Load Balancing & Task Scheduling Algorithms in Parallel Computing Environment: A Review

Raj Kumar[1], Prof. (Dr.) Vijay Dhir[2]

[1] Research Scholar, IKG PTU, Jalandhar, Punjab India,

[2] Professor CSE, M.K Group of Institutes, Amritsar, Punjab,

ABSTRACT

Parallel computing is a novel perspective in Grid computing environment. Parallel computing also plays a major role in cloud environment. For parallel computing to work efficiently, load balancing and task scheduling need to be managed accurately and precisely. In this review paper, we will study and analyze load balancing and task scheduling algorithms in parallel computing environment.

Keywords:- Cloud Computing, Grid Environment, Load Balancing Algorithms, Parallel Computing, Task Scheduling

I. INTRODUCTION

Parallel computing [1], [2] may be a variety of computation during which several calculations or the execution of different process area are carried out at the same time. Massive issues will typically be divided into smaller ones, which may then be resolved at an equivalent time. Before dividing the method, it is checked whether or not the method is divisible or not, if not, the method is dead and is not computed further and if is divisible then the processes are mapped among totally different processors.

Distributed computing [1], [2] is a field of applied science that studies distributed systems. A distributed system may be a model during which elements set on networked computers that communicate and coordinate their actions by passing messages among one another. The elements act with one another so as to attain a standard goal. It is a model during which elements of a software is shared among multiple computers to enhance potency and performance. Distributed computing simply implies that one thing is shared among multiple systems which can even be in numerous locations.

Advantages of parallel computing includes concurrency; save time; solve larger problems; load equalization and create a decent use of parallel hardware design. Disadvantages of Parallel computing includes Programming to focus on Parallel design may be a bit troublesome however with correct understanding and training you are ready to go; numerous code tweaking needs to be performed for various target architectures for improved performance; Communication of results may well be a tangle in certain cases; Power consumption is large by the multi core architectures; higher cooling technologies are needed just in case of clusters.

Therefore for parallel computing, Grid [1] and Cloud [2] environment, load balancing and task scheduling algorithms play a major role.

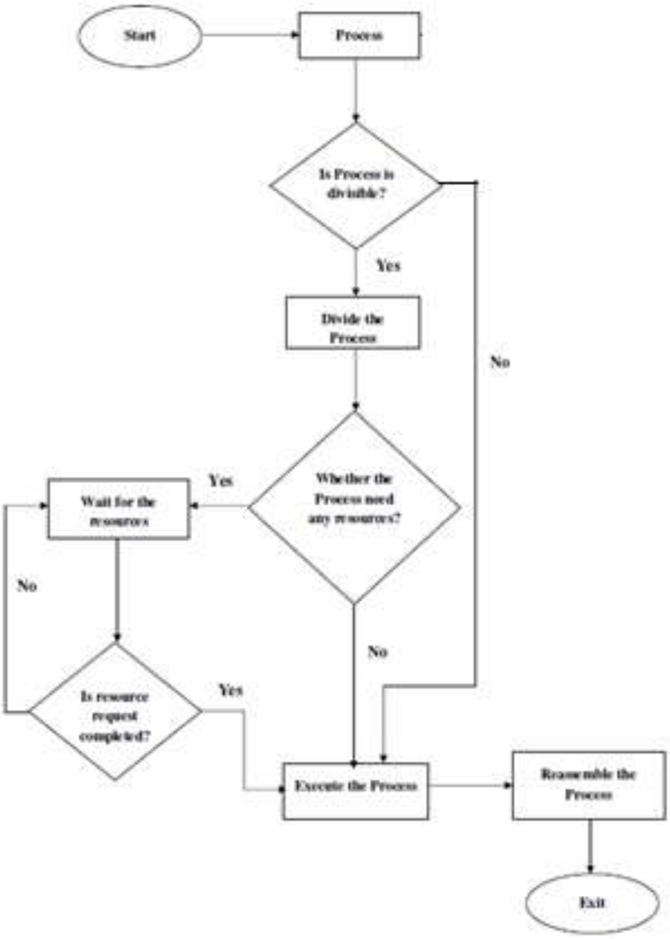

Fig.1: Phenomena of Parallel Processing

II. LOAD BALANCING

Computer utilization has increased the number of application tools that frequently uses the shared hardware and software package resources. (e.g. memory, processor, files etc.) This in turn increases the number of jobs to be submitted across web. Downside will be solved if we have tendency to distribute the applications across totally different computers, in such a way that it reduces the task latent period and also the overhead on one computer. Load balancing is correct distribution of applications over the web with the use of different kind of resources [4].

Load balancing algorithms [4] are categorized as static and dynamic scheduling. In static, no task is assigned to node after completion of program. The task to be assigned is provided to node before execution of program. Information concerned with process resources and task execution time is assumed to be well-known at compile time of scheduling process. Dynamic scheduling, throughout execution time relies on the re-distribution of processes among all the processors. Dynamic load balancing algorithmic [4] program assumes a no to previous information regarding job behavior.

1.1 Dynamic Load Balancing with Multiple Supporting Nodes (MSN)

Three approaches are put in good words during this algorithmic program, primary, centralized, and modified approach. In primary approach, ab initio methods are kept in queue or process may be assigned as they arrive. If these are in queue, processes are provided to primary nodes in one by one manner. From heavily loaded node Processes are migrated to light-weight node. Initial a light-weight node is checked within the same cluster, if appropriate node is not found then near cluster is searched and after that a needed node transfer takes place for completing the load transfer. On the other hand, one centralized node is given to cluster in centralized approach. Whenever a primary node is over loaded, initially, it searches for the light-weight primary nodes, if such primary node is found then load transfer occurs by balancing the load, Otherwise, one centralized node accommodates the overload of primary node. There is single node in centralized approach, so it processes the load at a very high speed via switching. Still, there are some drawbacks. A refined approach to remove all the drawbacks is there. Centralized node is splitted into smaller nodes known as supporting nodes (SNs). Suppose a process Pi is currently executed by SNi and a Primary node Ni is overloaded so that it finds a supporting node SNi suitable for transferring its overload [13].

1.2 Recent Neighbor Load Balancing Algorithm (RNLBA)

The problem associated with load balancing in grid is defined by assigning loads to a grid by taking care of the communication overhead while collecting load information. An economical dynamic load balancing algorithmic rule ‘Recent Neighbor’ (RN) has been conferred to tackle the existing challenges. RN performs inter cluster (grid) as well as intra-cluster load balancing [14].

In [14], Grid design is logically divided into three parts: Grid-Level, Cluster-Level and Leaf-Nodes. The clusters are totally interconnected within the grid. Multiple computing nodes referred to as leaf nodes are present in every cluster. The computing nodes within the cluster are heterogeneous. Grid cluster’s processing power is measured by taking the average speed of CPU across all computing nodes within the cluster. The tasks are supposed to be mutually independent, computationally intensive and may be executed at any cluster.

Load balancing is performed with two types of algorithms: Cluster-Level and Grid-Level. In Cluster-Level, current workload of the associated cluster is estimated by managing information from its own neighbors. Each and every Cluster-Level manager (CM) decides whether a load balancing operation has to start or not. If yes, then it tries to balance the workload among all its under loaded neighbors.

On the other-hand, if Cluster-Level fails to balance the load, then only, Grid-Level Algorithm works. The work tasks of all the overloaded clusters are provided to under loaded clusters in accordance with the communication cost and selection criteria. Under loaded clusters selected are the ones which need minimal communication cost in order to transfer tasks from the overloaded clusters.

1.3 A Load Balancing Policy for Heterogeneous Computational Grids (LBPHCG)

This policy tends to boost grid resource utilization and therefore maximizes outturn [15]. The prime focus of paper [15] is on the steady-state mode i.e. the number of jobs to be submitted is sufficiently high and besides this, the arrival rate of jobs doesn't exceed the grid overall process capability. The issues are self-addressed by the projected load balancing policy i.e. the computation-intensive and all the independent jobs has no communication between them. This paper [15] describes a two-level load balancing policy for the multi-cluster grid atmosphere i.e. local Grid Manager and site Manager Load balancing.

1.4 Grid Load Balancing using Intelligent Agents (IA)

The prime focus of paper [16] is on Grid load balancing with use of multi-agents and intelligent agent approaches. These approaches are accustomed to schedule local Grid resources and further doing a global Grid load balancing [16].

A Grid resource may be a digital computer or a workstation cluster. An agent at the Grid level provides Grid resources and services as a high performance computing power. Agents are the high-level abstraction of a Grid resource. Every agent consists of three main layers, from bottom to top: communication, coordination and local management layer. The local management layer acts as an agent for local grid load balancing. The coordination layer deals with the requests and organizes all the local information of the grid. The communication layer provides permit to act and deal with different agents. PACE performance prediction engine [16] which is a tool set for performance prediction is used by agents in Parallel and Distributed Systems.

1.5 Decentralized Genetic Algorithm (DGA)

Authors directed their analysis towards rushing up the convergence of genetic algorithms by using multiple agents and completely different universe to schedule sets of tasks [15]. The utilization of multiple ab initio search points within the downside space favors a high chance to converge towards a global optimum. Combined with the lookup services, this approach offers an answer to high reliability and scalability in demands [15].

SAGA Model is used by authors in [15] in order to submit scheduling request. The scheduler computes a near-optimal schedule which relies on the scheduling requests as well as the monitoring information gathered from Grid monitoring Service (MonALISA) [16]. The schedule as a ‘Request’ for task execution is then sent to the Execution Service. The solution provided by the scheduler is then received as a feedback by the user. It also provides information about the status of executed tasks. Moreover, the system will simply integrate new hosts within the scheduling method, and overcomes any failure situations by Discovery Service.

1.6 Prediction Based Technique (PBT)

Load balancing technique that may handle applications with heterogeneous cluster that cut back the typical response time is presented in this paper [12]. Three kinds of loads that are I/O, CPU, MEMORY are considered here. In this, Authors thought of a heterogeneous system with cluster computing platform within which a master node is liable for load balancing and for observing the available resources present in the node. Load manger consists of three modules: (1) predictor; (2) selector; (3) scheduler; Predictor is employed to predict the file I/O, CPU and memory needs of a task, for this author uses a statistical pattern-recognition technique.

The prediction is a weighted mean calculation of resource needs using the program's current state-transition model and therefore the actual resource usage in its most recent execution. Then foreseen value is fed to choose that is accustomed to select the best node among all nodes wherever the task can execute. Scheduler is accountable to dispatch the task to the node selected by the selector. Then task will send to that node and task will execute there. Load manager update the load status table. Preemptive migrations of tasks are not supported by this algorithmic program.

III. ANALYSIS OF VARIOUS LOAD BALANCING ALGORITHM

In MSN Algorithm, load is transferred locally as a result communication cost is reduced. However downside of this algorithmic program is that at ab initio phases, utilization of supporting nodes is decreases. So, resource wastage is more. RNLBGA takes less response time. It enhances the resource utilization as well as balances the load in an efficient manner.

Algorithm LBPHCG tried to reduce the job's mean response time and maximize the system utilization and outturn. Communication is required as long as a processing element joins or leaves its site. Intelligent Agent (IA) based approach has a benefit of the evolutionary algorithmic rule i.e. it is adaptive to changes within the system. It absorbs changes like addition or deletion of tasks or changes within the range of hosts. However, this algorithmic program cannot be utilized for an outsized scale, since complexity will increase exponentially with the number of hosts. Prediction based technique (PBT) in a cluster is used to attain the effective utilization of global disk resources. This may minimizes the typical cut down of all parallel jobs running on a cluster and scale back the typical response time of the roles. All these algorithms are summarized in TABLE 1 given below.

TABLE I. COMPARATIVE ANALYSIS OF EXISTING LOAD BALANCING ALGORITHMS IN GRID ENVIRONMENT

Approach	Strength	Drawback
MSN	Minimum traffic due to attached central node at each cluster	Initially process utilization at each supporting node is less
RNLBA	The parameters measured gives average response time, system load and communication delay	Need to make this algorithm for complex model nested of cluster
IA	It considers load index as a decision factor	It is not suited for dependent jobs

	for scheduling of jobs in a cluster and among clusters	
LBPHCG	It uses a combination of both intelligent and multi-agent processes	It can do further extension of the agent framework
DGA	It focuses on classes of independent tasks which avoids communication costs	Cannot handle data intensive programs
PBT	Better resource utilization and reduces the average response time	Not suitable for inter-dependent task

IV. SCHEDULING

Scheduling [19] is performed to increase the cloud performance. A task could embrace getting into data, processing, accessing software system, or storage functions. The information center classifies tasks in accordance with requested services and service-level agreement. By checking availability of servers, every task is assigned to them. The servers execute the requested tasks in turn and their result is provided to user after computation.

The jobs are submitted to cloud scheduler by associated users in task scheduling process. For getting the information of available resources, an inquiry is done by cloud scheduler. After inquiry, based on task requirement, different resources are allocated to tasks. Smart scheduling continuously assigns the virtual machines in an excellent manner.

An optimal scheduling algorithm [19] continuously improves the CPU utilization, turnaround time and cumulative outturn.

V. TECHNIQUES/ALGORITHMS TO DO TASK SCHEDULING

Many scheduling algorithms exist in distributed computing environment. Almost all these are applied in case of cloud environment also with some verification. Best system outturn and high performance computing are the utmost advantages. Cloud environment is not provided by traditional scheduling algorithms. In cloud environment, Job scheduling algorithms are often classified into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and on-line mode heuristic algorithms. In BMHA, Jobs are initially queued and picked up into a group when they arrive within the system. The scheduling algorithmic program can begins after a fixed amount of time. The main examples of BMHA primarily based formulas are; first come first Served scheduling algorithm (FCFS), round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm. By On-line mode heuristic scheduling algorithmic program, there is no need to make any group. Jobs are queued after they arrive within the system on regular basis.

5.1. First come first Serve Algorithm: Job within the queue that arrives first at the scheduler is served first. This algorithmic program is straightforward and fast.

5.2. Round Robin algorithm: Processes are sent in a FIFO manner however are given a restricted amount of cpu time referred to as a time-slice or a quantum. If a process doesn't complete its task before its CPU-time expires, the cpu is pre-empted and given to succeeding process waiting in a queue. The preempted process is then placed at the end of the queue.

5.3. Min–Min algorithm: This algorithm chooses tiny tasks to execute at first stage due to which large tasks have to wait long in a queue.

5.4. max – Min algorithm: This algorithm chooses massive tasks to be executed first, that in turn generate delays for tiny task which are not executed for a long time.

5.5. Most fit task scheduling algorithm: Task that fits best in prepared queue are executed first. This algorithm is not used widely because of high failure ratio.

5.6. Priority scheduling algorithm: Priority is assigned to each process and tasks are allowed to run. Tasks that have equal priority are arranged in FCPS order. The shortest-Job-First (SJF) algorithm could be a special case of general priority scheduling algorithm. Priority is the inverse of predicted cpu burst in SJF i.e. the shorter the cpu burst, the higher the priority and vice versa.

VI. ANALYSIS OF EXISTING SCHEDULING ALGORITHMS

The already mentioned task scheduling algorithms take into account completely different metrics for performing the scheduling operation. A good scheduling algorithm continually satisfies the needs of users. It provides good quality of service and simultaneously it should take into account the advantages at the cloud service.

It always try to cut back the cost and power consumption simultaneously providing better performance [41]. Load balancing and energy consumption are two main parameters that ought to be considered during a scheduling algorithm. It provides security to the users while providing services. Developing a far better algorithm by considering the combination of some important parameters together continually lead to a good scheduling algorithm which may be deployed in a cloud atmosphere for providing better cloud services to the users which may be considered as future enhancement [38]. An analysis on above scheduling ways and therefore the completely different scheduling parameters that are supported by them, their benefits and drawbacks are consolidated in the Table I; given below.

TABLE II. COMPARISON OF EXISTING SCHEDULING ALGORITHMS

Scheduling Method	Advantages	Disadvantages
First Come First Serve	Simple in implementation	Doesn’t consider any other criteria for Scheduling
Round Robin	Less complexity and load is balanced more fairly	Pre-emption is Required
Opportunistic Load Balancing	Better resource utilization	Poor makespan
Minimum Execution Time Algorithm	Selects the fastest machine for scheduling	Load Imbalanced
Minimum Completion Time Algorithm	Load balancing is considered	Optimization in selection of best resource is not Their.
Min-Min, Max-Min	Better makespan compared to other algorithms	Poor load balancing and QoS factors are not considered
Genetic Algorithm	Better performance and efficiency in terms of makespan	Complexity and long time Consumption
Simulated Annealing	Finds more poorer solutions in large solution space, better makespan	QoS factors and Heterogeneous Environments can be Considered
Switching Algorithm	Schedules as per load of the system, better makespan	Cost and time consumption in switching as per Load
K-percent Best	Selects the best machine for scheduling	Resource is selected based on the completion time Only
Sufferage Heuristic	Better makespan along with load balancing	Scheduling done is only based on a sufferage value
Benefit Driven, Power Best Fit, Load Balancing	Power consumption is reduced and cost is reduced even more number of servers used	Other QoS factors and completion time of tasks are less Considered
Energy efficient method using DVFS	Energy saving as per load in the system producing better makespan	Cost and Implementation complexity can be make better in future

DENS	Communication load is considered and job consolidation is done to save energy	Consider only data intensive Applications with less Computational Needs
e-STAB	Load balancing and energy efficiency is achieved based on traffic load, congestion and delay are avoided.	QoS factors can be considered For improvement in Overall Performance
Task Scheduling & Server Provisioning Improved Cost Based Algorithm	Energy is reduced meeting the deadline of tasks Resource cost and Computation performance is considered before Scheduling	Makespan and cost are less considered here Dynamic cloud Environment and other QoS attributes are not considered
Priority based Job Scheduling Algorithm	Priority is considered for scheduling. Designed based on multiple criteria decision making model	Makespan, consistency and complexity of the proposed method can be considered for Improvement
Job Scheduling based on Horizontal Load Balancing	Probabilistic assignment based on cost. Highest probable resource and task are selected for assignment.	Algorithm never mentions how the total completion time of the tasks will Remain
User Priority guided Min-Min	Prioritized is given to users improving load balancing and without increasing total completion time.	Rescheduling of tasks to perform load balancing will increase the complexity and Time
WLC based Scheduling	Dynamic task assignment strategy proposed, task heterogeneity is considered	Considering only load Balancing Feature
Cost Based Multi QoS Based DLT Scheduling	DLT based optimization model is designed for getting better Overall performance	Machine failure, Communication overheads and Dynamic workloads are not considered
Enhanced Max-Min Algorithm	Improves makespan and load balancing when large difference occurres in the length of longest task and other tasks or speed of processors	Parameters considered are limited and only theoretical analysis is Performed

VII. CONCLUSION AND FUTURE SCOPE

In this review papers we analyzed for different solutions and compared to each other by considering their pros and cons. The researchers can use these facts to develop better algorithms. In the above study it is found that some algorithms does not specified memory requirement of the jobs while submitting the jobs to the selected resources and some of algorithm does not considered communication cost, which we cannot neglect. Memory requirement of a job is vital in completing the execution of jobs at the selected resources within a time bound in realizing a real grid system.

Task scheduling is one of the most famous problems in cloud computing so; there is always a chance of modification of previously completed work in this particular field. In this paper various scheduling algorithms new namely A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing, A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment, An Greedy-Based Job Scheduling Algorithm in Cloud Computing etc.. have been studied and analysed.

REFERENCES

[1] Dhir, Vijay. "Alchemi .NET Framework in Grid Computing." Proceedings of the 3rd National Conference; INDIACom-2009 Computing For Nation Development at Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi. 2009.

[2] Dr. Vijay Dhir, Er. Gagandeep Kaur ,"Execution of cloud using freeware Technology" , International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), Vol 3, Issue 12, pp 22-29, December 2016.

[3] Haq, Abdul, and Munam Ali Shah. "Task Scheduling in Parallel Processing: Analysis." Journal of Biometrics & Biostatistics 6.5 (2015): 1.

[4] Sharma, Anuradha, and Seema Verma. "A SURVEY REPORT ON LOAD BALANCING ALGORITHM IN GRID COMPUTING ENVIRONMENT." Int. J. Adv. Engg. Res. Studies/IV/II/Jan.-March 128 (2015): 132.

[5] Tal Maoz, Amnon Barak, Lior Amar, "Combining Virtual Machine Migration with Process Migration for HPC on Multi-Clusters and Grids" in IEEE International Conference on Cluster Computing, The Hebrew University of Jerusalem, Israel, 2008 .

[6] Hans-Ulrich Heiss, Michael Schmitz, "Decentralized Dynamic Load Balancing: The Particles Approach", Department of Informatics, University of Karlsruhe, Germany, 1999.

[7] Ashok Singh Saira, Gautam Barua, "Distributed Route Control Schemes to Load Balance Incoming Traffic in Multihomed Stub Networks", IEEE ,Dept. of ComputerScience and Engineering Indian Institute of Technology Patna & Guwahati, 2010.

[8] Md. Abdur Razzaqu, Choong Seon Hong, "Dynamic Load Balancing in Distributed System: An Efficient Approach", Korea, 1991.

[9] M. Randles , Abu-Rahmeh , P. Johnson, A. Taleb-Bendiab, "Biased random walks on resource network graphs for load balancing", Springer,dec. 2009.

[10] K. Saruladha, G. Santhi, "Behavior of Agent Based Dynamic Load Balancing Algorithm for Heterogeneous P2P Systems", CSE Dept., Pondicherry Engineering College, Pondicherry, International Conference on Computational Intelligence and Multimedia Applications, 2007.

[11] Marina Petrova, Natalia Olano and Petri Mahonen, "Balls and Bins Distributed Load Balancing Algorithm for Channel Allocation" in IEEE, RWTH Aachen University Aachen, Germany, 2010.

[12] Parveen Jain, Daya Gupta, "An Algorithm for Dynamic Load Balancing in Distributed Systems with Multiple Supporting Nodes by Exploiting the Interrupt Service", ACEEE, ACADEMY PUBLISHER, Delhi College of Engineering, New Delhi, 2009.

[13] M. Kamarunisha, S.Ranichandra, T.K.P. Rajagopal, "Recitation of Load Balancing Algorithms In Grid Computing Environment Using Policies And Strategies - An Approach" International Journal of Scientific & Engineering Research, 2011.

[14] Jasma Balasangameshwara, Nedunchezhian Raju, "A Decentralized Recent Neighbour Load Balancing Algorithm for Computational Grid",The International Journal of ACM Jordan (ISSN 2078-7952), Vol. 1, No. 3, September, 2010.

[15] Sandeep Singh Waraich, "Classification of Dynamic Load Balancing Strategies in a Network of Workstations", Fifth International Conference on Information Technology: New Generations, 2008.

[16] Stephen A. Jarvis, Daniel P. Spooner, Helene N. Lim Choi Keung, Graham R. Nudd, "Performance Prediction and its use in Parallel and Distributed Computing Systems" in Proceedings of the International Parallel and Distributed Processing Symposium, 2003.

[17] George V. Iordache, Marcela S. Boboila, Florin Pop, Corina Stratan, and Valentin Cristea , "A Decentralized Strategy for Genetic Scheduling in Heterogeneous Environments", Springer-Verlag Berlin Heidelberg, 2010.

[18] I.C.Legrand, H.B.Newman (2003), "Monalisa: An Agent Based, Dynamic Service System to Monitor,Control and Optimize Grid Based Applications", European Center for Nuclear Research – CERN.

[19] Singh, Raja Manish, Sanchita Paul, and Abhishek Kumar. "Task scheduling in cloud computing: Review." International Journal of Computer Science and Information Technologies 5.6 (2014): 7940-7944.

[20] Tracy D. Braun, Howard Jay Siegel, Noah Beck, "A Comparison of Eleven Static Heuristics or Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, pp. 810-837, 2001

[21] J. Cao, D. P. Spooner, S. A. Jarvis, G. R. Nudd, "Grid Load Balancing Using Intelligent Agents", Future Generation Computer Systems, Vol. 21, No. 1, 2005, pp. 135-149.

[22] H. Izakian, A. Abraham, V. Snasel, "Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments," in Proc. of the International Joint Conference on Computational Sciences and Optimization, IEEE, vol. 1, 2009, pp. 8-12.

[23] S.Nagadevi1, K.Satyapriya2, Dr.D.Malathy3, "A Survey On Economic Cloud Schedulers For Optimized Task Scheduling", International Journal of Advanced Engineering Technology, 2013.

[24] Archana mantri, Suman Nandi, Gaurav Kumar, Sandeep Kumar, High Performance Architecture and Grid Computing Computing, International Conference HPAGC 2011, Chandigarh, India, July 2011, Proceedings.

[25] Weicheng Huai, Zhuzhong Qian, Xin Li, Gangyi Luo, and Sanglu Lu, “Energy Aware Task Scheduling in Data Centers, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications”, volume: 4, number: 2, pp. 18-38, 2013.

[26] Sung Ho Jang, Tae Young Kim, Jae Kwon Kim , Jong Sik Lee School, “The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing”, International Journal of Control and Automation Vol. 5, No. 4, December, 2012.

[27] Shenai Sudhir, Survey on Scheduling Issues in Cloud Computing, Procedia Engineering, Elsevier, 2012.

[28] M. Coli, P. Palazzari, “Real Time Pipelined System Design through Simulated Annealing,” Journal of Systems Architecture, vol.42, no. 6-7, 1996, pp. 465-475.

[29] Dzmitry Kliazovich1, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry and Samee Ullah Khan, “e-STAB: Energy-Efficient Scheduling for Cloud Computing Applications with Traffic Load Balancing”, IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing e-STAB, 2013.

[30] Dzmitry Kliazovich, Pascal Bouvry, Samee Ullah Khan, “DENS: Data Center Energy-Efficient Network-Aware Scheduling, Cluster Computing”, special issue on Green Networks, vol. 16, no. 1, pp. 65-75, 2013.

[31] Ning Liu, Ziqian Dong, Roberto Rojas-Cessa, “Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers”, IEEE 33rd International Conference on Distributed Computing Systems Workshops Task, 2013.

[32] Chia-Ming Wu, Ruay-Shiung Chang, Hsin-Yu Chan received, “A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters”, Elsevier, 2013.

[33] Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam, “Improved Cost-Based Algorithm For Task Scheduling in Cloud Computing”, IEEE, 2010.

[34] Qi Cao, Zhi-BoWei, Wen-Mao Gong, “An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing”, IEEE, 2009.

[35] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou, “A Task Scheduling Algorithm based on QoS driven in Cloud Computing”, Information Technology and Quantitative management, 2013.

[36] Mousumi Paul, Debabrata Samanta, Goutam Sanyal, “Dynamic Job Scheduling in Cloud Computing Based on Horizontal Load Balancing”, IJCTA, 2011.

[37] Huankai Chen, Frank Wang, Na Helian, Gbola Akanmu, “User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing”, IEEE, 2013.

[38] Lianying ZHOU, Xingping CUI, Shuyue WU, An Optimized Load-balancing Scheduling Method Based on the WLC Algorithm for Cloud Data Centers, Journal of Computational Information Systems, 2013.

[39] Monir Abdullaha, Mohamed Othmanb, Cost Based Multi QoS Job Scheduling using Divisible Load Theory in Cloud Computing, International Conference on Computational Science, ICCS 2013.

[40] Upendra Bhoi, Purvi N. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 4, April 2013, pp 259-264.

[41] Isam Azawi Mohialdeen, "Comparative Study Of Scheduling Algorithms in Cloud Computing Environment", Journal of Computer Science, 9 (2): 252-263, 2013.

[42] Vijay Dhir, Dr. Rattan K Datta, Dr. Maitreyee Dutta, "Grid Job Scheduling - A Detailed Study", International Journal of Innovative Research in Science, Engineering & Technology Vol.2, Issue 10, October 2013.

[43] Vijay Dhir, Dr. Rattan K Datta, Dr. Maitreyee Dutta, "Nimble@ITCEcnoGrid Novel Toolkit for Computing Weather Forecasting, Pi and Factorization Intensive Problems", International Journal of Computer Engineering & Technology (IJCET) , Vol:3 Issue:3, Dec 2012.

[44] Vijay Dhir, Dr. Rattan K Datta, Dr. Maitreyee Dutta, "Computational Grid based on Alchemi.NET framework", International Conference on Computer, Electrical, and SystemsScience and Engineering, Feb 10, 2009 WCSET 2009: World Congress on Science, Engineering & Technology Hong Kong March 23-25, 2009.