SURVEY OF GRID SIMULATION AND COMPUTING

Technological Advancement And Future Scope

Sujay Kakkad[1], Parikshit Chavan[2], Samyak Shamkuwar[3], Ajinkya Walimbe[4],
Prof. Kiran K. Joshi[5], Prof. Sowmiya Raksha Naik[6].
[1,2,3,4,5,6]Department of Computer Technology and IT,
Veermata Jijabai Technological Institute (V.J.T.I.), Mumbai, India.

Abstract—Peer-to-Peer or Grid computing is coming out as a next generation platform to solve large-scale problems in fields of science, commerce, and engineering. It is expected to involve many (heterogeneous) resources distributed in multiple organizations, administrations, and policies. The management of resources and their scheduling in such large-scale distributed systems is complicated and therefore, sophisticated tools for analyzing and tuning the algorithms before applying them to the real systems are needed. This paper combines and recapitulates the findings of prevalent research in this field.

Keywords—Grid Computing, Grid Simulation, Resource management, Resource allocation, Distributed Computing

I. INTRODUCTION

Grid Computing is a type of parallel and distributed system that allows the sharing, choosing, interchange, & collectivization of geographically separate "autonomous" computing systems. A primary strategy of grid computing is to use middleware to divide and break pieces of a program among several computers, sometimes in thousands. Grid computing involves computation in a fashion that is distributed, which may include the aggregation of large-scale clusters. The Global grid has been explained in Figure 1 below[1].

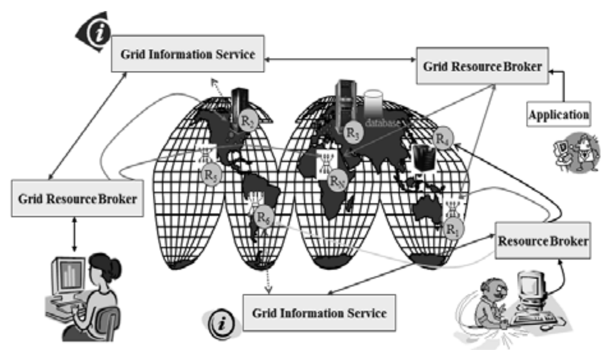

Figure 1: A Bird's eye view of Global Grid

II. ANALYSIS OF GRIDS

A. Simulation

Simulation is the only feasible way to study algorithms on large-scale distributed systems of heterogeneous computing systems. Unlike using the actual infrastructure in real time, simulation works ingeniously, without making the mechanism undesirably complex, by evading the overhead of co-ordination of real parts of the system. Simulation is also useful in working with very large hypothetical problems that would generally need involvement of a high population of active users and resources, which is very difficult to manage and build at large-scale research environment for exploration purposes[2].

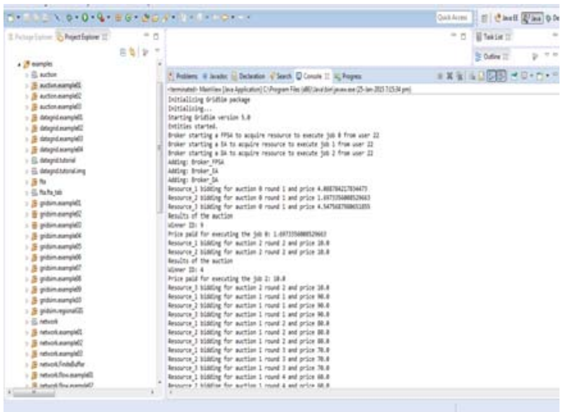

Figure 2: Screenshot of Gridsim Toolkit

B. Uses

Even if only one grid can be dedicated to a particular application, commonly a grid is used for a sundry of purposes. Grids are generally built with general-purpose grid middleware application libraries. E-commerce and E-science are main application areas of Grid Technology. These include the following:

- Utility Computing
- Collaborative Design
- High Performance Computing
- High Energy Physics
- Life Sciences
- Collaborative Data Sharing
- Financial Modeling
- Data center automation
- Drug Discovery

C. Challenges

The main Challenges in Grids are presented below.

- Security
- Uniform access
- Computational Economy
- System Management
- Resource Discovery
- Data Locality
- Network Management
- Application Construction
- Resource Allocation and Scheduling

III. LOAD BALANCING WITH FAULT TOLERANCE

D. Nanthiya and P. Keerthika propose an algorithm to enable efficient management of resources to tackle load problems [3]. The utilization of the Gridsim simulator has been key to verify the functioning of the algorithm[4].When a job is submitted to the Machine, the algorithm works at two phases namely:-Selection of fittest resource and load balancing algorithm.

A. Selection of Fittest Resource

Each job is submitted to the Machine in the grid environment with the user deadline. Then for each available job, ECT (Expected Completion Time) value for each resource is found. Then the resources that are satisfying the user deadline (less than or equal to ECT value of the job) alone are considered. The resource with highest fittest value is considered to be the fittest resource for that job.

B. Load Balancing algorithm

The load balancing mechanism is performed at all the three levels: Broker, Resource and Machine.The algorithm classifies the nodes at three levels into three lists: overloaded list, normally loaded list and under-loaded list based on the load threshold values namely:-PE level threshold, Machine level threshold and Resource level threshold. Overloaded list means set of nodes with over-commitment of jobs. Normally loaded list is a list of nodes, if any job is further submitted to these nodes then they are shifted to the overloaded list. The under-loaded list is a list of nodes which can be submitted with jobs. If the selected node is present in the under-loaded list, then the job is scheduled to the same resource else check for the presence of the next fittest resource in the under-loaded list. Before scheduling the job, expected load of the selected resource should be calculated by adding the load of the job with current load of that particular resource. If the expected load exceeds the load threshold value then it is assumed that the submission of job may lead to the overloaded node condition. If the expected load value below the load threshold, then the job can be submitted. Finally if the load is unbalanced at the PE level, then few jobs are selected and forwarded to the Machine level. If the load is unbalanced at the Machine level, then few jobs are selected and forwarded to the Resource level. Almost the load is balanced at the Machine level itself. By this way the load balancing mechanism is performed at each level considering the fault tolerance factors.

IV. SCHEDULING OF RESOURCES

Simulated Annealing is an algorithm that simulated the procedure of physical solidification. The most difference is its principle of reception, which was called "Metropolis" rule. It transmitted from one state to another not only according to the fact that the new state was better than the older one, but it also transmitted if the new and worse state was "hit" by a function of random. It made it jump out of local extremum. All the states made up of a link, called "Markov chain". The advantage is its capability of global optimization. The procedure of searching of the state of optimization was controlled by many factors. So, applying of SA should obtain those factors' appropriate value. Those parameters interacted tightly and controlled the optimization procedure. In order to reduce SA's overhead, hierarchical structure of grid scheduling should be adopted. But scheduling algorithms in each level may be the same. It should not only take load of machines (CPUs) into account, but also network's load should be considered, especially for data intensive application.

The algorithm run as follows:

(1) Request and discovery resources and filter them, according to characters of resource and demand from task, performance;

(2) Extract tasks to be scheduled; the condition here is that the number of tasks extracted should not be greater than the number of resources;

(3) Retrieve dynamic information of each resource, then estimated completion time of the task and its communication;

(4) Call simulated annealing algorithms in grid computing;

(5) Distribute the task scheduled to target resources and update resources' information;

(6) Compute the number of returning tasks;

(7) If there is more work to be scheduled, return step (2)

(8) Repeat (1)-(7) until all tasks completed [5].

The Sample Simulation Network and deployed resources is as given in figure 3 below[5].

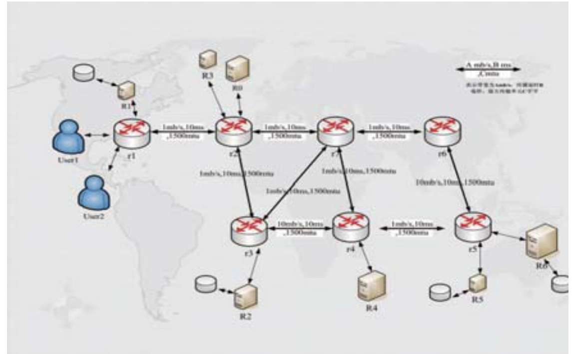

Figure 3: Sample Simulation Network and resources deployed

V. PRICING MODEL FOR GRIDS

Analysis of grid resources utilization from real grid trace data shows the feasibility of a financial option based model for pricing grid resources to attract more users for profitability for the grid provider while making the provision of good Quality of Service (QoS) to clients [6]. However, in the absence of the grid resource cost pricing scale, we simulate grid resources usage in order to justify our pricing model using GridSim toolkit. In this work a financial option based pricing model with GridSim framework was integrated and used as a grid simulation tool to price grid compute resources. Grid resources such as CPU cycles, network bandwidths, computing power or capacity, memory, disks present, throughput, processors, and various measuring and instrumentation tools are non-storable compute products. Pricing these compute products is challenging because of the specific characteristics of the grid resources: heterogeneity of resources (geographically dispersed ownership and time zones affects availability of resources) and volatility of resources since they exist as compute cycles).

The Algorithm:

1. Begin: GridSim;
2. Begin: Create grid scenario;
/* Create the environment scenario
and initialize the GridSim Toolkit*/
3. Start: for each grid resource do; /*Ri*/
4. Create new processing elements; /*PE*/
5. Create new machines;/*Mi*/
6. Create new resources Ri; /*where Ri have one
or more Mi that also have one or more PEs */
7. End: Create grid scenario
8. Begin: Create users' scenario
9. for each user do

10. create a grid task;
11. End: users' scenario;
12. Begin: bid and trinomial;
13. for each user do
14. for grid resource do
15. resource bid and utilization;
16. Apply trinomial;
17. compute option value;
18. End: resource use and trinomial;
19. Start GridSim simulation;
20. Obtain simulation data;
21. End GridSim simulation;
22. End:GridSim;

The work presented puts forward an important idea: feasibility of a financial option based model for pricing grid resources with emphasis to attract more users for cost recovering on grid infrastructure for the grid provider while there is uncompromised QoS for the users.

VI. QUALITY OF SERVICE

Quality of service is an important concept as it offers a basis for informing users about the sort of service they might expect. Often users will just wish to specify the time by which they would expect their job to be completed and would be happy to pay a predefined amount to achieve that. Other users may wish to reserve resources in advance for a particular time period. Others, who are more computationally aware, may wish to specify the services they expect more precisely through defining specific criteria. Currently, most Qualities of Services are defined as low level parameters that are, to the inexperienced computer user, vague and complex. Defining QoS at a higher level not only makes their measurement simpler but also helps users identify their requirements faster, more accurately and more realistically (Bhatti et al 2003; Rio et al2003, Albodour 2008 [7] . Another challenge is that of measuring Qualitative QoS. While Quantitative QoS such as bandwidth have standard measurement metrics that can be used, for example bandwidth can be measured by Mbps; Qualitative QoS such as reliability and availability do not and must therefore be defined with suitable metrics.

A. Quantitative QoS:

- Guaranteed number of Resources (Computational)
- Access period (Range of dates)
- Memory per core (in MB)
- Average power of single CPUs (in GHz)
- Storage required (in GB)
- Bandwidth Required (in Mbps)
- Short term storage requirements.
- Required time of completion (time deadline in hours)

B. Qualitative QoS:

- Resource availability (in %)
- Resource reliability (in %)

C. Other QoS considerations

Differing policies between different organizations and their resources have to be taken into account, as well as, authentication and authorization of users who may transcend their own domain into others in search of resources that meet their requirements for their applications, not to mention, the actual differing security domains within a single institutions. It is therefore vital for these considerations to be used in any simulation that tests an application that is to be run on top of a real Grid platform or system. It is for this purpose that we have distinguished multiple levels of users, meeting some of the access control demands of modern applications.

Varying QoS can be expressed and maintained through the identification a number of resource characteristics that affect performance. These types of characteristics can be used both to specify QoS and as metrics to measure performance against an agreed Service Level Contracts. Complications may arise if resources fail or if infrastructures across Grids differ and cause deterioration of service. In this case renegotiation of SLA may need to occur.
The essential extension of classes already present and the necessary introduction of new classes have been elaborated.

VII. GRID WORKFLOW SCHEDULING OPTIMIZATION

Workflow scheduling problem in grid environment optimises mainly based upon the time and cost constraints, but the main problems about, flexibility, stability, security and load balancing are not enough considered.

Redefining parameters of quality of service (QoS) and the model of grid workflow scheduling a rotary hybrid discrete particle swarm optimization (RHDPSO) algorithm has been put forward, in which double extremums are disturbed by the method of random time sequence based on rotation discretization, to control premature convergence and local optimum. The approach is used to generate an optimal solution so as to complete the multi-QoS constrained grid workflow scheduling as well as to utilize the resource in an efficient way. Empirical results reveal that the proposed approach can be applied for grid workflow scheduling. Simulation results prove that the RHDPSO algorithm has fast convergence, is very precise and is very robust, and can effectively restrain preterm convergence, As compared to DPSO [8].

VIII. Services Monitoring In Grid Market

A. Performance Metrics

The performance of Grid is not well defined, neither is the performance of Grid services. The obstacles are mainly the nature of Grid (heterogeneity, dynamism, wide distribution, etc) and these make traditional performance metrics not directly applicable to Grid environments. However, there has been pretty much work on Web services which can provide good references when we consider the performance features of Grids. In Grid market environments, the following metrics can be considered to monitor for Grid services:

- Availability: whether a Grid service is present and ready for immediate use. This is a very basic metric showing the status of a Grid service.
- Throughput: how many requests can be serviced in a given time period. Throughput can be an important concern by system administrators. The throughput could be limited by either the service itself of the system that it is running on.
- Latency (or response time): how much time elapses between the request and response is given by this. Response time is usually a concern by end users, and it can be affected by a lot of factors such as workload of the server, efficiency of the current service, network capability, actual degree of completion, and many more small factors.
- Scalability: How scalable the service is, in terms of number of service instances.

B. Other Metrics

There are many other concerns for Grid services, which may not be directly related to performance. The concerns could be from the view point of the consumer's commercial and non-technical considerations like:

- Service cost: Price of using the service. It should be provided by the service provider.
- Rate of failure: The failure rate reveals the number of failures for a given number of requests. This is one of the metric that should be monitored since the consumer needs to have an idea of how stable the service is, or how reliable the service is. The failure may not necessarily be caused by the service itself there can be other reasons like networks non fulfilment in communication.
- Rate of rejection: Sometimes the request for using the Grid services can be rejected. The reasons of rejection could be various: no agreement on the cost of the usage, unauthorized user, etc.
- Rate of request (or Popularity): The request rate reveals the popularity of the service to some extent. If other parameters are same, the higher the request rate, the more popular the Grid service is. This information can be used for marketing evaluation and other tasks.
- Service Guarantee: Service Guarantee tells consumer how high chance he can get the promised quality of service.

Liang Peng et al. proposed a Grid market framework to build commercial Grid market and support the commercialization of any Grid, for organizations to be able to share resources for profit. Such a framework is a must if the Grid is going to evolve from its current experimental status into professional or commercial use[9].

IX. Conclusion

Grid Computing finds its applications in the fields discussed above and provides a large scope for research in areas highlighted. From the survey we show how simulation has aided to ease in development of more efficient algorithms. Also, feasibility analysis of such projects can be tested through the use of the GridSim Toolkit.

Grid Computing forms the basis of Cloud environment creation and hence finds its way into applications of the future. Hence, commercially, Grids hold an essential position, especially in the field of e-commerce amongst other businesses.

REFERENCES

[1] Rajkumar Buyya and Anthony Sulistio, Service and Utility Oriented, Data Centers and Grid Computing Environments: Challenges and Opportunities for Modeling and Simulation Communities, Proc. of the 41st Annual Simulation Symposium (ANSS'08), April 13-16, 2008, Ottawa, Canada. Keynote Talk.

[2] Prof.K.K.Joshi, Prof. S.M. Ranbhise, ‘Simulation and Analysis of Cloud Environment’, International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014), vol. 2, issue 4, (Oct. – Dec. 2014)

[3] D.Nanthiya/ PG Student, P.Keerthika M.E.,/Assistant Professor (Sr.G), Computer Science and Engineering, Kongu Engineering College, Erode, India, ‘Load Balancing GridSim Architecture with Fault Tolerance’

[4] ‘Service and Utility Oriented Distributed Computing Systems: Challenges and Opportunities for Modeling and Simulation Communities’, Rajkumar Buyya, Grid Computing and Distributed Systems (GRIDS) Laboratory Department of Computer Science and Software Engineering University of Melbourne, Australia

[5] ‘Application of Simulated Annealing Algorithm to Grid Computing Scheduling based on GridSim’, Yuhua Guo, Xiaolin Wang, Department of Computer Science & Technology, Anhui University of Technology, Ma’anshan 243002,Anhui Province, China

[6] ‘Integrating a Financial Option Based Model with GridSim for Pricing Grid Resources’ David Allenotor, Ruppa K. Thulasiram, Department of Computer Science, University of Manitoba, Winnipeg, R3T 2N2, Canada

[7] ‘An Extension of GridSim for Quality of Service’- Reda Albodour, Anne James and Norlaily Yaacob, Distributed Systems and Modelling Research Group, Faculty of Engineering and Computing, Coventry University, Coventry, CV1 5FB, UK

[8] Qian Tao, Huiyou Chang, Yang Yi, Chunqin Gu, Yang Yu, Department of Computer Science, Sun Yat-sen University, Guangzhou, China- ‘QoS Constrained Grid Workflow Scheduling Optimization Based on a Novel PSO Algorithm’

[9] Liang Peng, Melvin Koh, Jie Song, Simon- ‘0-7803-9746-0/06’ Asia Pacific Science & Technology Center, Sun Microsystems, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798