Decision based Resource Selection in Grid Environment

Rajesh Kumar Bawa
Associate Prof., Deptt. of Computer Science
Punjabi University, Patiala

Gaurav Sharma
Assistant Prof.,Deptt. of Computer Engg.
JMIT Radaur,Yamunanagar

ABSTRACT
Grid computing allows users to locate computing resources and data dynamically during the computation. One of the main challenges in Grid computing is efficient selection of resources for the tasks submitted by users. Resource Selection is the most crucial phase in grid scheduling and resource management. The goal of selection is to identify list of authenticated resources that are available in the grid for job submission and to choose the best node. The challenges for the best resource selection involve analysis of several factors such as prediction time to run a job, access restriction to resources, and cost to use resources. In this paper we present a DBRS (Decision based Resource Selection) architecture that combines these influential factors and make the resource selection process more effective. We proposed the decision-making process which includes time utility function, price utility function and resource assessment and based upon these values we calculate multi attribute value. Then according to the multi utility values we rank the resources. The resource having highest multi utility values given highest rank and got selected for job submission.

General Terms
Grid computing phases, Job execution time prediction, Grid resource selection model

Keywords
Grid computing, Resource Selection, Ranking, Execution Time

1. INTRODUCTION
Grid computing is a type of parallel and distributed system that enables the distribution, selection and aggregation of geologically resources dynamically at run time depending on their availability, capability, performance, cost, user quality of self service requirement. The individual users can retrieve data, transparently, without taking into account the location, operating system, account administration, and other details. The details are abstracted, and the resources are virtualized. Grids functionally bring together globally distributed computers and information systems for creating a universal source of computing power and information [3]. A key characteristic of Grids is that resources (e.g., CPU cycles and network capacities) are shared among various applications, and therefore, the amount of resources available to any given application highly fluctuates over time. Resources are owned by various Virtual Organizations (VOs) and shared under locally defined policies that specify what is shared, who is allowed to share and under what conditions [1].To realize resource sharing, the Grid architecture need support several services and resource selection is one of them. Grid resource selection is a process that chooses suitable resources from the candidate resource set which is provided by the resource discovery mechanism. The existing methods of resource selection mainly evaluate some hardware parameters such as the frequency of CPU, the size of memory, the storage size of disk. They do not materialize the dynamic nature of gird resource. The dynamic nature of grid issues the unpredictability of desired resource for user. In grid environment, uncertainty of resource nodes and factitious tricks of user are ineluctability. In order to meet user's requirements of Quality of Service (QoS), ensure that task executes on the trust node and decrease the failure rate of task. The selection process should consider several factors, such as application minimal requirements, application run time, and resource access policies. In addition, it must consider the uncertainties associated with each resource and answer questions related to resource reliability, prediction, and cost to access a resource. The selection problem has been tackled in the literature in a number of different ways. Nevertheless, each approach solves this problem considering only one of the factors that can lead to the optimal selection. As far as we know, an approach that properly combines these important factors into an optimal grid resource selection is nonexistent. Due to above background, we propose a resource selection architecture that consists of resource discovery phase, filtering phase verification phase, job submission phase and decision making phase that applies decision making process by considered expected execution time, resource assessment and price utility function in grid environment to minimize the execution time/cost and maximize the performance.

2. RELATED WORK
Xiong and Liu in their paper [3] use an adjusted weighted average of amount of satisfaction that a user gets for each transaction. The parameters of the model include the feedback from transactions, the number of transactions, the credibility of feedbacks, the criticality of the transaction. Wang and Vassileva [4] use a naive Bayesian network which is generally used for representing and analyzing models involving uncertainty, to represent the trust of a user with a provider, the concept of trust being defined in terms of both the capability of the provider in providing services the reliability of the user in providing recommendations about other users. Cray, R [6],] Henderson, R. and D. Tweten [7],proposed method in which user-submitted jobs by finding resources that have been identified either explicitly through a job control language or implicitly by submitting the job to a particular queue that is associated with a set of resources. This manually configured queue hinders the dynamic resource discovery. The AppLeS framework [2] guides the implementation of application-specific scheduler logic, which determines and actuates a schedule customized for the individual application and the target computational Grid at execution time. Dongarra et al. developed a more modular resource selector for a ScaLAPACK application [1]. Since they embed the application-specific detail in the resource selection module, however, their tools cannot easily be used for other applications. Alunkal et. al. [5] select a server with

the best reputation for service delivery based on user feedback. However, instead of considering subjective information obtained from users, our approach verifies the resource reliability considering precise historical data from past job executions. In addition, it verifies resource access and predicts job execution time. The proposals by Dewey, Simon, and Brim et al are all *sequential* in the sense that they divide decision processes into parts that always come in the same order or sequence. Several authors, notably Witte (1972) have criticized the idea that the decision process can, in a general fashion, be divided into consecutive stages. His empirical material indicates that the "stages" are performed in parallel rather than in sequence. Although some efforts have been made for grid resource selection, these solutions aim at isolated foci concerning resource access or job execution prediction. The optimal selection continues to be an open problem and, for this reason, searching for new techniques that provide a suitable solution remains an important research topic.

3. PROPOSED ARCHITECTURE

This section presents the architecture for resource selection which considered five phases i.e. Resource discovery, filtering, Verification, decision making and job submission phase. The pictorial description of these phases is shown in the figure 1.

Resource discovery phase: Resource discovery is the basic component of Grid resource management which is the core of Grid and provides the list of resources available in the grid.

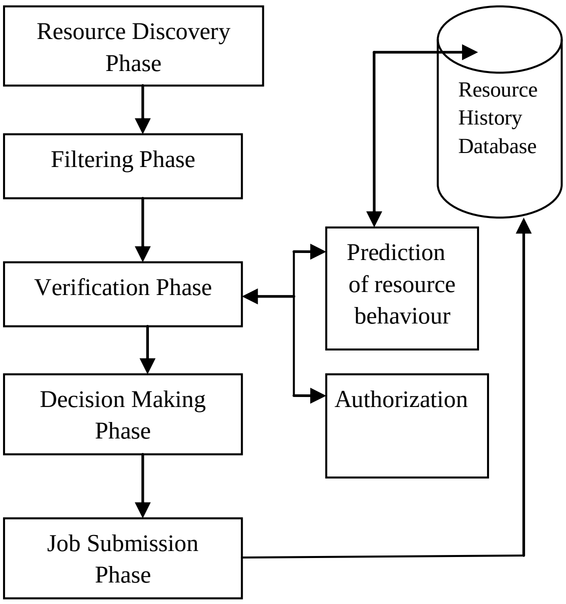

Fig. 1- DBRS Architecture.

Filtering phase:-The resources discovered are filtered depending on user needs i.e. minimum RAM size, maximum cost, minimum processor size. The output of this phase are list of resources which full fill the minimum criteria imposed by user.

Verification Phase:- In this phase prediction and authorization are used so that only reliable nodes are participate in selection process. In Prediction of resource behaviour, we predict the reliability of the resource based upon its past history i.e. how many time job is submitted to that resource and how many times it successfully executed the job. In authorization we check the resource certificates to ensure whether the resource is authorized or not for participating in selection process.

Decision making phase:- Fourth phase is decision making phase. In this phase finally the resource is selected for job submission. We assigned rank to each resource according to their performance, cost and execution time. The resources having better performance, less cost have given higher rank. In the subsequent section we will briefly explain that how the performance and cost is evaluated.

Job submission: Job submission phase is the last phase of our proposed architecture. In this phase, the job is submitted to selected resource i.e. having highest rank value. The successful completion or failure information is send to Resource History database.

Resource History Database: - It will store the history of the resources i.e. how many time job is submitted to resource and how many time he successfully executed the task. This information is used for prediction of the resource behaviour.

All these phases constitute the DBRS. But the most crucial phase is the Decision Making phase in which the resource gets selected. The performance of the grid is directly proportional to this phase. So, in order to make resource selection effective we have to make an efficient decision making process.

4. THE DECISION MAKING PROCESS

This section presents the decision making process which is based on a multi-attribute function that aggregates inputs from time utility function, resource assessment function and price utility function. The utility functions are based on the mathematical decision theory and resource assessment function is based on the node's performance. Using this multi-attribute function we rank the resource i.e. higher the multi attribute function value, higher the rank.

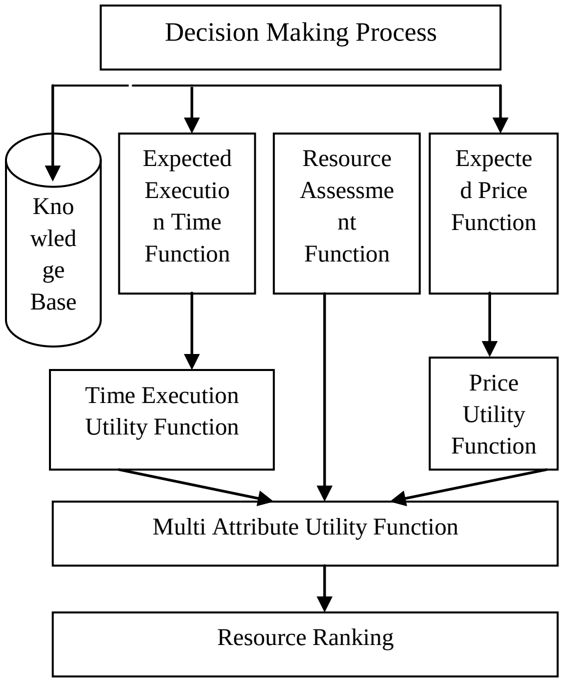

Fig. 2- Decision making Process

4.1 Utility function (U):

Utility Function is one the main concept of decision theory that is used in this paper. The decision theory is a mathematical framework for determining the best option of a decision problem providing optimal recommendation.

Utility function (U): A utility function designates a number to express a preference for a state. Utility is a function that maps states into numbers. The utility function for a variable can be defined by means of analytical expressions such as linear, exponential, or logarithmic functions [8].

The parameters used in calculating the utility functions, use a knowledge base that contains the information about the node. This information is used in prediction of job execution time about a node by using past cases.

4.2 Expected execution time function

The selection model receives the execution time from machines Mi. Function T corresponds to the expected value of T, E(T), as described in formulas (1) and (2).

$$T= E(T)=\sum_{i} ti\ Pr(ti) \qquad (1)$$

$$T=(t1*q)+(t1-p(s)*t1)(1-q) \qquad (2)$$

Where:
t1 = execution time in case of successful prediction;
t2 = execution time in case of unsuccessful prediction, and t2 = t1 –p(s)* t1;
p(s) = represents the prediction error rate for a degree s of similarity between the new and past cases;
q = prediction success degree;
pr = represents the minimal acceptable price to run a job in that node. [8].

4.3 Time execution utility function

Now the value of T is used in calculating the Time execution utility function U(T). In the function U(T) the alternative is more undesirable for higher values of T. The exponential function is commonly used in these situations.

$$\mathbf{U(T)} = \mathbf{e}^{-\tau T} \qquad (3)$$

Where:
T represents the job execution time prediction in milliseconds;
τ represents the maximum scale value considered (for predictions up to 24 hours, τ =1/86.400.000)[8].

4.4 Expected price function

Agents and users negotiate the price of resources. The function P corresponds to the expected value of P, E(P), as described in formulas (4) and (5).

$$P= E(P) =\sum_{i} p\ i\ Pr(p\ i) \qquad (4)$$

$$P = (z*p1)+ (1-z)(p1-g*p1) \qquad (5)$$

Where:
p1 = applied price in case of successful prediction;
p2 = applied price in case of unsuccessful prediction, and p2= p1 –g*p1;
g = discount in price p1 in case of unsuccessful prediction;
z = reliability degree in the service provision by machine Mi, where *z* is associated with evidence collected from agents. *z* represents the relation in Mi between the total past job executions that run faster or equal to predicted time, and the total previous job executions. The probability *z* is applied to the negotiated price[8].

4.5 Price utility function

Now the expected value of price is used in this utility function. In the function U(P) the alternative is more undesirable for higher values of P. The exponential function is commonly used in these situations.

$$\mathbf{U(P)} = \mathbf{e}^{-\sigma P} \qquad (6)$$

Where:
P represents the price obtained in the negotiations;
σ represents the maximum value considered in the scale p = [0…100],so σ = 1/100[8].

4.6 Resource Assessment Function

In the resource assessment function the resources are evaluated based on the RAM assessment, CPU assessment and the bandwidth assessment.

$$U(R) = \frac{(RAM_{Assess}+CPU_{Assess}+Bandwidth_{Assess})}{(W_{RAM}+W_{CPU}+W_{Bandwith})} \qquad (7)$$

Where:

$$CPU_{Assess}= \frac{W_{CPU}*(1-CPU_{load})*CPU_{Speed}}{CPU_{Min}} \qquad (8)$$

$$RAM_{Assess}= \frac{W_{RAM}*(1-RAM_{usage})*RAM_{Size}}{RAM_{Min}} \qquad (9)$$

$$Bandwidth_{Assess}=W_{Bandwidth}*(1-Bandwidth_{used})*Bandwidth_{Required}/Bandwidth_{Min} \qquad (10)$$

4.7 Multi-attribute function

The multi-attribute utility function, represents the preference for P or R or T, U(P,T,R), using preferences δ1,δ2and δ3 informed by the user. The multi-attribute utility function U(P,T,R) of each machine is calculated by formula (11).

$$\mathbf{U(P,T,R)} = \delta_1 \mathbf{*U(P)} + \delta_2 \mathbf{*U(T)}+\delta 3\mathbf{*U(R)} \qquad (11)$$

Where:
$\delta_1$is the user preference for P;
δ_2 is the user preference for T;
δ_3 is the user preference for R;
$\delta_1 + \delta_2 + \delta_3 = 1$;
U(P) is the computation of utility P on machine Mi;
U(T) is the computation of utility T on machine Mi.
U(R) is the computation of resource assessment function.

5. RESULTS AND DISCUSSION

This section presents a subset of our experiments that demonstrates the functionality and performance of our proposed selection approach. We performed our experiment on Grid simulator i.e GridSim ToolKit 5.2 which allows modelling and simulation of entities in grid computing systems-users, applications, resources, and resource schedulers for design and evaluation of algorithms [9]. We performed our experiment with 500 nodes, which are Pentium-4 based systems with CPU clock speed of 3GHz, 2GB RAM and windows XP operating system. In our experiment setup we submitted approximate 1000 jobs and compare the failure rate of our proposed DBRS approach with traditional resource selection approach. From Fig.3, it was observed that the failure rate for first 100 jobs is almost same in both approaches but as the number of jobs increases the failure rate of traditional resource selection approach is increases drastically. Since traditional resource selection approach follows simple match making algorithm, it does not pay attention to the efficiency and long term availability of the resources. Also in traditional approach, if the resource doesn't continuously service well, as long as it is available and fulfil the maximum requirement, it will be selected repeatedly, which leads to increase in the failure rate. And as a result, user may end up with low quality or inconsistent resources leading to disappointing results. However in DBRS based approach if some resource fails, the resource assessment value falls rapidly, leading into lower overall multi attribute value and it will no longer be chosen by the user.

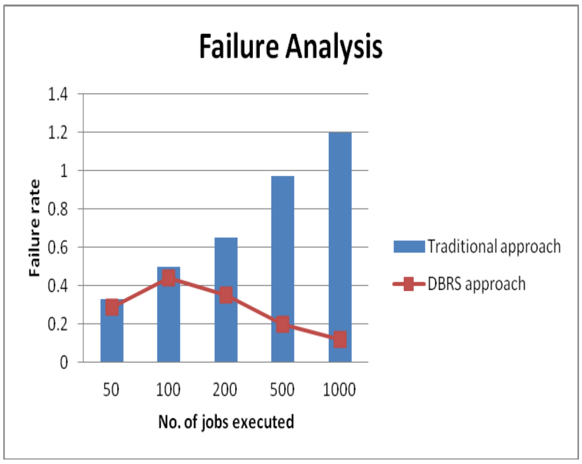

Fig. 3 Failure rate of both approaches

Figure 4 shows the trend of the two approaches for resource selection and allocation. The failure rate of jobs in traditional approach is holding a high failure rate between 70% to 50%, whether the number of jobs executed added or not. And failure rate of jobs in DBRS approach falls down soon to 18% when the number of jobs executed added. This is due to the efficiency of approach.

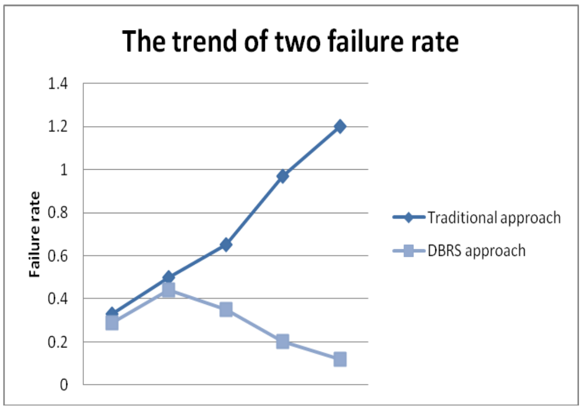

Fig. 4 Comparison of both approaches

6. CONCLUSIONS

The Grid environment provides a promising platform for the efficient execution of complex applications. Scheduling such kind of applications is somewhat cumbersome because target resources are heterogeneous in nature, and also their availability varies. So in this paper we proposed an architecture DBRS which is composed of five phases. Discovery and filtering phase provide a list of authenticated and eligible resources according to minimum requirement imposed by user which are further verified using prediction and authorization mechanism. The next phase is decision making process on which we select the best resource according to user need and assign rank to each resource. For making decision we calculate the expected execution time, resource assessment function, expected price of the job. Based upon all these we calculated multi attribute function. And the resources are ranked depending upon the value of multi attribute function. The top most resource will be selected and the job is being submitted to the resource. Also the study of failure rate comparison of the proposed approach reveals that the failure percentage of jobs gets lower.

7. REFERENCES

[1] Petitet, A., S. Blackford, and J. Dngarra, Numerical Libraries And The Grid: The GrADS Experiments With ScaLAPACK. 2001, University of Tennesse.

[2] Berman, F. and R. Wolski. The AppLeS project: A Status Report. in Proceedings of the 8th NEC Research Symposium. 1997. Berlin, Germany.

[3] Xiong L. , and Liu L. , (2004) 'PeerTrust: Supporting Reputation Based Trust for Peer-to-Peer Electronic Communities' , IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 7,pp 843-857.

[4] Wang, Y. and Vassileva, J. (2003) 'Trust and reputation model in peer-to-peer networks', Proceedings of the Third International Conference on Peer-to-Peer Computing, Linkbping, Sweden, pp. I 50- 157.

[5] V. Lo, J. Mache, and K. Windisch, "A Comparative Study of Real Workload Traces and Synthetic Workload Models for Parallel Job Scheduling," in 4th Workshop on Job Scheduling Strategies for Parallel Processing, Orlando, USA, ser. Lecture Notes in Computer Science (LNCS), vol. 1459. Springer, 1998.

[6] Cray, R., Document number in-2153 2/97. 1997, Cray Research.

[7] Henderson, R. and D. Tweten, Portable Batch System: External reference specification. 1996, Ames Research Center.

[8] Lilian Noronha Nassif, José Marcos Nogueira, Flávio Vinícius de Andrade, Distributed Resource Selection in Grid Using Decision Theory, Seventh International symposium on Cluster Computing and the Grid,2007.

[9] Buyya R, "A Grid simulation toolkit for resource modeling and application scheduling for parallel and distributed computing". www.buyya.com/gridsim/.