

OPTIMIZED RESOURCE PROVISIONING METHOD FOR COMPUTATIONAL GRID

P. Suresh and N. V. Keerthana

Department of Information Technology, Kongu Engineering College, Anna University,
Tamil Nadu, India – 638 052

ABSTRACT

Grid computing is an accumulation of heterogeneous, dynamic resources from multiple administrative areas which are geographically distributed that can be utilized to reach a mutual end. Development of resource provisioning-based scheduling in large-scale distributed environments like grid computing brings in new requirement challenges that are not being believed in traditional distributed computing environments. Computational grid is applying the resources of many systems in a network to a single problem 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 fulfill the user requirements considerably. The satisfaction of users while providing the resources might increase the beneficiary level of resource suppliers. Resource scheduling has to satisfy the multiple constraints specified by the user. The option of resource with the satisfaction of multiple constraints is the most tedious process. This trouble is solved by bringing out the particle swarm optimization based heuristic scheduling algorithm which attempts to select the most suitable resource from the set of available resources. The primary parameters that are taken in this work for selecting the most suitable resource are the makespan and cost. The experimental result shows that the proposed method yields optimal scheduling with the atonement of all user requirements.

KEYWORDS

Grid computing, Resource scheduling, HPSO

1.INTRODUCTION

Grid computing has come off as a computing platform to provide a vast amount of resource sharing to large-scale scientific applications. Computational grids can be set as an environment that organizes geographically distributed and heterogeneous resources in different administrative areas 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 induces a great effect on scheduling [6]. So, grid resource management has become one of the most important key concerns in the field of grid computing [7, 8]. A set of heterogeneous services is utilized in order to create the grid and, in addition, these savings are not done by a single soul but by different administrators in each of the systems. For the efficient procedure of a computational grid, several factors must be considered as load balancing, resource sharing and effective programming.

Grid resource management can be delineated as a process consisting identification of the prerequisites of the resources, matching resources to the 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 scheme is commanded to guide resource management decisions which include resource provisioning and scheduling by providing Quality of Service (QoS) metrics delivered to the users. These metrics introduce a number of challenging issues that need to be addressed such as resource provisioning-based resource scheduling and ensuring the scheduling of jobs on the trustworthy nodes [4]. Grid user often has a

limited control over grid resources and the resource handler is not always able to fulfill all the requirements due to the large number of resources as well as user's requests.

Grid scheduling is determined as the operation of making scheduling decisions involving allocation of tasks to resources over multiple administrative domains [4]. The end of this procedure is to maximize various optimization criteria such as machine usage, fairness and flow time or to guarantee non trivial QoS. Scheduling process should be flexible and fast so that it is capable to efficiently respond to dynamic changes in the grid environment (job arrival, failure, imprecise job runtime estimate, etc.). Effective scheduling in the grid environment is a complex problem which is not fully and efficiently solved in now a day's production system. Consequently, a sound approach is to search for suboptimal solutions, where sufficiently efficient algorithms exist. Nonetheless, production schedulers usually use very simple algorithms based on priority queues [8]. Referable to the limited and often imprecise information used by these algorithms their efficiency are frequently very depressed, especially when sufficient QoS is needed.

Heuristic methods often help to solve NP-complete problems [4]. Heuristic approaches can easily be used to grid scheduling problems because grid scheduling consists of various important issues such as heterogeneity of resources, the dynamic and independent nature of grid resources and finally the issue of different policies being followed by the resource providers and resource consumers for execution of their projects. The primary motif of this study is to propose a hyper-heuristic-based scheduling algorithm being able to be applied in grid environment and hence scheduling resources to the preferred jobs leading to the delivery of optimum results to the grid users.

Our contributions in this theme are: 1) Creation of Grid environment. 2) Workload and resource monitoring. 3) Implementation of multi-constraints strategy. 4) Heuristic Particle Swarm Optimization scheduling algorithm. 5) Result analysis.

2.RELATED WORK

Resource provisioning and scheduling are major things of computational grids and assist it achieve high performance in its execution environment. Referable to the heterogeneous and dynamic nature of grid resources, this basically has taken the strain of a large-scale optimization problem.

Hongbo Liu et al. [2] proposed a Fuzzy Particle Swarm Optimization (FPSO) for scheduling the job. This optimization algorithm dynamically generates an optimal schedule so has to complete the jobs within a minimal period of time as well as utilizing the resource in an effective fashion. The location and the velocity of jobs are represented in fuzzy matrices. The dynamic availability of resources and the cost of information transmission are not counted. Then it allocates more memory due to static allocation of resources.

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 more beneficial 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 protection.

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 price and minimum implementation time are addressed by this method. The performance of the grid is better by believing the median response time as a metric that is not taken in this fabric.

In grid computing, job scheduling is a complex undertaking. A good scheduling system must optimize QoS parameters such as price, utilization time and protection. The choice of services with minimal cost and protection is complex and finding the best pair of jobs and resources is an NP-complete problem [4]. To surmount these challenges, several methods for programming are introduced in the existing schemes. But they bear some problems like dynamic allocation of resources and efficiency of the organization, less implementation time, load balancing, minimum price and minimum makespan which are required to be seen in the projected scheme. Our proposed concept of hyper-heuristic-based resource scheduling algorithm minimizes the cost and makespan simultaneously. On the other hand, particle swarm-based heuristic has not been utilized for resource scheduling in the grid earlier and we are providing a new way while designing our resource scheduling algorithm.

3.PROPOSED WORK

3.1.Problem Definition

Grid resources are heterogeneous and dynamic in nature and the concept of job scheduling is the tedious task. Thus, an efficient scheduling should be packed away. Scheduling system must optimize QoS parameters such as price, usage time, security and makespan. Choice for inspection and repairs with minimal cost and security is more complex in the gridiron. 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 specified as a mapping of tasks to resources as: $S = J \rightarrow R$.

3.2.Creation of Grid Environment

The main elements 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 compute. Thus, the number of gridlets (jobs) is created which is received from the user. It holds all the information about jobs like length of the job, file size and output size. Then resource broker discovers the resources for scheduling strategies and job processing. Thus, 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 registered on to GIS, where GIS worked as an agent and collect all the relevant data about the resources.

In the proposed system, heterogeneous grid resources are considered for the allocation. The set of tasks would be garnered from the user along with their constraints. The multi-constraints gather from the user which in terms led to the successive execution of the set of jobs. Founded on these requirements job scheduling would be executed. 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 circle of resources. So the scheduling ensures a near-optimal satisfaction of QoS parameters. Consequently, the user now selects the most appropriate resources among all the resources shown that matches the user's budget and the other specifications

3.3. Multi-Constraints Strategy

In grid scheduling, the principal destination of the resource providers is to minimize the makespan whereas the destination of the user is to downplay the monetary value for power system application. Fitness function [4] 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 price of the job which executes on the resource and makespan is the completion time of the latest job and it is illustrated as:

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

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

where F_j is the finishing time of the latest job.

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

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

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

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

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

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

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

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

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

3.4. Heuristic Particle Swarm Optimization (HPSO) Scheduling

Here we present our HPSO scheduling algorithm. Resource provisioning is done with the thoughtfulness of the multi-constraints strategy evaluation. The user wishes to downplay 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 way. Another measure is cost, which brings up to the price of the job execution on the resource along with the security

assurance cost. HPSO is based along the biological behavior of the birds which foraging for food. This is executed with the thoughtfulness of the various tasks that are assembled from the users. The HPSO algorithm is presented in algorithm 1.

Algorithm 1: HPSO based scheduling Algorithm

Given n - set of resources and m - set of jobs
 Initialize the jobs randomly in the waiting line
For $i=1$ to n **do**
 For each job calculation fitness function for both planetary and local situations
 Fitness function = cost + Makespan
If (Fitness_{global} > Fitness_{local})
 Fitness_{global} = Fitness_{local}
Else
Return Fitness_{local}
 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

4.RESULT ANALYSIS

4.1.Environment Creation Setup

Table 1 Environment Creation Setup

Grid Resources	User Defined (1-5)
Machines for Each Resource	User Defined (1-5)
PEs and MIPS rating for each Resource	Random
Speed and Bit Rate for each Resource	Random
Price for Each Resource	Random
Gridlets	User Defined (10-40)
Length and File I/O size for each Gridlet	Random

Table 1 expresses the environment creation setup for power system system using GridSim. The resources and the machines are user-specified. The initial position of each resource is stored in the data server. Initially, grid user creates 10 gridlets. Job 1 arrives at job scheduler, and then the scheduler identifies the resource which is desirable for the peculiar task of computing the fitness function and load of the resource. Granting to the results, jobs are allocated to it. The operation will go on until all the chores in the queue are scheduled.

4.2.Parameter Evaluation

The accepted value of the simulation is presented in Table 2 and the number of grid lets is created for analyzing the parameter of the arrangement demonstrated in Table 3.

Table 2 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

Table 3 Performance Analysis based on Parameters

No. of Jobs	Resource Utilization (%)	User satisfaction (%) based on Cost	Makespan (ms)
10	92	95	133.876
20	94	93.5	171.364
30	96	94.5	187.56

The resource utilization parameter is analyzed in percentage with the number of jobs like 10, 20 and 30. The usage of the resource increases as the task increases because the resources are used more efficiently. The user satisfaction is analyzed with a percentage based on the price of the scheme. In which the user satisfaction slightly varies as the job increases because the price of the system increases when the resource usage increases. The overall performance of the system is calculated in millisecond. The makespan increases as the job increase because the time needed for all the tasks to finish processing take more time. By analyzing these parameters, the system performance is increased by reducing the response time, overall cost of the scheme, and the makespan. It also maximizes the resource use rate and throughput with the user satisfaction. It generates the success rate equals to 1. It implies that all the chores in the queue were scheduled completion.

5. CONCLUSION AND FUTURE ENHANCEMENTS

The significant problem addressed in the system is resource management, cost of the computing systems, and makespan of the tasks. To overwhelm the problem, the system integrates load balancing and optimal scheduling. It is settled by collecting the information of processing speeds, time, load and cost of the computing resources. Agreeing to this information the scheduler submits their jobs to the resource that fulfills the anticipated price of the user. The HPSO strategy of the scheduling system improves the resource usage rate, minimizes the cost of scheduling the jobs, and the finishing time of the jobs. The proposed system not only downplays 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 imaginations. It cuts down the resource availability which tends to produce the resource unusable and result in the loss of user submitted jobs. The active nature of the grid environment introduces challenging security. Users must be able to produce new services like

resources dynamically without administrator permission. These services should coordinate and interact with other services. Thus, the security should be provided by name the service with acceptable identity to the user i.e., user credentials.

REFERENCES

- [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] Abraham A, Liu H (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] Garg SK, Buyya R, 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] 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.

Author

Dr.P.Suresh is an Assistant Professor in Dept. of Information Technology in Kongu Engineering College, Tamil Nadu, India. His research interests lies in Grid Computing and C omputer Networking.



Ms. N.V.Keerthana is pursuing M.Tech - Information and Cyber Warfare (Dept. of Information Technology) in Kongu Engineering College, Erode, Tamil Nadu, India. She has completed her B.Tech in 2014 from Kongu Engineering College, Tamil Nadu, India.

