CHAPTER 6

PERFORMANCE ANALYSIS OF MULTI-TASKING BASED LOAD BALANCING DURING RESOURCE ALLOCATION IN CLOUD

6.1. INTRODUCTION

The principle objective of Interference Aware Resource Allocation (IARA) Technique on resource scheduling in cloud computing environment achieves better allocation of resources, based on its processing capability, electric power and network bandwidth. IARA Technique enhances the resource scheduling process in cloud computing environment by employing multi-tasking based resource scheduler. IARA Technique improves the resource scheduling with the adoption of sub-optimization process to resolve the cloud computing problem. IARA Scheduling Technique achieves both the allocation of resources as well as the utilization of the system resource. The next work, termed Adaptive Load Balancing Approach (ALB) addresses the high energy utilization problem and load imbalance challenges through clustering. ALB along with repetitive query messaging operations balances the load with the calculation of average energy and bandwidth consumption of each group after information updating. The final work on GCWM Scheduler intends for better cluster operation of similar workload using genetic principles. GCWM Scheduler focuses on minimizing the computational cost and complexities

arising during computation. GCWM Scheduler guarantees the multi tasking operation with efficient user communication.

Extensive level of experimental studies has been conducted to illustrate the efficiency and effectiveness of the proposed works, like IARA Technique, ALB Approach and GCWM Scheduler. Simulation experiments have been conducted with various conditions to perform the resource scheduler in cloud computing. Evaluation of IARA Technique is carried out in terms of interoperability, computational cost and scalability. The performance metric for evaluation of ALB approach is measured in terms of load imbalance factor, computational cost and clustering efficiency. Finally, GCWM Scheduler proves its performance in terms of generic based cloud services, multi-task clustering effect, computational cost and computational complexity.

6.2. ANALYSIS OF INTERFERENCE AWARE RESOURCE ALLOCATION TECHNIQUE IN CLOUD ENVIRONMENT

Multi-user cloud environment gains more focus with appropriate resource allocation principles. The main objective of IARA Technique in multi-user cloud computing is achieved in terms of proper resource allocation logics. Moreover, IARA Technique resolves the issues posed by the existing systems in range of improper resource allocation with high computational complexity. Most of the resource allocation problems arise due to interference. Therefore, IARA Technique discards interference with optimal sub-channel. The metrics, like computational complexity in resource allocation and time

required to allocate the resources are measured to verify the performance of IARA Technique. In addition, the interoperability parameter also intends to prove the superior functioning of IARA Technique through quantitative analysis.

In experimental evaluation, these three sets of performance metrics, like interoperability, computational complexity, and resource allocation time are considered to analyze the performance of IARA Technique, whereas the comparison of IARA Technique with the existing works, like Congestion Control Method of Daniel Warneke and Odej Kao, (2011) [11] and Dynamic Resource allocation method aims to over-perform the proposed work of Takuro TOMITA and Shin-ichi KURIBAYASHI (2011) [58]. Here, the quantitative analysis on IARA Technique uses CloudSim Simulator. CloudSim Simulator executes codes by means of Command prompt or CloudSim with Eclipse, Netbeans, etc. supporting a simple pass through. The CloudSim Simulator has been chosen as a simulation platform as it is a current Simulation framework in Cloud computing infrastructure. Cloud availability framework at broadcast phase performs the best analysis, based on the traditional configurations supported within the CloudSim. Compared to the simulation tool kits (e.g. SimGrid, CloudSim), JAVA CloudSim supports copy of on-demand virtualization, permitted with bandwidth and submission management.

IARA work simulates a data center comprising 100 heterogeneous physical nodes. Each node is presented with a CPU core, equivalent to 1000, 2000 or 3000 MIPS, 8 GB of RAM and 1 TB of storage. The bandwidth consumption by the nodes in cloud environment is defined according to the model. Moreover, in IARA model, a node consumes energy from 175 W with 0% CPU utilization, up to 250W with 100% CPU utilization. IARA Technique uses Statlog (Shuttle) Data Set from UCI repository with 9 attributes, all of which are numerically enhancing the data validation accuracy to 80%. The examples in the definite dataset are in time order, and this time order can most possibly be associated with clustering.

The users reveal suitable features for provisioning the 290 assorted VMs, that have the energy of the virtual data center. Each VM runs a web-application or any phase of application with erratic workload, which is designed to engender the utilization of the CPU, according to a consistently shared random variable. The application runs for 150,000 MI, that is equal to 10 min of its execution on 250 MIPS CPU with 100% utilization. Initially, the VMs are owned along with the demanded uniqueness assuming 100% CPU utilization. Each experiment has been run for 10 times. The performance of IARA Technique is evaluated in terms of interoperability, computational cost and scalability.

6.2.1. Quantitative Analysis for Interoperability

The main challenge of cloud platform is the interoperability issue. Interoperability is defined as the increased uptake in interference reach.

Interference in between the communication distracts the cloud communication in resource allocation. the lower the rate of interoperability, is the higher the performance of interoperability (I).

$$I = \frac{Success\ Rate\ in\ Communication}{Resource\ Allocation\ Time} - Hotspot * 100 \ \ldots (1)$$

Here, the interoperability measure is decided based on the equation (1). An optimal result of interoperability is detected on a minimum time period allocation of resources, triggering the hotspots. Interoperability is measured in percentage (%).

Table 6.1
Tabulation of Interoperability

No. of Resource Request	Interoperability (%)		
	Congestion Control Method	Dynamic Resource Allocation Method	IARA Technique
20	63	55	48
40	68	60	53
60	89	79	67
80	79	67	58
100	83	72	60
120	74	64	56
140	87	77	64
160	93	85	71

Table 6.1 describes the interoperability of the IARA Technique in comparison with the existing congestion control of Daniel Warneke and Odej Kao, (2011) and Dynamic Resource Allocation Method of Takuro TOMITA and Shin-ichi KURIBAYASHI, (2011).

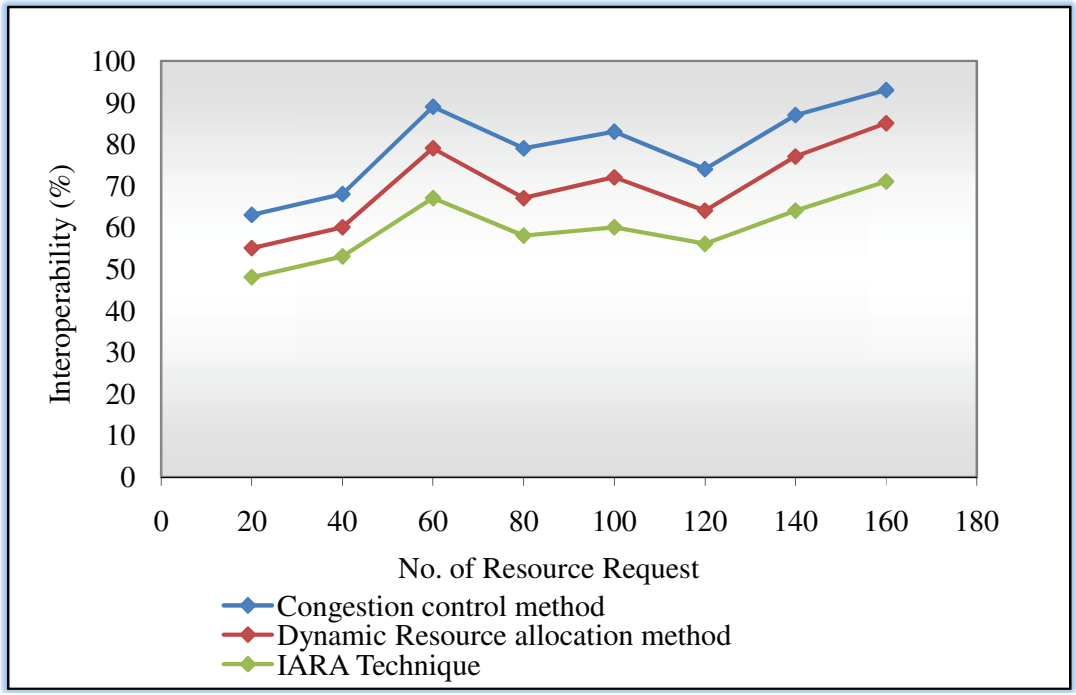

Fig 6.1 Measure of Interoperability

Fig. 6.1 describes the interoperability measure in interference negotiation for resource allocation with respect to the respective resource requests. IARA Technique reduces the interoperability rate by about 22-27% in interference minimization when compared with the Dynamic Resource Allocation in Takuro TOMITA and Shin-ichi KURIBAYASHI, (2011) [58] and nearly 11-16% when compared with the Congestion Control Method in Daniel Warneke and Odej Kao, (2011) [11]. The reason behind the lower

interoperability is that IARA Technique finds the expected time overlap between the transmissions of infrastructure and ad-hoc connection which form the cost function and thereby reduce the interference. However, both the existing systems face interference due to the presence of hotspots at irregular differences.

6.2.2. Measure of Scalability

Scalability means that cloud computing provides infinite processing and storage capacity. The cloud, which is consistent in that service provider facilitates access to applications and data even in remote areas in and around the cloud infrastructure.

Table 6.2
Tabulation of Scalability

No. of VM	Scalability (%)		
	Congestion Control Method	Dynamic Resource Allocation Method	IARA Technique
5	36	41	45
10	40	46	50
15	42	48	53
20	56	61	68
25	49	53	59
30	61	67	72
35	63	76	81
40	66	79	85

Scalability rate of the IARA Technique is compared with the existing works like Congestion Control Method in Daniel Warneke and Odej Kao, (2011) [11] and Dynamic Resource Allocation Method in Takuro TOMITA and Shin-ichi KURIBAYASHI, (2011) [58] as illustrated in table 6.2.

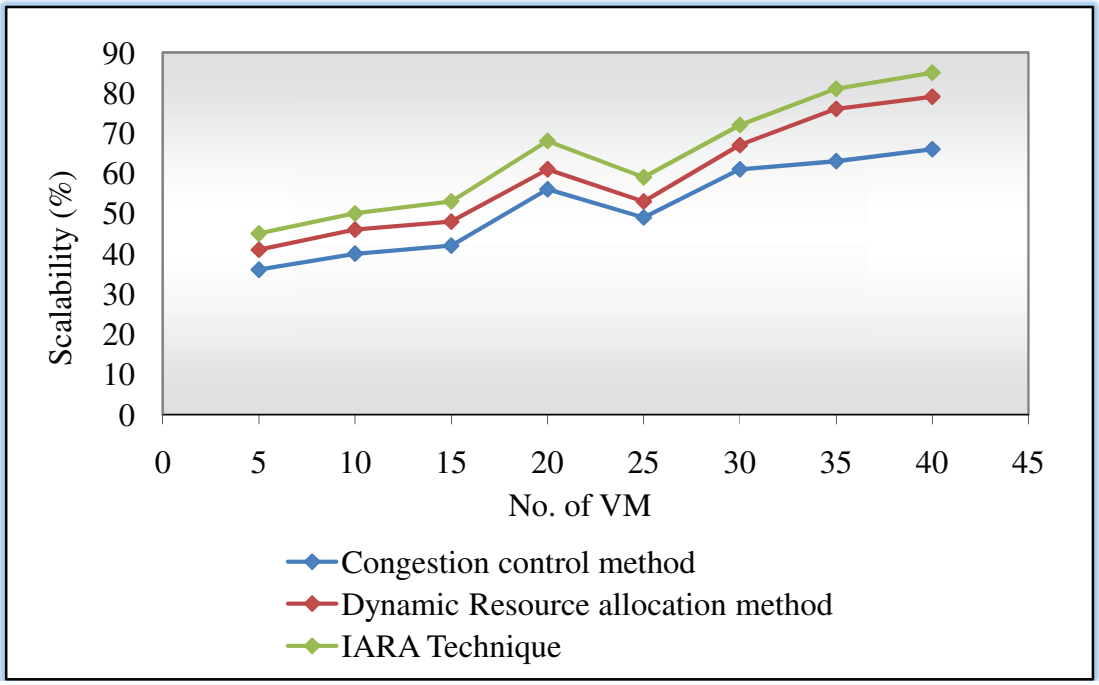

Fig 6.2 Measure of Scalability

Fig 6.2 describes the scalability with respect to the number of Virtual Machine, (VM) available. In Contrast to Congestion Control Method in Daniel Warneke and Odej Kao, (2011) [11] as well as Takuro TOMITA and Shin-ichi KURIBAYASHI, (2011) [58] in Dynamic Resource Allocation Method, the IARA Technique has approximately 18-28 % and 6-11 % better scalability. As IARA Technique carries out the dispersed process, which ensures the accepted out resource by individual node agents in the system, it satisfies the processing

and storage capacity. Moreover, every node running the tasks allocates resources to it that constantly achieves a cycle of placement, resource allocation and tuning using IARA technique.

6.2.3. Quantitative Analysis on Computational Cost

Computational Cost in resource allocation is defined as the time period required to assign jobs to opaque resources in heterogeneous cloud environment. For lesser time period allocation, higher performance of cloud systems is obtainable with lower computational cost.

Table 6.3
Tabulation of Resource Allocation Time

No. of VM	Computational Cost (ms)		
	Congestion Control Method	Dynamic Resource Allocation Method	IARA Technique
5	8.06	7.58	7.15
10	11.18	10.22	9.89
15	12.55	13.16	11.89
20	14.15	13.18	11.07
25	20.46	19.95	18.45
30	23.91	21.14	20.25
35	27.05	25.63	23.95
40	36.25	35.23	33.55

The computational cost of the proposed IARA Technique is compared with the existing Congestion Control Method and Dynamic Resource Allocation Method as shown in table 6.3. The computational cost is measured in milliseconds (ms).

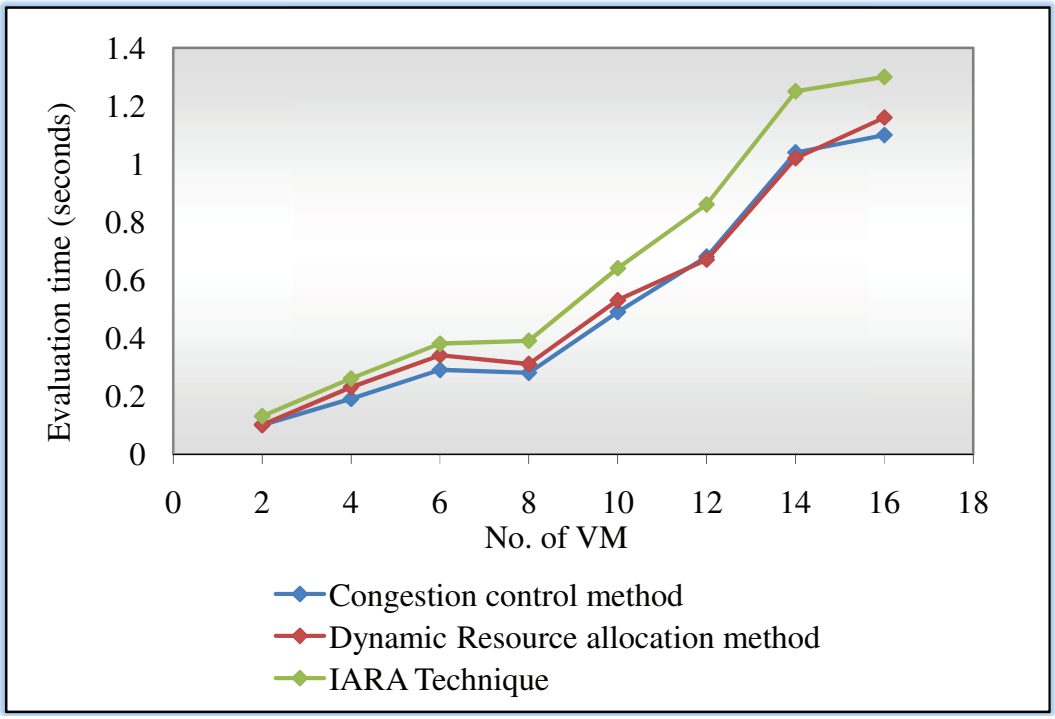

Fig 6.3 Measure of Computational Cost

Fig 6.3 describes the computational cost, based on the number of resources obtained in the environment. IARA Technique consumes computational cost of about 5-21% compared to the Congestion Control Method in Daniel Warneke and Odej Kao, (2011) [11] and 4-16% than Dynamic Resource Allocation Method in the Takuro TOMITA and Shin-ichi KURIBAYASHI, (2011) [58]. The main reduction in the computational cost of

IARA Technique in allocation of resources is that the dispersed process performs sub optimization with minimum processing speed and bandwidth within a specific period. In addition, before allocating the resources, the IARA technique will first monitor the activities of the task which highly reduces the additional computational cost involved in the datacenter. However, Congestion Control Method requires additional time in discovering the respective resources due to the fixed data centre. Even Dynamic Resource Allocation Method struggles to pre monitor the process, resulting in high computational cost in the final stage of delivery.

As a final point, the results conclude that IARA Technique is professionally good with resource allocation process in minimum computational cost, using the interference aware resource allocation method. Resource allocation process follows special hardware support, localized by the resource constrained cloud environment to enhance IARA Technique as it is justified with high scalability and lower interoperability metrics.

6.3. ANALYSIS OF ADAPTIVE LOAD BALANCING APPROACH IN CLOUD INFRASTRUCTURE FOR ENERGY CONSUMPTION

Adaptive Load Balancing (ALB) Approach efficiently clusters a group of servers in two main stages, like repetitive query messaging and clustering server group. Repetitive query messaging satisfies the performance metrics, like computational demand occurred during scheduled load using the mathematical instruction. The problem arising at load imbalance is resolved by

ALB on searching the most appropriate cluster group to share the load. Moreover, this resolves the engendered overhead during resource allocation. Energy resourceful scheduler employed in ALB optimizes the energy at each nodes during cloud computing. Further, ALB Approach equilibrates the data flows generated by the users with a minimum amount of bandwidth utilization. The better functioning of ALB is justifiable with certain metric analysis.

Experimental evaluations are conducted to analyze the performance of ALB Approach. For experimental evaluation purpose, a few set of metrics are considered for quantitative analysis. ALB Approach is implemented, using JAVA CloudSim Simulator. CloudSim Simulator runs under Command prompt or through CloudSim with Eclipse, Netbeans, etc. supporting an undemanding task. The specified CloudSim Simulator has been selected as a simulation platform as it is a present simulation structure in Cloud computing environments. Cloud availability structures at transmission layer carries out the optimal analysis, based on the custom configurations supported within the CloudSim. Compared to the simulation toolkits (e.g. SimGrid, CloudSim), JAVA CloudSim provides a copy of on-demand virtualization enabled bandwidth and submission management. Virtual machine simulates the data center comprising of 8 GB of RAM and 1 TB of storage. Energy consumption by the hosts is defined according to ALB Approach.

The user present needs to provision the 290 assorted VMs pack, which is the power of the virtual data center. Each VM runs a web-application or any kind of application with variable workload, which is modeled to generate the utilization of bandwidth according to evenly distributed random variable. ALB Approach uses Statlog (Shuttle) Data Set from UCI repository. The shuttle dataset contains 9 attributes all of which are numerical. Moreover, 80% of the data belongs to class 1 and they provide high accuracy in the validation process. The instances in the actual dataset are in time order, and this time order can most probably be related to clustering. ALB Approach is compared to Energy conservation in cloud infrastructures with Service Request Prediction (SRP) Model of Avinash Mehta., et al., (2011) and Homomorphic Verifiable responses Hash Index Hierarchy (HVHIH) of Yan Zhu., and Shanbiao Wang., (2012) [72] in terms of load balance factor, clustering efficiency and computational cost.

6.3.1. Measure of Load Balance Factor

The Load Balance Factor for ALB across multiple servers evaluates the maximal throughput used to avoid the overload and increases its reliability. Load Balance Factor of systems are measured in terms of Mega Bytes (MB).

Table 6.4
Tabulation of Load Balance Factor

Server Number	Load Balance Factor (MB)		
	SRP Model	HVHIH Mechanism	ALB Approach
Server 1	45	58	70
Server 2	77	82	90
Server 3	48	55	67
Server 4	56	68	78
Server 5	39	45	56
Server 6	55	65	77
Server 7	60	70	80

The table 6.4 describes the Load Balance Factor of ALB Approach against the SRP Model and HVHIH Mechanism.

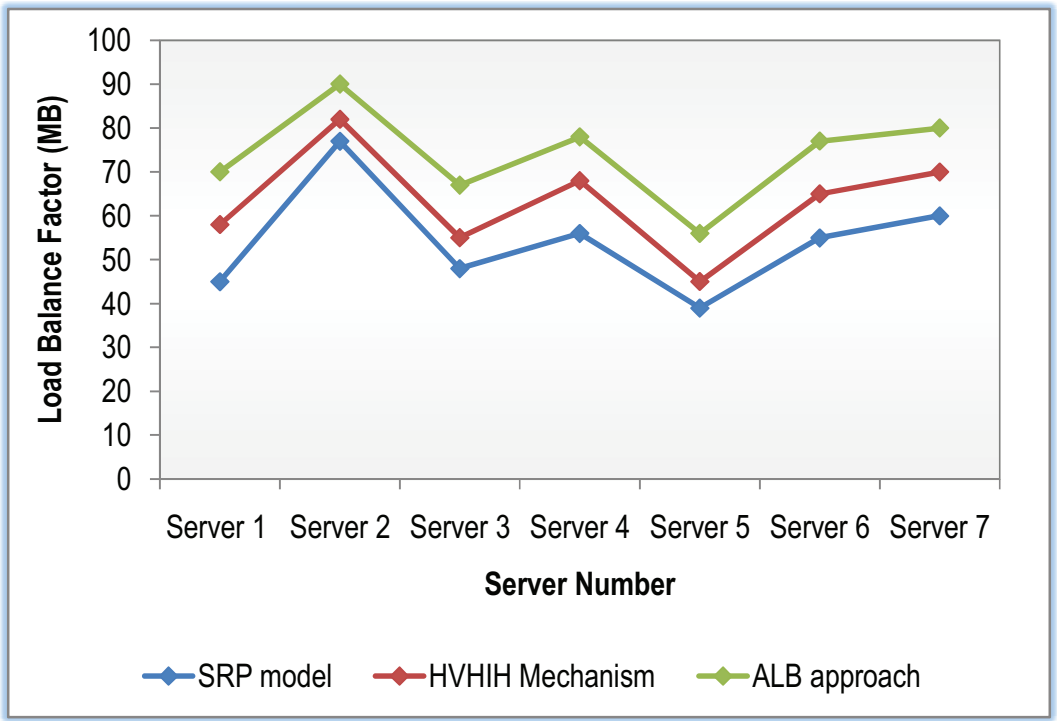

Fig 6.4 Measure of Load Balance Factor

Fig 6.4 illustrates the Load Balance Factor, using the three works, SRP Model, HVHIH Mechanism and ALB Approach. The Load Balance Factor of the ALB Approach is improved as it obtains the cluster head, each time when the imbalances occur with respect to the load threshold. The Load Balance Factor is 15 – 25 % improvement in ALB Approach when compared with SRP Model in Avinash Mehta., et al., (2011) and 8 – 12 % improvement in HVHIH mechanism in Yan Zhu., and Shanbiao Wang., (2012) [72]. This is because the ALB Approach verifies the load metrics against two thresholds in the member table to balance it, based on values. The usage of member table information for query processing in ALB Approach properly balances the load with better energy consumption and bandwidth utilization.

6.3.2. Quantitative Analysis of Clustering Efficiency

Clustering efficiency for ALB is defined with effective result on the grouping of similar servers to enhance the query processing result. Clustering efficiency of methods is measured in terms of percentage (%).

Table 6.5
Tabulation of Clustering Efficiency

No. of Frames	Clustering Efficiency (%)		
	SRP Model	HVHIH Mechanism	ALB Approach
10	85	86	95
20	84	90	96
30	83	89	95
40	81	87	95
50	80	90	95
60	79	86	96
70	75	90	96

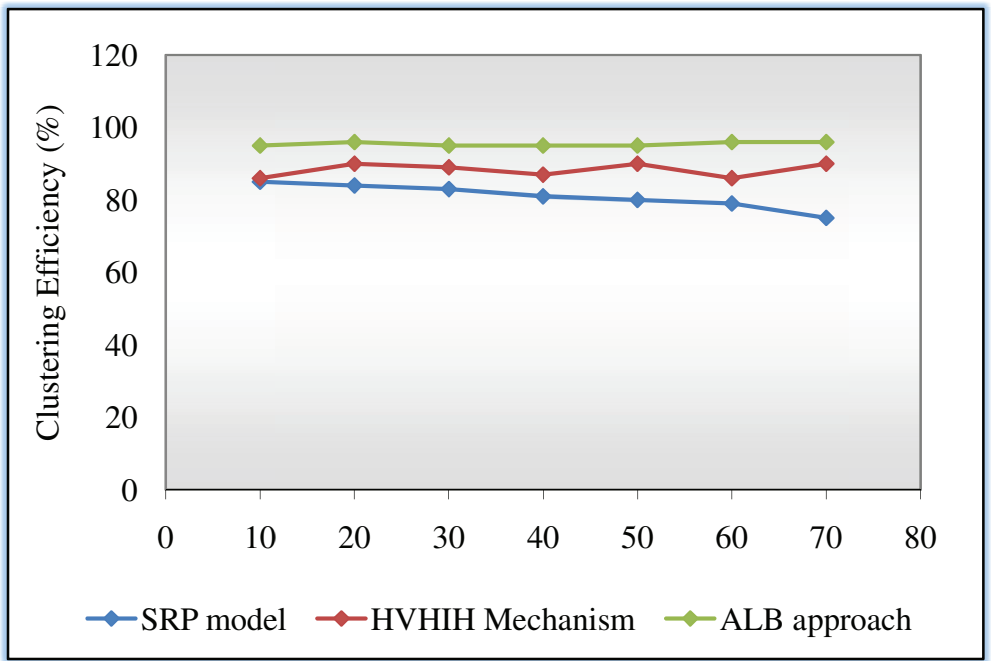

Fig 6.5 Clustering Efficiency Measure

Table 6.5 and Fig 6.5 describe the clustering efficiency, based on the frame count in the cloud infrastructure. The clustered group of servers in ALB Approach chooses a computing server to satisfy the computational demands. As a result, the ALB Approach improves its efficiency in clustering by approximately 8 – 18 % when compared with the SRP Model in Avinash Mehta., et al., (2011) and HVHIH Mechanism in Yan Zhu., and Shanbiao Wang., (2012) [72]. ALB Approach holds Cluster Head (CH) Server to maintain their member tables in order to control their member loads. Periodically, in each node the member of a cluster sends a message and communicates its energy and load values to the CH which updates its member table for efficient clustering. However, SRP Model and Homomorphic

verifiable responses and hash index hierarchy are not integrated with the clustering model and so it provides lesser efficiency when compared with the ALB Approach.

6.3.3. Measure of Computational Cost

The computational Cost of ALB Approach in cloud infrastructure is measured, based on the time period required to balance the load and time, required in resolving the server congestion. The Computational Cost (CC) is evaluated as given

$$CC = Time\ Taken\ (LB + SC_D) \qquad \ldots (2)$$

Where LB is time required in load balancing and SC_D denotes the discarding of server congestion. CC is measured in milliseconds (ms).

Table 6.6
Tabulation of Computational Cost

No. of VM	Computational Cost (ms)		
	SRP Model	HVHIH Mechanism	ALB Approach
10	1500	1350	1255
20	1600	1380	1210
30	2165	1750	1590
40	1920	1660	1530
50	1890	1515	1345
60	2620	2315	2085
70	1800	1535	1395

Table 6.6 depicts the computational cost values of ALB Approach in comparison to the existing SRP Model and HVHIH Mechanism.

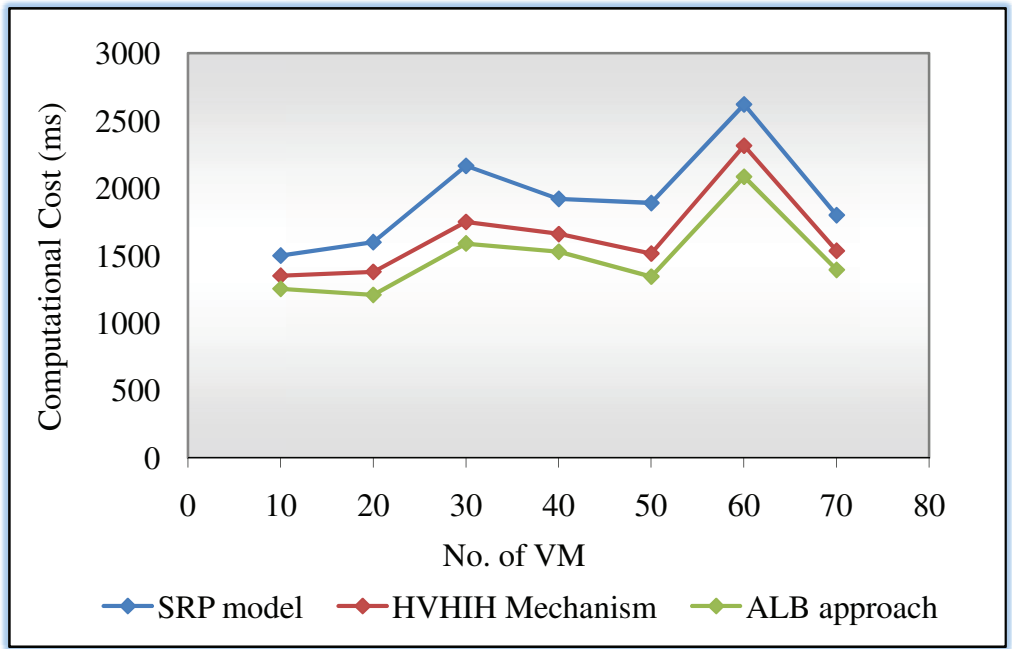

Fig. 6.6: Measure of Computational Cost

Fig 6.6 describes the computational cost of the system, such as SRP Model, HVHIH Mechanism and ALB Approach. Computational cost of ALB is approximately 16-28 % less in SRP Model of Avinash Mehta., et al., (2011) and 7-12% less using HVHIH Mechanism of Yan Zhu., and Shanbiao Wang., (2012) [72]. The computational cost in processing the service request is better than the SRP Model and HVHIH Mechanism as ALB of a frame 'i', is calculated through T f_i, where the maximum processing speeds of all links connecting a frame 'i' in a cloud group is minimized. Moreover, ALB scales

the measures of the available bandwidth in respect to the component $B\ c_i(t)$ and frame $Bf_i(t)$ with the component related to the size of the row minimizing the server congestion.

Finally, the experimental evaluation and computational cost in load balancing factor that is analyzed, offer better results in minimum time for service requests. This is achieved by the enhanced arrangements of the nodes by negotiating the load imbalance as well as the server congestion in cloud infrastructure, based on clustering. Simulation analysis reveals a significant improvement in terms of the computational cost, clustering efficiency and load balance factor.

6.4. ANALYSIS OF GENETIC CLUSTERING METHODOLOGY WITH WORKLOAD MULTI-TASK SCHEDULER

Genetic Clustering with Workload Multi-task (GCWM) Scheduler Scheme uses the genetic initial population and selection, crossover, mutation principles. GCWM Scheduler Scheme is performed to cluster the 'n' tasks for workload management. The multiple task requests from the users through the network link arrive at the server system in cloud environment. GCWM Scheduler Scheme performs clusters based on a similar task, with the aid of fitness function and communicates with each other effectively. Genetic Clustering based Workload Multi-task Scheduling Scheme uses multiple tasks for continuous scheduling at each time interval. GCWM Scheduler Scheme perform experimental evaluation using JAVA CloudSim simulator. CloudSim

simulator executes codes through Command prompt or through CloudSim with Eclipse, Netbeans, etc. supporting an easy task. The specified CloudSim Simulator has been chosen as a simulation platform as it is at present a famous simulation structure in Cloud computing environments. Cloud availability structures at transmission layer carry out optimal analysis, based on the custom configurations supported within the CloudSim. Compared to the simulation toolkits (e.g. SimGrid, CloudSim), JAVA CloudSim offers a copy of on-demand virtualization with enabled bandwidth and submission management.

Experimental machine is simulated with data center comprising 8 GB RAM and 1 TB of storage. The users present are in need of effective communication which is provisioned with a 290 assorted Virtual Machine (VM) pack. Each VM runs a web-application of any kind with variable workload, which is modeled to generate the utilization of the bandwidth, according to the uniformly distributed random variables. GCWM Scheduler uses Statlog (Shuttle) Data Set from UCI repository. The shuttle dataset contains 9 attributes, all of which are numerical. Approximately 80% of the data belongs to class 1. The instances in the actual dataset are in time order, and this time order can most probably be related to clustering. GCWM Scheduler is compared with the Cooperative Provable Data Possession (CPDP) scheme of Yan Zhu., et al., (2012) [71] and Multi-objective Scheduling (MOS) Scheme in Fan Zhanga., et al., (2013) [16]. The experiment is conducted on the

factors, such as generic based cloud services, multi-task clustering effect, computational cost and computational complexity.

6.4.1. Generic Based Cloud Services

Genetic Based Cloud Services denotes the effective cloud services on the system based on the genetic concepts. The cloud services provided are experimented on the CloudSim Simulator and measured in terms of percentage (%). More accurately the influence of generic based cloud services with respect to the cluster size is listed in table 6.7 and a comparison is made with two other existing schemes. Moreover, the generic based cloud services increases with the increase in the cluster size.

Table 6.7
Tabulation for Generic based Cloud Services

Cluster Size (KB)	Generic based Cloud Services (%)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
50	58	62	70
100	60	63	71
150	66	64	72
200	67	65	73
250	69	67	74
300	69	68	75
350	70	69	78

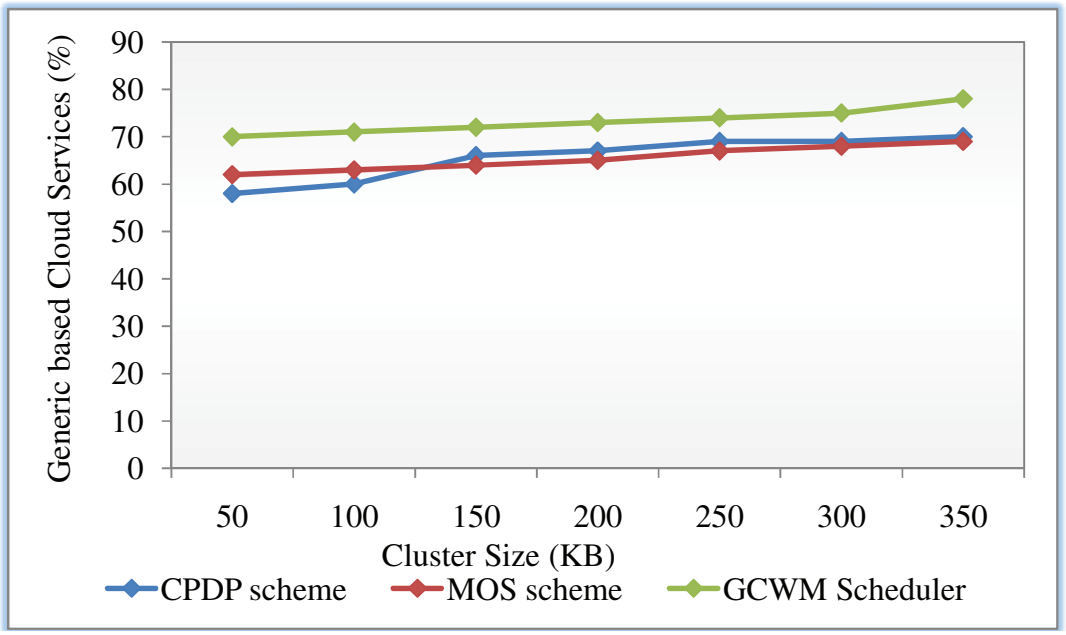

Fig 6.7 Generic Based Cloud Services Measure

Fig 6.7 describes the generic based cloud services, based on the cluster size and they are measured in terms of Kilo Bytes (KB). The initial population of chromosomes in GCWM Scheduler denotes the effective cloud services offered for about 7 – 20 % when compared with the CPDP scheme of Yan Zhu., et al., (2012) [71] and 10 – 13 % better when compared with the MOS scheme of Fan Zhanga., et al., (2013) [16]. All the tasks 'T' in the genetic cluster are assigned to the different physical machines in the cloud platform to serve the user on its effective communication path. The task is represented by a vector of size K and position 'i' represents a task from clients. The task from the client value between [1, n], where 'n' is the number of clusters, indicate effective cloud services.

6.4.2. Measure of Multi-Task Clustering Effect

Multi-task Clustering Effect defines clustering effectively on GCWM Scheduler. The clustering effect follows the initial population, selection, crossover and mutation operator. The results of Multi-Task Clustering Effect are illustrated in table 6.8. Moreover, GCWM Scheduler achieves a higher Multi-task Clustering Effect with an increase in the number of users ranging from 10 to 70.

Table 6.8
Tabulation of Multi-Task Clustering Effect

No. of Users	Multi-Task Clustering Effect (%)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
10	75	88	91
20	79	90	93
30	80	91	95
40	69	82	86
50	73	84	87
60	74	87	92
70	75	88	93

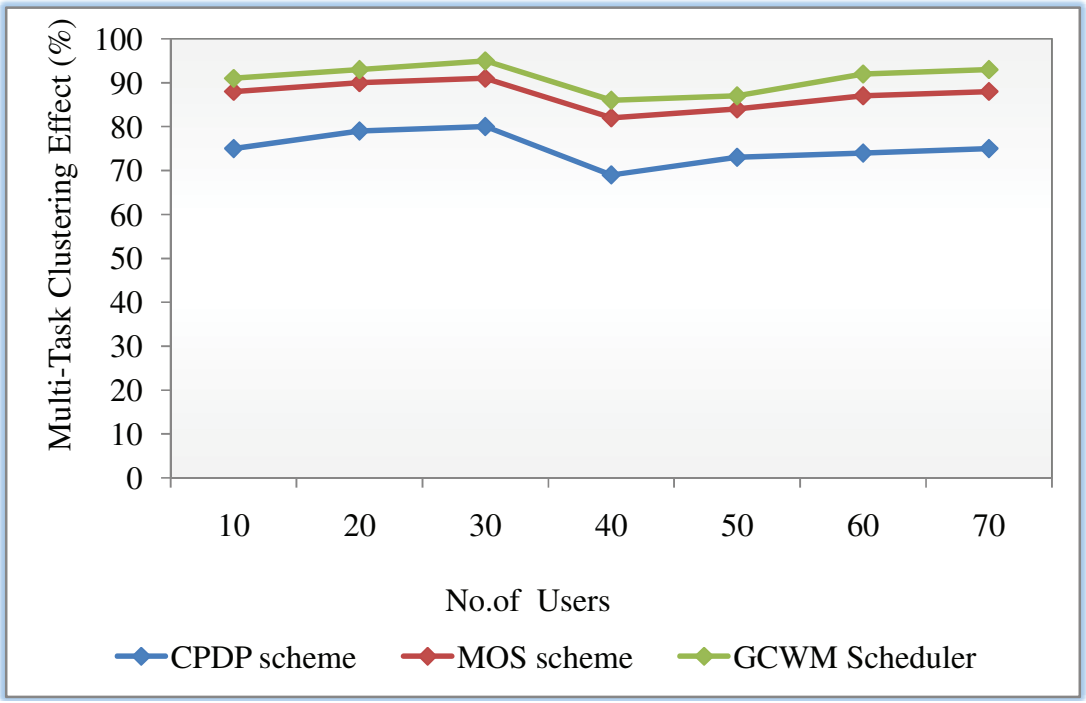

Fig 6.8 Measure of Multi-Task Clustering Effect

Fig 6.8 illustrates the Multi-task Clustering Effect, based on the number of users. GCWM Scheduler acts as a superior in keeping the server system to a perfect schedule using the client task as the best chromosome. The crossover probability 'μ' denotes the count of users in the cluster an improvement by 17 – 25 % when compared with the CPDP Scheme of Yan Zhu., et al., (2012) and 3 – 5 % when compared with the MOS Scheme of Fan Zhanga., et al., (2013). In addition, the crossover operator in genetic clustering uses the 'μ' probability of tasks for better multi-processing of clustering effect, resulting in better clustering efficiency.

6.4.3. Measure of Computational Cost

Computational cost refers to the time period required to perform the multi-task operations in a cloud environment by clustering a similar workload. Computational cost is measured in seconds.

Table 6.9
Tabulation of Computational Cost

No. of Users	Computational Cost (s)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
10	28	22	15
20	37	25	18
30	42	32	22
40	49	38	25
50	55	40	28
60	62	45	32
70	70	50	40

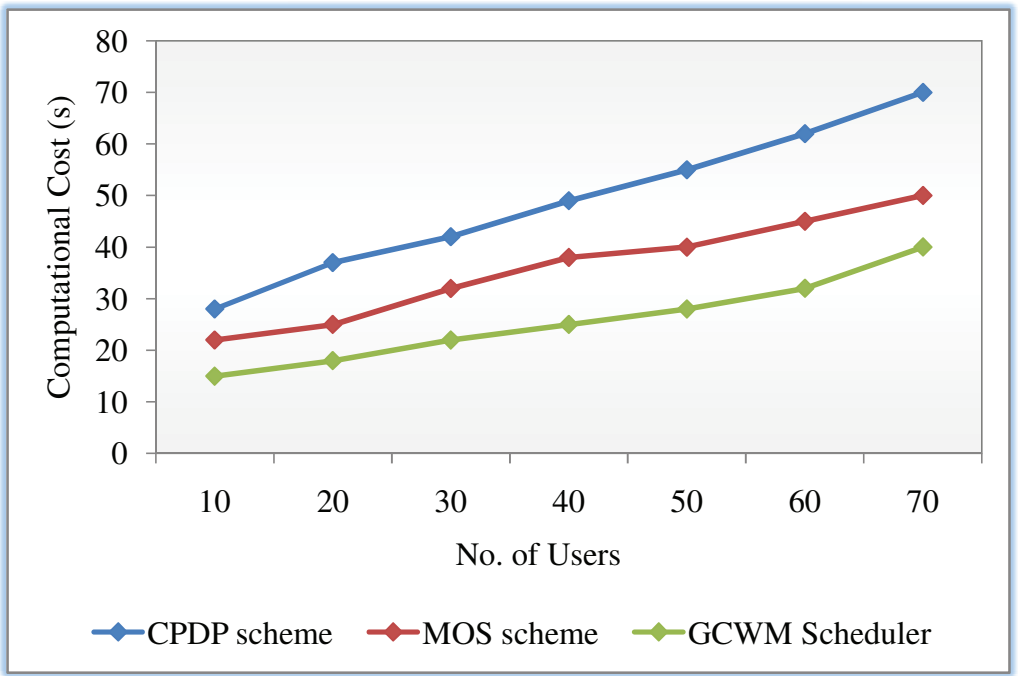

Fig 6.9 Measure of Computational Cost

Table 6.9 and fig 6.9 illustrate the computational cost, involved in Genetic Clustering Based Workload Multi-task Scheduling and the comparison being made with the two other existing schemes in respect to the number of users, ranging from 10 to 70. The results show that the computational cost increases with the increase in the number of users. But comparatively, the computation cost involved in evaluating multi-task scheduling, using the proposed GCWM Scheduler Scheme is low by 42-51% when compared to the CPDP scheme of Yan Zhu., et al., (2012) [71] and 20-34% low when using MOS Scheme of Fan Zhanga., et al., (2013) [16]. This is because the GCWM Scheduler Scheme is based on clustering of a similar workload, using the genetic concepts which minimizes the computational cost.

6.4.4. Measure of Computational Complexity

Complexity in terms of multi-tasking overhead, insufficient resource and overload involved during clustering is referred to as computational complexity. Computational complexity is measured in percentage (%).

Table 6.10
Tabulation of Computational Complexity

Cluster Size (KB)	Computational Complexity (%)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
50	65	70	78
100	69	72	82
150	72	74	84
200	74	76	85
250	77	80	88
300	80	83	90
350	85	85	91

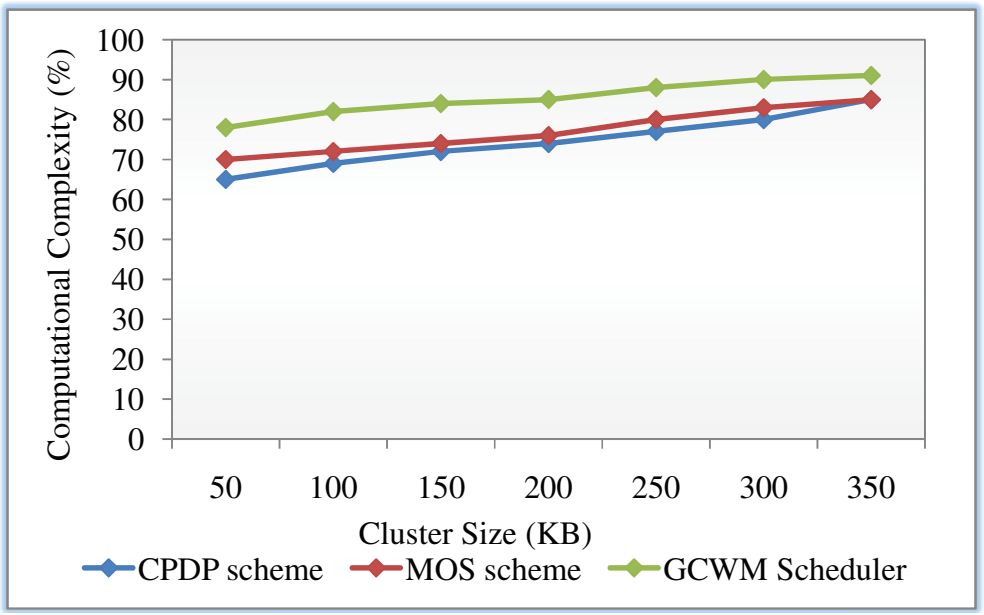

Fig 6.10 Measure of Computational Complexity

Table 6.10 and fig 6.10 illustrate the computational complexity involved with the varying cluster sizes of range 50 to 350 and a comparison has been made with two other existing methods. As illustrated in the figure, with the increase in the cluster size, the computational complexity also gets increased. Comparatively, the computational complexity using the GCWM Scheduler Scheme is less than the other two schemes because the GCWM Scheduler Scheme keeps the server system in such a way that it perfectly schedules the client task as the best chromosome and thereby reduces the computational complexity. The computational complexity is reduced by 12 – 20% when compared to the CPDP scheme of Yan Zhu., et al., (2012) [71] and reduced by 7 – 11% when compared to the MOS Scheme of Fan Zhanga., et al., (2013) [16].

Finally, Genetic Clustering Based Workload Multi-task Scheduling uses the distributed computing resources. The GCWM Scheduler acts superior in keeping the server system to perfectly cluster the client task as the best chromosome. GCWM continuously schedule the clients' tasks and ensures the multi-tasking operation with efficient users' communication by reducing the cost in terms of time involved during computation and the complexities involved in it.

6.5. SUMMARY

Performance analysis on IARA Technique, ALB Approach and GCWM Scheduler Scheme are conducted using JAVA CloudSim Simulator. CloudSim Simulator executes codes as a simulation platform in Cloud computing environments. Moreover, JAVA CloudSim supports copy of on-demand virtualization, permitting better bandwidth utilization and submission management. The centralized proposed systems uses Statlog (Shuttle) Data Set from UCI repository, which contains 9 numerical attributes. The proposed work justify better performances when compared to the existing schemes in cloud environment. The proposed work achieve better results with metrics, like interoperability, computational cost, scalability, load balance factor, clustering efficiency, generic based cloud services, multi-task clustering effect, communication cost and computational complexity. More specifically, IARA Technique reduces the interoperability rate by about 11-27% in interference minimization when compared with the existing methods. Similarly, ALB

Approach provides better load balance factor of about 8 – 25 % in contrast to the other existing schemes. Further, GCWM Scheduler improves the multi-tasking Clustering Effect by 17 – 25 % when compared with other existing works.