

I T F ' C R R N E C V I Q P U

Based on our earlier discussion, we can align Grid Computing applications to have common needs, such as what is described in (but not limited to) the following items:

- Application partitioning that involves breaking the problem into discrete pieces
- Discovery and scheduling of tasks and workflow
- Data communications distributing the problem data where and when it is required
- Provisioning and distributing application codes to specific system nodes
- Results management assisting in the decision processes of the environment
- Autonomic features such as self-configuration, self-optimization, self-recovery, and self-management

Schedulers

Schedulers are types of applications responsible for the management of jobs, such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of tasks, data management, event correlation, and service-level management capabilities. These schedulers then form a hierarchical structure, with meta-schedulers that form the root and other lower level schedulers, while providing specific scheduling capabilities that form the leaves. These schedulers may be constructed with a local scheduler implementation approach for specific job execution, or another meta-scheduler or a cluster scheduler for parallel executions. Figure 1.2 shows this concept.

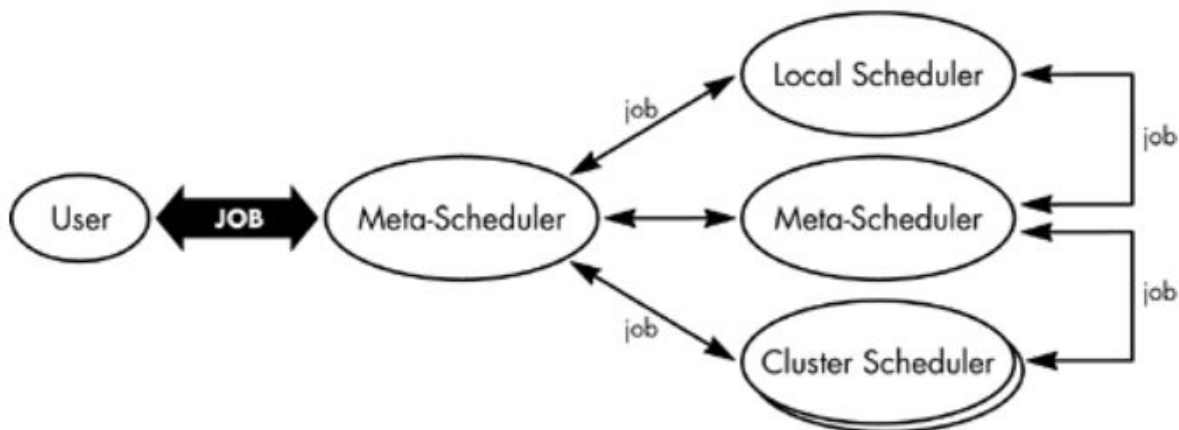


Figure 1.2. The scheduler hierarchy embodies local, meta-level, and cluster schedulers.

The jobs submitted to Grid Computing schedulers are evaluated based on their service-level requirements, and then allocated to the respective resources for execution. This will involve complex workflow management and data movement activities to occur on a regular basis. There are schedulers that must provide capabilities for areas such as (but not limited to):

- Advanced resource reservation
- Service-level agreement validation and enforcement
- Job and resource policy management and enforcement for best turnaround times within the allowable budget constraints
- Monitoring job executions and status
- Rescheduling and corrective actions of partial failover situations

Resource Broker

The resource broker provides pairing services between the service requester and the service provider. This pairing enables the selection of best available resources from the service provider for the execution of a specific task. These resource brokers collect information (e.g., resource availability, usage models, capabilities, and pricing information) from the respective resources, and use this information source in the pairing process. [Figure 1.3](#) illustrates the use of a resource broker for purposes of this discussion. This particular resource broker provides feedback to the users on the available resources. In general cases, the resource broker may select the suitable scheduler for the resource execution task, and collaborate with the scheduler to execute the task(s).

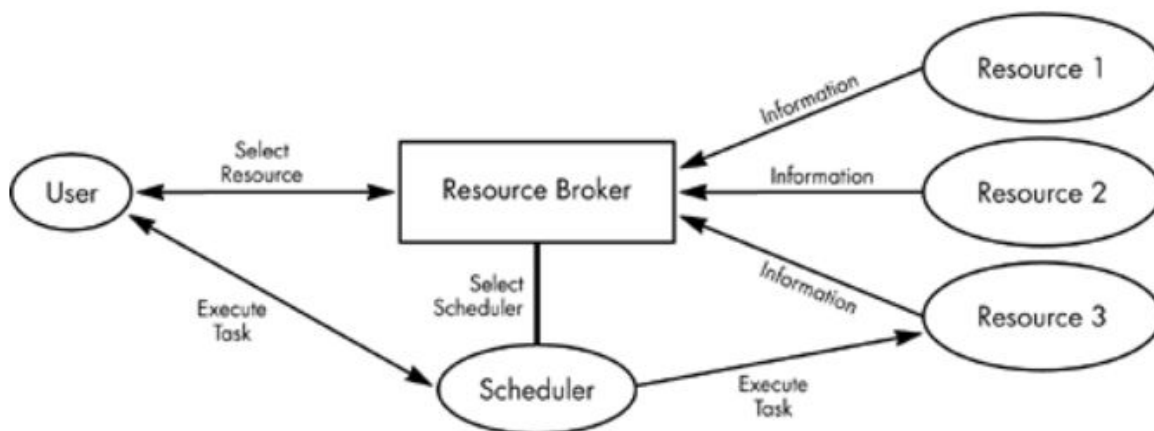


Figure 1.3. The resource broker collects information from the respective resources, and utilizes this information source in the pairing process.

The pairing process in a resource broker involves allocation and support functions such as:

- Allocating the appropriate resource or a combination of resources for the task execution
- Supporting users' deadline and budget constraints for scheduling optimizations

Load Balancing

The Grid Computing infrastructure load-balancing issues are concerned with the traditional load-balancing distribution of workload among the resources in a Grid Computing environment. This load-balancing feature must always be integrated into any system in order to avoid processing delays and over commitment of resources. These kinds of applications can be built in connection with schedulers and resource managers.

The workload can be pushed outbound to the resources, based on the availability state and/or resources, and can then pull the jobs from the schedulers depending on their availability. This level of load balancing involves partitioning of jobs, identifying the resources, and queueing of the jobs. There are cases when resource reservations might be required, as well as running multiple jobs in parallel. Another feature that might be of interest for load balancing is support for failure detection and management. These load distributors can redistribute the jobs to other resources if needed.

Grid Portals

Grid portals are similar to Web portals, in the sense they provide uniform access to the grid resources. For example, grid portals provide capabilities for Grid Computing resource authentication, remote resource access, scheduling capabilities, and monitoring status information. These kinds of portals help to alleviate the complexity of task management through customizable and personalized graphical interfaces for the users. This, in turn, alleviates the need for end users to have more domain knowledge than on the specific details of grid resource management.

Some examples of these grid portal capabilities are noted in the following list:

- Querying databases or LDAP servers for resource-specific information
- File transfer facilities such as file upload, download, integration with custom software, and so on
- Manage job through job status feedbacks
- Allocate the resources for the execution of specific tasks
- Security management
- Provide personalized solutions

Integrated Solutions

Many of the global industry sectors have witnessed the emergence of a number of integrated grid application solutions in the last few years. This book focuses on this success factor.

These integrated solutions are a combination of the existing advanced middleware and application functionalities, combined to provide more coherent and high performance results across the Grid Computing environment.

Integrated Grid Computing solutions will have more enhanced features to support more complex utilization of grids such as coordinated and optimized resource sharing, enhanced security management, cost optimizations, and areas yet to be explored. It is straightforward to see that these integrated solutions in both the commercial and noncommercial worlds sustain high values and significant cost reductions. Grid applications can achieve levels of flexibility utilizing infrastructures provided by application and middleware frameworks.

Source : <http://elearningatria.files.wordpress.com/2013/10/ise-viii-grid-computing-06is845-notes.pdf>