Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Novel Scheduling Approach for Load Balancing in Grid System

Mohit Agarwal[1],
[1,]Department of CSE,
IFTM University, Moradabad, Uttar Pradesh, INDIA

Mukul Pathak[2]
[2]Department of CSE,
KITM, Kurukshetra, Haryana, INDIA

Abstract— Grid computing is an enhanced version of distributed computing that consists of the coordination and involvement of geographically disparate heterogeneous resources. It contains large amount of computational tasks that require reliable resource sharing among multiple computing nodes. So, workload scheduling among nodes and resource management are challenging issues in the concept of Grid computing. In this paper, we have proposed a novel scheduling approach that will be used in the Load balancing technique to balance the workload in Grid system. In addition, a dynamic grid model, as a collection of clusters has been discussed. This novel scheduling approach compared with the other scheduling policy to show that our scheduling policy is capable of a higher resource utilization and reduction of the number of waiting jobs through the time.

Keywords: Grid computing, Scheduling, Load balancing, Clusters, Resource utilization, Dynamic grid model

I. INTRODUCTION

The concept of Grid was firstly introduced by Foster and Kesselman [1], who defined it as a distributed and decentralized computer environment composed of large number of heterogeneous resources which are managed by different owners like companies, universities or another business or scientific organizations. The purpose of the Grid technology is to manage large and heterogeneous computer environment that will allow an easy access to the Grid resources for various users. So, sharing in the Grid is not just a simple sharing of files but of hardware, software, data, and other resources [8].Grid computing can have variety of heterogeneous resources connected with each other. The constantly changing characteristic of the heterogeneous resources is known as dynamic resources. It allows them to submit their jobs into the system, guarantee them nontrivial Quality of Service (QoS) while hiding the complexity of the system itself by providing powerful but simple interfaces for the end user of the Grid [1]. On the other hand, scheduling in such a highly dynamic, distributed, heterogeneous and decentralized environment is an extremely difficult task if good performance, nontrivial QoS, scalability, etc required. Current scheduling techniques applied in Grids are based on the queuing systems of various types which are designed with respect to specific needs of Grid technology.

Today in grid system, Queue-based Approaches are used but this type of approach is time consuming as well as not efficient. FCFS (First Come First Serve) and EDF (Earliest Deadline First) are the examples of Queue-based Approaches. Here, in this paper, we are using schedule-based approaches that are used to solve the considered job scheduling problem. This approach is less time consuming and very efficient as compared to Queue-based Approach. BEST GAP and RANDOM SEARCH are the examples of schedule-based approach.

II. PROBLEM STATEMENT

In this paper, we define the problem of job scheduling in Grids or similar heterogeneous environments. Grid is a distributed, decentralized, heterogeneous and a highly dynamic computer environment consisting of various resources which are interconnected by computer network [2]. We now define some basic terminology concerning the Grid scheduling approaches. First, we have to distinguish between centralized and decentralized scheduling. By centralized scheduling we usually understand scheduling approach based on a single Local Resource Management System (LRMS). In this approach cluster is managed by a single scheduler which has a full control of all jobs and resources. Decentralized scheduling applies some hierarchical model and managed by one or more so called Meta-schedulers or Brokers. These brokers often manage different underlying centralized LRMSs [3]. Moreover, several brokers may communicate together, establishing a global Grid consisting of several local Grids or resources [3]. We can also distinguish between static and dynamic scheduling. In the static case, we assume that all information about jobs, resources, Grid topology and other constraints are known before the start of the scheduling process and do not change with time. In the dynamic case, Jobs and resources are appearing and disappearing in time, errors often occur and uncertain and imprecise information about jobs, resources or topology is available [9]. Finally, we have to distinguish between local and global scheduling. Local scheduling is used to assign the process (es) of the job to the CPU(s) on a single machine [4], while global scheduling decides where to execute given job. The main objective of this paper is to propose a novel scheduling policy for an efficient Load Balancing Algorithm in Grid environment. The main difference between existing Load Balancing algorithm and proposed Load Balancing is in implementation of Scheduling policies: selected by Selection Policy. Other aspects of any wide area network scheduler are its Transfer and Location policies [7].

III. PROPOSED METHODOLOGY

Cluster Level consists of a collection of computing nodes. Cluster-Level manager (CM) can fully control the computing nodes within it, but cannot operate the computing nodes of other clusters directly. The computing nodes within the cluster are referred as friends. CM maintains the load information along with registration information of its computing nodes. In Cluster-Level, each friend runs a CM. CM role is to balance the intra-cluster workload. A designated friend with highest CPU speed in each cluster is treated as the cluster server or master. Cluster System Monitor (CS) determines the load index of computing nodes and provides this information to CM.

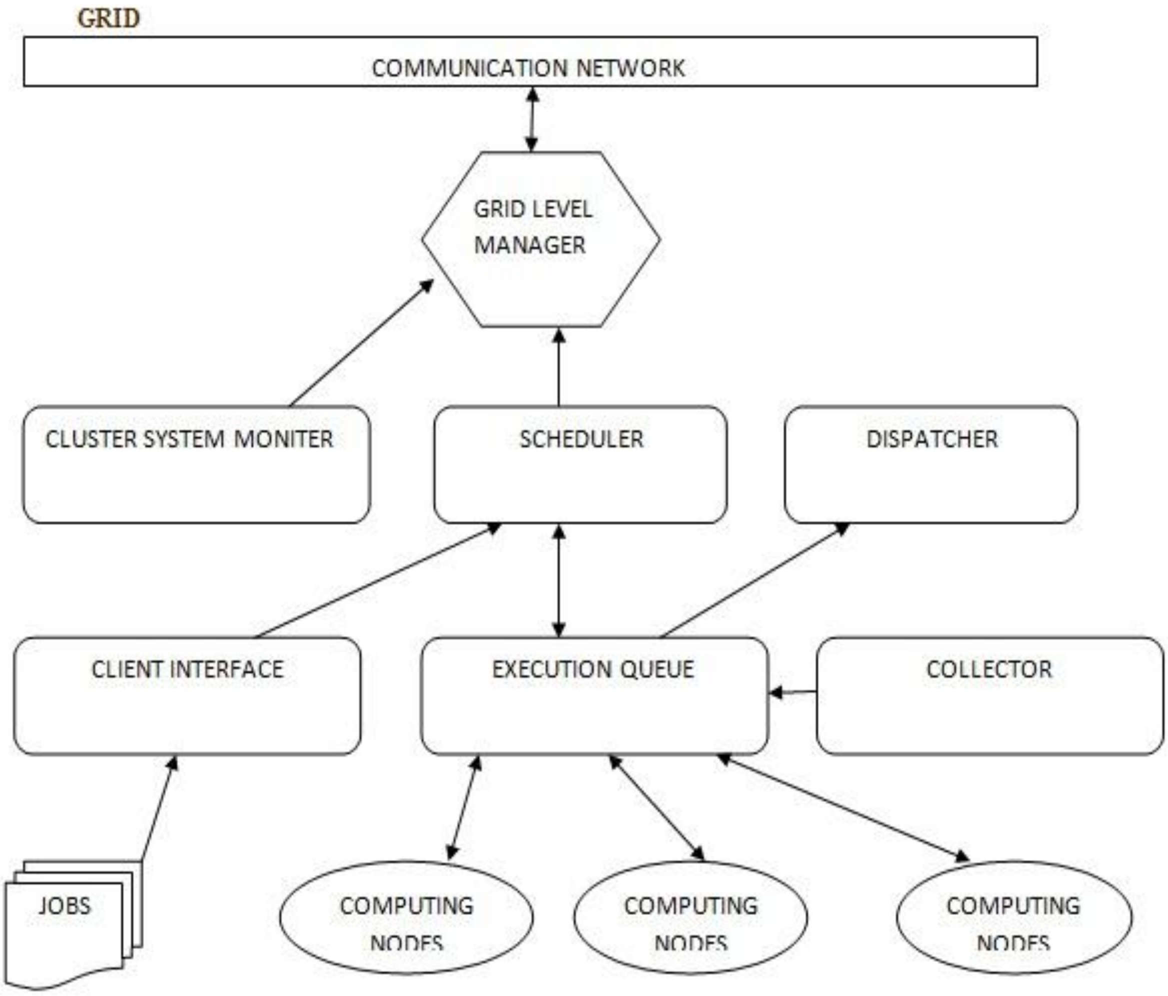

Figure1: Design of Load Balancing Model

Grid-Level consists of a collection of interconnected clusters. Grid-Level manager (GM) is responsible for load control among its clusters as shown in. GM maintains the load information along with registration information of neighbouring masters in the grid. Neighbours for each cluster are formed in terms of communication costs. GM calculates the minimum communication cost of sending or receiving jobs to/from remote clusters based on the information collected in the last exchange interval. Master of each cluster also runs the GM. K denotes the number of clusters in Grid-Level. Grid System Monitor (GS) determines the load index of masters and provides this information to GM. Client Interface provides a graphical user interface to the user for the submission of jobs. Scheduler is responsible for scheduling of submitted jobs. Dispatcher performs the dispatching of jobs to other masters. Collector is in charge of capturing jobs from other masters. Each neighbour of a cluster is responsible for completing the jobs assigned to them by their master. A decentralized job scheduling approach is used since the jobs generated by users are directly submitted to master. Scheduler runs as a sub-component of CM [6].

IV. IMPLEMENTATION DETAILS AND EXPERIMENTAL RESULTS

Through the experiment, we have compared three different scheduling algorithms: FCFS, EDF and Random Search + Best Gap scheduling policy. Figure 2, Figure 3 and Figure 4 presents graphs depicting the average machine usage per cluster (left) and the number of waiting and running jobs per day (right) as were generated by the Gridsim simulator during the experiment. These graphs nicely demonstrate major differences among the algorithms.
Following are some algorithm (BEST GAP+RANDOM SEARCH) which is used in my experiment.

Algorithm FCFS

```
stopping condition := false;
if queue is empty then
stopping condition := true;
end if
while stopping condition = false do
j := first job in queue;
if j in queue can be executed then
k := select cluster using FF policy;
remove j from queue and send it on k;
if queue is empty then
stopping condition := true;
end if
else
stopping condition := true;
end if
end while
```

Algorithm Earliest Deadline First

```
sort the queue according to dj in an increasing order;
execute FCFS;
```

Algorithm BestGap(job;criteria)

```
scheduleinitial := [schedule1;....; scheduleL]; schedulenew := ¢; schedulebest := ¢; t := 0; pointer := null;
for t := 1 to L do
if clustert is suitable to perform job then
schedulenew := scheduleinitial;
pointer := FindFirstGap(t; job);
schedulenew[t] := place job in the gap in the schedulenew[t] using the pointer;
if schedulebest = ¢ then
schedulebest := schedulenew; (schedulebest is correctly initialized)
else if AcceptCandidate(schedulebest ; schedulenew ; criteria) then
schedulebest := schedulenew;
end if
end if
end for
return schedulebest
```

Algorithm RandomSearch(iterations; time limit; stop event; criteria)

```
schedulebest := [schedule1;....; scheduleL]; schedulenew := schedulebest;
j := 0; run := true;
while (j < iterations and run) do
j := j + 1;
source := select random schedulenew[t] (1 < t < L) that contains at least one job;
if source = null then
return schedulebest; (schedule is empty so quit)
end if
job := select random job from source;
remove job from source;
if MoveJobRandomly(job; schedulebest ; schedulenew ; criteria) then
schedulebest := schedulenew; (move is accepted)
else
schedulenew := schedulebest; (move is rejected, job is returned back);
end if
if time limit exceeded or stop event detected then
run := false;
end if
end while
return schedulebest;
```

In this section we show the performance of the Gridsim simulator through several experiments. The GridSim toolkit provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers [5]. All experiments were performed on Intel Core I5 2.27GHz Laptop with 3GB of RAM. Unless otherwise indicated, the JVM (Java Virtual Machine) was limited by 1GB of available RAM.

The experiment involved 5000 jobs 14 clusters having 806 CPUs.

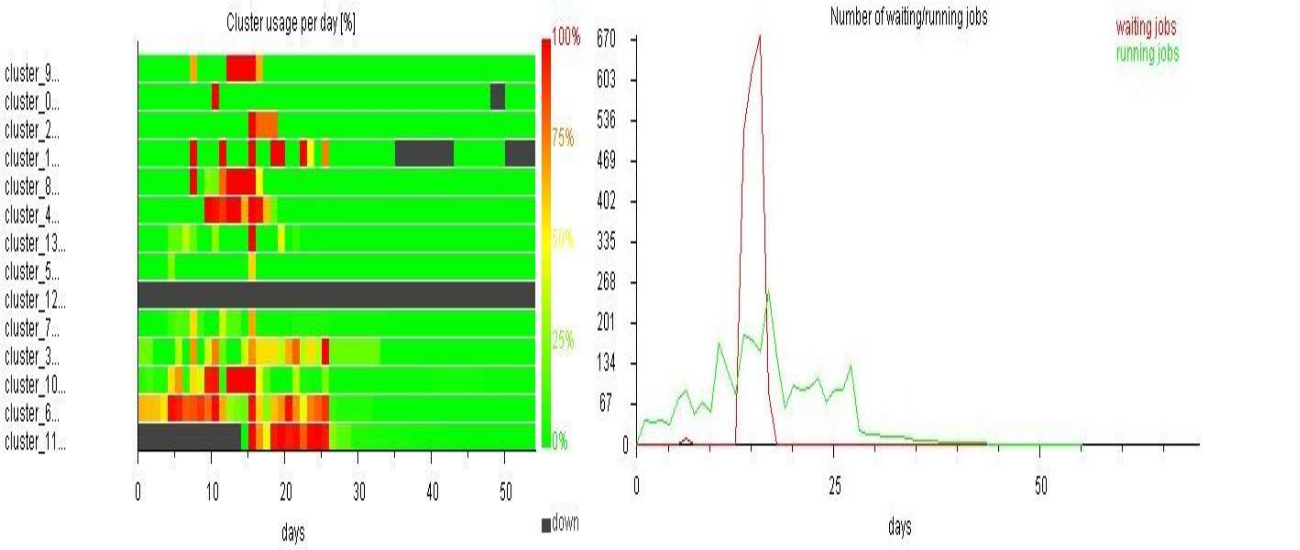

Figure2: FCFS Scheduling Result

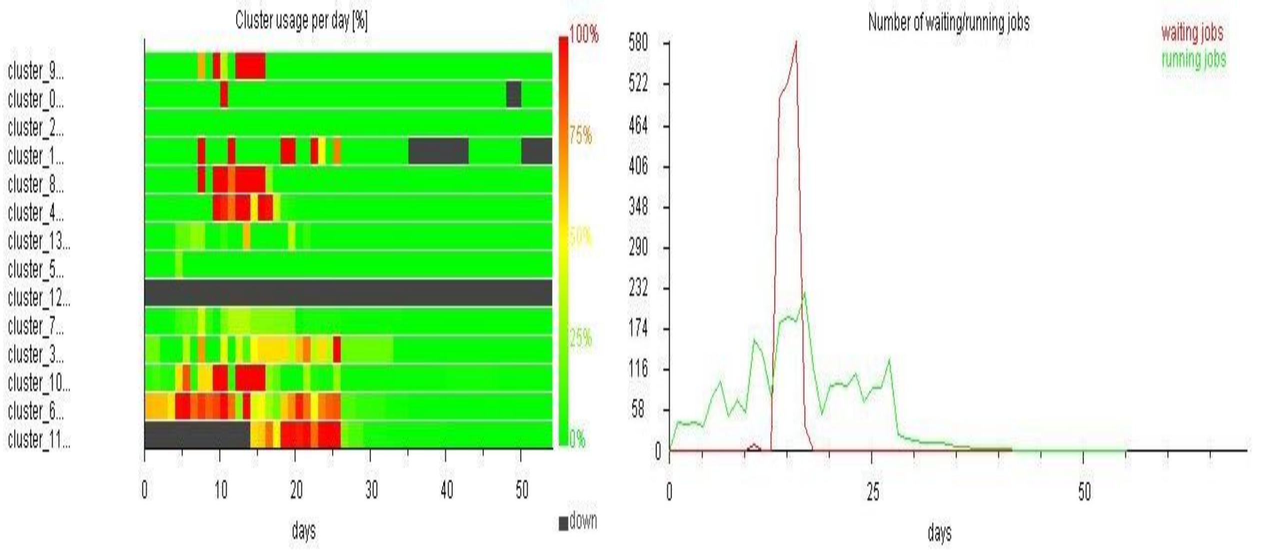

Figure3: EDF Scheduling Result

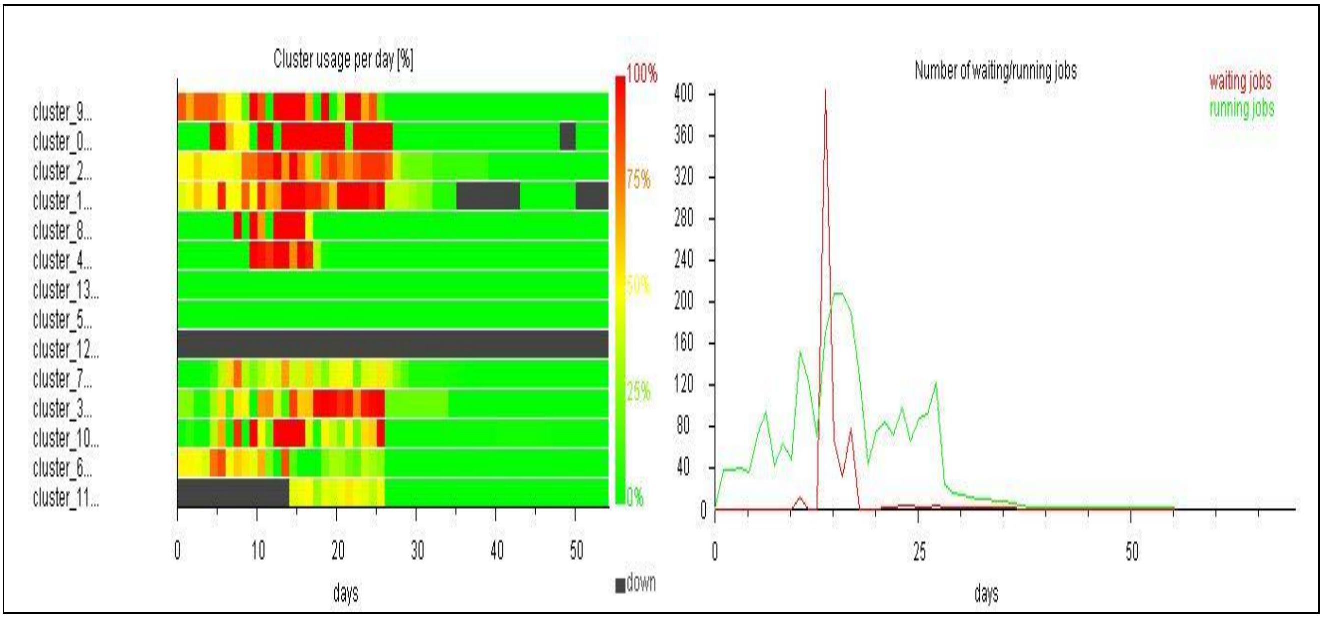

Figure4: BEST GAP + RANDOM SEARCH scheduling result

Concerning the machine usage as expected FCFS generates very poor results. FCFS is not able to utilize available resources when the first job in the queue requires some specific and currently unavailable machine(s). At this point, other more flexible" jobs in the queue can be executed increasing the machine utilization. This is the main goal of the EDF algorithm. As we can see, EDF is able to increase the machine usage by using the Earliest Deadline first approach. Still, EDF does not allow delaying the execution start of the first job in the queue, which restricts it from making more aggressive decisions that would increase the utilization even more. BestGap+RandomSearch algorithm does not apply such restrictions and thanks to the application of a more efficient schedule-based approach it produces the best results.

V. CONCLUSIONS AND FUTURE SCOPE

Grid Computing is definitely a promising tendency to solve high demanding applications and all kinds of problems. Objective of the grid environment is to achieve high performance computing by optimal usage of geographically distributed and heterogeneous resources. But grid application performance remains a challenge in dynamic grid environment. Resources can be submitted to Grid and can be withdrawn from Grid at any moment. This characteristic of Grid makes Load Balancing one of the critical features of Grid infrastructure. Therefore, there are a number of factors, which can affect the grid application performance like load balancing, heterogeneity of resources and resource sharing in the Grid environment. In this paper, we have focused on Load Balancing and tried to present the impacts of Load Balancing on grid application performance and finally proposed a efficient scheduling approach for Load Balancing algorithm in Grid environment. There may be certain future scope exist such as more complex models such as nesting of clusters need to be investigated. We may use another scheduling approach that could work better than the proposed scheduling approach and Experiments could be tried in a real environment.

ACKNOWLEDGMENT

It is a great pleasure for me to record my deep sense of gratitude and appreciation to Mukul Pathak., the paper supervisor for his valuable guidance, encouragement and healthy discussion during the paper of present research work.

.

REFERENCES

[1] Ian Foster and Carl Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgan,Kaufmann,1998.

[2] Ian Foster and Carl Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure, second edition. Morgan Kaufmann, 2004.

[3] Nicola Tonellotto, Philipp Wieder, and Ramin Yahyapour. A proposal for a generic Grid scheduling architecture. In Sergei Gorlatch and Marco Danelutto, editors, Proceedings of the Integrated Research in Grid Computing Workshop, pages 337-346. University di Pisa, 2005.

[4] Thomas L. Casavant and Jon G. Kuhl. A taxonomy of scheduling in generalpurpose distributed computing systems. IEEE Transactions on Software Engineering, 14(2):141{154, 1988.}

[5] Mr.V. P. Narkhede, Prof. S. T. Khandare , Fair Scheduling Algorithm with Dynamic Load Balancing Using In Grid Computing , Research Inventy: International Journal Of Engineering And Science Vol.2, Issue 10 (April 2013), Pp 53-57 ISSN(e): 2278-4721, ISSN(p):2319-6483, Www.Researchinventy.Com.

[6] Mukul Pathak,, Ajeet Kumar Bhartee, Vinay Tandon, An Efficient Scheduling Policy for Load Balancing Model for Computational Grid System, International Journal of Computer Engineering and Intelligent Systems, Vol 3, No.7, 2012

[7] N.Malarvizhi1 Dr. V. Rhymend Uthariaraj2, Hierarchical Load Balancing Scheme for Computational Intensive Jobs in Grid Computing Environment , 978-1-4244-4787-9/09/$25.00 ©2009 IEEE.

[8] Ian Foster , Carl Kesselman Steven Tuecke , The Anatomy of the Grid Enabling Scalable Virtual Organizations , Intl J. Supercomputer Applications, 2001.

[9] Jingyi Ma, A Novel Heuristic Genetic Load Balancing Algorithm in Grid Computing, Second International Conference on Intelligent Human-Machine Systems and Cybernetics, 2010