An Advance Reservation-Based Computation Resource Manager for Global Scheduling

1. National Institute of Advanced Industrial Science and Technology, 2 Suuri Giken

Hidemoto Nakada¹, Atsuko Takefusa¹, Katsuhiko Ookubo¹,², Tomohiro Kudoh¹

Yoshio Tanaka¹, Satoshi Sekiguchi¹
Large scale computation with Grid technology
- Resources are spanning on several sites
- Co-allocation of multiple resources is essential

Most sites employ batch queuing systems
- FCFS (First Comes First Served) + Priority
- Not suitable for co-allocation

![Diagram of Super Scheduler and Local Schedulers at SiteA, SiteB, SiteC]
Co-allocation of Computational Resources (1/2)

FCFS

FIFO scheduling

Jobs submitted at the same time not necessarily starts at the same time

SiteA

SiteB

SiteC
Co-allocation with Advance Reservation

One of the most easy way to co-allocate resources
- Specify a time slot and make reservations on all the resource in advance.
- Historically done by phone, fax, or e-mail to the site administrator.

![Diagram showing Super Scheduler connecting to Local Schedulers for SiteA, SiteB, and SiteC with Advance Reservation arrows.]

Super Scheduler

Advance Reservation

Local Scheduler
SiteA

Local Scheduler
SiteB

Local Scheduler
SiteC
Advance Reservation

Allocate time slot, independent of the queue
Key technologies for resource co-allocation

Super Scheduler

Commit Protocol

Co-allocation

Advance Reservation

Local Scheduler

SiteA

Local Scheduler

SiteB

Local Scheduler

SiteC

JSSPP '07
June 17th, Monterey

PluS Advance Reservation Mgr.
Contribution

Design and Implementation of Advance Reservation Manager PluS

- Plug-in module for existing queuing systems to enable advance reservation
- Propose two implementation methods
  - Scheduler Replacement Method
  - Queue Control Method
- Compare two methods
  - Queue Control Method is easy to implement
  - Overhead is substantial but acceptable
Overview of the talk

- **Design of Advance Reservation Manager PluS**
  - Generic configuration of queuing systems
  - Proposal of two methods
    - Scheduling Module Replacement Method
    - Queue Control Method

- **Evaluation: comparison of the two methods**
  - based on lines of codes
  - based on execution time

- **Conclusion**
What are Queuing Systems?

- Manages job execution on computational resources
  - Running job exclusively occupy the resource
    - c.f. time share
  - Manages accounting information
  - Most site uses some kind of this

Commercial implementations
- LSF, NQS, PBS Professional, LoadLeveler

Open source implementations
- TORQUE – based on OpenPBS, Cluster Resources Inc.
- Grid Engine – Sun Microsystems.
Typical Configuration of Queuing Systems

submit

Head Node

Master Server

Scheduling Module

Comp. Server

Comp. Node

Comp. Server

Comp. Node

Comp. Server

Comp. Node
Problem

- Open Source Queuing Systems typically do not support advance reservation capability.
- Commercial ones support it, but...
  - No chance to change the reservation policy.
  - Not suitable for research testbed.
How can we add Advance Reservation capability to existing queuing system?

- **Modify the Scheduling Module**
  - Requires deep understanding of the code. It is not easy even if the source is open.

- **Replace Scheduling Module**
  - Rather easy, if the communication protocol between Master Server is simple

- **Keep Scheduling Module as is and put some module outside**
  - Controls Queue from outside of the system
  - Not always possible depending on the queuing system capability

Controls Queue from outside of the system.
How can we add Advance Reservation capability to existing queuing system?

- Modify the Scheduling Module
  - Requires deep understanding of the code. It is not easy even if the source is open.
- Replace Scheduling Module
  - Rather easy, if the communication protocol between Master Server is simple
- Keep Scheduling Module as is and put some module outside
  - Controls Queue from outside of the system
  - Not always possible depending on the queuing system capability
How can we add Advance Reservation capability to existing queuing system?

- **Modify the Scheduling Module**
  - Requires deep understanding of the code. It is not easy even if the source is open.

- **Replace Scheduling Module**
  - Rather easy, if the communication protocol between Master Server is simple

- **Keep Scheduling Module as is and put some module outside**
  - Controls Queue from out side of the system
  - Not always possible depending on the queuing system capability
Summary of the two methods

Scheduling Module Replacement Method

- ‘Brain transplant’ – You can do anything you want
- You might have to re-implement all the capability of the existing scheduling module, if needed

Queue Control Method

- Not always possible, depending on capability of the target queuing system
  - ex. TORQUE
- Overhead might become an issue.
- Implementation cost will be relatively small
Implementation details of PluS Reservation Manager

- Implemented in Java
  - uses db4object as database backend
- Command line commands are implemented with shell script + Java
- Commands and the PluS module communicate with Java RMI

PluS Reservation Commands

RMI

PluS Reservation Manager
Reservation Related Commands

- **plus_reserve**
  - Requests for a reservation
  - In: start/end time, # of Nodes
  - Out: Reservation ID

- **plus_cancel**
  - Cancel a reservation
  - In: Reservation ID

- **plus_status**
  - Query status of the reservation
  - In: Reservation ID
  - Out: Status of the reservation

- **plus_modify**
  - Modify the reservation
  - In: Reservation ID, start/end time, # of Nodes
Reservation Usage Scenario

[Checklist]

Make a reservation
> plus_reserve -s 12:00 -e 14:00 -n 1
Reserve succeeded: reservation id is 14

Confirm the reservation with the reservation ID
> plus_status
id owner start end duration state
R14 nakada Feb 20 12:00 Feb 20 14:00 2h00m Confirmed

Submit a job with the reservation ID
> qsub -q R14 script
Scheduling Module Replacement

Reservation Request

PluS Scheduling Module w/ Rsv. Mgr.

Master Server

Comp. Server

Comp. Server

Comp. Server

Comp. Server
Advance Reservation with Queue Control

What are queues?
- Abstract ‘submit point’ for jobs
- Can be allocated for specific group of users
- Can be allocated for specific set of nodes

Advance Reservation by Queue Control
- Create Advance Reservation as a queue
- Activate the queue for specific time of period

Key Characteristics of the Method
- (Relatively) Easy to implement
- No need to understand internal protocol of the target system - means easy to catch up updates.
- Requires multiple invocations of command to control queues - overhead
Advance Reservation by Queue Control

Reservation Request

PluS Res. Mgr.

Scheduler

Master Server

Comp. Server

Comp. Server

Comp. Server

Comp. Server
Evaluation

- Easiness of implementation
  - Is the Queue Control Method really easier to implement?
  - Compare two methods with lines of codes

- Execution Overhead
  - How heavy is the Queue Control?
    - It might affect the response time of the upper layer modules
  - Compare execution time for reservation / cancellation
Lines of Code

- Common
- TORQUE specific
- GE Replace
- GE Queue Ctrl.

The diagram displays the distribution of Lines of Code across different categories: TORQUE Replace, GE Replace, and GE Queue Ctrl. The vertical axis represents the number of lines of code, ranging from 0 to 14,000, with specific values at multiples of 2000. The horizontal axis lists the categories, each with a bar chart showing the corresponding lines of code.
Note on the result

The replacing scheduling modules are not fully implementing the capability of the original TORQUE/Grid Engine scheduling module.

To fully implement them, it requires much more lines.
Lines of Code *if* we fully implement the existing capability

![Bar chart showing lines of code for TORQUE Replace, GE Replace, and GE Queue Ctrl.](chart.png)
## Comparison with Command Execution Time

### Experimental Environment
- Pentium III 1.4 GHz dual CPU
- Memory 2G byte
- Linux RedHat 8

### Measurement
- using ‘time’ command to measure the execution time of the plus commands
- 10 time trial. The average was calculated excluding the max. and min. values.

### Table: Execution Time

<table>
<thead>
<tr>
<th>Action</th>
<th>Make Reservation</th>
<th>Cancel Reservation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling module replacement</td>
<td>1.02</td>
<td>0.04</td>
</tr>
<tr>
<td>Queue Control</td>
<td>1.95</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Queue Control is slower.
- ‘qconf’ overhead

Execution time is 1 - 2 sec.
acceptable
Related work

Maui

- Freely available from Cluster Resources Inc.
- Replaces TORQUE Scheduling module

Catalina [Yoshimoto 05]

- SDSC (San Diego Supercomputer Center)
- Implemented in Python
- Replaces TORQUE Scheduling module
- All the jobs are scheduled with reservation
Conclusion

Proposed PluS, an Advance Reservation Manager

- Proposed two implementation methods
  - Scheduler replacement method
  - Queue control method
- Implemented for TORQUE and Grid Engine

Evaluated two methods

- Scheduler replacement is faster but more difficult to implement
- Queue control is slower but the overhead is acceptable
Current Status

- Administrators settable Advance Reservation Policy with Policy Description Language
  - Previous implementation:
    - Always prioritize jobs with Advance Reservation
    - Not suitable for production system.
  - Now it allows administrators to define ‘policy’ on acceptance of advance reservation request
    - Condor ClassAd as a policy language

Available from http://www.g-lambda.net/plus
Future Work

Application to other queuing systems

The queue control method will be easily applicable to other queuing systems, in theory.

Confirm this through porting PluS to other queuing systems

- LoadLeveler
- Condor
Acknowledgement

This work is partly funded by the Science and Technology Promotion Program’s “Optical Paths Network Provisioning based on Grid Technologies” of MEXT, Japan.

http://www.g-lambda.net/plus
PluS Implementation

3 implementations

- Scheduling module Replacement for TORQUE
- Scheduling module Replacement for Grid Engine
- Queue Control for Grid Engine
Scheduling Module Replace Method

submit — Head Node — reserve

Master Server

Scheduling Module

Comp. Server
Comp. Node

Comp. Server
Comp. Node

Comp. Server
Comp. Node
Queue Control Method

submit → Master Server → Scheduling Module → Comp. Server

reserve → PluS Rsv. Mgr.

qconf → Head Node

Comp. Node

Comp. Server

Comp. Node

Comp. Server

Comp. Node