save</label>
          <relief>GTK_RELIEF_NORMAL</relief>
        </widget>
      </widget>
    </widget>
  </widget>
</widget>

</GTK-Interface>
```

A.2 C source files

The source file of *main.c* and *callbacks.c* which have been generated by Glade and then modified. The *main.c* makes the control center appear at startup, and *callback.c* handles the signal (here the clicks on the button Job submission and Queue monitoring).

A.2.1 main.c

```
/*
 * Initial main.c file generated by Glade. Edit as required.
 * Glade will not overwrite this file .
 */

#ifdef HAVE_CONFIG_H
# include <config.h>
#endif

#include <gtk/gtk.h>

#include "interface.h"
#include "support.h"

int
main (int argc, char *argv[])
{
  GtkWidget *COSMOS_main;

  gtk_set_locale () ;
  gtk_init (&argc, &argv);

  add_pixmap_directory (PACKAGE_DATA_DIR "/pixmaps");
  add_pixmap_directory (PACKAGE_SOURCE_DIR "/pixmaps");

  /*
   * The following code was added by Glade to create one of each component
   * (except popup menus), just so that you see something after building
   * the project . Delete any components that you don't want shown initially.
   */
  COSMOS_main = create_COSMOS_main ();
  gtk_widget_show (COSMOS_main);

  gtk_main ();
  return 0;
}
```

A.2.2 callback.c

```
#ifdef HAVE_CONFIG_H
# include <config.h>
#endif

#include <gtk/gtk.h>

#include "callbacks.h"
#include "interface.h"
#include "support.h"


gboolean
on_COSMOS_main_destroy_event          (GtkWidget       *widget,
                                        GdkEvent        *event,
                                        gpointer         user_data)
{

  return FALSE;
}


void
on_COSMOS_main_destroy                (GtkObject       *object,
                                        gpointer         user_data)
{
   exit(0);
}


void
on_jsub_clicked                       (GtkButton       *button,
                                        gpointer         user_data)
{
  GtkWidget *Job_submission;

  Job_submission = create_Job_submission ();
  gtk_widget_show (Job_submission);

}

void
on_qmon_clicked                       (GtkButton       *button,
                                        gpointer         user_data)
{
  GtkWidget *Queue_mon;

  Queue_mon = create_Queue_mon();
  gtk_widget_show (Queue_mon);
}
```