Resource Selection in Computational Grid Based on User QoS and Trust

G.Kavitha † and V.Sankaranarayanan ††

†Research scholar, B.S.A. Crescent Engineering College, Vandalur, Chennai,Tamilnadu,India
††Professor & Director, B.S.A. Crescent Engineering College, Vandalur, Chennai,Tamilnadu,India

Abstract
Grid computing provides a virtual framework for controlled sharing of resources across institutional boundaries. As the resources from different administrative domains participate in task execution, selection of appropriate resources is a challenging task. We propose an efficient method of resource selection for distributed grid environments that is based on execution trust of a resource and the Quality of Service defined for the user. The resources selected are from the trusted list of resources that satisfy the user requirements of computation power for job execution with shorter response time and budget constraints imposed by the user. The selection strategy is based on user QOS parameters which improve the performance of jobs submitted to the grid and the utilization of resources that participate in the Grid. Simulation results show that the proposed model selects the appropriate resources among the available trustworthy resources with an improved power – cost ratio and maximizes the reliability of the resource.

Keywords
Grid computing, resource selection, trust, QOS parameter, computation power, response time, reliability.

1. Introduction

Grid Computing [1] [2] allows computing, storage and other resources that are geographically distributed and belong to different administrative domains to participate in a virtual organisation (VO). Grid computing provides the basic infrastructure required for sharing diverse sets of resources including desktops, computational clusters, super computers, storage, data, sensors, applications and online scientific instruments. Resources are virtualised so that members of the VO can execute their application on coordinated resources obtained by specifying the requirements, rather than specifying the individual resource to be used. Selection of resources for a user's task in such a heterogeneous environment is fundamentally important.

Grid is a dynamic environment where the state and availability of the resource changes from time to time. Hence, the resources are to be allocated to clients based on the dynamic status information of the resource.

Trust is subjective and it reveals the behaviour of the resource to some extent. A highly trusted resource may be overloaded and may not provide the needed compute cycles within the required time limit required for compute intensive client jobs. Therefore, selecting and allocating a resource based only on trust is not the efficient method of resource selection strategy. The proposed method identifies an appropriate resource that satisfies the user requirements of trust, time, cost limits and maximizes resource utilization in the grid by harnessing the maximum power that is delivered by the resources. The basic selection criterion from the user's view is whether adequate computational power is available for executing the client's application at an affordable cost.

Many earlier research works have addressed the problem of resource selection in heterogeneous grid environments. The focus is on allocating a client job on local schedulers of resources based on cost and load of the local resources. The proposed work differs from the previous work by a way of job allocation to a Meta scheduler of the grid considering the dynamic resource state information along with security requirements for the user's application. The security part is implemented using quantitative execution trust about the resource and has been addressed in detail in our previous research paper [5].

The present work is targeted to provide an optimal and performance aware resource selection algorithm (section 3) that identifies a suitable resource to complete the job with a minimum response time and leverage the compute power of the selected resource effectively thus improving the resource utilization in a grid environment (section 4). The proposed strategy maximizes the benefits of clients and resource providers maintaining the reliability of the client's task.

2. The Resource Selection Model

The resource selection model is developed to select a suitable and trustworthy resource among the available resources within a virtual organisation.

2.1. The Proposed System Model

Manuscript received March 5, 2011
Manuscript revised March 20, 2011

A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities. A computational grid consists of heterogeneous resources from different administrative domains. Every resource is characterized by its processor speed, processor configuration and memory size. Resources can execute multiple applications submitted by clients at the same time. Therefore, selection of appropriate resources for the client jobs is fundamentally important.

The proposed system is responsible for selection of resources such that the task execution meets the deadline with minimized cost availing the maximum computation power. The model works on user's behalf and completes the assigned task within the specified time duration and cost. For executing an application in a Grid environment the following parameters are estimated.

a. Resource requirements of a given application.
b. Computation capability of the resources.
c. Present state of the resource.
d. Cost of resource usage.
e. User preference based on trust.
f. Reliability of the resource.

The objective of the resource selection algorithm is to select an optimum resource with regard to the trust and QoS metrics requested by the user and offered by the resource provider. The user jobs are mapped to the resources that offer higher level of QoS in terms of available computational power, execution time and cost. The jobs submitted to the selected resources complete the job with an improved power-cost ratio and higher resource utilization. The proposed system model is based on the following assumptions:

a. Jobs are computation intensive.
b. Resources are allotted within a VO.
c. FCFS scheduling of jobs.
d. Jobs are not distributed over multiple resources.
e. Non pre-emptive task execution.

2.2 System Parameters

For every resource participating in the Grid the following parameters are defined.

1. *Job Parameters*: A user job is characterized by its Job ID, Workload, Execution deadline, Minimum trust level and Cost limit.
2. *Resource Capacity*: The computation capacity that can be offered by the resource.This is calculated as the number of processors of a particular type and the CPU speed of the processor. The resource capacity is specified in MIPS.
3. *Resource Status Information*: Real time status information about the state of resource, available computation power, job response time.
4. *Network Parameter*: Speed of network bandwidth link connection.
5. *Performance Metrics*: The performance metrics evaluated are power-cost ratio, reliability of resources and resource utilization in the Grid.

3. The Proposed Computational Grid Architecture

The proposed Computational Grid architecture is shown in fig.1. It is a three tier architecture with physical layer as the lower tier. The physical layer consists of resources from different physical organisations. The resources participate in one or more Virtual Organizations (VO) of the Grid. Each resource has different resource characteristics and a resource is a heterogeneous cluster that has multiple compute nodes for performing compute intensive tasks. A VO defines the available resources and the rules for accessing the resources.

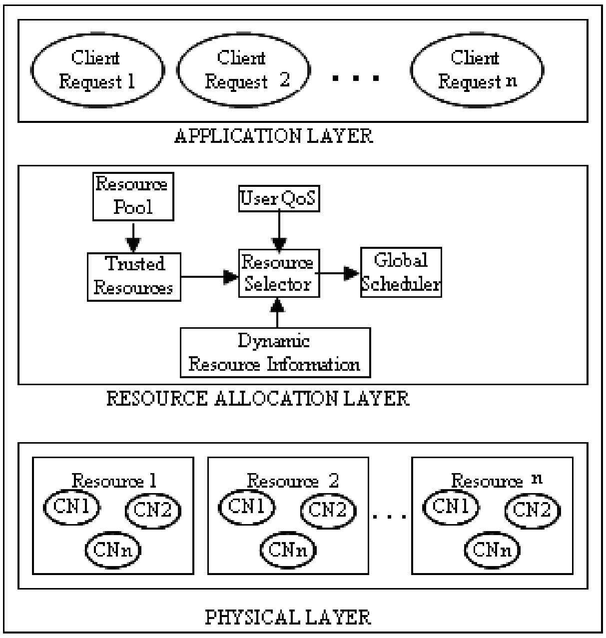

Figure.1. The Proposed Computational Grid Architecture

The middle tier is the Resource Allocation layer that performs selection of appropriate resources for a client's job. It has a resource pool, resource selector and a Global scheduler. The resource pool consists of the list of resources that participate in a computational grid. The list of resources that are trusted for performing a computation is found by a trust evaluation process briefly described in our previous work [5]. The client's job requests are given in the User Application layer. There exists a high variability in the jobs submitted by every client. We assume that one resource in a VO can execute the client's job.

In a Computational Grid, the status and capacity of the resources vary dynamically from time to time. Hence job submission should be based on trust worthiness of the resource along with its status, availability, computation capacity and job queue length. The resource attributes are matched with the user QoS requirements and if found satisfied the task is dispatched and scheduled on that resource.

3.1 Resource Selection Algorithm

The resource selection algorithm is derived upon the trustworthiness of the resource and the capability of the trusted resource. The trustworthiness is evaluated by the subjective and objective trust values about that resource [5] from the past history of transactions and present environment conditions. The capability is with respect to computation power, response time of jobs and cost involved in completing the client's task. The actual process of resource selection is detailed in figure 2.

The resource pool contains the set of resources derived from different physical organizations. A resource is a cluster with heterogeneous compute nodes that differ in hardware architectures, operating systems, computing power and resource capacity. The client's request contains the job descriptions namely, minimum trust level required, Job ID, size of the job, time and budget limits. Considering the job requirements, a resource is selected from the resource pool that has more than the minimum trust level given by the user. This provides entry level security to the client job submitted to the resource. Trust is evaluated based on the subjective and objective parameters and a quantitative value of execution trust is obtained. The overall trust value (OTV) is given by,

$$OTV = \alpha * SBT + \beta * OBT \qquad (1)$$

The values of the subjective trust (SBT) and the objective trust (OBT) are weighed equally to obtain the quantitative trust value. The weights α & β are assigned a value of 0.5 which may be varied depending on user requirements and OTV varies between 0 and 1.

After selecting a trustworthy resource from equation (1), we evaluate whether the selected resource has the required computation power to execute the client's task. The computation power is measured as the aggregate CPU speed across all computing nodes with in a resource Ri. The other factors that affect CPU performance apart from CPU speed are I/O bandwidth and cache architecture. In this work we consider the computation power of a processor is based upon CPU speed.

Input : Resources Ri, minimum trust level, time limit, cost limit, job Ji.

1. for every resource Ri,
   a. calculate Overall Trust Value (OTV) given the values of Subjective Trust (SBT) ,Objective Trust(OBT) , weights α and β.
   b. for each client's job request, the Minimum Trust Level (MTL) required for the job is given by the client.
      if (OTV >= MTL)
        select the resources and add it in the list L

2. for every resource Ri in L, the following values are ascertained based on the current conditions of grid.
   a. calculate the Available Computation Power (ACP) with the current values of Overall Computation Power (OVCP) and Occupied Computation Power (OCP).
   b. find the Expected ResponseTime (ERT) of Ji on Ri.
      Rank the resources based on the minimum response time and add it in Ranked List.

3. for every resource Ri in the ranked list,
   if(Expected Response Time <= Expected Execution Time) AND (Expected Completion Time <= Time Deadline)
     compute the Expected Cost of Resource (ECR) for the job Ji on resource Ri.
   else
     select the next resource from the ranked list and repeat step 3.

   If none of the resources in the list match the client's requirements, the job is put in wait state for a time duration T and repeat step 3.

4. Select the resource that has the minimum ECR among

Figure.2. Resource Selection Algorithm

The Overall Compute Power (OVCP) is the cumulative computation power that is provided by a resource Ri. The proposed method finds the Available Computation Power (ACP) of a resource and if required power is available then the resource is selected in a list L. The Available Computation Power for a resource is calculated as the difference between the Overall Compute Power of the resource and the Occupied Compute Power (OCP) of the resource at the given time instant.

This dynamic information given by the system is an important factor that helps to select the appropriate resources and hence the job migration is avoided. The computation power is expressed in MIPS (Million Instructions Per Second).

The Overall Compute Power of a resource Ri is given as,

$$OVCP_R = \sum_{j=1}^{m} N_j * Speed_j \quad (2)$$

Where, N_j represents the number of processors of a particular type j. $Speed_j$ is the CPU speed that indicates the number of instructions per second the CPU can process. Speed of every processor is represented by MIPS.

The OVCP specifies the total compute power that can be offered by the resource. Before allocating a task to a resource it is necessary to find the available compute power to avoid the run time failures due to insufficient computation power. The Available Computation Power of Ri is expressed as,

$$ACP_R = OVCP_R - OCP_R \quad (3)$$

The Occupied Compute Power is expressed as the sum of the computation power (CP) availed by the 'n-1' jobs that are currently in process at the resource Ri.

$$OCP_R = \sum_{i=1}^{n-1} CP_i \quad (4)$$

The second QoS parameter that is important in resource selection is the Expected Completion Time (ECT) for the job and is calculated prior to job submission. The ECT for the job involves two components as the Expected Response Time (ERT) of the job and the Expected Execution Time (EET) of the job.

$$ECT_n = ERT_n + EET_n \quad (5)$$

In equation 5, Expected Completion Time is computed for n^{th} job by considering the response time and expected execution time for that job. The job execution time in a distributed environment can be predicted using the weibull distribution. The weibull random variables are commonly used to model the runtime of the jobs as it approximates well to the actual run time of jobs [16].

In computational grid, the response time of a client's request is defined as the time taken by the resource to react or respond to the client's request. As the response time will be known only at the time of job submission, we consider the Expected Response Time (ERT) of the job prior to job submission. The expected response time is one of the factors that determine the completion time of the job. If the value of ERT is very high than the execution time of the job then the algorithm selects another appropriate resource that completes the task in specified time limits. The Expected Response Time for the n^{th} job is calculated as,

$$ERT_n = \sum_{i=1}^{n-1} AET_i + PET \quad (6)$$

Where, AETi represents the Approximated Execution Time (AET) for the i^{th} task present in the queue and PET represents the Pending Execution Time (PET) for the task that is currently running on the resource Ri. The computed value of expected completion time of job is compared with the time deadline provided by the user job. If the completion time is within the given time limit, then the resource is added in ranked list. The trusted resource after being evaluated for the computation power requirement and expected completion time of the job, now passes to the third QoS parameter which is the cost limit specified by user for task execution on a resource Ri.

The resource providers fix up the cost based on the demand for their resources in the grid. The cost is a dynamic factor and is changed by the resource provider from transaction to transaction. Hence cost of using a resource Ri is specified by the resource provider on joining the VO. The total cost spent on executing a job on the selected resource is calculated with respect to the amount of computation power used and the time during which the processors were engaged in performing the job.

Hence, whenever a job is to be submitted to a particular resource Ri, it is worthwhile to calculate the expected cost of execution based on the number of processors required and time spent on each processor. This evaluated cost parameter if found satisfactory for the user, allows the job to tbe submitted on that resource. The Expected Cost of Resource (ECR) for executing a job on Ri is formulated as,

$$ECR_i = \sum_{j=1}^{m} N_j * Hrs * Cost \quad (7)$$

Where, N_j represents the number of processors of type j and Hrs represent the number of hours for which these processors will be used and cost represents the usage cost of the processor per hour.

The Resource that satisfies all the mentioned parameters of Available Computation power, Expected Completion Time and Expected Cost of Resource is therefore selected for a client's task and the job is submitted. Simulation results show the satisfiability of the performance metrics as power-cost ratio, Resource Utilization and Reliability of jobs submitted to these selected resources.

3.2 Performance Metrics

The following metrics are evaluated to assess the performance of the resources in the grid environment.

1. Power-Cost ratio (PCR) : It is defined as the sum of computation power delivered to 'n' tasks to the cost incurred in completing the tasks.

$$PCR = \sum_{i=1}^{n} CP_i / Cost_i \quad (8)$$

2. Resource Utilization: It is defined as the percentage of processing power allotted to user's task out of the total processing power available at a selected resource node.

$$ResourceUtilization = \frac{\sum_{i=1}^{n} UCP_i}{TCP} \quad (9)$$

Where, UCP_i is the Utilized Compute Power for the i^{th} task and TCP is the Total Compute Power of the selected resource Ri.

3. Reliability: Reliability of a resource is defined as the number of jobs executed successfully to the total number of jobs submitted to the resource. The reliability of a resource R for n jobs is given as,

$$Reliability_R = \frac{\sum_{i=1}^{n} Jobtype * Jobstatus}{n} \quad (10)$$

Where, Jobstatus represents the status of job completion and Jobtype represents the type and size of the job.

4. Simulations and Discussions

The performance of the proposed algorithm is analyzed and discussed. The simulation was based on the grid simulation toolkit GridSim Toolkit 4.0 which allows modeling and simulation of entities in grid computing systems. In this simulation environment the Resource Selection model has been incorporated as the middle layer component for computing the values of the user QoS metrics according to the dynamic information of resources.The heterogeneous Grid environment is built by using various diverse resource specifications. The resources differ in their operating system type, number of processors, CPU speed, RAM memory, Baud rate. In GridSim, application jobs are modeled as Gridlet objects that contains all information related to the job and execution management.

For simulation purposes we have considered five heterogeneous resources with different characteristics such as number of processing elements (PE) in a machine, MIPS rating of a processing element, type of operating system and cost of using the machine. The simulation is done for different client's jobs and the performance metrics are evaluated. The simulation set up is shown in Table 1.

Table 1: Resource Information

Resource ID	CPU speed (GHz)	No.of processing elements (PE)	BW (MHz)	Cost ($/M)	Trust value
R1	2	5	2	3	0.7
R2	1.2	12	1.5	3	0.6
R3	1	10	1	3	0.5
R4	1.5	15	1.5	2	0.8
R5	1.2	10	1.5	2	0.5

In our resource selection and allocation scheme, the resources allocated to the previous requests cannot be preempted by the new request before the current jobs get completed and the resources released. We illustrate the resource selection and allocation process using the proposed method with five resource sites and fifty jobs. The jobs are characterized by the demands of compute power, trust level for security, time and budget constraints. The Information of resources used in the simulation is shown in Table I. The length of the jobs varies from 2000 MIPS to 50000MIPS. The same set of jobs is submitted to the Trust based resource selection model (TBS) and proposed User QoS based resource selection model (UQS) and their performance is analyzed. The inputs are the job demand and the available resources. The output is the selected resource for the job and its obtained performance metrics for the submitted job.

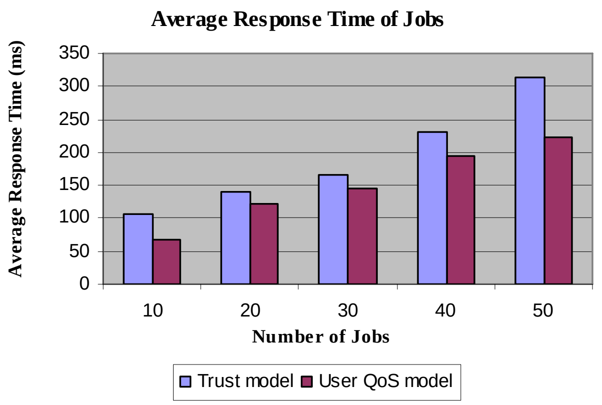

Figure.3. Average response time of jobs

The user's jobs when submitted to the resources selected by the resource selection algorithm, has a reduced average response time compared to the Trust model. This is shown in figure 3. For the first 10 jobs submitted, the average response time observed was 68 millisecond and as the input jobs varies from 10 to 50 there is a small increase in the response time as 223ms.But this is tolerable compared to the response time of trust model which is 315ms. The improvement is due to the fact that if a resource has a longer expected response time, the algorithm finds the next available resource satisfying the user demands. Also the incoming jobs are assigned to resources that better fits the required compute power and not to the resource that has the maximum compute power.

The proposed algorithm improves the success rate of the jobs submitted to the selected resource and hence the reliability of the resource is improved. For a set of 50 jobs submitted to the grid, the reliability of every resource is evaluated with respect to the successful completion of the job. Figure 4 illustrates the reliability of every resource in the grid environment when both the methods of resource selection is employed. The percentage of job failure is very much reduced as jobs are allocated considering the issues of security, availability, time and cost constraints. From our results, resources R1 and R4 are highly reliable as their trust values and computation capacity are high and connected to a high speed link. The jobs are assigned to resources that satisfy the trust level of the job and not only to highly trusted resources. Hence depending on the requirement, jobs are assigned to the other resources in the grid and achieve a good success rate.

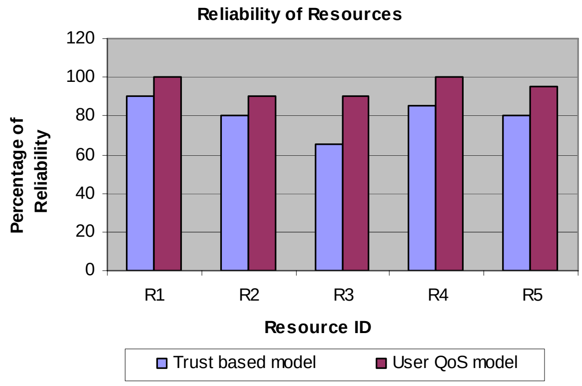

Figure.4. Reliability of Resources in Computation Grid

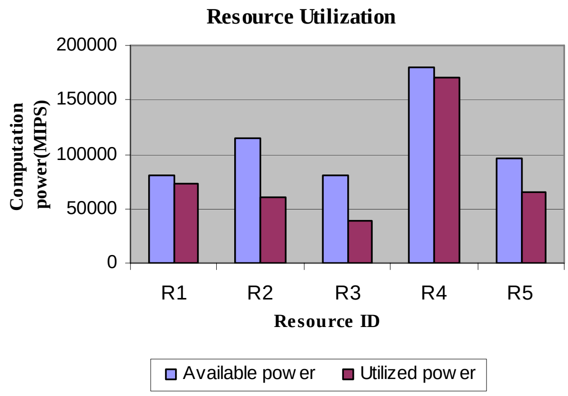

Figure.5. Resource Utilization in Computation Grid

The utilization of resources in the grid environment is given in figure 5.It is evident from the graph that maximum amount of resource power is utilized during a time interval. The available power is computed for a time duration of 8 seconds and the about 30 jobs of different sizes were submitted to the grid. The jobs were assigned to appropriate resources following the proposed algorithm and it was possible to achieve about 60%- 95% of resource utilization of every resource that participated in the grid.

The power-cost ratio is an important performance metric that is improved in our work. Figure.6. shows the obtained power-cost ratio of various resources on executing about 30 different jobs in the grid. From the results, resource R4 has the maximum power-cost ratio, where it has executed 2900 MI/$. The resource R4 has high computation capacity and is offered at a lower cost of 2$/min.

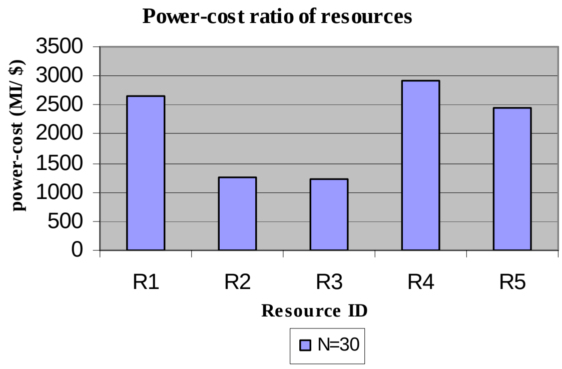

Figure.6. Power-Cost ratio of resources

Hence large jobs are assigned to R4 and executed utilizing more power at a low cost. Hence it has a high power-cost ratio among the other resources in the environment. Resource R1 achieves the next higher power-cost ratio as it is a highly trusted resource and performs high speed computation for small and medium jobs. The cost of using R1 is high. But if it satisfies the budget constraint of the client's job, it is assigned. The resource R1 has given a high success rate for the submitted jobs and has a power-cost ratio of 2600 MI/$. From the above results and discussions, we see that the resources if assigned appropriately before the submission of jobs, by considering the user QoS parameters, achieves a high success rate for the submitted jobs and also utilize the maximum amount of resource capability in the grid environment.

5. Related Work

In the existing Resource Management Systems (RMS) mapping of a resource strictly according to trust can cause severe load imbalance in large scale distributed systems [4].The authors propose a TRM algorithm that considers the execution cost, completion cost and security cost of executing a client's task on a trusted resource. The algorithm is based on subjective trust which is not well suited in a dynamic grid environment. Job scheduling is based on the minimum completion cost criteria. Further, the execution cost was computed for jobs with LOLO (low task and low machine) heterogeneity. In the proposed model in addition to minimum cost criteria, a minimum response time is also ensured.

An effective access model and management of trust in heterogeneous grid environment is discussed in [6]. Dynamic access control is performed by evaluating the trust of user entity. The utilization of resources in a Grid environment is based on the percentage of successful jobs executed on grid. Hence access control should not only enforce access control condition to resources but also validate and allocate the resources to the client jobs to improve the utilization rate of resources in Grids. In our model, the resource requirement and availability are ensured before job is assigned assuring a success rate always.

The variability of distributed resources and the variability of the application workload may have a dramatic performance impact on the grid environment. In [13], resource utilization in grid systems is improved by load balancing strategy. Load balancing decision is made during execution by the runtime state information of resources. Though the runtime estimation of resources is the best in correctly estimating the resource requirements, there is a time overhead for collecting the requirements. This delay in allocation of resources during runtime for a job is avoided in the proposed method.

A resource allocation policy based on competitive proportional cost is developed in [11]. A system model is described to coordinate resource allocation decision without assuming a prior cooperation. The objective of the work considers the benefit of the client and the resource provider. From the client's view the goal is to complete the job as quickly as possible by spending the least possible amount of money. The method is primarily based on maximizing the profit of competing grid resources by increasing or decreasing the price of any unit by a small amount after negotiation. Hence, the resource allocation approach is only price directed and not utility directed.

Many resource selection and allocation algorithms have been proposed for distributed grid systems. Unlike the other systems, this paper presents a resource selection strategy that maximizes the resource utilization by allocating jobs to resources that has the computation power required for the jobs rather than allocating jobs to resources that has maximum computation power. Security for the client jobs is implemented in the initial phase of resource selection by evaluating the trustworthiness of the resource and submitting the jobs to resources that satisfies the trust level specified by the user. Applying security in the resource selection process improves the success rate of jobs and reliability of the resources in the grid environment.

The existing resource selection algorithms are based on either time optimization or cost optimization or a mix optimization strategy. In this paper the selection algorithm is designed to select the resource that provides a minimum response time with a minimum execution cost among the available set of resources and thereby reduces the total completion time for the jobs submitted to the grid. Therefore, the proposed system selects the resources that better fit the client application needs satisfying the time and budget constraints.

6. Conclusions

Traditional resource selection approaches selects a good set of resources to satisfy an application's resource specification. The existing systems do not cover the aspects of security, quality and cost in the resource selection process. We have described the design of optimal and performance aware resource selection strategy that identifies a suitable resource based on the status information of the trusted resource. The jobs are mapped to the available trusted resources by considering the user mentioned QoS specified during the selection process.

Hence, resources are selected with respect to the QoS metrics as computation power, time and cost requested by the user and offered by the resource provider. The simulation result shows that the resource with higher compute power has a low average response time for the submitted jobs. Also the utilization and reliability of the resources are improved and thus achieves a high success rate for the client's task. The future work is to select multiple resources that are required for client jobs in a secure and efficient manner.

References

[1] Foster. I, Kesselman.C and Tuecke.S, "The anatomy of the grid: Enabling scalable virtual organizations", International Journal of High performance computing Applications,V.15 n.3,August 2001, pp. 200- 222.

[2] Foster.I , Kesselman.C , Nick.J, Tuecke.S, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Technical report, Open Grid Service Infrastructure WG,Global Grid Forum, June 22, 2002,pp 37-46.

[3] Ching Lin,Vijay Varadharajan, Yan Wang, Vineet Pruthi, "Enhancing Grid security with trust management", Proc.Of IEEE International Conference on ServicesComputing , Shanghai, China ,September 2004, pp 303-310,.

[4] Farag.A.Azzedin and Maheswaran.M , "Towards Trust Aware Resource Management in Grid Computing Systems", IEEE International Symposium on Cluster Computing and the Grid , Berlin, Germany., 2002, pp 1-6,

[5] G.Kavitha, V.Sankaranarayanan, "Secure Resource Selection in Computational Grid based on Quantitative Execution Trust" , Proc.Of International Conference on Information, Communication, Electrical and Computer Engineering, Singapore, December 2010, pp 149-155
[6] Xudong Ni , Junzhou Luo, " A Trust aware access control in service- oriented grid environment", International Conference On Grid and Cooperative Computing ,2007, pp 417-422.
[7] Fatos Xhafa, Ajith Abraham "Computational models and heuristic methods for Grid scheduling problems", Future Generation Computer Systems, volume 26, 2010, pp 608-621
[8] Thamaraiselvi.S, Balakrishnan.P, Kumar.R, Rajendar.K, "Trust based Grid scheduling algorithm for commercial grids" ,International Conference on Computational Intelligence and Multimedia Applications, sivakasi,India ,2007, pp 545-558.
[9] Yao Wang, Julita Vassileva ," A Review on Trust and Reputation for Web Service Selection" , 27th International Conference on Distributed Computing SystemsWorkshops, IEEE computer society, Washington, DC, USA, 2007, pg.25.
[10] Marty Humphrey , Mary R..Thompson ," Security Implications Of Typical Grid Computing Usage Scenarios ", Proc. Of 10th IEEE International Symposium on High Performance Distributed Computing, San Fracisco,CA August 2001, pg.0095.
[11] Chunlin Li, Layuan Li , "Competitive proportional resource allocation policy for computational grid",Future Generation Computer Systems volume 20,2004, pp.1041-1054.
[12] Abdul-Rahman.A and Hailes.S, Supporting trust in virtual communities," Proceedings of the 33rd Hawaii Int'l Conference on System Sciences, Maui, Hawii, Jan. 2000, vol.1,pg.6007.
[13] Po-Jung Huang, You-Fu Yu, Kuan-Chou Lai, Chao-Tung Yang , "Distributed Adaptive Load Balancing for P2P Grid systems", 10th International Conference on Pervasive Systems, Algorithms and Networks, 2009, pp 696-700.
[14] Bendahmane. A, Essaidi.M , A. El Moussaoui, Younus.A , "Grid Computing Security Mechanisms: State-Of-The-Art", International Conference on Multimedia, Computer andSystems,Ouarzazate, 2009, pp 535-540
[15] Gong Youg,Yao Li, Zhang wei-ming, Sha Ji-chang, Wang "Methods for Resource Allocation via agent Coalition Formation in Grid Computing Systems" , IEEE International Conference on Robotics, Intelligent Systems, Signal Processing, October 2003, pp.295-300.
[16] Ozan Sonmez, Nezih Yigitbasi, Alexandru Iosup "Trace-Based Evaluation of Job Runtime and Queue Wait Time Predictions in Grids", International conference on High Performance Distributed Computing, June 2009,Munich, Germany.

G.Kavitha was born in India in 1978. She received her B.E. in Electronics & Communication Engineering in 2000, M.E. in Computer Science & Engineering in 2002 from Madras University. She is doing Ph.D in Computer Science & Engineering in Anna University, Chennai. She has a teaching experience of 10 years. She is currently working as an Assistant Professor in Information Technology Department at B.S.A.Crescent Engineering College, Chennai,India.

Dr. V.Sankaranarayanan was born in India in 1942. He received his B.E. in Electical Engineering from Annamalai University in 1966, M.Sc. (Engg) in High Voltage from college of Engineering, Guindy, Chennai and Ph.D. in computer applications from IIT (Madras) in 1979. He worked as professor in Anna University for about 3 decades. He was then deputed as Director Tamil Vitual University, from 2004 - 2006 Chennai. He is currently Director at B.S.Abdur Rahman University Chennai, India.