



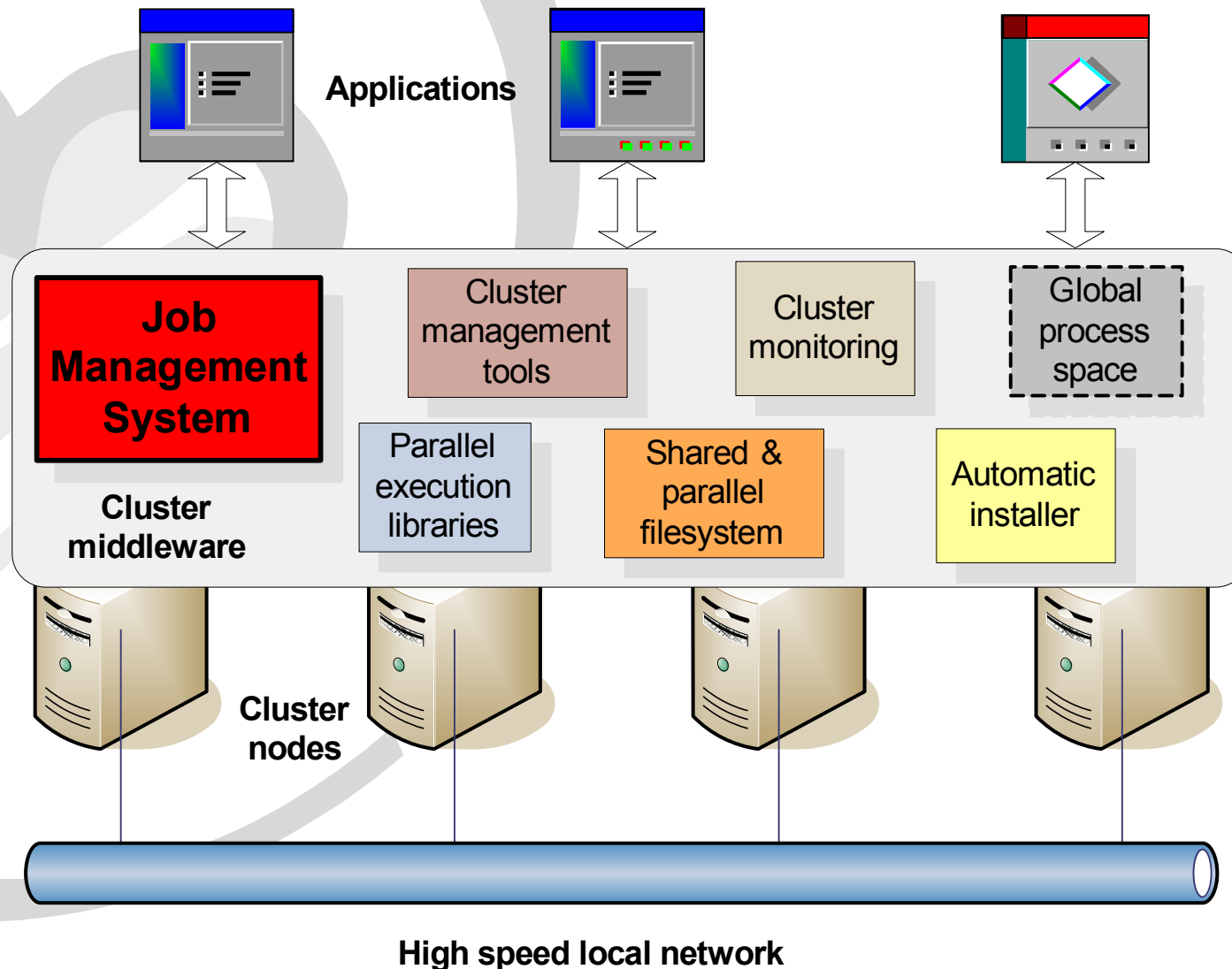
Job Management Systems Analysis

E. Imamagić, B. Radić, D. Dobrenić

Overview

- Introduction
- Job Management Systems
- JMS Architecture
- Components
- Functionalities
- JMS's overview
- Conclusion

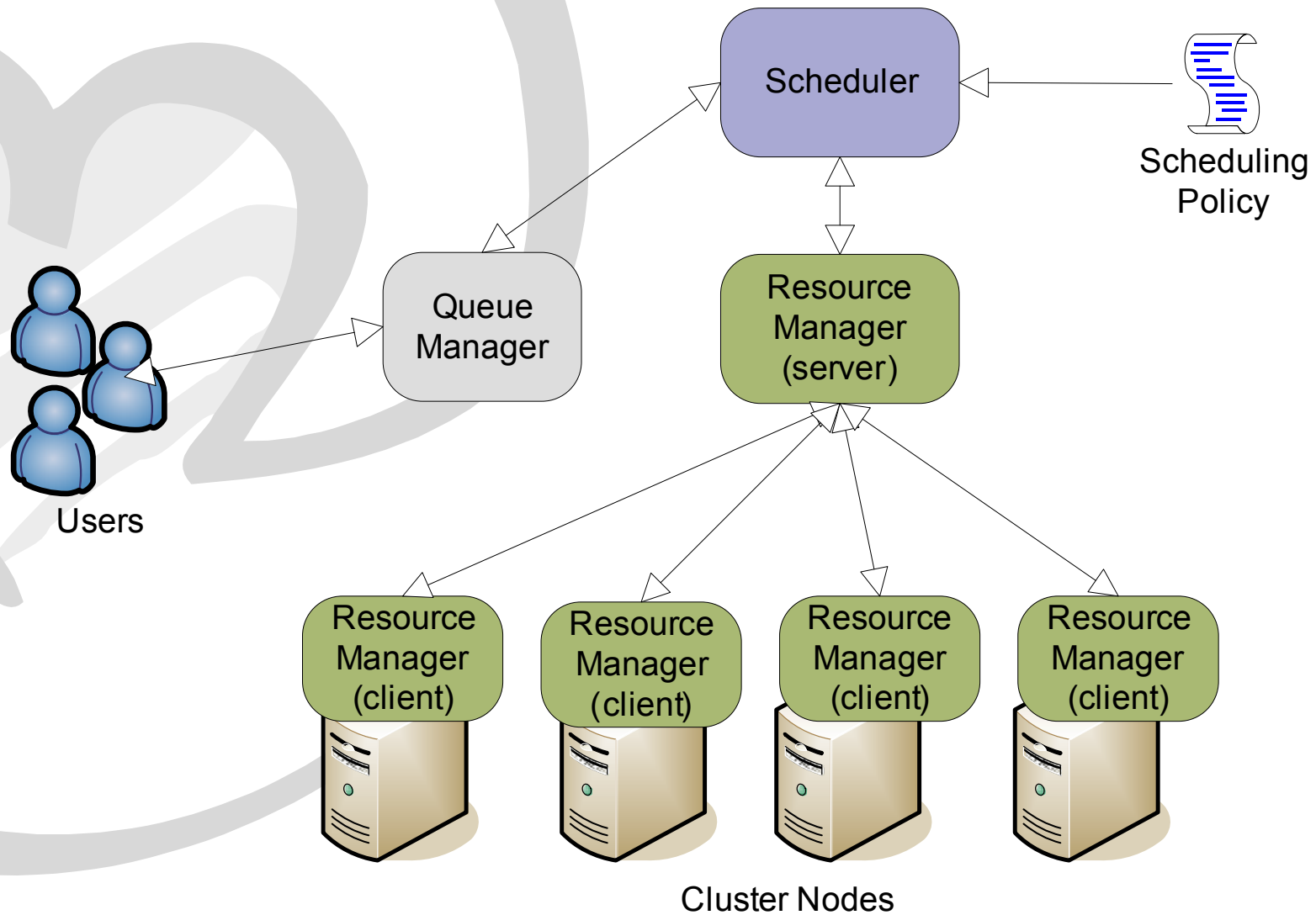
Introduction



Job Management Systems

- What is Job Management System (JMS)?
 - Cluster component responsible for users' jobs control
 - Workload Manager, Batching System, Resource Management System, Queuing System
- Why do we need JMS?
 - Resource usage policy enforcement
 - Resource utilization
 - User interface to computer cluster
- Why is JMS important for Grid?
 - Grid interface
 - Fine-grained resource usage control

JMS Architecture



JMS Components

- Queue Manager
 - User interface
 - Queue management
- Scheduler
 - Job scheduling
 - Policy implementation
- Resource Manager
 - Job execution, monitoring and control
 - Resource monitoring and utilization

Functionalities

- General
 - User interface (GUI, CLI, web portal, API, **DRMAA**)
 - Platforms supported (arch. & OS)
 - Installation method (package, installation scripts, binaries, cluster distribution)
- Job Support
 - Job type (serial, parallel, interactive, workflow, job array)
 - Integration with parallel execution libraries
 - **Job description**
- Security
 - AAA

Functionalities (cont.)

- Queue Management
 - Multiple queues definition
 - Job control (canceling, holding)
 - Job history & statistics
- Scheduling
 - Standard policies (fair share, backfilling)
 - Custom defined policies
 - External Scheduler (e.g. Maui)
 - **Advance reservation**
 - Preemption

Functionalities (cont.)

- Resource Management
 - Node configuration
 - Checkpointing
 - Job migration
 - Dynamic load balancing
 - Fault Tolerance
 - CPU harvesting

JMS overview

	Pro	Con
Condor	• complex jobs • checkpointing • process migration • fault tolerance	• not designed for parallel jobs • no custom sched. alg. • no sched. alg. for all job types
PBS	• parallel lib. Integration • integration with Maui • multiple queues • advanced reservation • job preemption	• limited GUI • no complex job types • limited node configuration • no load balancing • unknown future
SGE	• GUI • job arrays • rich node configuration • fault tolerance • load balancing	• parallel lib. Integration • limited integration with Maui • no multiple queues • no advanced reservation • no job preemption

Conclusion

- JMS are crucial part of cluster
- All analyzed systems provide great deal of functionalities
- Grid integration
 - PBS and Maui are widely supported (Globus, UNICORE)
 - SGE ver. 6 has all needed features
 - Condor provides many tools for Grid integration
- Grid & cluster future
 - Service Layer Agreement (SLA) support
 - WS Agreement (ex. OGSI Agreement)
 - DRMAA implementation