Balancing the load between computing cluster, where one cluster containes high load and other cluster may have no load or little load on the system. A simple load balancing method is used for balancing a load between computing clusters that are far away from each other which is proposed by C. Chauu and W. C. F. Ada [4].

To balance the load among computing nodes a novel algorithm was presented in high performance computing [HPC]. The association between the entropy and the program executing time was evaluated. This shows that the algorithm is effective because of the PCE and execution time is highly associated which is proposed by H. Y. Sun, W. X. Xie and X. Yang [5].

Scalable algorithms are analyzed for balancing the load and mapping to help the concurrent and distributed computer system. 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 which is proposed by A. Heirich [6].

For the cost-effectiveness and an increase in performance the parallel computing and its methods are used. There is a rapid development in microprocessor availability and designing which is now performing highly on cluster systems. In this, Message Passing Interface [MPI] is used in a cluster system for cost-effectiveness. They use the cluster system for availability of resources and shelf hardware & software by H. Zhou and S. X. Luo [7].

Balancing the load between the hypercube architecture the improved method of Dimension Exchange Method [DEM] is used in Heterogeneous Computing Clusters. In this the master node is assumed to be connected as hypercube. For simple performance, it uses or assume the same size of a computing cluster. In practice, it can provide a closed optimal solution, but theoretically do not provide quality for balancing the load which is proposed by Siu-Cheung Chau, Ada wie-Chee fu [8].

3. Proposed System

3.1 System Architecture

There are some cluster systems which are having a set of different configuration. The figure2 below shows the overview in which task is taken as input and Master node will handle the scheduling and migration and it also handle Novel algorithm and the calculation of entropy value. Each server node performs Benchmark and forward the result to master node. And them the Master node will collect the final result.

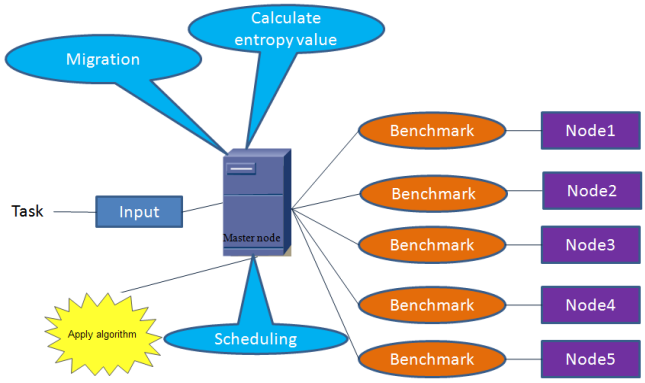

Figure 2: Conceptual overview of proposed system

Benchmark:
The benchmark is a way to judge the level of Memory load and CPU load of processor capacity. It is performed when the system will start for checking processor capacity. It sends Memory load and CPU load of each system to master node continuously, which helps in distributing task and calculating the entropy value . It is also useful to find the under-loaded system or overloaded system. If there is no load on any system, then also performs benchmark for checking processor capacity.

Consider an example: If server node S_1 have memory Load M_L and CPU Load C_L then it sends it to the Master node A. The task can be shared on different multiple server node. The mathematical model defined as follows:

For balancing the load between nodes, consider Set of load $L= \{L_1, L_2,\ldots.., L_n\}$, Set of Server node= $\{S_1, S_2,\ldots.., S_m\}$, Set of Current Server Load= $\{SL_1, SL_2,\ldots.., SL_m\}$, were set of load L can be mapped to the set of Server node S for finding a function f (L).

Now, to execute tasks, 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…

$$t_i=\sum_o \in_{f(Li)(i=0,1,..n)to}$$

If m=1, where m is the number of server nodes, 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.

What is Entropy?

It is relevant to the relative load factor, which is the ratio of load of the nodes with the full load of the system. And the load can neither be calculated by the number of tasks, nor can be measured with the calculation of tasks. If the system has n numbers of nodes and time t, the load of P node is L_p and a relative load factor is

$$q_k = L_p/\sum_i L_{i(i=1,2,\ldots n).}$$

Mathematicaly the Entropy value is calculated as,

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

The Novel algorithm can make the system load to balanced in a short period towards the trend of entropy increase. Entropy may used for showing the randomness of material, where the increase of entropy value shows the aim of balancing the load that is the average distribution of load. So the balancing of load is achieved by increasing the value of entropy. It also makes full use of server resources and avoids unequal distribution of the load. If the entropy value become maximum, the execution of a task can be done in minimum time.

3.2 Implementation Details

The traditional algorithms are Round Robin, Least-connection and Weighted Round Robin scheduling. In Round Roubin, the process is kept in circular queue or it is also called as ready queue. The new process is added to the tail of the queue. It is similar to First Come First Serve (FCFS) scheduling algorithm. A Time slice is set for each process to execute.

In weighted Round Robin Scheduling, the node can treated as different processing capacities where weight is assigned to each node. By default weight is 1. Consider one example:- four node, A,B,C and D have weights 4,3,2,1 respectively. A Scheduling sequence will be ABCDABCABA in a period of scheduling.

In least Connection scheduling Algorithm,the request is directly received from the network to the node with the least number of established connections. For this it needs to count live connection for each node. To achieve the target of the load balance in cluster system, the following operations are performed:

3.2.1Collect Information of load and process the load

Collect the node load information to back-end services. For this, the proposed system uses Simple Moving Average method. Using collected information about load, calculate the average of the load and then allots time to each system for completing the task. The graph represents time and CPU or Memory space. To calculate the entropy value for further process it needs to collect information from each server, i.e. the load of each server node.

3.2.2Choose the Scheduling Policy:

The traditional algorithms are Round Robin , First Come First Serve, Least connection scheduling, weighted round robin scheduling policy. The Novel algorithm is based on the variation trend of entropy, so the scheduling is based on the changes in the value of the entropy. The scheduler collects relative load factor from each node and the entropy value of the system from the monitor node, then it calculates the changes in the value of the entropy according to the calculation of the scheduled tasks, it assumes that before doing the scheduling, the tasks have been assigned for scheduling to the nodes and after that it select a node. After every scheduling, the entropy value of the system is increased maximum. The system will achieve a stable state at the moment, so the load balance of the system is achieved.

3.2.3Migration Strategy

The performance of each task is different, so the time needed for each task is different. Sometime some of the tasks are completed, but few tasks are still performing, so it leads to the change in load factor of the node that is changing with entropy value. From above, value of entropy is reduced and then the system will become unbalanced at that time, so there is a need 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 of each server node as well as to increase the entropy value.

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 from that cluster system.

3.2.4Implementation of Migration Policy

The migration depends on the system entropy value, if it is increased, 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.

The below diagram shows the sequence of operations for achieving dynamic load balance.

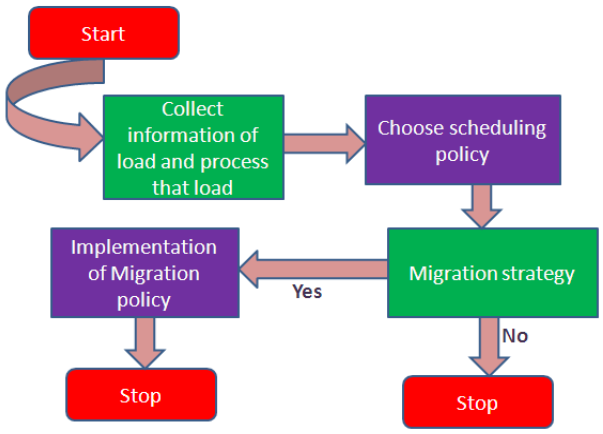

Figure 3: Sequence of Performance

The below steps shows the execution of the system:-

1) Initially the task taken as input for further processing to find the output in minimum time with balancing the load between the system node.
2) The Task is divided into subtask depending on the size of entropy and simultaneously the benchmark is performed continuously for each system and it gives load information on each different configuration of systems to the master node, which helps in distributing task.
3) The master system will find the status of the system nodes, i.e. under load, overloaded and idle system node and also provides time for each task. And after Calculates the entropy value of each system and then system performs scheduling and apply Novel algorithm which is based on entropy value.

4) If any server node in cluster system is not returning a result within a given time, then the system will decide to perform migration or not.
5) After that it collects the result, the load after scheduling and the load after migration.
6) The load balancing is achieved by increasing the performance of the system by executing the task in the shortest time.

Novel algorithm

Whenever the task is arrived, using the collected information about the system, then it calculates the entropy value. Using Calculated entropy value, select a minimum entropy value; if the entropy value is minimum then it performs migration , if not then shows the system is balanced.

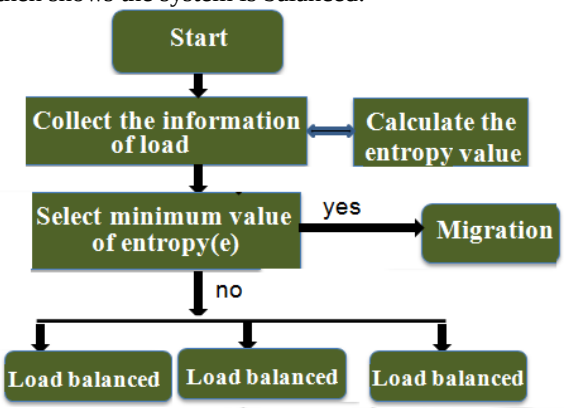

Figure 4: Novel Algorithm

3.2 Expected Results

The Entropy based cluster system for balancing the load which can speed up the process and execute the result within minimum time. The cluster system with Novel algorithm depends on Entropy, which provides very good results as compared to the traditional algorithm. The comparison between the result of traditional algorithm, i.e. load after migration and scheduling to the result of Novel algorithm i.e. load after migration and scheduling. It can also perform better for a large number of cluster systems. The figure 5 below represents how system is performing and how results are collected after the migration and scheduling.

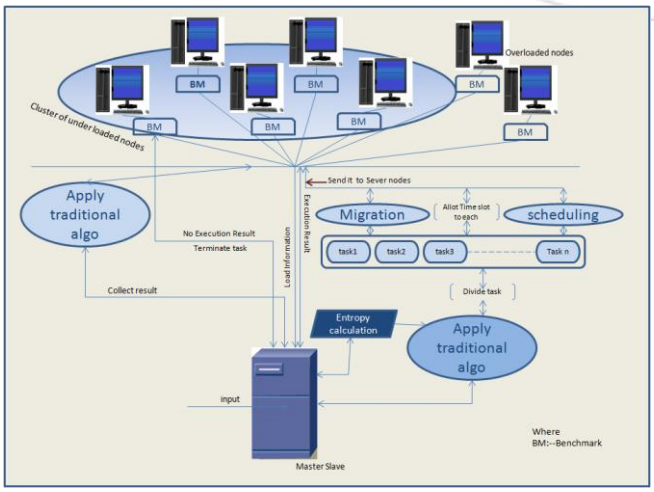

Figure 5: Graphical representation

4. Conclusion

Previously, some techniques have been developed for a Homogeneous cluster system like concept which depends on entropy for balancing the load which was implemented successfully but precisely not on different types of cluster systems. 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 objective is to achieve load balancing faster and better.

The novel algorithm will be better 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. The 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.

References

[1] 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)
[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] 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).
[4] S. C. Chauu and W. C. F. Ada, "Load balancing between computing clusters", Proceedings of the 4th International Conference on Parallel and Distributed Computing, Beijing, China, (2003)
[5] 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).
[6] A. Heirich, "Analysis of Scalable Algorithms for Dynamic Load Balancing and Mapping with Application to Photo-realistic Rendering", Caltech, Pasadena, (1998).
[7] 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).
[8] Siu-Cheung Chau, Ada wie-chee fu, "Load Balancing Between Heterogenous Computing Clusters" in National Sciences and Engineering Research Council of Canada. [2004].
[9] Vatsal Shah , Kanu Patel, "Load Balancing algorithm by Process Migration in Distributed Operating System", in year 2012.
[10] Jiani Guo, Laxmi Narayan Bhuyan," Load Balancing in a Cluster-Based Web Server for Multimedia Applications", in year 2006