CoreGRID

Managed by

European Research Consortium for Informatics and Mathematics
ERCIM

Research Challenges in Grids
Session: Scheduling and Monitoring

Ramin Yahyapour
CoreGRID / University Dortmund

Bridging Global Computing with Grid (BIGG)
Sophia Antipolis, 28/29.11.2006

CoreGRID
Institute on Resource Management and Scheduling

Information Society
Technologies

Motivation

- There is obvious need for smart resource management solutions for distributed environments across different administrative domains.

HPC Grid Computing

- Parallel Computing
- Cluster Computing
- Desktop Computing

Enterprise Grids

- Business Services
- Virtualization
- IT Management

Ambient Intelligence

Ubiquitous Computing

Mobile Grids

- Smart Phones, PDA, Mobile Devices

Necessary base technologies are available and systems are already in use!

But many advanced functions are still missing; implementation and use is still difficult

Different Views

Scientific Environments

- Often job-oriented
- Mostly batch
- Many jobs
- No or limited business model
- Performance oriented
- Level of service is of secondary interest

Enterprise Environments

- Often long-term commitments
- Commercial applications
- Business services are considered
- Mostly within a company
- Cost consideration is an issue: EBIT, TCO
- High reliability/QoS requirements
- Interest in Virtualization, SLAs

Both deal with a distributed service infrastructure that needs efficient management considering similar side constraints.

General Challenges (for Scheduling and Monitoring)

- More resources types
  - Compute, data, network, storage, licenses,… arbitrary services
- More complex job requests
  - Co-allocation of several resources for a request
- Limited Information, limited control
  - Privacy concerns, issues of autonomy
- Different access policies
  - Resource access limited by resource providers
- Support for business models
  - or accounting aspects
- Scalability
  - thousands to millions of resources
- Reliability
  - Dynamic environment requires fault-management
- Manageability
  - Automatic, seamless management, self-*
- Efficiency
  - Optimization of resource utilization, quality of service
- Security, Dependability

Scheduling and Resource Management

- The smart and automatic utility management is still not available!
  - "The grid should be invisible"
  - easy-to-use
  - efficient
  - reliable
  - broader applicability to many use-cases
  - self-*

Scheduling is a key part of this smart management.
(aka brokerage, matching, negotiation, resource mgmt, workload mgmt, utility mgmt)

Assessment:

- Need for sufficient theoretical understanding, analysis, models
- Need for technical foundations and practical solutions
- Both are improving!

Multi-level Scheduling

In a Grid, resources typically belong to different providers/owners.

- They usually have different policies and constraints.
- These policies are often enforced by local management systems.
  - high degree of heterogeneity

Grid vs. local resource management systems

- Grid-RMS:
  - has to deal with many heterogeneous resources
  - in a highly dynamic environment
  - without exclusive control over any resources
  - decentralized
- Local resource management
  - typically one or few homogeneous resources
  - in a static configuration
  - within a single administrative domain

Scheduling now occurs on several levels

Alternatives

Case 1: Specialized resource management and scheduling dedicated to a single application domain.

- Goal: high efficiency
- Cost: higher development effort

The RMS is adapted to:

- application and its workflow
- resource configuration

Only certain types of jobs and resources are considered.

Case 2: Generic Grid RMS that is open for many applications

- Goal: lower development effort
- Cost: may be less efficient

The RMS is adapted to:

- Generic interfaces are required that are adapted to several front- and backends.

Broader applicability
Needed for proliferation

Hot Topic: Interoperability and Standardization

- Need for well-defined interfaces to core services:

Access to information

Status monitoring

Allocation/Scheduling

SLA Management

Execution Management/Provisioning

Accounting and Billing

Job, Resource, Performance Models

- OGF, W3C, OASIS, de-facto standards
- Support for different implementations while maintaining interoperability between these implementations

Hot Topic: Monitoring

The small, idealistic Grid is not realistic!

Status monitoring:

- job and resource condition
- SLA status

Autonomic aspects:

- detection of unexpected changes
- allows prediction of system behavior
  - related to an individual job
  - and to general demand
- Automatic triggering of re-scheduling/re-allocation

Ontology and semantics?

How to monitor?

How to draw conclusions from information?

Live with unreliable information

Translate between information models

Combine monitoring, notification, reasoning, triggering of actions

Aspects:

flexibility

scalability

Hot Topic: Suitable Scheduling Models

Simple job submission/distribution is not sufficient!

Different application scenarios

- Online problem
- parallel, sequential jobs
- Batch, interactive, reserved
- co-allocation and orchestration
- workflows

Provider policies

- access, cost, security

User/application policies

- scheduling objectives
- cost/budget management
- Deadlines

Cooperation between RM systems

Support for different (= individual) algorithms and strategies

How to support different application scenarios?

What are optimization goals?

How to include different strategies?

Multi-level scheduling

Multi-criteria optimization

Coordinated scheduling (workflows/co-allocation)

Using service-level agreements

Interoperability of schedulers

Market-oriented/economic scheduling

Aspects:

flexibility, ease-of-use

support for business models

efficiency

Hot Topic: SLA Management

Best effort is not sufficient anymore!

Service Level Agreements are cornerstones for future management:

- reliability
- orchestration of services
- quality of service
- business models
- accountability
- billing

How to model SLAs?
How to create/negotiate SLAs?
How to manage SLAs?
How to monitor SLAs?

WS-Agreement/Negotiation
SLA scheduling/optimization
Negotiation strategies

Aspects:
reliability
support for business models

Conclusions

Key challenges:

- Interoperability
  - connecting different Grid-RMS implementations and types
  - and between Grid-RMS and local RM systems
  - standardization of core interfaces
  - interaction and negotiation between different systems
- Scalable Monitoring/Scheduling
  - general architecture should cover thousands or millions of resources
  - predict, recognize and manage faults or performance problems
- Intelligent Scheduling Models
  - Reliable, automatic and efficient self-management
  - SLA Management
  - Support for complex application scenarios: Co-allocation, workflows, virtualization
  - flexible optimization goals: cost, QoS, …

The Grid scheduling system should be invisible to the user and provide a pervasive, reliable common architecture allowing different implementations while maintaining interoperability.

Thanks for the attention!