

Load Balancing in Cluster System using Entropy Value

Pratibha S. Yalagi
Walchand Institute of
Technology
Solapur

Mitali M. Mohire
Walchand Institute of
Technology
Solapur

Sulabha S. Apte
Walchand Institute of
Technology
Solapur

ABSTRACT

In a clustered system, the general problems in load balancing are: ill-planned task allocation, poor performance, long response time, and low throughput. The work focuses on a novel algorithm of load balancing, which is based on the entropy value for both wired and wireless connections. To improve the performance of a system the scheduling and migration policy are calculated on entropy value. The factors like the configuration of server and client, benchmark performance, time of communication between client and server systems and the system speed are considered. This novel algorithm is compared with traditional algorithm, like Round robin and the results shows that the time of execution reduces and the system performance is increased.

Keywords

Load balancing, entropy value calculation, clustered system.

1. INTRODUCTION

Load balancing allows use of multiple paths to balance the traffic across the network. It has advantages in ease of capacity planning; also it can help to minimize traffic between systems by distributing them across multiple nodes and to handle node failure by offering alternative paths. Across network traffic the powerful tool is load balancing.

Today, the Internet is very important for each person who is belonging to Computer industry and it is developing rapidly. Whereas the new applications, the Exponential increment of data processing, the more tasks of web services, the data flow and the strength of the network which is calculated, especially the core one has increased significantly. In parallel and distributed systems, the Load balancing is a critical issue to ensure faster processing and better utilization. The work focuses on improving load balancing in the clustering system using the concept of 'Entropy'. Entropy is just like a state function, the changes in the value of entropy are same for any process which goes from initial state to final state. Entropy helps to distribute task between the clustered systems. Basically the concept of entropy is in physics, it is used for evenness of energy distributions in the space. In this paper entropy is consisting of the corresponding load factor, which is the proportion of the load of nodes to the full load of the system nodes. Where the load can be calculated by the number of the task and measured by the calculation of tasks.

For balancing the load, single server would not able to satisfy the users growing demands, so the cluster systems becomes the primary choice with systems having good scalability as well as high performance. Load balancing is technique which allows equal distribution of workload between different server system nodes or hard drive. Redistribute the work load between the nodes of the systems is one of technique of load balancing to improve the usage of resources and reply time.

A cluster is the set of connected computer that are working together and has different configuration of hardware and

software's. It is a group of independent servers running on different configuration and considered a single system. Sometimes it is difficult to understand and study how to hire a task reasonably. So balancing the load between the cluster systems became a main goal for allocating resources. Hence, the proposed novel algorithm for balancing the load depends on the change in value of entropy on the cluster system.

The objective of the work is:

- To reduce the reply time
- To measure processor speed
- To maximize the throughput.
- To improve the system performance.

2. BACKGROUND STUDY

There are various algorithms for balancing the load. For example Round Robin algorithm is used in Web Logic Server for clustered object. Random Load balancing is only recommended for Homogeneous Cluster.

1) Depth First Search Algorithm:DFS i.e. Depth first search algorithm is used to evaluate in the parallel system for dynamic load balancing algorithm. In this, Linux Operating system and Solaris Operating System with formation of 10 numbers of clusters are used and parallel execution is done in the clusters of heterogeneous 6 SUN workstations using the message passing interface [2].

2) Genetic Algorithm: Load balancing optimization algorithm based on genetic algorithm, which is only used in the laboratory, but practically it can't be able to perform or to solve the complex needs [3]. The genetic algorithm is used for balancing the load with five components in distributed system. The components are genotype, chromosomes, crossover and mutation [13]. All jobs have set same priority and Natural Selection strategy is used with Genetic Algorithm and comparing with round robin algorithm [14].

3) Dimension Exchange Algorithm: Some organization having computing clusters which is placed on different places. Balancing the load between computing cluster, where one cluster contains high load and other cluster may have no load or little load on the system. The improved DEM for load balancing is quite suitable for computing clusters that are quite far apart, the problem of having computing clusters of various sizes [4]. In Heterogeneous Computing clusters, the Dimension Exchange Method is used for balancing the load between the hypercube architecture. In this Master terminal is considered to be connected to hypercube. Same size of computing cluster is used for simple performance. Theoretically, it can't provide a better result for balancing the load, but practically it provides optimal solution [8].

4) Entropy Based Algorithm for Homogeneous System: In Homogeneous clustered system, there are same types of configuration like hardware and software are used to balance the load. The system is totally depends on the calculation of entropy value. In this threshold value is also calculated for comparison with entropy value. The scheduling and migration strategies are performed based on entropy value. And the traditional algorithm is used for comparison with an entropy based algorithm. But the system is only for the homogenous cluster system not for the heterogeneous clustered system [1].

5) Round Robin VM Load balancing Algorithm: VM load balancing, the previous allocation state which is requested from user are not saved. This problem is resolved by using round robin approach. For this two data structure are used, hash map and VM list to store a state. Updating data structure is required for each user entry [15]. In concurrent and distributed computer system, the scalable algorithm is used for balancing the load and mapping. There is absence of central thread to control and there is no need of centralized communication. Distributed and concurrent computer system is derived with the help of graphs of spectral properties: first graph shows the communication between the process in problem of mapping and second for network link between computers as well as if the number of the computer is increasing then the cost should not to be increased for balancing the load. Lastly, it checks the result comparing with other result [6].

For the cost-effectiveness and for improving performance of parallel computing some methods are used. One of the method is Message Passing Interface [MPI] is used in a cluster system for cost-effectiveness. There is a rapid development in microprocessor availability and designing which is now performing highly on cluster systems. For this the cluster system for availability of resources and shelf hardware & software are used [7]. To balance the load among computing nodes a novel algorithm was presented in high performance computing [HPC].

3. SYSTEM DESIGN

The Round Robin and Random Algorithm sends the task randomly without checking the capacity or performance of slave machines. Hence the system becomes unstable so it is required to check whether selected system is able to execute task or not. Using Entropy, this problem is solved.

The proposed Algorithm is for Cluster system, which has four main parts load information, benchmark, scheduling, and migration. Clustered system consists of different configuration of nodes. The system uses novel algorithm for balancing the load where the algorithm is based on the entropy value.

Benefits of Entropy: When executing a task on different machine, it is required to know whether the selected machine is capable or able to perform that task. This type of situation is easily handled using Entropy. It is not only used to know the overall System performance but also Load on the server systems.

The framework is shown in figure1.

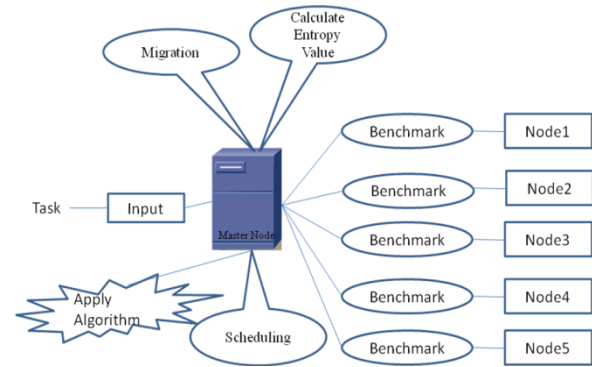


Figure 1 System Framework

In Figure1, Task is taken as input to the system and one task is considered as number of subtask. Task is java programs. Each node having different configuration and benchmark is performing on each node separately. The Master node is connected to all nodes and forms a cluster system with low load for execution of the subtask. Entropy is calculated by Master node for performing scheduling and Migration policy. The algorithm is based on the entropy value and each node sent the result back to the master node.

4. IMPLEMENTATION

For balancing the load between cluster systems, the operations are divided into two steps: first is load scheduling and second is migration. To perform load scheduling, all tasks are equally distributed to the slave nodes in the cluster to balance the system load. And migration migrate the load from overloaded node to lightly loaded nodes. Here task is considered java program and result is, comparing a time required to execution of task using both algorithms. Procedure follows following steps:

- 1) Entropy
- 2) Collect Information of load and process the load
- 3) Benchmark
- 4) Choose Scheduling policy
- 5) Migration strategy
- 6) Novel Algorithm
- 7) Algorithm Steps

4.1 Calculation of Entropy

For synchronization, the information is collected from each node as a status, this may cause traffic congestion and maximize the load between the systems by using the traditionally dynamic modeling. In this paper Master node is introduced for handling the load information. For calculating entropy value of the system, first it needs to find load of each slave node.

If the system consists of n numbers of nodes and t as time, relative load factor (q_k) is

$$q_k = L_p / \sum_i L_i (i=1,2,\dots,n).$$

Where,

L_p :- load of P node

L_i :- load on the server nodes.

n:- no. of nodes

Example:

Consider Total Entropy= 0, If suppose there are two system master (M) and server machine (S).

System has Entropy value $M = 32.974565$ and $S = 67.534314$ now divide the master node entropy value by 100,

$(M \text{ by } 100) = 0.32974565$.

Now find the relative load factor $q_k = M / (\text{sum of all server system entropy})$.

Now finally find total entropy

Total Entropy = Total Entropy+ $q_k * \text{Math.log}(1 / q_k)$;

So the total Entropy = $0 + 0.003280761 * \text{Math.log}(1/0.003280761)$.

So Final Entropy value i.e. Total entropy= 0.00814825 i.e. 0.008 .

Mathematically the Entropy ($H_{(t)}$) value at a time t is calculated as,

$$H_{(t)} = \sum_{p=1}^n [q_k * \ln(1/q_k)]$$

Where,

n :- number of nodes,

q_k :- is relative load factor.

Increase value of Entropy shows the system is balanced by using the novel algorithm within a short period of time.

4.2 Collect Information of load and process the load

Information about system load is collected for back-end services. Each slave node needs to send the load information to the master node. For this, the proposed system uses simple moving average method. Simple moving average method, calculates the average load of the system. Using this load information, Master node calculates the average of the total load. After that it allots time slice to each slave node for executing the task. For calculating the value of entropy, there is need to gather load information from each server node.

Example of simple moving average method is shown with the help of figure 2. The graph shows the CPU usage and Memory Usage of the system. For further processing, the average of both CPU and memory value is taken.

If Server Node S has Memory Load M_L and CPU Load C_L , then server node S sends load info to the Master node A . The task may be shared on the different multiple server node. The simple moving average method performs following steps.

- 1) Take M_L & C_L load for finding average load and mention some window size. The all load values is taken into buffer and dividing that using the window size and use this value as average value.

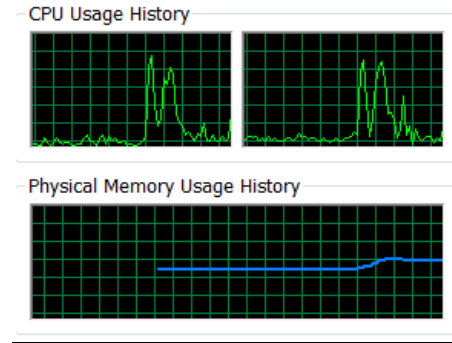


Figure 2 CPU and Memory Usage

- 2) The next step is mathematical model defined as follows: For load balancing between nodes, consider Set of load $L = \{L_1, L_2, \dots, L_n\}$, Set of Server node = $\{S_1, S_2, \dots, S_m\}$, Set of Current Server Load = $\{SL_1, SL_2, \dots, SL_m\}$, were set of load L can be mapped to the set of Server node S for finding a function $f(L)$.

Now, to execute task, it requires time (t). If execution of task L_o requires time t_o on the server node S_i then the time needed for execution of all the tasks on server node S_i is:

$$t_i = \sum_{o \in f(L_i)(i=0,1,\dots,n)} t_o$$

Where L_i = set of load

t_o = time

If the number of server nodes (m) = 1, then it indicates that there is only one server node present and execution of all tasks should be done serially and required time is the sum of all the time which is shown by t_1 .

If m is more than 1 then it that shows there are more than one server nodes and tasks can be shared with multiple nodes and required time is represented as t_m .

The aim is to execute tasks within a minimum time.

4.3 Benchmark

The way of judging level of the load like CPU Load and Memory load of processor capacity is called Benchmark. Benchmark help to the master node for distributing the task over the network. Using the benchmark result, it can also help to find the system status like under loaded, overloaded. It performs on the each system for knowing the processor capacity. Following example shows the benchmark. Considering the example:

The system start time $S = 1449250667642$ and end time $E = 1449250681017$ then the benchmark $B = 0.7476635514$ of that system.

$$\text{Benchmark} = 10000.0 / (E - S)$$

4.4 Choose the Scheduling Policy

The novel algorithm depends on the change in entropy value which is calculated by master node, so the scheduling is also depends on the value of entropy. The Scheduler collects the relative load factor of each node and entropy value from the master node. Then calculate the value of entropy changing according to the scheduled tasks. Before doing the scheduling it assumes that the task has been scheduled to the server nodes and then after it will select a node. Changes in entropy value are arriving after every scheduling. The load balancing of the system is achieved when the system will achieve the stable state.

Consider the system having the fifty tasks and Consider the five nodes ABCDE having result after scheduling 11, 11, 16, 15, and 10 respectively by using Round Robin Scheduling Algorithm. And having 10, 13, 12, 15, 13 respectively by using novel algorithm. Using a novel algorithm the system entropy value is 1.6010 and using Round Robin Scheduling Algorithm system entropy value is 1.5922 [3]

Graphically representation of above example i.e. loads after scheduling policy shown in figure 3. The diamond lines shows the load by using round robin algorithm and dotted lines show the load by using novel algorithm.

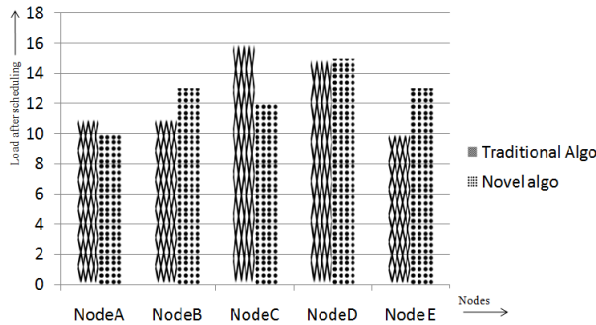


Figure 3. Graphically representation

4.5 Migration strategy

The time slot needed to execute each task is different because of their differences in the performance. Some of the tasks are done but some time some of the tasks are still running, which affects on the relative load factor of the server node so it leads to change in entropy value. At that movement, the system becomes imbalanced.

From Figure 3, the value of entropy is reduced and then the system will become unbalanced at that time, so migration is needed to perform the migration. Migration is based on entropy value, if the system is unbalanced then performing migration means transferring the tasks from high-load nodes to low-load nodes, to balance the load on each server node as well as to increase the entropy value. System resources and time is needed for performing migration.

If in case, one node hasn't returned the result within the time period, then it will terminate that task and that work is transferred to the idle or under load node in the cluster system.

4.5.1 Implementation of Migration policy:

The migration depends on the system entropy value; the decrease in the entropy value will increase the percentage of migration. The load-balance position is upgraded significantly after the migration. The scheduler decides whether to perform migration or not. To ensure that, after the migration the entropy value will be increased if not, the migration will not be performed.

Figure 4 shows the sequence of operations and flow of the system..

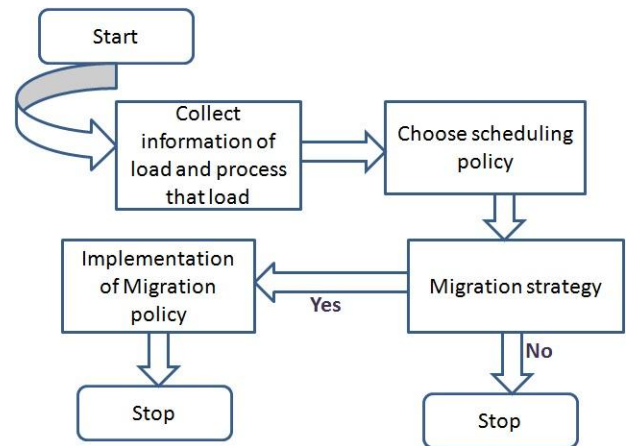


Figure 4 Sequence of operations

4.6 Novel algorithm

Whenever the task is arrived, the entropy value is calculated by using the collected information of the system. Using calculated entropy value, select a minimum entropy value; if the entropy value is less then it performs migration, and if not then shows the system is balanced. Figure 4 shows sequence of operations and the following algorithmic steps describes these operations.

Algorithm Steps: The below steps shows execution of system

Step1: The task is taken as an input to the system and execution of tasks is done in a minimum period of time for balancing load between the terminals of a clustered system.

Step2: Consider there are 10 subtasks which are considered as one big task. Calculation of the entropy value is done by the Master node and depending on that value, the task is distributed between the server nodes. Simultaneously the benchmark is also running on each server node and providing information of each system to the master node for further processing, which will help for distributing the task.

Step3: Status of the node will be calculated by the master node. The nodes can be under loaded, overloaded or doing some little work or idle. The System also provide time slot to each system to execute task within a time slot. Calculation of the entropy value is done by the system, depending on that value the migration and scheduling policy will perform. For this policy the novel algorithm is applied, which depends on the value of entropy.

Step4: If any one of the server node is not returning a result within a time period, then perform the migration policy which is done or not is decided by the master node of the clustered system.

Step5: After this the result is collected by the Master node.

Step6: The system achieves the load balancing by executing the task within minimum time and increasing the performance.

5. EXPERIMENTAL RESULTS AND ANALYSIS

The Entropy based cluster system for balancing the load which can speed up the process and execute the result within a minimum time. Novel algorithm depends on entropy, which provides very good results as compared to the traditional algorithm. The result of both algorithm i.e. the result of

traditional algorithms, load after migration and scheduling is compared with the result of novel algorithm, load after migration and scheduling. It can also perform better for a large number of cluster systems.

The figure 5 represents the overall system design with the phases of information collection, migration, scheduling etc.

Initially the traditional algorithm (Round Robin) is applied and results are collected. Then the novel algorithm is tested and the time required for both algorithms is taken for comparison. The results collected are better for novel algorithm than the traditional Round Robin algorithm.

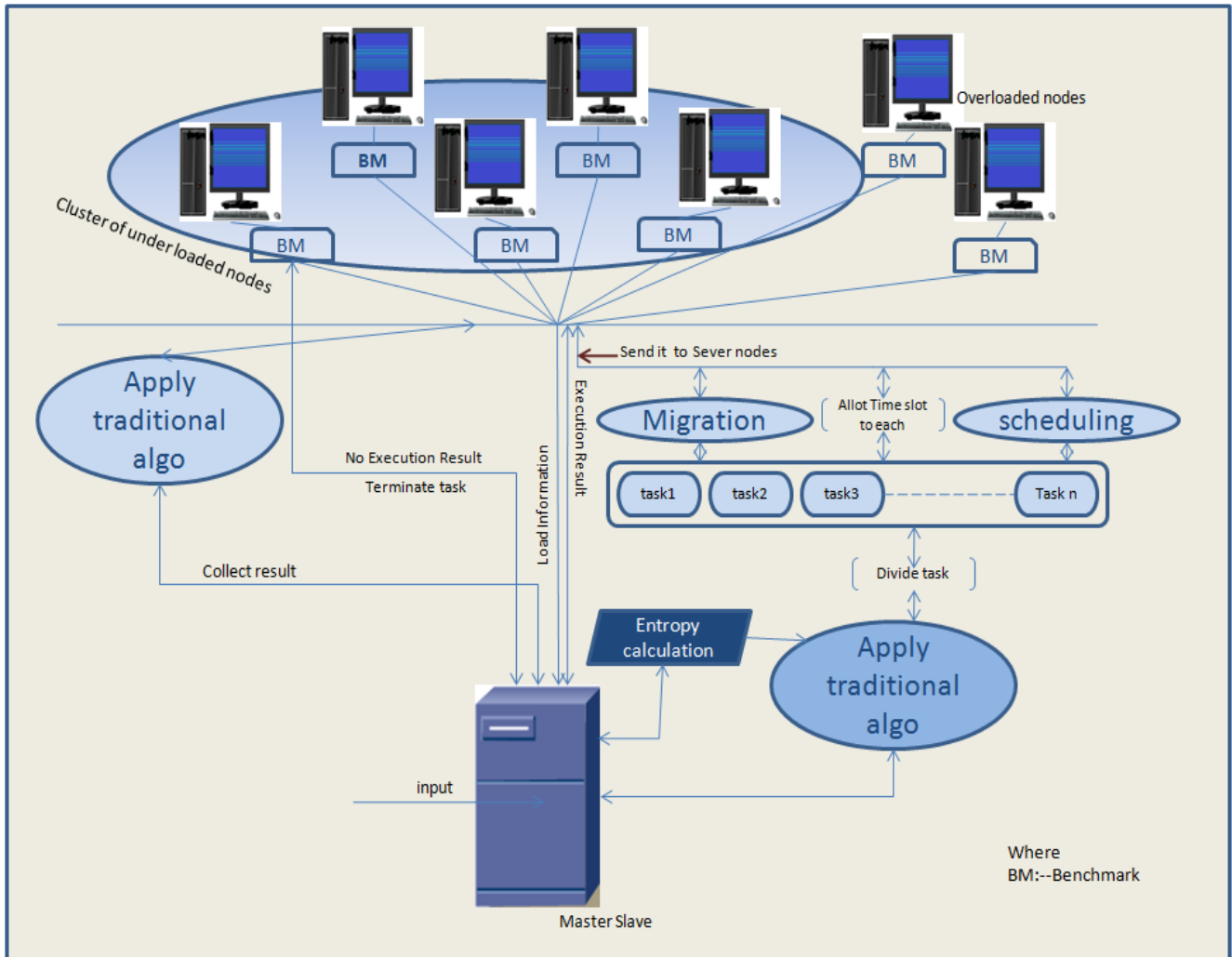


Figure 5 Overall System Design

The Graph in figure 6 shows that the big tasks require more time to execute in traditional algorithm as compared to novel algorithm.

The time (in ns) required to execute task is described in Table 1 using novel algorithm and Round Robin algorithm. The task is taken as java program of expression evaluation and simple NP hard problem.

Table 1 Table shows results

Using Novel Algorithm	Using Traditional Algorithm
926.85	1754.22
8878.36	12114.36
14234.67	21126.95
1688.52	25611.96

25217.63	17503.44
3102.91	10004.05
11294.95	23210.72
3649.06	26003.43
293.95	3753.4

Another Result is shown in figure 7 which contains execution of small tasks. The time (in ns) required to execute task is described in table 2 using novel algorithm and Round Robin algorithm.

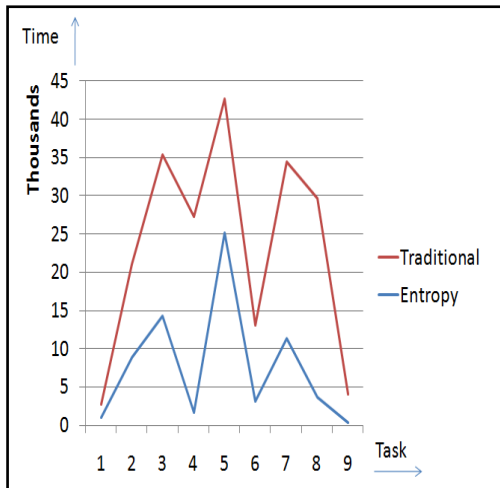


Figure 6 Analyzed result of both algorithm

Table 2 Table shows results

Using Algorithm	Novel	Using Algorithm	Traditional
	18.29		32.97
	111.98		222.71
	90.18		5806.77
	136.82		235.36
	677.76		3672.05
	927.22		1207.4
	653.31		8508.48
	625.55		7123.85
	4982.8		6260.16

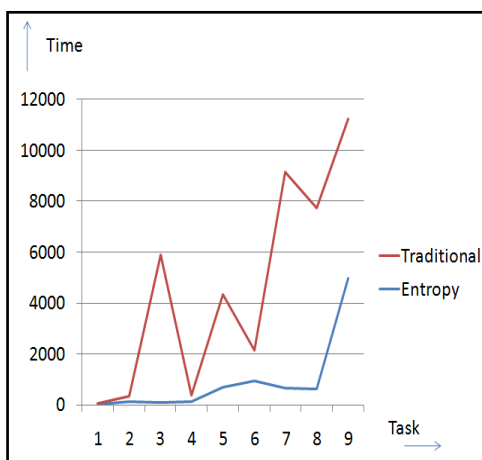


Figure 7 Analyzed result of small task using both algorithm

6. CONCLUSION

In this paper, a proposed novel algorithm based on the changes of entropy value in cluster system for balancing load, gives the basic properties of a system, using the entropy, it can measure the status of load balancing. The scheduling and migration are performed on the basis of change in entropy value. The main goal is to achieve load balancing faster and better.

The novel algorithm provides better result than the traditional algorithm for throughput, response time, increasing performance and degree of the system load balancing, which means that the novel algorithm is useful to a certain term to balance the load in cluster systems. Benchmark also gives the processor speed, capability of the processors. And it gives the information related to processor which tells whether it is able to perform next task or not. The further implementation can be done in Rack Aware System to balance the load.

7. REFERENCES

- [1] Mitali Mohire and Pratibha Yalagi "A Novel Algorithm based on the change in entropy value for load balancing in cluster system", IJSR volume 7 issue 4(2015).
- [2] A. M. I. Mohammed and X. D. Lu "Performance of dynamic load balancing algorithm on cluster of workstations and PCs", in year 2002.
- [3] "A Load Balancing Algorithm based on the Variation Trend of Entropy in Homogeneous Cluster", International Journal of Grid and Distributed Computing, pp.11-20, in year(2014)
- [4] Z. H. Liu and X. L. Wang, "Load balancing algorithm with genetic algorithm in virtual machines of cloud computing", Journal of Fuzhou University(Natural Science Edition), vol. 4, no. 40, (2012).
- [5] S. C. Chau and W. C. F. Ada, "Load balancing between computing clusters", Proceedings of the 4th International Conference on Parallel and Distributed Computing, Beijing, China, (2003)
- [6] H. Y. Sun, W. X. Xie and X. Yang, "A Load balancing algorithm based on Parallel computing entropy in HPC", Journal of Shenzhen University Science and Engineering, vol. 1, no. 24, (2007).
- [7] A. Heirich, "Analysis of Scalable Algorithms for Dynamic Load Balancing and Mapping with Application to Photo-realistic Rendering", Caltech, Pasadena, (1998).
- [8] H. Zhou and S. X. Luo, "The Computer Cluster and the Parallel Computer Environment based on the Networks", Computing Techniques for Geophysical and Geochemical Exploration, vol. 4, no. 23, (2001).
- [9] Siu-Cheung Chau, Ada wie-chee fu, "Load Balancing between Heterogeneous Computing Clusters" in National Sciences and Engineering Research Council of Canada. [2004].
- [10] Vatsal Shah , Kanu Patel, "Load Balancing algorithm by Process Migration in Distributed Operating System", in year 2012.
- [11] Jiani Guo, Laxmi Narayan Bhuyan, " Load Balancing in a Cluster-Based Web Server for Multimedia Applications", in year 2006.

- [12] J. Uma, V. Ramasamy, A. Kaleeswaran “Load Balancing Algorithms in Cloud Computing Environment -A Methodical Comparison “<http://ijarct.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-2-272-275.pdf>
- [13] Purnima Shah, S. M. Shah ”Load Balancing in Distributed System using Genetic Algorithm”, IJCA
- [14] Kousik Dasgupta, Brototi Mandal “A Genetic Algorithm based Load balancing strategy for cloud computing”, Science Direct.
- [15] Nusrat Pasha, Dr. Amit Agarwa “Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment”, IJARCSSE volume4, May 2014.