Indian Journal of Science and Technology

New Roadmap for Elastic Grid Resource Matchmaking

R. Surendran[1*] and B. Parvatha Varthini[2]

[1]Research Scholar, Department of Information Technology, Sathyabama University, Chennai, Tamil Nadu, India; surendran.mtech.it@gmail.com
[2]Professor & Head, Department of Computer Applications, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India; parvathavarthini@gmail.com

Abstract

Dynamic grid computing possesses the power of several resources in one to solve the problems optimally at run time. Main focus of grid computing is on matchmaking and scheduling. The proposed system finds the resources which are suitable for a given job, from the available grid resources. A Elastic Grid design system is proposed to design the topology for a grid network, more number of the quality of service attributes used, optimal fuzzy scheduling, profitable resource allocation increase perfection of grid balancing, coordinated and optimized resource sharing, decrease the monitoring and migration work and detect the fault tolerance grid resource from the available resources based on Optimal Resource management Techniques Operations (ORMTO).

Keywords: Dynamic Grid Computing, Grid Resource Matchmaking, ORMTO, Fuzzy Scheduling, Grid Balancing.

1. Introduction

Dynamic grid computing is a process of share the grid balanced resources and services in a large-scale distributed network [1]. Grid computing major roles are resource selection is easily performed based on collection operations and allocation for a job via efficient scheduler [2]. Grid ranking is the process to rank the available grid nodes (resource) based on their QoS (Quality of Service) parameters (mainly time and resource consuming) [3], and then predict the best performing ones for a job [4]. Resource matchmaking is sophisticated technique to find the fault tolerance, perfect quality, best grid balanced resource selection to a correct valuable trust based job. Resource matching provides a pairing service between the service requester and the service provider. Resource matching should allocate the appropriate resource or a combination of resources for a task execution and also support user deadlines and economic constraints for scheduling optimizations [5]. Grid Resources are service, software, platform, infrastructure, supercomputer, Data base, virtual organization, virtual machine, clusters, server, personal computer, web cam, printer, scanner. Based on Quality of Service (QOS) parameters, sorts the particles (resource) then hash table use to format the job allocation, here resource ID is hash table index and key values are multiple jobs [6].

2. Related Works

Existing systems have a limited innovation for resource matchmaking process [7]. From toward optimal deployment of communication-intensive cloud applications [8], clustering-based cloud node selection process is executed. Here considers the relationship (i.e., high response time) between cloud nodes is an advantage and disadvantage is topological structure is not well defined [9]. Here optimal resource discovery and resource selection is an advantage and disadvantage is searching the resources not faster [10]. Scheduling

*Corresponding author:
R. Surendran (surendran.mtech.it@gmail.com)

phase is not follow the dynamic and intelligent grid component selection [11], need to improve the accuracy of grid ranking and improve the customer satisfaction and highly dynamic in nature, and dynamic grid load balancing [12] and fault tolerances technique service time is not good. Motivation of the proposed system is developed from the issues are to provide an enhanced grid balancing for job via accurate grid ranking and perfect service time and to design a well structured hybrid topology for fault tolerant grid resource selection service. Scheduler and the results reinforce its adequacy for dealing with the lack of accuracy in the estimation of communication demands so that scheduling decisions can lead to poor performance [13]. Imprecise input data impose special challenges to grid scheduling [14]. In existing methods are adaptable for homogeneous computing system and in all the times the scheduling leads to the overhead of monitoring and task rescheduling [15].

3. Proposed Work

3.1 Aim

To design a well defined hybrid topology for the grid network, more number of the quality of service attributes used, optimal fuzzy scheduling, profitable resource allocation, increase perfection of grid balancing, coordinated and optimized resource sharing, decrease the monitoring and migration work and detect the fault tolerance grid resource from the available resources.

3.2 Architecture

This work is to device an efficient model for resource matching in complex grid computing environment. Resource matching is a key task is any grid computing service and it involves various levels of complexity. In this paper, resource matching based on computational economics [16], where each resource is treated as a product in the economic market and a demand value is assigned to them. We also consider other criterion such as complexity, size and type of the task to identify is priority and need of resource [17]. Figure 1 represents the architecture for an elastic grid resource matchmaking process [7, 17]. Based on the priority of the task and the demand value of the resources the matching process goes on [18]. A high priority task is essentially matched with a resource of high demand as it indicates that resource is efficient.

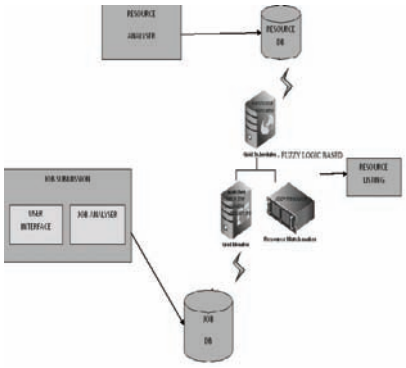

Figure 1. Architecture for an elastic grid resource matchmaking.

3.2.1 Resource Matchmaking

The user submits a resource specification to the resource broker, which returns a set of physical available resources matching the specification with one resource managers. Figure 2 describes the normal queue, priority queue and reservation queue jobs are organized by Resource matchmaking. Resource matchmaking sorts the particles (resource) then represented in the data base table. Data Base table use to format the job allocation, here resource ID is index and values are multiple job at average execution time. Resource Co-Allocation technique applied in the table for the purpose of simultaneous execution of same job in different resources.

3.2.2 Optimal Fuzzy Scheduling

Optimal Fuzzy Scheduling depends on application demands and resource availability. To produce effective results, schedulers need to take into account the difficulty in estimating the demands of applications [19]. A scheduler based on fuzzy optimization, both computational and communication demands are expressed as fuzzy numbers.

Figure 3 represents the architecture of the project, starting with the creation of Directed Acyclic Graph; the nodes are being extracted with the corresponding information and can compute the communication demands, host capacity, cost constraint using the fuzzy scheduler, which

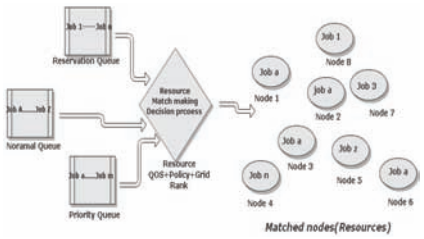

Figure 2. Resource matchmaking process.

can be able to determine the starting time of the task and data transfer time.

3.3 Modules

3.3.1 Job Submission and Evaluation

First the user uploads the job to which the resource is to be allocated in to the matchmaking engine.

- The analysis involves finding out the type of the job, size of the job and computational complexity of the job.
- This information is used in the matchmaking process to find if a system has the particular type of resource required for the job.
- The analyzed information is stored in a job database which is accessed by the matchmaker process.

3.3.1.1 Job Evaluation

- The first step is identifying the type and size of the job submitted.
- The type is identified by extracting the extension of the file and comparing it with a list of file types in the database. The second step is to calculate the lines of code uploaded job.
- This value is uploaded into the job database which is then used in assigning priority.

3.3.1.2 Job Database

- This database has two tables namely file type table and Job info table.
- The file type table has two fields, the extension and type field, where each extension has the type of name mapped to it.

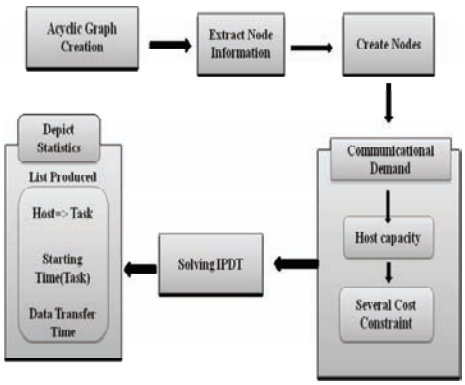

Figure 3. Optimal Fuzzy Scheduling.

- The job info table stores all the results of the job evaluation, which is user later for resource allocation.
- The job info table has the following fields, JobName, JobType, JobSize, JobLoC, JobPriority.

3.3.2 Available Resource Updating

The second module deals with the admin work of updating the resources those are available in the grid and ready to service the jobs. This is done through the resource entry interface. The admin enters the resource ID, name, proximity and type. The proximity value is the hop value of the resource from the central grid system. This proximity value is used in assigning resource to a job, jobs with low priority are assigned resources with lesser proximity value and jobs with high priority value are assigned resources at maximum proximity value, the medium priority jobs have an intermediate value. The job type is selected from a list of the jobs that the grid services.

The demand value is used to service jobs at a better quality. It helps assign high priority jobs to those resources that have a high demand value which in turn means a good quality resource.

- This phase involves obtained information about the resources available in the grid system.
- This information is stored in the resource database and is accessed by the matchmaker.

The second phase the resource matchmaking process is getting the available resources to which the job can be matched.

- Getting the available resource involves updating the resource database with the information regarding the resource.
- This work is carried out by the admin who enters the resource id, name and the proximity of the resource from the central system.

The resource info table contains the following fields, ResourceID, ResourceName, ResourceProximity, ResourceType, ResourceDemand.

3.3.3 Optimal Fuzzy Scheduling

The input is formulated via DAG with nodes (vertex) and edges. The edges are used to represent the processing demand of i^{th} and j^{th} task. Using graphics mode the graphs are constructed. Host is numbered from 0 to m.

As per graphs the related nodes are constructed using GRIDSIM tool. By extracting the node related attributes such as the edges are being connected. The graphs that are represented are called as grid topology and the dependencies among the task, the node connectivity's etc., are being found out.

3.3.4 Resource Matching

The final module is the actual resource matching process. Resource matching is a critical task in any grid computing system. It involves selecting the best suitable matching is done based on the conditions for each level of priority. We check for the proximity of the resource from the central grid system, then the job type with the type of service the resource handles and the demand value in case of high priority jobs. For low and medium priority jobs we check for the proximity value and service type match only. In case of low priority job we assign resources at a fixed minimum proximity, for medium priority a fixed average proximity value and for high priority job the maximum proximity value. For type match, we just compare the job type with the resource service type value. This value is used while looking for a resource for a high priority job, those resources that have a high demand value are assigned to high priority jobs and hence serving them with the best resource.

- The final process is matching the appropriate resource for the given job.
- Here the job analysis information from the job database and resource analysis information from the resource database are used.
- Primarily the closest resource to the client is matched. Whenever a resource is assigned it is assumed to be having a greater economic value and a counter is used to maintain this information.
- Based on this information job that arrive in the future can be assigned resourced based on their priority.
- To perform the matching process we make use of Job DB and Resource DB.
- The resource DB is populated with the available resources that are ready the service the jobs.
- Once the job is uploaded and the evaluated that value is used to calculate the priority of the job.

By default the jobs are assigned to the resources with the closest proximity value

- Every time a resource is utilized its demand value is increased, and hence when a high priority job needs to be serviced the demand value is considered and the resource with the highest demand is assigned.
- It is necessary to analyze if a particular resource provides service for that particular type of job.
- Job priorities are low, medium or high and the resources are assigned accordingly.
- The proximity value tells us how close the resource is to the central system, as the closes resource can be reached quicker
- Once all the criterions match a job is assigned a resource from the available resources.

4. Algorithm

Step 1: Resource detail

- Admin enters the resource ID, name, proximity and choose the type of service provided. Resource details updated into the database.

Step 2: Job submission

- The job engine identifies the following about the job. job size, LoC and job type.
- Priority value is calculated based on the following conditions.
  If Job size > sminmid and <smaxmid and LoC > Lminmid and < Lmaxmid, the priority if assigned as medium.
- If Job size > smin and LoC> Lmin, the priority is assigned as High.
  Else the priority is assigned as low.

Step 3: Resource Matching

- Resource matching is done based on following conditions.
  If priority is low and proximity <= plow and job type and resource service type match, the corresponding resource is matched.
- If priority is medium and proximity <=pmid and job type and resource type match, the corresponding resource is matched.
- If priority is high and proximity <=pmax and job type and resource type match, the corresponding resource is matched.
  Else No match found message is shown.Where,
- sminmid, smaxmid are minimum and maximum size values for medium priority job.

- Lminmid, Lmaxmid are minimum and maximum LoC values for medium priority job.
- smin, Lmin are minimum size and LoC values for high priority job.

Proximity is the hop distance of the resource from the central system.

5. Experiment

5.1 Grid Resource Registration in ORMT's Matchmaker

In java, resources are registered with all details in ORMT's matchmaker based on authorized ID, Name, proximity, type.

In Figure 4, resources are registered with all details in ORMT's matchmaker based on authorized ID, name, proximity, type. This is the available resource updating interface. The admin enters the information about a resource through this interface.

5.2 User Interface – Directed Acyclic Graph Creation

Based on Figure 5, The input is formulated via Directed Acyclic graph (DAG), which comprises nodes (as oval shape) and edges between the nodes. The edges are used to represent the processing demand of i^{th} and j^{th} task. Using graphics mode the graphs are constructed and the Host is numbered from 0 to m.

5.3 Initializing All Grid Nodes and Mapping the Tasks

In Figure 6, as per graphs the related nodes are constructed using GRIDSIM tool. By extracting the node related attributes such as the edges are being connected. The graphs that are represented are called as grid topology and the dependencies among the task, the node connectivity's etc., are being found out.

5.4 Identifying the Scheduling Task

Here the host capacity and the available bandwidth of the nodes are being estimated and compute the time of data transfer and the processing time of the task and represent the Time taken by the k^{th} host to execute single instruction and Evaluate QoI the quality of information index. Figure 7 shows: QoI represents the level of confidence of demand estimation.

5.5 Calculating the Limits and Uncertainty and Lists Generation

Based on Figure 8, The IPDT-Fuzzy scheduler is based on a fuzzy optimization formulation. This formulation is then transformed into a crisp equivalent model based on an integer programming formulation. The output of the

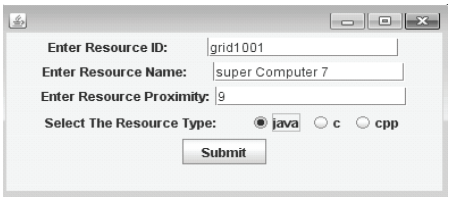

Figure 4. Grid Resource Registration.

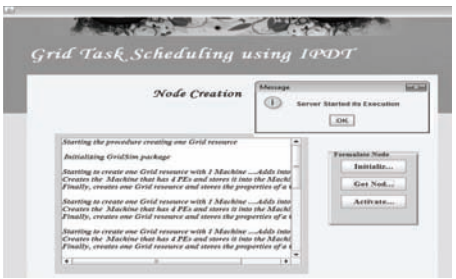

Figure 5. Create a grid network from DAG.

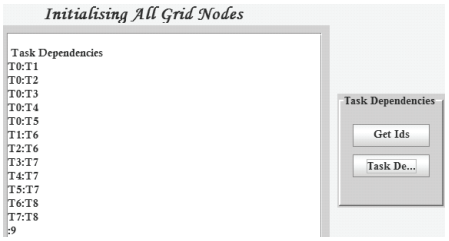

Figure 6. Initializing all grid nodes.

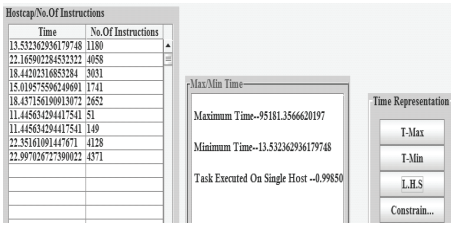

Figure 7. Scheduling the task.

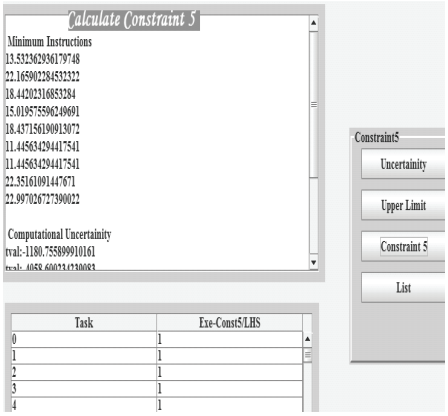

Figure 8. Calculate the constraint.

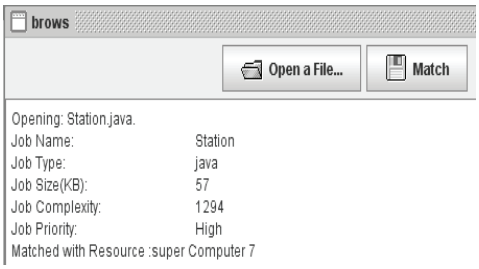

Figure 9. Predict the perfect resource.

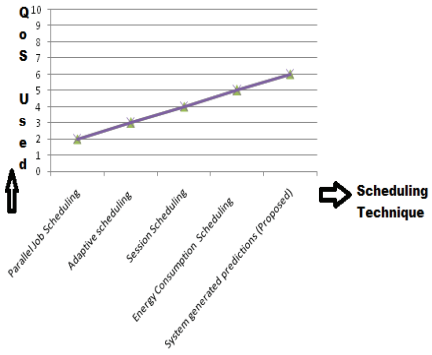

Figure 10. Proposed Scheduling algorithm.

IPDT-Fuzzy scheduler is a list which provides information about the host on which each task should be executed, the starting time of that task, and the time when data transfer should take place.

5.6 Browser Predict the Perfect Resource from ORMT'S Matchmaker

Browser predicts the matched resource from ORMT'S matchmaker for their job. This window shows the information about the job such as name, type, size, complexity and priority. To match the uploaded job, we click on the match button, and the system displays the resources that a best matched to that specific job. In Figure 9, Browser predict the matched resource from ORMT's matchmaker for their job.

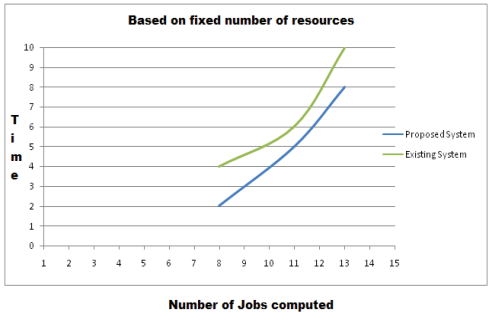

Figure 11. Proposed system comparison.

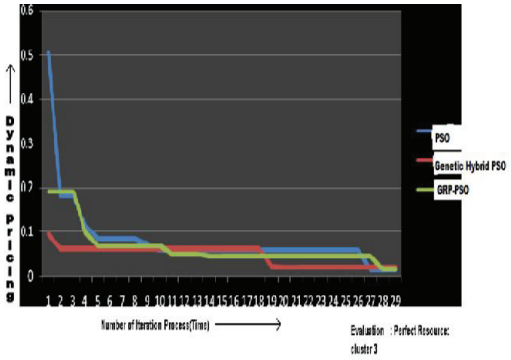

Figure 12. Proposed best node selection.

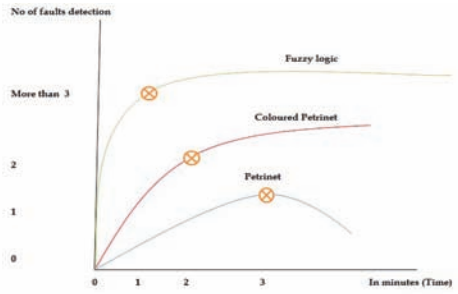

Figure 13. Proposed fault detection.

5.7 Comparison Chart for ORMT

Scheduling the resource co-allocation request based on system generated predictions through Discovery service and Priority based on fairness with user history. In Figure 10, The proposed scheduling algorithm support greater number of QoS parameters. Such as, time, energy, user history, grid knowledge necessity, sharing, and availability.

In Figure 11, The proposed system compare with existing system for job completion from time factors. In the proposed system, minimize time more number of job completion. Figure 12 discuss the best node selection algorithms based on price.

Figure 13 refer the fault detection techniques based on Iteration and member function.

PetriNet	Coloured PetriNet	Proposed Fuzzy Logic
Only one fault detection at a time	More than one fault at a time	More than 3 faults at a time
Cannot use in Real system	Cannot use in Real system	Can use in Real System

6. Conclusion

Grid balanced resource matchmaking process achieve coordinated and optimized resource sharing, enhanced security management, cost optimizations. Improved performance substantially, backup plan in case the system fails even partially, maintain the system stability, accommodate future modification in the system and avoid a situation where some nodes are heavily loaded while others are idle or doing little work. Resources can be filtered by the reservation manager and freely available matches can be considered for matchmaking.

7. References

1. Joseph J, and Fellenstein C (2003). Grid Computing Book, PHI, Chapter 1, Prentice Hall PTR Publisher, 1–7.
2. Barzegar B, Esmaeelzadeh H et al. (2011). A new method on resource management in grid computing systems based on QoS and semantics, Indian Journal of Science and Technology, vol 4(11), 1416–1419.
3. Xiaoshan H, Sun X et al. (2003). Qos guided min-min heuristic for grid task scheduling, Journal of Computer Science and Technology - Grid Computing, vol 18(4), 442–451.
4. Naik V K (2007). Prediction based resource matching for grid environments, US 2008/0172673A1.
5. Wu M, and Sun X (2003). A general self-adaptive task scheduling system for non-dedicated heterogeneous computing, 2003 IEEE International Conference on Cluster Computing, 2003 Proceedings, 354–361.
6. Agarwal A, and Kuma P (2011). Multi dimensional QOS oriented task scheduling in grid environment, International Journal of Grid Computing & Applications, vol 2(1), 28–37.
7. Qiang G, Heng-wei Z et al. (2011). A grid resource matching algorithm, Computer Science and Network Technology, International Conference, 142–145.
8. F Pan, Wang J et al. (2011). Toward optimal deployment of communication-intensive cloud applications, 2011 IEEE International Conference on Cloud Computing (CLOUD), 460–467.
9. Chtepen M, Claeys F H A et al. (2009). Adaptive task check pointing and replication: toward efficient fault-tolerant grids, IEEE Transactions on Parallel and Distributed systems, vol 20(2), 180–190.
10. Nordin M I B, Abdullah A B et al. (2011), Goal-based cloud broker for medical informatics application: a proposed goal-based request and selection strategy, International Conference on Telecommunication Technology and Applications, 35–40.
11. Blythe J, Jain S et al. (2005). Task scheduling strategies for workflow-based applications in grids, Cluster Computing and the Grid, 2005, CCGrid 2005, IEEE International Symposium, vol 2, 759–767.
12. Ning X, and Shaohua Y (2013). A load-balanced crosspoint-queued switch fabric, China Communications, vol 10(2), 134–142.
13. Sadasivam G S, Rajendran V V (2009). An efficient approach to task scheduling in computational grids, International Journal of Computer Science and Applications, Techno Mathematics Research Foundation, vol 6(1), 53–69.
14. Batista D M, Drummond A C et al. (2008). Scheduling grid tasks under uncertain demands, SAC '08 Proceedings of the 2008 ACM symposium on Applied computing, 2041–2045.
15. Nazir B, Hassan M F et al. (2009). Adaptive task scheduling strategy for economy based grid, International Symposium on Computing, Communication, and Control, 164–169.
16. Wolski R, Brevik J et al. (2004). Grid resource allocation and control using computational economies, 1–27, Available from: http://www.cs.ucsb.edu/~rich/publications/gc-book.pdf
17. Brooke J, Fellows D et al. (2004). Semantic matching of grid resource descriptions, Lecture notes in computer science, vol 3165, 240–249.
18. Hu Z, Hu Z et al. (2010). A service-clustering-based dynamic scheduling algorithm for grid tasks, International Journal of Grid and Distributed Computing, vol 3(3), 53–66.
19. Rao K R M, Ramachandram S et al. (2011). A reliable distributed grid scheduler for independent tasks, International Journal of Computer Science, vol 8(2), 213–223.