International Journal of Electronics Communication and Computer Engineering
Volume 2, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

# A Bee Optimization Algorithm for Scheduling a Job Dynamically in Grid Environment

**M. Prakash, P. Rajeswari**

***Abstract*— Grid computing is based on large scale resources sharing in a widely connected network. Grid scheduling is defined as the process of making scheduling decisions involving allocating jobs to resources over multiple administrative domains. Scheduling is the one of the key issues in the research. Matchmaking is a key aspect in the grid environment. Matching a job with available suitable resources has to satisfy certain constraints. Resource discovery is one of the key issues for job scheduling in the grid environment. The proposed Bee optimization algorithm is to analyze Quality of Service (QoS) metrics such as service class, job type in the heterogeneous grid environment. QoS parameters play a major role in selecting grid resources and optimizing resources effectively and efficiently. The output of the proposed algorithm is compared with max-min and min-min algorithm.**

*Keywords* **—Bee Optimization, GIS, Resource Broker, Scheduler, Scheduling**

## I. INTRODUCTION

With the abrupt development of networking technology, grid computing, which enables large-scale resource sharing and alliance, has embarked as a promising distributed computing presumption. It is a standard and software for sharing of remote resources and concert. Mainly used for high performance computing jobs. It delivers non diminutive qualities of service such as response time, throughput, and availability. Grid is a utility or infra-structure for complex, huge computations, where remote resources are amicable through web (internet), from desktop, laptop, mobile phone. The most obvious resource included in a grid is a processor, but grids also subsume sensors, data-storage systems, applications, and other resources. A few machines in a department to groups of machines organized as a hierarchy are composed to form a grid that can be built in all sizes. The machines that have the same hardware architecture and same operating system, are connected on a local network, form a simple grid. This type of simple grid uses homogeneous systems, and may be used for specialized applications. Because the machines have the same architecture and operating system, choosing application software for these machines is usually simple. The next movement would be to include heterogeneous machines. The grid may grow to be hierarchically organized to reduce the disputation implied by central control, increasing scalability. The consequences of grid computing are security, fault tolerance, scheduling. The basic security components are comprised of mechanisms for authentication, authorization, and confidentiality of communication between grid computers. Security challenges in a grid environment are Integration, Interoperability and Trust Relationship.

The phases of scheduling are resource discovery, system selection, job scheduling. In resource discovery, scheduling interaction involves determining which resources are available to a given user. It involves selecting a set of resources, information gathering. This phase is done in three steps: authorization filtering, job requirement definition, and filtering to provide the minimal job requirements. In System Selection phase, for schedule the job, a single resource must be selected, given a group of possible resources. All of the resources will provide the minimum requirements for the job. This selection is generally done in two steps: gathering detailed information and making a decision.

In Job scheduling, select best suitable machines in a grid for user jobs. By taking static restrictions and dynamic parameters of jobs and machines, scheduling system generates job schedules for each machine in the grid. It ascertains proficiency of grid. Steps in Job Scheduling are Job Submission, Preparation Tasks, Monitoring Progress, Job Completion, and Clean up tasks. In Job submission, once resources are chosen, the application can be submitted to the resources. The preparation stage may involve setup, staging, claiming a reservation, or other actions needed to prepare the resource to run the application. The progress of users and their application are monitored, and possibly change their intuition about where or how it is executing. By consecutively querying the resource for status information, monitoring is done. Reschedule the process, if a job is not making competent work up. When the job is finished**,** the user needs bulletin message. Users generally do the clean up tasks after a job is run, or by including clean-up information in their job submission scripts.

The rest of this paper is organized as follows. In section 2, we briefly discuss literature survey. Architecture is described in section 3. Algorithm is described in section 4. Section 5 concludes the paper.

## II. RELATED WORK

Matchmaking has been an issue of Multi Agent Systems to find a suitable agent for a specific dispute. Resource discovery, resource allocations are to be issues in grid computing. Service discovery is a traditional method. It includes name matchmaking and keyword matchmaking [1]. Matchmaking in Multi Agent Systems involves semantic service matchmaking using concept relationship and distance of word to determine semantic similarities of advertisements and requests. Matchmaking [11] in Multi Agent Systems does not involve other resource types and matchmaking results are exact that is only “false” and “true” are allowed. The grid matchmaking process involves three types of agents namely requesters, providers, and a matchmaking service. A new approach to matchmaking service executes a matchmaking algorithm and returns a set of ranked resources to requester. Requester chooses a resource from the set and contacts the corresponding resource provider. This improves the efficiency of matchmaking without sacrificing the quality of matchmaking results.

Task scheduling is mapping of tasks to selected group of resources which may be distributed in multiple administrative domains. Objective is to execute user's task request as early as possible. A particular task found an available resource on some agent, but the resource had been found by some other tasks and state of the resource was not modified in a period of time, then the resource was actually unavailable. Agent Request Table is the new technique [2], it contains agent ID, task information. This table is to assist the discovery process that occurred by the submission of idle resource. By using this table, task can be completed as quickly as possible.

An enhanced ant colony optimization algorithm is used for jobs and resources in grid computing. By effectively taking advantage of large amount of distributed resources, minimize execution time of computational jobs [3]. Local pheromone trail update and trail limit values are issues. Stagnation in grid computing occurs when all jobs require or are assigned to the same resources and due to high workload of resources. Algorithm has been implemented in grid system architecture, which consists of four main components namely the grid information server, grid resource broker, jobs and resources. Value of pheromone updated on each resource is controlled and ensures that already assigned optimal resources will not be taken for newly submitted jobs are the advantages of this technique. Completion time for each job is minimized by using ant colony algorithm.

Multiple Criteria Resource Selection algorithm is developed for grid computing environment. Scheduler selects resource that gives shortest time for job completion for each job, including time for file staging, resource queue waiting and job execution. A new approach in selecting a resource for job execution is proposed. Multiple criteria like network bandwidth, workload, processing power are considered to select a resource for job execution [4]. Performance is compared with single criteria resource selection algorithms. Resource selection methods based on any one criteria is insufficient for best decision, because resources are heterogeneous and harnessed by different administrative domains. Multiple Criteria Resource Selection algorithm maintains its superiority and selection of optimal resource over many single criteria resource selection algorithms.

A Quality of Service (QoS) aware Workflow Management System ensures workflows that meet the predefined QoS requirements and optimize the requirements accordingly. Workflow management systems have been used to support various types of e-science workflows on grid. Business Process Execution Language and Web Service-Choreography are powerful workflow languages for developing workflows based on web services [5]. Complication occurs due to dynamic nature of resources appearing and disappearing without notice and the load on these resources changing over time. For interactions between grid components to enable resource oriented workflows, QoS constraints are used. Workflow management Systems should be able to accept, monitor, and control the QoS provided by users. Two approaches are used such as loose (soft) guarantees, and strict (hard) guarantees. Loose guarantees are suitable to applications that can perform adaptation and self-healing operations to cope with requirements that are not met in practice. Strict QoS requires certainty of delivery of prescribed level of service, involving both client and the service provider. QoS provisioning relies on both strict and loose QoS guarantees. Workflow management Systems provides a level of QoS through resource selection from prior information along with the use of advanced reservation.

A hybrid approach to the load balancing of sequential tasks under grid computing environments is proposed. Task allocation and load balancing represent a common problem for most grid systems. A centralized load balancing approach can function either based on averages or instantaneous measures according to type of information on which load balancing decisions are made [6]. Average based schemes achieve a thorough and holistic load balancing in handling sequential tasks in computing grids and it affects total task execution time which degrades overall system performance. Instantaneous measures can yield an immediate and direct decision for newly arrived task without a prior knowledge but it might perform agitatedly for sequential task load balancing due to lack of an overall knowledge of the tasks to be scheduled. A new approach is developed to combine the best features of both measures to form the hybrid approach. This approach focuses on using a combination of static and dynamic load balancing strategies. Two algorithms are used in this approach. Combine a First-Come-First-Served algorithm with Genetic algorithm. The First-Come-First-Served reduce system response time, Genetic algorithm aims at desirable load balance across all nodes in a computational grid. This gives better performance than its counterparts.

### *A. Proposed Work*

Bee Optimization Algorithm analyzes job type, service class in grid environment. Service class consists of QoS factors such as processor capability, system memory, storage capacity, bandwidth, and delay. A user who has a higher service class level can utilize better resources. A matchmaking service uses a matching algorithm to evaluate a matching function. The output of the algorithm is compared with max-min and min-max algorithm for better performance.

International Journal of Electronics Communication and Computer Engineering
Volume 2, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

## III. GRID ARCHITECTURE FOR JOB SCHEDULING

The Grid users are not granted to submit the job directly to the grid resources. Initially they have to register in the grid portal. Only the authorized or authenticated users are permitted to submit the job. After getting registration, user has to be checked for authentication. Authenticated grid user details are registered in the User Resource Information database (URIDB).

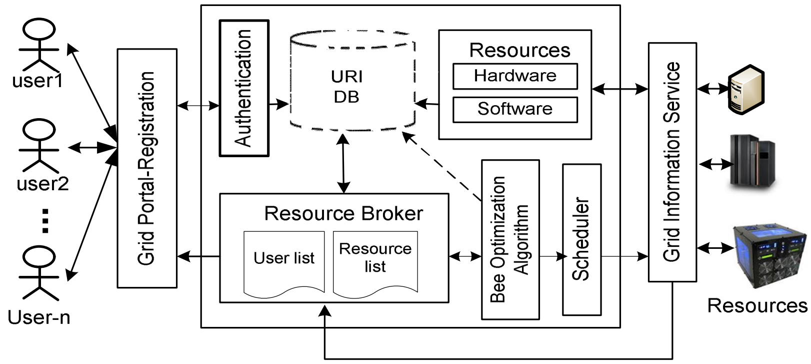


Fig.1.Bee Optimization Architecture for Grid Job Scheduling

Grid Information Service [8] checks for availability of resources. Job scheduling architecture is shown in Fig.1. User Resource Information Database separates the user list & resource list. Resource broker matches the user requirements with resource availability [10]. If it matches, Bee Algorithm had done matchmaking[11] process. If it doesn't match, resource broker intimate to user. Scheduler dispatches the job via. Internet. Resource Broker got the status of job details from Internet. QoS metrics such as service class, job type, deadline type are optimized. Service class consists of QoS factors such as processor capability, system memory, storage capacity, bandwidth, and delay. A user who has a higher service class level can utilize better resources. In Job Type [9], the job can be classified into 2 categories; indivisible job and divisible job. The indivisible job could not be divided into subjobs because those have strong relations among sub-processes. Divisible job can be considered as a task set of sub-jobs with no relation. This Service class and Job Type parameters are taken into account for comparison with max-min and min-min algorithms.

## IV. BEE OPTIMIZATION ALGORITHM FOR JOB SCHEDULING

The job scheduling is done by using Bee Optimization Algorithm (BOA). The BOA is discussed below

1. Initialize total number of resources randomly.
2. Scrutinize fitness of total number of resources.

$$\text{Fitness}(P_i) = \frac{1}{\text{makespan}} \times \text{Average Utilization}$$

$$\text{Utilization}(P_i) = \frac{\text{Completion time}(P_i)}{\text{makespan}}$$

Where P denotes the processor of the system (average utilization calculated based on individual performance of processor)

3. While (stopping criterion not met)
4. Select Grid sites for neighbourhood search
5. Procure users for selected grid site (more Number of users for best grid sites) & interpret fitness.
6. Select suitable user from each grid site.
7. Assign remaining user to search randomly & scrutinize fitness.
8. End while.

The resources available in the various grid sites are to be initialized. Available resources may be either hardware nor software. Depends upon the availability of the resources evaluate fitness [7] of total number of resources. Calculate makespan as the difference between completion time and start time. Utilization of the processes is calculated and estimates the average. Select the grid sites for neighbourhood search. Procure grid users for selected grid sites, more number of users for best grid sites and evaluate fitness. Select suitable users form each grid site. Scrutinize fitness for remaining user. Repeat the process until stopping criterion not met. At the end of this optimization process the exact fitness of the resource is found then the scheduler is responsible to dispatch the job to matched resources in the grid.

## V. CONCLUSION AND FUTURE WORK

Grid users access the memory as a resource. The grid users enter the data and timer interval is setup. For each user, memory access table is obtained. After execution, user-read table and user-lookup table are updated. User-read table gives the details about the user data. User-lookup table gives details about the memory location of grid user data. In the next phase, processor, CPU resources will be simulated. Matchmaking will be done using Bee algorithm. QoS parameters such as service class, job type should be optimized. To achieve better performance, comparative study of Bee algorithm with max-min, min-min algorithms will be performed.

## REFERENCES

[1] Xin Bai, Han Yu, Yongchang Ji, and Dan C. Marinescu, "Resource Matching and a Matchmaking Service for an Intelligent Grid", World Academy of Science, Engineering and Technology, pp: 104-107, 2005.

[2] Ding S. L., J. B. Yuan and J. B., "An algorithm for agent-based task scheduling in grid environments." International Conference on Machine Learning and Cybernetics, pp: 2809-2814, 2004.

[3] Ku Ruhana Ku-Mahamud, Husna Jamal Abdul Nasir, "Ant Colony Algorithm for Job Scheduling in Grid Computing", Fourth Asia International Conference on Mathematical/Analytical Modeling and Computer Simulation, pp: 40-45, 2010.

[4] Malarvizhi Nandagopal and Rhymend Uthariaraj, "Performance Analysis of Resource Selection Algorithms in Grid Computing Environment", Journal of Computer Science, vol.7, no. 4, pp: 493-498, 2011.

[5] L Guo, A S McGough, A Akram, D Colling, J Martyniak, M Krznaric, "QoS for Service Based Workflow on Grid", London e-Science Centre, Imperial College London, London, UK.

[6] Yajun Li, Yuhang Yang, Maode Ma, Liang Zhou, "A hybrid load balancing strategy of sequential tasks for grid computing environments", Future Generation Computer Systems 25 (2009), pp: 819-828.

[7] Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote, "Order-Based Fitness Functions for Genetic Algorithms Applied to Relevance Feedback", Journal of the American society for information science and technology, pp: 152–160, 2003.

[8] Karl Czajkowskiy, Steven Fitzgeraldz, Ian Fosterx, Carl Kesselman, "Grid Information Services for Distributed Resource Sharing", IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), pp:1-14, 2001.

[9] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, Prachet Bhuyan, "A Survey of Job Scheduling and Resource Management in Grid Computing", World Academy of Science, Engineering and Technology, pp: 461-466, 2010.

[10] Jun Zhang, Chris Phillips, "Job-Scheduling with Resource Availability Prediction for Volunteer-Based Grid Computing", London Communications Symposium 2009.

[11] M Prakash, T Ravichandran, "Resource Based Scheduling Approach for Grid Environment" International Conference on Future Engineering Trends, pp.296-299, April 2009.

## AUTHOR'S PROFILE

### M.Prakash

received the B.E Degree in Computer Science and Engineering from University of Madras and M.E Degree from Sathyabama University, Chennai, India. He is pursing PhD in Faculty of Computer Science Engineering, Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India. Presently he is working as a Assistant Professor in the Department of Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, India. His research area includes Grid Computing and Distributed Computing. He published around 10 papers in National and International conferences and journals. He is the life member in professional society of ISTE and CSI.

### P. Rajeswari

has completed her B.E in Information Technology at P.S.R Engineering College affiliated to Madurai Kamaraj University, India. Currently she is pursuing her M.E. in Computer Science and Engineering at Rajalakshmi Engineering College, affiliated to Anna University of Technology, Chennai, India. She has one year experience in teaching. Her areas of interest include Grid Computing and Computer Networks.