CHAPTER 5

WORKLOAD MULTI-TASK SCHEDULER WITH GENETIC CLUSTERING IN CLOUD ENVIRONMENT

5.1. INTRODUCTION

The main issue in cloud environment is the multi-task computing scenario as they contain huge volumes of datasets and loosely coupled tasks. The data possession scheme using the Cooperative Provable Model (CPDP), based on homomorphism provides reliability by automatically maintaining the multiple photocopy of information. More specifically, the CPDPS scheme for large data files needs to resolve the cluster network model for dynamically updating the CPDP metrics. Another scheduling scheme based on the Multi-objective (MOS) Scheme is actively designed and performed using the ordinal optimization (OO) method for clouds. However, MOS also need different memory and disk requirements to balance the workload, while performing multi-tasking.

In order to increase the performance of the workload management during multi-tasking, this proposes work developed Genetic Clustering with Workload Multi-task (GCWM) Scheduler Scheme. The main objective of GCWM scheduler is to cluster the similar workload using genetic principle. Moreover, GCWM Scheduler focuses on minimizing the computational cost and complexities arising during computation. In general, GCWM Scheduler Scheme is applied to cluster 'n' tasks with the initial population (i.e.,) tasks, selection, crossover and mutation operators for the workload management. The fitness function in GCWM Scheduler Scheme Cluster Similar task in cloud zone and communicate with each other successfully. GCWM Scheduling Scheme employes distributed computing resources to test the effectiveness of clustering. GCWM Scheduler guarantees the multi-tasking operation with an efficient user communication. GCWM Scheduler Scheme performs an experimental evaluation using JAVA CloudSim simulator, while GCWM Scheduler uses Statlog (Shuttle) data, set from UCI repository. The shuttle dataset contains 9 attributes, all of which are numerical. The performance of GCWM Scheduler Scheme in measured on factors like throughput, workload management efficacy, and relative cost.

5.2. MULTI-TASKING IN CLOUD ENVIRONMENT

Cloud computing is efficient to carry the users process to the offered system resources, like storage and software. Cloud computing holds a solution check for resource arrangement in a real-time environment. In such virtualized environments, both the Virtual Machine (VM) and hosted applications needs to be configured on-the-fly to adjust the system dynamics. Moreover, the main essential part is to coordinately point all the physical servers concerned in cloud areas to evaluate the independent optimization components.

The progress of the service providers in the cloud environment, arranges the cognizance with every client deals to the cloud has varied with the increase in the infrastructure layer, release of software and development models. The importance of cloud computing has been imagined from different places including grid, autonomic and efficacy computing and changed into an innovative design model. With reliable design, architectural model and feature issues in cloud computing, there requires a quantity of security, consisting of centralized security, optimal scheduling of data and processes with higher accessibility rate. However, the cloud supports the different advantages of the users, while a lot of the expected risks are calculated proficiently due to the infrastructures significant characteristics.

Multi-tasking denotes the sharing of resources in cloud environment. Various features of the information system are used, which commonly include memory, programs, networks and data. Cloud computing is based on a framework design in which the resources are used commonly. The resources are shared with the multiple users, which are comparable resources at the system level, host level, and application level. Although users are present in the remote area at a virtual level, the hardware is not divided. With a professional multi-tasking design model, a software application is forecasted to virtually partition its data. In addition, the configuration is also separated as each client network related computational resource organization works with a customized virtual application instance.

5.2.1. Scheduling of Multi-tasking Workload

Scheduling of Multi-tasking Workload for large data measures is a testing concern, as it needs a significant quantity of comprehensive metric and iterations. Hence, real-time scheduling is necessary with an increase in the throughput of multi-task scheduling. The complexity depends on attaining a sequence of the best, thus far the reactive schedules. In active principles, such as virtual clusters in cloud, the scheduling must be tasked with rapid sufficiency to remain in speed with the random fluctuations the workloads to optimize the complete system performance. Multi-tasking schedules the task class as a group process that is of similar kind and is implemented parallelly.

Haiyan Guan., et al., (2013) [23] discuss that the resources required for computation in the network-related model for cloud computing environment uses an appropriate process virtualization platform termed, Condor-based Process Virtualization Platform. Condor-based process mainly deals with vast quantities of huge data. However, the virtualization model fails to support a parallel distributed network based processing workflow in cloud computing environments. The reason behind the failure is that there are some break-lines in the middle of the filtering results of the adjacent blocks. A post-adjustment essentially removes those sharp break-lines according to the accuracy required by the users. Therefore, suitable virtualization needs an optimal workload management, specifically in multi-tasking.

In general, the clustering model is also helpful in performing many-tasks. Basically, the task class behaves as a set of tasks that are of a similar kind and performs at the same time. Considering a specific number of task classes in all, the index is traced. A task within one task class is either mutually dependent, such as technical workflow or self-governing of each other, such as systematic simulations with various metrics. Tasks across the different task classes are independent. The conceptual view of multi-tasking in cloud is elaborated in fig. 5.1.

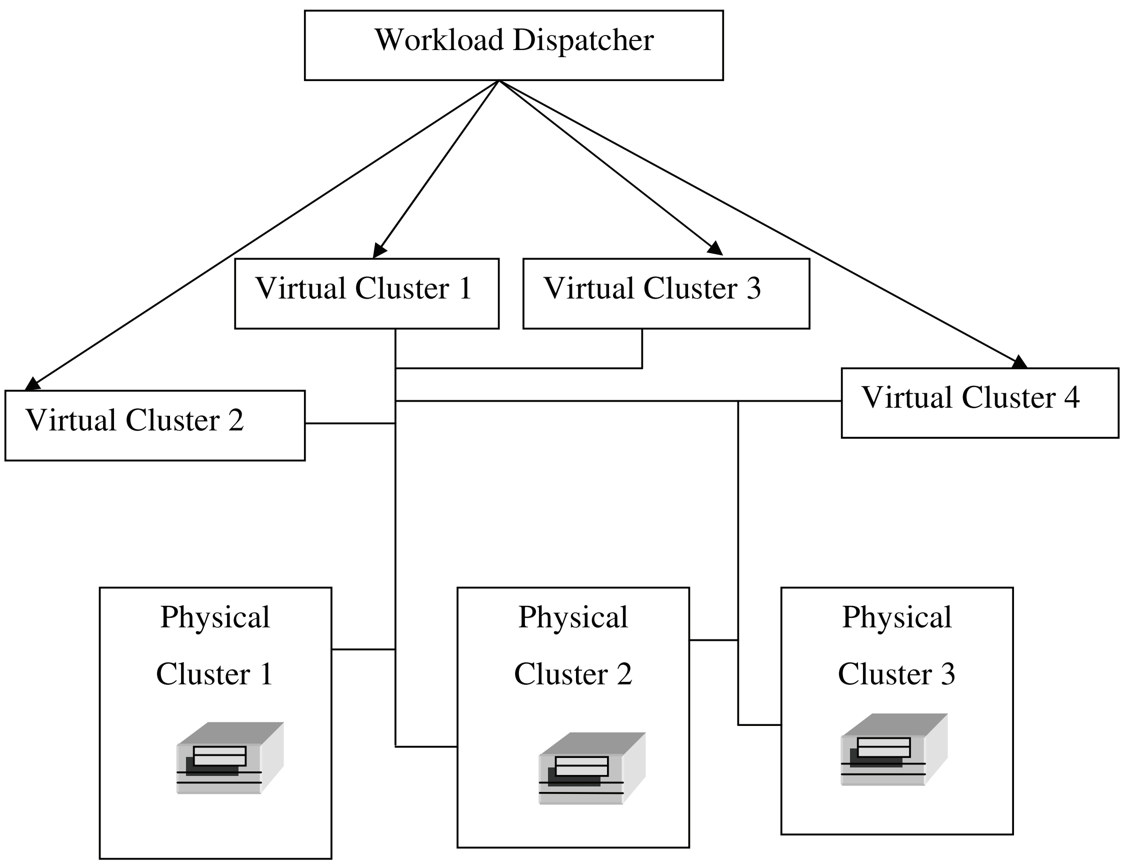

Fig. 5.1 Conceptual View of Multi-tasking in Cloud with Virtual Cluster Resource Allocation

Multi-tasking in cloud environment is handled with the help of virtual machines in virtual cluster as shown in Fig. 5.1. Moreover, fig. 5.1 clearly depicts the conceptual view of a virtual cluster resource allocation model for workload execution in a virtualized cloud platform. In general, physical clusters with each cluster groups form the VMs. The VMs are partitioned into virtual clusters, based on cloud structure. The workload dispatcher disperses the class task to each virtual cluster to balance the load.

Yan Zhu., et al., (2012) [70] discuss that most of the methods focus on multi-tasking based workload management. One such method is Cooperative Provable Data Possession (CPDP) Scheme, based on homomorphism demonstrable response and hash index hierarchy. CPDP proves the security of the scheme, based on multi-prover zero-knowledge proof system. Moreover, the scheme CPDP provides finiteness, knowledge-specific and optimal parameters with zero-knowledge system. However, CPDP is unable to support the high throughput due to lower workload management issues. Fan Zhanga., et al., (2013) [16] analyse that, the scheduling scheme for multi-objective (MOS) is specifically designed, based on the Ordinal Optimization (OO) method for clouds. But, MOS exploits most of memory and disk spaces, causing high computational cost and high relative cost.

5.3. GENETIC CLUSTERING METHODOLOGY WITH WORKLOAD MULTI-TASK SCHEDULER

Genetic Clustering with Workload Multi-task (GCWM) Scheduler Scheme uses the genetic initial population, selection, crossover and mutation concepts. GCWM Scheduler Scheme is applied to cluster the ‘n’ tasks for workload management. The multiple task requests from the client through the network connection reach the server system in cloud environment. GCWM Scheduler Scheme employs a fitness function to cluster a similar task in the cloud zone and communicates with each other effectively. Genetic Clustering Based Workload Multi-task Scheduling Scheme uses multiple tasks for the continuous scheduling at each time interval. The architecture diagram of Efficient User communication cloud computing using Genetic Clustering with Workload Multi-task (GCWM) Scheduler Scheme is demonstrated in Fig 5.2.

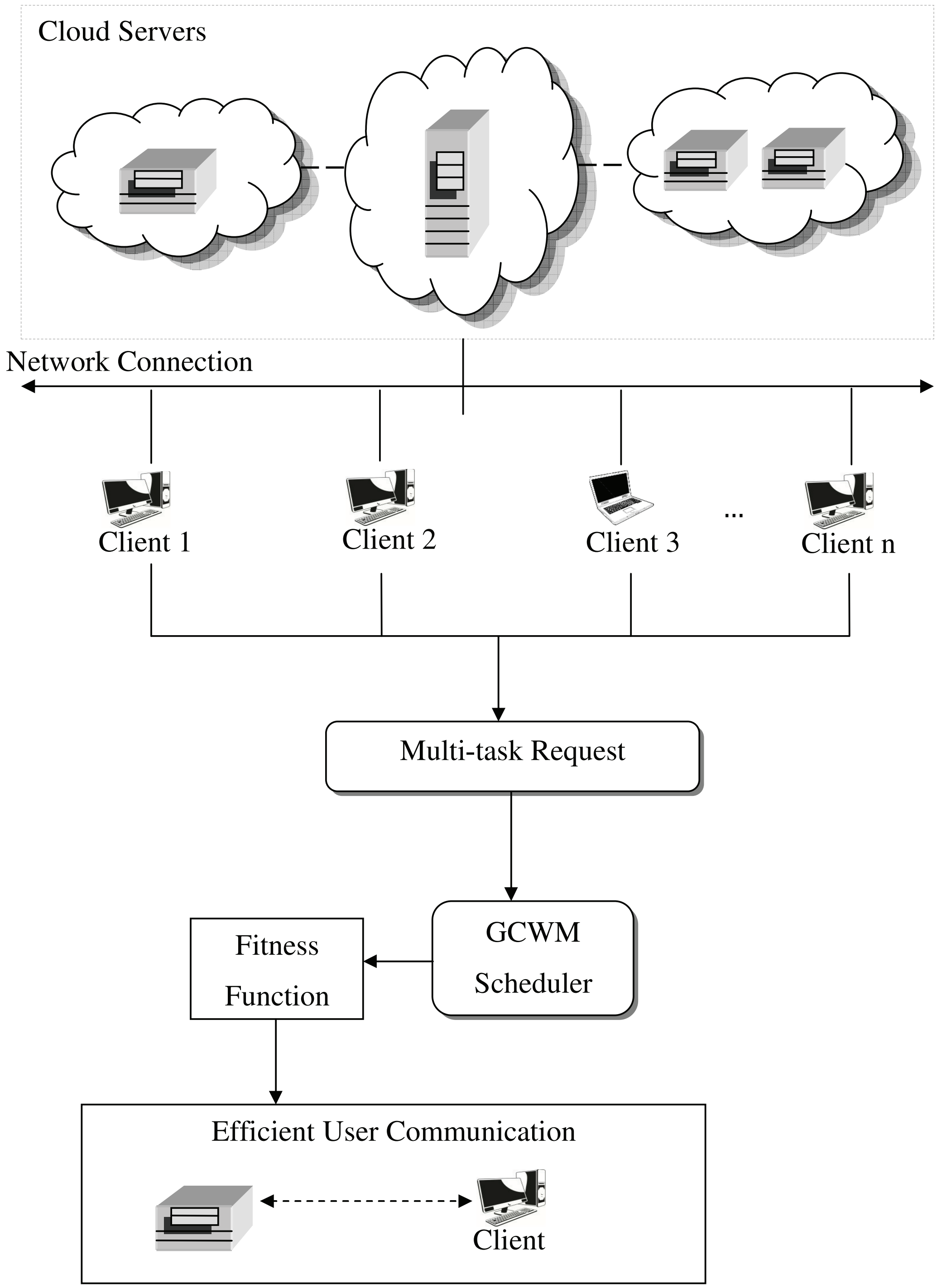

Fig. 5.2: Architecture of Efficient User Communication Cloud Using GCWM Scheduler Scheme

Fig 5.2 describes the workload balance using the Multi-task scheduler with genetic clustering. GCWM Scheduler Scheme executes fast scheduling of the multiple tasks of the clients. With the logic of the genetic clustering principle, the server implements different client task as requested by the clients successfully. In order to achieve dynamic scheduling, the client 1, 2, 3,…,n sends multiple tasks request. On the other hand, GCWM Scheduler schedules the task using the cluster network model. The main communication results in success by adopting the fitness function into GCWM Scheduler Scheme. The fitness function of each client task is forecasted against a threshold value. Moreover, the threshold value in GCWM scheduler scheme is measured, based on the genetic concepts of clustering. If the threshold value is not reached to the specified point, then the client task will be assigned to different clusters.

GCWM Scheduler Scheme mainly concentrates on initial population, fitness computation, selection, crossover, and mutation. These metrics are performed for a maximum number of generations to successfully schedule the multiple tasks. In cloud services, GCWM Scheduler Scheme reduces the computational complexity using the dynamic scheduling for the multiple tasks from the clients. The real time development of GCWM Scheduler Scheme is optimal at every stage. In addition, the overall performance is enhanced and it is justified with the help of experimental results.

5.3.1. Workload Partition in Cloud Services Using Genetic Clustering Process

The process of genetic clustering in cloud services begins with the initial partition of the workload on the client workload, based on the server resources. The key objective of the genetic concept is to offer an initial representation of task from the clients; Selection, crossover, and mutation operator are carried out in GCWM Scheduler for effective communication.

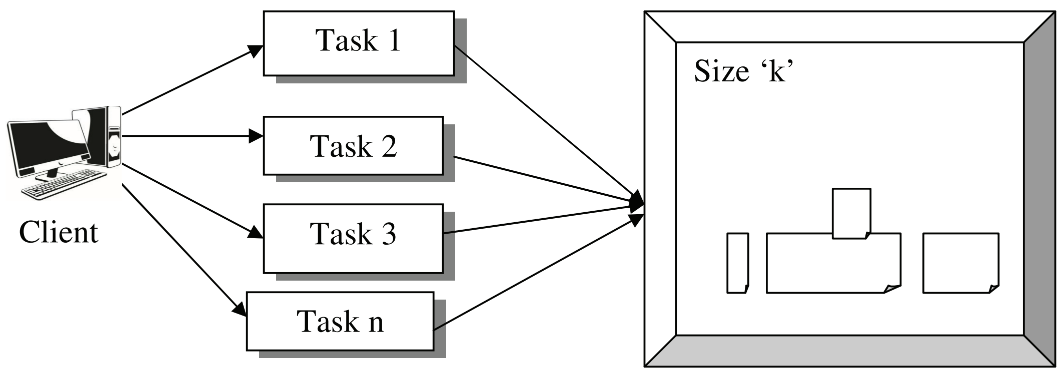

Fig 5.3 Client Multi-task Representation in Cloud

The representation of obtaining the multiple tasks from the clients is depicted in Fig 5.3 with a help of the task number. Depending on the multiple tasks assigned, the scheduler represents the task by a vector of size K, where each position in 'i' represents a task from clients. The task from the clients is valued between [1, 2,3,.., n], where 'n' is the number of clusters, representing which genetic cluster with position 'i' belong to. The initial population of the chromosomes in GCWMS Scheduler Scheme indicates the initial task requested from the client to the server system with the help of network links.

Each chromosome in the population (i.e.,) cluster group is indicated by an ‘n’ cluster, which is occupied by ‘K’ vector size.

The selection method used in GCWM Scheduler Scheme is the process of selecting the fitter. The fitter denotes the client task chromosomes to execute the cluster operation using genetic operators. GCWM Scheduler Scheme manages the server system in a better way that the server perfectly schedules the client task. The better server scheduling is achieved with the optimal chromosome, reducing the computational complexity. The crossover operator in genetic clustering uses the ‘μ’ probability of tasks for better processing. The crossover probability ‘μ’ denotes the count of tasks in the cluster. As the count enhances in the genetic cluster group, effective development of multi-task scheduler is provided using GCWM Scheduler Scheme.

The mutation operator in GCWM Scheduler Scheme uses consistent mutation to select a task that travels randomly from the client to the server system, based on the request time priority. Each chromosome performs mutation with a probability of ‘μ’ seconds. The ‘μ’ seconds is used to forecast the request time between the different client systems and the server in cloud zone. The fitness function in GCWM Scheduler uses the process of selection, crossover and mutation for a maximum number of generations for effective communication in the cloud services.

The fitness function for each client task is measured with the aid of a threshold value. The threshold value used in GCWM Scheduler Scheme uses the genetic logics, based on clustering. If the threshold value is not reached to the specified point, then the client task is moved to the different clusters. The genetic cluster is then updated to the mean threshold points of the respective clusters. As a result, in genetic clustering 'T' is measured for each chromosome (i.e.,) task, as T=0. For n = 1 to 'N', the genetic clustering ideas are used for different groups of clusters to execute the specific task of the client. For all the tasks from the clients,

$$\mu \ = T + \ \mu seconds \qquad \ldots (1)$$

The count of client tasks in the genetic cluster 'μ' is evaluated. Cluster 'μ' is the sum of T that denotes the tasks requested to the server system in cloud environment. In addition, 'μseconds' represents the request time of the different client system to the server in cloud zone. The optimal clustering is based on the genetic operations performed till the last generation i.e., request from all clients attains the solution. GCWM Scheduler Scheme, at each request offers an optimal result with the genetic cluster center. The genetic cluster center contains the complete cluster label, named for easy access of the resources, even when the multi-task is scheduled up in the queue.

5.3.2. Multi-task Scheduler in Workload Balancing

Once the genetic clustering task is completed, the process of workload multi-task scheduler is carried out. GCWM Scheduler Scheme adopts a dynamic multitasking schedule in order to define multi-tasking as a set of tasks. A set of tasks of the same type is considered in the genetic cluster group and it performs in parallel. After considering the 'T' task in all the genetic clusters, the index should be measured. The index in genetic cluster is denoted as t, i.e., $t \in [1, T]$.

A workload is scheduled dynamically in GCWM Scheduler Scheme. Moreover, the load scheduling is provisioned at continuous time periods, without any elapsed time. The scheduler 'i' with $\theta\ (q_i)$ is indicated in the genetic cluster with the schedule space 'S'. The schedule in 'n' cluster is represented as,

$$\theta\ (q_i) = \{\theta_1(q_i), \theta_2(q_i)\ \theta_3(q_i) \ldots \ldots \theta_n(q_i)\} \qquad \ldots (2)$$

Where, $\theta_n(q_i)$ is the number of genetic cluster 'θ_n' with the q_i tasks for processing in cloud zone. The task 'q_i' at time 't' is the execution time for specific task in the GCWM Scheduler Scheme. The schedule space 'S' is the space where the multi-task is implemented on the provided server, resulting in effective scheduling. The dynamic scheduling of multi-task for each client in cloud using GCWM Scheduler Scheme is shown in Fig 5.4.

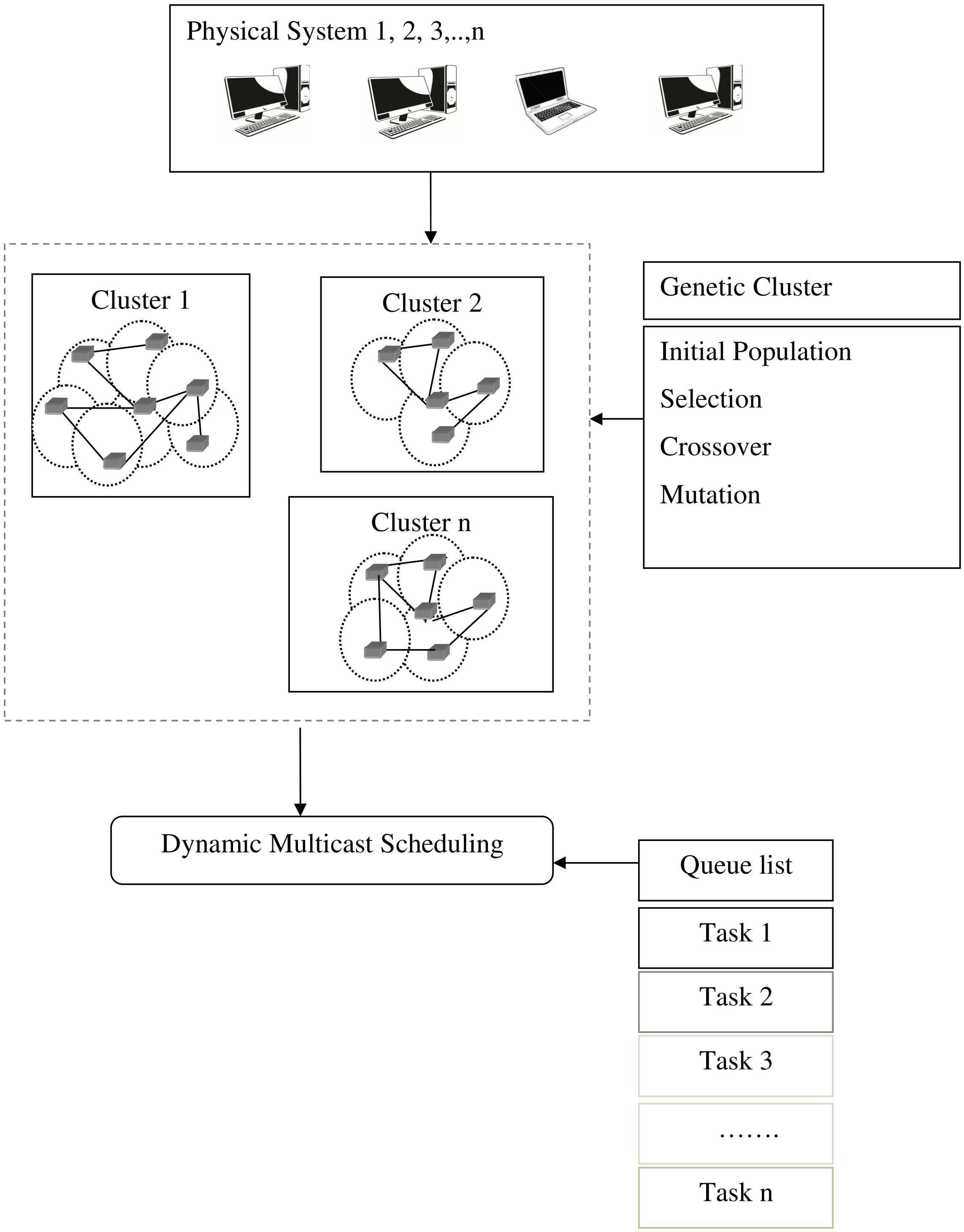

Fig 5.4 Clustering Operation in GCWM Scheduler Scheme

The server system in the cloud zone schedules the multi-task effectively and measures the units which are built on the top of the physical machines. A permanent set of client systems are offered in GCWM Scheduler Scheme, so that the scheduling method can organize them into 'n' clusters. The cluster formation in each genetic cluster holds several tasks with similar threshold values. All the tasks 'T' in the genetic cluster are assigned to the different physical machines in the cloud platform so as to prepare the user for an effective communication path. The dynamic scheduling in GCWM follows the queue list, where the queue is arranged, based on the client request time.

5.3.3. Algorithmic Steps in GCWM Scheduler

The GCWM Scheduler accepts the resources and satisfies the appropriate requests with minimum energy utilization. Moreover, GCWM Scheduler Scheme takes the maximum number of generations for dynamic scheduling of the multiple tasks from the client environments. The algorithmic description of GCWM Scheduler Scheme provided below describes the step-wise description.

Input: $'T'$ tasks, 'μ' seconds request time of the different client system to the server in cloud zone, 'n' number of task in cluster group, 'K' vector size of the task.

Begin

// Initial Task Request

Step 1: T ← 0, set the initial population with vector size 'K'

Step 2: Evaluate Initial Task Request in cloud.

// Selection Operator

Step 2: Client task chromosomes perform cluster operation

Step 3: Perfectly schedule the client multiple task as the best chromosome.

//Crossover Operation

Step 4: 'μ' probability of task in cluster

Step 5: Effective development of multi-task scheduler

//Mutation Operator

Step 6: Selects a task that moves randomly using Uniform mutation

Step 7: Chromosome undergoes mutation with a 'μseconds' probability

Step 8: Compute the request time of the different client system to server

// Fitness Function

Step 9: If (Specified threshold value attained)

Step 10: Assigned to particular cluster group

Step 11: Else

Step 12: Client task is assigned to different clusters

// Dynamic Scheduling

Step 13: Queue list with multiple task follow genetic cluster concept

End

Output: Best communication to the cloud users

The algorithmic step in GCWM Scheduler performs the initial task request, selection, crossover mutation operator, fitness function and dynamic scheduling more elegantly. The algorithmic step describes briefly the genetic cluster with multiple task workload scheduling on the cloud zone for improved throughput. The user communication is made robust with genetic cluster network model development. At last, dynamic Scheduling and multi-tasking workloads in cloud computing issues are addressed.

5.4. EXPERIMENTAL EVALUATION

Genetic Clustering with Workload Multi-task (GCWM) Scheduler Scheme performs an experimental evaluation using JAVA CloudSim Simulator. CloudSim Simulator executes codes through the 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 at present it is a famous simulation structure in Cloud computing environments. Cloud availability structures at transmission layer carry out an 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, enabled with bandwidth and submission management.

An experimental machine is simulated with data center comprising 8 GB RAM and 1 TB of storage. The users present need for effective communication, which is provisioned with a 290 assorted Virtual Machine (VM) pack. Each VM runs a web-application of different kind with variable workload, which is modeled to generate the utilization of the bandwidth according to the uniformly distributed random variable.

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 a in time order, and this time order can most probably be related to clustering. GCWM Scheduler is compared against the Cooperative Provable Data Possession (CPDP) Scheme of Yan Zhu., et al., (2012) [70] and Multi-Objective Scheduling (MOS) Scheme of Fan Zhanga., et al., (2013) [16]. The experiment is conducted on the factors, such as throughput, workload management efficacy and relative cost.

5.5. RESULT ANALYSIS

In the result analysis section, GCWM Scheduler is compared against the existing Cooperative Provable Data Possession (CPDP) Scheme of Yan Zhu., et al., (2012) [70] and Multi-objective Scheduling (MOS) Scheme of Fan Zhanga., et al., (2013) [16]. The evaluation value given below through table and graph describes the GCWM Scheduler in cloud infrastructure so as to enhance the communication level.

5.5.1. Throughput

Throughput is the rate of a successful message delivery over a communication channel in cloud infrastructure. GCWM Scheduler throughput is formalized as,

$$Throughput = Threshold\ value + \left(\frac{M}{T}\right) \qquad \ldots (3)$$

Throughput is described as the rate at which a cloud system generates and produces services per unit of time. 'M' denotes the number of machines used in the cloud infrastructure and 'T' denotes the time taken to perform the process. The threshold value is measured to be "70" in the GCWM Scheduler, for computation.

Table 5.1
Tabulation of Throughput

No. of Machines	Throughput (tasks/sec)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
5	58	65	72.5
10	59	66	73.33
15	60	67	73
20	62	69	75
25	63	70	74.11
30	65	71	76
35	66	72	77.5

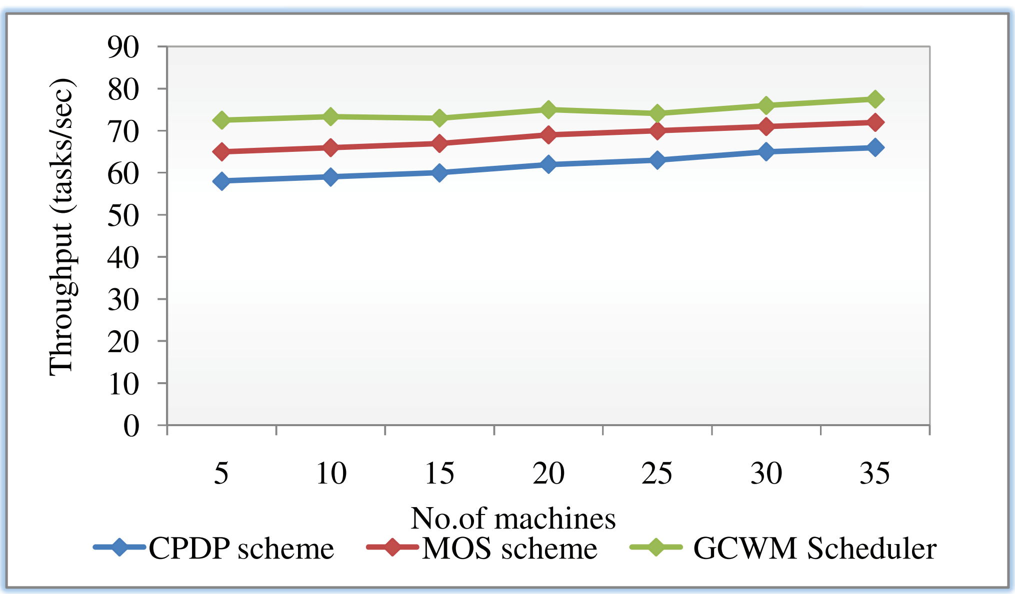

Fig 5.5 Measure of Throughput

Fig 5.5 describes the throughput based on the machine count in the cloud infrastructure. In genetic clustering, task ‘T’ is calculated for each chromosome starting from T=0. The fitness function of each client task is measured with the threshold value. Genetic cluster is updated with the mean threshold points of the respective clusters for better result. Accordingly, the throughput value is improved by 16 – 25 % in GCWM Scheduler when compared with the CPDP Scheme of Yan Zhu., et al., (2012) [70]. Similarly, GCWM Scheduler throughput is improved by 5 – 11 % when compared with the MOS Scheme of Fan Zhanga., et al., (2013) [16]. Moreover, GCWM Scheduler Scheme at each request provides the best result with the genetic cluster center to increase the throughput. The genetic cluster center involves the complete cluster label name for easy access of resources, even when the multi-task is scheduled up in the queue, enhancing the throughput results.

5.5.2. Measure of Workload Management Efficiency

Multiple-workloads are managed in the GCWM Scheduler, where the amount of work that the computer processor in the cloud has been given to do at a particular time. Multiple workloads share the resources and complete the work task effectively. The workload management efficiency is measured in terms of percentage (%).

Table 5.2
Tabulation for Workload Management Efficiency

Load Size (KB)	Workload Management Efficiency (%)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
25	74	83	90
50	75	84	89
75	76	86	90
100	79	87	91
125	82	89	92
150	83	89	94
175	83	89	95
200	84	89	96

The Workload Management Efficiency of GCWM scheduler is compared with the existing two schemes, namely Cooperative Provable Data Possession (CPDP) scheme of Yan Zhu., et al., (2012) [70] and Multi-objective Scheduling (MOS) Scheme of Fan Zhanga., et al., (2013) [16] is provided in table 5.2.

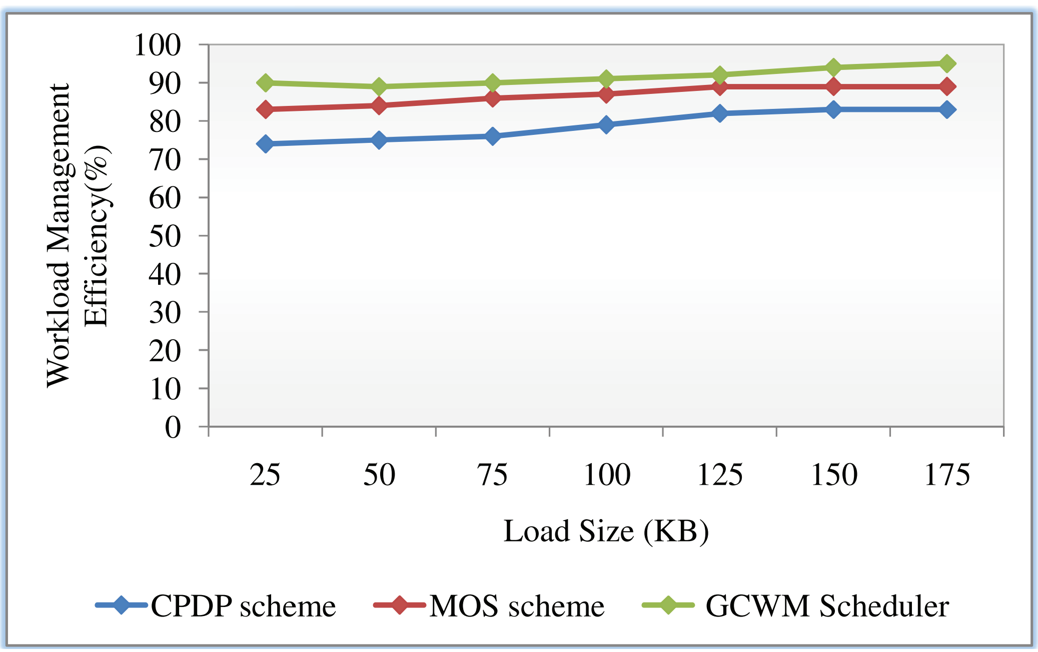

Fig 5.6 Workload Management Efficiency Measure

The real time development of GCWM Scheduler is optimal at every stage and the overall performance is enhanced as demonstrated in fig 5.6. GCWM Scheduler has an improved 12 – 21% workload management when compared with the CPDP Scheme of Yan Zhu., et al., (2012) [70]. Moreover, GCWM Scheduler improves the workload management using the dynamic scheduling of the tasks from the clients, increasing it by 3 – 7 % more than MOS Scheme of Fan Zhanga., et al, (2013) [16]. The main reason behind the better workload management in GCWM Scheduler is the initial task request handled by the initial population, the fitter selection to perform cluster the better processing by the crossover operator and finally the task travel management by the mutation operator. Additionally, the fitness function in GCWM Scheduler uses the process of selection, crossover and mutation for a maximum number of generations for the workload management in an effective communication in support of the cloud services.

5.5.3. Average Relative Cost

Average Relative cost is defined as the cross product of the time taken to execute per task and the task quantity count number. The average relative cost is measured as

$$Average\ relative\ cost = Time\ taken\ to\ execute\ per\ task * Task\ Count\ number \qquad \ldots (4)$$

The minimum the measure of the relative cost, the higher the performance of the cloud service. The average cost count is measured in terms of milliseconds.

Table 5.3
Average Relative Cost Tabulation

Task Count	Average Relative Cost (ms)		
	CPDP Scheme	MOS Scheme	GCWM Scheduler
2	125	112	102
4	457	429	396
6	1025	952	880
8	1783	1627	1550
10	2633	2573	2400
12	3974	3633	3540
14	5211	4915	4800

The average relative cost of our scheme and comparison made with two other existing schemes namely, Cooperative Provable Data Possession (CPDP) scheme and Multi-objective scheduling (MOS) Scheme is listed in table 5.3.

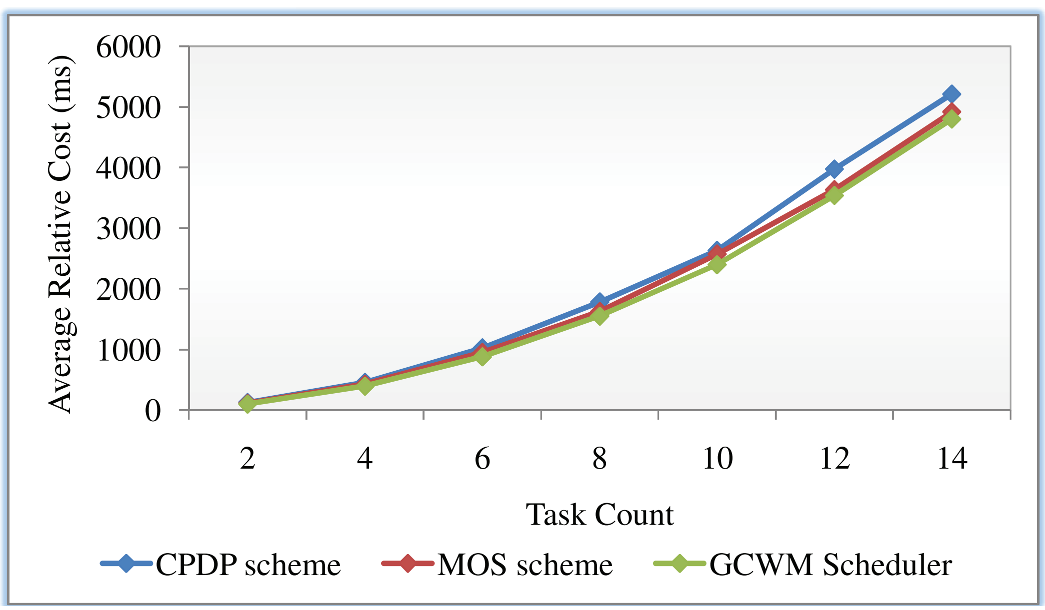

Fig 5.7 Average Relative Cost Measure

Fig 5.7 describes the average relative cost, based on the task count. The task count is computed with the equation (4). The schedule with 'n' cluster reduces 7 – 18 % of the average relative cost when compared with the CPDP Scheme of Yan Zhu., et al., (2012) [70]. The workload is scheduled dynamically in GCWM Scheduler and it is provisioned at continuous time periods avoiding the elapsed time supporting the lower relative cost. Moreover, the scheduler 'i' with θ (q_i) is represented in the genetic cluster with the schedule space 'S'. The dynamic scheduling in GCWM Scheduler follows the queue list where the queue is arranged, based on the client request time and thereby reducing the cost by 2 - 8% when compared with MOS scheme of Fan Zhanga., et al., (2013) [16].

At last, GCWM scheduling exploites the distributed computing resources. The GCWM Scheduler behaves as if the advantage is in keeping the server system to completely cluster the client process as the optimal chromosome. GCWM constantly schedules the clients' tasks and promises the multi-tasking operation with efficient users' communication by reducing the relative cost in terms of time involved during computation.

5.6. SUMMARY

Genetic Clustering with Workload Multi-task Scheduler resolves the issues and viability in managing the computing cluster. The clustering holds an indefinite number of tasks which justify that the multi-cloud execution of a computing cluster is feasible in terms of scalability. In addition, cluster evaluation efficacy is enhanced by GCWM Scheduler, where the fitness function clusters the identical process in the cloud zone and communicates with each other effectively. GCWM Scheduler schedules the workload dynamically, which is provisioned at successive time periods, without any elapsed time. The schedule space performs multi-tasking on the provided server, which results in effective scheduling of GCWM by reducing the computational cost and minimizes the complexities where the server completely schedules the client task with the optimal chromosome. The fact proves that the GCWM Scheduler implementation of a computing cluster is viable from the throughput view point and workload management efficiency. CloudSim Simulator is used to prove better performance of GCWM Scheduler in terms of 5-25% improved throughput, 3-21% better workload management efficacy and 2-18% reduced relative cost.