Resource Management and Scheduling

Mechanisms in Grid Computing

Edgar Magaña Perdomo

Universitat Politècnica de Catalunya
Network Management Group
Barcelona, Spain

emagana@nmg.upc.edu
http://nmg.upc.es/~emagana/
Outline

• Introduction

• Resource Management in Grid Computing

• Scheduling in Grid Computing

• Our Approach

• Policy-based Management System

• Heuristic Methodologies

• Conclusions
• **GRID Computing:**

– Grid Systems are interconnected collections of geographically distributed and heterogeneous resources harnessed together to satisfy different computational needs by a great variety of users.

– Exploiting underutilized computational resources (reducing costs)

– Allowing very large computation through virtual parallel machines (speed up)

– The essential mode to implement the Grid is through Virtual Organizations (VO).
Introduction: Grid Computing (II)

• Two points of view:
  – User’s view: An analogy to power grid
  – Grid Developer’s view: Account management, resource management, application scheduling, and application/development environments; all parts should work in a secure way
  – and more…

• Grid Computing Taxonomy:
  – Scale: Intragrid, Intergrid; cluster, campus grid, global grid
  – Application/Development Environment: computational/data grid, information grid, and knowledge grid
Next Generation Grid Computing Tendencies:

– Services should be provided to users regardless of network technology, administrative domain or operative platform.

– Effective access to large amount of computing, network and storage resources, reducing procurement, deployment, maintenance and operational cost.


– Changes in Grid Services: WS-RF
Main Problems:

- Swift and dynamic allocation and reservation of computational resources
- Allocation of network resources per service
- Configuration of resources on fly
- Deployment of distributed services in heterogeneous and multi-domain networks
- Communication mechanisms with Grid/Web Services
Grid Computing

Resource Management

and

Scheduling Mechanisms
Resource Management in Grid (I)

• **Basic Requirements:**
  - Members should be trustful and trustworthy.
  - Sharing is conditional.
  - Should be secure.
  - Sharing should be able to change dynamically.
  - Discovery and registering of resources.
  - Can be peer to peer or client/server.
  - Same resource may be used in different ways.
  - Well defined architecture and protocols.
Resource Management in Grid (II)

• Grid is affected by continuous innovations:
  – Schemas conversion technologies
  – Allow data to move between different systems technologies
  – Intercommunication between different network domains
  – Services on the net need specific resources requirements
  – Grid need to handle resources in more dynamic way
  – Grid Services will require to co-ordination and orchestration of resources at run time
  – QoS Aware
Resource Management Architecture:

- **Super Scheduler**
  - Job Queuing
  - Queuing
  - Queuing

- **Local Scheduler**
  - Queuing

- **Resource Information Service**

- **Collection/monitoring**

- **Agents**
  - Resource

- **Agents**
  - Resource
Resource Management in Grid (IV)

• Resource Management Challenges:
  – Satisfactory end-to-end performance through multiple domains
  – Availability of computational resources
  – Handle of conflicts between common resources demand
  – Fault-tolerance
  – Inter-domain compatibility (P2P)
• **Difficulties in Resource Management:**
  – Autonomy
  – Heterogeneity
  – Dynamics

• **Grid Resources:**
  – Computing Power
  – Disk Space
  – Memory Space
  – Network Bandwidth
  – Software, etc.
• Stages of Grid Resources Management:
  – Phase 1: Resource Discovery:
    » Find available resources
  – Phase 2: Systems Selection:
    » Allocate the resources
  – Phase 3: Job Execution
    » Run the job
    » Log the resource usage
    » Release the resources
    » Charge the user

Scheduling (Job Shop)
Scheduling Problem (I)

- **Formal Definition:**
  - A finite set of \( n \) jobs
  - Each job consists of a chain of operations
  - A finite set of \( m \) machines
  - Each machine can handle at most one operation at a time
  - Each operation needs to be processed during an uninterrupted period of a given length on a given machine
  - Purpose is to find a schedule, that is, an allocation of the operations to time intervals to machines, that has minimal length
Scheduling Problem (II)

• A Grid ($G$) consists of “n” Nodes:

• A Node ($N$) consists of “m” Resources ($R$):

\[
\tilde{N}_i = \begin{bmatrix} R_1 & R_2 & R_3 & \cdots & R_m \end{bmatrix}
\]

\[
\tilde{G} = \begin{bmatrix}
\tilde{N}_1 \\
\tilde{N}_2 \\
\tilde{N}_3 \\
\vdots \\
\tilde{N}_n
\end{bmatrix}
\]

\[
\tilde{G} = \begin{bmatrix}
R_{11} & R_{12} & R_{13} & \cdots & R_{1m} \\
R_{21} & R_{22} & R_{23} & \cdots & R_{2m} \\
R_{31} & R_{32} & R_{33} & \cdots & R_{3m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
R_{n1} & R_{n2} & R_{n3} & \cdots & R_{nm}
\end{bmatrix}
\]
Traditional Scheduling (I)

• Complexity Analysis:

\[ C = \binom{n}{k} = \frac{n!}{k!(n-k)!} \]

\[ Ct = \sum_{k=1}^{n} \binom{n}{k} = \sum_{k=1}^{n} \frac{n!}{k!(n-k)!} \]

• Binomial Analysis:

\[(1 + x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k \]

\[(1 + x)^n = \binom{n}{0} + \binom{n}{1} x + \binom{n}{2} x^2 + \binom{n}{3} x^3 + \ldots + \binom{n}{n} x^n \]

If \( x = 1 \)

\[(1 + 1)^n = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{n} \]

\[2^n = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{n} \]

\[2^n - 1 = \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{n} \]
Approaches – Traditional Scheduling (II)

- Algorithm Analysis:

\[ g(n) = 2^n - 1 \]
\[ f(x) = O(g(n)) \]
\[ f(x) = O(2^n - 1) \]
\[ f(x) = O(2^n) - O(1) \]
\[ f(x) = O(2^n) \]
Approaches – Traditional Scheduling (III)

- Most Common Algorithms Complex Times:

<table>
<thead>
<tr>
<th>n</th>
<th>log(n)</th>
<th>n*log(n)</th>
<th>n^2</th>
<th>2^n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>0.7</td>
<td>3.5</td>
<td>25</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>1.0</td>
<td>10</td>
<td>100</td>
<td>1024</td>
</tr>
<tr>
<td>20</td>
<td>1.3</td>
<td>26</td>
<td>400</td>
<td>1,048,576</td>
</tr>
<tr>
<td>50</td>
<td>1.7</td>
<td>85</td>
<td>2,500</td>
<td>1.12590*10^{15}</td>
</tr>
<tr>
<td>100</td>
<td>2.0</td>
<td>200</td>
<td>10,000</td>
<td>1.26765*10^{30}</td>
</tr>
<tr>
<td>200</td>
<td>2.3</td>
<td>460</td>
<td>40,000</td>
<td>............</td>
</tr>
<tr>
<td>500</td>
<td>2.7</td>
<td>1,349</td>
<td>250,000</td>
<td>............</td>
</tr>
<tr>
<td>1,000</td>
<td>3.0</td>
<td>3,000</td>
<td>1*10^6</td>
<td>............</td>
</tr>
</tbody>
</table>
Our Approach (I)

• Main Target:

“Development of Algorithms for Scheduling Computational Resources belonging to the Grid Computing Infrastructure based on Heuristic Methodologies”
Our Approach (II)

• Complementary Targets:

  – To **implement** the resulting algorithm into the environment of a Policy-Based Grid Network Management Architecture

  – To guarantee certain **Quality of Service (QoS)**

  – To offer a **rapid and cost-effective** access to large amounts of computing, memory, software, network and storage resources across multiple domains

  – **Scheduling of resources** regardless of network technology, operative platform or administrative domain

  – Evaluate the **performance** of the Resource Management System
**Policy-Based Management Approach**

**Definition:**

- Policies are rules of the type: `if <condition> then <action>`
- Automated and flexible way of expressing the system behavior
- The system behavior can be dynamically adapted according to business needs
- Naturally fitted to specify the business objectives
- Technology independent and Standardized by the IETF

---

Policy-Based Management System

• IETF Policy framework and architecture:

Functions of the PDP
- Interpret policy
- Detect policy conflicts
- Receive policy decision requests from PEPs
- Determining which policy is relevant
- Send policy elements to PEPs

Functions of the PEP
- Metering (auditing of policy compliance)
- Applying actions according to PDP decisions
1.- Service Level Agreement (SLA)

– Grid Infrastructure Providers (GIP) and Grid Services Consumers (GSC)

– QoS Levels:
  » Diamond
  » Gold
  » Silver
  » Bronze
PBMS - High Level Functionality

2.- Grid Service Requirements – NL Policy

- SLA Service Requirements
- Web Services – Resource Framework Requirements
  - Service Descriptor – WS-Resource Properties Document
  - Grid Service Instance by UDDI Registry
- Inter-Domain Communications (P2P)

3.- NL Policy Analysis

- Policy Manager to PDP Manager
- Resources Availability by Resource Manager
- Resources Status by Monitoring System
4.- Selection of Grid Target Nodes

- **Resource Manager** contacts several times with Monitoring System
- **Resource Manager** computes a set of available nodes offering their resources

5.- Grid Nodes Configuration

- QoS PDP creates the EL policies
- QoS PEP enforces the EL policies in the specific nodes
- Service PDP – PEP respectively, keep the functionality of the service according the service requirements
Simulated Annealing

Tabu Search

Evolutionary Algorithms
• **Simulated Annealing:**
  
  – It has the ability to avoid becoming trapped at local minima.
  
  – The algorithm employs a random search.
• **Tabu Search:**

  - Meta-strategy for guiding known heuristics to overcome local optimality.
  - Iterative technique which explores a set of problems solutions \((X)\).
  - Moves from solution \(s\) to another solution \(s'\).
  - These moves are performed with the aim of efficiently reaching optimal solution by the evaluation of some objective function \(f(s)\).
  - Only parts of the neighborhood are explored.
  - It might be worth returning after a while solution visited previously to search in another direction.
**Evolutionary Algorithms (Genetic Algorithms):**

- Adaptive methods that can be used to solve optimization problems.
- Based on the evolutionary process of biological organism.
- Their search is constrained neither by the continuity of the function under investigation, nor the existence of a derivate function.
- Past data and results are used to determine future results.
- They could be running in parallel way
State of the Art (I)

• **GRAM and GARA:**
  
  – They are part of the Globus Toolkit.
  
  – Use low level primitives.
  
  – Mechanism to obtain reservation in advance.
  
  – Complex commands to inexpert users.
  
  – Need for upper layer middleware.
  
  – Just work under Globus.
  
  – Some problems handing novelty Grid Services specifications.
State of the Art (II)

• **Nimrod-G:**
  - A resource broker for managing, steering, and executing task farming applications on the Grid based on deadline and computational economy.
  - Based on users’ QoS requirements, Nimrod-G Broker dynamically leases services at runtime depending on their quality, cost, and availability.
  - Uses Globus – MDS, GRAM, GSI, GASS.
  - Generic Dispatcher & Grid Agents.
  - Transportation of data & results.
State of the Art (III)

**Condor-G:**

- Condor is a high-throughput scheduler.

- Condor-G uses Globus Toolkit libraries for:
  - Security (GSI)
  - Managing remote jobs on Grid (GRAM)
  - File staging & remote I/O (GSI-FTP)
  - Grid job management interface & scheduling

- Supports single or high-throughput apps on Grid

- Personal job manager which can exploit Grid resources
Publications (I)

Journals:


Congresses and workshops:


Publications (II)

Congresses and workshops:


Books:

Algorithms Performance Analysis

• **Analytical Methods:**
  – Worst case and average performance analysis.
  – Bounds.

• **Empirical Testing:**
  – By testing a heuristic across a wide range of problem instances.

• **Statistical Inference:**
  – The problem of estimating a parameter of a statistical population from sample information.
Conclusions (I)

• **Scheduling Algorithm Approach**
  
  – Sub-optimal algorithm for scheduling computational resources
  
  – Managing heterogeneous resources
  
  – It might reduce scheduling computational resources times
  
  – Support for dynamic, reconfigurable on demand, secure and highly customizable computing storage and networking environments
  
  – Three services resources parameters:
    
    » Users requirements
    
    » Resources availability
    
    » Grid services requirements
Conclusions (II)

• Integration with Policy-Based Grid Network Management System
  
  – An architecture taking advantage of the synergy obtained by coupling Policy-based technology and Heuristic Methodologies  
  
  – Simplifies grid services deployment and management 
  
  – Optimal management of the network resources 
  
  – Scalable architecture as well as automate 
  
  – QoS Architecture 
  
  – Deployment and activation of Grid Services in all planes
Thank You!!!
Any Questions ??

Edgar Magaña

http://nmg.upc.es/~emagana/
emagana@nmg.upc.edu

Managing Grid Computing