Grid Computing Systems: A Survey and Taxonomy

Material for this lecture from:

“A Survey and Taxonomy of Resource Management Systems for Grid Computing Systems, K. Krauter, R. Buyya, M. Maheswaran, CS Technical Report.

Introduction

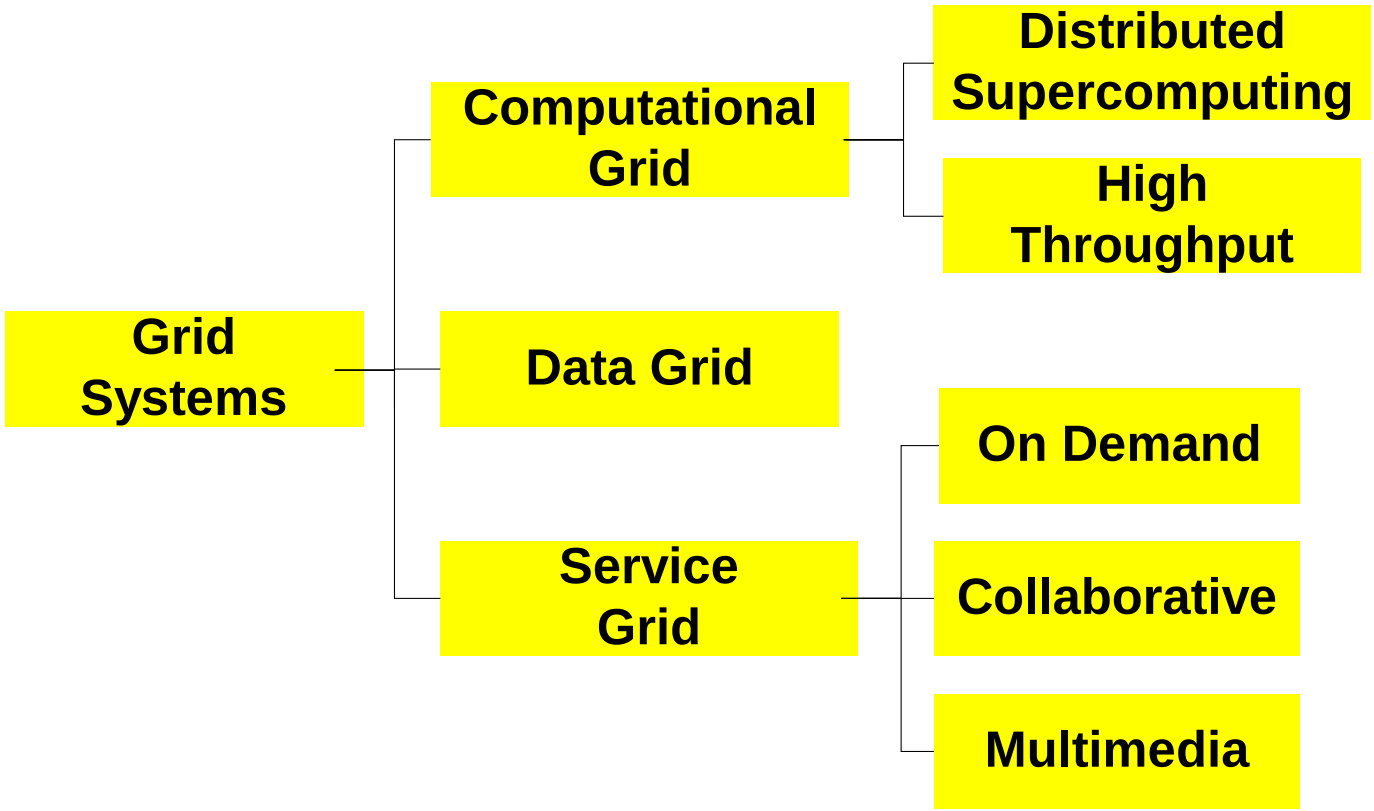
- Network Computing System:
  - a virtual system that is formed by machines and networks that agree to work together by pooling their resources
- Grid is a generalized network computing system that is supposed to scale to Internet levels and handle data and computation seamlessly

Introduction

- Resource management in Grid systems involves managing the basic elements
- Grid elements:
  - processing elements: uniprocessors, multiprocessors, handhelds, ..
  - storage elements:
  - network elements:

Introduction

- Grid systems can be classified depending on their usage:



Introduction

- Computational Grid:
  - denotes a system that has a higher aggregate capacity than any of its constituent machine
  - it can be further categorized based on how the overall capacity is used
- Distributed Supercomputing Grid:
  - executes the application in parallel on multiple machines to reduce the completion time of a job

Introduction

- Grand challenge problems typically require a distributed supercomputing Grid – one of the motivating factors of early Grid research – still driving in some quarters
- High throughput Grid:
  - increases the completion rate of a stream of jobs arriving in real time
  - ASIC or processor design verifications tests would be run on a high throughput Grid

Introduction

- Data Grid:
  - systems that provide an infrastructure for synthesizing new information from data repositories such as digital libraries or data warehouses
  - applications for these systems would be special purpose data mining that correlates information from multiple different high volume data sources

Introduction

- Service Grid:
  - systems that provide services that are not provided by any single machine
  - subdivided based on the type of service they provide
- collaborative Grid:
  - connects users and applications into collaborative workgroups -- enable real time interaction between humans and applications via a virtual workspace

Introduction

- Multimedia Grid:
  - provides an infrastructure for real time multimedia applications -- requires the support quality of service across multiple different machines whereas a multimedia application on a single dedicated machine can be deployed without QoS
  - synchronization between network and end-point QoS

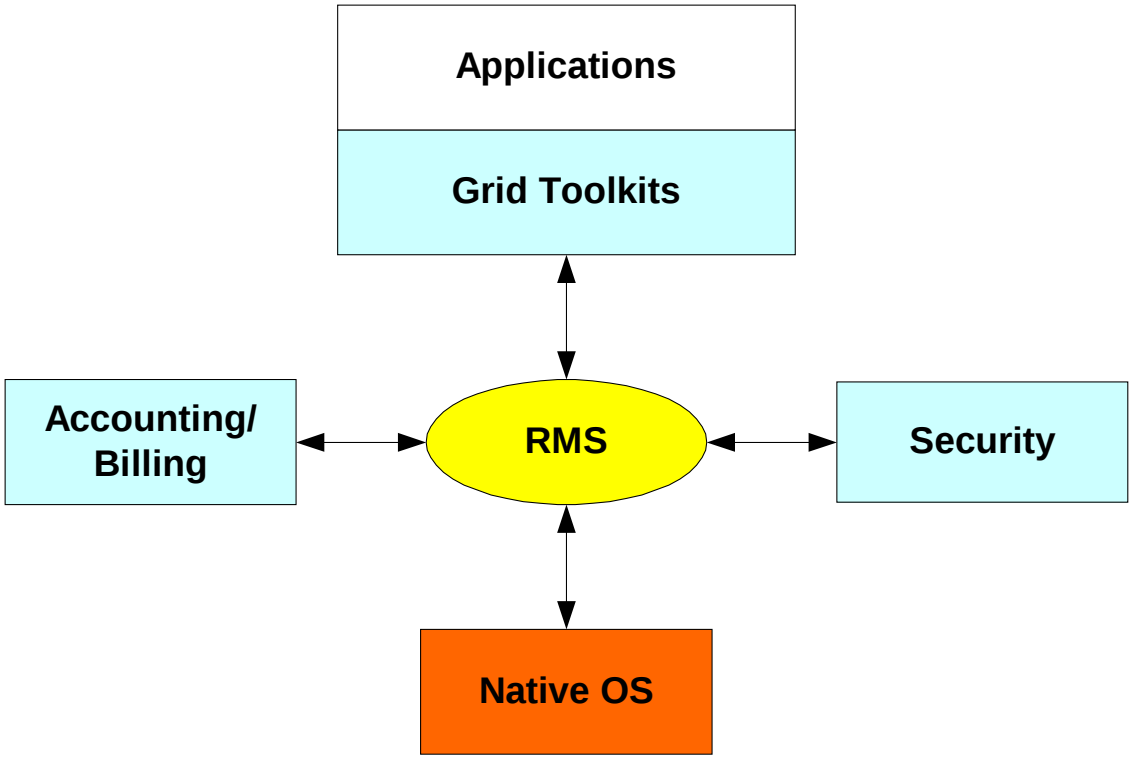
Introduction

- demand Grid:
  - category dynamically aggregates different resources to provide new services
  - data visualization workbench that allows a scientist to dynamically increase the fidelity of a simulation by allocating more machines to a simulation would be an example

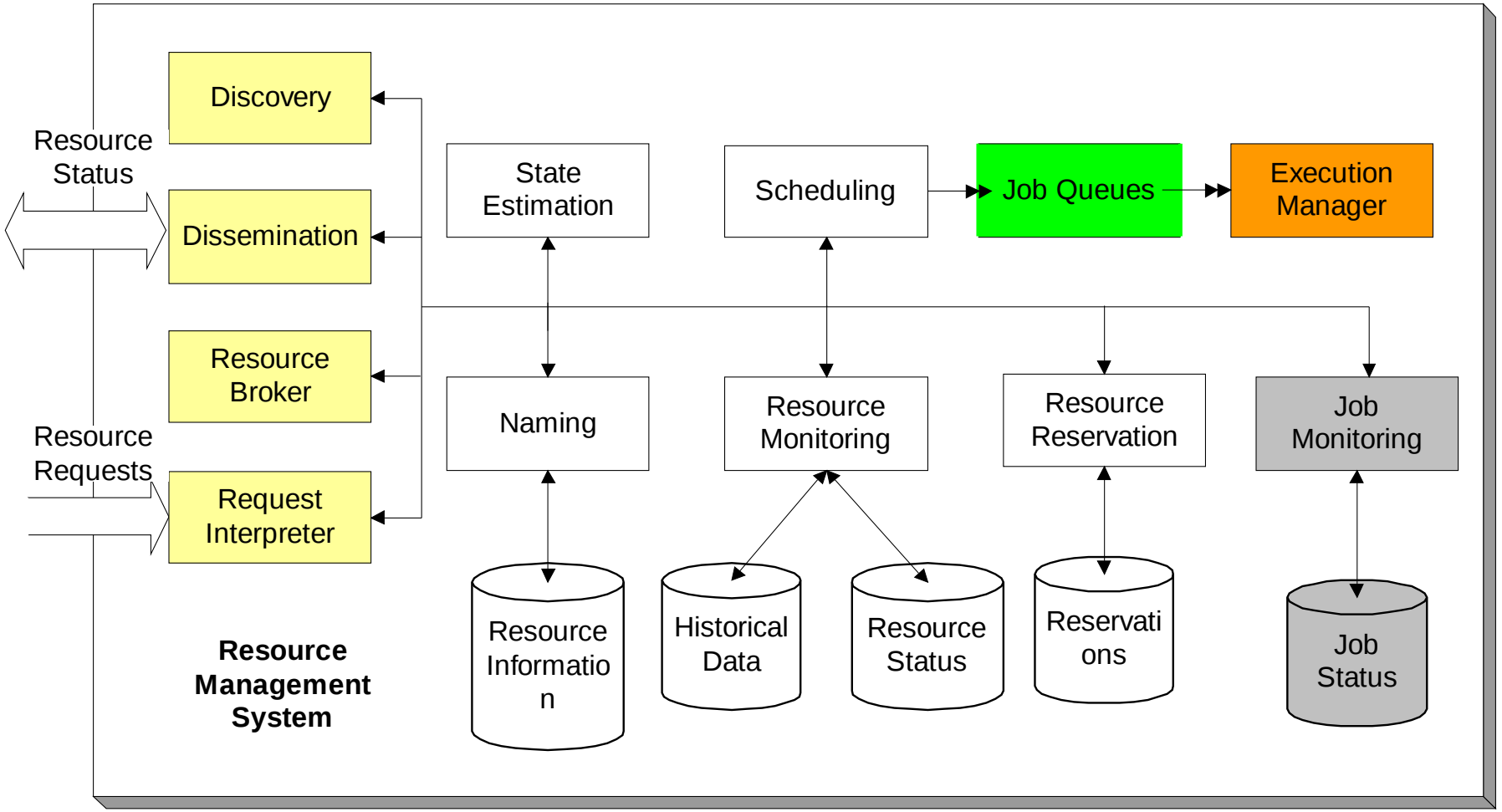
Abstract Model for a Grid RMS

- resource to refer to the entities that are managed by the RMS and jobs to refer to the entities that utilize resources
- architectures of existing resource management systems are quite different
- an abstract model of resource management systems provides a basis for a comparison between different RMS architectures

Abstract Model for a Grid RMS



Abstract Model for a Grid RMS



Abstract Model for a Grid RMS

- three different types of functional units:
  - application to RMS interfaces
  - RMS to native operating system and hardware environment
  - internal RMS functions
- application to RMS interfaces provides services that end-user or Grid applications use to carry out their work
- RMS to native operating system or hardware environment interface provides the mechanisms that the RMS uses to implement resource management functions

Abstract Model for a Grid RMS

- internal RMS functions identify the functions that are implemented as part of providing the resource management service
- resource dissemination, resource discovery, resource broker and request interpreter function provide the application to RMS interfaces
- RMS to native operating system interfaces are provided by the execution manager, job monitoring, and resource monitoring functions

Abstract Model for the RMS

- internal RMS functions are provided by the resource naming, scheduling, resource reservation and state estimation
- Resource information is distributed between machines in the Grid using a resource information protocol
- This protocol is implemented by the resource and dissemination functions
- Application resource requests are described using a resource description language or protocol that is parsed by the resource interpreter into the internal formats used by the other RMS functions

Abstract Model for the RMS

- The resource dissemination function and resource discovery function provide the means by which machines within the Grid are able to form a view of the available resources and their state
- resource naming function is an internal function that enforces the namespace rules for the resources and maintains a database of resource information
- The structure, content, and maintenance of the resource database are important differentiating factors between different RMS

Abstract Model for the RMS

- naming function interacts with the resource dissemination, discovery, and request interpreter so design choices in the namespace significantly affect the design and implementation of these other functions
- flat namespace would impose a significantly higher level of messaging between machines in the Grid even with extensive caching

Abstract Model for the RMS

- request interpreter accepts requests for resources, they are turned into jobs that scheduled and executed by the internal functions in the RMS
- job queue abstracts the implementation choices made for scheduling algorithms
- scheduling function examines the jobs queue and decides the state of the jobs in the queue

Abstract Model for the RMS

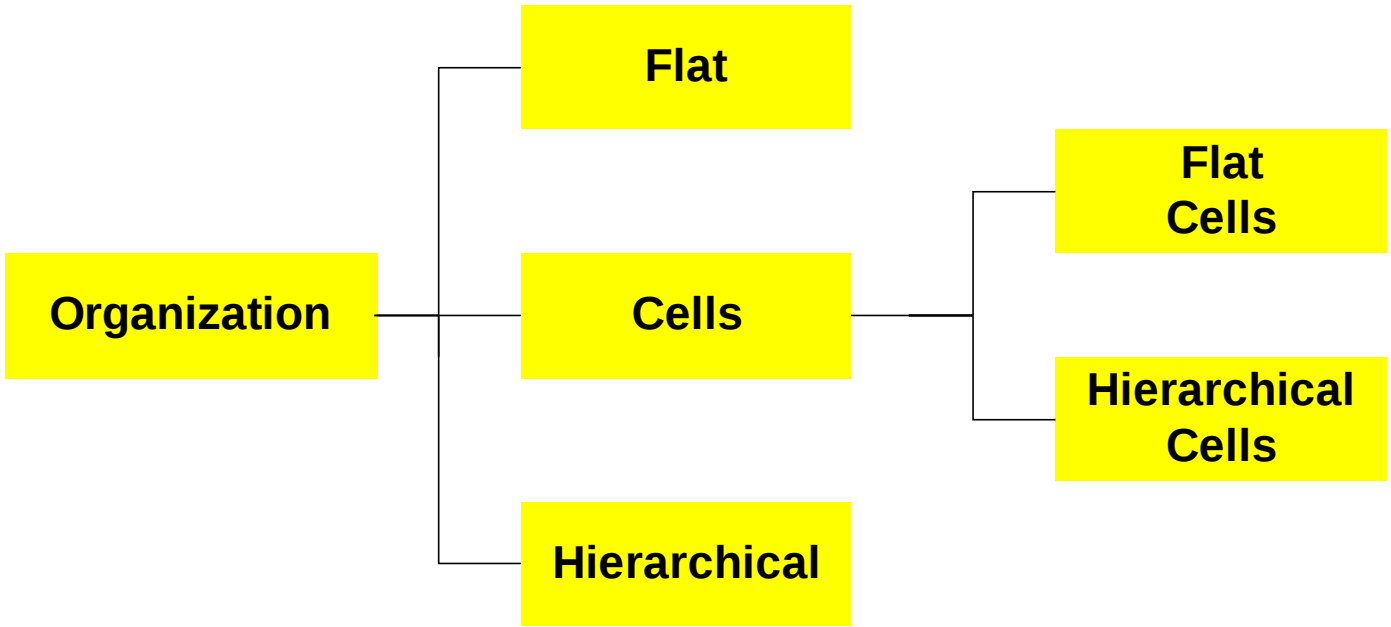
- The scheduling function uses the current information provided by the job status, resource status, and state estimation function to make its scheduling decisions
- scheduling function examines the jobs queue and decides the state of the jobs in the queue
- The scheduling function uses the current information provided by the job status, resource status, and state estimation function to make its scheduling decisions

Abstract Model for the RMS

- state estimation uses the current state information and a historical database to provide information to the scheduling algorithm
- execution manager does not control the execution of the jobs on a machine other than initiating the job using the native operating system services

Machine Organization

- Traditionally machines were organized as either in a centralized or decentralized organization
- Different classification is shown below



Machine Organization

- flat organization all machines can directly communicate with each other without going through an intermediary -- no current Grid systems use this type of organization but previous generation systems in a cluster environment used a flat organization
- hierarchal organization machines in same level can directly communicate with the machines directly above them or below them, or peer to them in the hierarchy -- most current Grid systems use this organization since it is scalable to some extent

Machine Organization

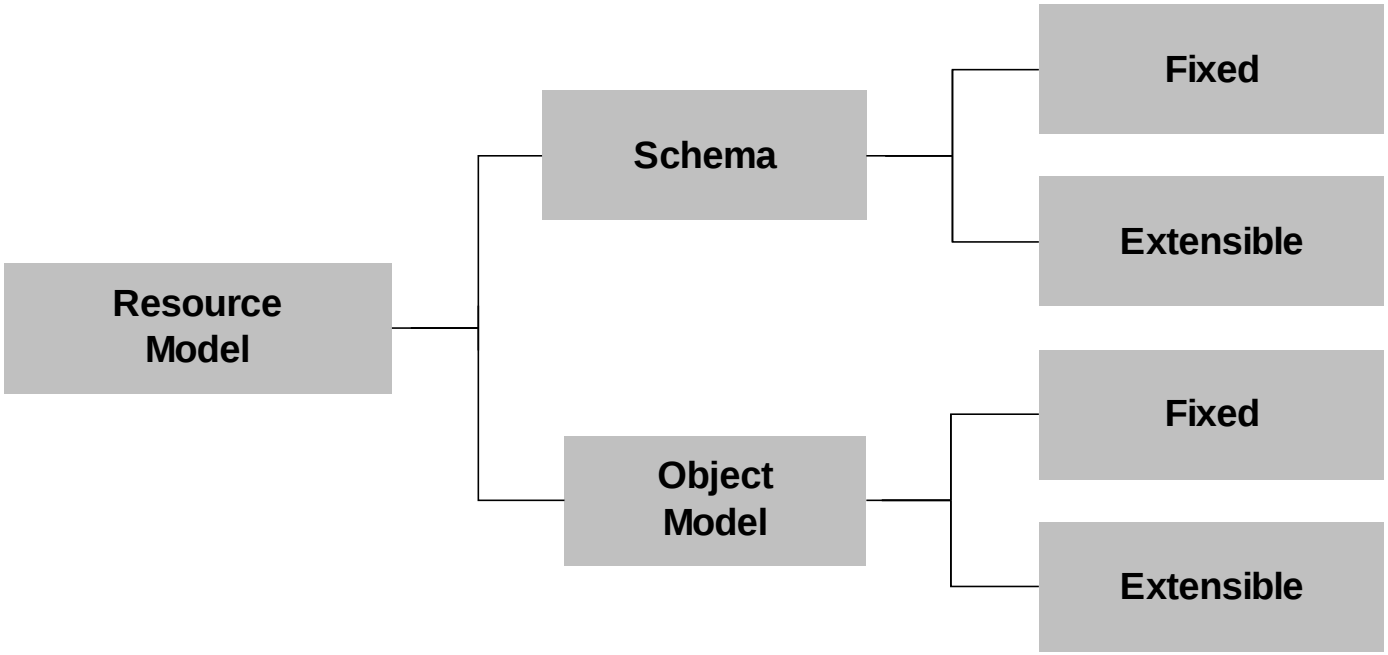
- cell structure, the machines within the cell communicate between themselves using flat organization
- designated machines within the cell function acts as boundary elements that are responsible for all communication outside the cell
- internal structure of a cell is not visible from another cell, only the boundary machines are

Machine Organization

- cells can be further organized in a flat or hierarchical structures
- Grid that has a flat cell structure has only one level of cells whereas a hierarchical cell structure can have cells that contain other cells

Resource Model

- resource model determines how applications and the RMS describe and manage Grid resources

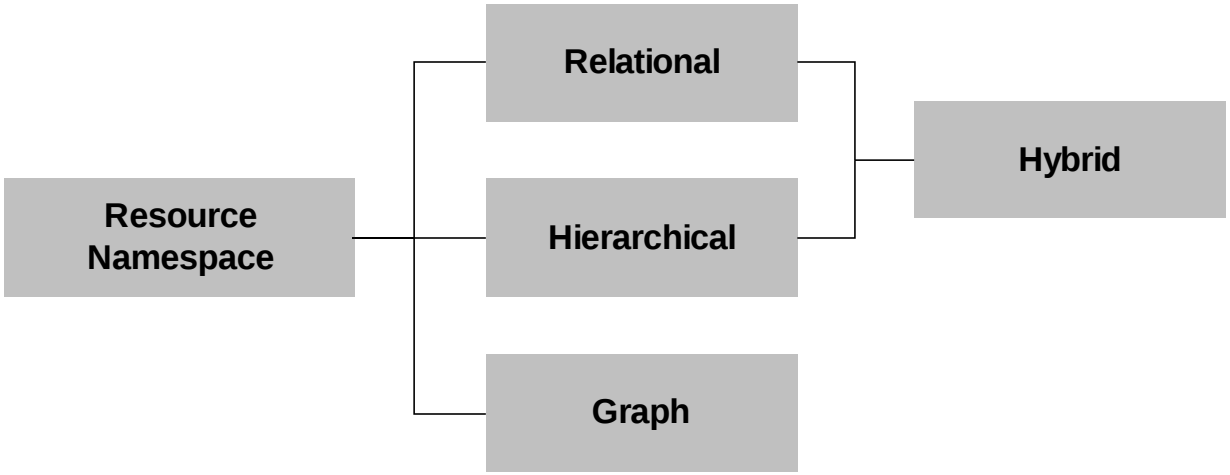


Resource Model

- the resource descriptions and resource status data store are integrated with their operations in an active scheme or if they function as passive data with operations being defined by other components in the RMS
- Condor “classad” approach using semi-structured data approach is in the extensible schema category

Resource Naming Model

- organization of the resource namespace influences the design of the resource management protocols and affects the discovery methods



Resource Naming Model

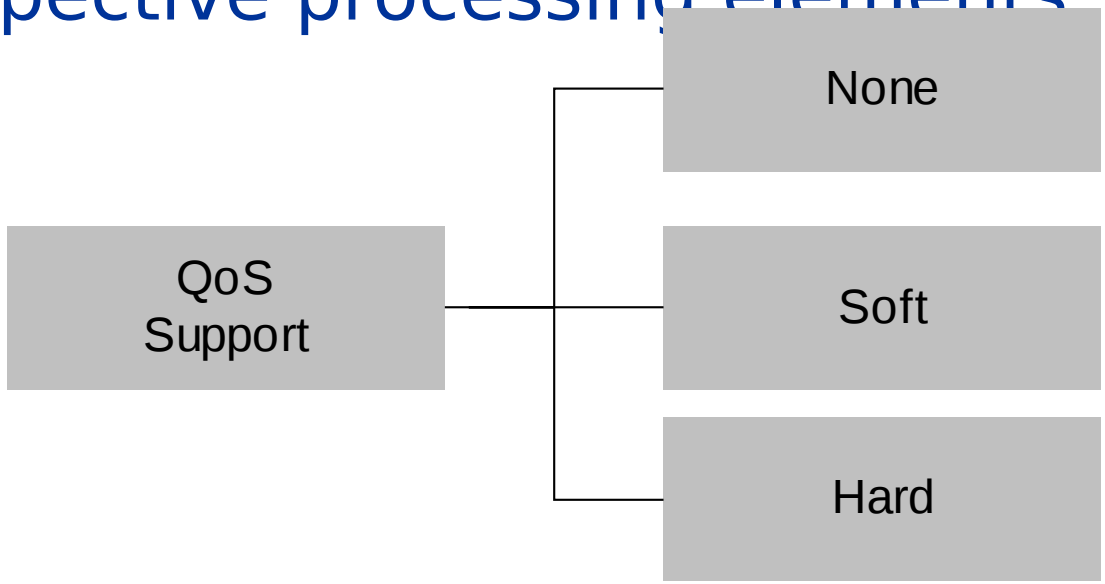
- flat namespace the use of agents to discover resources would require some sort of global strategy to partition the search space in order to reduce redundant searching of the same information
- relational namespace divides the resources into relations and uses concepts from relational databases to indicate relationships between tuples in different relations

Resource Naming Model

- hierarchical namespace divides the resources in the Grid into hierarchies
- graph-based namespace uses nodes and pointers where the nodes may or may not be complex entities

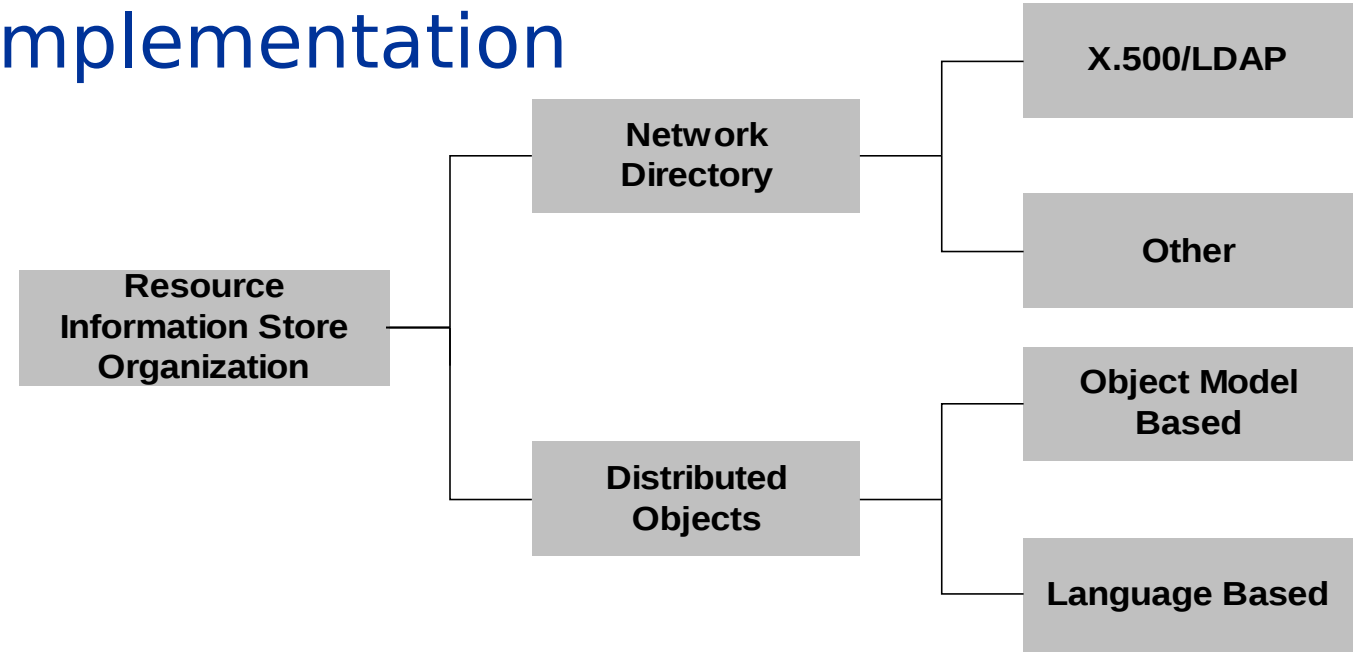
QoS Model

- inefficient to guarantee network QoS and not be able to ensure the application components that are communicating over this link have performance guarantees on their respective processing elements



Resource Info. Store Model

- organization determines the cost of implementing the resource management protocols since resource dissemination and discovery may be provided by the data store implementation

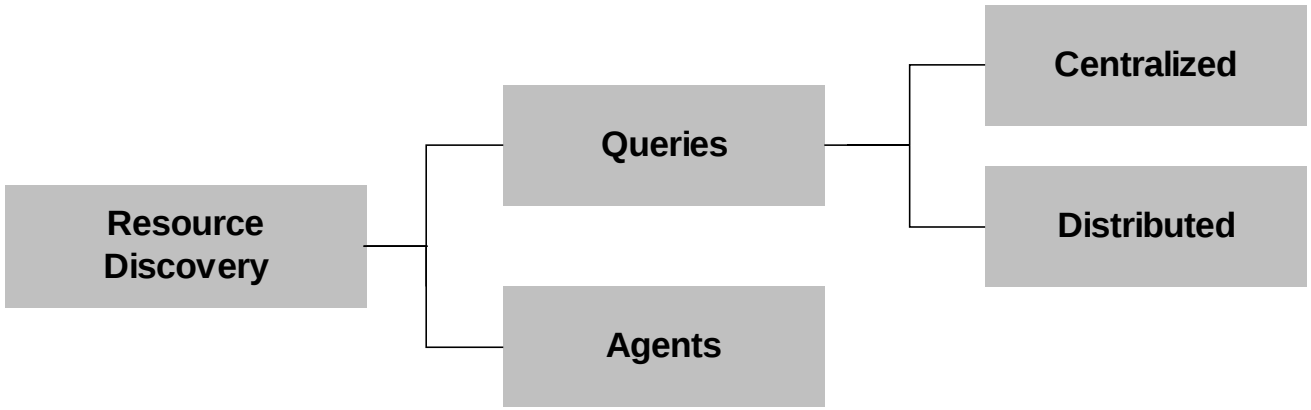


Resource Info. Store Model

- Distributed object data stores utilize persistent object services that are provided by language independent object models such as CORBA or a language based model such as that provided by persistent Java object implementations
- Network directories data stores are based on X.500/LDAP standards or utilize specialized distributed database implementation

Resource Discovery Model

- Network directory based systems mechanisms such as Globus MDS use parameterized queries that are sent across the network to the nearest directory, which uses its query engine to execute the query against the database contents



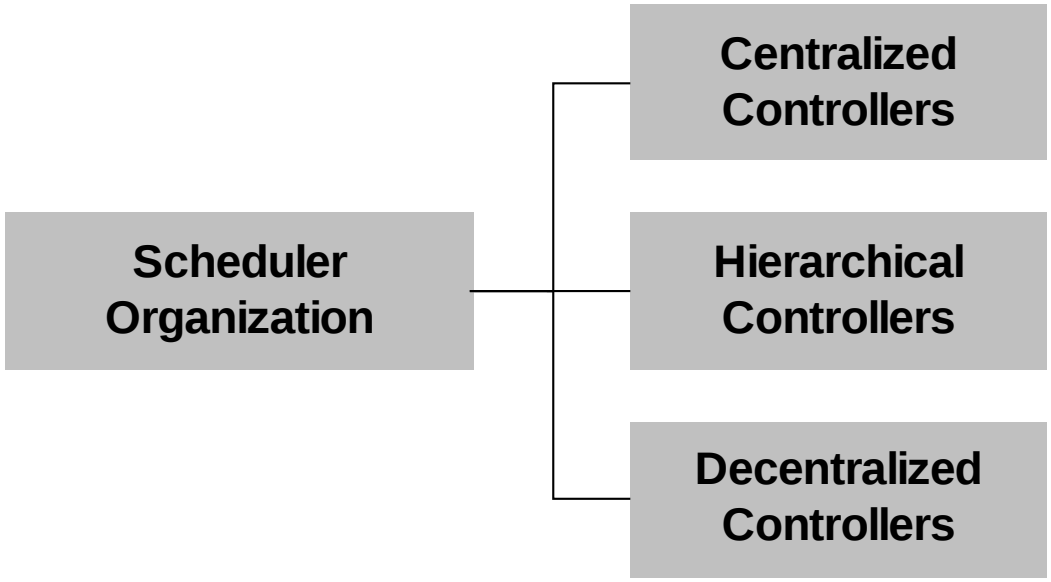
Resource Discovery Model

- Query based system are further characterized depending on whether the query is executed against a distributed database or a centralized database
- Agent based approaches send active code fragments across machines in the Grid that are interpreted locally on each machine

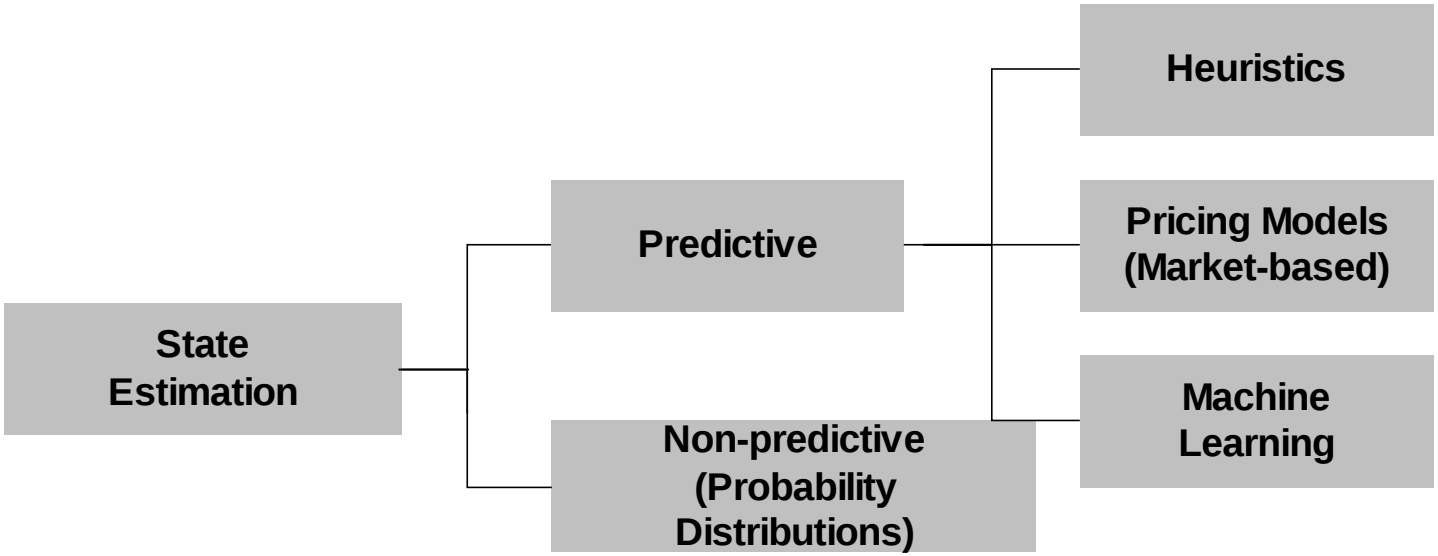
Resource Dissemination Model

- Universal awareness:
  - Each node has complete awareness of the entire system
- Neighborhood awareness:
  - Each node is aware of nodes that lie within a predefined network vicinity
- Distinctive awareness:
  - Each node is aware of nodes within a vicinity and are also aware of nodes outside that vicinity if they are “important”

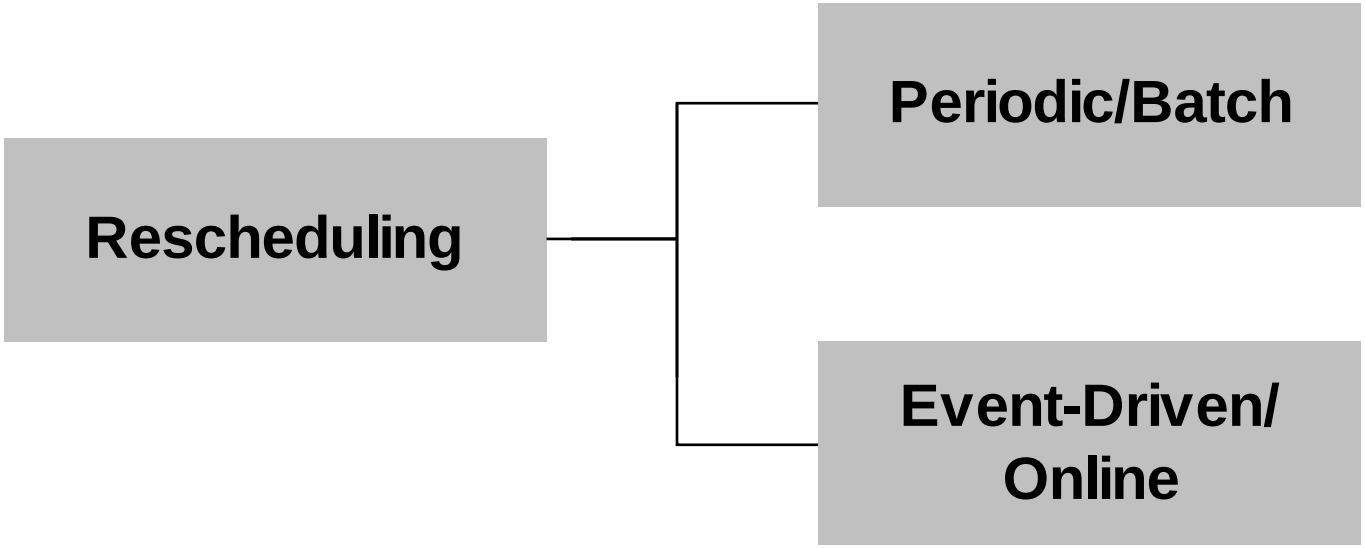
Scheduler Organization Model



State Estimation Model



Rescheduling Model



Scheduling Policy