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/

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Outline

- Introduction
- Resource Management in Grid Computing
- Scheduling in Grid Computing
- Our Approach
- Policy-based Management System
- Heuristic Methodologies
- Conclusions

Introduction: Grid Computing (I)

- **GRID Computing:**
  - Grid Systems are interconnected collections of geographically distributed and heterogeneous resources harnessed together to satisfy different computational needs by 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

Introduction: Grid Computing (III)

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.
- Network Performance: Fault-tolerance, Reliability, Scalability, Flexibility and Persistence.
- Changes in Grid Services: WS-RF

Introduction: Grid Computing (IV)

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)

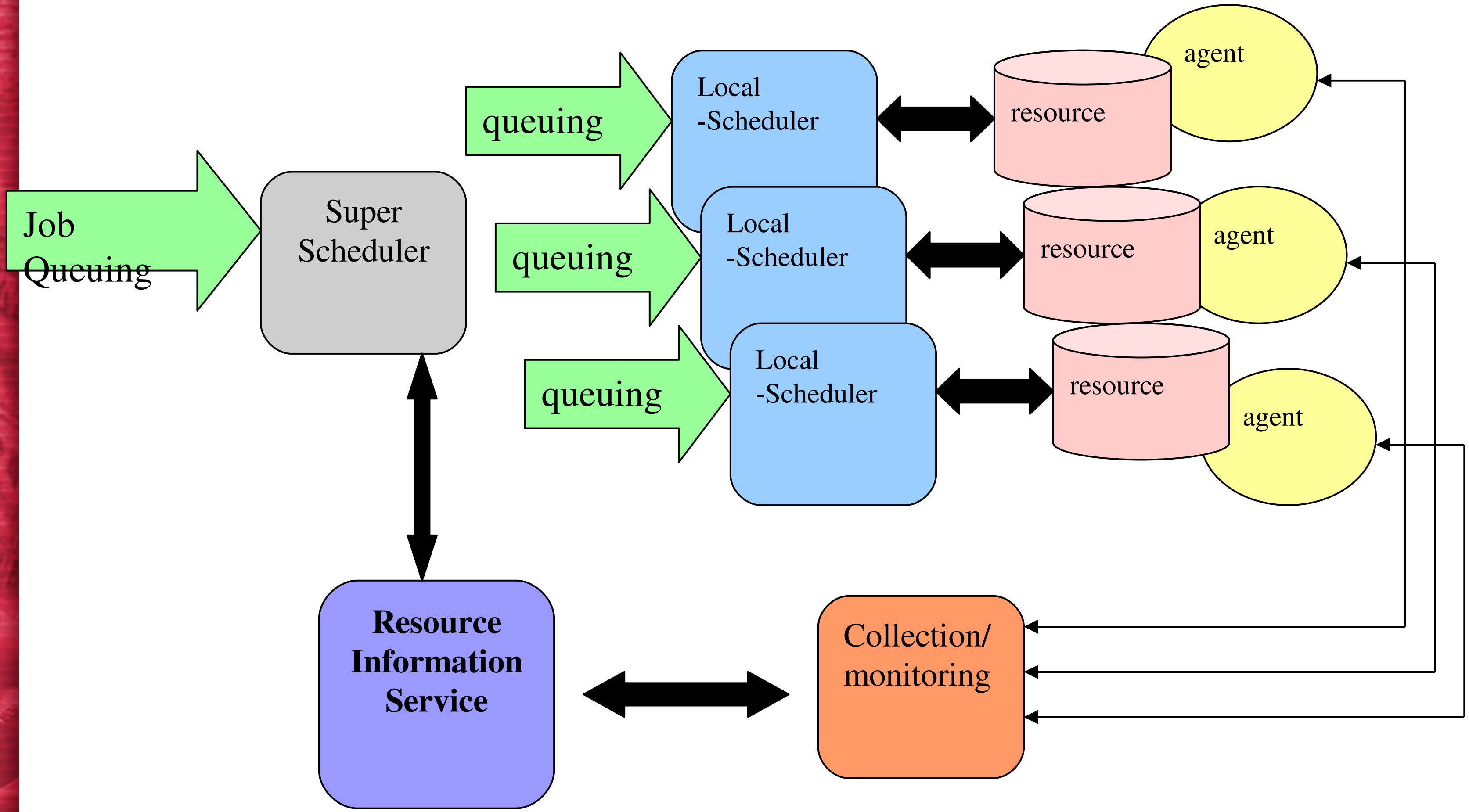
- **Basic Requirements:**
  - Members should be trustful and trustworthy.
  - Sharing is conditional.
  - Should be secure.
  - Sharing should be able to change dynamically.
  - Discovery and registering of resources.
  - Can be peer to peer or client/server.
  - Same resource may be used in different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

- **Resource Management Architecture:**



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)

Resource Management in Grid (V)

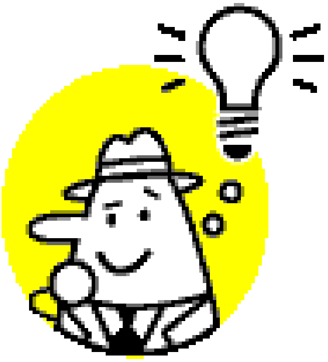
- **Difficulties in Resource Management:**
  - Autonomy
  - Heterogeneity
  - Dynamics

- **Grid Resources:**
  - Computing Power
  - Disk Space
  - Memory Space
  - Network Bandwidth
  - Software, etc.

Resource Management in Grid (VI)

- **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) (covers Phases 1–3)



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):**

$$\vec{N}_i = \begin{bmatrix} R_1 & R_2 & R_3 & \cdots & R_m \end{bmatrix}$$

$$\vec{G} = \begin{bmatrix} \vec{N}_1 \\ \vec{N}_2 \\ \vec{N}_3 \\ \vdots \\ \vec{N}_n \end{bmatrix}$$

$$\vec{G} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1m} \\ R_{21} & R_{22} & R_{23} & \cdots & R_{2m} \\ R_{31} & R_{32} & R_{33} & \cdots & R_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{n1} & R_{n2} & R_{n3} & \cdots & R_{nm} \end{bmatrix}$$

Traditional Scheduling (I)

- **Complexity Analysis:**

$$C = \binom{n}{k} = \left(\frac{n!}{k!(n-k)!}\right) \longrightarrow Ct = \sum_{k=1}^{n}\binom{n}{k} = \sum_{k=1}^{n}\frac{n!}{k!(n-k)!}$$

- **Binomial Analysis:**

$$(1+x)^n = \sum_{k=0}^{\alpha}\binom{n}{k}x^n$$

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

If x=1 $$(1+1)^n = \binom{n}{0} + \binom{n}{1}1 + \binom{n}{2}1 + \binom{n}{3}1 + \ldots + \binom{n}{k}1$$

$$2^n = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{k}$$

$$2^n - 1 = \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{k}$$

Complexity Equation

$$\sum_{k=1}^{n}\binom{n}{k} = 2^n - 1$$

$$Ct = \sum_{k=1}^{n}\binom{n}{k} = 2^n - 1$$

Approaches – Traditional Scheduling (II)

- **Algorithm Analysis:**

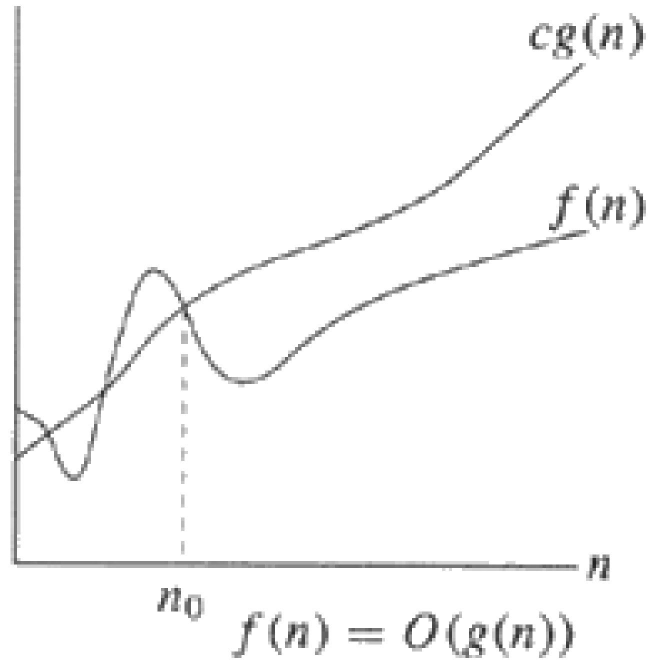
$$g(n) = 2^n - 1$$

$$f(x) = O(g(n))$$

$$f(x) = O(2^n - 1)$$

$$f(x) = O(2^n) - O(1)$$

$$\boxed{f(x) = O(2^n)}$$



Approaches – Traditional Scheduling (III)

- **Most Common Algorithms Complex Times:**

n	log(n)	n*log(n)	n^2	2^n
1	0	0	1	2
5	0,7	3,5	25	32
10	1,0	10	100	1024
20	1,3	26	400	1,048,576
50	1,7	85	2.500	$1.12590*10^{15}$
100	2,0	200	10.000	$1.26765*10^{30}$
200	2,3	460	40.000	…………
500	2,7	1.349	250.000	…………
1.000	3,0	3.000	$1*10^6$	…………

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

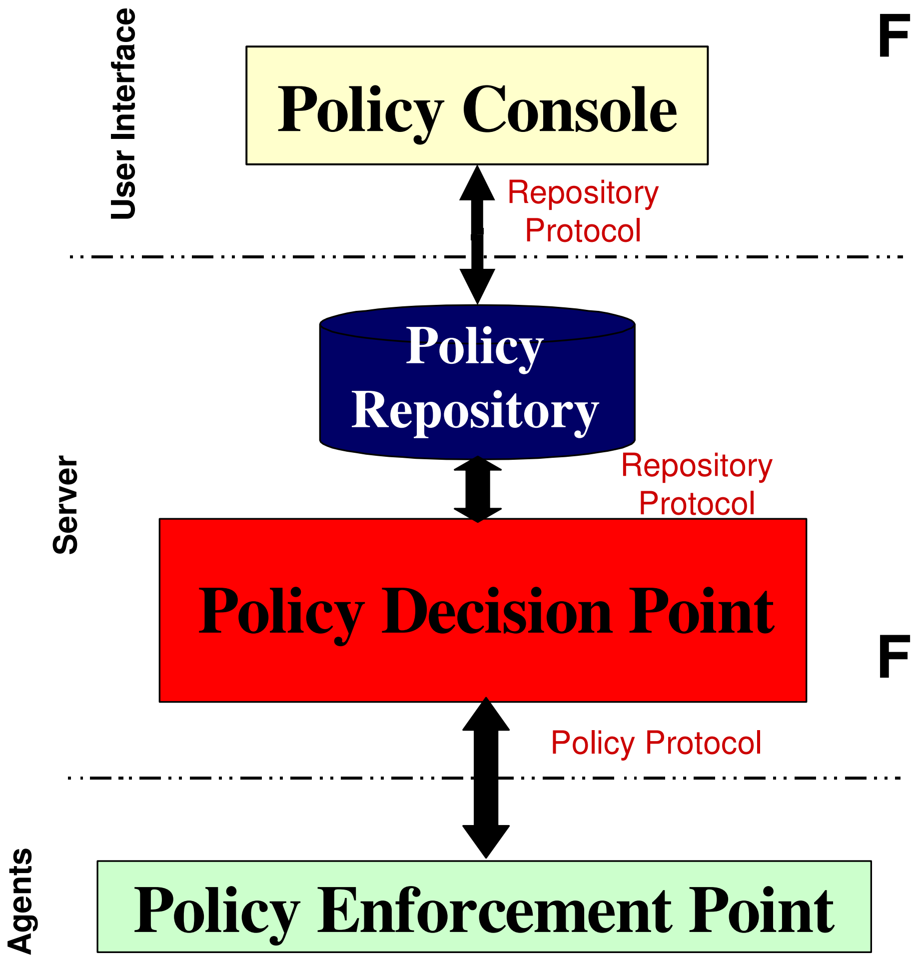
Policy-Based Management Approach

- **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

E. Magaña and J. Serrat, "QoS Aware Policy-Based Management Architecture for Service Grids", 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE/ETNGRID 2005). Linkoping University, Sweden, June 13-15, 2005.

Policy-Based Management System

- **IETF Policy framework and architecture:**



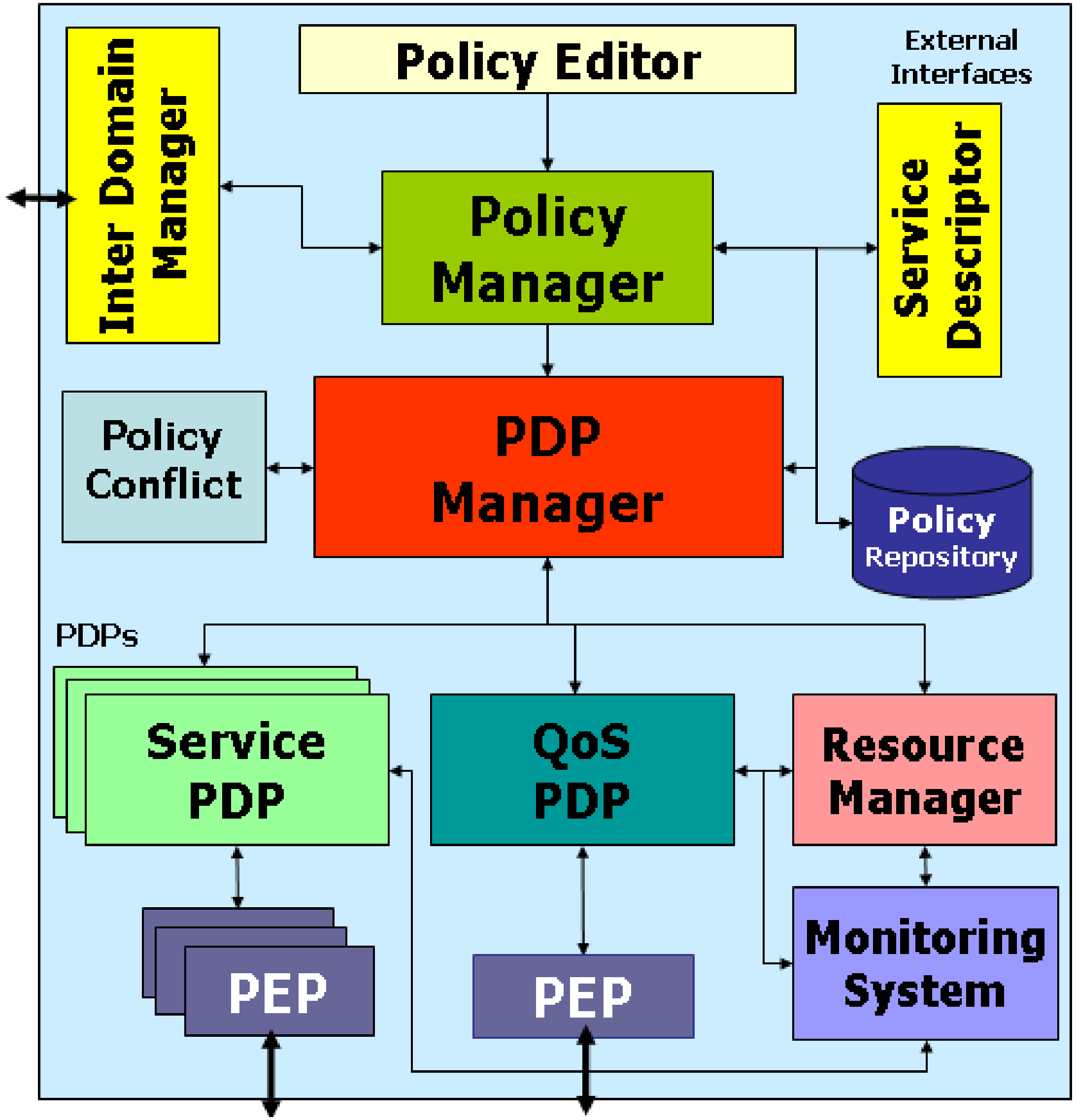
Functions of the PDP

- 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

PBMS Components Architecture



PBMS - High Level Functionality

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**

PBMS - High Level Functionality

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

Approaches - Heuristic Methodologies

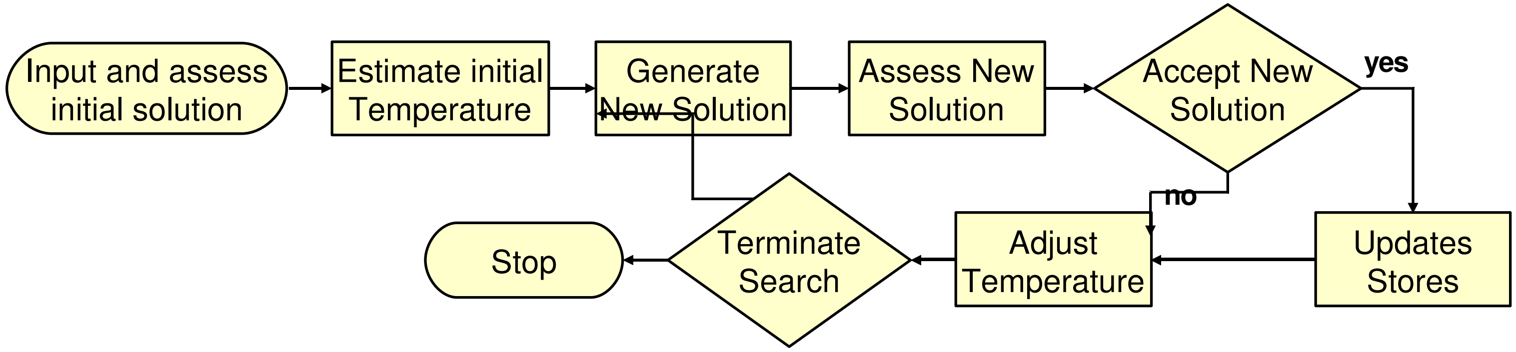
Simulated Annealing

Tabu Search

Evolutionary Algorithms

Approaches - Heuristic Methodologies (I)

- **Simulated Annealing:**
  - It has the ability to avoid becoming trapped at local minima.
  - The algorithm employs a random search.

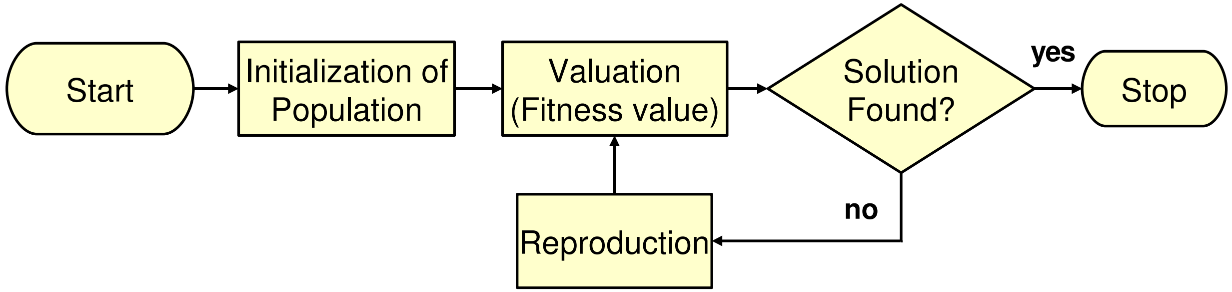


Approaches - Heuristic Methodologies (II)

- **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.

Approaches - Heuristic Methodologies (III)

- **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:

- C. Tsarouchis, S. Denazis, Chiho Kitahara, J. Vivero, E. Salamanca, **E. Magaña**, A. Galis, J. Mañas, Y. Carlinet, et.al., *"Policy-Based Management Architecture for Active and Programmable Networks"*, IEEE Network, May 2003, Vol.17, No.3.

Congresses and workshops:

- **E. Magaña** and J. Serrat, "QoS Aware Policy-Based Management Architecture for Service Grids", 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE/ETNGRID 2005). Linkoping University, Sweden, June 13-15, 2005.

- **E. Magaña** and J. Serrat, "QoS Aware Resource Management Architecture for OGSA Services Deployment", 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), 15-19 May. Nice - Acropolis, France. (Poster)

- **E. Magaña** and J. Serrat, "Adaptive Management Middleware for Grid Services and Applications Based on Policies", Workshop on Adaptive Grid Middleware (AGridM2004), Antibes Juan-les-Pins, France, September 29, 2004.

- **E. Magaña**, E. Salamanca, J. Serrat, *"A Proposal of Policy-Based System Architecture for Grid Services Management"*, Active and Programmable Grids Architectures and Components (APGAC'04), 7-9th June 2004, Kraków, Poland.

Publications (II)

Congresses and workshops:

- Y. Nikolakis, **E. Magaña**, M. Solarski, A. Tan, E. Salamanca, J. Serrat, C. Brou, A. Galis, *"A Policy-Based Management Architecture for Flexible Service Deployment in Active Networks"*, International Working Conference on Active Networks (IWAN03) Kyoto, Japan 2003.

Books:

- Co-author and editor of chapter 15 of *"Programmable Networks for IP Service Deployment"*, Artech House Books, May 2004, ISBN 1-58053-745-6

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 Metodologies
  - 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