An Advance Reservation-Based Computation Resource Manager for Global Scheduling

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

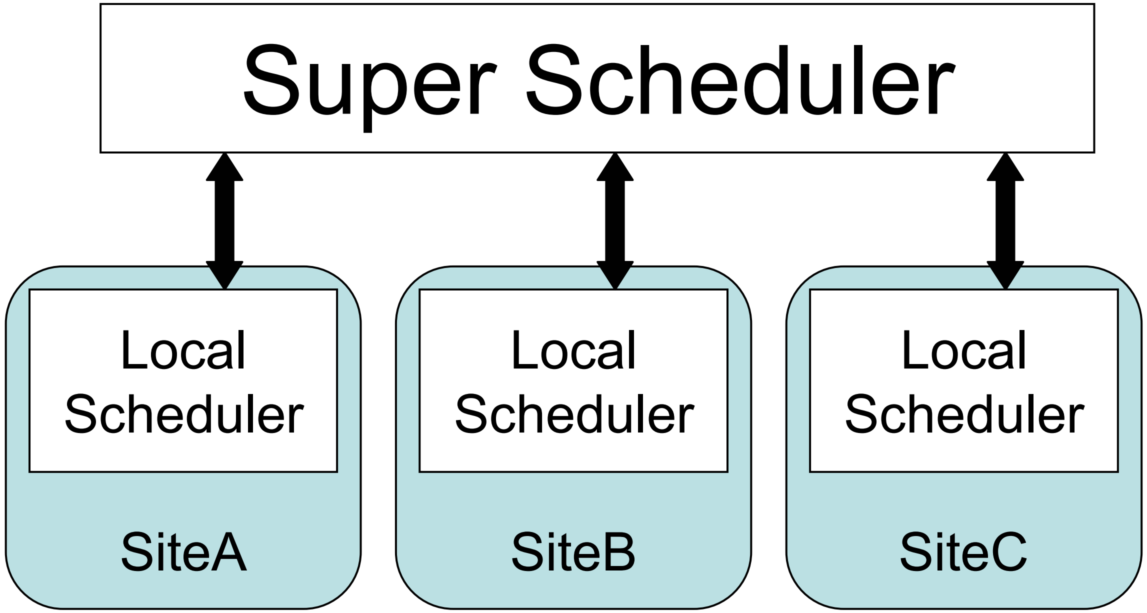
Hidemoto Nakada[1], Atsuko Takefusa[1],

Katsuhiko Ookubo[1,2],Tomohiro Kudoh[1]

Yoshio Tanaka[1], Satoshi Sekiguchi[1]

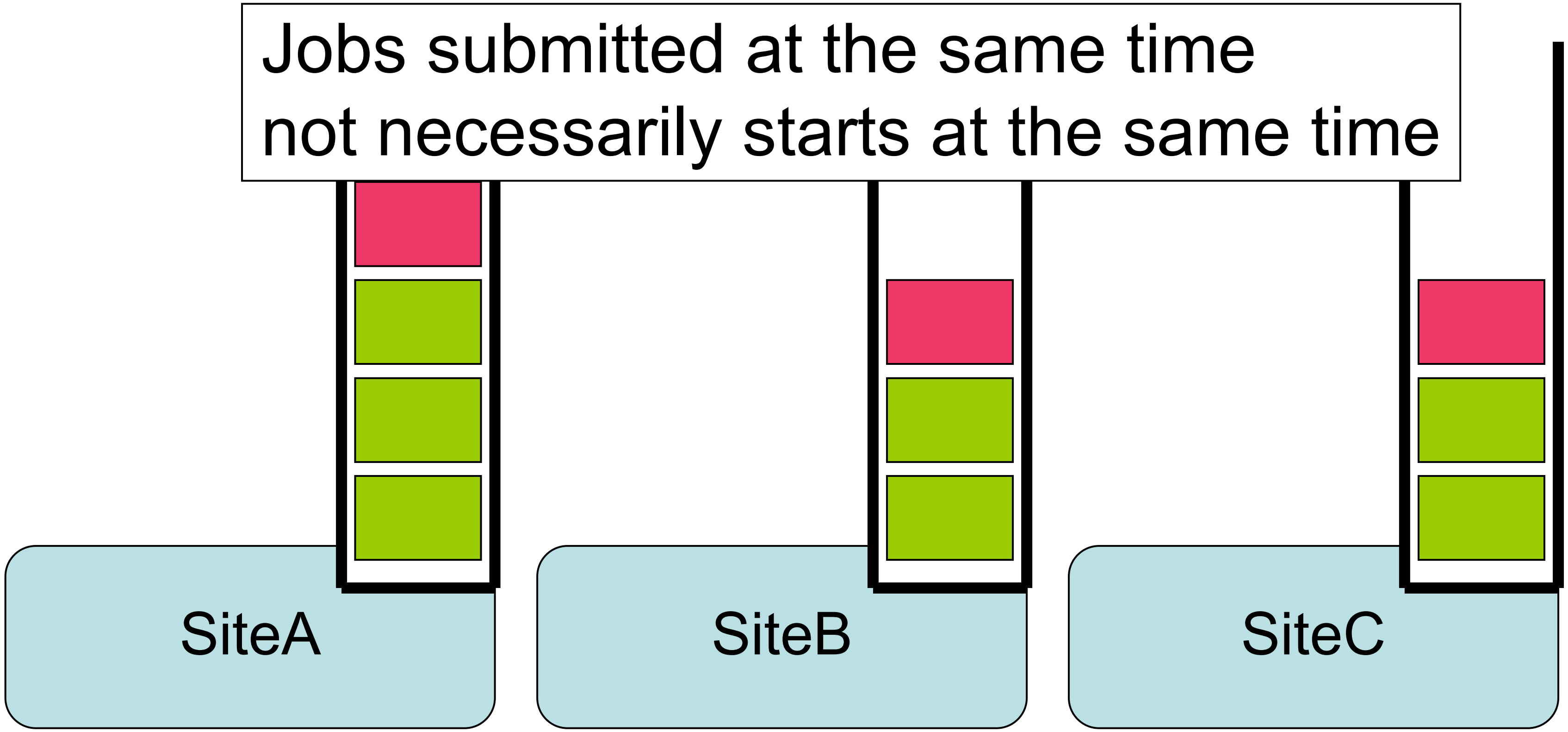
Background

- Large scale computation with Grid technology
  - Resources are spanning on several sites
  - Co-allocation of multiple resources is essential
- Most sites employs batch queuing systems
  - FCFS (First Comes First Served) + Priority
  - Not suitable for co-allocation



Co-allocation of Computational Resources (1/2)

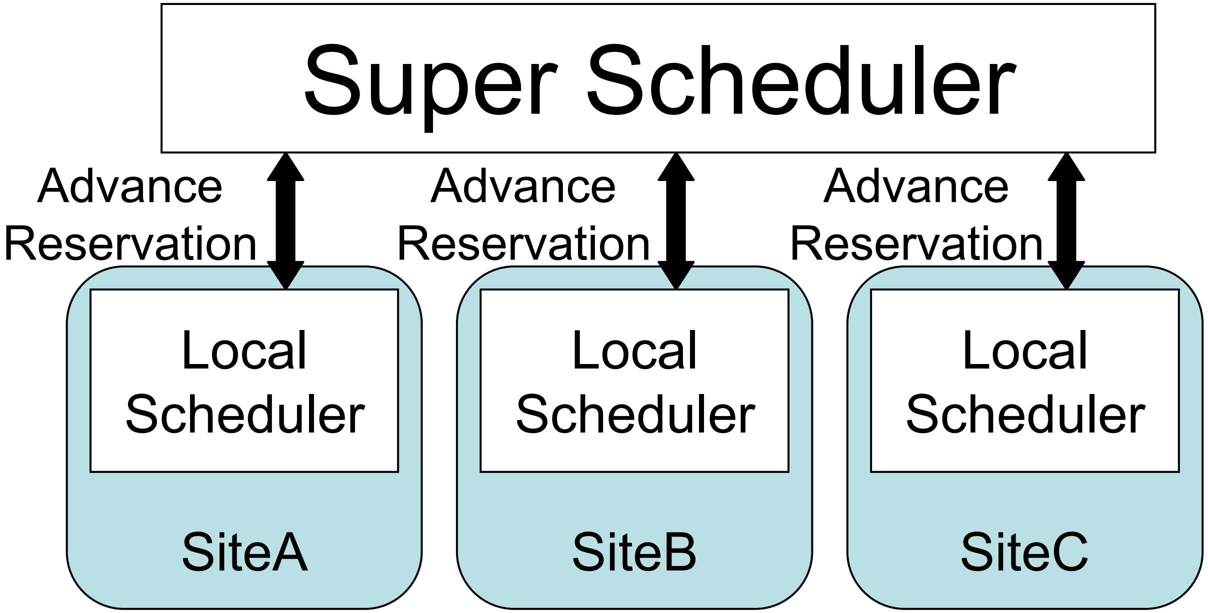
- FCFS
  - FIFO scheduling



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

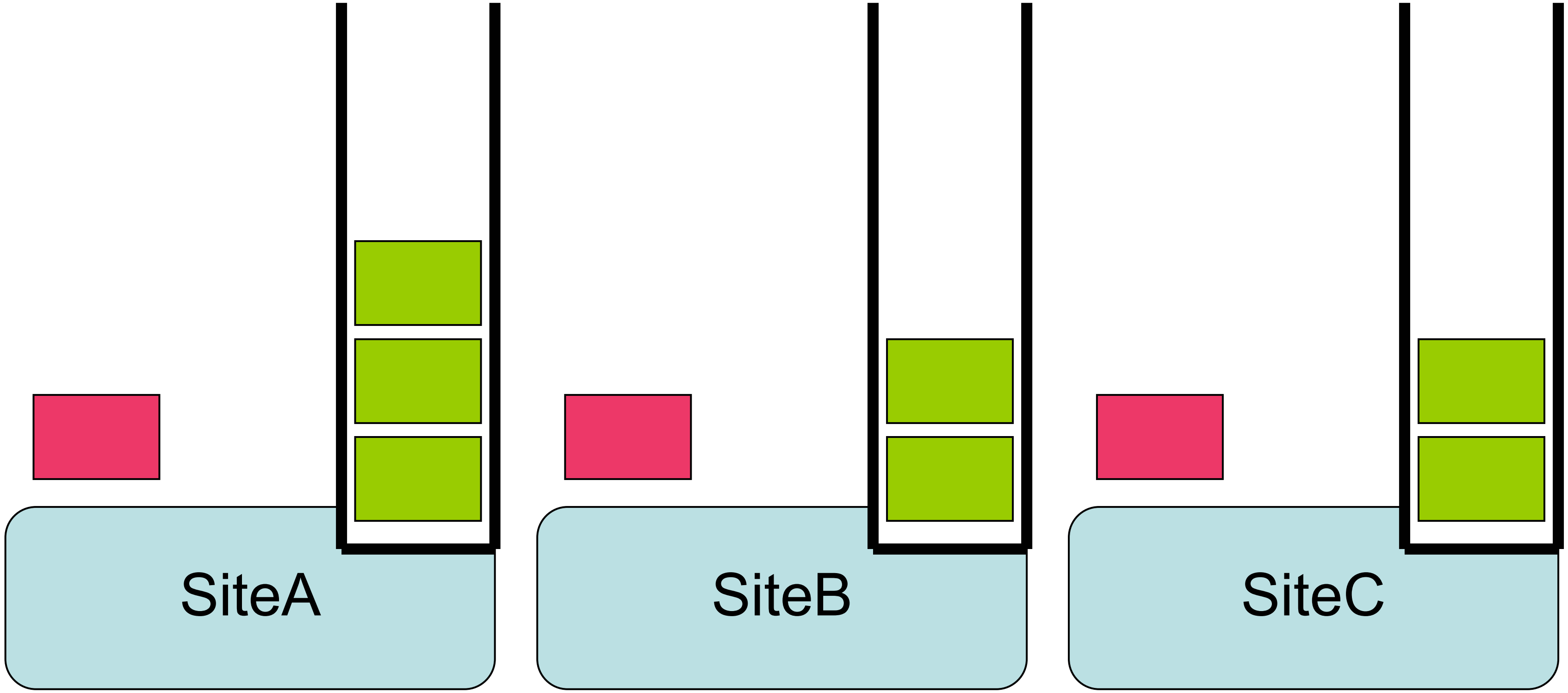
Co-allocation with Advance Reservation

- 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



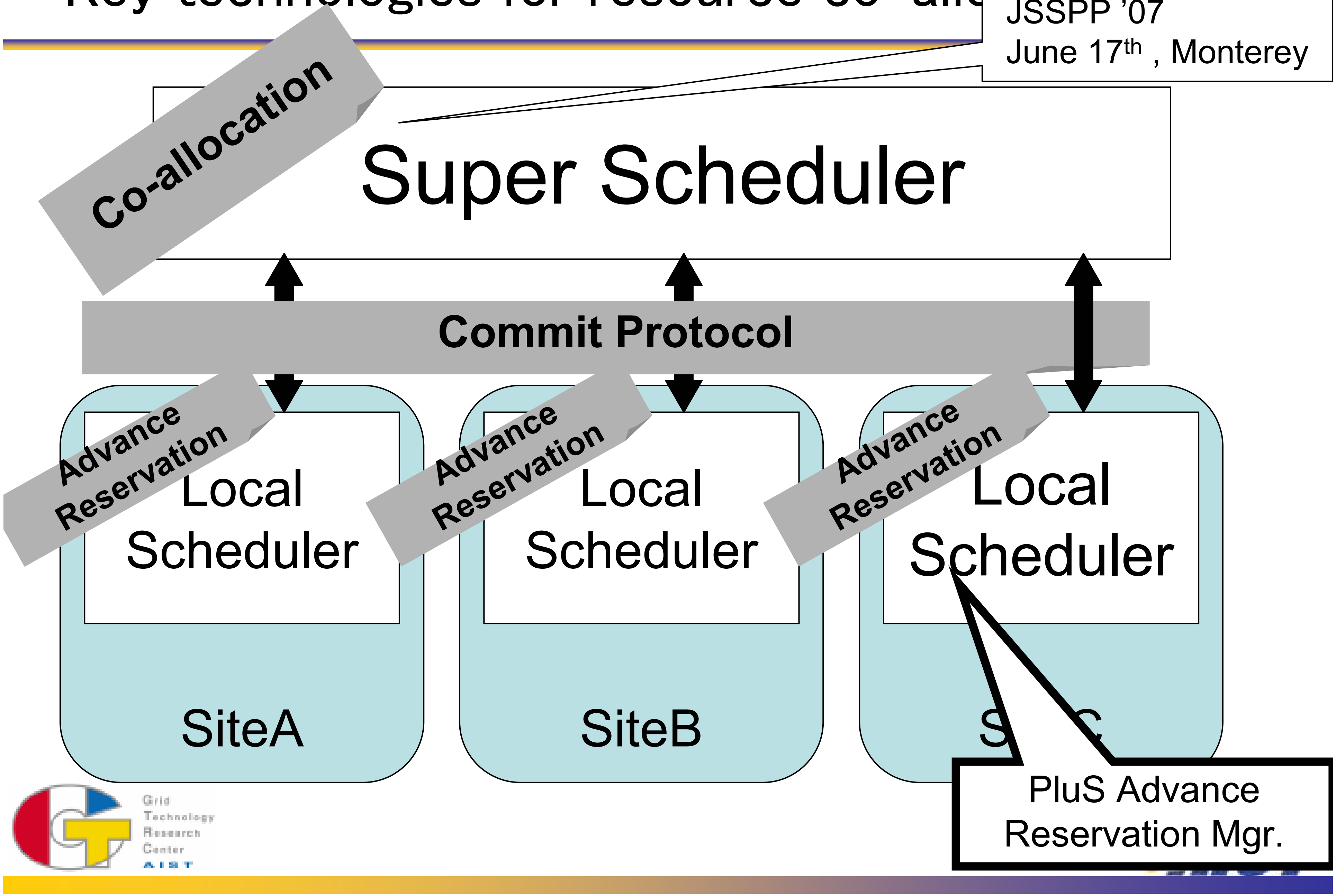
Co-allocation of Coputational Resources（2/2）

- Advance Reservation
  - Allocate time slot, independent of the queue



Key technologies for resource co-allocation

JSSPP '07
June 17th , Monterey



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

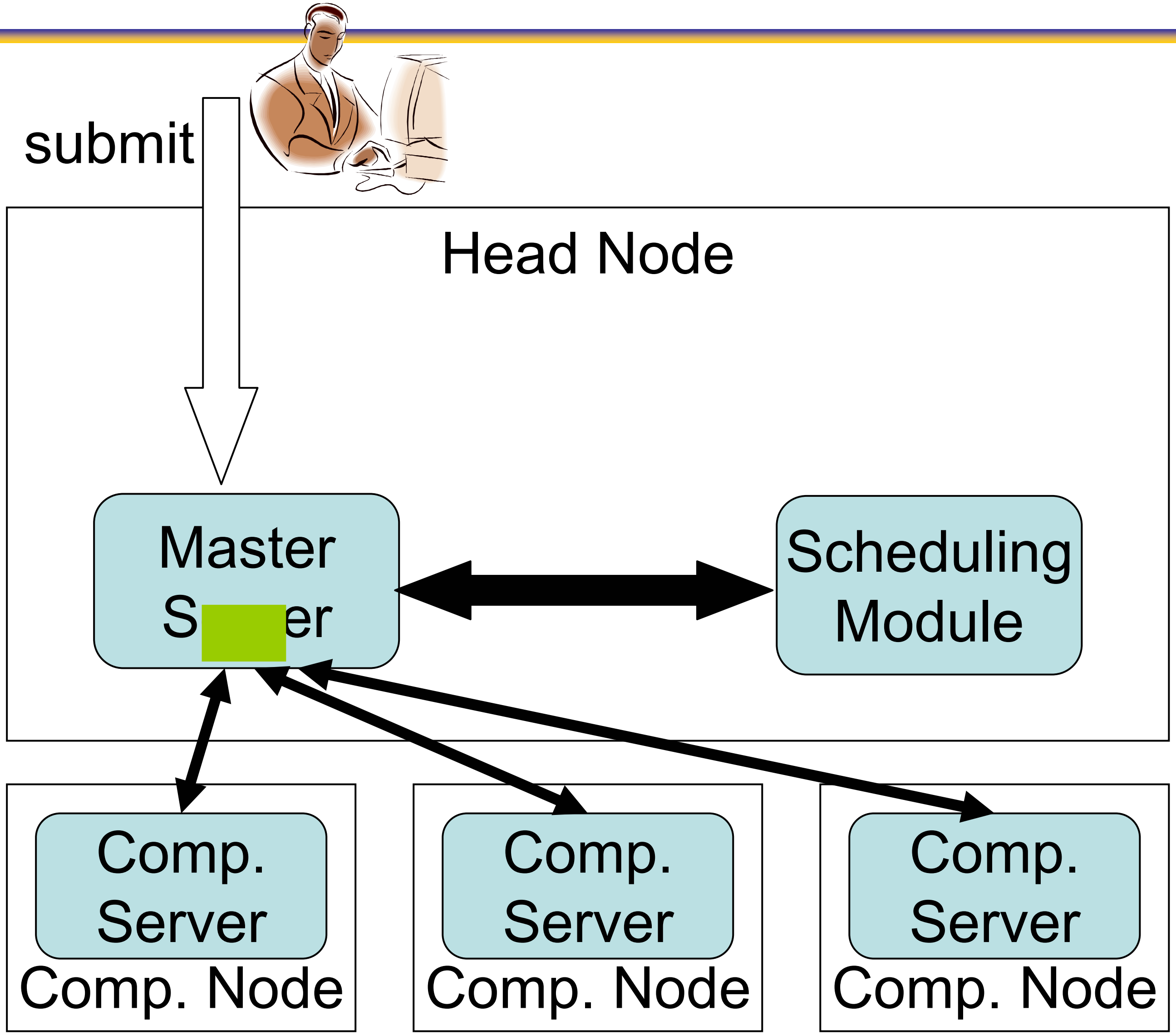
Overview of the talk

- 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



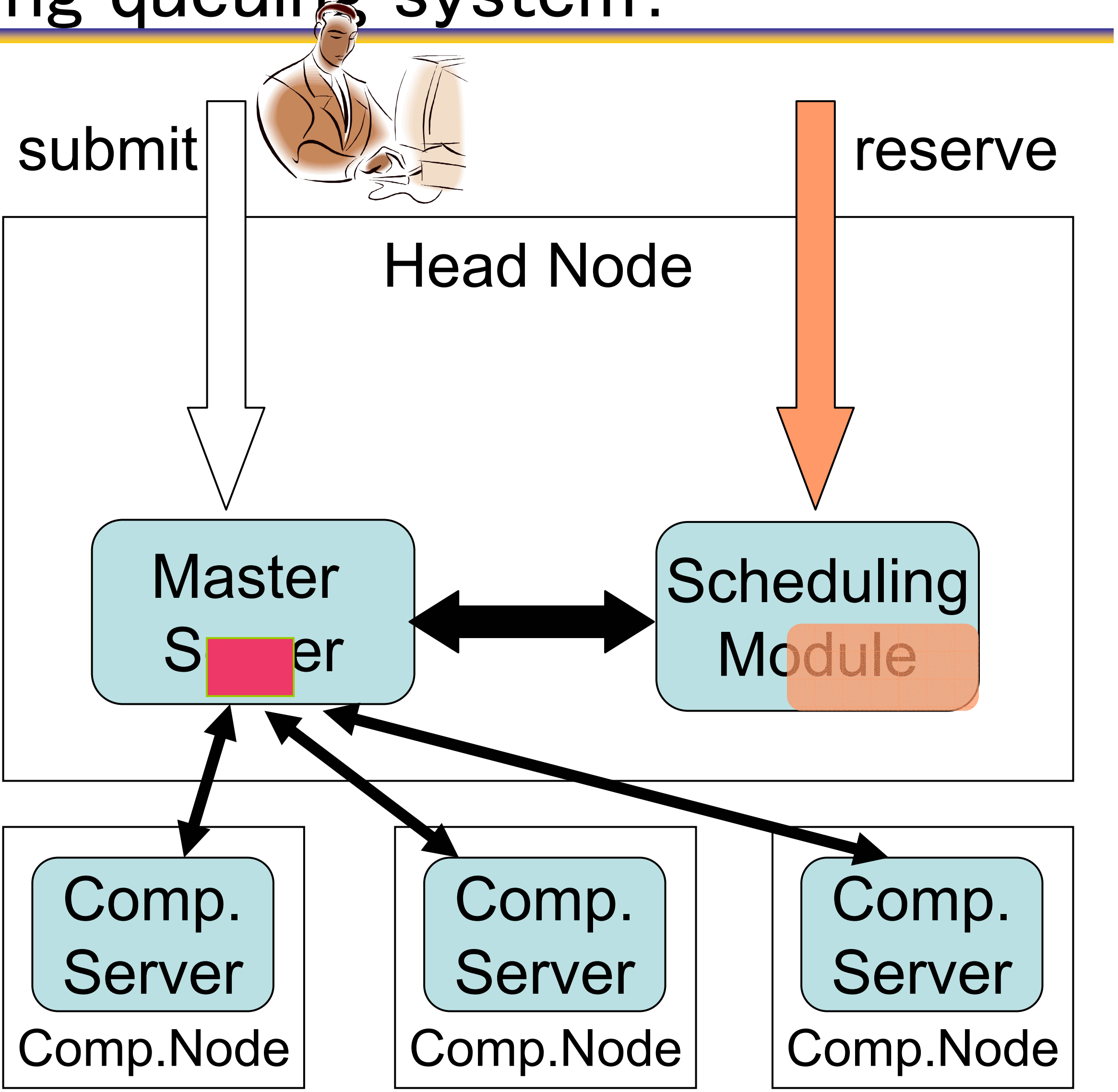
Problem

- Open Source Queuing Systems typically does not support advance reservation capability

- Commercial ones support it, but ..
  - No chance to change the reservation policy
  - Not suitable for research testbed

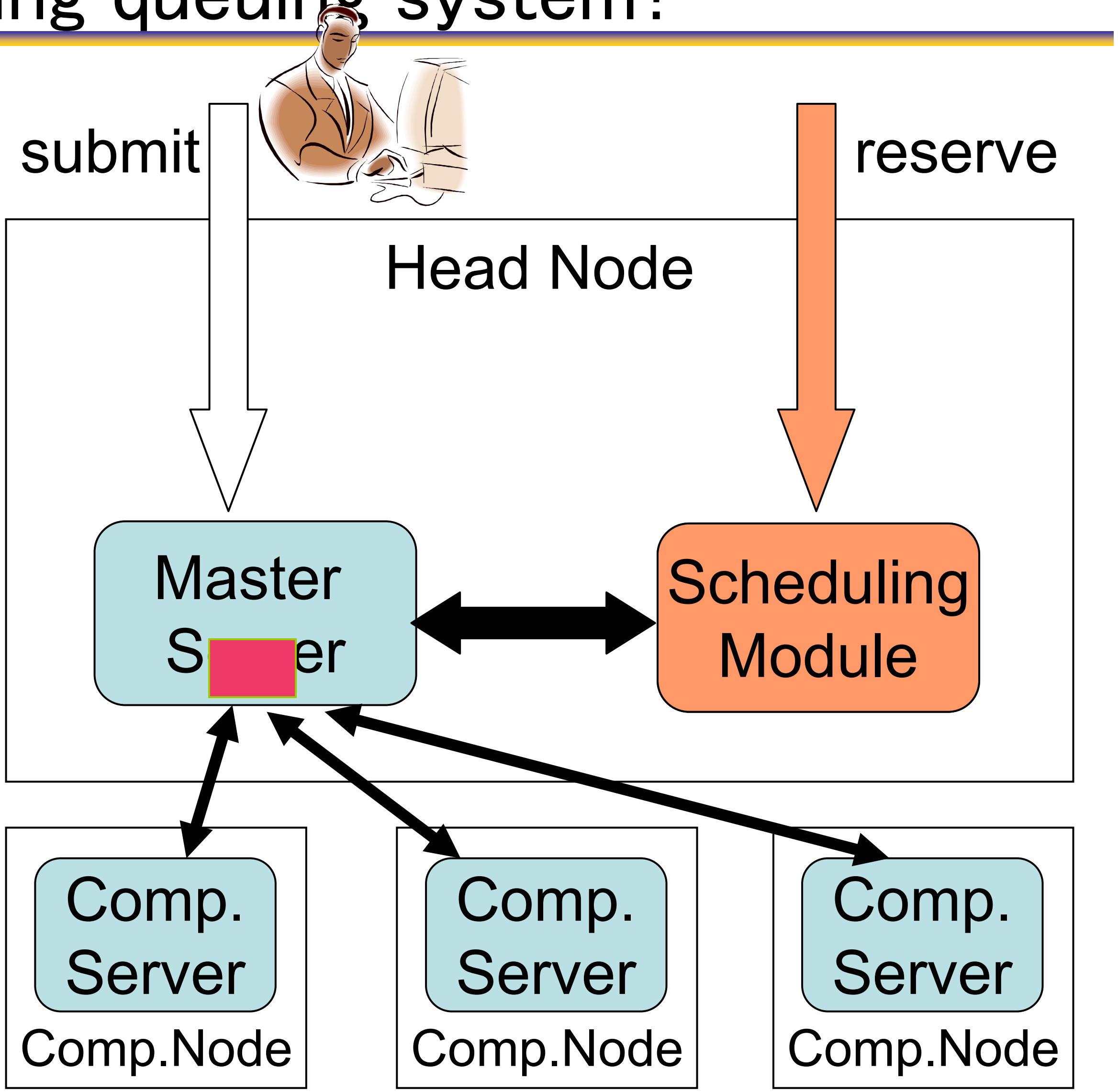
How can we add Advance Reservation capability to existing queuing system?

- 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 out side of the system
  - Not always possible depending on the queuing system capability



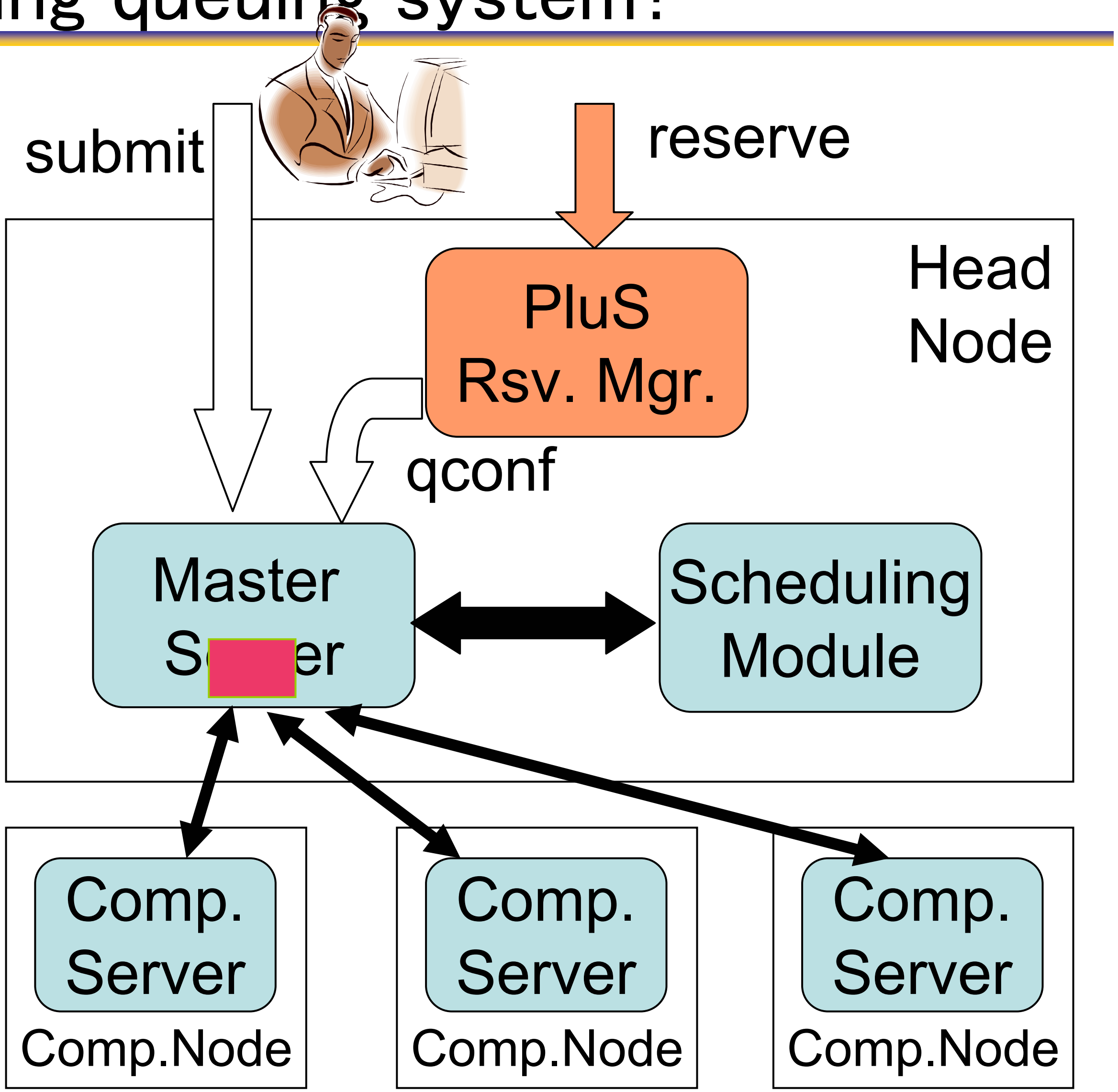
How can we add Advance Reservation capability to existing queuing system?

- 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



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 Rep

TORQUE
Grid Engine

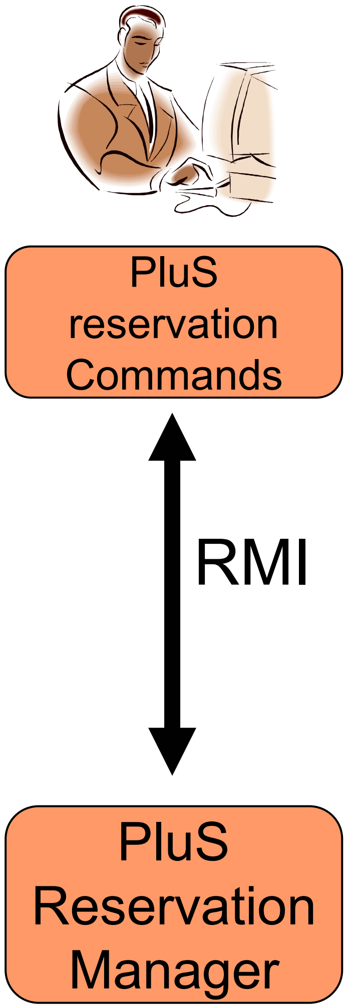
  - ‘Brain transplant’ – You can do anything you want
  - You might have to re-implement all the capability of the existing scheduling module, if needed
- Queue Control Method
  - Not always possible, ity of the targe queuing system

Grid Engine

    - 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

- Make a reservation

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

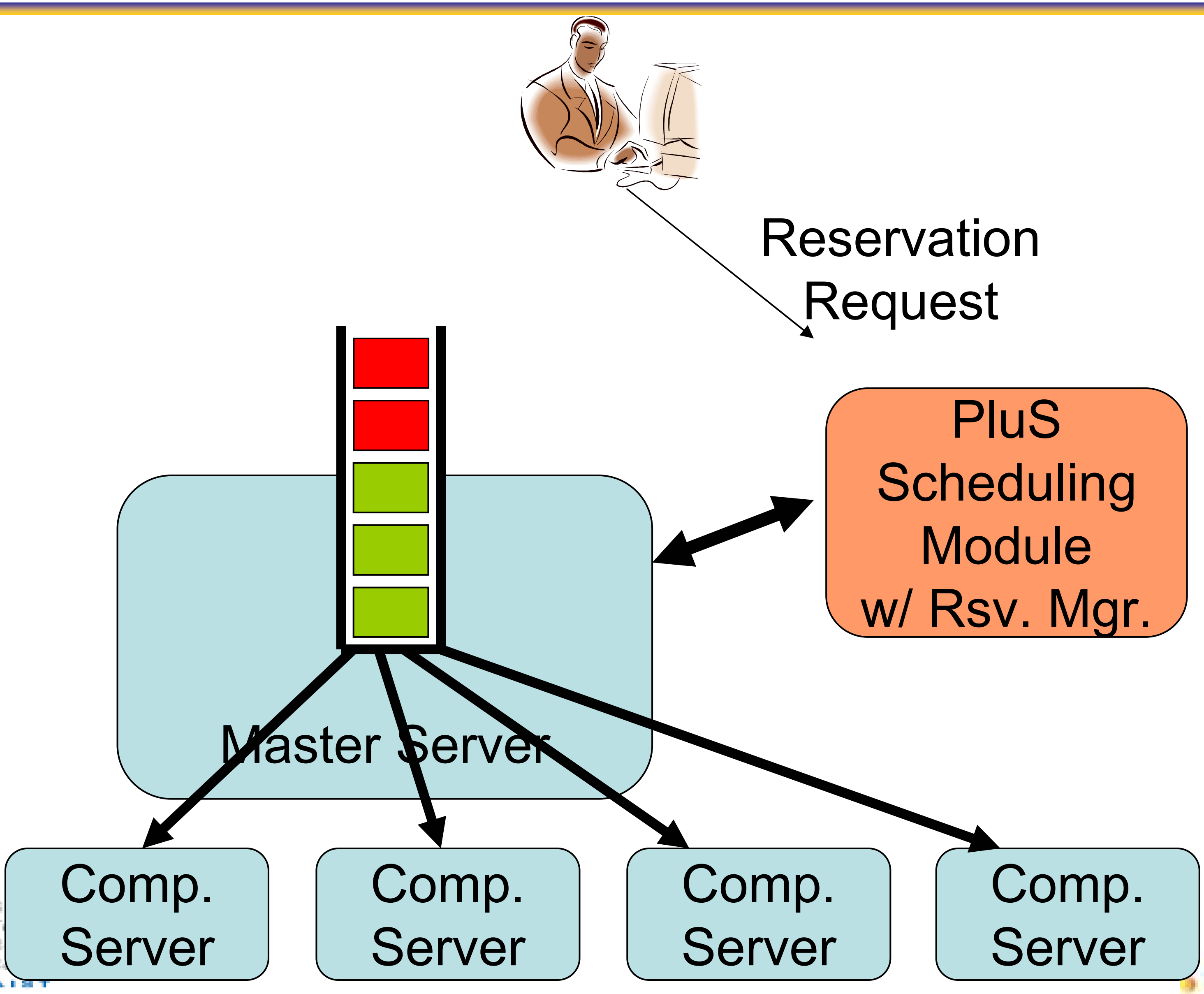
- Confirm the reservation with the reservation ID

```
> 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
```

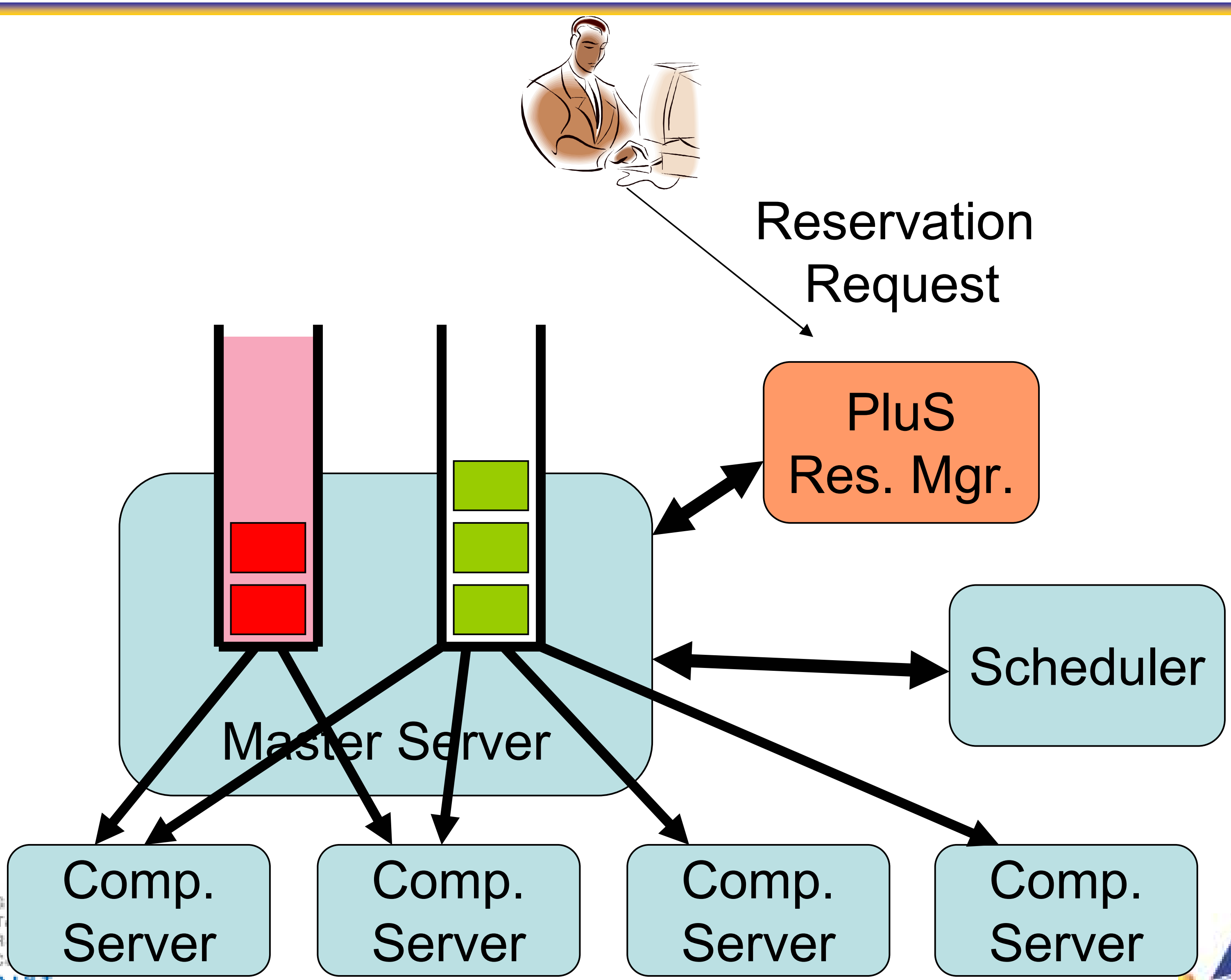
Scheduling Module Replacement



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

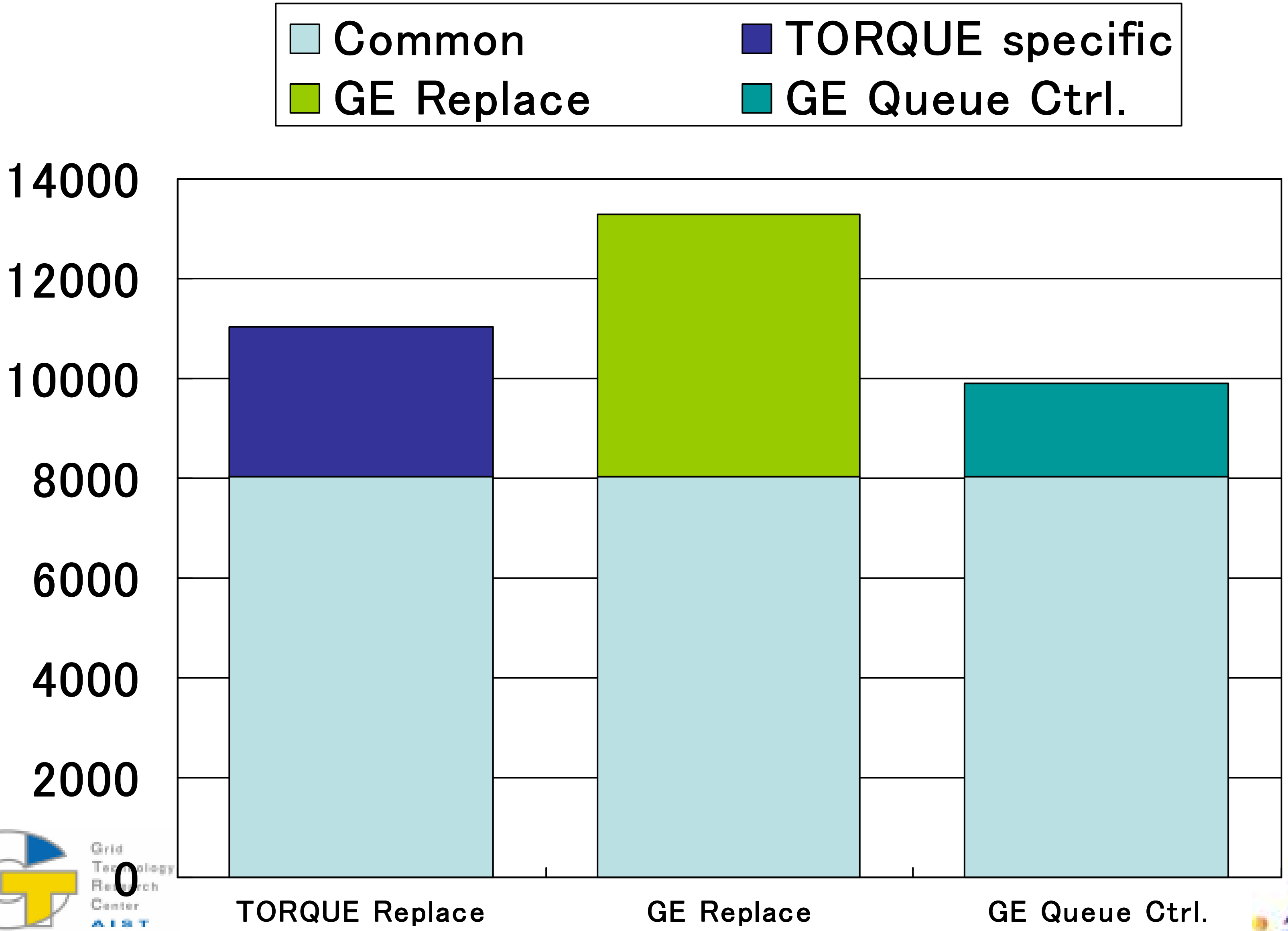
Advance Reservation by Queue Control



Evaluation

- Easisiness of implementaion
  - Is the Queue Control Method really easier to implement?
  - Compare two method 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

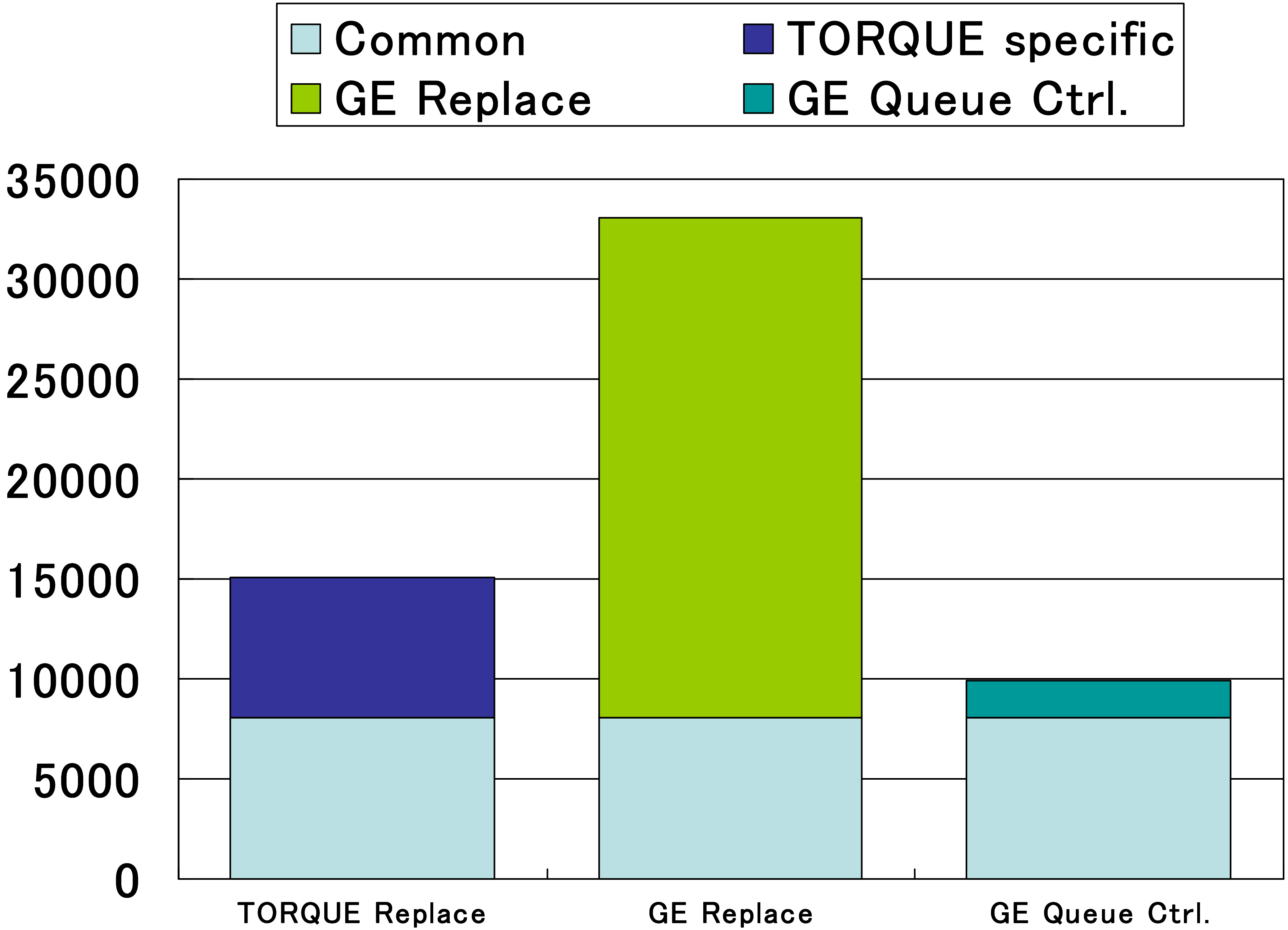


Note on the result

- 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



Comparison with Command Execution Time

- **Experimental Enviro**
  - Pentium III 1.4 GH
  - Memory 2G byte
  - Linux RedHat 8
- **Measurement**
  - using ‘time’ command ure the execution time of the plus commands
  - 10 time trial. The aver as c min. values.

Queue Control is slower.
- ‘qconf’ overhead

Execution time is 1 - 2 sec.
acceptable

	M e Reservation				Cancel Reservation			
	Ave.	Dist.	Min.	Max.	Ave.	Dist.	Min.	Max
Scheduling module replacement	1.02	0.04	0.91	1.54	0.92	0.00	0.85	1.03
Queue Control	1.95	0.02	1.76	2.25	1.02	0.00	0.97	1.11

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

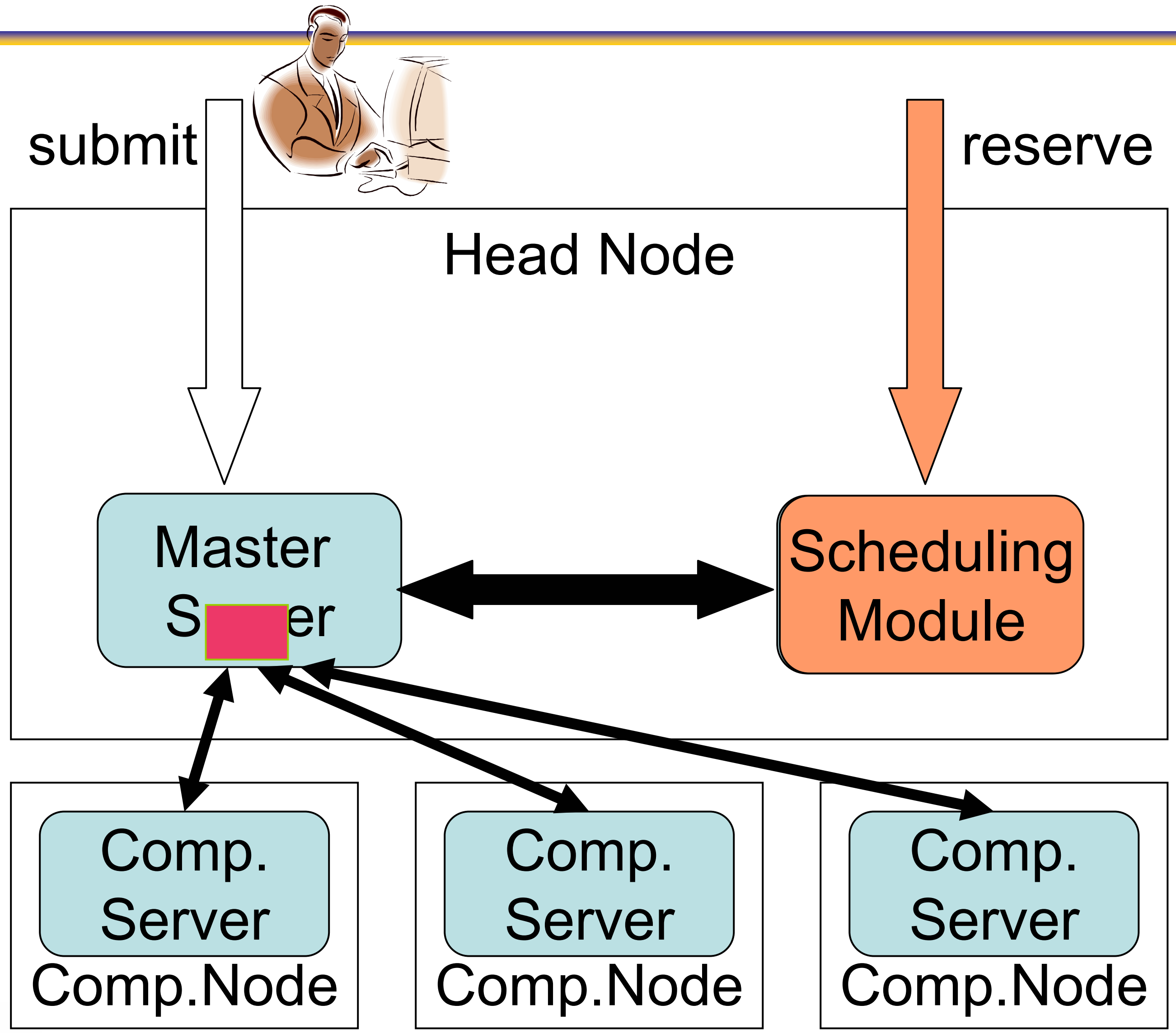
This work is partly funded by the Science and Technology Promotion Program's "Optical Paths Network Provisioning based on Grid Technologies" of MEXT, Japan.

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

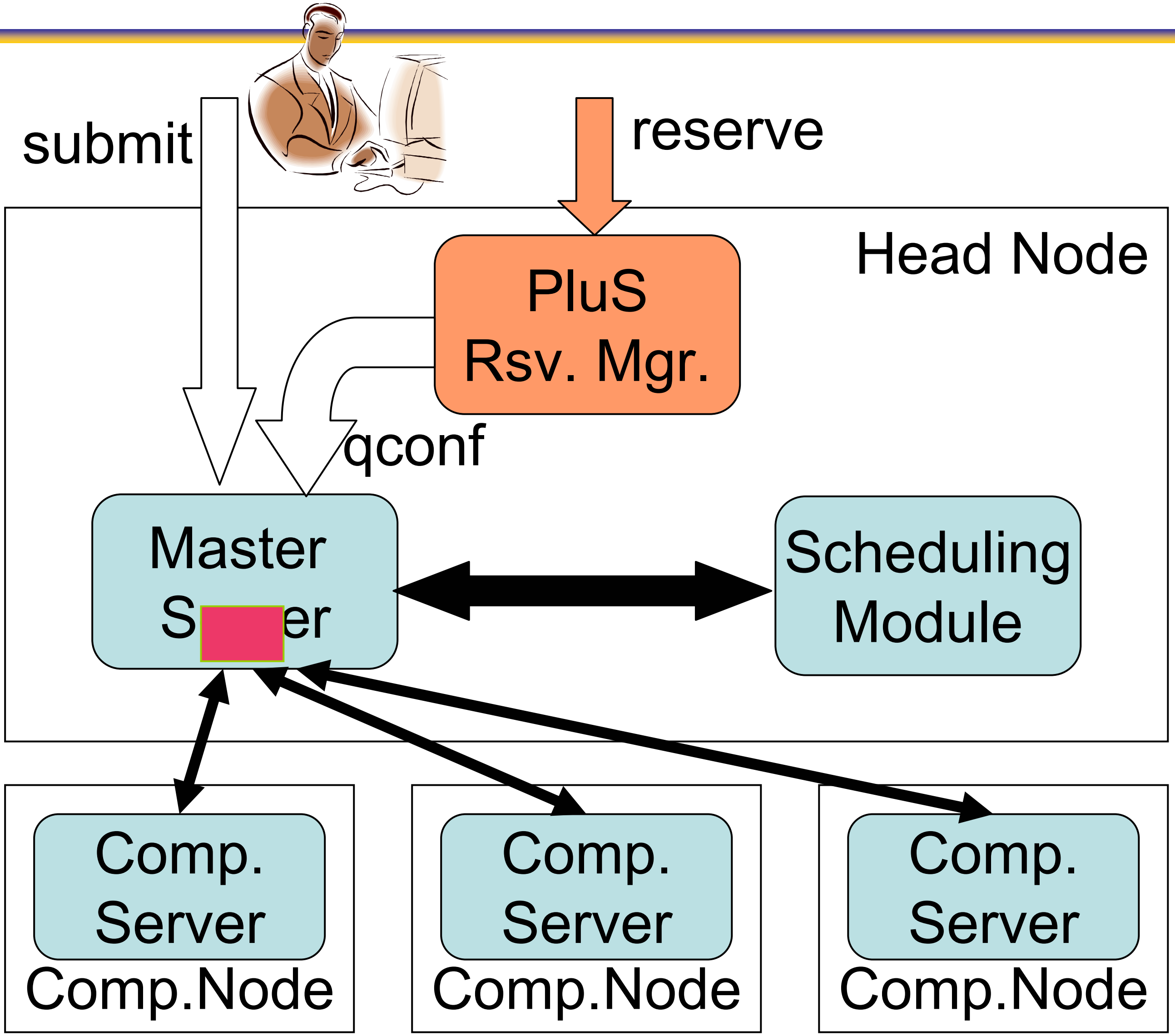
PluS Implementation

- 3 implementations
  - Scheduling module Replacement for TORQUE
  - Scheduling module Replacement for Grid Engine
  - Queue Control for Grid Engine

Scheduling Module Replace Method



Queue Control Method



submit

reserve

Head Node

PluS Rsv. Mgr.

qconf

Master S er

Scheduling Module

Comp. Server

Comp.Node

Comp. Server

Comp.Node

Comp. Server

Comp.Node