Hybrid Job Scheduling Algorithm With Process Threading Technique To Maximise Network Performance In A Cloud Environment

Rajpreet Kaur[1], Er. Diljit Kaur[2]
Reseach Scholar[1]
Assistant Professor[2]
Department of Information Technology, CEC Landran, Mohali(Punjab) CEC Landran

Abstract- Cloud server always working under heavy load due to processing requests of millions of users worldwide. Heavy load affects the performance of cloud server and cause high energy consumption and time for processing their Jobs. Existing algorithms reduces energy consumption, but still, it's much more. The proposed algorithm is a hybrid of two techniques known as multi-queue scheduling algorithm and P-Thread algorithm. Both of them are scheduled in such a way to reduce energy consumption for tasks execution on the cloud network. The overall working is divided into two different modules in this research. The first module is used to handle all the possible queues on the network with the help of multi- queue scheduling algorithm. All the queues handling and execution process at the same time is done by the first module of the proposed model. Another important module is Job in the job queues. These are handled with the help of a P-thread algorithm which makes the scheduling process more efficient and less time-consuming. The P-Thread algorithm executes all the tasks in a queue with multi-threading technique. This process reduces the waiting time between all the possible solutions and provides maximum throughput and reduce execution cost. The proposed approach compared with other scheduling approaches in terms of energy consumption, execution time and processor utilization. The results shows that performance of ourproposed approach is better in all the secnerio when compared to other existing approaches.

Keywords- Cloud Servers, cloud computing, P-thread and Multi threading algorithm, maximum throughput.

I. INTRODUCTION

Cloud computing is internet associated mode of super-computing. It is a type of communal environment, which normal puts the large system groups together by using numerous means:

a) Distributed
b) Virtualization etc.

It gives clients to variety of storage, networking and computing resources in the cloud computing infrastructure via the internet; a client puta lot of information andcontacts of computing with the help of its own computer[1].Cloud computing is consists of a number of resources that are different with one Another via some means and cost of performing jobs in the cloud using the resources of cloud is different[2], sothe scheduling of jobs in the cloud is different from the traditional methods of scheduling and so the scheduling of jobs in the cloud need better attention to be paid because the services of cloud depends on them. Job scheduling plays a key role to improve flexibility and reliability of systems in the cloud.The mail aim for job scheduling is to allocate jobs to resources in an optimized manner so as the cloud user jobs can be executed on given time bound.Cloud computing is a complete new technology. It is the development of parallel computing, distributed computing grid computing, and is the combination and evolution of Virtualization, Utility computing, Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS). Cloud is a metaphor to describe web as a space where computing has been pre installed and exist as a service; data, operating systems, applications, storage and processing power exist on the web ready to be shared [3]. To users, cloud computing is a Pay-per-Use-On-Demand mode that can conveniently access shared IT resources through the Internet. Where the IT resources include network, server, storage, application, service and so on and they can be deployed with much quick and easy manner and least management and also interactions with service providers. Cloud computing can much improve the availability of IT resources and owns many advantages over other computing techniques. Users can use the IT infrastructure with Pay-per-Use-On-Demand mode; this would benefit and save the cost to buy the physical resources that may be vacant.

II. JOB SCHEDULING

The Job management is the fundamental concept of cloud computing systems job scheduling issues are identifies with the effectiveness of the entire cloud computing framework. Job scheduling will be a mapping component from client assignments to the proper determination of assets and its execution [4]. Job scheduling is adaptable and helpful. Jobs and job streams can be planned to run at whatever point required, taking into account business capacities, needs, and requirements. Job torrents and procedures can set up every day, week after week, month to month, and yearly ahead of time[5].

The task scheduling objectives of Cloud processing are given ideal job scheduling clients, and given the complete cloud framework throughput and measure the QoS at the same time. Subsequent are the needs of job scheduling in cloud computing [6,7].

- Quality of Service-It is the description or measurement of the overall performance of a service, such as a telephony or computer network or a Cloud computing service, particularly the performance seen by the users of the network. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as error rates, bit rate, throughput, transmission delay, availability, jitter, etc[8]. The meaning of constrains here is applying QoS that users need and balancing between these QoS and fairness among the tasks. Many algorithms are improved to satisfy the requirement of cloud computing. One of these algorithms is task scheduling algorithm that calculates priority to each task depending on its attributes and then schedules first tasks that have high priority [9].
- The reduced execution time-Jobs can be apportioned into diverse modules as indicated by the needs of clients, and after that set the best running time on the ground of distinctive objectives for every job. It will enhance the QoS of task scheduling indirectly in a cloud environment [10].
- Enhanced throughput of the system-Mainly for distributed computing frameworks, throughput is a measure of framework commission planning, reorganization execution, and it is likewise [9] an objective which must be considered in plan of achievement advancement. Build throughput for clients and cloud suppliers would be advantageous for both of them[11].

III. RELATED WORK

Mansouri, Najme, GhDastghaibyfard, and A. Horri2011,[12]worked ondata grid is a geographically distribute situation that deals with large – scale data concentrated problematic. The main difficulties in data grid are job scheduling and data management. Normally, Job scheduling in grid has been intended from the perception of computational grids. In data network, the operative scheduling strategy should deliberate both computational and data storage possessions. In this paper a new job scheduling technique, called combine scheduling approach is planned that considers numeral of jobs to come in the queue, position of necessary data and the capacity of sites.Yang, Bo, XiaofeiXu, Feng Tan, and Dong Ho Park2011[13]worked as, cloud computing service concerned with features advance a new way of service provisioning called usefulness based computing. Though, toward the applied application of commercialized Cloud, they happenstance two challenges:

i) Nor is no well-defined job scheduling algorithm for the Cloud that contemplates the system state in the upcoming, mainly under overloading conditions;
ii) The current job scheduling algorithms under helpfulness computing standard do not take hardware or software failure, and retrieval in the Cloud into account.

In an effort to address these experiments, they familiarize the failure and recovery situation in the Cloud computing articles and suggest a Reinforcement Learning based algorithm to make job scheduling, fault supportable while maximizing efficacies attained in the long term.Rodero, Ivan, FrancescGuim, and JulitaCorbalan, 2009 [14] in this paper, describes and evaluates our synchronized grid scheduling plan. They take as an orientation the FCFS job scheduling strategy and the matchmaking method for the resource collection. They also current a new job scheduling policy based on backfilling that aims to progress the workloadaccomplishment, performance, avoiding punishment and the SLOW coordinated resource collection policy that deliberates the average bounded slowdown of the possessions as the main parameter to perform the reserve selection.Aparnaa, S. K., and K. Kousalya,2014 [15]in this paper, the standard algorithms does not deliberate the memory restriction of each cluster , which is one of the main possessions for scheduling data rigorous jobs. Due to this the job failure rate is also very high. To offer an explanation to that problematic Improved Adaptive Scoring Job Scheduling algorithm is presented. The jobs are recognized whether it is data intensive or computational intensive and based on that the jobs are planned. The jobs are owed by computing Job Score along with the memory condition of each cluster. Due to the active nature of grid environment, each time the status of the resources unconventionalities and each time the Job Score is totalled and the jobs are assigned to the most apposite properties.Keqin Li ,2004 [16]In this paper, they associate the performance of numerous job scheduling and mainframe allocation algorithms for grid computing on meta process. They appraise the performance of 128 mixtures of two job scheduling algorithms, four initial job ordering strategies, four processor provision algorithms, and four Meta computers by extensive reproduction.SaadBani-Mohammad ,2012 [17] In this paper described as, the concert of non-contiguous provision can be knowingly affected by the job scheduling approach used for determining the order in which jobs are particular for execution. In this paper, the routine of the well-known Greedy Offered Busy List non-contiguous apportionment strategy for 2D mesh-connected multi-computers is re-visited considering several significant job scheduling strategies. These are the First- Come-First-Served, Out-of-Order, and Window- Based job scheduling strategies. They are likened using detailed flit-level imitations. General simulation consequences based on synthetic and real assignment models indicate that the Window Based job scheduling approach exhibits good presentation when the scheduling window size is large and weighty system masses.Babbar, Davender, and Phillip Krueger,1994[18]In this paper; they exist the consequences of a achieve" study of all the proposed policies known to the authors. Initially, each of these

allocation plans was further for use with First- Come- First-Served job scheduling. In this paper they also suggest and appraise new alternatives of these strategies using the Scan scheduling correction.

IV. EXISTING ISSUES IN JOB SCHEDULING

Most data centers, by design, consume massive amounts of energy in an incongruous wasteful manner, interviews and documents show. As a result, data centers can waste 90 % or more of the electricity they pull off the grid, they are more power consumption and wastage of time. So there is a strong need of optimization above three factors CPU utilization, response time and no. of jobs executed per time[19,20].

- The current research concern is the unwanted power utilized, energy consumed and more time consumed in data[21].
- In realism, Service providers make high quality use of IaaS and PaaS for developing their services without consideration of physical hardware, while operators also can access on-demand and pay-peruse services anywhere in Cloud computing. But one of the major issue in data center's found is to manage optimum energy, power usage in the systems[22].
- In gap scheduling [23] the device goes into a sleep state whenever it is idle, and the objective is to minimize the total number of transitions (because the time spent in the active state is fixed by the input jobs). Though minimizing total power (as in power minimization) is the most natural measure, minimizing the number of transitions (as in gap scheduling) seems stronger from the point of view of approximation algorithms.
- In all previous work on these problems, tasks have arrival times, deadlines, and processing times, and the goal is to find a pre-emptive schedule that satisfies all deadlines. We consider a generalization, called multi-interval scheduling, in which each task has a list of one or more time intervals during which it can execute [24] (e.g., when the necessary resources are available), and the goal is to complete every task. One practical special case of this generalization, also considered.In this paper, is multiprocessor scheduling where each task can run on any processor, and individual processors can go into the sleep state. (To see that multiprocessor scheduling is a special case of interval scheduling, view the processor executions as laid out one after the other, so that idle gaps correspond; then a task with an arrival time and deadline becomes executable in an arithmetic sequence of time intervals.) This problem is particularly interesting given the 46 increasing prevalence of multi-core and multiprocessor architectures[25].

V. PROPOSED MODEL

The proposed algorithm is a hybrid of two techniques known as multi-queue scheduling algorithm and P-Thread algorithm. Both of them are scheduled in such a way to reduce energy consumption for tasks execution on cloud network. The overall working is divided into two different modules in this research.

- First module: It is used to handle all the possible queues on the network with the help of multi- queue scheduling algorithm. All the queues handling and execution process at the same time is done by the first module of the proposed model.
- Second module: Another important module is used to handle Jobsin the job queues. These are handled with the help of P-thread algorithm which makes the scheduling process more efficient and less time consuming. The P-Thread algorithm executes all the tasks in a queue with multi-threading technique. This process reduces the waiting time between all the possible solutions and provide maximum throughput in less execution cost.

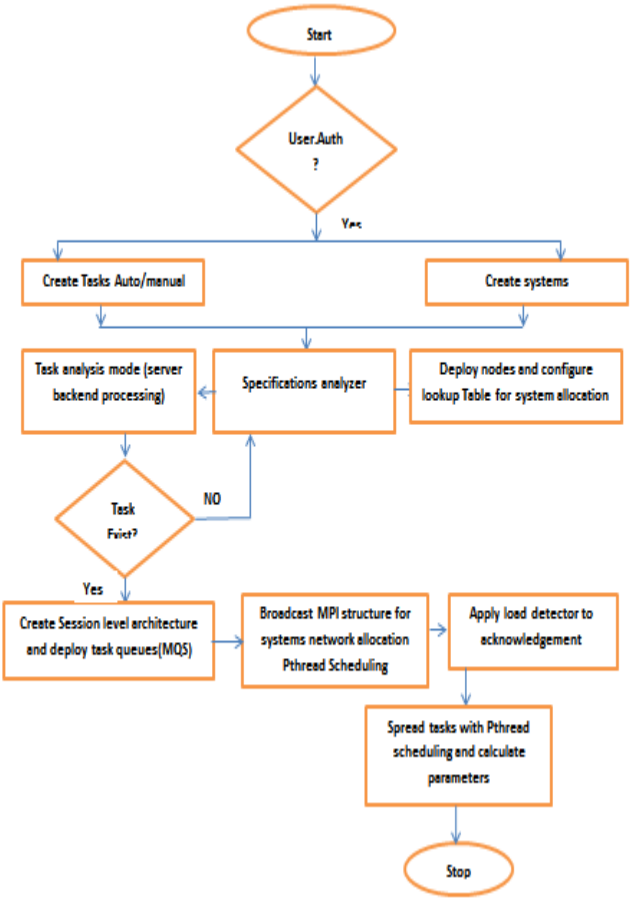

Fig. 1: Proposed Flow Chart

The proposed approach is having various steps which are used to create a virtual environment for processing tasks and compute their execution parameters. The various tasks and systems are created in tasks and system creation phase. The

process of system creation is used to create a virtual network for processing input tasks. Another phase that creates tasks which are used to form queues on cloud servers. After processing system and tasks the specification analyzer is used to generate various specification parameters for tasks and systems in a cloudenvironment. The specifications are used to estimate the execution results of input tasks based on network processing environment. The next step for processing in hybrid technique is MQS algorithm which used to provide the outer module processing as discussed. All the queues are executed at a time to reduce the waiting and response time of execution. The inner structure is managed with the help of P-thread algorithm along with MPI technique. It reduces the waiting time with the help of the message passing interface and multi-threading technique. The message passing interface reduces the network cycles for system allocates and tasks allotment. The proposed approach tested in several test cases as compared to other existing approaches. In all the test cases the results with proposed approach are shows enhancement in job scheduling.

VI. RESULTS AND DISCUSSIONS

In this section, we discussed the explain the results and research work done in the .NET simulation tool with SQL database used.The proposed approach is having various steps which are used to create a virtual environment for processing tasks and compute their execution parameters.

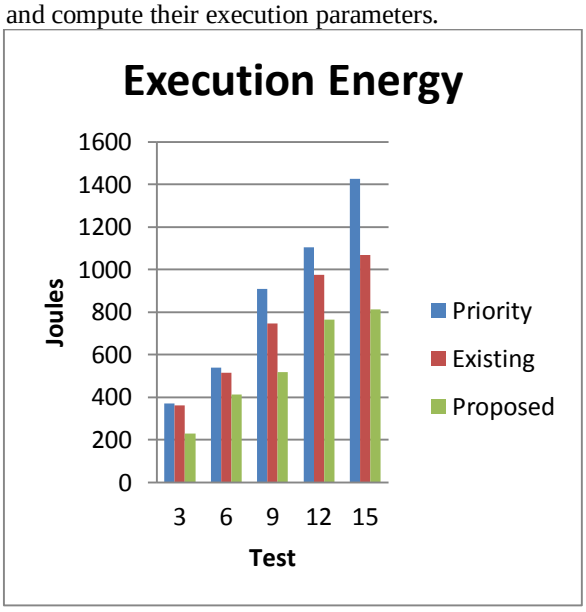

Fig. 2: Execution Energy

The above figure shows energy consumption for various test cases and comparison with other existing techniques. The energy consumption of the hybrid algorithms is less as compared to the base and other standard algorithm like a priority. The stable performance of the proposed algorithm describes the better performance in all the cases.

Table 1: Comparison between priority , existing and proposed methods

Algorithm	3	6	9	12	15
Priority	370	540	910	1103	1426
Existing	362	514	746	975	1068
Proposed	229	412	519	765	813

The above table shows various energy values for test cases in a cloud environment. The processing results shown less energy consumption of the proposed approach in all the possible cases as compared to other scheduling approaches.

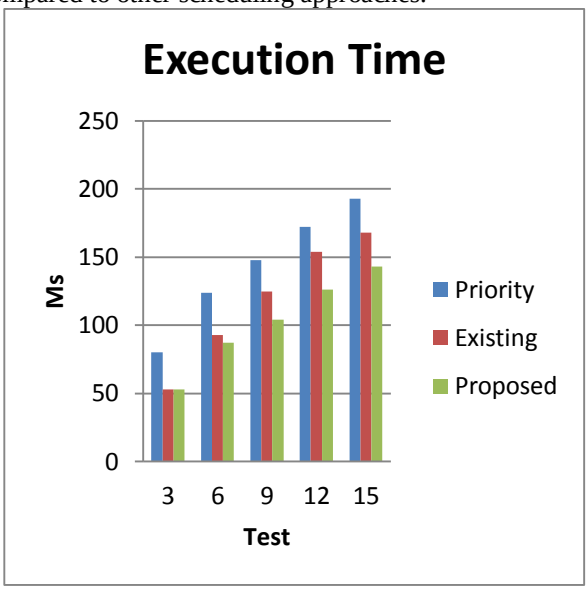

Fig. 3: Execution Time

The execution time also shown the end to end delay of execution for cloud users. The less execution time shows fast processing speed of the network. It is also attached to the energy consumption of the processing. An increment in the processing time causes more energy consumption because the energy consumption also affected by the waiting time of processing. Here in the figure the processing time shows better results in all the cases as compared to other existing approaches.

Table 2. Comparison between priority, existing and proposed work in execution time

Algorithm	3	6	9	12	15
Priority	80	124	148	172	193
Existing	53	93	125	154	168
Proposed	53	87	104	126	143

Above table used to show tabular form of execution results and shows difference between various test cases in terms of execution time. The proposed algorithm consumes less time as compared to the other existing approaches in a job scheduling environment.

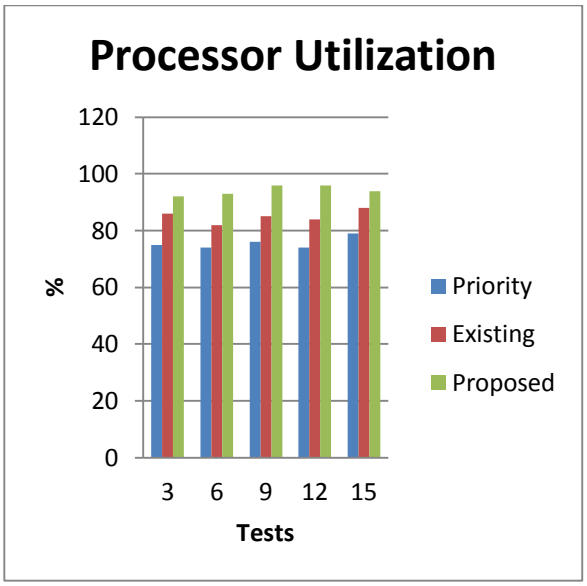

Fig. 4: Processor Utilization

Processor utilization parameter is used to check the efficiency of the scheduler that how efficiently scheduler used their systems. More utilization shows better performance of scheduling algorithm. Here in the figure several test cases performed to check the performance of scheduling algorithm. In all the cases hybrid approach shows better performance and utilize the network properly.

Table 3. Comparison between priority, existing and proposed work in processor utilization

Algorithm	3	6	9	12	15
Priority	75	74	76	74	79
Existing	86	82	85	84	88
Proposed	92	93	96	96	94

In percentage the network utilization is shown in the table. The several algorithms are compared in this table in scheduling environment. The performance of the proposed approach is higher than the other as it utilizes the network more efficiently as compare to other scheduling techniques.

VII. CONCLUSION AND FUTURE SCOPE

In this research paper two techniques,i.e. multi-level queue scheduling Algorithm and P-thread Algorithm have been taken into consideration by replacing the already existing techniques. The system was planned with the prime focus on optimization so as to achieve accurate results.Job Scheduling is intended to accomplish not solitary competent processing as well as the use of computing infrastructure, but also minimize energy consumption. It is important for confirming in which the future development of Cloud-computing is quite sustainable. Then, Cloud-computing with progressively prevalent front-end client devices interrelating with back-end data centers will cause an enormous escalation of energy usage. To address this issue, data center resources necessary to be accomplished in an energy-efficient manner to drive Green Cloud computing has been proposed in this work. Multilevel queue scheduling algorithm and P-Thread Scheduling Algorithm has been utilized. And the results are being evaluated using energy, Time and processor utilization parameter. The proposed approach performing well in all the cases and save maximum energy for job execution in the cloud environment.

In future scope, the performance of the Hybrid algorithm is good in case of multiple user and multi-system network. In future algorithm can be optimized with an optimization technique. In this, if any the technique can arrange the queues and minimize the waiting time of tasks over a cloud network. Then the cost of execution and response time will automatically reduce.

VIII. REFERENCES

[1]. Min, Dugki, and Matt W. Mutka. "Efficient job scheduling in a mesh multicomputer without discrimination against large jobs." Parallel and Distributed Processing, 1995. Proceedings. Seventh IEEE Symposium on. IEEE, 1995.

[2]. Kim, Jong-Seon, and Daniel C. Lee. "Weighted round robin packet scheduler using relative service share." Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE. Vol. 2. IEEE, 2001.

[3]. Mishra, Manoj Kumar, PrithvirajMohanty, and G. B. Mund. "A modified grouping-based job scheduling in computational grid." Engineering (NUiCONE), 2011 Nirma University International Conference on. IEEE, 2011.

[4]. Tang, Wei, Dongxu Ren, Zhiling Lan, and Narayan Desai. "Adaptive metric-aware job scheduling for production supercomputers." In Parallel Processing Workshops (ICPPW), 2012 41st International Conference on, pp. 107-115. IEEE, 2012.

[5]. Zeng, Chengkuan, Jiafu Tang, and Huabo Zhu. "Two heuristic algorithms of job scheduling problem with inter-cell production mode in hybrid operations of machining." Control and Decision Conference (CCDC), 2013 25th Chinese. IEEE, 2013.

[6]. Zhu, Chunsheng, Xiuhua Li, Victor CM Leung, Xiping Hu, and Laurence T. Yang. "Job scheduling for cloud computing integrated with wireless sensor network." In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, pp. 62-69. IEEE, 2014.

[7]. Ananth, Alaka, and K. Chandra Sekaran. "Game theoretic approaches for job scheduling in cloud computing: A survey." Computer and Communication Technology (ICCCT), 2014 International Conference on. IEEE, 2014.

[8]. Aparnaa, S. K., and K. Kousalya. "An Enhanced Adaptive Scoring Job Scheduling algorithm for minimizing job failure in heterogeneous grid network." Recent Trends in

Information Technology (ICRTIT), 2014 International Conference on. IEEE, 2014.

[9]. Karthick, A. V., E. Ramaraj, and R. Ganapathy Subramanian. "An efficient multi queue job scheduling for cloud computing." Computing and Communication Technologies (WCCCT), 2014 World Congress on. IEEE, 2014.

[10]. Cheng, Dazhao, Jia Rao, Changjun Jiang, and Xiaobo Zhou. "Resource and deadline-aware job scheduling in dynamic hadoop clusters." In *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pp. 956-965. IEEE, 2015..

[11]. Gupta, Abhishek, H. S. Bhadauria, Annapurna Singh, and Jagdish Chandra Patni. "A theoretical comparison of job scheduling algorithms in cloud computing environment." In *Next Generation Computing Technologies (NGCT), 2015 1st International Conference on*, pp. 16-20. IEEE, 2015.

[12]. Mansouri, Najme, GhDastghaibyfard, and A. Horri. "A Novel Job Scheduling Algorithm for Improving Data Grid's Performance." P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011 International Conference on. IEEE, 2011.

[13]. Yang, Bo, XiaofeiXu, Feng Tan, and Dong Ho Park. "An utility-based job scheduling algorithm for cloud computing considering reliability factor." In *Cloud and Service Computing (CSC), 2011 International Conference on*, pp. 95-102. IEEE, 2011.

[14]. Rodero, Ivan, FrancescGuim, and JulitaCorbalan. "Evaluation of coordinated Grid scheduling strategies." High Performance Computing and Communications, 2009. HPCC'09. 11th IEEE International Conference on. IEEE, 2009.

[15]. Aparnaa, S. K., and K. Kousalya. "An Enhanced Adaptive Scoring Job Scheduling algorithm for minimizing job failure in heterogeneous grid network." Recent Trends in Information Technology (ICRTIT), 2014 International Conference on. IEEE, 2014.

[16]. Li, Keqin. "Experimental performance evaluation of job scheduling and processor allocation algorithms for grid computing on metacomputers."Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International. IEEE, 2004.

[17]. Bani-Mohammad, Saad. "On the performance of job scheduling for noncontiguous allocation in 2D mesh-connected multicomputers."Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean. IEEE, 2012.

[18]. Babbar, Davender, and Phillip Krueger. "A performance comparison of processor allocation and job scheduling algorithms for mesh-connected multiprocessors." Parallel and Distributed Processing, 1994. Proceedings. Sixth IEEE Symposium on. IEEE, 1994.

[19]. Huang, Daochao, Peng Du, Chunge Zhu, Hong Zhang, and Xinran Liu. "Multi-resource packing for job scheduling in virtual machine based cloud environment." In *Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on*, pp. 216-221. IEEE, 2015.

[20]. Jin, Shiyuan, Guy Schiavone, and DamlaTurgut. "A performance study of multiprocessor task scheduling algorithms." The Journal of Supercomputing43.1 (2008): 77-97.

[21]. Ding, Shun-Li, Jing-Bo Yuan, and Jiu-Bin Ju. "An algorithm for agent-based task scheduling in grid environments." Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. Vol. 5. IEEE, 2004.

[22]. Kaur, Rajveer, and SupriyaKinger. "Analysis of Job Scheduling Algorithms in Cloud Computing." International Journal of Computer Trends and Technology (IJCTT) 9.7 (2014): 379-386.

[23]. Sarkinen, Scott A., and Scott A. Davidson. "Multi-service queuing method and apparatus that provides exhaustive arbitration, load balancing, and support for rapid port failover." U.S. Patent No. 7,151,744. 19 Dec. 2006.

[24]. Nichols, Kathleen M. "Method and apparatus for providing differentiated services using a multi-level queuing mechanism." U.S. Patent No. 6,944,128. 13 Sep. 2005.

[25]. Veerasamy, BalaDhandayuthapani, and G. M. Nasira. "Java Native Pthread for Win32 Platform." *Computing and Communication Technologies (WCCCT), 2014 World Congress on*. IEEE, 2014.