ISSN : 2350-0743

www.ijramr.com

International Journal of Recent Advances in Multidisciplinary Research
Vol. 04, Issue 11, pp.3080-3086, November, 2017

RESEARCH ARTICLE

TASK SCHEDULING ALGORITHMS FOR PARALLEL PROCESSING

***Gargi Patil, Gursimar Kaur, Sukriti Jain and Gopichand, G.**

School of Computing Science and Engineering, Vellore, Tamil Nadu, India

ARTICLE INFO

Article History:
Received 09th August, 2017
Received in revised form
28th September, 2017
Accepted 04th October, 2017
Published online 30th November, 2017

Keywords:

Parallel computing, Task scheduling, Advantages, Disadvantages, Factors considered.

ABSTRACT

In computing, Scheduling refers to a set of policies which define the order of execution of processes. The scheduling for the processor is done to keep it as busy as possible. Parallel computing includes scheduling of multiple processors, and managing the resources for all the processors. The scheduling in multiprocessors is more difficult than scheduling in a single processor unit. In scheduling of multiple processors it should be ensured that any processor should not be overloaded and any processor should not be under loaded. So the overall system should be balanced. In parallel processing system, as there will be multiple processors as well as multiple queues, so there is need of scheduling multiple queues simultaneously. The scheduling algorithms for multiple processors should be capable of scheduling all the queues optimally. A descriptive study of different types of task scheduling algorithms along with concise analysis of different parameters is considered in this paper.

INTRODUCTION

The paper mainly focuses on various task scheduling algorithms.A task is simply any small chunk of work to be completed in allotted time period. The different tasks provided by the users to the task scheduler are dispatched based on the resource availability. Various parameters contribute for the task scheduling to improve theoverall process efficiency. The task inputted by the user can be of any form which may include processing, querying data, accessing software. The servers are assigned to the specific tasks. The main job of the processor is to process the task and provide the output or result to the user.
The various jobs submitted by the users to the scheduler are analysed based upon the status of resources available and their associated properties. Then, the resource allocation to various tasks is done based upon task requirement. The tasks are assigned to various virtual machines. Assignment of tasks to the processors for good scheduling must be optimal and efficient. A good scheduling algorithm improvises maximum CPU-utilization, cumulative throughput and turnaround time. Various parameters can be considered in different ways for task scheduling process. The scheduling can be done statically in which resources are allocated at compile time or dynamically where allocation is done at runtime.

Literature Survey: Cloud computing gives us a gist ofdelivery and supplement replica for Information Technology services which depend internet on pay as per usage foundation. In this paper, the authors have written about a scheduling algorithm, which reduces execution rate andimplementation time as well.

Corresponding author:* **Gargi Patil,
School of Computing Science and Engineering, Vellore, Tamil Nadu, India.

An upgraded hereditary algorithm is established by merging two already available scheduling algorithms for the scheduling processes, considering the magnitude of their computing and computational complexity volume of elements that are going to be processed. The authors have conducted experiments with these algorithms and the results obtained by experiments show that, under the high load conditions, the proposed method gives a good performance (Braun, 2001). This paper is unique in its approach as it has worked on 11 algorithms that are highly useful. Choosing the optimal task scheduling algorithm to utilize in a required surroundings, however, is still a huge problem, because collations are often accompanied by various underlying hypothesis in all of the original studies of each of the heuristic.Therefore, a group of eleven heuristics have been chosen, adapted, used, and studied under a number of general assumptions by the authors. It is understood that these algorithms obtain a mapping using statistical methods. The paper also assumed that a meta-task is mapped and that the aim is to reduce the overall time of execution of the meta-task (Braun, 1997). In this paper, the authors have proposed a novel task scheduling related algorithm with an aim to achieve the purposes of better functioning, quick running time and scalability. The proposed algorithm is nothing but a parallel algorithm which gives rather useful and quick fix results in less time. By making the list for scheduling and using it as a chromosome, the required scheduling list can be generated using this algorithm, which will give us the optimal schedule. The most important fact of this algorithm resides in its 2 important operators: the proper mutationand crossover. The following operators have been successful in combining the fundamental parts of better scheduling lists to build improved lists. The proposed algorithm that has been given in this context is constructed through an extremely careful comparison

with two methods which are highly popular with respect tothe time complexityand performance. It outshines both of the heuristics whilst using comparatively lesser running-time in all situations (Maheswaran, 1999). Task Scheduling and Dynamic mappingalgorithms for the group of tasks which are not dependent, with the help of distributed-computing systems are scrutinised. 2 different mapping types algorithms are considered: the batch mode and the immediate mode. 3 of the recent heuristics, one for the batch-mode and two others for the immediate-mode, are introduced by the authors as a part of their research. Parallel learning is performed to understand mentioned heuristics with few already existing heuristics. The simulation results tell us the appropriate dynamic task scheduling algorithm to utilize in available surroundings. All of this relies on various factors that are mentioned in the paper at the later stages (He, 2003). Task scheduling is an indivisible part of the world of computing. Along with the arrival and the up rise of the ubiquitous computing andGrid technology, new problems arise in task-scheduling on the basis of factors like way of service, security, and the very less centralised control amongst the given distributed administrative domains. In this paper, the authors have given a novel Quality of Service inspired Grid computing based task scheduling algorithm. The unique methoddesigned by the authors on the basis of a generic adaptive heuristics task based algorithms that have Quality of Service leadership. The authors have studied and analysed the given algorithm in a simulated-Grid surroundings. These experimentally determined outputs show us that the new Quality of Service guided algorithm can lead to very high performance improvement for a large number of applications (Mathew, 2014). The performance and effectiveness of the world of cloud computing and its various structures have always relied on the execution of the tasks of the user that the cloud system retrieves from the system. .A significant role is played by scheduling of user tasks in improvising the cloud service performance. Scheduling of tasks is one amongst the most important categories of scheduling to be applied in the domain of computing considering the type of data it handles. This paper gives us a detailed understanding of different task scheduling methods that are being used in the world of cloud and all its subsidiaries. A detailed analysis of different scheduling factors used in these given methods is also scrutinised in this paper (Pandey, 2010). virtualized resources are provided to applicants dynamically with the help of task scheduling algorithmsfactors such as cost arising from transfer of data between resources and the overall cost of execution must also be taken into consideration in addition to execution time.The authors of the paper have presented a particular particle optimization based algorithm to schedule applications taking computation cost and transmission cost into account. The author varying the computation and communication costs experiments the workflow application cross verifying the cost savings by implementing PSO and another existing algorithm 'Best Resource Selection' i.e.(BRS). And finally showed that PSO saves 3 times the cost than BRS and also the work load is better distributed in PSO (Singh, 2014). A single criteria alone cannot be based upon to schedule tasks but a number of rules and regulations need to be followed which can be termed as agreement amongst cloud users and providers. This agreement meets the expectations of the user in terms of quality of the service. Since there are numerous tasks running in the provider's side it is a hefty job for the provider to meet all the requirements of the user. Viewing the task scheduling problems for finding an optimal allocation of set of subtasks among the available group of resourcesto achieved the expected goal for the processes. In this paper the author have compared different algorithm based on their adaptability, feasibility, suitability for cloud computing and a hybrid approach has been provided to enhance the performance (Sindhu, 2015). The scheduling algorithms used in the field of Computer Science, particularly in the field of operating systems, should assign the tasks in a way where there is a proper balance between improving the efficiency and quality of service and also simultaneously maintaining the effectiveness and equality among all the jobs. The authors in this paper aim at scrutinising the different task scheduling methods. A good task scheduling method will help greatly in proper utilization of the allocated resources. Many scheduling methods have been developed carefully by the researchers like PSO (Particle Swarm Optimization), GA (Genetic Algorithm), Min-Min, Priority based Job Scheduling Algorithm and Max-Min. This paper has given a review of certain papers on management and task scheduling in cloud computing (Shinde, 2017). The essence of the real-time scheduling algorithms present in the world of computing have a direct impact on the real-time system's functioning. The authors have studied popular scheduling algorithms such as Deadline Monotonic, Rate Monotonic, Least laxity First, Earliest Deadline First, Group Priority Earliest Deadline First and Group Earliest Deadline First for periodic task. Each of these scheduling algorithms have been used in particular situations and have their own advantages and disadvantages. The authors have observed that the choosing of a particular scheduling algorithm is very much essential in obtaining a real-time system. The paper is concluded by discussing the outcomes of the Real-Time scheduling heuristic algorithm survey.

Proposed Work

The main objective of this paper is to study the significance of task scheduling algorithms for parallel computing and analysing existing algorithms. The study will be done on the basis of of performane metrics, parameters considered in each algorithm, advantages and disadvantages of the algorithms.

Overview

Task scheduling is generally categorized into two subparts as distributed scheduling and centralized scheduling. Distributed scheduling is the part which deals with resource management component of a system which moves tasks around the processors in order to achieve load balance and improved overall performance. Task scheduling can be performed in two ways and in two different environments- homogeneous environment and heterogeneous environment, on dependent tasks or independent tasks. This type of scheduling is called centralized scheduling, which is the type that consists of a single scheduler who does all the mapping, on the other hand distributed scheduling partitions the tasks on different schedulers. Distributed scheduling is very complex for implementation but as a result of distributed work load the processor cycles are saved. It is very easy to implement compared to distributed scheduling, but it has a major drawback of single point failure and thus lack scalability tolerance. Further, distributed scheduling is generally classified into heuristic algorithms and hybrid algorithms. Static

scheduling algorithms assume that all tasks come together at the same time instance and they are not dependent on the model resource's states and availability. First in First out, First come First serve, genetic algorithm are examples of static scheduling algorithms. The first come first serve methods generally collect the tasks and queues them tillthe resources are available. Once the resources are available assigning of task is done amongst them on the basis of their arrival time. In dynamic scheduling, the nature of tasks is dynamic. In thistasks come at different time instances and it depends on the model processor's state. MCT, MET, k-percent are some examples of dynamic task scheduling algorithms. The classification of the task schedulling algorithms for parallel computing is given below,

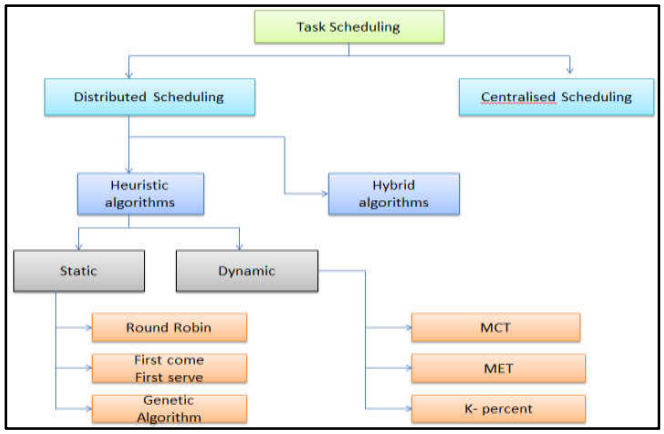

Figure 1. Classification of Task Scheduling Algorithms

The ultimate aim of task scheduling is to get best throughput from the available system resources and the bettercomputing performance. From the programmer's aspectStatic scheduling is very easy to implementon the other hand dynamic scheduling is better for real world problems. Dynamic scheduling reduces the amount required to be paid for running the scheduler.

Study of Task Scheduling Algorithms

1. MCT-Minimum Completion Time

Minimum Completion-Time is abbreviated as MCT. This task scheduling technique allocates tasks to the processors (in case of cloud computing, Virtual Machines) or resources on the basis of the best completion time which can be predicted for the randomly ordered tasks. This technique calculates the completion time and find the processor which has the Minimum Completion Time for the specific job. The completion time is mathematically found by summing together the execution time and the ready time of the resource.

Completion-Time (Ct) = Execution-Time(Et) + Resource-Ready(Rt)

Every task is allocated to the processor or to the resource that has the least Ct. With this algorithm, some tasks are assigned to the processors or resources which do not have minimum Et. The minimum Ct algorithm aims to collaborate the advantages of OLB algorithm and MET algorithm as well as avoids their drawbacks.

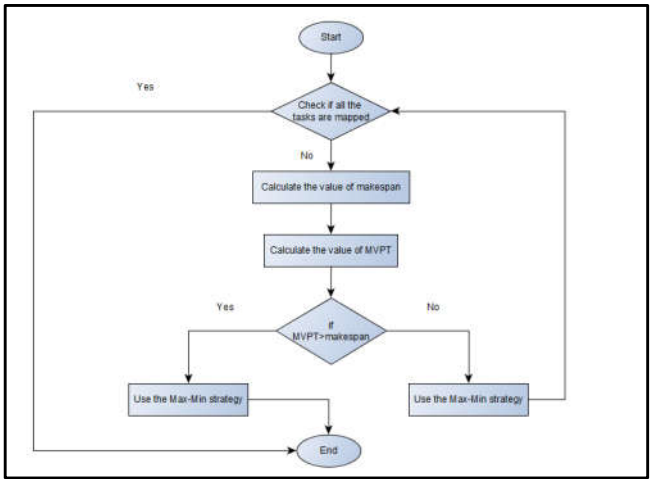

Figure 2. MCT Flowchart

2. MET-Minimum Execution Time

Minimum Execution Time is abbreviated as MET. This algorithm allocates the jobs to the processors (in case of cloud computing, Virtual Mac hines) or resources on the basis of the best completion time which can be predicted for the specific job without considering the availability of resources. The main idea of this algorithm is to allocate a task to the processor or resources on the basis of the minimum Et. Itsometimes results to highimbalance of load.This is because of the assignment without considering the availability of the resources. This algorithm assigns tasks on the processors on the basis of which processor takes less tome for execution. It chooses the best processor for execution but does not check for the availability of resources during scheduling. It leads to load imbalance.

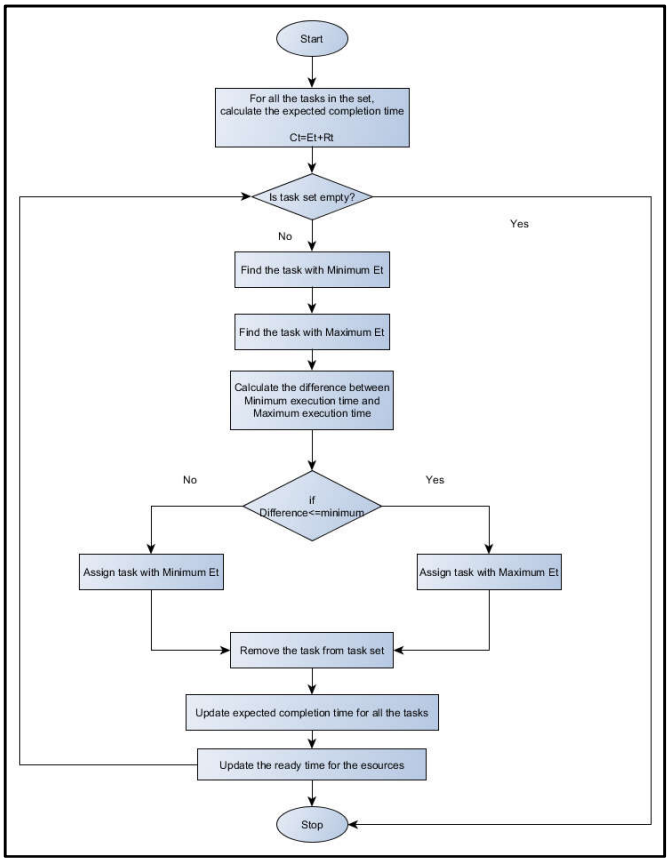

Figure 3. MET Flowchart

3. Min-Min and Max-Min

The Min-min algorithm firstly chooses the smallest task from among the given tasks and then the assigned processor for the smallest task provides the least completion-time (fastest processor) for the task. The algorithm hence increases the total completion-time and the makespan of all tasks, but the load of the processors before scheduling is not considered. In here, the task's execution-time and expected completion-time are similar. The longer tasks have the probability to wait, for the completion of smaller tasks, butoverall throughput of the proposed modelis improved.

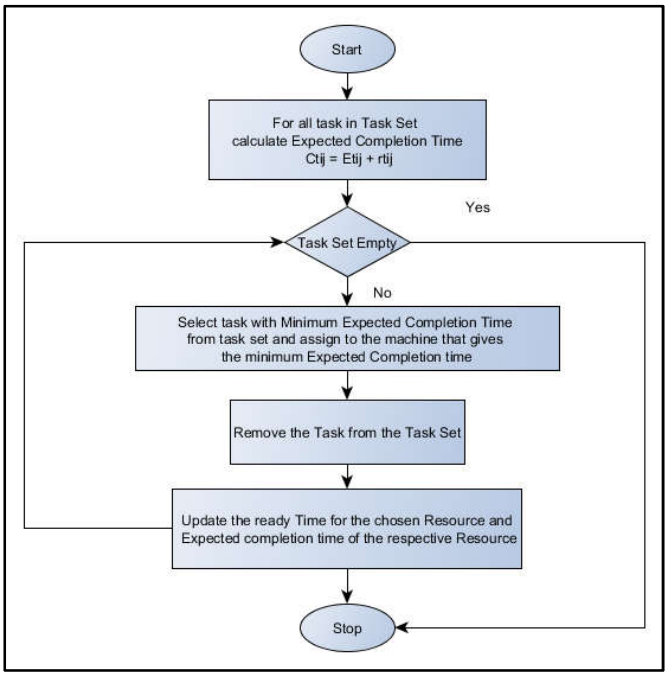

Figure 4. Min-Min Flowchart

Max-Min which has similarity to previously mentioned min-min algorithm, selects the maximum completion time also referred as longest task, that is scheduled first on the best processor available. The algorithm depends on minimum completion time of that specific task from the available processors.Starving of smaller tasks may also occur, alsothere is no consideration for load balancing. The makespan and the throughput of the system are also increased in comparison to the min-min method as the makespan of all the tasks is decided by the longest task. The advantage of Max-Min over min-min is that the execution of the longer task can be done first and in faster processors and simultaneously smaller tasks on other possible processors can be executed parallelly. Hence, the max-min results in a balanced load and better makespan than min-min approach. These algorithms have better makespan in comparision to other algorithms while the major drawback includes poor load balancing and it does not consider QoS factors.

4. Round robin

Round robin is preemptive job scheduling algorithm in which each process is allocated with a particular time to execute called quantum. That particular time is known as time quantum which is assigned to each and every processor equally and in a circular order. Time quantum is not based on priority of processes and hence equal time is given to all processors. The first process to arrive is executed for the given quantum gets preempted after completion of particular time period and another process executes for given time period.

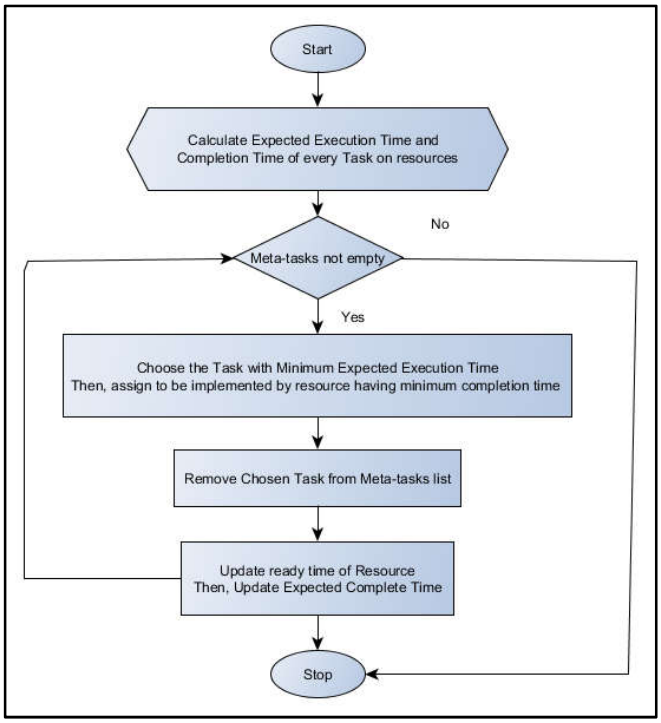

Figure 5. Man-Min Flowchart

For fair process scheduling round robin scheduler follows time sharing which gives equal time slot to processors and the scheduler interrupts the process if it is not completed within the time assigned to it and the job is resumed next time when quantum is given to that process. This state of the process is saved and stored in memory. It helps in resuming from the point where process was interrupted. If quanta are larger than the job size in that case process with large jobs will be given priority over other processes. Round robin can be used as an alternative process for first come first serve.Round robin scheduling is done by multiplexer, switch or router where all three have separate queue for data flow and flowing of this data can be analyzed from its source and destination.In a periodically repeated manner this algorithm allows the data packets to take their turn in active flow of data. If one flow is out then another data packet will take its place hence it prevents the link resource from being unused and also helps in work conserving.

Deficit weighted round robin Network scheduler use this algorithm. In this algorithm a scheduler is provided with n number of flows and q quantum's. All the non-empty queues are scanned by DRR and when a a non empty queue is selected its counter get increment by its quantum value and value of deficit counter becomes the maximum no of bytes which can be sent in one go. If the size of packet is smaller than the deficit counter that packet is sent and now the value of counter is decremented by size of the packet. When finally the counter value becomes insufficient or queue is empty then the value of deficit counter will be reset to zero and scheduler will go to another queue.

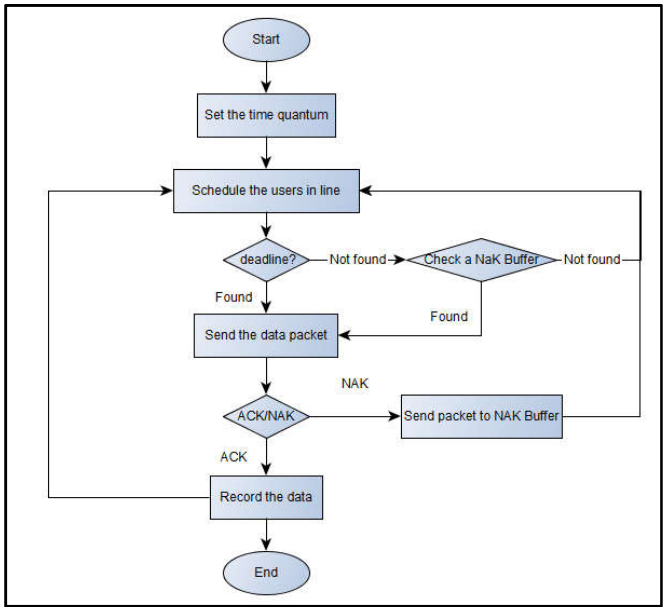

Figure 6. Round Robin Flowchart

Weighted round robin In this algorithm network face controller has connection or packet flow and each packet in turn has it own packet queue. In WRR classification of packets is done into various services like real time, file transfer and it is then assigned to a queue specifically for that particular service class. In this more than a single packet can be send in one round when the bandwidth of queue is higher or each queue is allowed to send only single packet but queues with higher bandwidth can visit multiple times in a single round of service. It is the best scheduling algorithm for getting better response time.It is starvation free.In this algorithm each process get equal time to get execute. But according to round robin all processes are equally important because of which it sometimes produces bad results.It is time consuming specially when there are many processes.

5. Opportunistic load balancing

In parallel computing load balancing is a technique that distributed heavy workload evenly across all the nodes. Balancing level in a system is influenced by ordering of tasks in a queue. In opportunistic load balancing work load is assigned to all the nodes in free order. It is a static load balancing so it has nothing to do with the current workload on the virtual machine. It tries to keep each node as busy as possible. Opportunistic load balancing assigns tasks to next available machine arbitrarily in considerate about the execution time of task on next available machine. In this Heuristic load balancing arbitrarily chooses cloudlet from a cloudlets batch which is then allocated to next VM which is estimated to be available irrespective of cloudlet executing time on that VM making this whole process poor because task progress in slow manner as it does not calculate or keep a check of current execution time of the node. So instead of that we can use simple scheduling algorithm with load balancing because minimize the make-span with maximum resource utilization in a low computational complexity grid system. It has Better resource utilizing ratio.Its advantage is Proper load distribution. But in this algorithm expected execution time is not considered. It does not calculate current time execution of nodes.

6. Genetic Algorithm

Genetic algorithm is a type of heuristic search algorithm that is developed based on natural evolution. Genetic algorithms are known to perform near optimal scheduling where the tasks are resources assigned based on schedules which are individual solutions. The algorithm decides which resource should be allotted to the task. Genetic algorithms are based on the concept of biology of population spawning. The steps followed in this algorithm are evaluation, selection, cross over, and mutation. The final results' schedule is a list of tasks called chromosome. Genetic algorithms are well known methods used for finding huge solution spaces.Each process is depicted the vector within which every block in the vector is the task in array of tasks.value of the block depicts location of the machine in which task mapping is done.Every job depicts a job. There exists a fitness value for each chromosome depicting the overall execution time belonging to the tasks derived from the mapping of tasks done to the resources consisting of that particular chromosome. This algorithm makes use of previous results to improve its current ability to create accurate mappings, and hence results in survival of the fittest. The function of fitness is utilized to observe nature of the individual in that population as indicated by given enhancement objective.The genetic algorithm minimizes time hence giving profit to the service provider and also reduces the cost of maintenance of the resources.

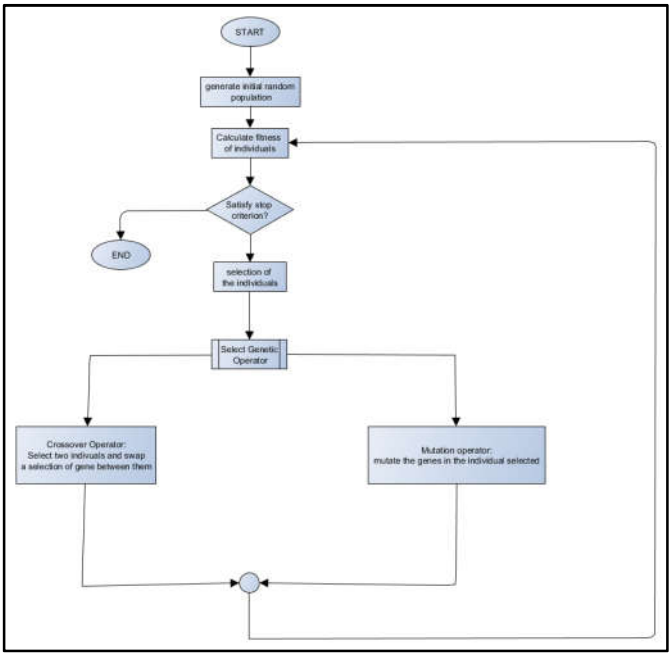

Figure 7. Genetic Algorithm Flowchart

7.Earliest Deadline first algorithm (EDF)

Earliest deadline first is a dynamic scheduling algorithm where the priority of a task kaaps changing during its execution.The priority of a task is related inversely to its deadline, the smaller the deadline higher the priority.

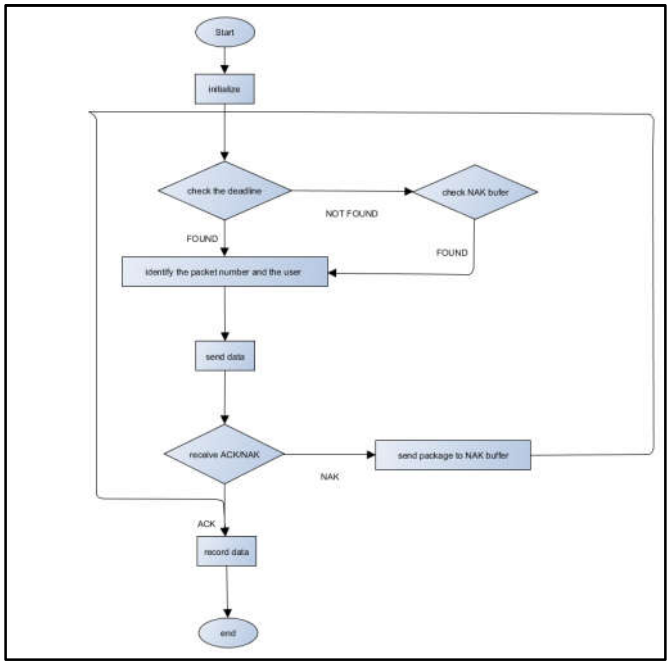

Figure 7. Genetic Algorithm Flowchart

So the highest priority job has the smallest deadline.if two tasks have the same deadline then one of those is chosen at random.EDF is an optimal scheduling algorithm that provides 100% cpu utilization.EDF minimizes the miss ratio thus providing best performance.EDF guarantees every deadline at higher loading.however,when the system is overburdened the number of processed that miss their deadlines is highly unpredictable.EDF has low number of context switches and moreover there is no need to assign priorities manually.The optimality of EDF is very high and the cpu can be utilized completely.But EDF is not provided by commercial RTOS,because of certain disadvantages.It is highly unpredictable and less controllable and also requires more overhead to be implemented.

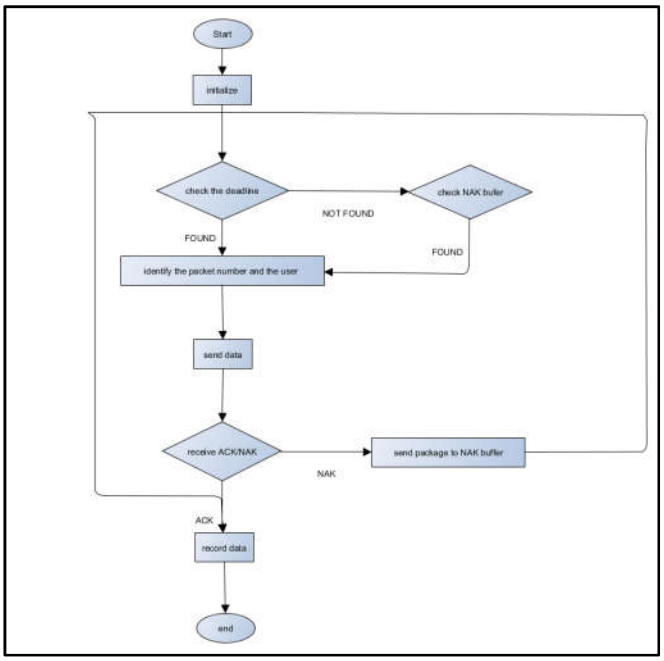

Figure 8. EDF Flowchart

RESULTS

The various task scheduling algorithms were studied referring different papers and their proposed work on parallel computing for task scheduling. The advantages and disadvantages of each algorithm along with the method of approach was studied in detail. It was found that each algorithm has certain constraints by which it is bounded to perform. Some algorithms prove to be superior than other algorithms in certain factors.

Conclusion

Task scheduling is an important aspect of parallel and distributed computing. It is very important to schedule the tasks properly in order to achieve the efficiency and best predictable performance. The task scheduling algorithms for parallel processing are indeed specific to conditions and the environment in which they need to be executed. Each algorithm has a specific way of approach and implementation which is efficient to some systems while not others. Hence, an appropriate algorithm which suits the input requirements along with the working environment should be executed for efficient results.

Future Work

The future work would include implementing these algorithms in different computing environments and trying to analyse their performance.Also, based on the observations obtained, the decision as to which algorithm suits best for specific environment for maximum output gain and high efficiency can be concluded.

REFERENCES

Adam, T.L., Chandy, K.M. and Dickson, J. 1974. "A Comparison of List Scheduling for Parallel Processing Systems," Communications of the ACM, vol. 17, Dec. pp. 685-690.

Anousha, S., & Ahmadi, M. 2013. An Improved Min-Min Task Scheduling Algorithm in Grid Computing. *Grid and Pervasive Computing Lecture Notes in Computer Science,* 103-113. doi:10.1007/978-3-642-38027-3_11

Arezou Mohammadi and Selim G. Akl," Scheduling Algorithms for Real-Time Systems", Technical Report No. 2005-499, July 15, 2005.

Braun, T. D., Siegel, H. J., Beck, N., Bolo ni, L. L., Maheswaran, M., Reuther, A. I., Robertson, J. P., Theys, M. D. Yao, B., Hensgen, D. and Freund, R. F. 2000. ``A Comparison Study of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems," Technical Report TR-ECE-00-4, School of Electrical and Computer Engineering, Purdue University, Mar.

Braun, T. D., Siegel, H. J., Beck, N., Bölöni, L. L., Maheswaran, M., Reuther, A. I., ... & Freund, R. F. 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed computing*, *61*(6), 810-837.

Etminani, K. and Naghibzadeh, M. 2007. "A Min-min Max-min Selective Algorithm for Grid Task Scheduling,"The

Third IEEE/IFIP International Conference on Internet, Uzbekistan.

Etminani, K. and Naghibzadeh, M. 2007. A Min-Min Max-Min selective algorihtm for grid task scheduling. *2007 3rd IEEE/IFIP International Conference in Central Asia on Internet*. doi:10.1109/canet.2007.4401694

Haluk, T., Salim, H., and Wu, M.Y, "Performance-effective and low complexity task scheduling for heterogeneous computing", IEEE Transaction on Parallel and Distributed Systems, vol. 13, no.3 pp: 260-274, Mar 2002.

He, X., Sun, X. and Von Laszewski, G. 2003. QoS guided min-min heuristic for grid task scheduling. *Journal of Computer Science and Technology*, *18*(4), 442-451.

Hemamalini, M. 2012. Review on grid task scheduling in distributed heterogeneous environment. *International Journal of Computer Applications*, *40*(2), 24-30.

Hermann Kopetz, "Real-Time Systems: Design Principles for Distributed Embedded Applications", Springer, second edition.

Hou, E.S.H., Ansari, N. and H. Ren, 1994. "A Genetic Algorithm for Multiprocessor Scheduling," IEEE Trans. Parallel and Distributed Sys., vol. 5, no. 2, Feb. pp. 113-120.

Jane W.S. Liu, Real-Time Systems, Pearson Education, India, pp. 121 & 26, 2001

Kaur, S. and Verma, A. 2012. An efficient approach to genetic algorithm for task scheduling in cloud computing environment. *International Journal of Information Technology and Computer Science (IJITCS)*, *4*(10), 74.

Kwok, Y. K. and Ahmad, I. 1997. Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm. *Journal of Parallel and Distributed Computing*, *47*(1), 58-77.

Liu, C. and James Leyland, January 1973, "Scheduling algorithm for multiprogramming in a hard real-time environment", *Journal of the Association for Computing Machinery,* 20(1): 46-61.

Madni, S. H., Latiff, M. S., Abdullahi, M., Abdulhamid, S. M. and Usman, M. J. 2017. Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. *Plos One,12*(5). doi:10.1371/ journal.pone.0176321

Maheswaran, M., Ali, S., Siegel, H. J., Hensgen, D. and Freund, R. F. 1999. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. *Journal of parallel and distributed computing*, *59*(2), 107-131.

Mathew, T., Sekaran, K. C. and Jose, J. (2014, September). Study and analysis of various task scheduling algorithms in the cloud computing environment. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on* (pp. 658-664). IEEE.

Pandey, S., Wu, L., Guru, S. M. and Buyya, R. (2010, April). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *Advanced information networking and applications (AINA), 2010 24th IEEE international conference on* (pp. 400-407). IEEE.

Peter Brucker, "Scheduling Algorithms", Springer, fifth edition.

Reda, N. M., Tawfik, A., Marzok, M. A., & Khamis, S. M. 2015. Sort-Mid tasks scheduling algorithm in grid computing. *Journal of Advanced Research,6*(6), 987-993. doi:10.1016/j.jare.2014.11.010

Sasikaladevi, N. 2016. Minimum Makespan Task Scheduling Algorithm in Cloud Computing. *International Journal of Grid and Distributed Computing,9*(11), 61-70. doi:10.14257/ijgdc.2016.9.11.05

Shinde, V. and Biday, S. C. 2017. Comparison of Real Time Task Scheduling Algorithms. *International Journal of Computer Applications*, *158*(6), 37-41.

Sindhu, S. 2015. Task scheduling in cloud computing. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, *4*.

Singh, R. M., Paul, S. and Kumar, A. 2014. Task Scheduling in Cloud Computing. *International Journal of Computer Scienceand Information Technologies (IJCSIT)*, *5*(6), 7940-7944.

Sousa, T., Silva, A. and Neves. A. 2004. Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*, 30(5-6):767–783, 2004.

Verma, S. Kaushal, "Deadline constraint heuristic based genetic algorithm for workflow scheduling in cloud," Forthcoming article in international journal of grid and utility computing.

Yin H., Wu H., Zhou J. 2007. "An Improved Genet ic Algorithm with Limited It erat ion for Grid Scheduling", IEEE Sixt h Internat ional Conference on Grid and Cooperat ive Comput ing, 2007. GCC 2007, Los Alamitos, CA, pp. 221-227.

****