IMPROVED APPROACH IN LOAD BALANCING FOR PARALLEL COMPUTING

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

[1] Research Scholar, IKGPTU, Kapurthala, Punjab, India, er.duggal@gmail.com
[2] Professor CSE, Punjab,India drvijaydhir@gmail.com

Abstract: *In almost every aspect of today's world, parallel computing is playing as is a very noteworthy role in meeting the rising needs of the digital world. The purpose of this paper is to show that if we combine the logic of Weighted Round Robin algorithm with Minimum Completion Time algorithm, then newly developed hybrid algorithm provides us optimal results like shorter makespan, lowest execution time, lowest energy consumption, lowest latency, highest throughput in comparison to Round Robin (RR), Opportunistic Load Balancing (OLB), Minimum Execution Time (MET) and Minimum Completion Time Algorithm. Thus, hybrid algorithm is more reliable to schedule the gives tasks or jobs to nodes of computing system as well as it can also remove the load imbalance problem to greater extent. With these optimal results, we can say that hybrid scheduling and load balancing can finish the workload within given deadline.*

Keywords: ***Keywords***- *Parallel Computing, Load Balancing, Task Scheduling, MET, MCT, OLB.*

I. INTRODUCTION

The major reason for balancing the load is to tap the potential of underutilized processing power within the parallel and distributed networks [1]. So, its better idea to shift extra load from heavily loaded nodes to under loaded nodes without disturbing the remaining clients in a network. Other motivations for scheduling and load balancing include attaining high accessibility, signifying adaptability and tapping execution power of all nodes. Advanced accessibility can be attained by shifting tasks from the machines that are more prone to failure to the more reliable machines [2]. Thus, the execution of tasks will remain unaltered even after some of the nodes goes down off the network. The accessibility is indispensible during the process reconfiguration as some tasks are needed to continue in working condition (like servers) [3][4]. At last, some machines have high processing capability in a distributed network and these machines should always be targeted for processing purpose although the processes can be started on regular machines [5][6].

So, load balancing algorithms play the most important role to enable efficient use of multi-processor systems by nearly equalizing the load among processors [7,8]. Balancing of the load among the computing nodes does not keep any node in idle state. A load balancing algorithm will always improve the performance of the computing system than having load balancing algorithm[9][10-11]. Thus, without load balancing algorithm the basic objective of a computer network for sharing resources will not be fulfilled effectively [12].

II. RELATED WORKS

Scheduling and load balancing is achieved on the knowledge of numerous performance evaluations, which will increase the performance of parallel tasks computations [14]. The author shows that the task deals with the data processing, software accessing and storage utilities. The efficient software classifies process reliable with the service-level arrangement and demanded sources [15]. Every process in the parallel processing is then allocated to the one of the available processing servers. The meta-heuristic task scheduling; delivers an outstanding amplification through which it practices the data for achieving scheduling results. Heuristic methods can work as static or dynamic [9] [10]. In addition, such cyclic methods like round robin deals with the FIFO method to perform such scheduling tasks. In addition, it works on the resource management for each task using specific time intervals [16] [17]. Subsequently the processes are queued till the probability for processing of the tasks is achieved. Also the balancing load is also one of the heuristic methodology based scheduling process which also deals with the scheduling of the processes to the following accessible apparatuses based on completion time of the execution [12][13].

Round Robin (RR) It allocates the tasks to the nodes in First-in-First-Out (FIFO) manner for a specific time slice and thereafter it snatches ir from node and allocate next task available in ready queue of that node. Thus it is pre-emptive in nature. [16].

Opportunistic Load Balancing (OLB) it allocates the given job to the workstation that is ready next. Thus, it tries to keep maximum resources as busy as possible. But main drawback of this method is that poor makespan is resulted as this strategy does not take in to account execution-time of the process [17].

Minimum Execution Time (MET) That machine, which will take the lowest execution time for the process, will be the first to be scheduled. Because it does not take into consideration the availability of the machine before assigning job to it, load balancing is not achieved in effective way. Time-Complexity of this method is O (m) if we consider m as number of processes [18].

Minimum Completion Time (MCT) That machine, which will take the lowest completion time for the process, will be the first to be scheduled. Because it takes into consideration the availability of the machine before assigning job to it, load balancing is achieved in effective way. Time-complexity of MCT is similar to MET [18].

III. PROPOSED WORK

Round Robin Algorithm does not take into account different processing capabilities of the machines which results in load imbalance. This shortcoming can be taken care of by using the Weighted Round Robin (WRR) where larger processes are assigned to higher priority queues.

MCT covers the limitations of OLB and MET by considering the ready time of the resource to maximizing the overall resource utilization.

Proposed Algorithm (WRR+MCT) combines the benefits of WRR and MCT. It is pre-emptive in nature to balance the load among processors at run time in a better way. So, its dynamic algorithm in nature.

IV. RESULTS AND DISCUSSION

In this research we have attained whole simulation in MATLAB. We have used MATLAB because it’s an efficient strong technical computing tool which is used to analyze the algorithm and execution in an effectual manner. So below are the results and discussions of the proposed method.

Thus, using MATLAB, Simulators for Round Robin (RR), Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT) and Proposed Hybrid Approach (WRR+MCT) are designed. Data Set consisting of random arrival times of processes, the power consumption by these processes and other data traffic is used as input for analyzing performance of all these simulators. Each Simulator needs input as number of processes for which scheduling is to be done so that the load is optimally balanced among various parallel processors and the appropriate simulation time needed by the simulator.

The experimental results obtained from these simulators are shown as follows:.

The proposed algorithm steps are as follows:

```
Step_1: Start.
Step_2: Set the processes so that $J_i = 1$ to n and Simulation time
            Where J is No. of processes and n stands for limit of the total processes.
Step_3: Generate random initial energies $E_x$ and the arrival time of the process $T_s$.
Step_4: Store the ID's of processes in one array.
Step_5: Deployment of the queues Q [p] such that p = 1, 2, 3, 5 and Machines M[s] such that s = 1, 2, 3, 4, 5.
Step_6: Generate the burst_time of CPU to complete the processes.
Step_7: Initialization the array C[z] where $z \leq 1 \leq n$ to complete the processes.
Step_8: Assigning weights W[x] to the queues and select the high priority queue having high weight.
Step_9: Assign jobs to the high queues and execution through machines having high bandwidth for the minimum
completion time of jobs.
Step_10: Calculate time wanted for the process, store its Id for the current execution of the process.
Step_11: For i = 1: job count

            If C(t)< T (j)
Evaluate the power consumption, completed processes and completion time for the task.
            End If

            End for

Where C (t) = current time and T (j) = time required per job.

Step_12: Evaluate the process for the incomplete jobs after the burst time.
Step_13: Evaluate the minimum completion time process to find the jobs which are executed and took minimum time to
execute.
Step_14: Evaluate the power consumption to execute the number of jobs.
Step_15: Evaluate the latency, throughput and execution time taken by machines to execute the jobs parallel.
Step_16: Stop.
```

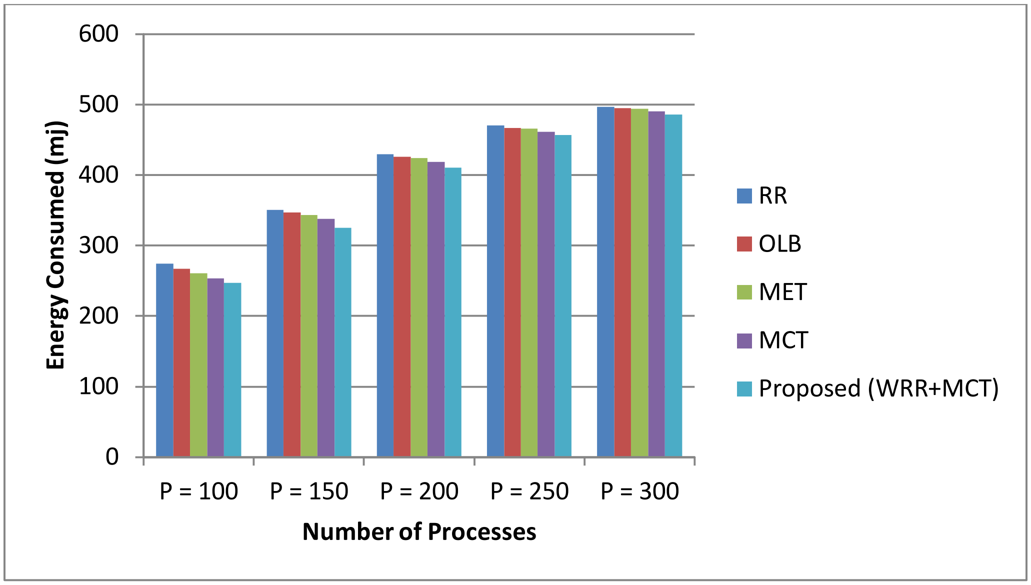

Fig 2: The energy consumption pattern of different algorithms

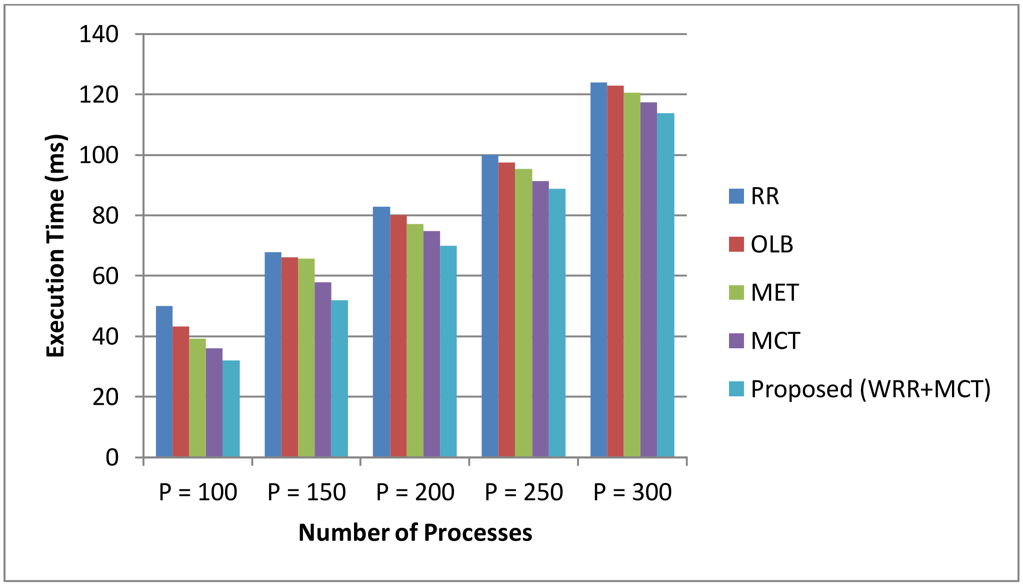

Fig 3: Shows that the actual execution time taken by proposed method is less than the expected execution time considered by the other traditional methods.

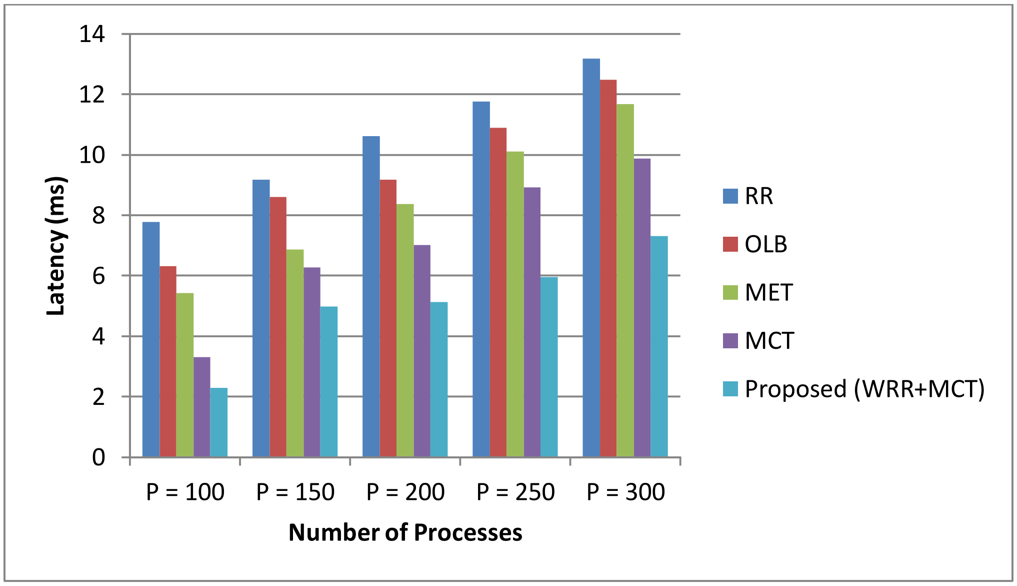

Fig 4: shows that the latency in execution of all processes is lowest in proposed method to give real time solutions.

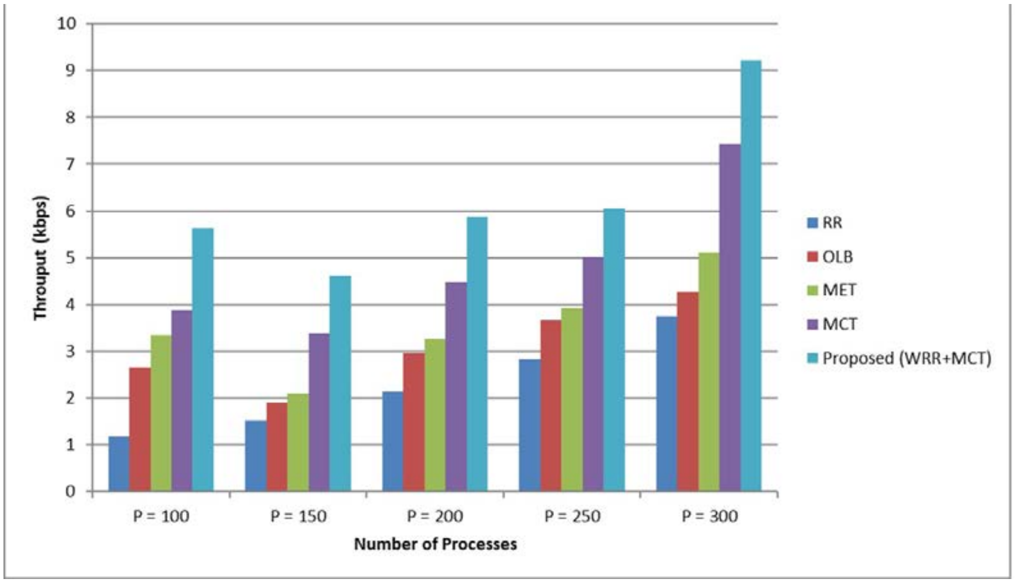

Fig 5: shows the highest throughput achieved by proposed method in terms of kilo bits per second (kbps) in evaluation to other methods.

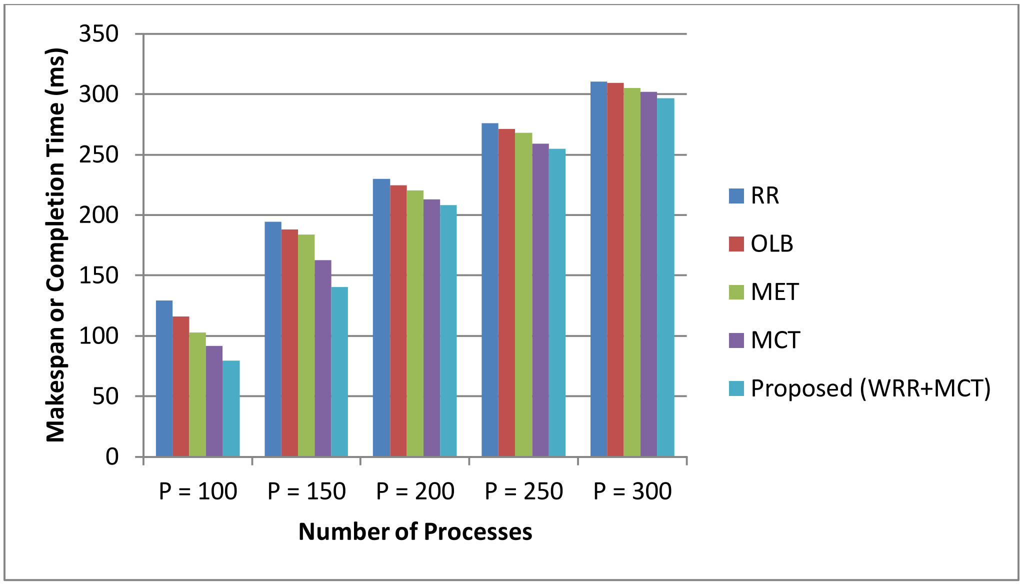

Fig 6: shows the minimum completion time taken by hybrid method in execution of all the processes in given simulation time. As we know makespan is the total time taken by an algorithm for execution of given tasks from earliest start time of first process to latest finish time of last process. The makespan for given input data set is also obtainable from this figure and clearly proposed hybrid method is attaining lowest makespan than other traditional methods.

All these above five metrics viz. Energy Consumption in mJ, Execution Time in ms, Latency in ms, Throughput in kbps and makespan or completion time in ms are quantitative in nature while reliability and deadline are taken as qualitative metrics.

Thus, it is clear that the performance of our proposed method is coming better and is achieving high efficiency in evaluating all the number of jobs by the machines to provide all the necessary resources to the users. The proposed hybrid method can attain optimal results which increase the efficiency of the parallel computing tasks in an effectual manner.

V. CONCLUSION

This paper deals with the parallel computing method using the hybridization method using weighted round robin and minimum completion time for the scheduling and the load balancing of the computing system to achieve least energy consumption, least execution time, least latency, high throughput and least makespan or completion time.

Future research work may include the communication cost in message passing from source to destination node in the communication network. As such techniques can be explored which can provide optimal results considering communication cost over links while calculating the above said performance metrics.

REFERENCES

1. V. Dhir, G. 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, 2016.
2. Saanjay Kumar Panda, “TBS: A Threshold Based Scheduling in Grid Environment”, 3rd IEEE International Advanced Computing Conferenece (IACC), pp 23-27, 2013.
3. A. Kaminsky, “BIG CPU, BIG DATA: Solving the World's Toughest Computational Problems with Parallel Computing”, Create Space Independent Publishing Platform, 2016.
4. T. Deepa, Dhanaraj Cheelu, “ A comparative study of static and dynamic load balancing algorithms in cloud computing”, International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017.
5. D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, A. Y. Zomaya. "CA-DAG: Modeling communication-aware applications for scheduling in cloud computing", Journal of Grid Computing 14, no. 1, pp: 23-39, 2016.
6. S. Singh, I. Chana. "A survey on resource scheduling in cloud computing: Issues and challenges." Journal of grid computing 14, no. 2, pp: 217-264, 2016.
7. S.J. Ahmad, C. S. Liew, E. U. Munir, T. F. Ang, S. U. Khan, "A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems", Journal of Parallel and Distributed Computing 87, pp: 80-90, 2016.
8. A. Haq, M. A. Shah, "Task Scheduling in Parallel Processing: Analysis", Journal of Biometrics & Biostatistics, 2015.
9. A. Sharma, S. Verma, "A Survey Report on Load Balancing Algorithm in Grid Computing Environment", Int. J. Adv. Engg. Res. Studies/IV/II/Jan.-March 128, pp:132 -138, 2015.
10. R. M. Singh, S. Paul, A. Kumar, "Task scheduling in cloud computing: Review." International Journal of Computer Science and Information Technologies 5.6, pp: 7940-7944, 2014.
11. S. Nagadevi, K. Satyapriya, Dr.D.Malathy3, “A Survey on Economic Cloud Schedulers for Optimized Task Scheduling”, International Journal of Advanced Engineering Technology, 2013.
12. W. Huai, Z. Qian, X. Li, G. Luo, and S. 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.
13. D. Kliazovich, S. T. Arzo, F. Granelli, P. Bouvry, S. U. 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.
14. S. H. Jang, T. Y. Kim, J. K. Kim , J. S. L. School, “The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing”, International Journal of Control and Automation Vol. 5, No. 4, December, 2012
15. S. Sudhir, “Survey on Scheduling Issues in Cloud Computing”, Procedia Engineering, Elsevier, 2012.
16. A. mantri, S. Nandi, G. Kumar, S. Kumar, High Performance Architecture and Grid Computing, International Conference HPAGC 2011, Chandigarh, India, July, Proceedings, 2011.
17. V. Dhir, "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.
18. M Hemamalini, M V Srinath, “Performance Analysis of Balanced Minimum Execution Time Grid Task Scheduling Algorithm”, Internaional Journal of Communication and Networking System, Vol 5, Issue 2, pp 96-100, 2017.
19. Shahbaz Afzal, Kavitha Ganesh, “A Taxonomic Classification of Load Balancing Metrics: A Systematic Review”, 33rd Indian Engineering Congress, Udaipur, 2018: Technical Volume, pp: 85-90, 2018.