A Survey on the Allocation of Grid Resources Using Evolutionary Algorithms

T.Kokilavani, R.Raja Priya

Abstract— **Grid computing is a in grid environments. The optimal use of task scheduling increases the efficiency, throughput, and decrease the average turnaround time. This paper surveys, the evolutionary part of distributed computing which distributes the tasks to a network of computers linked together to offer preeminent computational resource. Task Scheduling is more important in the acquisition of high consumption of resources in the arena of grid computing and attaining consumption of resources task scheduling algorithm in grid computing that will benefit for the researchers to carry out the future work in that area and develop a better algorithm.**

Index Terms— **Grid, Architecture, Task Scheduling, Evolutionary Algorithms.**

I. INTRODUCTION

Grid computing connects numerous computing resources located at various places to achieve a specific goal. Grid can be viewed as a process like distributed computing with non-interactive tasks that deals with large amount of files. The performance of grid is distinguished from the existing computing resources as its nodes are set to perform different types of tasks or workloads[1]. The resources of the grid are seemed to be more heterogeneous and dispersed around the world. A single grid can be used to accomplish variety of tasks. The size of the grid is large, and, in general, it is constructed using grid middleware software applications.

Grid is one of the kinds in distributed computing formed by a network of loosely coupled systems to perform huge computing tasks.

Grid computing relies on the resources of other computers such as storage, processor, power supplies and network interfaces that are connected to the network by a conventional mode of internet. This would in turn contrast with the notion of traditional supercomputers as those computers have many processors connected through a high-speed local computer bus.

A.Grid Architecture

Grids started off in the mid-90s to address large-scale computation problems using a network resource-sharing machines that deliver the computation power only by supercomputers and large clusters at that time. The major motivation was that these high performance computing resources were cost effective, so the starting point was to use federated resources that could compute, network resources from multiple geographically distributed institutions, and such resources are generally heterogeneous and dynamic.

Grids focused on integrating existing resources with their operating systems, hardware, local resource management, and security infrastructure. In order to support the creation the so called “Virtual Organizations” a logical entity within which distributed resources can be discovered and shared as if they were from the same organization.

Grids define and provide a set of standard protocols, middleware, toolkits, and services built on top of these protocols. The generic architecture of grid environment interconnects the resources through internet that are available to offer services to the needy one.

Figure 1 depicts the general architecture of grid environment with which the resources are represented as super computer, data mart, and application or web services and so on.

Fig. 1 Generic Architecture of Grid Computing

II. TAXONOMY OF GRID SYSTEM

The grid system can be majorly classified into three types such as computational grid, service grid and data grid. These functionalities are designed to enhance the effective management of resources in grid environment [2].

Computational Grid contains systems that have a greater computational capacity that can be assigned for single applications than the capacity of any participatory machine in the network. These systems can be further subdivided into supercomputing or distributed and high throughput categories depending on the capacity utilized. To reduce the completion time of a task, grid executes parallel computation over multiple machines at the same time.

Service Grid offers systems that provide numerous services that are not possibly provided by any single system. Service grid offers various collaborative, on-demand, and multimedia Grid systems. Collaborative grid system helps in connecting users and applications in collaborative networks for enabling real time interaction between applications and

T.Kokilavani, Assisstant Professor, Department Of Computer Science, St.Joseph’s College(Autonomous), Tiruchirapalli-620 002, Tamilnadu, India.

R.Raja Priya, Research Scholar, Department Of Computer Science, St.Joseph’s College(Autonomous), Tiruchirapalli-620 002,Tamilnadu, India.

users in a virtual workspace. On-demand grid system dynamically allocates resources to provide services. On the other hand, multimedia grid system acts as a platform for providing real-time video streaming applications.

Data Grid allows systems to provide an infrastructure for delivering data services from data sources such as data warehouses or digital libraries that could be distributed in a large network. The difference between the data service provided by the computational grid and data grid is that the post one offers a specialized infrastructure
to applications for storage management and data access than the prior.

III. TASK SCHEDULING IN GRID COMPUTING

The resources in computational grid mainly offer computational or processing power to the nodes that it is shared with. The benefits of grid computing encompasses the reduction of computational cost, minimum task completion time and turnaround time that helps in increasing the productivity. One primary issue associated with the efficient utilization of heterogeneous resources in a grid environment is task scheduling. In general, the computational grid is responsible for monitoring the management of resources, access uniformity, task scheduling and security as the resources are globally connected. The potentials of computational grid can also be stretched in the manipulation of business computations, clustering variety of applications and accounting.

One of the vital roles of grid computing is to choose the most appropriate resource for tasks scheduling to complete its execution either in terms of reducing cost or time. Task scheduler is used to manage both tasks and resources; the task of the task scheduler is to select the appropriate resource for the task with which it can be assigned [3]. The main objective of task scheduling is to make use of the efficient utilization of resources for sequence execution of tasks. The proper allocation of tasks to the processors would in turn reduce the completion time the application. Fig 2 depicts the framework of the task scheduler in grid computing, where the user initially makes a request to the grid application (GA) for the allocation of suitable resource for its tasks, the grid application refers the grid information registry to know the status of the grid resource and chooses a suitable resource for the user request and assigns the tasks to the resource. Finally, when the task is done the GA returns the results to the requested user.

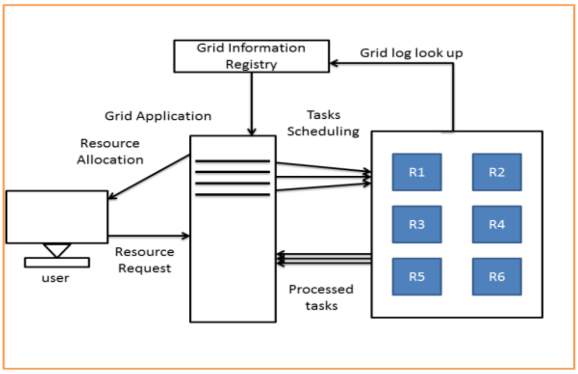

Fig 2.Framework of Grid Task Scheduler

A.Types of Task Scheduling

There are many types of scheduling algorithms available in grid environment to offer better results in terms of allocation of resources. Table 1, shows the different types of grid scheduling with their description[4]-[5].

Table 1. Types of Grid Scheduling

S. No	Types of Grid Scheduling	Description
1	Immediate Mode	Tasks are scheduled as soon as they are getting into the system
2	Batch Mode	Tasks are grouped into batches and allocated to the resources
3	Preemptive	Current execution of the tasks can be interrupted and the task can be migrated to another resource.
4	Non-Preemptive	Task should entirely be completed in the same resource
5	Static Scheduling	information of resources and tasks are assumed to be known in advance
6	Dynamic Scheduling	The tasks are entering dynamically and the scheduler has to work on the allocation of resources.
7	Independent Scheduling	Tasks are partitioned into almost independent parts which can be scheduled independently.
8	Centralized Scheduling	Contains single task scheduler on one instance , all information collected here
9	Hierarchical Scheduling	Tasks are managed at three levels to improve load balance
10	Decentralized Scheduling	No central instance is responsible, tasks are allocated using distributes schedulers

IV. EVOLUTIONARY GRID SCHEDULING ALGORITHMS

This paper reviews the most recent works carried out on the implementation of task scheduling through evolutionary grid scheduling in terms of the successful allocation of grid resources. Evolutionary algorithms provides well and approximate results to all sort of problems because they preferably do not make any assumption about the underlying facts, this generality is shown by successes almost in all domains. Evolutionary algorithms often interchange biological and computational principles and vice versa to produce optimized results to a problem. Fig. 3 shows the working processes of evolutionary algorithms.

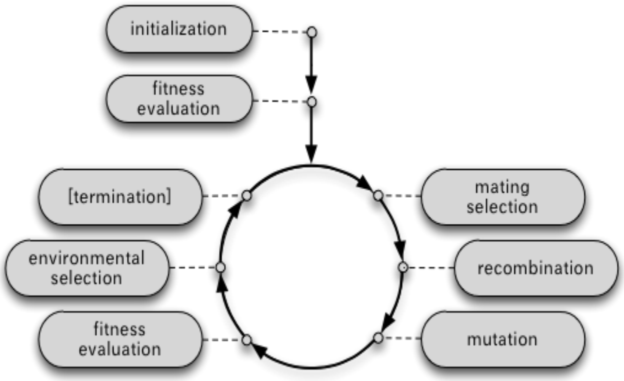

Fig 3. Methodology of Evolutionary Algorithms'

A. *Firefly Algorithm*

Yousif et al [6], presented a meta heuristics method based on firefly algorithm for scheduling tasks on grid computing. The authors dynamically created an optimal schedule to complete tasks within minimum makespan and flow time. In this work, the author treated each firefly as a candidate solution in a vector form, with n elements; they assumed all tasks were independent and preemption was not allowed and also assumed that the tasks and resources were ranked in ascending order based on the tasks' length and the processing speeds respectively. The speed of each resource is expressed in the form of MIPS (Million Instructions Per Second), and the length of each task in the number of instructions. They compared the results with Min-Min and Max-Min heuristic algorithm and showed that their method achieved less makespan and flowtime.

B. *Multiple Ant Colony Algorithms*

Umarani et al [7], proposed a new concept of multiple ant colony optimization scheduling algorithm for the allocation of best suitable resources to each task with minimal makespan and execution time. Multiple ant colony algorithms were employed to avoid local optima problem. The author considered both positive and negative feedbacks in searching solutions. The modified pheromone allocated the resource and obtained optimal shortest path.

C. *Genetic Algorithm*

Carretero et al [8], presented genetic algorithms based schedulers for efficiently allocating tasks to resources in a Grid system. They presented an extensive study on the usefulness of GAs for designing efficient Grid schedulers when makespan and flow time were minimized. Two encoding schemes had been considered and most of GA operators for each of them were implemented and empirically studied. Their experimental study showed that the GA-based schedulers outperform existing GA implementations in the literature for the problem and also revealed their efficiency when makespan and flowtime were minimized either in a hierarchical or a simultaneous optimization mode; previous approaches considered only the minimization of the makespan.

Moreover, the author identified GAs versions worked best under certain Grid characteristics, which was very useful for real Grids. They proved GA-based schedulers were very fast.

D. *Cuckoo Search Algorithm*

Rabiee et al [9], proposed a method called Cuckoo Optimization Algorithm (COA) to solve task scheduling in grids computational design and discussed the implementation and results of the proposed algorithms. The results were compared and analyzed with the existing evolutionary algorithms grid computing algorithms such as GA, ACO and PSO. The results showed was more efficient and better performing compared with Genetic and Particle Swarm Optimization.

E. *Particle Swarm Optimization Algorithm*

Izakian et al [10], introduced a Particle Swarm Optimization (PSO) algorithm, for grid task scheduling. Their proposed task scheduler aimed at minimizing makespan and flow time simultaneously. The goal of scheduler was to minimize the two parameters and compared with the existing fuzzy PSO method in order to evaluate its efficacy.

The experimental results showed that the presented method was more efficient and this method can be effectively used for grid scheduling.

Table 2. Summary on Reviewed Algorithms

Algorithm	Merits	Demerits
Scheduling Tasks On Grid Computing Using Firefly Algorithm[6]	➢ Less makespan time than min-min, max-min methods ➢ Creates dynamic and optimized task scheduling	The proposed work was not experimented in real grid environment
Efficient Multiple AntColony Algorithm for Task Scheduling In Grid Environment[7]	➢ Allocates best suitable resources to each task ➢ Minimize makespan and execution time	Suffers from local optima
Genetic Algorithm Based Schedulers For Grid Computing Systems[8]	➢ Identified the best GA version with respect to the Grid Characteristics ➢ Very fast ➢ Allows dynamic schedulers in batch mode for a short time	Does not support dynamic scheduling
Task Scheduling in Grid Computing with Cuckoo Optimization Algorithm[9]	➢ Generated an optimum schedule for minimizing time	Experimented only with small scheduling problem
A Novel Particle Swarm Optimization Approach for Grid Task Scheduling[10]	➢ Outperforms fuzzy PSO in terms of makespan and minimum execution time	Limited quality of service constraints

V. CONCLUSION

Grid computing can be widely used in applications like business, economics, medical, research, science and other areas. This paper presents a review on the contributions of evolutionary algorithms to address the challenges of grid computing such as resource and task scheduling. Most of the reviewed algorithms focused only on two parameters called makespan and flowtime, which have the direct impact towards the efficiency. This paper highlights the role of evolutionary computing algorithms in improving the overall performance of grid computing by means of increasing the throughput, reduced completion time of grid resources and so on. Discovery of new grid scheduling algorithms should focus on the effective search on grid resource over the network within minimum makespan than the existing algorithm.

REFERENCES

[1] Seema Kumari, Gulshan Kumar, “Survey on Task Scheduling Algorithms in Grid Computing”, International Journal of Computer Applications (0975 – 8887), Volume 115– No. 15, April 2015.

[2] Klaus Krauter, Rajkumar Buyya, Muthucumaru Maheswaran, “A taxonomy and survey of grid resource management systems for distributed computing”, Software—Practice And Experience Softw. Pract. Exper. 2002; 32:135–164 (DOI: 10.1002/spe.432).

[3] Dipti Sharma, Mr. Pradeep Mittal, “Task Scheduling Algorithm for Computational Grid in Grid Computing Environment”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013.

[4] Kumar, Mr Rakesh, and Navjot Kaur. "Task Scheduling in Grid Computers." Ijarcsse (2006).

[5] Dr. D.I. George Amalarethinam, P. Muthulakshmi, “An Overview of the Scheduling Policies and Algorithms in Grid Computing”, International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 2, No. 2, April 2011.

[6] Adil Yousif, Abdul Hanan Abdullah, Sulaiman Mohd Nor, Adil Ali Abdelaziz, “Scheduling Tasks On Grid Computing Using Firefly Algorithm”, Journal of Theoretical and Applied Information Technology, 30th November 2011. Vol. 33 No.2.

[7] S.Umarani, L.M.Nithya, A.Shanmugam, “Efficient Multiple Ant Colony Algorithm for Task Scheduling In Grid Environment”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3388-3393.

[8] Javier Carretero, Fatos Xhafa, Ajith Abraham, “Genetic Algorithm Based Schedulers For Grid Computing Systems”, International Journal of Innovative Computing, Volume 3, Number 6, December 2007.

[9] Maryam Rabiee, Hedieh Sajedi, “Task Scheduling in Grid Computing with Cuckoo Optimization Algorithm”, International Journal of Computer Applications (0975 – 8887) Volume 62– No.16, January 2013.

[10] Hesam Izakian, Behrouz Tork Ladani, Kamran Zamanifar, Ajith Abraham, “A Novel Particle Swarm Optimization Approach for Grid Task Scheduling”, Information Systems, Technology and Management. Springer Berlin Heidelberg, 2009. 100-109.