Asian Research Consortium

Asian Journal of Research in Social Sciences and Humanities
Vol. 6, No. 9, September 2016, pp. 1430-1443.

ISSN 2249-7315
A Journal Indexed in Indian Citation Index

DOI NUMBER: 10.5958/2249-7315.2016.00881.9

Category:Science and Technology

Asian Journal of Research in Social Sciences and Humanities

www.aijsh.com

Hybrid Genetic Algorithm based Proficient Resource Management and Scheduling in Grid Computing

G. Saravanan*; V. Gopalakrishnan**

*Research Scholar,

Anna University,

India.

**Associate Professor,

Government College of Technology,

Coimbatore, India.

Abstract

Grid networking is an aggregation of geographically dispersed computing, storage, and network resources, coordinated to deliver improved performance, higher quality of service, better utilization, and easier access to data. It enables virtual, collaborative organizations, sharing applications and data in an open, heterogeneous environment. Scheduling is the process that selects which job in the queue should be considered next. Grid Scheduling is the process of making scheduling decisions involving allocating jobs to resources over multiple administrative domains. The goal of scheduling is to minimize the make-span by finding an optimal solution. In the present Grid Networking environment, the scheduling approaches for VM (Virtual Machine) resources only focus on the current state of the entire system. Most often they fail to consider the system variation and historical behavioral data which causes system load imbalance. In existing system is based only on future load prediction mechanism. Based on this factor VM resource allocation is done. During this VM migration, there is no suitable criterion for unique identification and location of VM that means which VM is migrated and where to be migrated. In this paper a Grid Booster Algorithm is using. In this system VM allocation is based on resource weight [a value indicates capacity of each resource]. Based on these weights a VM resource allocation mechanism has proposed, which is considering both resource weight and future prediction. To produce a better approach for solving the problem ofVM resource migration in a Grid Networking environment, this paper demonstrates Acclimatize Genetic Algorithm based VM resource migration strategy that focuses on system load balancing.

Keywords: Grid Networking, Grid Booster Algorithm, Acclimatize Genetic Algorithm.

1. Introduction

Gridis essentially a collection of distributed computing resources (within one or multiple locations). The computer resources are aggregated to act as a unified processing resource (virtual super-computer). The process of interconnecting these resources into a unified pool involves the coordination of the usage policies, job scheduling, queuing issues, Grid security, as well as user authentication. Grid Networking is to interconnect and utilize available processor, storage, or memory subcomponents of distributed computing systems to solve larger problems more efficiently. The benefits of Grid Networking are (1) cost savings, (2) improved business agility by decreasing the time (delivering actual results), and (3) enhanced collaboration and sharing of resources among jobs. In recent years, the researchers have proposed several efficient scheduling algorithms that are used in Grid networking to allocate grid resources with a special emphasis on job scheduling.

Grid Networking focuses on providing a computing infrastructure that leverages system virtualization to allow multiple Virtual Machines (VM) to be consolidated on one Physical Machine (PM).VMs are loosely coupled with the PM they are running on; as a result, not only can a VM be started on any PM, but also, it can be migrated to other PMs in the Grid centre. Migrations can either be accomplished by temporarily suspending the VM and transferring it, or by means of a live migration in which the VM is only stopped for a split second. With the current technologies, migrations can be performed on the order of seconds to minutes depending on the size and activity of the VM to be migrated and the network bandwidth between the one. The ability to migrate VMs makes it possible to dynamically adjust data centre utilization and tune the resources allocated to applications. Furthermore, these adjustments can be automated through formally defined strategies in order to continuously manage the resources in a data centre with less human intervention. We referred to this task as the process Acclimatize Genetic Algorithm in a data centre. The design goals are follows.

Performance Objective: We consider computational& memory resources and the objective is to achieve reasonable fairness among jobs for computational resources under memory constraints.

Adaptability: The resource allocation process must dynamically and efficiently adapt to changes in the demand from jobs.

Scalability: The resource allocation process must be scalable both in the number of resources in the Grid and the number of jobs that the Grid hosts. This means that the resources consumed per machine in order to achieve a given performance objective must increase sub linearly with both the number of machines and the number of jobs.

This system fits well in a grid for efficiently parallelizing incoming set of tasks using large data. In this, a job initiates one virtual machine (VM) and on the go, based on the no: and complexity of subtasks, further VMs are allocated and de-allocated. It introduces an idle queue between the Job Manager and the Task manager (VM). Whenever a VM becomes idle, it joins the queue. When a job approaches the Job manager, it fetches the first VM in the queue and allocates the job to it. Hence this method reduces communication overhead and thereby improves throughput of the data processing system. Virtual machine related features such as flexible resource provisioning, and isolation and migration of machine state have improved efficiency of resource usage and dynamic resource provisioning capabilities.

1.1 Virtualization

The goal of Grid Networking is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The grid aims to cut costs, and help the users focus on their core business. The main enabling technology for Grid Networking is virtualization. Virtualization generalizes the physical infrastructure, which is the most rigid component, and makes it available as a soft component that is easy to use and manage. By doing so, virtualization provides the agility required to speed up the operations, and reduces cost by increasing infrastructure utilization. On the other hand, autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process and reduces the possibility of human errors.

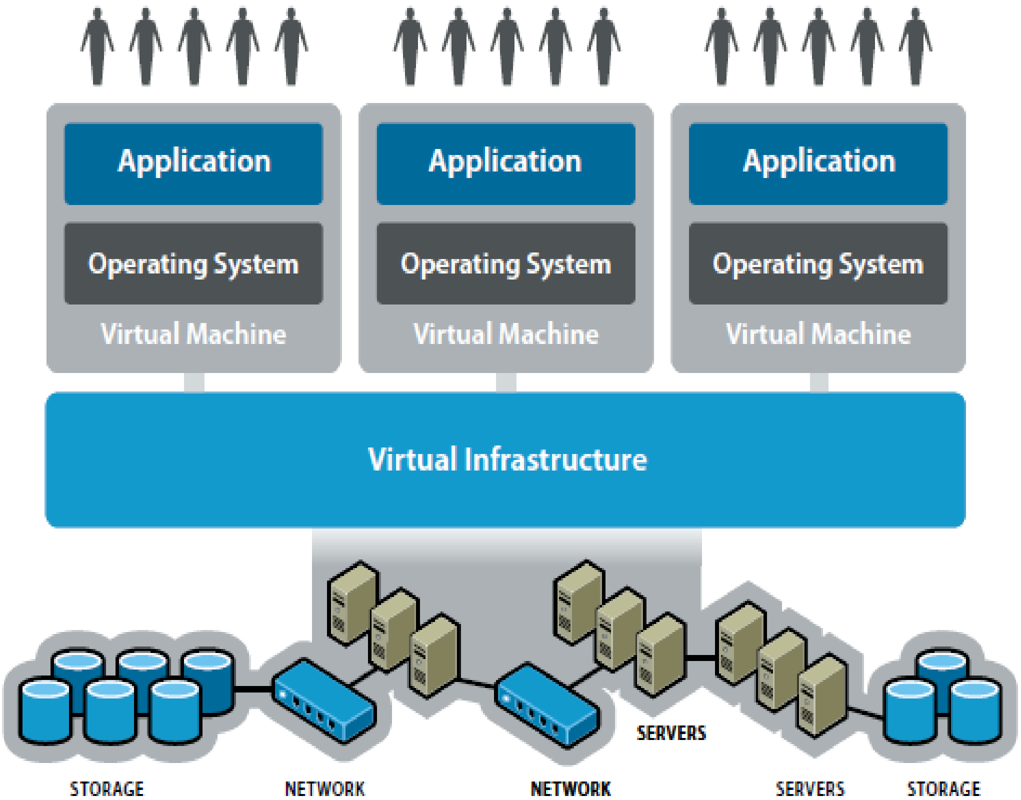


Figure (1.1) Virtualization

Efficiency is readily available in Virtualized Environments and Machines. On a Grid Networking platform, dynamic resources can be effectively managed using virtualization technology with virtualized server instances. Whereas some resources like CPU, memory, and instance storage are dedicated to a particular instance, other resources such as the network and the disk subsystem are shared among instances. If each instance on a physical host tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is under-utilized, it is often able to consume a higher share of that resource while it is available. This helps to solve problem of heterogeneity of resources and platform irrelevance.

1.2Requirement of Adaptive Genetic Algorithm

Adaptive Genetic algorithm is a random searching method that has a better optimization ability and internal implicit parallelism. It can obtain, and instruct the optimized searching space and adjust the searching direction automatically through the optimization method of probability. With the advantages of adaptive genetic algorithm, this paper presents a balanced scheduling strategy of VM resources in Grid Networking environment. By considering the current states and historical data, this method will compute in advance its influence over the entire system.

2. System Architecture

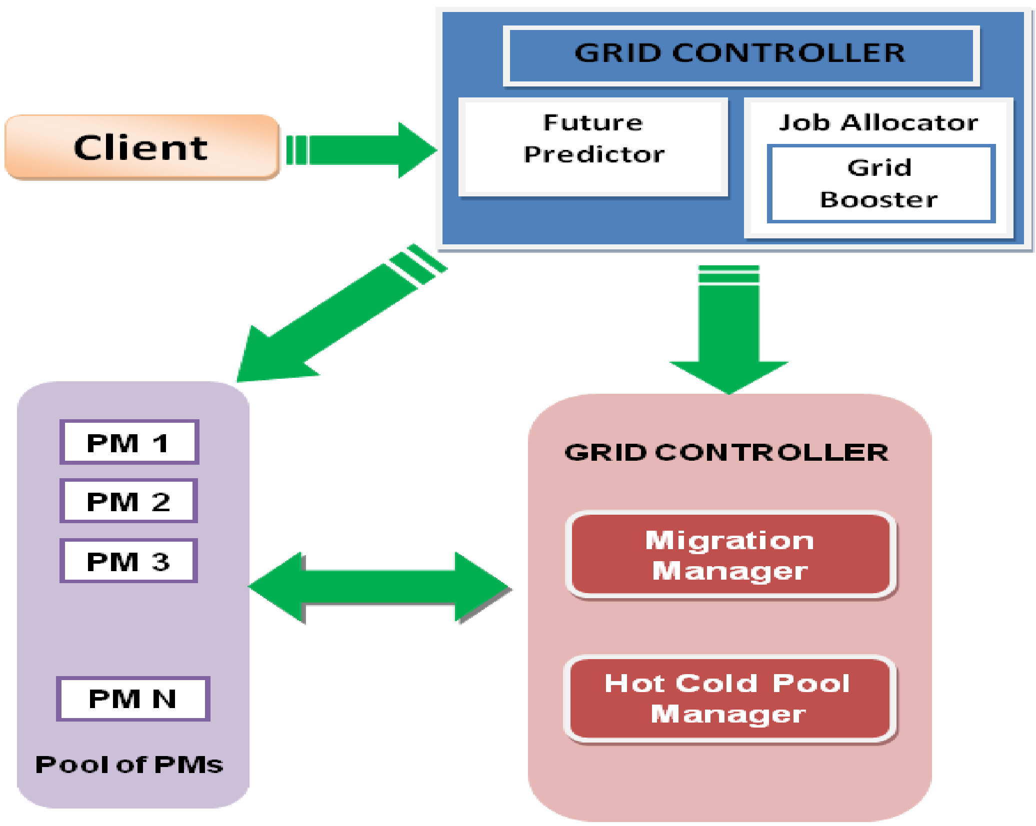


Figure (2.1) System Architecture

This paper also addresses a fundamental problem in virtual machine (VM) resource management: how to effectively profile physical resource utilization of individual VMs. Our focus is on extracting the utilization of physical resources by a VM across time, where the resources include CPU (utilization in CPU cycles), memory (utilization in memory size). Correct VM resource utilization information is tremendously important in any autonomic resource management that is model based. Hence, resource management is completely based on resource mapping across virtual machines.

This proposal provides a scheduling strategy to enable effective load balancing by using Grid Booster Algorithm. This method used in this proposal will compute its influence on the system in advance. Efficient load balancing in a grid is challenging since running machines have the problem of load imbalance due to resource variation in heterogeneous environment. For heterogeneous systems nodes have different processing capabilities, dynamic load balancing methods are preferred. This approach makes load balancing decision based on node weights which varies on each machine i.e. distribute load on the nodes at run time. Modern parallel computing hardware demands increasingly specialized attention to the details of scheduling and load balancing across hetero-geneous execution resources in grid Networking. To produce a better approach for solving the problem of VM resource migration in Grid Networking environment . When current VM resources are allocated to every physical node and will opt for the deployment that will have the least load on the system. This paper demonstrates an acclimatize genetic algorithm based VM resource migration strategy that focuses on system load balancing. This algorithm computes the impact in advance, that it will have on the system after the new VM resource is deployed in the system, by utilizing historical data and current state of the system. It then picks up the solution, which will have the least effect on the system. By doing this it ensures the better load balancing and reduces the number of dynamic VM migrations.

2.1 Job Allocation

Clientsubmit job requests to the Grid Controller. It will distribute the job among virtual machines (VMs). Then the Job will be executed on the different physical machines (PMs). See figure (2.2).

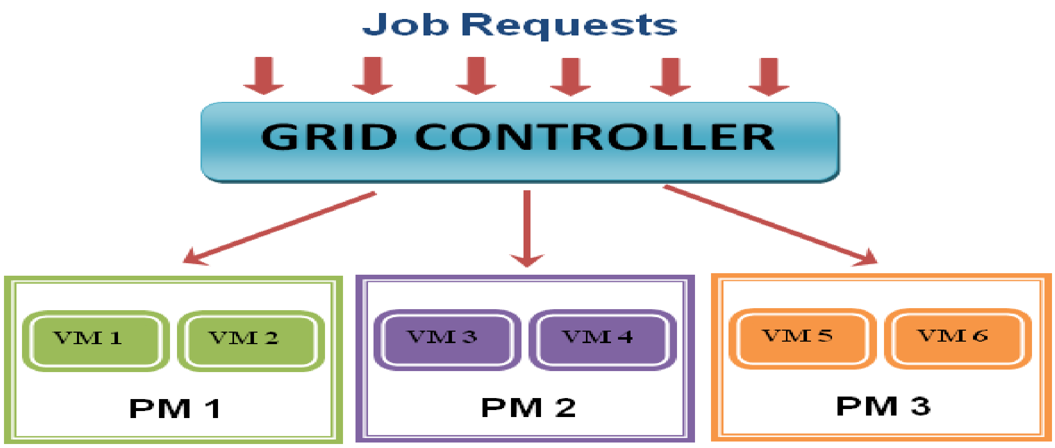


Figure (2.2) Job Allocation

While allocation it will consider the results from the future predictor. After the allocation, PMs will do the jobs independently. After the successful completion of job, the results will be sent back to the Grid controller. The future predictor will predict the future resource needs of VMs based on the past statistics. See figure (2.1). One solution is to look inside a VM for application level statistics, e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible instead of making our prediction based on the past external behaviours of VMs. It measures the load on every minute and predicts the load in the next minute. Based on the prediction result, the given jobs will be distributed to the Grid resources.

2.2 VM Model

The relationship between the physical machines and the VMs can be seen from figure 1.1 Consider P as a set of all the physical machines in the entire system, where P = {P1, P2, P3 …. PN}. N is total number of the physical machines and an individual physical machine can be denoted as Pi, where i denote the physical machine number and range of i is (1 <= i <= N). Similarly, we have a set of VMs on each physical machine Pi, Vi = {Vi1, Vi2, Vim} here m is the number of VMs on the physical server i . If we want to deploy VM V on the present system, then we have a solution set denoted by S = {S1, S2, S3 …. SN}, it represents the mapping solution after VM V is assigned to each of the physical machines. When the V is arranged with the physical machine Pi we get the mapping structure denoted as Si.

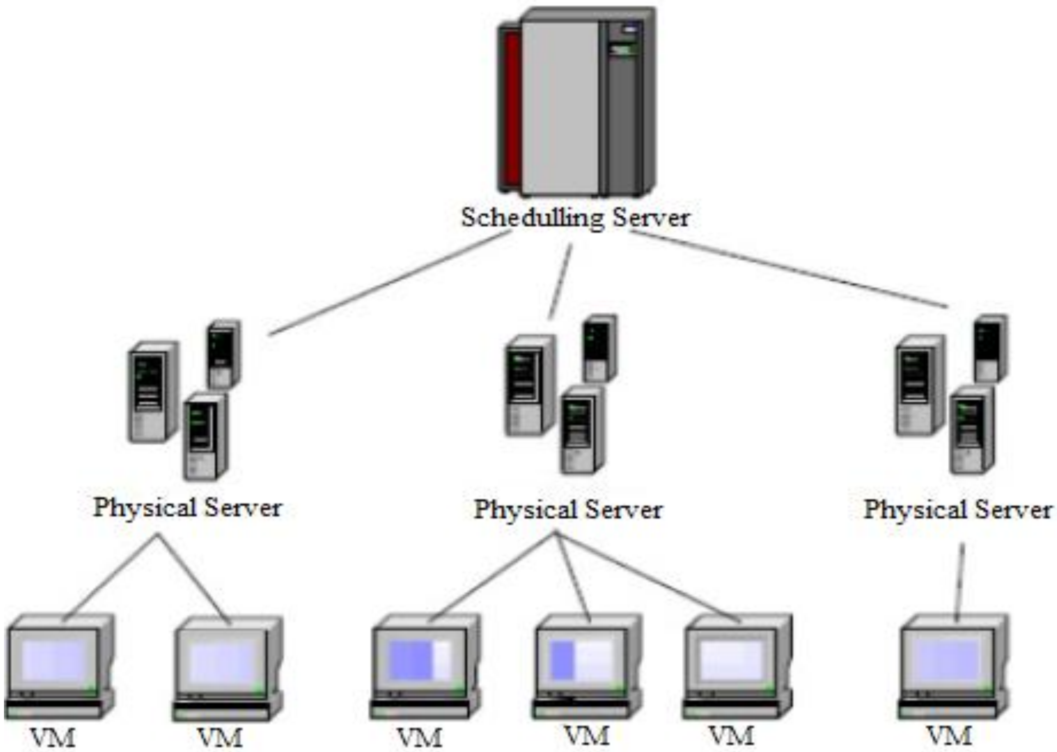


Figure (2.3) System Structure

3. Algorithms Used

- Grid Booster algorithm for Job Allocation.
- Acclimatize Genetic Algorithm for resource provisioning during job migration situation for minimizing migration cost.

3.1 Grid Booster Algorithm

This algorithm has two phases. First phase is used for calculating weight of the node in the parallel processing system. Second phase is used for estimating job proportion required by each parallel machine based on weight metrics.

Notations:

WCPU - CPU Weight.

WMem - Memory Weight.

WLoad - Weight for current load.

Resource factors for calculating weight for a particular machine is tabulated as follows:

Table (3.1) Resource Information

Resource	Sample Value for a Machine	Weight Constant
CPU Speed(GHZ)	2.4	0.5
Main Memory(GB)	2	0.4
CPU Utilization	0.013	0.5
Memory Utilization	0.536	0.4

3.1.1 Node Weight Calculation Procedure

i. Collect resource information from each parallel node.

ii. Calculate Weight for each resource separately by excluding negative factors.

iii. After estimating WCPU, WMem and WLoad, determine the weight of a node by using the formula.

Weight of Machine = WCPU + WMem – Wload

3.1.2 Job Allocation Procedure

i. Calculate the required job Percentage for a machine using the formula.

For a Node X, Wx = (Wx/ ∑ Wi.).

Where $\sum$ Wi is the total weight of all the nodes.

ii. Divide the job based on job proportion and distribute the job among parallel nodes. Thus Job (load) is submitted by the user and it is distributed among parallel nodes in parallel processing system and output is generated.

3.2 Acclimatize Genetic Algorithm

This algorithm has three phases.

i. Selection

- Randomly generate initial sequences
- Each sequence may contain job as well as node.
- Selection process - generate 8 sequences.
- First copy 4 sequences from initial sequences.
- Generate 4 remaining sequences as randomly select sequences from initial sequences with best fitness value.

ii. Cross Over

- Cross over process- Generate 8 sequences.
- First copy 4 sequences from selection sequences.
- Remaining 4 sequences are generated as combining the "selection sequences".

iii. Mutation

- Generate 8 sequences.
- First copy 4 sequences from Cross over sequences.
- Remaining 4 sequences are generated as randomly select a sequences from above & Perform some interchange in the sequence.
- Sort these sequences on rank base, select first n sequences with min weight.

$$R = \sum_{i=1}^{N_j} \frac{\frac{\boldsymbol{Wi}}{\boldsymbol{Max(Wi)}}}{\boldsymbol{N_j}} + \sum_{i=1}^{N_j} \frac{\frac{\boldsymbol{Max(Pi) - Pi}}{\boldsymbol{Max(Pi)}}}{\boldsymbol{N_{vm}}} + \sum_{i=1}^{N_{vm}} \frac{\frac{\boldsymbol{Ti}}{\boldsymbol{Max(Ti)}}}{\boldsymbol{N_{vm}}} + \sum_{i=1}^{N_j} \frac{\frac{\boldsymbol{Min(Ci) - Ci}}{\boldsymbol{Max(ci)}}}{\boldsymbol{N_j}}$$

Equation (3.1) Weight Calculation

R - Rank

N_j - Number of jobs

W_i - Weighting time of i^{th} Job

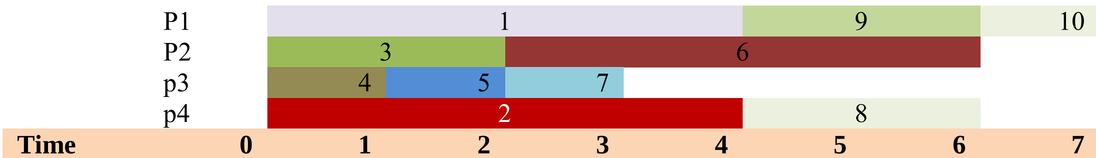
P_i - Communication Cost of i^{th} Job

N_{vm} - Number of Jobs

T_i - Idle time i^{th} VM

C_i - Migration cost of i^{th} Job

Consider Following Allocation



Waiting Time of Job 9: 6

Here highest waiting time 7 (job 10)

PW (Percentage of waiting time) =Sum (waiting time for each job) / (highest waiting time * number of jobs)

Idle time of p4=1, Highest idle time=4

PI (Percentage of idle time) =Sum (idle time for each node) / (highest idle * number of nodes)

PP (percentage of communication cost) =Sum (communication cost of each node) / (Maximum communication cost * number of jobs).PP (percentage of replication cost) =Sum (replication cost of each node) / (Maximum replication cost * number of jobs).

Rank= PC + PP + PW + PI

Here we select sequence with minimum rank. In this way, AGA (Acclimatize Genetic Algorithm) generates new sequence. During this generation, AGA minimizes rank. Here rank is the sum of waiting time, communication cost, replication cost and idle time. So these generations minimize above factors.

4. Experimental Results & Performance Analysis

4.1 The Expression of Load

The summation of all the running VMs on a physical machine can be termed as the load of the physical machine. If we assume that the T is the best time duration to monitor historical data i.e., from current time to the last T minutes will be the historical data zone, which will be used to solve the load balancing problem.By using the variation law of the physical machine load, we can split T into n subsequent time intervals. Therefore, T becomes [(t1- t0), (t2 - t1),..,(tn - tn-1)].

The time interval k can be defined as (tk – tk-1). If the load of a VM is stable in all the periods then V(i, k) refers to the load of VM i in the interval k.

By using the above definition, we can define the average load of VM on physical server Pi in time cycle T is

$$Vi(i,T) = \frac{1}{T}\sum_{k=1}^{n} V(i,k)\ x\ (t_k - t_{k-1})$$

Equation (4.1) Average load of VM on Pi in time T

By using the above average load definition of VM, we can calculate the load of the physical machine for last T interval by adding all the loads of the VMs present on the physical machine. So, the expression to compute the load of physical machine Pi is as follows:

$$P(i,T) = \sum_{j=1}^{mi} V_i(j,T)$$

Equation (4.2) Load of the PM for last T interval

The Grid Booster algorithm provides an effective dynamic load balancing method in heterogeneous grid environments. In order to test the performance of this work, a graph is plotted with Job size or load on X axis and response time on Y axis.

Table (4.1) Resource Information

Job Size (Load) Bytes	**Grid Booster (Response Time)ms**	**Existing System (Response Time) ms**
3965001	14976.5	19513.5
7933009	29421.5	24394
15864174	66621	89658
31730654	113075	192336

Parallel processing setup is constructed with 3 different machines. Each machine ishaving different processing capacity. Data sets are obtained from giving inputs of different job size and response time is noted. That is shown in table 4.1.

From the above data, performance diagram is plotted separately for Grid booster and exponentially weighted moving average (EWMA). It is shown in figure (4.1). From the graph it is clear that as load increases, proposed system performed exceptionally well and response time of job is considerably reduced. In existing system, as load increases response time is also increased linearly. That is response time is very high for task having larger job size.

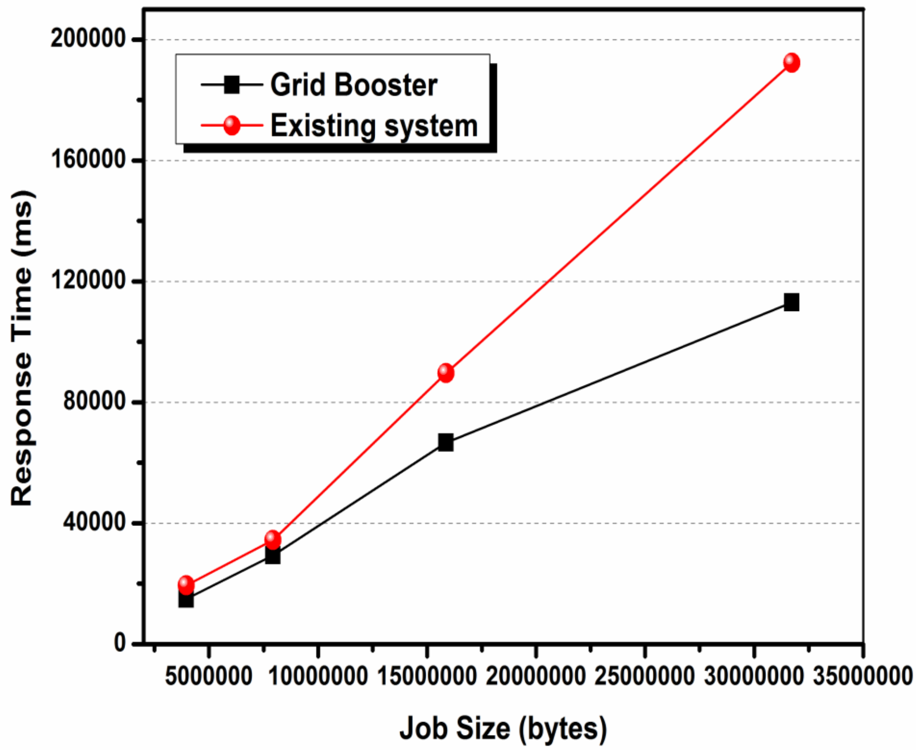


Figure (4.1) Job Allocation

To analyse the behaviour of this acclimatize genetic algorithm with VM scheduling, we have got a scheduling details within 0.5 micro seconds for following inputs.

Table (4.2) Scheduling

No : of jobs	2000
No: of nodes	30
No: of GA rounds	100
No : of chromosomes	20

After a no of rounds of various inputs, we can realize that as number of rounds increases we got better scheduling data. i.e. scheduling job with minimum waiting time, Consumer cost, provider idle time and maximum provider benefit. The VM loads are balanced on every physical node and also the system load variation parameter is also achieved. As a result, we can conclude that the above explained algorithm has quite a better globally astringency and it can come closer to be the best solution in very little time. These results are shown in following table.

Table (4.3) Experimental Results

SL.No	Weight	Comm.-Cost	Rep- Cost	Idle-Time	Waiting-Time
1	42272	2680	3111	4404	13532
10	38121	2408	2842	3209	12505
20	37036	2336	2769	3334	12478
30	36384	2293	2724	3387	12239
40	35770	2253	2679	3415	12095
50	34578	2174	2598	3480	12061
60	33960	2134	2555	3444	11579
70	33011	2070	2489	4363	11488
80	31916	1998	2415	4255	11325
90	31144	1947	2363	4303	11156
100	30102	1878	2295	4360	10835
110	29239	1821	2239	4402	10543
120	28258	1756	2170	4470	10619
130	27423	1704	2117	2722	9964
140	26612	1649	2062	3453	9733
150	25692	1588	2000	3505	9609
160	24827	1531	1938	3438	9452
170	24254	1495	1903	2716	9133
180	23691	1458	1863	2739	9092
190	23096	1418	1825	2774	9096
200	22382	1370	1779	2928	9087

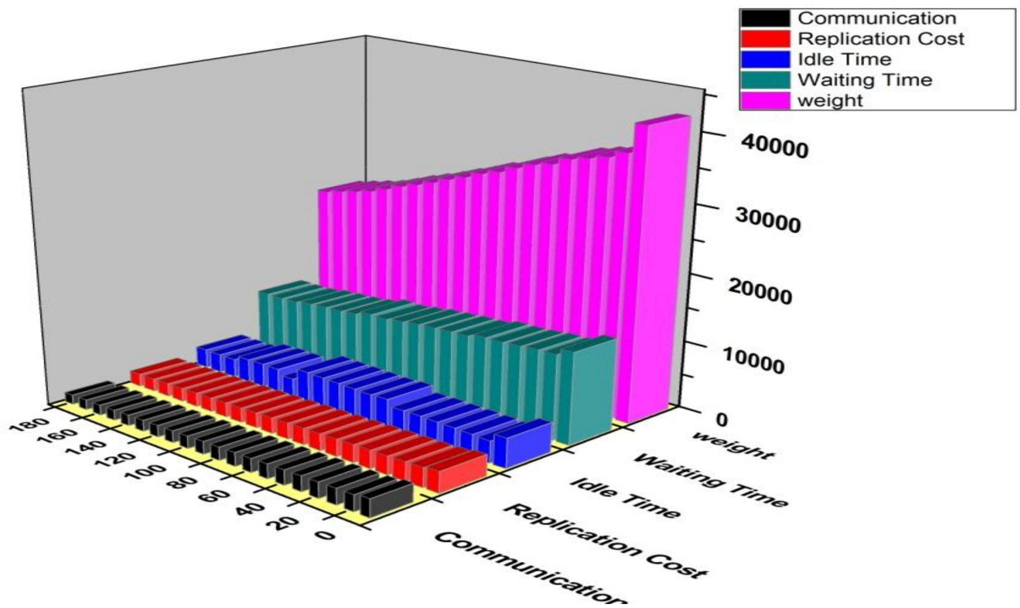


Figure (4.2) Performance Diagram for different Parameters against Number of Iteration

5. Conclusion

Most of the firms are moving to Grid Networking environment now days. Moving to Grid Networking is clearly a better alternative as they can add resources based on the traffic according to the cost model. But for Grid Networking to be efficient, the individual servers that make up the data center grid will need to be used optimally. Even an idle server consumes about half its maximum power. With the emergence of Grid Networking in the past few years, map reduce has seen tremendous growth especially for large-scale data intensive computing. The lack of a separate power controller in map reduces frameworks post an interesting area of research to work on. This paper addresses the issue of job migration for clusters of nodes that run Map-Join-Reduce jobs. So migration cost based job reconfiguration algorithm is proposed that dynamically reconfigures the job in accordance with the workload imposed on it. The work for implementation of a new framework Migration based reconfiguration of nodes, a system that extends and improves Map Reduce runtime framework to efficiently process complex data analysis tasks with migration using Genetic Algorithm is in progress.

References

Dr. K. Somasundaram, R. Natchadalingam, "Combinational Adaptive Advanced Scheduling Algorithm for Grid Networking"**,** European Journal of Scientific Research, ISSN 1450-216X Vol. 87,No 4, pp.573-584,2012.

DavideAdami, Christian Callegari, Stefano Giordano, Michele Pagano, "A Three-Dimensional Resource Scheduling Algorithm for a Network-aware Grid",IEEEGlobecom 2010 proceedings, pp. 978-1-4244-5638-3, 2010.

MustafizurRahman, Rajiv Ranjan, and RajkumarBuyya, "Dependable Workflow Scheduling in Global Grids"10th IEEE/ACM International Conference on Grid Computing, pp.978-1-4244-5149-4, 2009.

SaeedFarzi "Efficient Job Scheduling in Grid Computing with Modified Artificial Fish Swarm Algorithm" IACSIT- International Journal of Computer Theory and Engineering, Vol. 1, No. 1, pp. 1793-8201,2009.

Xuelin Shi, Ying Zhao, "Evolutionary Grid Scheduling Algorithm with Predictive Resource Optimization", IEEE Computer Society- The Fifth Annual ChinaGrid Conference, pp .978-0-7695-4106-8, 2010.

WaelAbdulal, S. Ramachandram, "Reliability-Aware Genetic Scheduling Algorithm in Grid Environment", IEEE Computer Society- International Conference on Communication Systems and Network Technologies,pp. 978-0-7695-4437-3,2011.

TarekHelmyZeehashamRasheed, "Independent Job Scheduling by Fuzzy C-Mean Clustering and an Ant Optimization Algorithm in a Computation Grid"IAENG International Journal of Computer Science pp. 37:2, IJCS_37_2_03, 2010.

Miguel Camelo, YezidDonoso, Harold Castro "MAGS – An Approach Using Multi-Objective Evolutionary Algorithms for Grid Task Scheduling" International Journal Of Applied Mathematics And Informatics, Issue 2, Volume 5, 2011.

AmelHamdi, Lynda Zitoune, V´eroniqueV`eque "MS-DFGA : Malleable Scheduling with Dynamic Max Flow and Greedy Algorithms in Grid Networks" IEEE Computer Society- Sixth Advanced International Conference on Telecommunications,pp. 978-0-7695-4021-4,2010.

SrdanVukmirovi, AleksandarErdeljan, LendakImre, NemanjaNedi "Neural Network workflow scheduling for large scale Utility Management Systems"IEEE-pp. 978-1-4244-6588-0,2010.

JayantUmale, Dr. SunitaMahajan "Optimized Grid Scheduling Using Two Level Decision Algorithm (TLDA)" Proceedings of the 1st International Conference on Parallel, Distributed and Grid Computing (PDGC-2010),pp. 978-1-4244-7674-9,2010.

Mr.P.Mathiyalagan,U.R.Dhepthie ,Dr. S.N.Sivanandam "Grid Scheduling Using Enhanced PSO Algorithm" (Ijcse) International Journal on Computer Science and Engineering, Vol. 02, No. 02, 140-145, ISSN : 0975-3397,2010.

D.Maruthanayagam, Dr. R.Uma Rani "Enhanced Ant Colony System Based On Rasa Algorithm In Grid Scheduling" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4), 1659-1674, ISSN: 0975-9646, 2011.

Xuelin Shi, Ying Zhao "Stochastic Model and Evolutionary Optimization Algorithm for Grid Scheduling" IEEE- CAS, Seventh International Conferenceon Fuzzy Systems and KnowledgeDiscovery (FSKD 2010), pp. 978-1-4244-5934-6, 2010.