ISSN: 0976-3104
SPECIAL ISSUE: Computer Science

ARTICLE

TOWARDS OPTIMAL ALLOCATION OF RESOURCES IN CLOUD MODIFIED MAPREDUCE USING GENETIC ALGORITHM

D. Arivudainambi[1], D. Dhanya[2*]

[1]Department of Mathematics, Anna University, Chennai, INDIA

[2]Research Scholar, Anna University, Chennai, INDIA

ABSTRACT

Background: *Cloud computing is one among the unfathomable knowledge exploration in the field of Information Technology. A virgin as well as a task emerging area of study, cloud computing provides an innovative model for the distribution of multiple applications in various organizations.* ***Methods:*** *The development and growing popularity of cloud computing indicates the evolution in the way IT infrastructure and services are distributed and expended. This increased use of cloud computing resulted in resource problems which have to be solved for better usage of the clouds. In this paper, we present an efficient method for solving resource problems in the cloud using modified map reducing algorithm.* ***Results:*** *Here we employed a Map Reduce programming model with GA for distributed parallel computing and execution on a virtualization process which is used to detect non-sufficient reductions in the execution time and to detect the decrease in the computing time.* ***Conclusions:*** *GA is compared with existing methods such as Clustering large applications(CLARA), Partioning around medoids(PAM), Clustering large applications based on randomized search(CLARANS) in which the simulation results of GA aim on improving the solutions both in execution and computation.*

KEY WORDS
Cloud computing, MapReduce, resource allocation, genetic algorithm, clustering algorithms.

Received: 18 March 2017
Accepted: 30 May 2017
Published: 5 July 2017

***Corresponding Author**
Email:
dhanya9870@gmail.com

INTRODUCTION

Cloud computing is a large scale distributed computing paradigm in which a pool of computing resources are available to the users via the Internet [1]. Cloud computing is a metaphor used by technology or IT Services companies for the delivery of computing requirements as a service to a homogeneous community of end-recipients. Recently, the cloud computing has been gaining lots of interests from researchers and IT industry due to its many abilities to provide flexible, dynamic IT infrastructures, QoS guaranteed computing environments, configurable software services and the state-of-the-art enabling cloud computing technologies [11]. Cloud computing clusters provide a large scale computing environment for scientific users. However, a large scale biological application often involves various types of computational tasks which can benefit from different types of computing clusters [9]. The use of cloud computing technology has gained popularity in recent years, and many companies are currently moving their business to the cloud, by deploying their services and executing their workloads in private or public clouds according to the application requirements or the particular business models [2].

In recent years, the application of high performance and distributed computing in scientific practice has become increasingly widespread. Among the most widely available platforms to scientists are clusters, grids, and cloud systems [10]. A storage cloud provides storage services, while a compute cloud provides computer services [13] It is important to note that we are implicitly considering the possibility of a hybrid deployment, i.e., the resources can be placed in different clouds, this multi-cloud setup can be suitable for deployment of independent virtual resources or for loosely-coupled multi-component services with no or weak communication requirements [3]. Data analytics play a key role in planning, problem solving, and decision support tasks. Data analytics applications typically process large amounts of data from both operational and historical data sources and the processing is primarily read-only with occasional batch inserts [5].

Scientific computing is a field of study that applies computer science to solve typical scientific problems. It is usually associated with large scale computer modeling and simulation and often requires a large amount of computer resource [7]. Providers such as Amazon, Google, Sales force, IBM, Microsoft, and Sun microsystems have begun to establish new data centers for hosting cloud computing applications in various locations around the world to provide redundancy and ensure reliability in case of site failures [4]. Many task computing (MTC) is novel paradigm, which tries to solve the problem of executing multiple parallel activities in multiple processors. Multiple processors may refer to a large cluster or a cloud [12]. The cloud storage service (CSS) relieves the burden of storage management and maintenance. However, if such an important service is vulnerable to attacks or failures, it would bring irretrievable losses to users since their data or archives are stored into an uncertain storage pool outside the enterprises [6].

Map Reduce is a programming model used for processing large data sets in a highly-parallel way. Users specify the computation in terms of a “map” function that processes a key/value pair to generate a set of intermediate key/value pairs, and a “reduce” function that merges all intermediate values associated with the same intermediate key [18]. It is one of the most popular programming models designed to support the development of such applications [16]. Map-Reduce is attracting a lot of attention, proving both a source for inspiration as well as the target of polemic by prominent researchers in databases. In database terms, map reduce is a simple yet powerful execution engine, which can be complemented with other data

IIOAB JOURNAL
ISSN: 0976-3104

storage and management components as necessary [14]. The map-reduce paradigm has been drawing the attention of the computational biology community particularly in the last year. However, majority of these applications are limited to simply distributing the data which are then handled by existing sequential software [20].

Hadoop is a popular open source cloud computing framework that has shown to perform well in various usage scenarios. Its Map reduce framework that offers trans-parent distribution of compute tasks and data with optimized data locality and task level fault tolerance; its distributed file system (DFS) offers a single global interface to access data from anywhere with data replication for fault tolerance [15]. On the other hand, data analysts in most companies, research institutes, and government agencies have no luxury to access large private Hadoop/ Map Reduce clouds. Therefore, running Hadoop/ Map Reduce on top of the public cloud has become a realistic option for most users [19]. One of the biggest merits of cloud storage is that users can access data in a cloud anytime and anywhere, using any device [8]. The simplicity of map reduce, its wide-spread usage, and its ability is in capturing the primary challenges of developing distributed applications [17].

In [21] Semantic web an emerging area to augment human reasoning has been proposed. Various technologies were being developed in those arenas which have been standardized by the World Wide Web Consortium (W3C). One such standard was the Resource Description Framework (RDF). Semantic web technologies can be utilized to build efficient and scalable systems for cloud computing. Hadoop's mapreduce framework is used to answer the queries. The results show that they can store large RDF graphs in hadoop clusters built with cheap commodity class hardware. Furthermore, their framework was scalable and efficient and can handle large amounts of RDF data, unlike traditional approaches.

In [8] storage in the cloud is proposed with the SPKS scheme for cloud storage services. It allows the CSP to participate in the decipherment, so that the user could pay less computational overhead for decryption. Furthermore, it was searchable encryption scheme; the CSP could search the encrypted files efficiently without leaking any information. It was probable that the proposed schemes have semantic security against adaptive chosen plaintext attacks.

In [12] the performance of Hadoop is been analyzed an implementation of mapreduce programming model for distributed parallel computing, executing on a virtualization environment comprised of 1 + 16 key pair values running the VMW. A set of experiments using the standard Hadoop benchmarks has been designed in order to determine whether or not significant reductions in the execution time of computations are experienced on the virtualization platform on a departmental cloud. The results mainly focused on the computation time. In [23] a cloud-computing based evolutionary algorithm is been presented using a synchronous storage service as a pool for exchanging information among the population of solutions. The multi-computer was composed of several normal PCs or laptops connected via Wi-Fi or Ethernet. The effect of how the distributed evolutionary algorithm reached the solution when PCs was tested whether that effect also translates to the algorithmic performance of the algorithm. To this end different problems were addressed using the proposed multi-computer, analyzing the effects that the automatic load-balancing and synchronization had on the speed of algorithm successful, and analyzing how the number of evaluations per second increases when the multi-computer includes new nodes.

The paper is organized as follows. Section 2 summarizes some of the formal definitions of the problem, section 3 presents the related work, section 4 presents the genetic algorithm proposed in this paper for solving the resource problem in the clouds using the map reduction algorithm, section 5 contains the experimental part of the paper where the performance of the proposed approach is evaluated and finally the conclusions are drawn in section 6.

RELATED WORK

Many definitions exist resource management techniques for cloud networks available in the literature.
K. Meri et al. [24] illustrated cloud-based evolutionary algorithms: An algorithmic study. This work shows the effect of how distributed evolutionary algorithm reaches the solution when new PC are added and tested and whether the effect translates the algorithmic performance. Different problems were addressed by analyzing the effect of load balancing and synchronization and are measured by number of evaluation per second when new nodes are added. The Evolutionary parallel algorithm solves the proposed problem and it was feasible both in homogeneous and heterogeneous environment.

Kai Zhu et al. [25] proposed Hybrid Genetic Algorithm for Cloud Computing Applications. An effective load balancing strategy improves the task throughput of cloud computing. Virtual machines are selected as a fundamental processing unit of cloud computing where the resources increases and vary energetically by the utilization of virtualization technology. As a result performance of load balancing has become complicated and is complex to be achieved. Multi-agent genetic algorithm (MAGA) a hybrid algorithm of GA was proposed whose performance is far superior to that of the traditional GA. The experimental results of paper shows the advantage of MAGA over traditional GA, where load balancing problem is solved in cloud computing moreover better performance is achieved.

Shin-ichi Kuribayashi et al. [26] proposed the Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments. This research develops a resource allocation model where both processing ability and bandwidth are allocated simultaneously to each service request and rented out on an hourly basis. The allocated resources are dedicated to each service request. An optimal joint multiple resource allocation method is proposed based on the above resource allocation model where it reduces the request loss probability and as a result, reduces the total resource required, compared with the conventional allocation method. The simulation result shows that the proposed method allocates resources fairly among multiple users efficiently.

Fang Yu et al. [27] illustrated Quantitative Analysis of Cloud-based Streaming Services. As online streaming services significantly increase in recent years, it becomes a critical issue to offer quality service via systems that benefit from cloud computing developments. The study on quantitative analysis to estimate and further improve the quality of cloud-based streaming services, deriving theoretical results on operation characteristics of queuing models with mild assumptions. By simulating the continuous-time Markov chain, according to the adopted operations rules for VMs, we also get performance indicators such as the average lag time and interruptions that a customer may experience under different environmental settings.

PROPOSED METHODOLOGY FOR SOLVING RESOURCE PROBLEMS IN A CLOUD

Cloud computing is a rapidly developing area which offers a possible increase in the flexibility and efficiency of the service providers. With the help of cloud computing data's can be provided through shared computing resources and can be easily accessed through the internet. These include applications, development tools and servers. Amidst of all these advantages cloud offers, the major drawback has been the resource problem. The major intention is to develop a system which can solve the resource problems in the clouds. For this purpose we have used map reducing algorithm which proves to be an efficient method for reducing the over usage of resources such as execution and computing time.

In our proposed method, we implemented a Map Reduce Programming model for distributed parallel computing and execution on a virtualization process which is used to detect non-sufficient reductions in the execution time and to detect the decrease in the computing time. For an efficient mapper, Genetic Algorithm (GA) is used for optimizing the parameters of map reduce algorithm. A map and reduce algorithm is used to abstract the complex parallel computations. The Apache Hadoop model is used for map reduce. The proposed method is explained in detail in the below sections.

The HADOOP distributed file system stores a large amount of datasets and also reduces the Map Reduce jobs by computing the clusters within the key pair values. HDFS stores files across a collection of servers in a cluster. Files are decomposed into blocks, and each block is written to more than one of the servers. HDFS ensures data availability by continuously monitoring the servers in a cluster and the blocks that they manage. The key pair values in Hadoop classified as Name key pair values and Data key pair values. Name key pair values will allocate a data request to data key pair values from the master to the workers. The Name key pair value monitors jobs through the Job Tracker process, and the name key pair values execute and track tasks through Task tracker. Name key pair value can be replicated to avoid single point failure. HDFS deliver inexpensive, reliable, and available file storage. But the major component of Hadoop is the parallel data processing system called Map Reduce.

Hadoop Database for Proposed Method

The database forms the basic part in the map reducing process. In our proposed method, the database is the various documents that are selected for performing the map reduce programming. The databases are selected such that the word document contains the more frequent words at regular interval so that the map reduce framework can be effectively performed. We extract the keywords from the documents and based on their frequency, the sets are formed. Then, we compute the relevancy between these keyword sets. Keyword-based relevancy measure relies on the idea that the content of a document can be characterized by a set of keywords that is a set of words expressing the most significant concepts in the respective document. The relevant measure devised contains two parts, where the first part is based on frequent keywords and the second part is based on the remaining keywords of the documents.

1. Frequent keyword-based similarity: The motivation behind this approach is that the most significant words are likely to be referred repeatedly, or, at least, more frequently than unimportant words. In practice, the words that are frequently occurring in a document have more expressive power in the file. Based on this, we have designed a frequent keyword-based similarity measure that gives more importance to the frequent keywords rather than infrequent keyword.
2. Keyword-based similarity: The relevant measure is not a best measure if it is only based on frequent keywords to categorize a web page. To overcome such a situation, we also incorporate keywords other than the frequent words for finding their suitable category. The importance of this keyword-based similarity measure is relatively less compared with the frequency based similarity measure.

Map reduce Programming Models

The Map Reduce programming model is usually used to develop a highly parallel application that process and generate a large amount of data. Map Reduce provides high scalability and reliability because of the division of the work into smaller units. The data are mainly given to a master key pair value, which is responsible for managing the execution of applications in the cluster. After submitting a job, the master initializes the desired number of smaller tasks or units of work, and puts them to run on worker key pair values. First, during the map phase, key pair values read and apply the map function to a subset of the input data. The map's partial output is stored locally on each key pair value, and served to worker key pair values executing the reduce function.

Map Reduce is extremely appropriate for huge data searching and processing operations. For traditional clusters, the model has shown excellent I/O features, which is apparent from its successful application in large-scale search applications by Google. Data in the Map Reduce framework are usually delineated in the form of key-value pairs <key, value>. The primary step of the computation is the map function where the framework reads input data and optionally changes it into proper key-value pairs. The second step which is the map phase, where on each pair <k, v> a function g which returns a multiset of new key-value pairs is applied. The function is expressed as below,

$$g(<k,v>) = \{<k_1, v_1>, <k_2, v_2>, \cdots, <k_n, v_n>\} \qquad (1)$$

Where k is the key and v is the value. In the reduce phase, all pairs that are generated in the preceding step are grouped according to their keys and their values are reduced using a function which is given by,

$$h(\{<k_1, v_1>, <k_2, v_2>, \cdots, <k_n, v_n>\}) = <k, v> \qquad (2)$$

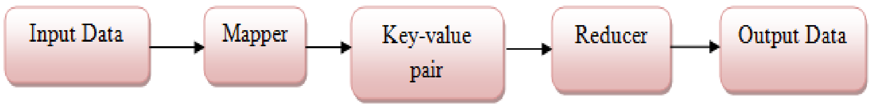

Fig. 1: Map-Reducing Process.

The input dataset which is a document is first applied to the mapper, which generates the key value pair. Here the key is the document name and value is the document content. The same words and the number are grouped as a key-value pair and this is given to the reduce function. The reduce function, then obtains all the pairs of word with the same key and value and counts the number of pairs in the document and then reduces the count by considering the same key-value pair as a single one. Once the map reduces step is over next the data are stored in HDFS. When the program calls MapReduce function, the following sequence of actions occurs sequentially.

The MapReduce function first splits the input files into a number of sections. Then it starts producing a number of copies of the program on cluster of key pair values. Among this, one is the master, which assigns the work to the remaining key pair value which is the workers. There are P map task and Q reduce task. The master splits the task to appropriate workers. The worker who is assigned each map task will read the content of the corresponding split input and generates the key/value pair for that input data and passes each to the Map function. Here we can optimize that key/pair value by employing GA. This can be described in the next section. The intermediate key/value pair produced by the Map function is then given to the memory for storage. The location is then passed to the master key pair value which then forwards this location to reduce workers. When the reduce workers get the location of the intermediate data output, it reads the entire intermediate data and based on the keys they are sorted such that data with the same key are collected into one group. The key and intermediate value set are passed by the workers to the Reduce function. The reduce function produces the final output of the Map Reduce model. This process continues until the entire map and reduce tasks are completed. After the completion of the process; the outputs are stored in the HDFS. The MapReduce program provides the way for better processing as it provides more resources to be available for storage.

The various drawbacks that exist in the related works are being sorted out by our proposed method and based on our proposed algorithm better resource problem solving strategies are obtained. The advantages of our proposed algorithm include,

- The Data involved in one Map/Reduce job is lesser. Hence lesser load on the buffers and better I/O.
- Datasets of any size can be utilized provided there is space available on the HDFS.
- Any number of datasets can be used given enough space on the HDFS.
- MapReduce may be easier for users to adopt for simple or one-time processing tasks.

EVOLUTIONARY ALGORITHMS

Camel In 1975, John Holland introduced a new way to solve problems with computers: Genetic algorithms (GAs) [22]. The GA is a heuristic search technique that simulates the processes of natural selection and evolution. GAs tends to find good and novel solutions to hard problems in a reasonable amount of time. The selection, crossover, mutation and fitness functions are discussed below.

Genetic Algorithm

IIOAB JOURNAL ISSN: 0976-3104

Genetic algorithms are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms. According to the principles of natural selection and survival of the fittest; natural populations are evolved in many generations. By simulating this process, genetic algorithms are able to use solutions to real world problems, if they have been suitably encoded. The genetic algorithm usually works as given below,

Generation and Selection

In genetic algorithm, a population is created with a group of individuals or chromosomes.

$$D=\{D_0, D_2, D_3, \cdots, D_{n-1}\} \quad , 0 \le j \le N_p - 1 \qquad 0 \le k \le n-1 \tag{3}$$

Here D represents the database with collection of key pair values. Among them we have to select the optimal one. The selection process decides which of the chromosomes from the population will be selected for crossover to create new chromosomes. This new chromosome will now include with the population to determine the next selection. The individuals with more fitness value will be selected. The selection is based on the fitness of the individuals.

Pseudo code for chromosome generation:

```
function Chrom = Generate_Chrom(Psize)
 Chrom =[];
 % Psize = 10;
 for t1 = 1 : 10
     Chrom =[Chrom ; randi([1,Psize])];
 end
```

Crossover

After selecting the individuals the next step is the crossover where two parents are made to mate each other. In crossover there are various types in which two point crossovers is more commonly used method. Here the offspring is produced by selecting two crossover point and the genes in between these two points are interchanged from the parents to form new offspring's.

Pseudo code for Crossover:

```
                        Function C = CrossOver (C, Psize)
for t = 1 : size(C,1)
                        %     tmp1 = C (t,1);
                        %     tmp2 = C (t+1,1);
                        %     C (t+1,1) =  randi([1,Psize]);
                           C (t,1) =  randi([1,Psize]);
                        end
```

Mutation

After crossover, new set of populations are produced. In order to provide individuality to each chromosome, mutation operation is performed where we replace any value with a new value to form a new individual. Among different types of crossovers, the two point crossover is selected with the crossover rate of CR.

In the two point crossover, two points are selected on the parent chromosomes using the eqns. The genes in between the two point's c1 and c2 are interchanged between the parent chromosomes and so Np/2 children chromosomes are obtained. The crossover point's c1 and c2 are determined as follows.

$$c_1 = \frac{\left|D_k^{(j)}\right|}{3} \tag{4}$$

$$c_2 = c_1 + \frac{\left|D_k^{(j)}\right|}{2} \tag{5}$$

Pseudo code for Mutation:

```
                    Function M = Mut(M,Psize)
               r = randi([1 2],1);
  for t = 1 : r
               r1 = randi([1 size(M,2)],1) ;
              M (t,r1) =randi([1,Psize]);
                    End
```

Fitness function

IIOAB JOURNAL ISSN: 0976-3104

After mutation the fitness of each individual is founded and the individual with high Fitness values are selected as the final solution. In our proposed method, the genetic algorithm is used to find out the suitable key pair value for transmission. Here the key pair value is represented as the bit of chromosome. The [Fig. 2] shows flow diagram of genetic algorithm for our proposed method. Initially generate 'N' number of chromosomes (key pair values). Next fitness for each of the individuals is calculated. In our proposed method the fitness value is calculated in relation to the distance between the key values. The key pair values with less distance are selected. Here a Euclidean distance from each key pair value to the next closest key pair values is calculated. It is given by the expression,

$$Fitness(F) = \frac{1}{\min}\left(E_d\left(m,n\right)\right) \quad (6)$$

Where m and n are the source and the destination pair values respectively.

Once fitness of the key pair values is calculated we proceed to next step in the genetic algorithm. Next is the selection process where the two chromosomes for crossover and mutation are selected. This selection is based on the fitness of the chromosome. More fit the chromosomes are more the chance for selection. There are various methods of selection in genetic algorithm. In our proposed method we use the roulette wheel selection method. The roulette wheel selection is used for selecting potentially useful solutions for recombination. In roulette wheel selection the chromosome with higher fitness value when compared to others are selected to form the new offspring's. The probability of selecting the 'k'th value is given by,

$$P_k = \frac{F_k}{\sum_{t=1}^{N} F_t} \quad (7)$$

Where the total number of values, P_k is the probability of the 'k'th key pair value and F_k is the fitness of the 'k'th value.

After selecting the solutions crossover and mutation are performed. In the proposed method we utilized two-point crossover method. Two points are selected in the parent chromosomes as R1 and R2 and the genes in between these two points are interchanged to form new offspring's. After the crossover operation, mutation is applied to the newly formed offspring's in order to make each individual independent of the other. After the mutation operation finally the fitness value of the newly formed individuals is calculated. By calculating the fitness value for each individual or key value pair, the values are analyzed and the key pair values with higher fitness values are selected as the most suitable key pair value for transmission. After selecting the optimal value, the map reducer process will be proceeded to get maximum resources.

SIMULATION RESULTS

A series of simulations are carried to evaluate the performance of the proposed GA to solve the resource allocation problems. In our experiment we have utilized the Hadoop map reduce in order to reduce the number of repetitions of the words for more than one time, so the resource memory can be utilized for more storage process. A large set of the databases is collected with different words and the key value pair for each line has to be found out. Here genetic algorithm is used to select the key value pair based on the fitness of the key value pair selected. We have used certain database where different key value pairs are generated based on the number of repetitions of the words. The above key value pair is selected using the map reduce programming and these are based on the availability of the words in the database. Next the selection of the key value is done with the help of the genetic algorithm. The process of selecting the key value pair depend on the fitness of the selected pair. Here the fitness of the key value calculates base the number of times the key word appears in a specific line of the document database. In our experiment we have taken the above keywords as the key value pair and they showed to be fit for selection thus reducing the extra time required in the process of analyzing the document thus saving the resources in a better way when compared to other methods.

It has been seen that our proposed method of map reduce using the genetic algorithm has shown a remarkable reduction in the execution time when compared to other existing method. The map reduces without using the genetic algorithm has shown that it takes more execution time when compared with our proposed method of map reduce using the genetic algorithm. Based on the parameters experimentally determined, the effect of data size, number of clusters, degree of cluster distinctness, degree of cluster asymmetry and level of data randomness at the execution time and clustering quality of the four fast clustering algorithms were evaluated empirically. A discrete event simulator has been developed to evaluate the transition time of the reconfiguration plan obtained by the reduced map approach with GA and the performance of the proposed algorithm in consolidating size of resources.

Transition Time of the Reconfiguration Plan

In the experiments, three types of resources are simulated: CPU, memory and I/O, and three types of VMs are created: CPU-intensive, Memory-intensive, and I/O intensive VMs. A Virtual cluster (VC) consists of the same type of VMs. For the CPU-intensive VMs, the required CPU utilization is selected from the range of [30%, 60%], while their memory and I/O utilization are selected from the range of [1%, 15%].

ISSN: 0976-3104

The VMs are first generated in physical nodes according to the above method. A node is not fully utilized and will have a certain level of spare resource capacity. The service rate of requests of each VM is calculated using the performance model. The workload manager is used in the experiments. The arrival rate of the incoming requests for each VC is determined so that the VCs' QoS can be satisfied. The average execution time for each type of requests is set to be 5 seconds, and the QoS of each VC is defined as 90% of the requests' response time is no longer than 10 seconds. A VC's workload manager (LM) dispatches the requests to VMs, and therefore the request arrival rate for each VM can be determined. Then the developed GA is applied to consolidate VMs so that the spare resource capacity in nodes can converge to a smaller number of nodes. After the GA obtains the optimized system state, the reconfiguration plan is constructed to transfer the Cloud from the current state to the optimized one.

The average time for deleting and creating a VM is 20 and 14 seconds, respectively. The migration time depends on the size of VM image and the number of active VMs in the mapping nodes. The migration time in our experiments is in the range of 10 to 32 seconds.

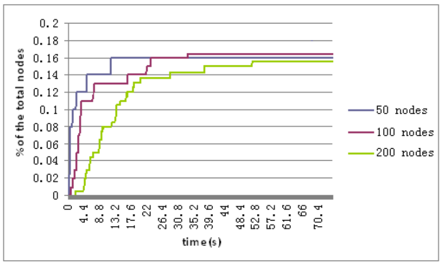

Fig. 2: The quantity of nodes saved as the GA progresses.

[Fig. 2] shows the number of nodes saved as the GA progresses. In the experiments in [Fig. 2], the number of nodes varies from 50 to 200. The experiments aim to investigate the time that the GA needs to find an optimized system state, and also investigate how many nodes the GA can save by converging spare resource capacities. The free capacity of each type of resource in the nodes is selected randomly from the range [10%, 30%] with the average of 20%. The number of the VMs in a physical node is 3. The number of the VCs in the system is 30. As can be observed from [Fig. 2], the percentage of nodes saved increases as the GA runs for longer, as to be expected. Further observations show that under all three cases, the number of nodes saved increases sharply after the GA starts running. It suggests the GA implemented in this paper is very effective in evolving optimized states. When the GA runs for longer, the increasing trend tides off. This is because that the VM-to node mapping and resource allocations calculated by the GA approaches the optimal solutions. Moreover, by observing the difference of the curve trends under a different number of nodes, it can be seen that as the number of nodes increases, it takes the proposed GA longer to approach the optimized state.

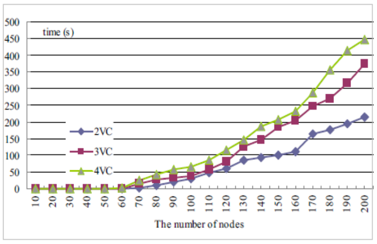

Fig. 3: The execution time of the proposed reconfigure GA algorithm for different number of nodes and VCs.

[Fig. 3] shows the time it takes for the proposed algorithm approach to find the optimal reconfiguration plan under a different number of nodes and different number of VCs. The optimized system states are computed by the GA. The average spare capacity in nodes is 15%. It can be seen from this figure that the time increases as the number of nodes increases and also as the number of VCs increases. When the number of nodes is 200 and the number of VCs is 4, the time is 450 seconds, which is unbearable in real systems. That is why a proposed approach is necessary to quickly find the sub-optimal reconfiguration plan for the large scale of systems. A GA is developed to compute the optimized system state and consolidate resources. The modified map reduce model is then developed to transfer the Cloud from the current state to the optimized one computed by the GA.

Effect of Resource Data Size

The parameters for the synthetic data generation program are summarized in [Table 1]. The default values of these parameters for synthetic resource data sets were n=3000, k=20, t=0.2, a=1, and r=1%. Depending on the type of experiment conducted, the respective parameter was varied, while the rest of parameters adopted their default values. For example, to evaluate the data size effect on the target

IIOAB JOURNAL ISSN: 0976-3104

clustering algorithms, the parameter n took values from 500 to 7000, while default values were used for the rest of the parameters.

Table 1: Parameters and default values for synthetic data generation

Symbols	Meanings	Defaults
n	Number of resources in a cloud server	3000
k	Number of clusters	20
t	Degree of cluster distinctness	0.2
a	Degree of cluster asymmetry	1
r	Level of data randomness	1%

Synthetic data sets were generated for various data sizes, ranging from 500 to 7,000 (i.e., n=500, 1,000, 2,000, ..., 7,000). Remaining parameters received their default values, as defined in [Table 1]. [Fig. 4] shows the performance of the target clustering algorithms as a function of the data size.

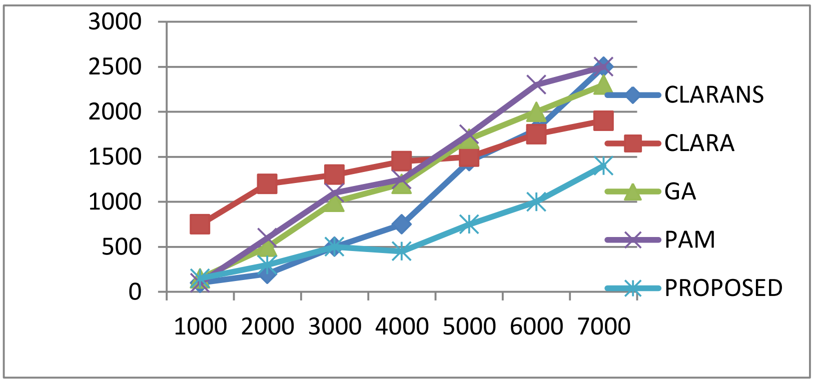

Fig. 4: Effect of Resource data size.

As shown in [Fig. 4], PAM and CLARA slightly outperformed the others in terms of clustering quality when given only a small data size (i.e., fewer than 1,000).When the data size was increased, the clustering quality of the proposed algorithm degraded as compared to that of others. In terms of execution time, our proposed algorithm is more efficient than CLARA, CLARANS, GA and PAM when the data size was more than 3000 sec. As the data size increased, CLARA increased its execution time increased, but our proposed algorithm is a very good solution for the large resource data size communication between cloud servers.

Table 2: Comparison of execution time of the proposed method using genetic algorithm and existing methods

Resource Data size	Execution time(sec) CLARANS	CLARA	GA	PAM	Proposed Algorithm
1000	100	750	150	100	150
2000	200	1200	500	600	300
3000	500	1300	1000	1100	500
4000	750	1450	1200	1250	450
5000	1450	1500	1700	1750	750
6000	1800	1750	2000	2300	1000
7000	2500	1900	2300	2500	1400

In the [Table 2], the execution time for the proposed and existing methods is shown. As it indicates, the execution time for the proposed method improved over time and this proved to be more effective when compared with the existing methods. Based on the execution time for different numbers of nodes the graph is plotted for comparing the performance of different algorithms which are used in map reduce programming. As shown in [Fig. 4], the execution time for our proposed method of map reducing using the genetic algorithm proved to be more reliable when compared to the other algorithms.

CONCLUSION

The Genetic algorithm, Partioning around Medoids, Clustering Large Applications, and Clustering large applications based on randomized search is discussed to solve resource problems in the cloud. Here we

IIOAB JOURNAL
ISSN: 0976-3104

employed a Map Reduce programming model for distributed parallel computing and execution of virtualization process which is used to detect non-sufficient reductions in the execution time and to detect the decrease in the computing time. The proposed method proved to be efficient, one when considering the outcome compared to other algorithms. The graph shows the execution time is reduced to a large instant when compared to the existing method. The simulation results confirm to be an efficient method in reducing the resource problems that occur in the cloud computing. Further enhancement can be done to reduce execution time for larger networks with the number of node and to testify the reliability and securability of the resource allocation where the execution time decreases considerably that enhances the horizon for further study.

CONFLICT OF INTEREST

There is no conflict of interest regarding this manuscript

ACKNOWLEDGEMENTS

There is no acknowledgement regarding this manuscript.

FINANCIAL DISCLOSURE

There is no financial support for this manuscript.

AUTHOR CONTRIBUTION

I Arivudainambi contribute to development and growing popularity of cloud computing indicates the evolution in the way IT infrastructure and services are distributed and expended. This increased use of cloud computing resulted in resource problems which have to be solved for better usage of the clouds.

I Dhanya contribute to present an efficient method for solving resource problems in the cloud using modified map reducing algorithm. Here we employed a MapReduce programming model with GA for distributed parallel computing and execution on a virtualization process which is used to detect non-sufficient reductions in the execution time and to detect the decrease in the computing time.

REFERENCES

[1] Arivudainambi D, Dhanya D. [2013] Improving MapReduce Performance in Cloud Using Genetic Algorithm, 1060-1067, Conference Proceedings, International Conference on Mathematics and Computer Engineering.

[2] Chaisiri S, Lee BS, Niyato D. [2009] Optimal Virtual Machine Placement across Multiple Cloud Providers, IEEE Asia-Pacific Services Computing Conference (IEEE-APSCC). 103-110.

[3] Lucas-Simarro JL, Vozmediano RM, Montero RS, Llorente IM. [2013] Scheduling strategies for optimal service deployment across multiple clouds, Future Generation Computer Systems. 29:1431-1441.

[4] Tordsson J, Montero RS, Vozmediano RM, Llorente IM. [2012] Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers, Future Generation Computer Systems. 28:358-367.

[5] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. [2009] Cloud computing and emerging IT Platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems. 25: 599–616.

[6] Mian R, Martin P, Vazquez-Poletti JL, [2013] Provisioning data analytic workloads in a cloud, Future Generation Computer Systems , 29:1452-1458.

[7] Zhu Y, Hu H, Ahn GJ, Yau SS. [2012] Efficient audit service outsourcing for data integrity in clouds, The Journal of Systems and Software. 85:1083– 1095.

[8] Srirama SN, Jakovits P, Vainikko E. [2012] Adapting scientific computing problems to clouds using MapReduce, Future Generation Computer Systems. 28:184–192.

[9] Liu Q, Wang G, Wu J. [2012] Secure and privacy preserving keyword searching for cloud storage services, Journal of Network and Computer Applications. 35:927–933.

[10] Pallickara SL, Pierce M, Dong Q, Kong CH. [2009] Enabling Large Scale Scientific Computations for Expressed Sequence Tag Sequencing over Grid and Cloud Computing Clusters, In Proceedings of the 8th International Conference On Parallel Processing And Applied Mathematics.

[11] Seinstra FJ, Maassen J, Van Nieuwpoort RV, Drost N, Kessel TV, Werkhoven BV, Urbani J, Jacobs C, Kielmann T, Bal HE. [2011] Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds, Computer Communications and Networks. 167-197.

[12] Kim M, Lee H, Cui Y. [2011] Performance Evaluation of Image Conversion Module Based on Map Reduce for Transcoding and Transmoding in SMCCSE, IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing. 396-403.

[13] GonzaLez-Ve´Lez H, Kontagora M. [2011] Performance Evaluation Of Mapreduce Using Full Virtualisation On A Departmental Cloud, International Journal of applied mathematics and computer science. 21(2):275–284.

[14] Grossman R, Gu Y. [2008] Data mining using high performance data clouds: experimental studies using sector and sphere, In Proceeding of the 14th ACM SIGKDD International conference on Knowledge discovery and data mining, New York. 920-927.

[15] Papadimitriou S, Sun J. [2008] DisCo: Distributed Co-clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining, In Proceedings of the 8th IEEE International Conference on Data mining, Hawthorne. 512-521.

[16] Zhang C, Sterck HD. [2009] CloudWF: A Computational Workflow System for Clouds Based on Hadoop, Cloud computing. 5931:393-404.

[17] Jin C, Buyya R. [2009] Map Reduce Programming Model for .NET-based Distributed Computing, In Proceedings of the 15th International Euro-Par Conference on Parallel Processing, Berlin. 417-428.

[18] Sehgal S, Erdelyi M, Merzky A, Jha S, [2011] Understanding Application-Level Interoperability: Scaling-Out Map Reduce Over High-Performance Grids and Clouds , Future Generation Computer Systems, 27: 590-599.

[19] Marozzo F, Talia D, Trunfio P. [2008] Adapting Map Reduce for Dynamic Environments Using a Peer-to-Peer Model , In Proceedings of the 1st Workshop on Cloud Computing and its Applications.

[20] Tian F, Chen K. [2011] Towards Optimal Resource Provisioning for Running Map Reduce Programs in Public Clouds, In Proceedings of the IEEE International Conference on Cloud Computing (CLOUD). 155-162.

[21] Yang X, Zola J, Aluru S. [2011] Parallel Metagenomic Sequence Clustering via Sketching and Maximal Quasi-clique Enumeration on Map-reduce Clouds, In Proceedings of the 2011 IEEE International Conference

on Parallel & Distributed Processing Symposium (IPDPS). 1223-1233.

[22] Husain M, McGlothlin J, Masud MM, Khan L, Thuraisingham B. [2011] Heuristics Based Query Processing for Large RDF Graphs Using Cloud Computing, IEEE Transactions on Knowledge and Data Engineering. 23(9):1312-1327.

[23] Holland JH. [1975] Adaptation in natural and artificial systems: an introductory analysis with applications to biology, Control and artificial intelligence. Ann Arbor: University of Michigan Press. 175-177.

[24] Patel K, Annavaram M. [2013] NFRA: Generalized Network Flow Based Resource Allocation for Hosting Centers, IEEE Transactions On Computers. 62(9):1772-1785.

[25] Meri K, Arenas MG, Mora AM, Merelo JJ, Castillo PA, Garcia-Sanchez P, Laredo JLJ. [2012] Cloud-based evolutionary algorithms: An algorithmic study, Natural Computing-Springer. 12(2):135-147.

[26] Zhu K, Song H, Liu L, Gao J, Cheng G. [2011] Hybrid Genetic Algorithm for Cloud Computing Applications, IEEE Asia-Pacific Services Computing Conference. 182-187.

[27] Kuribayashi SI. [2011] Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments, International Journal of Research and Reviews in Computer Science(IJRRCS). 2(1).

[28] Yu F, Wan YW, Tsaih RH, [2013] Quantitative Analysis of Cloud-based Streaming Services, IEEE International conference on service computing (SCC). 216-223.

[29] Salvadores M, Zuo L, Imtiaz SMH, Darlington J, Gibbins N, Shadbolt NR, Dobree J. [2010] Market Blended Insight: Modeling Propensity to Buy with the Semantic Web, Lecture Notes in Computer Science. 5318:777-789.