An Efficient Hybrid Scheduling and Load Balancing Approach in Parallel Computing Systems

Raj Kumar[1], Prof. (Dr.) Vijay Dhir[2]
[1]*Research Scholar,* [2]*Professor CSE, M.K Group of Institutes,*
[1]*IKG PTU, Jalandhar, Punjab India,* [2]*Amritsar, Punjab, India*

Abstract- Parallel computing is a very significant research area which is new and emerging filed that holds contribution significantly in various areas of technology and industries. The goal is to make the aggregate computing power of a vast array of resources available to a single programmer and the application. With that kind of computing power, problems could be solved that aren't even currently attempted, and much larger versions of existing problems could become tractable. This paper deals with the efficient hybrid approach using weighted round robin and minimum completion task scheduling for the successful execution of task with high throughput and efficient completion of jobs with less energy consumption of machines and the proposed performance is evaluated with traditional techniques named as minimum completion time, round robin and opportunistic load balancing.

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

I. INTRODUCTION

Parallel computing deals with the computationswhere a lot of calculations or the performanceimplementations of the tasks or processes are passedand executed instantaneously [1].Hugedifficulties can be separated into reduced ones, which are further solved in an efficient manner at the same interval of time. There are variousdistinct parallel computing systems. They are bit by bit, instruction by instruction and comprises of job parallelism process. The process of parallelism has been employed to achieve high computing performance, but itsacquisition is inhigh interest because of its physical restrictionsthat prevents scaling of the frequency [2]. As the consumption of the power by the employed computers is a big concern in the parallel computing, these types of processes or systems have become the leadingarrangement in the architecture of the computer which takes the form of high core processors. The tasks based on these computing are closelyinterrelated to the process of the synchronized computing which can be used frequently together, though these two are totally distinct [3][4].It is conceivable to work parallel without the action of the concurrency like the process of multitasking by using time sharing arrangement with the use of single processing unit. In this arrangement, a computational process is broken in numerous phases which can be less or more according to the application requirement that can be handledindividually and whose outcomes are collective in natureonthe time of completion [5]. In comparison, in the arrangement of concurrent computing, the processes do not relate with thechores. The separate processescan be varied in nature and often deal with interrelated communication of the tasks at the time of execution [6].

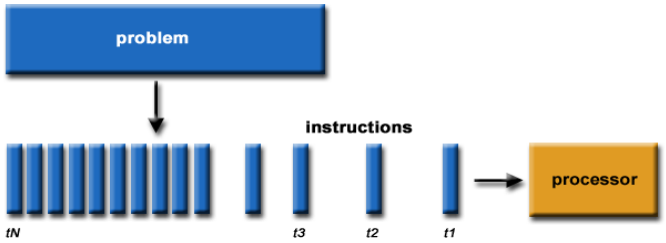

Fig.1: Parallel computing generalized process

The research based on parallel computing is discovered in the mid 90's using some efforts which arebasicallycontinuedor progressed parallel with eachother. Clustering process is the main concern of the parallel computing [20]. The clusters deal with the high performance arrangementwhich is the choice ofvarious industries with highelasticity, consistency, scalability and costing performance over different work stations. The process of clusteringbased on parallel computing arrangement, has one centralized node and more than onecluster nodes which are the processing elements. Cluster performance can beeffectual usingaddition of more efficient nodes which are involvedwith the master node [21]. Numeroustrainings have establishedwith a large scale collectionarrangements. Progressively, parallel handling is the cost-effective scheme for the efficientexplanationor solutions of complex data handling difficulties. The appearance of low-cost parallel processorslike multiprocessors based desktops and the workstations which have made parallel procedures applicable, as they have such software ethics for portable or movable parallel software design [22][23].

The effectualexplanationsand advantages ofhaving parallel computing in real time scenarios are as follows:

1. Time Saving
2. Solving complex problems
3. Execution of multiple tasks with in the same time intervals [24][25]

There also some other advantages which are responsible of achieving efficient results are as follows:

1. Advantage of management ofresources: Having the availability of such resources on WAN arrangements, the users are having advantage of management of resources over the internet [26].
2. Savings of cost: Having computing resources at lower prices, the parallel computing is able to share cheap resource with the users over the internet [27].
3. Reducing the memory problems: Solitary computers have limited resource of the memory. For complex management of the data and problems, by using memories of various computers can reduce these problems [15].
4. Transmission data rates at high speeds - Therapidityof serial systems is openly dependent on how speedily the data can transfer at higher rates. Growinghustlesand speeds increases the closeness of handlingfundamentals [16].
5. Financial Limitations: It is progressivelycostly to make a particular processor speedy. Using a superioramount of fast servicemainframes to attain the betterroutine is less luxurious [17]

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 [7]. The author shows that the task deals with the data processing, software accessing and storage utilities. The efficient software classifies processreliable with the service-level arrangement and demandedsources [8]. Every process in the parallel processing is then allocated to the one of the availableprocessing 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]. Also such cyclic methods like round robin deals with the FIFO method to perform such scheduling tasks. Also it works on the resource management for each task using specific time intervals [10][11]. 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].Also the Minimum Execution Time (MET) and Minimum Completion Time (MCT) are based on heuristic procedures in which assigned processes on the processors work efficiently and execute tasks in less execution time [14].

III. PROPOSED WORK

The proposed algorithm steps are as follows:
Step 1: Start
Step 2: Initialize Jobs such that J_i = 1 to n and Simulation time
Where J = number of tasks and n is the number of tasks limit
Step 3: Generate random initial energies E_x and the arrival time of the process T_s
Step 4: Store the ID's of the tasks in the 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, ..,5
Step 6: Generate the burst time for the completion of the job
Step 7: Initialization the array C[x] where x = 1 to N for the completion of number of process.
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: Evaluate the time needed for the job and store the id of the job 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

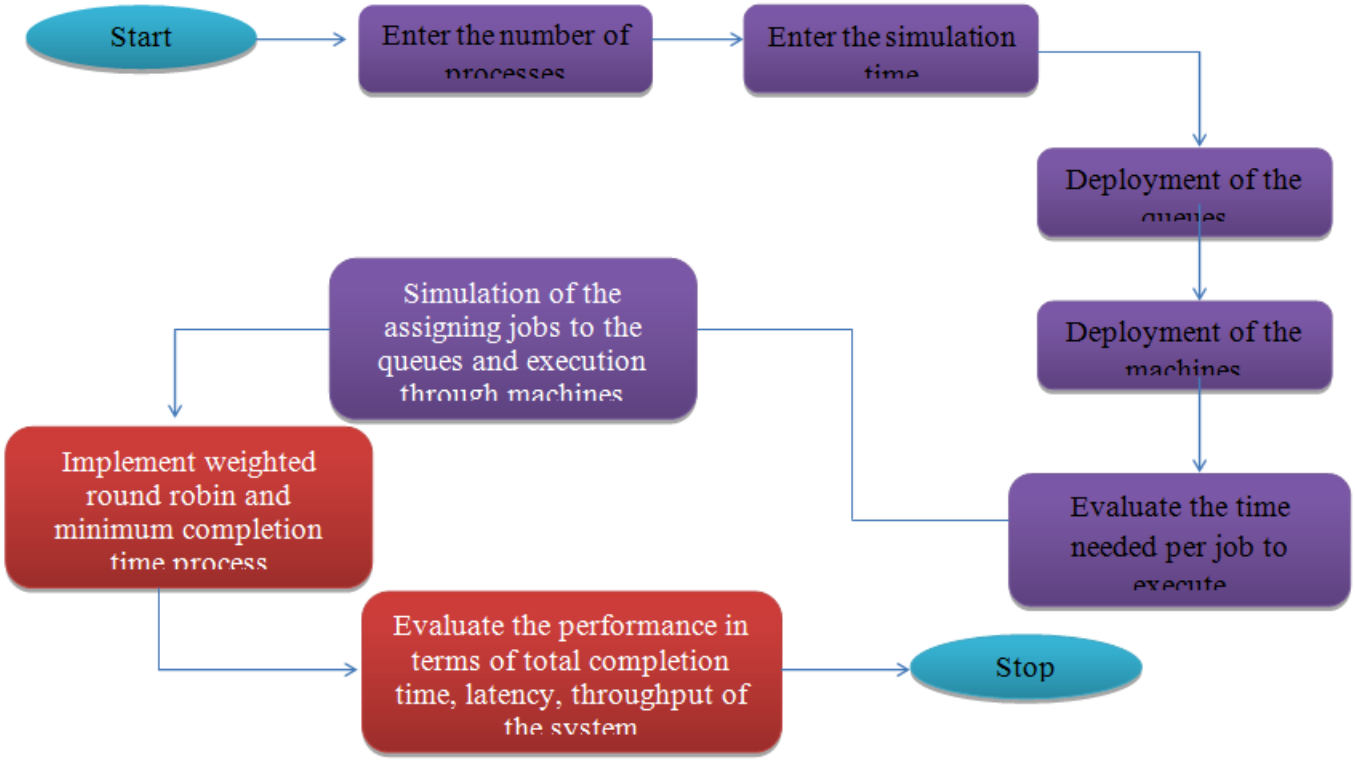

Fig.2: Proposed Flow Diagram

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 approach.

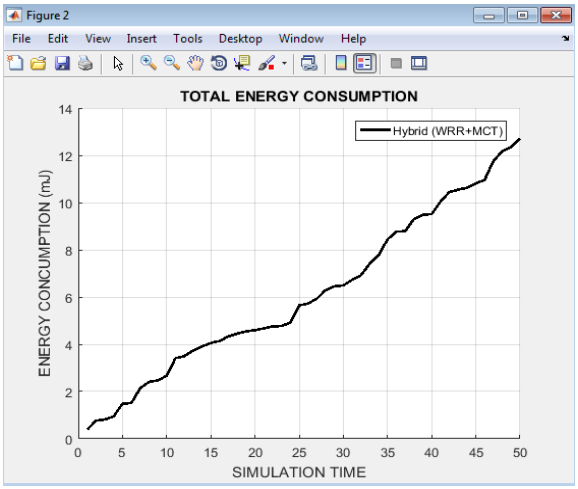

Fig.3: Energy consumption

The fig 3 shows the energy consumption of the system to execute the number of the tasks at given time simulations and shows that our proposed approach is able to achieve less energy consumption which shows the robustness of the machines to execute the number of jobs.

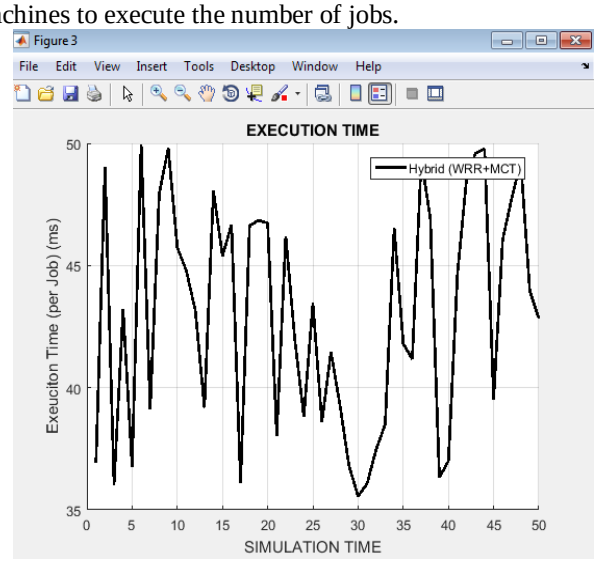

Fig.4: Execution time (ms)

The fig 4 shows the execution time of the system to complete the total number of jobs with our proposed hybrid approach in an efficient manner and shows that system is taking less execution time with high performance of the system.

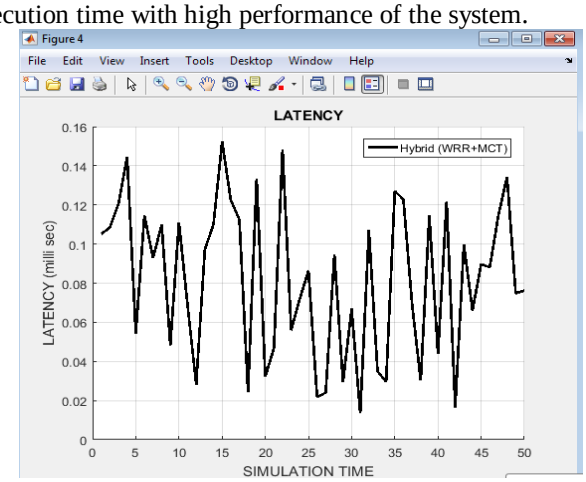

Fig.5: System Latency

The fig 5 shows the latency of the system which shows that our proposed approach is able to achieve less latency with less error rate probabilities to execute the number of tasks in the efficient manner to achieve less execution times

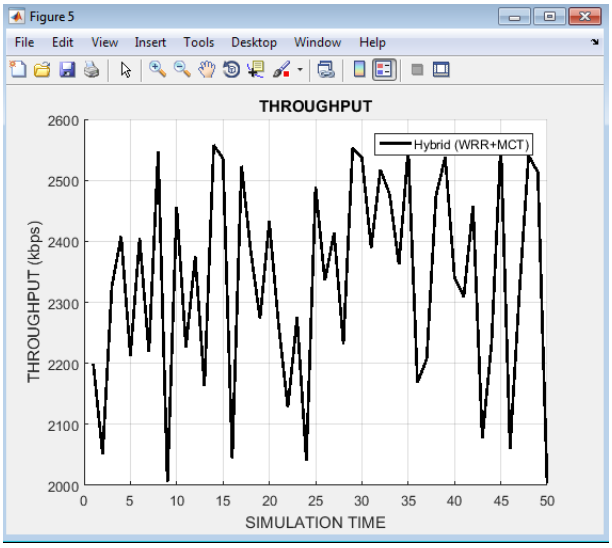

Fig.6: System Throughput

The fig 6 shows the throughput of the system that shows the execution of the jobs in successful manner. The throughput must be high and the latency must be low which reduces the execution time of our machines to achieve high efficiency.

Table 1: Performance using Round Robin

EC (mJ)	ET (ms)	LT (ms)	TH(kbps)
1400	42.84	550	2.3

Where EC = Energy Consumption in mJ
ET = Execution Time in ms
LT = Latency in ms
TH = Throughput in kbps
EC = Energy Consumption

Table 2: Performance using MCT

EC (mJ)	ET (ms)	LT (ms)	TH(kbps)
817.8	42	70	19

Table 3: Performance using OLB

EC (mJ)	ET (ms)	LT (ms)	TH(kbps)
3100	42	900	20

Table 4: Performance using Hybrid Approach

EC (mJ)	ET (ms)	LT (ms)	TH(kbps)
13	0.43	0.08	2000

From above tables we can see that the performance of our proposed approach is coming far 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 approach is able to achieve less execution time, low latency, high throughput and less energy consumption which increase the efficiency of the parallel computing tasks in an effectual manner.

V. CONCLUSION

This paper deals with the parallel computing approach using the hybridization approach using weighted round robin and minimum completion time for the scheduling and the load balancing of the computing systems to achieve less execution time and high throughput. Our proposed approach is able to achieve less latency to execute the number of the jobs assigned to the machines and also the high throughput with less execution time for the high efficiency of our computing systems. Our proposed system is able to achieve high throughput of 2550 kbps and also the latency of 0.085 (ms) and execution time of 43 ms. The machines are able to achieve 13 mJ of energy consumption which shows that our proposed approach is highly efficient to achieve parallel execution of the tasks in an efficient and effectual manner.

The future work deals with the optimizations scenarios if the execution of the machines fails to implement the jobs assigned to it due to any halt condition

VI. 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]. A. Kaminsky, "BIG CPU, BIG DATA: Solving the World's Toughest Computational Problems with Parallel Computing", Create Space Independent Publishing Platform, 2016.
[3]. 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.
[4]. 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.
[5]. 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.
[6]. A. Haq, M. A. Shah, "Task Scheduling in Parallel Processing: Analysis", Journal of Biometrics & Biostatistics, 2015.
[7]. 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.
[8]. 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.
[9]. S. Nagadevi, K. Satyapriya, Dr.D.Malathy3, "A Survey on Economic Cloud Schedulers for Optimized Task Scheduling", International Journal of Advanced Engineering Technology, 2013.
[10]. 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.
[11]. 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.
[12]. 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
[13]. S. Sudhir, "Survey on Scheduling Issues in Cloud Computing", Procedia Engineering, Elsevier, 2012.
[14]. A. mantri, S. Nandi, G. Kumar, S. Kumar, High Performance Architecture and Grid Computing Computing, International Conference HPAGC 2011, Chandigarh, India, July, Proceedings, 2011.
[15]. 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.
[16]. A. S. Saira, G. Barua, "Distributed Route Control Schemes to Load Balance Incoming Traffic in Multi-homed Stub Networks", IEEE ,Dept. of Computer Science and Engineering Indian Institute of Technology Patna & Guwahati, 2010.
[17]. G. V. Iordache, M. S. Boboila, F. Pop, C. Stratan,V. Cristea, "A Decentralized Strategy for Genetic Scheduling in Heterogeneous Environments", Springer-Verlag Berlin Heidelberg, 2010.
[18]. V. Dhir, "Alchemi .NET Framework in Grid Computing", Proceedings of the 3rd National Conference; INDIACom-2009 Computing For Nation Development at BharatiVidyapeeth's Institute of Computer Applications and Management, New Delhi. 2009.
[19]. M. Petrova, N. Olano and P. Mahonen, "Balls and Bins Distributed Load Balancing Algorithm for Channel Allocation", IEEE, RWTH Aachen University Aachen, Germany, 2010.
[20]. J. Balasangameshwara, N. Raju,"A Decentralized Recent Neighbour Load Balancing Algorithm for Computational Grid", International Journal of ACM Jordan (ISSN 2078-7952), Vol. 1, No. 3, September, 2010.
[21]. P. Jain, D. 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.
[22]. M. Randles , A. Rahmeh , P. Johnson, A. T. Bendiab, "Biased random walks on resource network graphs for load balancing", Springer, 2009.
[23]. S. S. Waraich, "Classification of Dynamic Load Balancing Strategies in a Network of Workstations", Fifth International Conference on Information Technology: New Generations, 2008. 11
[24]. 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
[25]. S. A. Jarvis, D. P. Spooner, H. N. L. Choi Keung, G. R. Nudd, "Performance Prediction and its use in Parallel and Distributed Computing Systems", in Proceedings of the International Parallel and Distributed Processing Symposium, 2003.
[26]. I.C.Legrand, H.B. Newman, "Monalisa: An Agent Based, Dynamic Service System to Monitor, Control and Optimize Grid Based Applications", European Center for Nuclear Research – CERN, 2003.
[27]. M. Coli, P. Palazzari, "Real Time Pipelined System Design through Simulated Annealing," Journal of Systems Architecture, vol.42, no. 6-7, pp. 465-475, 1996.