

Enhanced Utilization and Allocation of Grid Resources

Grid computing systems comprise of several resource nodes in a networked system. Each resource node may further comprise of a single machine or several machines forming another network. By utilizing the computing power of all these machines in this grid computing system, calculations that usually require super computers may be realized by the combined computing capability of all the machines in the system.

Presently, grid computing has become one of the effective ways for the sharing of resources and computing power over the internet. Several resource nodes of a grid computing system may be able to utilize each other's spare or idle computing power. An example of such sharing is when a resource node is located in a differing time zone, say at night where the computing power of the machines in that node are idle. Another resource node currently requiring extra computing power may be able to send jobs to this idle node to speed up the calculations or even to free up computing power in its own node for more crucial applications. Furthermore, present methods for assigning jobs to nodes are mainly based on a round robin queuing system that is disadvantageously arbitrary and takes no consideration of the characteristics of the idle or available resource nodes.

Our proposed solution thus provides a method and system for monitoring and managing grid resources in a grid computing environment with a plurality of resource nodes and at least one resource broker/supervisor node. The resource broker serves to retrieve information stored in an attribute repository about the resource nodes. All the resource nodes would send and update this information on the attributes and status of the resource nodes to the attribute repository periodically. The step of job brokering starts with the step of checking the attribute repository for attributes and status for all resource nodes in the grid computing environment. The information retrieved by the resource broker from the attribute repository may be attributes such as CPU utilization, Random Access memory (RAM) capacity or Virtual Memory (VM) Capacity. The status of the resource nodes may also be retrieved.

Next, the step of receiving a first priority attribute and a second priority attribute for the matching of resource nodes that may be best suitable for performing the job. These first and second priority attributes may be decided by a user based on the requirements of the job. For example, the job submitted may require very large RAM capacity for handling large amounts of variable data, the first priority attribute will then be RAM. The second priority may then be CPU utilization, depending on the user's requirements for the job. The step of matching available resource nodes corresponding to the first and second priority attributes is then performed. Following which the step of recommending the matching resource node to the job submitter is then determined.

Summary

This paper describes an effective mean of determining the optimum resource for job execution in a Grid environment that will further improve the performance of job computation which Grid Computing offers. In summary, we proposed a solution with the following features :

- An improved method for Grid job submission, monitoring and management
- Monitoring Grid processes and switching between Grid resources in the event of a process failure
- Job brokering to determine the "Best match" grid resource to perform the job request
- Distributing jobs to available resources whereby the number of jobs exceeded the number of resources
- Notification in the event of a process failure through SMS, Email and messaging.

Primary author: Mr TAN, Nam Beng (Nanyang Polytechnic)

Presenter: Mr TAN, Nam Beng (Nanyang Polytechnic)

Track Classification: Massively Distributed Computing and Citizen Sciences