SCHEDULING OF DYNAMIC TASKS IN WIRELESS GRID COMPUTING USING OPTIMIZED FINAL STERNNESS PRIORITY RULE (OFSPR) ALGORITHM

P. Vijaya Karthik[1], V. Vasudevan[2]
Assistant Professor, Department of Information Technology,
Kalasalingam University, Krishnankoil-626 126, Tamil Nadu, India

ABSTRACT

Over the past several years, Grid computing is emerged as one of the most important and scalable alternatives to high performance supercomputing. Dynamic nature of grid computing is difficult to come up with near optimal solutions to effectively schedule the tasks in grids. This paper proposes a novel scheduling strategies and dynamically schedules the tasks by using Optimized final Sternness Priority Rule (OFSPR). When more than one job arrives at the same cycle time, Optimized Final Sternness Priority Rule (OFSPR) Algorithm is used at that moment. Simulation results presents that our approach is efficient for scheduling the tasks.

KEYWORDS – *Grid Computing, First Come First Serve (FCFS), Priority Schedule, Dynamic Scheduling, Optimized Final Sternness Priority Rule (OFSPR) Algorithm.*

I. INTRODUCTION

Grid Computing is a form of distributed computing, where loosely coupled and heterogeneous nodes donates their unused processor cycle to form a pool of process environment. In recent years grid computing is emerged as one of the most important alternatives to process computive-intensive tasks. Scheduling is important but challenging tasks for grid computing. The main advantage of grid computing is that inexpensive computing nodes are coupled together to produce resources at lower costs.

The principal challenge involved in grid computing environment is that optimal scheduling the tasks that dynamically enters the grid is a NP-hard problem. Performing effective scheduling in the grid is one of the key factors for achieving high performance in grid environments.

This paper proposes a task-scheduling algorithm which can operate effectively. Tasks scheduling strategy Optimized Final Sternness Priority Rule (OFSPR) Algorithm is used. The proposed algorithm is designed by combining backfilling procedure with optimized priority rule algorithm. The Optimized Final Sternness Priority Rule (OFSPR) scheduler is designed to manage newly arrived jobs by the grid users to the grid systems. The new jobs arriving are sorting by using First Come First Serve (FCFS) in the waiting queue. This waiting queue is then checked whether the first job in the queue can fit in the first hole of the machine. When more than one job exists at the same cycle, priority rule is applied for allocating the jobs to the machine. If there exist one gap that can be filled by a new job backfilling approach is used for this purpose. Simulation results show the effectiveness of our approach.

This paper is organized as follows. Section II describes the related work that is carried out in the past. Section III introduces the proposed work in this paper. Section IV describes the simulation results of our proposed work. Section V describes the conclusion of our approach.

Grid computing facilitates flexible, secure, coordinated large scale resource sharing among dynamic collections of individuals, institutions, and resource sharing in a geographical distributed area.

It is an evolving Technology of set of open standards for Web services and interfaces that make services, or computing resources, available over the Internet. These days the grid technologies are used on homogeneous clusters, and heterogeneous clusters and they can add value on those clusters by assisting, for example, with scheduling. The criteria for Grid Computing involves by coordinating the resources that are not subject to centralized control. It uses standard, open, general-purpose protocols and interfaces and delivers nontrivial qualities of service. The main applications of Wireless Grid computing in the field of Medicine, computationally-intensive scientific, mathematical, and academic problems like drug discovery, economic forecasting, seismic analysis, e-commerce.

The main components of Grid are

1) Grid Information Server
2) Global Grid Resource Broker
3) Local Grid resource Broker
4) Grid Users
5) Grid resources like computers, laptops, Servers, Printers.

The architecture of the Grid and how the scheduling takes place is mentioned in the proposed work section. The rest of the other algorithm were compared and it was graphically displayed in the Simulation Results section of this paper and we have proved this Optimized Final Sternness Priority Rule (OFSPR) is the best one for scheduling in the wireless grids.

II. ARCHITECTURE OF GRID

The role of Global Grid Resource Broker is the client Registration of jobs to process and the role of Resource nodes is to donate the resources at local Grid resource Broker and process the client request as per the instruction given by Local Grid Resource Broker. All the resource stastics like resource node, resource node size, resource header information will be collected from all the LGRB by Grid Information server and it is forwards to the GGRB . The main component in which scheduling will takes place in Global Grid Resource Broker. This GGRB provides all the information like resource type, resource variants, resource allocations and the corresponding nodes like nodes 1, node2, node3 and the information of the nodes will be acquired by GGRB. The Grid Scheduling takes place in the time sequence. To provide the efficient scheduling with the available resources is the one of the top issues in the Grid Computing environment. The Advantage of Scheduling includes Effective usage of all Grid resources, High throughput can be obtained, Decreased turnaround time, Users made responsible for providing input on schedule, consequences of effects of an increased workload. Each and every resources has its own policy and accountability.

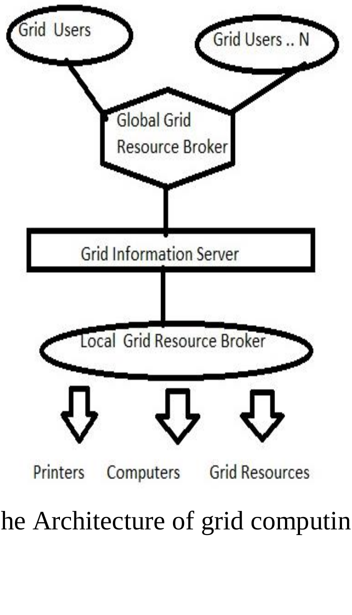

Figure 1: The Architecture of grid computing environment

III. RELATED WORK

Significant research has been made in the past to study the problem of optimal job assignment in distributed environment such as grids. Gap filling techniques plays an important role in grid computing environment for scheduling the tasks. This gap filling technique is derived from backfilling algorithm. The main purpose of backfilling is to improve the system utilization. Backfilling technique improves the resource utilization by filling the jobs into the gaps available in the queue. Jobs which are considered lower in the queue are moved towards the idle machines without affecting the execution of the jobs by moving them to the top of the queue. One of the Backfilling techniques is simple one because it moves the simple job to the top of the queue. Some research work has been carried out that combines the backfilling with priority algorithm.

Zafril Rizal M Azmi et.al. (2005), has done a work by combining backfilling with shortest job first algorithm. This algorithm rearranges the jobs in the queue based on the increasing order of the execution time of the jobs.

Dan et.al.,(2012) also combines backfilling with shortest job first algorithm but he uses different approach for carrying out his research. These techniques were computationally expensive because each time the scheduler has to reconstruct the queue when a new job arrives into the system.

Klusacek et.al.(2011), had find out the solution for the problem by providing an incremental technique for backfilling approach. This technique works by taking the last computational schedule as the starting point and contains information up to date. This technique avoids unwanted costs for constructing a schedule.

IV. PROPOSED SOLUTION

In this paper we have proposed a novel scheduling strategy for scheduling the tasks. The proposed were designed by combining the backfilling technique with the priority rule algorithm. IH-PR scheduler was designed in order to manage the newly arrived jobs that are submitted by the grid users to the grid systems. The newly arriving jobs are sorted by using the Optimized Final Sternness Priority Rule (OFSPR) Algorithm policy in the waiting queue. This queue checks whether the first job in the queue can fit into the first hole found in the machine. When more than one job arrives at the same cycle time, at that moment priority rule is applied to allocate the jobs to the selected machines. If there exists even one gap that can be filled by a new job, then simple backfilling approach is used for scheduling the jobs. Compared to that of the traditional method backfilling approach not only considers small job but also be applicable to all new jobs arriving into the systems. This mechanism is used to evaluate the makespan. The sternness is like the severity assigned to the failure of jobs in arrival time.

V. THE OPTIMIZED FINAL STERNNESS PRIORITY RULE (OFSPR) ALGORITHM

1. Get the total number of resources
2. Assign the total number of resources to a variable.
3. for i=0 to total number of resources do
4. if the number of processors requested by the job< number of processors actually available then
5. break;
6. else allocate the job to suitable machine
7. if suitable gap is found in the machine then
8. insert the dynamically available jobs into the machine based on the capacity of the gap size and also provide priority for filling the gap
9. end if
10. else if no suitable gap is found in the machine then allocate the jobs to another machine by applying priority rule algorithm
11. end if

VI. SIMULATION RESULTS

The simulation result for our approach is obtained by using GridSim toolkit. GridSim is one of the software platforms that allow the users to model and simulate the characteristics of grid resources and network with different configurations. It allocates the incoming jobs based on the space and time shared mode. It is also responsible for scheduling the computive or data intensive jobs.

It provides a well-defined interface for implementing different resource allocation algorithms. The resource allocation is done by using resource broker. The simulation result shows better results compared to that of the previous work carried. Minimizing Response Time: Response time is also known as flow time. Response time is the sum of final time of all tasks. Response time and makespan are two important objectives to be considered in scheduling. Minimization of makespan results in maximization of response time.

Maximizing Resource Usage: Maximizing resource usage in grid computing is another important performance factor to be considered. Usage is the percentage of resources actually occupied compared to that of the resources available for use. Low usage means the resource is idle and it is wasted.

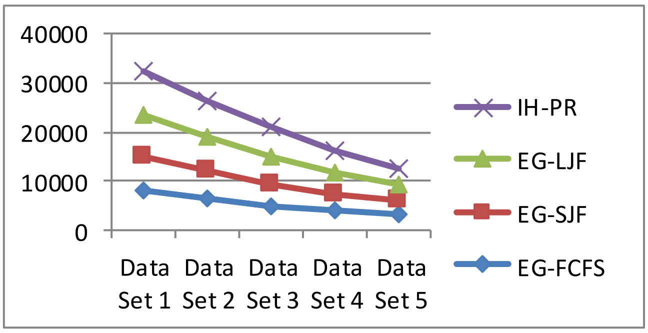

Figure 2: Graph showing comparison of flow time between different algorithms.

This graph in figure 2 provides a comparison between different algorithm used in the past and our proposed work carried out. This graph shows that our proposed one is superior compared to the previously used algorithms.

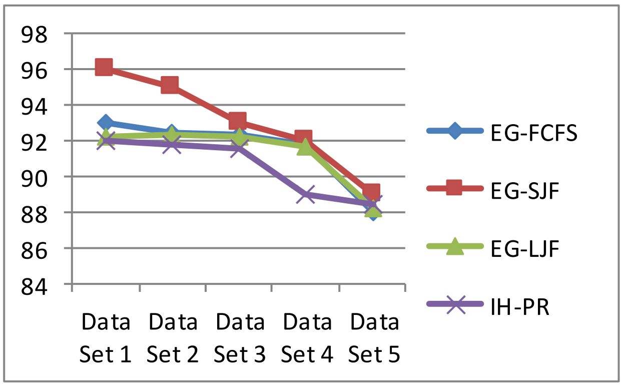

Figure 3: Graph showing comparison of machine usage by different algorithms.

The graph in figure 3, shows the usage of the machines by different algorithms. Also shows that our proposed method use less CPU time for computation than that of other algorithms provided in the past.

VII. Conclusion

Thus in this paper we have proposed a novel scheduling strategy which can schedules the tasks that arrives dynamically in the queue by using Optimized final sternness Priority Rule(OFSPR)policy. When more than one task arrives at the same cycle time, priority algorithm is used at that moment for scheduling the tasks in the suitable machine. Compared to previous approach this strategy is considered not only for smaller jobs but also for more number of newer jobs arriving into the system. Hence we can conclude based on the simulation results that our new proposed algorithm “Optimized final sternness Priority Rule (OFSPR) is superior when compared to the rest of other algorithms.

VIII. FUTURE WORK

In future, we can extend the work like when more than multiple task arrives at the different cycle time, Extension priority algorithm is used at that moment for scheduling the tasks in the suitable machine. Compared to previous approach this strategy is also can be implemented in the cloud meta scheduler which can also be implemented in the Cloud Environment.

REFERENCES

[1]. Amir Vahid Dastjerdi, Sayed Gholam Hassan Tabatabaei, and Rajkumar Buyya, A Dependency-aware Ontology-based Approach for Deploying Service Level Agreement Monitoring Services in Grid, Software: Practice and Experience, Volume 42, Number 4, Pages: 501-518, ISSN: 0038-0644, Wiley Press, New York, USA, April 2012.

[2]. Balachandar R. Amarnath, Thamarai Selvi Somasundaram, Mahendran Ellappan, Rajkumar Buyya, Ontology-based Grid Resource Management, Software: Practice and Experience (SPE), Volume 39, Number 17, Pages: 1419 - 1438, ISSN: 0038-0644, Wiley Press, New York, USA, Dec. 10, 2009.

[3]. Rodrigo N. Calheiros, Rajkumar Buyya, and Cesar A. F. De Rose, Building an automated and self-configurable emulation testbed for grid applications, Software: Practice and Experience (SPE), Volume 40, Number 5, Pages: 405-429, ISSN: 0038-0644, Wiley Press, New York, USA, April 25, 2010.

[4]. Mustafizur Rahman, Rajiv Ranjan, and Rajkumar Buyya, Cooperative and Decentralized Workflow Scheduling in Global Grids, Future Generation Computer Systems, Volume 26, Number 5, Pages: 753-768, ISSN: 0167-739X, Elsevier Science, Amsterdam, The Netherlands, May 2010.

[5]. SungJin Choi and Rajkumar Buyya, Group-based Adaptive Result Certification Mechanism in Desktop Grids, Future Generation Computer Systems, Volume 26, Number 5, Pages: 776-786, ISSN: 0167-739X, Elsevier Science, Amsterdam, The Netherlands, May 2010.

[6]. Marcos Dias de Assuncao, Alexandre di Costanzo and Rajkumar Buyya, A Cost-Benefit Analysis of Using Cloud Computing to Extend the Capacity of Clusters, Journal of Cluster Computing, Volume 13, Number 3, Pages: 335-347, ISSN: 1386-7857, Springer, Netherlands, September 2010.

[7]. Saurabh Kumar Garg, Rajkumar Buyya, and Howard Jay Siegel, Time and Cost Trade-off Management for Scheduling Parallel Applications on Utility Grids, Future Generation Computer Systems, Volume 26, Number 8, Pages: 1344-1355, ISSN: 0167-739X, Elsevier Science, Amsterdam, The Netherlands, October 2010.

[8]. Srinivasan, S., et al. Characterization of backfilling strategies for parallel job scheduling. in Parallel Processing Workshops, 2002. Proceedings. International Conference on. 2002.

[9]. Xhafa, F. and A. Abraham, Meta-heuristics for Grid Scheduling Problems, in Metaheuristics for Scheduling in Distributed Computing Environments, F. Xhafa and A. Abraham, Editors. 2008, Springer Berlin / Heidelberg. p. 1-3

[10]. Yu, J. and R. Buyya, A taxonomy of scientific workflow systems for grid computing. SIGMOD Rec., 2005. 34(3): p. 44-49.

[11]. Zafril Rizal M Azmi et al., Scheduling Grid Jobs Using Priority Rule Algorithms and Gap Filling Techniques,2011

[12]. Dalibor Klusacek “Dealing with Uncertainties in Grids through the Event-based Scheduling Approach”2012.

AUTHORS

P.Vijayakarthick received B.E degree in Information Technology and M.E in Computer science in 1998 and 2005 respectively. He is working as Assistant Professor in the Department of Information Technology, Kalasalingam University, India. He has published 5 Research papers in various International Journals. His current research interests include Distributed Computing, Grid Computing and Cloud Computing. He is an active member of ISTE.