Ranking Based Resource Selection in Grid Environment

[1]Rajesh Kumar Bawa, [2]Gaurav Sharma
[1]Punjabi University, Patiala Punjab, India
[2]J.M.I.T. Radaur Engg. College, Yamuna Nagar, India
[1]rajesh_k_bawa@yahoo.com, [2]gaurav13@gmail.com

***Abstract*— In recent years the development of Internet and Grid technology has led to a remarkable increase in the number of resources to which a user, program, or community may have access. When selecting a resource a user must first consider whether the resource has the required capabilities (e.g. processor architecture, network bandwidth etc.), and second whether it is likely to complete the task successfully. The first case is simple, either a resource has the required capabilities or it does not. Those that do not can be rejected, but the second case become more complex when the number of resources which satisfy user demands is quite large, selecting the best resource to run a specific job remains one of its main problems. When a large number of resources are fulfills the user criteria then problem to choose a right resource is quite similar to choose a right page in information retrieval. So a good ranking mechanism is required to solve the problem of Resource selection. In this paper a ranking mechanism is propose which is based upon certain factors like affordability, Success-rate, Self protection capability, user feedback to solve the problem of resource selection in grid environment.**

Keywords: Grid, Resource Selection, ranking, affordability, Success-rate, Self protection capability, user feedback.

I. Introduction

Grid computing aims to connect large numbers of geographically and organizationally distributed resources to increase computational power, resource utilization, and resource accessibility. In order to effectively utilize grids, users need to be connected to the best available resources at any given time. In small, single organization grids, users may be able to adequately select resources for simple requests on their own. Even small grids, however, are in constant flux with resource availability changing minute by minute due to user activity and system failures. Large, multi-organization grids are much more chaotic where resource types, connectivity, support levels, software environments, availability, etc. may vary greatly with a greater probability of resource failures. In such an environment, users cannot possibly keep up with the configuration and status of the grid nor potentially even which resources they may have access to, thus they will tend to choose only those resources they are directly familiar with. This leads to imbalanced resource utilization, which results in longer wait times and a decrease in productivity. To maximize the benefits of grid computing, users must be provided with automatic resource selection and ranking resources to meets his requirements [12]. Grid computing is concerned with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources.To support application execution in the context of the Grid, a Grid Resource Broker is desirable. Grid Resource Brokering is defined as the process of making scheduling decisions involving resources over multiple administrative domains. Grid Resource Discovery and Selection is an essential and crucial part of grid resource brokering, which provides adequate available resources for other grid resources management. Resource discovery is a previous process before resource selection. The goal of resource discovery is to identify a list of authenticated resource objects that are available for resource users. Once the list of possible target resource objects is known, the second phase is to select those resource objects that best suit the constraints and conditions imposed by the user, such as CPU usage, RAM available or disk storage. Resource selection involves a set of factors, such as application minimal requirements, application run time, and resource access policies [1]In addition, resource selection must consider uncertainties associated with each resource and answer questions related to resource reliability, prediction error probability, and cost error probability to access a resource. Grid technology allows resource sharing among several entities, but selecting the best resource to run a specific job remains one of its main problems. The Challenges for the best resource selection involves analysis of several factors such as affordability of resources, success rate of resources and cost to use resources [2]. This paper emphasizes on improving the current resource selection technique in grid scheduler by taking into account the quality and reliability of resources.

II. Related Work

Resource selection process is used to choose one or more resource objects from the resource objects list of candidates for a given resource requirement. Since all resource objects in the list could meet the minimum requirements imposed by the job, so an approach is needed to choose the best resource object to execute the job. Many techniques has been proposed so far. Lilian, José and

Flávio Vinícius de Andrade [2] use decision theory in resource selection, which combines the probability and utility theories. Past evidence is used to calculate resource reliability and Utilities are associated with user preference. The prediction module PredCase, can give the user guarantees about when his job will finish and allows advance job scheduling. PredCase presents a small error prediction percentage and improves case retrieval performance. The decentralized restriction verification model, POLAR, can verify the resource workload, SLA parameters, and user access.The decision model groups this information and selects the best machine to run a job.The Xuan Wang And Ling-Fu Kong[5] proposed a resource selection algorithm based on the multi-goals.He proposed resource credit model and then established resource provider model. Based on these models, a dual-credit is introduced to describe the behaviors of the resource and its provider objectively. In the next place, he considered a time-cost synthetic measurement function joins into the MGCS to meet the user requirements for response time and cost. In the end he proposed improved scheduling architecture, which supports the credit evaluation. Jianhua, Hui Peng and Juhong[1] Tie proposed a resource selection algorithm based on binary variable, which take the similarity of hardware and software resources into account based on a quantitative evaluation between the resource objects list of candidates and a given resources requirement. YU Huashan and XU Zhuoqun [6] proposed a two-layer framework for selecting and accessing resources on computational Grids. It groups resources across organizations in an application-oriented manner, and virtualizes each group of resources to be a set of stateful WEB services. These WEB services provide the capabilities for combining resources within, as well as across, organizations to prepare customized job execution environments as required. Peggy Lindner, Edgar Gabriel and Michael M. Resch [7] present a novel approach for resource selection based on the estimation of the execution time of an application using a simulator. Given a pool of machines, the Grid Configuration Manager (GCM) tests several possible combinations of machines fulfilling the requirements of the end user regarding the required computational resources. Chen Chen, Gu Li-ze, Niu Xin-xin, Yang Yi-xian [8] proposed a Resource selection technique based on trust management system. In a trust management system, a node is usually named an OP (operation point), the node requesting resources is called an CP (control point) and the node supplying resources is an PP (participate point). All entities can register into the trust management system. In grid environment, each domain includes a trust agent, several trust subagents and other entities. A trust agent calculates the trust values and supply trust services. After each transaction, the subagent in charge of the domain should collect related trust information, update the trust value of each node, and generate a new temporary trust value. Then the agent would accumulate every temporary trust value offered by each subagent, and calculate the general trust value of the nodes. In this system, the trust agent or subagent will collect the trust information in grid, calculate the trust values according to a mathematic model and store the values in a database. When choosing resources, the subordinate of the trust agent would make use of trust service, negotiate with the agent, gain the trust value list and choose suitable resource according to trust values and local strategy. Also, further negotiation with the chosen resource should be proceeded in order to identify whether the resource is available and embed this trust value in a scheduler's scheduling algorithm in order to selecting the most trustful resource provider to the users. They considered not only the trust value of the resource but also the local strategies of the resource provider and user. And the approach can get rid of the bad node with malicious behavior in history. The experiments on performance comparison of the trust based scheduler against the trust unaware scheduler reveals that the failure percentage of jobs gets lower. Tyng-Yeu Liang, Siou-Ying, Wang I-Han Wu [9] proposed a novel algorithm of resource selection for computational grids based on resource availability prediction using frequent workload patterns. The basic idea of this algorithm is to discover frequent workload patterns (FWPs) according to the load history of resources by means of the data mining technique, and then select resources for grid jobs according to the prediction of resource availability in a given time period by using the association rules derived from FWPs. In order to achieve a high prediction precision, the proposed algorithm dynamically adapts its prediction in resource availability by flushing old workload data. The simulation Experiment has shown that it can effectively reduce the performance degradation of the proposed algorithm when the time series of resource workloads become irregular. P.Varalakshmi, S. Thamarai Selvi, P.Kanchana and Sarah Waziah [10] proposed three-tier framework where the Regional Resource Administrators (RRAs), resource brokers and the SPs are arranged into three-tiers. An RRA selects a 'suitable' SP for a consumer request, based on the trust indices and Quality of Services of the SPs nominated by various brokers. RRAs derive their compensation from registration, renewal, and audit charges paid by the broker community, not from individual transactions. RRAs serve the consumer community in a manner similar to DNS, in an 'unbiased' and 'trustworthy' manner. The proposed architecture not only supports a choice of SP based on reputation (trust-index) but also on credentials (policy). Thereby the consumer is assigned with a trustworthy SP and the transaction is free from runtime failure, as the policies have been matched with. In this model, the trust indices of each of the entities (consumers, SPs and brokers) are computed based on the feedback provided by other entities after each transaction. These trust-indices of brokers, consumers and SPs are updated dynamically at the RRA's and the broker's sites respectively, to ensure trustworthy services and to quicken the selection of 'suitable' SPs. Noorisyam Hamid et al.[3] proposes a resource selection technique based on resource ranking. They emphasizes on improving the current resource selection technique in grid scheduler by taking into account the quality and reliability of both users and resources. Their Quality-based Grid Resource Discovery *(Q-GreD)* aims at providing a grid resource discovery which takes into account the quality and reliability of both users and resources. Their resource selection technique i.e Resource Ranking is based on Google's search technique PageRank. Google's PageRank is a numeric value that represents the importance of a page on the web. A page has a high rank if the sum of its backlinks is high. Similar idea can be applied to grid where submission of jobs to resources indicates the backlinks. Resource can obtain higher ResourceRank score, if many users from different organizations submit jobs to that resource or there exist users with high ResourceRank using the resource. They calculate the ResourceRank using the damping factor. After that they incorporates ResourceRank into the rank equation in Condor ClassAd. Hence, ResourceRank becomes a new constraint that must be considered when matchmaking is performed. Then the matchmaking between the resources are done on the basis of some rules.

III. OUR APPROACH

The Grid computing provides the ability to access, utilize, and control a variety of underutilized heterogeneous resources distributed across multiple administrative domains. The resources of grid computing vary in level from a small number of large clusters to millions of PC-class machines. Efficient Resource Selection is the fundamental requirement in Grid applications.In our proposed work we present a resource selection technique that mainly aims on selecting a quality and a reliable resource. It makes use of the usage score of a resource and some of the trust factors of the resource. Depending on these factors broker makes the resource selection decision.

A usage score [3] is calculated on the basis of submission rate of the job. If the resource of an organization links to the resource of another organization means that it is casting a vote as an indication that the other resource is good. In other words, higher the rate of submission of jobs means higher the usage score. But this technique relies on the fact that how many times the job submitted on this particular resource and based on this it calculates the rank of the resource. It does not consider the background and quality of the users and resources involved. So, users may end up with low quality or inconsistent resources leading to disappointing results. We can improve by taking into account the other factors which highlight the quality of the resource rather than considering only the importance or usage of the resource, we can also emphasize on Affordability, Success Rate, Bandwidth and self protection capability, user feedback of the resource. These are known as trust factors [8]. Based on these factors, we can improve the selection process of the resource and retrieve the result with more accuracy.

ResourceRank Score or Usage score or Backlink score: It determines the reliability of the resource. It is defined as the sum of the ResourceRank score of the resources which provides the job submission [4].

$RR(A)=(1-d)+d\ (RR(T_1)\ /C(T_1)\ +..+\ RR(T_n)\ /C(T_n)\ )$

Where,

$RR(A)$ = ResouceRank of resource A

$RR(T_1)$ = ResourceRank of resource T1 which uses the resource A

$C(T_1)$ = The number of times user in the organization T1 submits job to current resource in an organization

d = damping factor which usually set to0.85

Affordability (A_A): The ratio between the number of times a resource being available to the grid and the number of attempts made to access the resource. It is defined as:

$A_A= N_t / N_c$

Where,

N_t = Number of times resource being available to grid resource A

N_c = Number of attempts made to access the resource

Success Rate (S_A): The number of successful executions of jobs by a computational resource against the total number of jobs submitted to the resource that indirectly reveals the expertise of the resource provider. It is defined as:

$S_A= J_S / J_T$

Where,

J_S = Number of jobs successfully executed by the resource.

J_T = Total Number of jobs executed by the resource.

Bandwidth (N_A): The speed of connectivity of the resources with the requesting node. The values to N_A are assigned according to the following table [11]:

TABLE I: BANDWIDTH ASSIGNMENT

Bandwidth Obtained from metascheduler Mbits/Sec	Value assigned to N_A
B>100	1.0
10<B<100	0.75
1<B<10	0.75
B<1	0.25

User Feedback (F_A): After each job, the requesting node returns the result of the job and the operation situation of resource to resource provider. According to the feedback, the value is calculated using the following table [11]:

TABLE II: FEEDBACK ASSIGNMENT

Feedback levels	Code	Value
Excellent	F1	1.0
Very High	F2	0.8
High	F3	0.6
Medium	F4	0.4
Low	F5	0.2
Very Low	F6	0.01

Self-protection Capability (SPC): The self-protection capability of an entity is calculated by aggregating the security factors. The values of these factors differ in the range between 0 and 1, which are determined by trust strategy. Based on their contributions to security, a proportion is given to each security factor as a final point aggregated to compute the self-protection capability. The security factors and their proportions are listed in the following table:

TABLE III: SECURITY FACTORS AND THEIR PROPORTION

Security Factors	Proportion(P)
IDS Capabilities	0.825
Anti-virus Capabilities	0.85
Firewall Capabilities	0.9
Authentication Mechanism	0.8
Secured File Storage Capabilities	0.7
Interoperability	0.6
Secured Job Execution	0.75

Based on these security factors SPC is calculated as:

$$SPC = \frac{\sum_{i=1}^{n} P(i)}{n}$$

Where,

n = the total number of factors.

P(i) = the proportion.

Based on all these factors we can calculate the rank of the resource as follows:

$$Rank(A) = \frac{RR(A) * A_A * S_A * F_A * N_A * SPC}{6}$$

Now, according to this calculated Rank the jobs are submitted to the resource. As higher the rank of the resource, higher the priority. The usual scheduler (i.e which uses only resource ranking) follows simple match making algorithm, it does not pay attention to the past service history of the resource and consider only the usage score as a result a resource which has higher usage score is repeatedly selected although it has the poor past records in terms of trust (affordability, success rate,user feedback, self-protection capability and bandwidth) and user may end up with low quality or inconsistent resources leading to disappointing results. So in ranking a resource all these factors are necessary. However in the proposed approach if some resource fails the rank value falls rapidly leading into lower overall trust and it will no longer be chosen by the user.

IV. CONCLUSIONS AND FUTURE WORK

Grid computing aims to connect large numbers of geographically and organizationally distributed resources to increase computational power, resource utilization, and resource accessibility. In order to effectively utilize grids, users need to be connected to the best available resources at any given time but the heterogeneous and dynamic nature of Grids, however, leads to numerous technical problems, of which resource selection is one of the most challenging. The effective and efficient exploitation of Grid computing facilities needs highly advanced and reliable resource management systems. The proposed approach evaluate the rank of the resource by combining the factors like affordability, success rate, user feedback, self-protection capability and bandwidth and the usage score. Also this approach can get rid of the bad node with malicious behavior in history. So this approach gives a reasonable way of selecting the resources that selects the resource based on trust as well as certain factors. Future work include considering more dynamic factors to make more effective selection process.

REFERENCES

[1] J. Adu, H. Peng And J. Tie "A Resource Selection Algorithm Based On Binary Variable In Grid" International Symposium On Intelligent Information Technology Application Workshops, IEEE 2008.

[2] L. N. Nassif, J. M. Nogueira And F. Vinícius De Andrade "Distributed Resource Selection In Grid Using Decision Theory", Seventh IEEE International Symposium On Cluster Computing And The Grid (Ccgrid'07).

[3] N. Hamid, F. Haron and C. H. Yong "Resource Discovery using PageRank Technique in Grid Environment" Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06).

[4] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine". In Proceedings of the 7th World Wide Web Conference, Brisbane, Australia, 1998.

[5] X. Wang And L-F Kong "Study On Grid Resource Selection Method Based On Multi-Goals" Proceedings Of The Sixth International Conference On Machine Learning And Cybernetics, Hong Kong, 19–22 August 2007.

[6] Y.U. Huashan and X.U. Zhuoqun "Standardizing Resource Selection and Access on Computational Grids" Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06).

[7] P. Lindner, E. Gabriel And M.M. Resch "Performance Prediction Based Resource Selection In Grid Environments" High Performance Computing and Communications, Springerlink 2007.

[8] C. Chen, G. Li-ze, N. Xin-xin,Y. Yi-Xian "An Approach for Resource Selection and Allocation in Grid Based on Trust Management System" International Conference on Future Information Networks, 2009 IEEE.

[9] T-Y Liang S-Y Wang I-H Wu "Using Frequent Workload Patterns in Resource Selection for Grid Jobs" 2008 IEEE Asia-Pacific Services Computing Conference.

[10] P. Varalakshmi, S. T. Selvi, P. Kanchana and S. Waziah "Three-Tier Grid Architecture with Policy-Based Resource Selection in Grid" IEEE-International Conference on Signal processing, Communications and Networking, 2008.

[11] S. ThamaraiSelvi, P. Balakrishnan, R. Kumar, K. Rajendar "Trust Based Grid Scheduling Algorithm for Commercial Grids", International Conference on Computational Intelligence and Multimedia Applications 2007.

[12] P. Z. Kolano "Surfer: An Extensible Pull-Based Framework for Resource Selection and Ranking" Proceeding. Of the 4th IEEE/ACM Intl. Symp. On Cluster Computing and the Grid, Chicago, IL, Apr. 19–22, 2004.