ISSN 2229-5518

Description of various Scheduling Techniques in Grid Computing Environment

Heena Setia, Abhilasha Jain

Abstract: Grid Computing is selection, sharing of wide range of heterogeneous resources for solving complex computations. Grid provides a parallel programming framework containing dynamic resources. Grid scheduling means allocation of jobs to resources present in infrastructure for faster execution. Grid scheduling algorithms have been a subject of powerful research over the last. However, evaluation and comparative analysis of these algorithms are often hard to perform. This is caused by many issues, like difficulties in acquire access to large scale IT infrastructure and lack of resource management facilities. Therefore, Grid scheduling algorithms have been often tested in simulation environments. This paper describes some existing scheduling algorithms for allocation of resources in grid computing environment.

Keywords: Grid Computing; Scheduling; Waiting Time; Completion time; Cost of computing; makespan; User deadline, Priority, Performance.

1. Introduction:

[1] Grid provides the facility to share, select & distribute work on resources present in the grid infrastructure. Applications with heavy use of resources, applications using sub applications and applications using special devices are candidates of grid. Main application areas of grid are medical, multinational enterprise, bio informatics, E-governance and E-learning. For perfect grid environment design features should include heterogeneity, autonomy, scalability and availability. The various standards of grid environment are OGSA (open grid service architecture), OGSI (open grid service interface), WSRF (web services resource framework) etc. Various tools are available for stimulating the behaviour of grid that are OptorSim (simulation of various HEP scenarios includes storage resources, computing resources and schedule for job allocation and the n/w), GRIDSIM (simulates different classes of dynamic resources, broker, users, gridlets, GIS) and SIMGRID (simulation of parallel applications in heterogeneous distributed environment).

Types of grid are computational grid(for applications containing complex computations), Data grid(for applications containing large-scale data) & network grid.

Scheduling problem in grid environment is viewed as NP-complete problem, where application is being assigned to resources to optimize overall performance and execution time. Scheduling algorithm adopting resource-centric scheduling objective (CPU utilization, throughput) and application-centric scheduling objective (performance, deadline) to achieve required level of QOS. There are two modes of scheduling: Batch & Online mode heuristic scheduling algorithms.

Scheduler architecture is described as:

Users pass their applications with necessary information (like length of jobs) to grid scheduler for allocation. Then scheduler allocates jobs to appropriate resources by considering the characteristics of resources which it obtains from GIS.

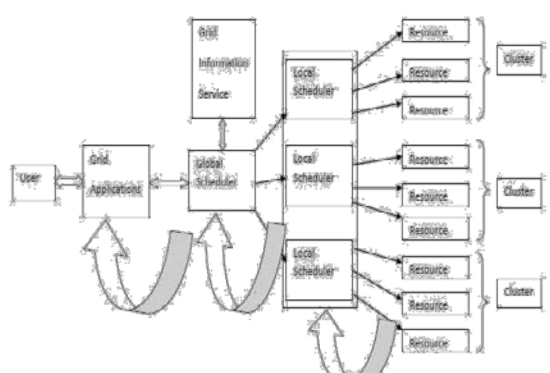

Figure 1.1: Architecture of scheduler

Performance evaluation metrics that should be considered are system utilization, throughput, turnaround time, job slowdown, economic profit, makespan, scheduling time and speed up.

This paper is organized as follows. Section 2 contains a description of various job scheduling techniques in Grid

---

- *Heena Setia is currently pursuing masters degree program in Computer Science & engineering in GZS campus college of engineering and technology, Bathinda(Punjab), India, PH-+91-7822991313. E-mail: heenasetia@rocketmail.com* environment. Section 4 will describe merits and demerits of scheduling techniques & Section 4 offers conclusions and implications for future work.

2. Description of various scheduling techniques:

Many task scheduling algorithms are developed for achieving required level of performance in Grid Computing. Some of the previous techniques in grid scheduling are given below:

2.1 ACO (Ant Colony Optimization) Algorithm for grid scheduling [2]:

Metrics consider: Processing speed, Load & Bandwidth of resources.

ACO is bio inspired algorithm. The main goal of this algorithm is to minimize computation time of each job. It is inspired by colony of ants that work together for finding the shortest path between nest and food place. Each ant when moving, deposit a chemical substance, called pheromone on ground. The path with high pheromone value should consider as shortest path. In ant colony algorithm, ant treated as job and submitted to grid system for scheduling. Resource broker collects information about resources from GIS. Characteristics of resources based on pheromone value (capacity of resources) that include processing speed, load and bandwidth of resource. Initially pheromone value for each resource based on execution time and transmission time. And matrix is made of all resources based on previously calculated pheromone values. The resource with highest pheromone value, allocated first. Local updates are done in matrix when job allocated to particular resource of machine and global updates are done when all jobs are allocated.

ACO algorithm has been used in grid computing because it is easily adapted to solve both static and dynamic optimization problems in job scheduling.

2.2 A Bee Colony Task Scheduling Algorithm in Computational Grids [3]:

Metrics consider: Maximum delay, Execution Time, Deadline.

Scheduling of tasks (gridlets) to appropriate resources in grid environment is NP-Complete problem. This paper proposed an algorithm which uses artificial bees for schedule submitted tasks to grid resources. Maximum delay and execution time of tasks are two main metrics in this algorithm. In BCO, artificial bees represent agents, work together and exchange information to reach the specific goal. Here, bee system having two essential components: food source and foragers. Foragers consist of two types that are unemployed and employed foragers (memorize location of food source). Unemployed foragers are further of two types: scout bees (having no knowledge of food source, searching spontaneously) and recruit bees (start searching by using knowledge of scout bees). The BCO algorithm based on population of bees in which population of bees seeks the best solution. The algorithm having two pass: forward and backward pass. When tasks submitted by user , they sent to resource named PBBS (priority base bee scheduling), it selects the task with highest priority and minimum delay. For each resource available in grid, calculate the data transmission and grid computation time and select that resource which execute the task with minimum completion time. Repeat the process until it receives all tasks. Then, resource randomly selects the algorithm (FCFS, LJF) for schedule tasks exists (forward pass). For each task, send all information about task to knowledge base, and according to objective function value of task, choose suitable algorithm which results in lowest finish time (backward pass). Execute task on resource based on nearest deadline. After execution updates in knowledge base are done.

Results show that this algorithm not only reduce the maximum delay and finish time of tasks but also satisfy the deadline and priority requirements of tasks.

2.3 Integration of genetic and tabu search algorithm for grid scheduling [4]:

Metrics consider: Makespan, Waiting Time, No. of iterations.

They consider two algorithms that are used to solve scheduling problems in grid environment. They proposed an algorithm that combines genetic algorithm and tabu search method (new method of meta-heuristic). Integration genetic-tabu search algorithm (IGTS) used tabulist formed in tabu search along with genetic algorithm to solve SJMM problems. Genetic algorithm based on natural selection (like genes in human being changes continuously according to environment through the process of breeding). The stages of genetic algorithm are: establishment, calculation of fitness value, selection, regeneration and creation of new population. TS does not use random solution, it uses memory structure that allows continuous search until better solution is not found. In short, GS uses all the possible solutions available, TS uses individual solution at first and then move. IGTS combines both and removes limitations of GA.

ISSN 2229-5518

Experimental results show that it reduces makespan and waiting time of jobs.

2.4 A non co-operative game theory approach to optimise workflow scheduling in grid computing [5]:

Metrics consider: Execution time & Processing Cost**.**

Task (gridlet) scheduling and resource allocation are very important for reliable performance and throughput in grid computing environment. The proposed workflow scheduling mechanism provides an optimal algorithm which minimize time and cost of computing. In this algorithm, scheduler/broker in grid architecture is treated as player. Broker can use any random algorithm for task allocation which gives maximum profit. In this broker use two functions: utility and payment function. Broker set lower limit to profit value for avoiding negative values. In proposed grid architecture, user submits their task to broker in grid environment and receive time and cost factor according to characteristics of Gridlet. The best scheduling solution is provided to user by broker. If user accepts solution, Gridlet executed and cost is paid to broker, else of user neglects solution, broker waits for next round. Broker mostly used genetic NSGA-II algorithm for allocation and results show that it is best strategy for scheduling a workflow.

2.5 An efficient scheduling algorithm for gird computing with periodical resource reallocation [6]:

Metrics consider: Reallocation, Processing time & Quality Constraint.

To meet time and quality constraint in grid computing environment, they provide a mechanism of task migration (at time of running) to utilize available computing power and to balance the load on resources. In this two models are developed (task model and computation model). In task model, task (Gridlet) treated as parallel application which can further divide into sub-tasks. These sub-tasks are divided into groups according to the data dependencies present in them. Division into groups is done as there is no dependency in tasks of one group. The o/p of one group of sub-tasks can be i/p to another group. Sub-tasks are execute in the form of stages starts from stage zero. The state and results of stages stores at local memory where it executes. In computation model, broker allocate group of sub-tasks to computing nodes and periodically pulls all information regarding resources from GIS and try to minimize execution time at each stage of execution. This algorithm named as periodically reinforcing and reclaiming algorithm (PR^2). Mapping (task with computing modes) is done by stage by stage , not as in fine grained parallelism, because size of sub-tasks are not small. The execution phase of sub-tasks goes through two phases:

Communication phase: For accessing i/p data.

Computation phase: For performing computation.

Sub-tasks are migrating to other computing nodes if it leads to smaller layer execution. Results show that the proposed algorithm smooth the workload in grid computing environment.

2.6 Adaptive scheduling in a mobile ad-hoc grid for time-sensitive computing [7]:

Metrics consider: Maximum workflow, Multi dimension cost function, User deadline.

This paper considers scheduling in mobile-grid environment in popularity of wireless technologies. The proposed scheduler provides real-time/dynamic allocation of tasks. It utilizes two optimization heuristics: workflow and cost function. It overcomes the challenges that are faced in P2P connectivity. The layout of mobile grid includes set of active local clusters that are subscribed to same grid environment authority. The architecture includes:

Local Grid (cluster) - Containing active devices that can communicate wireless.

Wigo - Wireless grid operator acts as a proxy within cluster that manage services of cluster.

Subscribers - Nodes in the grid can be service provider or receiver.

Tasks - Gridlets are time-sensitive and should be completed before deadline.

Schedules - Assign approximate resources to tasks. The mobile grid scheduler includes arrival queue, planned queue, failed queue, scheduling unit, routing unit and operates on three states that are super state, operations state, and update state.

The proposed algorithm provides capabilities that can maximize the possibilities of tasks being completed from their deadline. This accommodates three dimensions of this problem that are deadline, mobility and power dependency. It also uses cost functions in workflow that is the key advantage in scheduling tasks.

ISSN 2229-5518

2.7 Research on novel dynamic resource management and job scheduling in grid computing [8]:

Metrics consider: Scalability, Fault tolerant, Load factor.

This paper proposed a novel dynamic resource management and scheduling of jobs for gaining scalability, robustness and high performance. It is based on computational grid, so the main factor for assigning job here is computing power. It arranges all resources in two layered HST (heap-sort tree). Every node in system having two kinds of agents: ARA (autonomy representation agent) and NSMA (node state monitoring agent). ARA constructs them into top-HST according to the computing power available. The top node (root) has the highest CP (computing power). NSMA on each node randomly obtain computing power of node for which benchmarks are designed and constructs them into sub-HST.

When a job is requested for resource, it selects the resource with highest CP, if job assign to resource successfully, then computing power of that resource set to zero. If job is not assigned, then it reports error. After successful job submission, system is ready for new request.

Experimental results show that this novel agent based algorithm provide a good model for integrate agent technology with grid environment & improves performance efficiently.

2.8 A dynamic clustering heuristic for job scheduling on grid computing systems [9]:

Metrics consider: Communication cost, Priority, Mutex between jobs, Resource capacity**.**

In this paper, authors consider heterogeneous characteristics of jobs and proposed an efficient

algorithm for allocation in grid computing. It considers two heuristics: Cluster-threshold and Metrics for grid system. In cluster threshold, algorithm divides into two phases: cluster phase (eliminate the inter processor communication cost by finding pair of jobs which needs communication between them by assigning them on same processor) and modulate phase (migrations of jobs according to load on processor). Communication & processing cost, job priority matrix, job mutex matrix and job schedule matrix are the appropriate metrics in this algorithm. Dynamic allocation is main consideration in this paper in which alternate for existing jobs is considered and cost is replaced by alternate's cost. Heuristic consider five phases:

Phase 1 - Grouping (cluster) of jobs which are attached or need IPC.

Phase 2 - According to load, create heap of resources from light weight to heavy weight. Light weight resource should be on top and considered first.

Phase 3 - Deals with jobs which are not scheduled according to the previous metrics (deadline & priority).

Phase 4 - Migrate jobs according to load on resources (balance, light load, heavy load).

Phase 5 - Remove jobs which are scheduled or done, update metrics and add new jobs and scheduled them.

Experimental results show that dynamic heuristics scheduling is an optimal solution for scheduling which provide efficient performance with load balancing. This heuristic obtains optimal solution that is stable for heterogeneous grid environment.

2.9 Stochastic workflow scheduling with QOS guarantees in grid environment [10]:

Metrics consider: Dynamically adjust workload, Turnaround time, Cost of computation.

This paper provides global optimal scheduling solution for grid computing, having ability to dynamically adjust workload (according to resource characteristics) and gives a solution that provides required quality of service level for an application. Infrastructure also aims to minimize the cost of application execution. Stochastic workflow scheduling means formulation is based on queuing theory. It uses mechanism including various parameters which predict future workload on resource. So that it can assign application to resource directly without undergoing scheduling mechanism. It reduces the turnaround time for application.

This method is suited for enterprise and scientific theory where there are small sets of tasks. In this paper, application is actually set of dependent/independent tasks which are scheduled. Task is treated as individual service invocation, and these services are assigned to resources with predefined stages. Algorithm considers linear, hybrid and parallel tasks. It implements weighted random routing approach where proportion of jobs routed to proportion of service rate provided by resource. By different parameters like workflow arrival rate, service rate, etc., it predicts future workloads and then give appropriate solution for scheduling. Average cost per allocation is also considered for providing required QOS level. The results presented that this setup reduces the cost of computation and minimize scheduling time and achieve sufficient QOS level.

2.10 A new resource scheduling model with bandwidth aware job grouping strategy in grid computing [11]:

Metrics consider: Processing time, Resource Utilization.

This paper presents an NRSM algorithm for scheduling in grid environment. Maximizing utilization of resources and minimizing processing time are two main goals of this algorithm.

Grid system is a collection of clusters (group of resources) connected through high-speed n/w. The resources in cluster may be of any type and dynamic in nature (may enter or leave the grid, fails or stay idle). Here, selection of resources for job based on MHT , having two layers, grid level and cluster level. Grid level is called supervisor node having highest CP (computing power) and cluster level having all other nodes.

There are two main agents: Executor agent and supervisor agent. Executor agent obtain characteristics of all resources and select supervisor node, and also select a backup node which has 2nd highest CP and handle execution of jobs. Supervisor node manages co-ordination of clusters and grouped jobs based on computation and communication capacity of resource. It also receives the results from executor node. Grouping of jobs is based on strategy that processing requirements of grouped jobs should not exceed capacity of resource. When job arrives, it is allocated to top resource having highest CP and after allocation updates in status of resource is done. Results show that NRSM using NHT and grouping of jobs is able to minimize processing time of jobs and better utilization of resources.

2.11 Hybrid Adaptive Meta-Scheduling System For Grid Computing[12]:

Metrics consider: Makespan, Turnaround time, Job priority & Deadline, Resource Capacity.

For solving scientific and engineering problems, more computing power is needed. Grid is a system that allows sharing of heterogeneous resources for solving complex problems. Existing scheduling strategies consider either job side or resource side metrics. This paper proposed a meta scheduler in which both job and resource are taken into account and utilize the power of grid more efficiently. In this, jobs are prioritized based on location seem that consider both user and system priority and resource and prioritized based on computing power. The goal of this scheduler is to assign maximum number of jobs to minimum amount of resources, which is a very complex task. Resource discovery, resource selection, scheduling generation and job execution are the scheduling stages. In proposed system, when jobs arrive, system assigns priorities to jobs based on both user and resource characteristics, then sort jobs according to priorities. The job with highest priority is assigned to resource with highest computing power to achieve minimum turn-around time. Each and every time when job arrives, computing power of resources are calculated. Thus, it minimize the turnaround time and hence the performance can be increased by improving makespan.

2.12 FastMap: A Distributed Scheme For Mapping Large Scale Applications Onto Computational Grids [13]:

Metrics consider: Scale factor, Communication between tasks.

This paper proposed heuristic, called FastMap to map large applications on to dynamic resources present in computational grid. Previous existing systems use graph portioning of application, this paper attempt to solve problem from clustering perspective.

This approach focuses on scale factor in environment. It uses genetic algorithm for mapping. It stimulates FastMap as two level scheduler tree where first level act as top level scheduler and other is lower level schedulers. They test FastMap on large TIG, representative of a NASA application consisting of 50,000 nodes. The top level scheduler act as a root of the tree which consists of lower level schedulers as children (schedule resource belonging to different domains). All resources of grid make hierarchical structure, but actually all are interconnected. The parallel applications which are to be scheduled consider two aspects that are processing time and communication time. Distributed mapping scheme consists of three phases:-

(1) Task clustering

(2) Cluster mapping

(3) Recursive distribution

This is the responsibility of top level scheduler to equal the number of task clusters and number of resource cluster. In this paper, they developed a distributed mapping strategy for large parallel applications and uses greedy clustering

ISSN 2229-5518

approach to reduce communication between tasks which improves performance of FastMap.

2.13 A Fuzzy differential evaluation algorithm for job scheduling on computational grids [14]:

Metrics consider: Makespan, Job Length, Scaling Factor.

This paper proposes a novel fuzzy approach using DE (differential evaluation) for scheduling jobs on resources in computational grid. The main goal of this approach is to reduce execution time of jobs. In scheduling mapping of various jobs to specific time intervals on grid resources is done. In proposed approach, The length of job (number of cycles) and speed of each resource (cycle per unit time) are assumed to be known. Problem consists of mapping of jobs with given processing time on resource available. It provides a solution by generating fuzzy scheduling matrix (represents degree of membership) of particular resource to particular job. It, then calculate the makespan of individual, by determining allocation using highest membership.

Experimental results shows that this approach improves performance and provide optimal solution for scheduling in computational grid. It also evaluate the performance against GA, SA and differential evaluation and shows that it provides more optimal solution.

3. Merits and demerits of above mentioned techniques:

Technique with Ref no.	**Merits of scheduling technique**	**Demerits of scheduling technique**
Ant Colony[2]	Solve both static and dynamic optimization problems, reduce completion time of jobs, occupy memory of entire colony which helps in choosing best allocation	Difficult to understand theoretically, cost of processing is not considered, coding part is also difficult
Bee Colony[3]	User satisfaction(complete task before deadline), consider priority requirements, improves makespan	Large memory requirements, random decisions are made for allocation which are not independent, cost and performance are not considered
Genetic & Tabu Search Integration[4]	Tabu search with GA improves performance (avoiding local maxima), improve makespan & waiting time	Tabu search with GA makes algorithm slower, load & cost factor are not considered
Non Co-operative Game Theory [5]	Minimizes time and cost of computing, encourage scheduler to choose best algorithm for allocation	Load factor is not considered
Periodically Resource Reallocation[6]	Computing power is considered to balance load on resource, task migration is the main feature	Economic factor is not considered
Scheduling for mobile ad-hoc[7]	Real time scheduling which considered mobility & battery dependency issues, flow optimization & cost of computing is managed	Difficult to understand theoretically, time of computing is not considered
Novel dynamic Scheduling[8]	More scalable, robust & fault tolerant 2-layered HST system, balance load & improves makespan	Cost of computing is not considered, difficult to understand agent based model
Dynamic Cluster Heuristic[9]	Consider both job and resource characteristics for allocation	Difficult to clustering jobs according to communication cost, time consuming process
Stochastic workflow Scheduling[10]	Ability to dynamically adjust workload improves performance, more accurate & efficient allocation decision with specified QOS level	Analysis & coding part is difficult
New Resource Scheduling Model[11]	Maximizes resource utilization & minimizes computation time, job grouping strategy used which is easy to understand	Cost of computing is not considered, complex agent based architecture
Hybrid Meta Scheduling[12]	Consider both job & system priorities, improves performance and makespan	Cost of computing & load balancing are not considered

Fastmap[13]	Execution time and mapping(job to optimal resource) time are improved, mapping strategies are discussed for getting an optimal solution	Cost of computing load factor are not considered
Fuzzy Logic[14]	Improves makespan(maximum jobs completed in minimum time)	Cost of computing and load factor are not considered

Conclusion:

This paper shows description of various resource allocation techniques with their performance evaluation metrics. At the end, it presents merits and demerits of those techniques. Developers can use this paper for improvement in mentioned techniques or they can implement any of the existing according to their infrastructure requirements.

References:

[1] Raicu, I., Foster, I. T., & Zhao, Y. (2008, November). Many-task computing for grids and supercomputers. In *Many-Task Computing on Grids and Supercomputers, 2008. MTAGS 2008. Workshop on* (pp. 1-11). IEEE.

[2] Ku-Mahamud, K. R., & Nasir, H. J. A. (2010, May). Ant Colony Algorithm for job scheduling in grid computing. In *Mathematical/Analytical Modelling and Computer Simulation (AMS), 2010 Fourth Asia International Conference on*(pp. 40-45). IEEE.

[3] Mousavinasab, Z., Entezari-Maleki, R., & Movaghar, A. (2011). A bee colony task scheduling algorithm in computational grids. In *Digital Information Processing and Communications* (pp. 200-210). Springer Berlin Heidelberg.

[4] Darmawan, I., Priyana, Y., & Joseph, I. (2012, October). Grid computing process improvement through computing resource scheduling using genetic algorithm and Tabu Search integration. In *Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on*(pp. 330-334). IEEE.

[5] Yaghoobi, M., Fanian, A., Khajemohammadi, H., & Gulliver, T. A. (2013, August). A non-cooperative game theory approach to optimize workflow scheduling in grid computing. In *Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on* (pp. 108-113). IEEE.

[6] Lin, C. C., & Shih, C. W. (2008, July). An efficient scheduling algorithm for grid computing with periodical resource reallocation. In *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on* (pp. 295-300). IEEE.

[7] Morsy, H., & El-Rewini, H. (2013, May). Adaptive scheduling in a mobile ad-hoc grid for time-sensitive computing. In *Computer Systems and Applications (AICCSA), 2013 ACS International Conference on* (pp. 1-8). IEEE.

[8] Li, F., Qi, D., Zhang, L., Zhang, X., & Zhang, Z. (2006, June). Research on Novel Dynamic Resource Management and Job Scheduling in Grid Computing*. In *Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on* (Vol. 1, pp. 709-713). IEEE.

[9] Liu, L., Yang, Y., Shi, W., Lin, W., & Li, L. (2005, November). A dynamic clustering heuristic for jobs scheduling on grid computing systems. In*Semantics, Knowledge and Grid, 2005. SKG'05. First International Conference on* (pp. 4-4). IEEE.

[10] Afzal, A., Darlington, J., & McGough, A. S. (2006, October). Stochastic workflow scheduling with QoS guarantees in Grid computing environments. In *Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference* (pp. 185-194). IEEE.

[11] Sharma, R., Soni, V. K., & Mishra, M. K. (2010, July). A New Resource Scheduling Model with Bandwidth aware Job Grouping Strategy in Grid Computing. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on* (Vol. 4, pp. 324-328). IEEE.

[12] Nithyapriya, D., & Krishnamoorthy, N. (2013, February). Hybrid adaptive meta-scheduling system for grid computing. In *Information Communication and Embedded Systems (ICICES), 2013 International Conference on* (pp. 356-360). IEEE.

[13] Jain, A., Sanyal, S., Das, S. K., & Biswas, R. (2004, June). FastMap: a distributed scheme for mapping large scale applications onto computational grids. In *Challenges of Large Applications in Distributed Environments, 2004. CLADE 2004. Proceedings of the Second International Workshop on* (pp. 118-127). IEEE.

[14] Rao, C., & Babu, B. R. (2014). A Fuzzy Differential Evolution Algorithm for Job Scheduling on Computational Grids. *arXiv preprint arXiv:1407.6317*.