SECURITY DRIVEN MULTI-CONSTRAINED SCHEDULING FOR COMPUTATIONAL GRID

N. V. Keerthana[1] and Dr. P. Suresh[2]

[1]PG Scholar, Department of Information Technology, Kongu Engineering College, Tamil Nadu, India
email: nvkeerthu@gmail.com

[2]Assistant Professor, Department of Information Technology, Kongu Engineering College, Tamil Nadu, India
email: sureshme@gmail.com

Abstract

Grid computing is a collection of heterogeneous, dynamic resources from multiple administrative domains which are geographically distributed that can be applied to achieve a common goal. Development of resource provisioning-based scheduling in extensive distributed environments like grid computing brings in new requirement challenges that are not being considered in traditional computing environments. Computational grid is applying the resources of many systems in a pervasive network to a single issue at the same time. Grid scheduling is the method by which work specified by some means is assigned to the resources that complete the work in the environment which cannot satisfy the user requirements considerably. The resources may leave the grid network at any time resulting in occurrence of faults and vulnerability. The satisfaction of users while providing the resources might increase the beneficiary level of resource providers. Resource scheduling has to meet the multiple constraints specified by the user. The selection of resource with the satisfaction of multiple constraints is the most tedious process. This problem is resolved by introducing the particle swarm optimization based heuristic scheduling algorithm which attempts to select the most suitable resource from the set of available resources. The main parameters that are considered in this work for selecting the most suitable resource are the makespan and cost. The experimental result proves that the proposed method yields optimal scheduling with the satisfaction of all user requirements.

Keywords: Resource scheduling, PSOH, Fault tolerance, security

1 INTRODUCTION

Grid computing has come forth as a computing platform to offer a huge measure of resource sharing to extensive scientific applications. Computational grids can be defined as an environment that organizes geographically distributed and heterogeneous resources in various administrative domains with different security policies into a single computing system. Due to heterogeneous and dynamic nature of the resources in the grid, resources are inexorably unreliable, which causes a large effect on scheduling [6]. So, grid resource management has become one of the most important key concerns in the area of grid computing [7, 8]. A lot of heterogeneous services are used in order to create the grid and, in addition, these services are not managed by a single person but by different administrators in each of the organizations. For the efficient operation of a computational grid, various factors must be considered as load balancing, resource sharing, failure rate, security and efficient scheduling.

Grid resource management can be defined as deployment of necessary requirements of the resources, matching resources to the various applications followed by resource allocation and scheduling as well as monitoring the grid resources finally to run grid applications as efficiently as possible [5]. A grid resource management system is required to take decisions which include provisioning of resources and scheduling by providing Quality of Service (QoS) metrics delivered to the users. These metrics introduce a bit of challenging events that require to be addressed such as provisioning-based resource scheduling and assuring the scheduling of jobs on the trustworthy nodes. Resource provisioning is needed before scheduling the resources or execution of grid applications as Grid user often experiences a limited control over grid resources and the resource manager is not always able to satisfy all the requirements due to the large number of resources as well as user's requests.

Grid scheduling is defined as the process of assigning jobs to resources over multiple administrative domains [4]. The objective of this process is to maximize various optimization criteria such as machine utilization, user satisfaction and flow time or to guarantee non trivial QoS. Scheduling process should be flexible and fast so that it is able to efficiently react to dynamic changes in the grid environment (job arrival, failure, accurate job runtime estimate, etc.). Finding the favorable solution is a Nondeterministic Polynomial time (NP) -complete problem [4] which is practically difficult for larger sets of jobs. Therefore, a heuristic approach is to search for suboptimal solutions, where sufficiently systematic algorithms exist. However, schedulers usually use very simple algorithms based on priority queues. Due to the limited and often broad information used by these algorithms their efficiency is often very low, especially when sufficient QoS is required.

The NP-complete problems are often solved by heuristic methods. This approach can easily be applied to grid scheduling problems because grid scheduling consists of various significant events such as heterogeneity of resources, the dynamic and independent nature of grid resources and ultimately the subject of different policies being followed by the resource schedulers and users for execution of their tasks. The main motive of this work is to propose a heuristic-based scheduling algorithm being able to be applied in grid environment and hence scheduling the resources to the preferred jobs leading to the delivery of optimum results to the grid users. Due to dynamicity of the resources, the constraints such as failure rate and security were considered.

Our contributions in this paper are: 1) User authentication. 2) Workload and resource monitoring. 3) Multi-constraints strategy. 4) Particle Swarm Optimization based Heuristic (PSOH) scheduling algorithm. 5) Result analysis.

2 RELATED WORK

Resource provisioning and scheduling are major things of computational grids and help it achieve high performance in its execution environment. Due to the heterogeneous and dynamic nature of grid resources, this basically has taken the form of a large-scale optimization problem. The purpose of encompassing security is to prevent unauthorized disclosure of data.

He Xiao Shan et al. [9] discussed about the min-min heuristic for grid task scheduling. To improve the significant performance of the system, the new QoS guided min-min heuristic algorithm is implemented. This model addressed with the match of QoS request from the application and QoS provided by the diverse resources in the grid. It considers only the bandwidth constraint for the task scheduling. Since, the efficiency and the utilization is maximized, the security constraint is not addressed.

The Fuzzy Particle Swarm Optimization (FPSO) algorithm [2] dynamically generates an optimal schedule so has to complete the jobs within a minimum period of time as well as utilizing the resource in an efficient way. The position and the velocity of jobs are represented in fuzzy matrices. The dynamic availability of resources and the cost of data transmission are not considered. So it allocates more memory due to static allocation of resources.

The meta-heuristic technique [10] improves the performance of the system, an intelligent ant has been used and a new concept called ant level load balancing is implemented. A new ant is created when the system is overloaded. So it allocates more memory based on the decision making algorithms and the efficiency of the system decreases due to the creation of many new ants.

Hesam Izakian et al. [1] represented a Novel Particle Swarm Optimization (NPSO) algorithm for grid scheduling. This method minimizes the makespan and flowtime simultaneously by regularly updating the resources status in the grid information system. Thereby it yields a better result than FPSO by dynamically scheduling the tasks in the distributed grid environment. This method does not deal with the constraints such as user expected cost and security.

Visalakshi et al. [13] dealt with Hybrid Particle Swarm Optimization (HPSO) method for solving the Task Assignment

Problem (TAP) which is an NP-hard problem. It has been developed to dynamically schedule heterogeneous tasks on to heterogeneous processors in a distributed environment. The nature of the task is independent and non-pre-emptive. The HPSO yields a better performance than the normal Particle Swarm Optimization (PSO) when applied to the task assignment problem and it also compared with Genetic Algorithm (GA).

Rajni Aron et al. [3] constructed a resource provisioning framework based on QoS parameters. A resource provisioning policy based on QoS parameters is required for efficient grid resource provisioning. The motivation of this method stems for the challenges in managing and an efficient utilization of grid resources. The constraints including QoS satisfaction, minimum cost and minimum execution time are addressed by this method. The performance of the grid is better by considering the average response time as a metric that is not considered in this framework.

Rajni Aron et al. [4] proposed a hyper-heuristic approach for resource provisioning based scheduling in grid environment. This method presents an efficient strategy for secure scheduling of jobs on appropriate resources. This hyper-heuristic scheduling algorithm has been designed to schedule jobs effectively on available resources without violating any security norms. This algorithm provides reduced execution time, increased security and reliability. The dynamic allocation of the resources and the resource monitoring is not carried out by this method.

In grid computing, job scheduling is a complex task. A good scheduling system must optimize QoS parameters such as cost, utilization time and security. The selection of services with minimum cost and security is complex and finding the best pair of tasks and resources is an NP-complete problem [4]. To overcome these challenges, various methods for scheduling are introduced in the existing systems. But they have some problems like dynamic allocation of resources and efficiency of the system, less execution time, load balancing, minimum cost, minimal security and minimum makespan which are needed to be considered in the proposed system. Our proposed concept of heuristic-based resource scheduling algorithm minimizes the cost and makespan simultaneously. Also, particle swarm-based heuristic has not been applied for resource scheduling in the grid earlier and we are offering a fresh style while designing our resource scheduling algorithm.

3 PROBLEM DEFINITION

Grid resources are heterogeneous and dynamic in nature and the concept of job scheduling is the tedious task. So, an efficient scheduling should be carried out. Scheduling system must optimize QoS parameters such as cost, utilization time, security and makespan. Selection of services with minimum cost and security is more complex in grid. To formulate the problem, the set of *n* resources are considered as $\{r_1, r_2 \dots r_n\}$ and the set of *m* jobs are considered as $\{j_1, j_2 \dots j_m\}$. The jobs are considered as atomic and independent and can be executed in parallel with other available jobs. Scheduling function is defined as mapping of jobs to resources as: $S = J \rightarrow R$.

4 PROPOSED SYSTEM

In the proposed system, heterogeneous grid resources are considered for the allocation. The set of jobs would be gathered from the user along with their constraints. The multi-constraints are gathered from the user which in terms led to the successive execution of the set of jobs. Based on these requirements job scheduling would be done. Grid scheduling being a two-step process as follows: 1) Identifies the required set of resources as per the user requests. 2) Maps the jobs to the actual set of resources. Thus the scheduling ensures a near-optimal satisfaction of QoS parameters. Accordingly, the user now selects the most appropriate resources among all the resources shown that matches the user's budget and the other specifications.

4.1 User Authentication

The primary components of grid environment are grid user, grid resources, resource broker and GIS. Initially, the grid user interacts with the resource broker and sends their task to computation. So, the number of gridlets (jobs) is created which is get from the user. It contains all the information about jobs like length of the job, file size and output size.

Then resource broker locates the resources for scheduling strategies and task processing. So, the number of heterogeneous resources with dynamically varying cost, Central Processing Unit (CPU) available, CPU speed and memory available are created and those resources are enrolled on to GIS, where GIS worked as an agent and collect all the relevant information such as resource availability, node capacity and provide it to the resource broker to make the scheduling decision.

Authentication is the mechanism of checking the identity of the user to avoid the attempts of user to alter the system resources or affect the operation [11]. Authentication is implemented by entering userID and password before entering into grid by the user. Then the authentication server audits with its database whether the user credentials were proper. If it matches, the system allows the user to submit their jobs for scheduling.

The "RSA" algorithm is the public key cryptosystem. The maximum size of data that can be encrypted by RSA is 245 bytes, making ideal for securing information and also fast, free alternative to existing encryption algorithms.

The original RSA scheme is described in [12]. The key encryption process is done by RSA small-e encryption algorithm [14]. It is similar to original RSA except the public exponent. After the generation of keys, the resource provider wants to identify the users. Hereby the authentication plays a vital role.

For authentication process modified Diffie Hellman algorithm id used. The computed value by resource provider and user does not matches the grid user will be declared as unauthorized user.

4.2 Workload and Resource Monitoring

Resource monitoring would be done periodically to update the most current situation of every resource. While monitoring the grid resources, information like memory, CPU and the processing capacity of the users would be updated periodically [4]. The event monitoring system would find the states of all machines and that will be stored in the database for the final efficient allocation process. The expected cost to schedule each job on the resource is assumed to be supplied by the user.

4.3 Multi-Constraints Strategy

In grid scheduling, the main goal of the resource providers is to minimize the makespan whereas the goal of the user is to minimize the cost for grid application. Fitness function is calculated as:

$$\text{Fitness function} = \Theta\text{Cost} + \delta\text{Makespan} \quad (1)$$

where $0 \leq \Theta < 1$ and $0 \leq \delta < 1$ are weights to prioritize components of the fitness function. The cost of the job which executes on the resource and makespan is the completion time of the latest job and it is illustrated as:

$$\text{Cost} = \min\left(c(r_k)\right) \text{ for } 1 \leq k \leq n \quad (2)$$

$$\text{Makespan} = \max(F_j) \quad (3)$$

where F_j is the finishing time of the latest job.

Completion time of the job must be defined before calculating the makespan. Completion time indicates the time in which the resource can complete the execution of all the jobs. The completion time is the summation of Availability Time (AT) and the Expected Time to Complete (ETC) is calculated as:

$$\text{Completion time} = \text{AT}\ (r_k) + \text{ETC}\ (j_i, r_k)(4)$$

where AT (r_k) is the availability time of the resource and ETC (j_i,r_k) is the expected time to complete for job j_i in the resource r_k.

The current load of the resource is calculated by summing all the length of the jobs submitted to the particular resource with their Million Instructions Per Second (MIPS) rating and AT of the resource. The formula is illustrated as:

$$\text{Load}_{r_k} = \frac{\sum_{j=1}^{n} Length}{MIPS_{rk} \times AT_{rk}} \quad (5)$$

The resource utilization is the total amount of resources utilized by the user for executing the jobs in the system. The formula is given as:

$$\text{Resource utilization} = \text{Load}_{r_k} \times 100 \quad (6)$$

The success rate is defined as the percentage of jobs scheduled successfully from the amount of jobs submitted for scheduling. The formula is given as:

$$\text{Success rate} = \frac{\text{No.of Jobs Succeeded}}{\text{No.of Jobs Submitted}} \qquad (7)$$

4.4 Particle Swarm Optimization Based Heuristic (PSOH) Scheduling

Here we present our PSOH scheduling algorithm. Resource provisioning is done with the consideration of the multi-constraints strategy evaluation. The user wishes to minimize the cost whereas the resource provider wishes to minimize the makespan. Smaller values of makespan indicate that the scheduler is planning the jobs in an efficient manner. Another criterion is cost, which refers to the cost of the job execution on the resource along with the security assurance cost. PSOH is based on the biological behavior of the birds which foraging for food. This is done with the consideration of the various jobs that are gathered from the users. The PSOH algorithm is presented in algorithm 1.

Algorithm 1: PSOH based scheduling Algorithm

```
Given n - set of resources and m - set of jobs
Initialize the jobs randomly in the queue
For i=1 to n do
    For each job calculation fitness function for both global and local position
    Fitness function = cost + Makespan
    If (Fitness_gbest > Fitness_lbest)
        Fitness_gbest = Fitness_lbest
    Else
        Return Fitness_lbest
    Allocate the jobs to the resources with minimum Fitness and minimum load
End for
Apply heuristic
While there are unscheduled jobs in the queue do
    Get the jobs from the queue
    Repeat the process until all jobs are allocated
```

5 RESULT ANALYSIS

5.1 Parameter Evaluation

The assumed values for simulation are shown in Table 1.

Table 1 Assumed Values for Simulation

No. of Resources	No. of Machine per Resource	No. of PEs per Machine	MIPS per Machine	Bit Rate per machine (bps)
5	4	1-5	400-800	10^2-10^3

The resource utilization parameter is analyzed in percentage with the number of jobs like 10, 20 and 30. The utilization of the resource increases as the job increases because the resources are used efficiently. Figure 1 shows the resource utilization performance in the system.

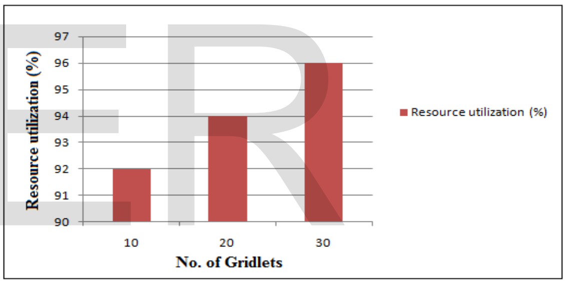

Figure 1 Performance based on Resource Utilization

The user satisfaction is analyzed with percentage based on the cost of the system. In which the user satisfaction slightly varies as the job increases because the cost of the system increases when the resource usage increases. Figure 2 shows the user satisfaction of the system.

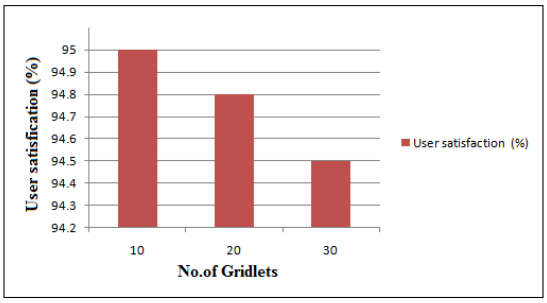

Figure 2 Performance based on Cost

The overall execution of the system is calculated in millisecond. The makespan increases as the job increase because the time taken for all the jobs to finish processing take more time. Figure 3 shows the makespan calculation of the system.

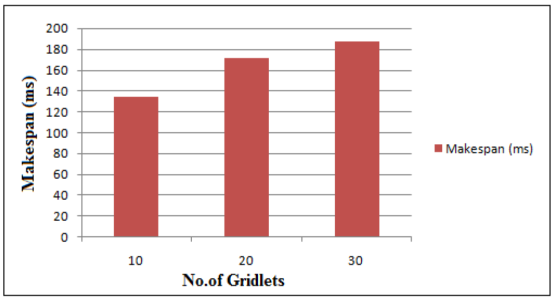

Figure 3 Performance based on Makespan

By analyzing these parameters, the system performance is increased by reducing the response time, overall cost of the system, and the makespan. It also maximizes the resource utilization rate and throughput with the user satisfaction. It yields the success rate equals to 1.

6 CONCLUSION AND FUTURE ENHANCEMENTS

The significant problem addressed in the system is resource management, security, cost of the computing systems, and makespan of the jobs. To overwhelm the problem, the system integrates load balancing and optimal scheduling with user authorization. It is determined by collecting the information of processing speeds, time, load and cost of the computing resources. The major problem is masquerade i.e., illegal user will act as authorized one. This will be overwhelmed by providing authentication mechanism. According to this information the scheduler submit their jobs to the resource that satisfies the expected cost and credentials provided by the user. The PSOH strategy of the scheduling system improves the resource utilization rate, minimizes the cost for scheduling the jobs, and the finishing time of the jobs. The proposed system not only minimizes the cost but also minimizes the makespan.

In future, the ability of a system to perform its function correctly even in the presence of failure of the resources. It reduces the resource availability which tends to make the resource unusable and result in the loss of user submitted jobs.

RFERENCES

[1] Abraham A, Izakian H (2009), 'A novel particle swarm optimization approach for grid job scheduling', Information Systems, Technology and Management, Vol.21, pp.100-109.

[2] Liu H, Abraham A (2006), 'Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm', In: Proceedings of 10th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, England, pp.500-507.

[3] Aron R, Chana I (2012), 'Formal QoS Policy Based Grid Resource Provisioning Framework', Journal on Grid Computing, Vol.10, No.4, pp.249–264.

[4] Aron R, Chana I, Abraham A (2015), 'A hyper-heuristic approach for resource provisioning-based scheduling in grid environment', Journal on Supercomputing, Vol.71, pp.1427-1450.

[5] Buyya R, Abramson D, Giddy J, Stockinger H (2002), 'Economic models for resource management and scheduling in grid computing', Concurr Comput, Vol.14, pp.1507–1542.

[6] Foster I, Kesselmen c (2004), 'The Grid: blueprint for a future computing infrastructure', Morgan Kaufmann Publishers, USA.

[7] Buyya R, Garg S K, Siegel HJ (2010), 'Time and cost trade-off management for scheduling parallel applications on utility grids', Future Gener Comput System, Vol.8, No.26, pp.1344–1355 (ISSN: 0167–739X, Elsevier Science, Amsterdam, The Netherlands).

[8] Garg S, Konugurthi P, Buyya R (2008), 'A linear programming driven genetic algorithm for metascheduling on utility Grids', In: Proceedings of the 16th International Conference on Advanced Computing and Communication (ADCOM 2008, IEEE Press, New York, USA). Chennai, India.

[9] He X, Sun X, Laszewski G (2003), 'A Qos Guided Min-Min Heuristic for Grid Task Scheduling', Journal on Computing Science and Technology, Vol.18, No.4, pp.442-451.

[10] Moshen A S, Deldari H, Doori B M (2008), 'Balancing Load in Computational Grid Applying Adaptive Intelligent Colonies of Ants', Journal on Informatica, Vol.32, pp. 327-335.

[11] Preetha Evangeline D (2012), 'Survey on Grid Security Models and Mechanisms', International Journal on Computational Sciences & Applications (IJCSA), Vol.2, No.6, pp.57-63.

[12] Ronald R, Adi S, Leonard M A (1978), 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems', ACM Trans. on Communications, Vol.21, pp.120-126.

[13] Visalakshi P, Sivanandam S N (2009), 'Dynamic Task Scheduling with Load Balancing using Hybrid particle Swarm Optimization', International Journal Open Problems on Computing and Mathematics, Vol.2, No.3, pp. 475-488.

[14] Faraz F M, Maen T A, Omidreza K (2013), 'A Hybrid Encryption Algorithm based on RSA Small-e and Efficient –RSA for Cloud Computing Environments', Journal of Advances in Computer Network, Vol.1, No.3, pp.238-241.