Grid Scheduling Algorithms: A Comparative Analysis

Seema Rani
Assitant Professor
N.M. Govt. PG College, Hansi

Ritika
Lecturer, N.M. Govt. PG College,
Hansi

ABSTRACT
Grid computing is one of the distributed computing where the resources of various computers are shared to solve a particular problem. Grid can be used for variety of purposes. Job scheduling is used to schedule the applications submitted by the user to appropriate resources in grid environment. It allows best match of resource for a particular job. In this paper various job scheduling algorithms are compared. The job scheduling algorithms are compared and contrasted.

General Terms
Grid Computing, Grid Scheduling, Distributed Computing.

Keywords
Makespan, Throughput, SETI, Crossover, Mutation, Minimum Completion Time

1. INTRODUCTION
Grid computing[1] enhances computing facilities over internet. Grid computing is a type of parallel and distributed systems that enable sharing. Grid computing has emerged from distributed computing where the resources of many computers in a network are used to solve a single problem at the same time. The difference between the conventional high performance computing such as cluster computing and grid is that grid tends to be loosely coupled, heterogeneous and geographically distributed. Though grid can be dedicated for a particular application, it can also be used for a variety of purpose. Search for Extraterrestrial Intelligence (SETI) is an example of Grid Computing Application. Since the user's machine is not able to process the jobs either because of resource or hardware constraints, the user can use the grid system for running the job. The user submits the set of jobs to the job scheduler and the job scheduler splits the job depending on certain factors and gives it to the machines having available resources on the grid. The machines will complete the task and final result will be given to the user. In order to achieve this job scheduling strategies has to be followed. Job scheduling and resource scheduling are the two main necessities in grid computing. In job scheduling, the job scheduler has to find the appropriate resource for the job that the user submits[2]. It has to find the best machine in grid to process the user job.

2. GRID SCHEDULER
Grid has two main schedulers such as local schedulers and grid schedulers. The **local schedulers** work in local computational environment and hence it is reliably, fast connection, works in uniform environment and also takes full control of the homogeneous resources.

The **grid scheduler[3]** is a software component in charge of computing and mapping of task to grid resources under multiple criteria and grid environment configurations. Grid Information service, Global Resource Allocation Manager, Monitoring and Discovery service, Local Resource Management service are the major components of grid scheduler. All these components must work coordinately in order to achieve better scheduling order.

Grid Schedulers are the top level schedulers also called as meta-schedulers. They are responsible for orchestrating resources that are managed by different local schedulers.

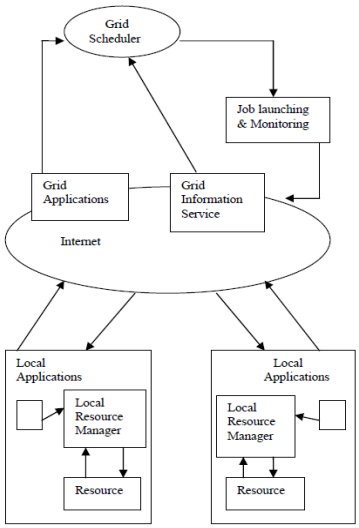

Figure : Grid Scheduling Architecture

Grid Scheduler (GS) are responsible for receiving applications from Grid users, select resources for the applications and finally generate application-to-resource mapping . Grid Information Service (GIS) collects and predicts the resource state information. An example of a GIS is Monitoring and Discovery system (MDS). A Local Resource Manager (LRM) is responsible for local scheduling inside a resource domain and reporting resource information to GIS.

3. GRID SYSTEM CLASSIFICATION
The grid systems are classified into three types. They are computational grid, data grid and service grid[4]. The **computational grid** have a higher aggregate computational capacity available for single applications than the capacity of any consistent machine in the system. This computational grid is a collection of distributed heterogeneous resources which can be used as an ensemble to execute large scale applications.
The **data grid** is a category of grids which deals with the controlled sharing and management of large amount of distributed data and also provide an infrastructure for synthesizing new information from data repositories such as digital libraries or data warehouses that are distributed in wide area networks.
The **service grid** is a category of grids which provides services that are not provided by a single machine.

4. JOB SCHEDULING IN GRID SYSTEM

Scheduling can also be classified into static and dynamic scheduling.

In **static scheduling[5]**, before execution the jobs are assigned to the suitable machines and those machines will continue executing those jobs without interruption.

In **dynamic scheduling**, the rescheduling of jobs is allowed. jobs executing can be migrated based on the dynamic information about the workload of the resources.

Grid system controls and coordinates the integrity of the grid by balancing the usage of reliable and unreliable resources among its participants providing better quality of service. Grid computing is a form of distributed computing where by resources of many computers in a network is used at the same time, to solve a single problem. The Job Scheduling: The aim of job scheduling system is to select the best machines in a grid for various users jobs. For each machine, job schedulers are generated by the management and scheduling system. Effective computation and Job Scheduling is becoming one of the main challenges in grid computing.

5. A COMPARATIVE ANALYSIS OF SCHEDULING ALGORITHMS

5.1 Genetic Algorithm

Genetic Algorithm[6] is an array of searching algorithms it searches a solution space of optimal solution to a problem. The main characteristic of the GA is how the search. Actually, the algorithm creates a *population* of possible solutions to the problem and lets them *evolve* over multiple generations to find better and better solutions.The better solution of generation is evaluated according to the fitness value and the candidates with better fitness values are used to create further solutions through crossover and mutation processes . Here we consider two parameters the total make span and throughput of the environment. The goal of this algorithm is to minimize the makespan of tasks and thereby increase the throughput of the environment. GA's methods and operators. Here we represent the detail of particularization of methods and operators of the GA for the scheduling problem based on the two schedule representations .

5.1.1 Generating the initial population.

In GAs, the initial population is usually generated randomly. Besides the random method, we have used two *ad hoc* heuristics, namely the *Longest Job to Fastest Resource - Shortest Job to Fastest Resource* (LJFR-SJFR) heuristic and *Minimum Completion Time* (MCT) heuristics.

These two methods are aimed at introducing more diversity to the initial population.

The LJFR-SJFR heuristic tries to optimize alternatively both values of makespan and flowtime.

5.1.2 In the MCT method

Each job is assigned to the machine in which the job obtains the minimum completion time (note that this is not the same as the minimum execution time. Jobs are considered for allocation at random.

Fitness, Crossover and Mutation operators are used for implementation of the Algorithm

The main disadvantages of genetic algorithms are the disruption of good sub-solutions by crossover and mutation operations.

And undesired population diversity loss by selection operations, which constantly decreases the variety of its specimens.

5.2 Task Graph Based Scheduling

The main objective of task graph concept is time optimization. Task graph is used for allocating the suitable resources to the grid environment[4]. Resource allocation and job scheduling have great importance in grid computing. Adequate information of available resources is based on these functions. Timely get resource status information is of great importance in ensuring overall performance of grid computing. In order to minimize the execution time in grid environment task graph concept is used. A processor completes the execution of the assigned task and the next task event is generated. At this event, the global scheduler assigns the outgoing communications of the finished task to channels. Each of the task having its own fixed execution time. If anyone gets completed before the fixed time, then the remaining time gets wasted. To avoid this, online scheduling was used.

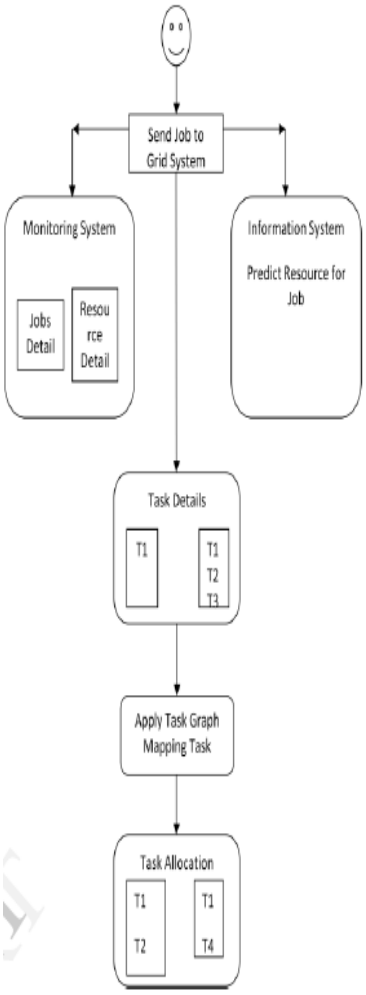

Fig. 2.Architecture of Task Graph

Figure 2 shows the Architecture of task graph. Initially the user sends a job to the grid system. The monitoring system monitor the job based on the job details and the resource details. The monitor service is monitoring the resource node. Then it manages the resource sensors and generates the resource monitoring data. At the same time the information systems predict the resource for job. The information node interacts with the users and runs the storage and query. The task details executes only in its specified time. If any task finished its execution before or after its fixed execution time it means that the scheduling is not efficient. After this applying

the task graph mapping to the grid system and allocate the tasks based upon resources free in the grid system.

5.3 Ant Colony Algorithm

The ant colony algorithm for job scheduling in grid aims at submitted jobs to resources based on the processing ability of jobs as well as the characteristics of the jobs[7]. Ant colony algorithm is the bio-inspired heuristic algorithm, which is derived from the social behavior of ants. Ants work together to find the shortest path between their nest and food source. When the ants move, each ant will deposit a chemical substance called pheromone. Using this pheromone, the shortest path is found. The same concept is used to assign jobs in grid computing. When a resource is assigning a job and completes, its pheromone value will be added each time. If a resource fails to finish a job, it will be punished by adding less pheromone value. The issue here is the stagnation, where there is a possibility of jobs being submitted to same resources having high pheromone value. In this ant colony algorithm the load balancing method is proposed to solve the issue of stagnation. The amount of information deposited on components may depend on the quality of the solution found. ACS has been applied to number of combinatorial optimization (CO) problems such as traveling salesman problem (TSP), Job-shop Scheduling Problem (JSP), Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP), etc.

The algorithm is as follows
(i) The user will send request to process a job
(ii) The grid resource broker will find a resource for the job
(iii) The resource broker will select the resource based on the largest value in the pheromone value matrix
(iv) The local pheromone update is done when a job is assigned to a resource.
(v)The global pheromone update is done when a resource completes a job
(vi) The execution result will be sent to the user

Advantage of Ant colony algorithm is that it is Less affected by initial poor solutions due to selection of random path.
It Can be used in dynamic applications such as Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP).

Disadvantage of Ant Colony Algorithm is that it performs much slower than the other scheduling algorithms. The reason behind that ACO computes several times for searching the optimal resource that is allocated to process the job. Hence, the calculation time of scheduling consumes more resources.
The time to convergence is uncertain but convergence is guaranteed.

5.4 Community – Aware Scheduling Algorithm (CASA)

CASA is a decentralized dynamic heuristic metascheduling algorithm. In CASA, jobs can be rescheduled. In order to overcome the stagnation a probabilistic approach has been used to assign jobs so that the jobs are evenly distributed to all other resources. CASA is a two phase algorithm[8]. The first phase is the job submission phase where each node receives the jobs that are submitted by local user. Consider a node A, it receives the job, it acts as a initiator node and requests all other nodes using the REQUEST message.The other nodes who are willing to take the job will reply through ACCEPT message. The node A will evaluate the other participating nodes using the historic data and selects the appropriate node and submits the job to it. The second phase is the dynamic rescheduling phase, the node which received the job will look for the job
which has large enough waiting time and has not been selected recently in the local job queue. That job will be rescheduled to the other nodes.
5 algorithms are discussed in CASA. They are:
(i) Job distribution
(ii) Job delegation request acceptance
(iii) Job assignment
(iv) Job rescheduling
(v) Job rescheduling request acceptance

5.5 Particle Swarm Optimization Algorithm

It is population based heuristic approach for optimization. It imitates the behavior of flying birds in the search space
Initially a swarm of particles is generated randomly.
Each particle has an updating position vector and updating velocity vector[9]. Changing velocity enables the particle to search around it individual best position and updating global best position .The flying particle in space adjusts its position according to its own experience and the experience of its neighbors.

Advantages of the particle swam optimization is that
the optimization method is not generation based. Each particle represents an independent solution . This allows an easy parallelization and hence more computations per time unit are possible

Disadvantage is that the method suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction.
Then the method cannot work out the problems of scattering and optimization and the method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

6. COMPARISON BETWEEN SCHEDULING ALGORITHMS

S. No.	Genetic Algorithm	Task Graph Based Algorithm	Ant Colony Algorithm	Community Aware Scheduling Algorithm	Particle Swarm Optimization
1.	It is based on the evolutionary process for large search space	The main objective is time optimization. In order to minimize the execution time in grid environmrnt task graph concept is used	It imitates the behavior of the ants	It is decentralized dynamic heuristic metascheduling Algorithm.	It imitates the behavior of flying birds in large search space

2.	The better solution of generation is evaluated according to the fitness value	The better solution is evaluated with successive monitoring of grid resources.	The better solution is evaluated on the basis of amount of pheromone deposited on the path.	The better solution for the Stagnation problem is probabilistic approach	The better solution is evaluated according to the position of particle in the space
3.	The candidates with better fitness values are used to create further solutions through crossover and mutation processes.	The Schedular attempts to raise the Throughput based on the availability of resources.	The scheduler attempts to find optimal distribution of wok unit to processing nodes.	CASA is a two phase algo. First phase is job submission phase. Second phase is dynamic rescheduling phase.	The flying particle in n dimension space adjusts its position according to its own experience and the experience of its neighbors.
4.	Each chromosomes represent the feasible solution, a fitness value represents the completion time of task	Task graph is used for allocating the suitable resources to the grid environment.	The ants represent the work units,food represents the computational power and the path represents the mapping.	CASA is inspired by the motivation of enable grid scheduling for the scope of the overall grid, instead of each single node.	Each dimension represents a task and each particle represents a feasible solution

7. CONCLUSION

Grid computing is a promising tendency to solve high demanding applications and all Kinds of grid problems. The main objective of grid environment is to achieve high performance computing by optimal usage of geographically distributed and heterogeneous resources. The success of grid computing depend on the effective utilization of the system for the various computing intensive job. Grid computing can solve complex task in shorter time and utilizes the hardware efficiently. To make the grid work efficiently , best job scheduling strategies have to employed job scheduling is the foremost step in grid computing where the user's Job are scheduled to different machines. The various scheduling algorithm and their main objective , better solution etc. have been studied and compared here.

8. REFERENCES

[1] I.Foster, C. Kesselman, and S.Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Application,15(3):200,2001

[2] R. Buyya and M. Murshed, "Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing." Concurrency and Computation : Practice and Experience, vol.14, pp 1175-1220,2002

[3] E.Huedo, R.S. Montero, and I.M. Liorente. The GridWay framework for adaptive scheduling and execution on grids scalable computing :practice and experience,6(3):1-8,2005

[4] S.Gokuldev,Lija Anna Mathew Task Graph Based Scheduling And Resource Monitoring In Grid Computing vol.2 Issue 5,May-2013

[5] M.Chtepen, "Dynamic scheduling in grids system," Sixth Firw PhD Symposium, Faculty of Engineering, Ghent University, 2005

[6] Javier carretero, Fatos Xhafa Genetic algorithm based schedulers for grid computing ICIC international volume 3,number 6

[7] Ku Ruhana Ku-Mahamud Husna Jamal Abdul Nasir , "Ant Colony Algorithm for Job Scheduling in grid Computing", in 2010 Fourth Asia International Conference on Mathematical/Analytical Modeling and Computer Simulation,IEEE computer Society 2010

[8] Y.Huang, A.Brocco, N. Bessis P. Kuonen, B. Hirsbrunner,"Exploring Decentralized Dynamic Scheduling For Grids and Clouds using the Community-aware Scheduling algorithm",in:2011 Future Generation Computing Systems Elsevier

[9] Kennedy J and R ,C Eberhart (1995) 'particle swarm OPTIMIZATION' proc IEEE int 'l Conf neural Network.