ISSN:0975-9646

Improvised Genetic Approach for an Effective Resource Allocation in Cloud Infrastructure

M. Durairaj[#1], P. Kannan[#2]

[#1,2]*School of Computer Science, Engineering and Applications, Bharathidasan University*
Tiruchirappalli, India

***Abstract*— Allocation and schedule of virtual machines based on the requisite of cloud users is a challenging crucial chore in cloud services especially in IaaS (Infrastructure as a Service). Whenever the virtual machines requests are increased or decreased, the resources have to be balanced to attain optimal resource utilization. In this paper, we propose an approach namely Effective Cloud Resource Allocation Using Improvised Genetic Approach, which directs to accomplish better virtual machine allocation across cloud servers for maintaining vertical elasticity and minimizing response time. The proposed approach is focused on elasticity and Scheduling to improve resource allocation mechanism in cloud computing. This paper not only focuses the resource utilization problem, but also discusses our innovative algorithm called Enhanced Genetic Algorithm (EGA) using Multipurpose Mutation Operator. The proposed algorithm makes the effectual use of mutation operator to avoid local optimum problem. It repairs infeasible solutions and handles local search efficiently. The result shows that the EGA provides an optimal solution and proves better performance compared to the existing algorithms. Our method exemplifies that there is a substantial improvement in response time and also reduction in VM (Virtual Machine) migration count.**

***Keywords*— Cloud Computing, Virtualization, Elasticity, Resource Allocation, Scheduling, Genetic Algorithm.**

I. INTRODUCTION

Cloud computing is a new prototype and comprehensive virtualization [1] [6] system of prominent distributed computing in elastic manner. It delivers resources such as virtual machines, data storage, processing power, and networks as an on demand service rather than as an IT product with security [5]. It helps many corporate, educational, research and development sectors to reduce cost, time and focus on core development of project work rather than expending time on IT infrastructure consequences. The succeeding definition[2] of cloud computing has been originated by the NIST (National Institute of Standards and Technology): Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud computing model has five substantive features such as on-demand self service, broad network access, resource pooling, rapid elasticity and measured service. It has four deployment models private cloud, community cloud, public cloud and hybrid cloud. The cloud computing service models are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). According to Berkeley report [3] defines: ‘‘Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service’’.

Elasticity [8] is the concept of utilizing cloud resources (virtual machines, storage, networks, platforms, and applications) in more flexible fashion. In point of fact, the profound feature in cloud is elasticity which supplies cloud resources in an elastic manner with on demand workload changes by managing the ability to scale up and scale down of cloud system resources. National Institute of Standards and Technology (NIST) [4] defines “Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.” Cloud computing resources are shared, heterogeneous and platform independent. So the resources have been definitely wasted if the cloud resources are not shared in right order. Resource allocation is a significant approach to improve resource utilization in cloud and mobile cloud environment.

Resource allocations is done with the help of scheduling which is used for supplying resources efficiently and effectively among cloud servers based on request from cloud users and availability of resources. Scheduling algorithms are applied for proper resource utilization, reduce virtual machine migration count, decrease the waiting time for resources and to assure the resources are balanced equally among the servers or datacenters. The resources are allocated in the server based on the request to create virtual machine by cloud users. Scheduling refers to the group of procedures to assure the allocation of resources by a scheduler. Optimum cloud resource scheduling serves both cloud service provider and cloud user. The users acquire gain in terms of cost and response time. The providers obtain profit in terms of resource utilization. There have been several types of scheduling algorithms available in cloud computing system. But there are no existing algorithms which focus on scheduling with elasticity in static and mobile cloud environment.

Genetic algorithms arise from the evolutionary principles of the nature population. It is a rapidly growing field of artificial intelligence. These are [5] stochastic searching method that has a more beneficial optimization power and inner implicit parallelism. This algorithm is initiated with a set of outcomes called population. Outcomes are transferred to form a new population from parent population according to their fitness. The outcomes, which has higher fitness is more possible to reproduce and should represent as a genome based on the problem.

Once we formulate a genome, then the genetic algorithm [7] [9] creates a population of outcomes. It employs genetic operators such as crossover and mutation to evolve the outcomes in order to detect the best one. This algorithm frequently alters a population of individual outcomes. It randomly selects offspring's from the current population and applies them as parents to generate individuals for the succeeding generation in each step. Finally, the population acquires towards an optimal solution on consecutive generations. GA is used to reduce virtual machine migration count and extremely handles resource allocation, when more VM allocation requests occur. The improvised genetic algorithm produces a virtual machine scheduling strategy with elastic manner in static and mobile cloud computing systems.

In this research work, an improvised mutation based genetic with elastic cloud resource allocation approach with the help of genetic algorithm is proposed. The GA based elastic cloud resource allocation approach focus on scheduling virtual machines based on the availability of cloud resources besides their response time.

The remaining work of the paper is formed as follows. In Section II, we present the related works. In section III, we explain the problem description and proposed system. Results and discussions are presented in section IV. Section V gives the conclusion and future work of the paper.

II. Related Works

A. Vouk [10] presents load balancing as the main challenge because of the scalability of cloud computing resources. It widely takes over by the enterprises and academia to utilize resources in effective way. Duy et al. [11] proposed to find out the accuracy resource prediction of host load utilization level by using Back propagation Artificial Neural Network (ANN) prediction approach in grid computing environment. Scheduling virtual machines using genetic algorithm in cloud was proposed by Gu et al. [12] to indicates how it is efficient compare with Least-load technique.

Yang Xu et al. [13] proposed a new model for load balancing to ameliorate the performance of cloud computing in distributed application level. A thought of four different resource allocation approaches and three algorithms has been explained and discussed by Gomoluch J et al. [14]. The approaches are state based, pre-emptive, non pre-emptive and model based. The objective is minimizing the communication operating cost of auctions in both non preemptive and state based system model. The remaining two approaches are flexible type.

Optimal Virtual Machine Placement (OVMP) algorithm was proposed by Chaisiri et al. [15]. It decides the optimal placement of virtual machines using a linear programming model. Grewal et al. [16] proposed a rule-based approach is utilized for cloud applications resource scaling in hybrid cloud computing environment. Their approach focused on reactive approach, threshold values, QoS attributes and predefined policies.

Marston et al. [17] described that service level agreements play a vital role in the corporate, research development and academic environment for virtual machine. They also focused on penalties model, infrastructure service systems, datacenter resource optimization in cloud. Antonescu et al. [18] depict a framework, which focus on processing the workload concurrently by analyzing physical resources in distributed enterprise data systems.

Prediction based virtual machine allocation algorithm was presented by Roy et al. [19]. It uses a second rate autoregressive travelling average prediction technique for optimizing the usefulness of the cloud application over an average prediction horizon. Antonescu et al. [20] presents virtual machine scaling algorithms for optimally notice most suitable scaling conditions using models of distributed applications from workload benchmarks.

Kousik Dasgupta et al. [21] proposed a genetic algorithm based new load balancing strategy to effectively utilize resources and balance the cloud infrastructure load. The proposed algorithm compared with the available approaches like Round Robin, First Come First Serve and Stochastic Hill Climbing search algorithm. An enhanced load balancing approach to avert deadlocks was carried out by EpoMofolo et al. [22]. It is happening within the cloud serves while migration of virtual machines in cloud environment. This algorithm is applying the wait time and hop time procedures to enhance load balancing.

Buyya et al. [23] delivered a modeling which contains procedures for simulating large scale infrastructure and network connections in cloud computing environments using cloudsim. Nguyen et al. [24] proposed an elaborated architecture to operate the allocation of virtual machines in dynamic manner. Allocation of virtual machine based on structural constraint aware mechanism and algorithm were presented by Jayasinghe et al. [25] to enhance availability and performance on IaaS environment.

Di Costanzo et al. [26] demonstrated virtualization technology to permit resource allocation among grids. The intergrid mechanism allocates resources based on the time constrain. In cloud, resource allocation are based on monetary compensation and not timely constrained. Rodrigo N. Calheiros et al. [27] presented cloud coordinators agents, which is used to maintain reliability, performance and scalability of elastic cloud applications in intercloud. Evaluating and estimating the cost profit for policies of resource allocation and provision using cloud computing have a significant impact rather than the traditional performance metrics [28].

Proper efficient resource utilization [29] [30] particularly reduce power consumption, carbon emission rates, global warming and operational cost in cloud data center. Vanderster et al. [31] concentrate resource allotment challenges from resource provider side, which focus on the admission control policies and gaining of profit based scheduling. Most of the previous works are focused to minimize the cost in resource scheduling. Furthermore, most of the existing approaches are suitable only for distributed computing environment and not for cloud computing. This work had focused on elastic virtual machines resource allocation in cloud infrastructure level. The proposed approach supports cloud elasticity, intercloud datacenter support and better load balancing.

III. Problem Statement

The objective of the resource allocation problem is to minimize the response time which heavily depends upon the execution time of the scheduling algorithm. Here, we proposed the enhanced genetic algorithm which reduces the execution time and also concentrate on improving the resource utilization rate by equally distributing the load. To provide a better elastic resource allocation at any time, the elastic resource allocation problem can be formulated as apportioning N number virtual machines requests submitted by cloud users to M number of servers in the cloud computing system. Each cloud server will have processing unit and memory utilization vector showing current memory and processing unit utilization status. Each virtual machine is allocated by the server based on service level agreements and resource availability of cloud provider. There are various constrains on the allocation of virtual machines in server.

1. All VMs must be allocated in the given server
2. Each virtual machine maintained by elastic queue
3. Virtual machine size should be less than the server size
4. One virtual machine should be placed in one server
5. Ensure that there is no ideal space

Thus the problem of elastic resource allocation in cloud has been simplified as an optimization problem which has the objective to improve the load distribution which can be mathematically expressed as follows:

$$Minimize\ CRLDV = \frac{\sum_{j=1}^{m}(AASS_j - \overline{AASS})^2}{m} \qquad (1)$$

Subject to,

$$AASS_j = HS_j + AR_j \quad \forall j \qquad (2)$$

$$HS_j = \sum_{i=1}^{n} X_{ij} B_{ij} \quad \forall j \qquad (3)$$

$$S_j \geq HS_j \quad \forall j \qquad (4)$$

$$B_{ij} = \begin{cases} 1 \text{ if VM i is running on server j} \\ 0 \text{ otherwise} \end{cases} \qquad (5)$$

$$\overline{AASS} = \frac{\sum_{j=1}^{m} AASS_j}{m} \qquad (6)$$

Here, CRLDV denotes the cloud resource load distribution variance, *AASS* refers to the actual allocation size of the server variable which calculates the size of running virtual machines and new incoming virtual machines of size after scheduling in cloud server. S_j denotes the remaining space in the cloud server *j*. AR_j denotes the size of running virtual machines in server *j*. HS_j refers to the size of new requests of virtual machine allocation from waiting queue or cloud user. X_{ij} stands for the memory space occupied by virtual machine *i* in server *j*. B_{ij} is a binary variable which indicates either the virtual machine *i* is running on server *j* (1) or not (0). The response of virtual machine resource allocation operations on the given server is mentioned as a schedule.

IV. Proposed Work

A) Genetic Algorithm

1. Introduction:

The Genetic Algorithm is an optimization algorithm which uses the technique of computerized search based on natural selection and genetics. This algorithm was conceived (mid-sixties) and published (1975) by Prof. John Holland of University of Michigan. The core thought behind genetic algorithm is to commence with randomly generated chromosomes and carry out the "survival of the fittest" scheme in order to develop better solutions.

The advantages of Genetic algorithm are to operate with a coding of variables, processes a number of population points at the same time. A distinctive genetic algorithm procedure consists of population, fitness evaluation, chromosome selection, employing genetic operators such as crossover, mutation, inversion, immigration and termination. A Genetic algorithm commences its search with a stochastic group of candidate solutions which normally coded in binary representations. Every chromosome is allotted a fitness value with respect to the objective function of the optimization problem.

Thenceforth, the members of the previous population are altered and the new population is created by using three operators such as selection, crossover, and mutation. Genetic algorithm works continuously by applying these three operators successively in each generation till the stopping criterion is reached. Due to the global perspective, simplicity of implementation and inherit parallelism, the genetic algorithm has been used as a very successful optimization tool for many real world problems.

Selection is a function for choosing genomes from chromosome to evaluate and measure the worth of chromosomes. It is an important stage in genetic algorithm. Roulette wheel, rank selection and steady state selection techniques are commonly used in selection stage. Crossover is the strategy combines two chromosomes and produces two new offsprings which represent the next population. Choosing of right crossover and mutation based upon the encoding procedure and according to the problem requirements.

Premature convergence is a vital challenge in the most optimization problems. This problem states that the situation where nearly all of the chromosomes in the population share the same fitness value. So, it is needed to mutate genes in chromosomes using mutation operator which is able to explore new areas. The proposed algorithm modifies the mutation operator effectively and efficiently to avert local optimum problem, fix infeasible solutions and neighborhood search.

B) *Working cycle of Genetic Algorithm*

Genetic algorithm uses an iterative optimization procedure and operates with a number of solutions instead of functioning with a single solution.

C) *The procedures of the Genetic Algorithm*

The pseudo code of the Genetic Algorithm

```
Step 1   Input:
         Maximum number of generation, population size,
         chromosome size, mutation probability, crossover
         probability
Step 2   Creation of initial population.
Step 3   Loop until maximum generation reached
Step 4   Evaluate the fitness of each individual with respect to
         the objective value
Step 5   Apply the genetic operators selection, crossover and
         Mutation
Step 6   End
Step 7   Output:
          The results
```

D) *Improvised Genetic Algorithm for elastic virtual machine resource allocation*

Generally, in the most of real world problems, the surface of search space is not easy to identify. If the problem had many local optimum peaks, it may often trap with the pre-mature convergence problem. Moreover, the stall generation also led to pre-mature convergence problem. So in this work, an improvised genetic algorithm is proposed to avoid pre-mature convergence problem.

1) Chromosome representation: Encoding the solution as a chromosome is vital part in the proper working of the Genetic algorithm. Binary encoding is one of the chromosome representation techniques which use the chromosome mapping of the target variables to the string code. In addition to that, it is the most suitable chromosome representation to avoid infeasibility as much as possible even slightly more number of bits is required compare with other representations from the perspective of storage. The applicability of accessing crossover and mutation operator has been used in easy manner for this representation. Two chromosome representations are suitable for this problem. One is VM-Server based value representation and the second one is binary representation as shown in Fig. 1 and Fig. 2 respectively. The size of bits is required for storage in VM-Server based value representation is 2 × No of VMs whereas in binary encoding J(no of server) × k(No of VMs) size of bits is required for storage.

(4,1)	(5,1)	(6,2)	(7,1)	(8,3)	(9,3)

Fig. 1 VM-Server Pair Representations

100	100	010	100	001	001

Fig. 2 Binary Encoding

These two chromosome representations had drawn for the example schedule shown in Fig 3 respectively.

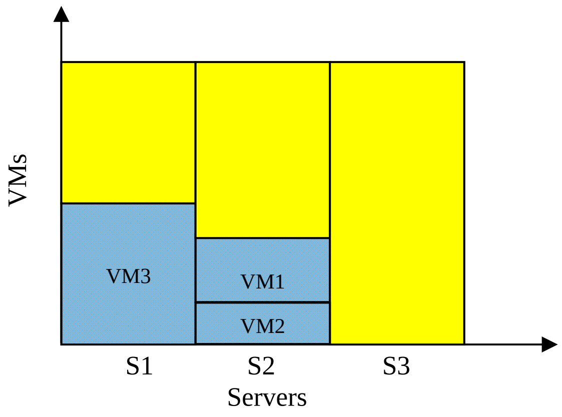

Fig. 3: Already Allocated VMs

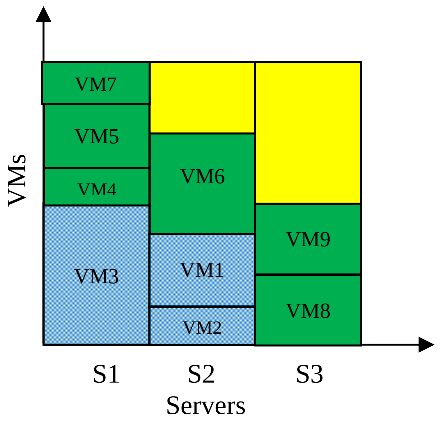

Fig. 4 VMs after Schedule

TABLE I
VMs AND SIZE

Number of VMs	Size of VMs
VM1	1GB
VM2	512MB
VM3	2GB
VM4	512MB
VM5	1GB
VM6	1.5GB
VM7	512MB
VM8	1GB
VM9	1GB

This example schedule has considered three cloud servers and nine virtual machines. Each server is considered to have 4 GB of memory size and virtual machine sizes are varied based on the cloud user's request. The size of the virtual machines is listed in table 2. Fig. 3 shows that virtual machines VM1, VM2 and VM3 already scheduled and allocated for resource utilization. Fig 4 shows how the new VMs requests from the user are scheduled in the servers.

2) Creation of Initial Population: Initial population generation is the first step in genetic algorithm. Each chromosome is assessed and allotted a fitness value according to the fitness function. This work takes the inverse value of the objective function as the fitness value. A chromosome is developed by assigning virtual machine to available cloud server repeatedly and thereby a solution is developed. Additionally, this algorithm checks whether the schedule is feasible or not. If not feasible, the algorithm uses mutation as repairing strategy to convert feasible solutions.

3) Fitness: Fitness function is creditworthy for evaluating how a potential solution is good and relative from one to another. It returns a fitness value or positive integer value which shows how near to the optimal solution. If the fitness value is higher, the solution will be better.

4) Selection: Selection is one of the significant elements to be taken while implementing the genetic algorithm. The selection procedure is used to develop child chromosome from parents for the succeeding generation and determines what sort of solutions will be employed in genetic algorithm operations. The selection operator is cautiously developed to ensure that population member with higher fitness have the larger probability of being taken for mutation. Even, the unfitter elements of population still have a little probability of being chosen is also important. Moreover, the selection pressure in the genetic algorithm must be chosen in such a way to ensure that the search process is global and does not simply converge to the nearest local optimum. Among the various available selection schemes, rank based roulette wheel selection has been used in this work. This selection scheme chooses the chromosomes based on fitness rank. This selection procedure maintains diversity and dynamic selection pressure at the same time provide better result.

5) Crossover: Crossover is a procedure to produce new offsprings from parental chromosomes by replicating selected bits from each parent strings. In single-point crossover, crossover degree is determined randomly by fragmenting common crossover point on both parent chromosomes. Afterwards, a new offspring is produced by swapping the fragmented part on both parent strings. The restricted Single point crossover is implemented in this paper. The crossover mechanism is exemplified in Fig. 5. Let consider an example of two parent chromosomes with 18 binary variables each. The chosen crossover degree is 9. After the crossover the new offspring's are produced.

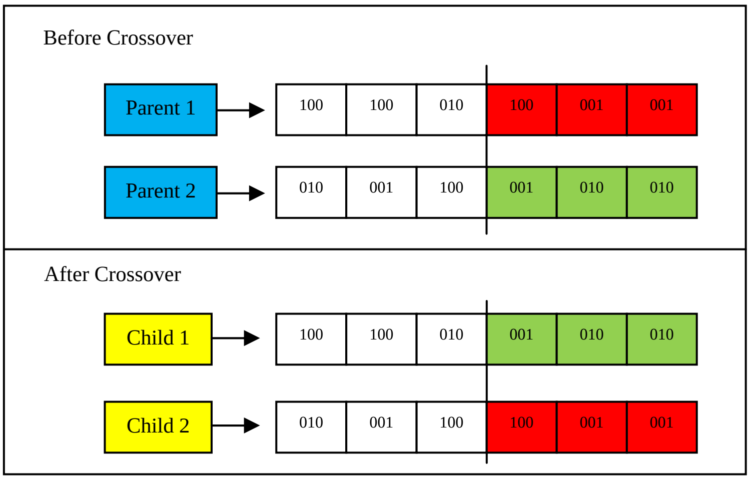

Fig. 5 Single-point crossover

In this example, the restricted single-point crossover parental chromosomes represent how virtual machines are scheduled from VM4 to Vm9 in cloud server. Chromosome strings or virtual machines are represented in three different colors such as blue, green and yellow. Blue color represents already scheduled virtual machines VM1, VM2 and VM3. Green color indicates presently what are the VMs from VM4 to VM9 are in the scheduling process. How much free resources are available in cloud server are shown in yellow color. Parent chromosome 1 strings are presented in Fig. 6 as a form of virtual machines and cloud servers before crossover occur. The parent chromosome 2 is shown in Fig. 7 where VM9 is not fit in server 1 and so the infeasibility is occurred. To avoid infeasibility, mutation is used as the repairing strategy in this situation.

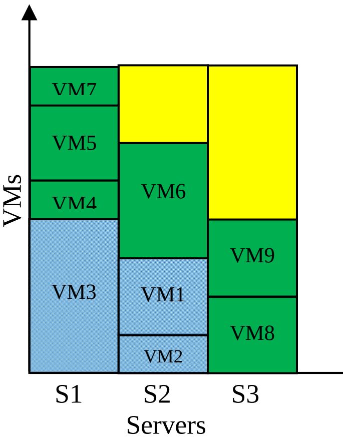

Fig. 6 Parent 1

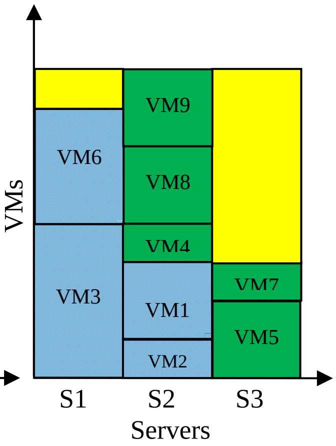

Fig. 7 Parent 2

After crossover, the offspring chromosomes are illustrated in Fig. 8 and Fig. 9. From the Fig. 9, it can be observed that child 1 represents an infeasible schedule. So to avoid such infeasible schedule, vertical elasticity is applied to improve resource allocation.

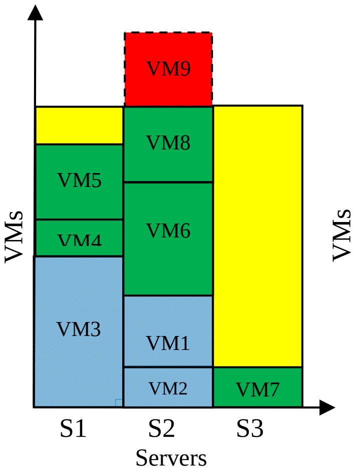

Fig. 8 Child 1

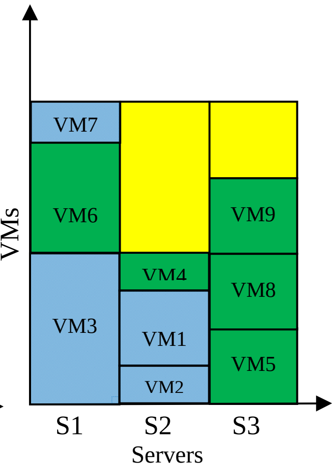

Fig. 9 Child 2

6) Mutation: Mutation is a crucial genetic operator that arbitrarily alters either the gene value or locus of a gene in a chromosome for maintaining the multifariousness in the population. The global search is good in genetic algorithm but dumb to converge. So in order to increase the convergence speed of the genetic algorithm, it has to be combined with a valuable local search method. Even though, local search is beneficial at fine tuning but there are possibilities to trap in local optima. So, in order to get away from local optima and carry fine tuning, the enhanced genetic algorithm is using global as well as local search. Local search in this circumstance can be conceived of as a neighbourhood search algorithm. Multipurpose rotation based mutation operator is adapted in this paper which is depicted in Fig.10. The neighbourhood of a chromosome refers to the set of schedules convertible from the chosen chromosome by rotating the gene position. While applying the mutation operator, it chooses the best solution from the neighbourhood of the current solution. A solution is said to be the local best solution if it has the least objective value than any other solution in the neighbourhood.

In this paper, multipurpose rotation based mutation operator is not only used as local search technique but also served as the repairing strategy for infeasible chromosomes that produced after crossover and attempted to preserve the diversity. Furthermore, this EGA checks whether the stall generation is occurred or not on a periodical basis while in the execution of the algorithm. In addition to, the same mutation strategy is used to overcome the stall generation and hence this EGA algorithm preserves the diversity among the chromosomes in the population.

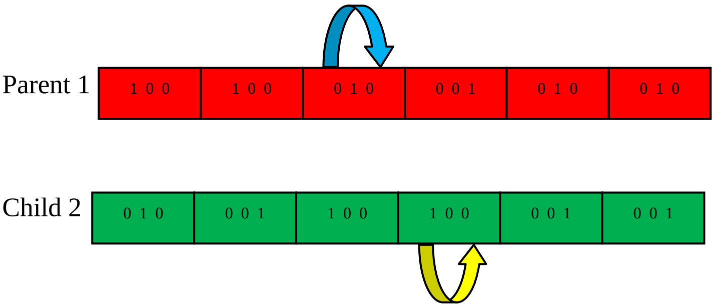

Fig. 10 Rotation Based Mutation

The Fig. 11(a&b) depicts the workflow of proposed improvised genetic approach for effectual resource allocation in cloud computing environment.

V. Results and Discussion

The enhanced genetic algorithm provides an effective resource distribution with vertical elasticity in cloud computing environment. In order to examine the performance of this proposed work, we analyzed the load variance with 50 iterations. The results are presented in Fig 12. The load variance are gradually reduced which indicates that the fitness is good.

The proposed algorithm considerably reduces the response time of virtual machines while comparing with Round Robin and Genetic algorithm for a time of day which is shown in Fig 13. This algorithm uses multipurpose mutation strategy which indirectly concentrates VM migration with vertical elastic resource allocation.

The different level of workload results show that the proposed algorithm effectively handles the workloads with low response time and vertical elasticity compared to other existing techniques. The algorithm is experimented with three different level of workload such as low, medium and high.

It is evident from the Fig. 14 that the performance of algorithm is not reduced even in high workload which supports the algorithm in providing resources effectively in cloud computing environment.

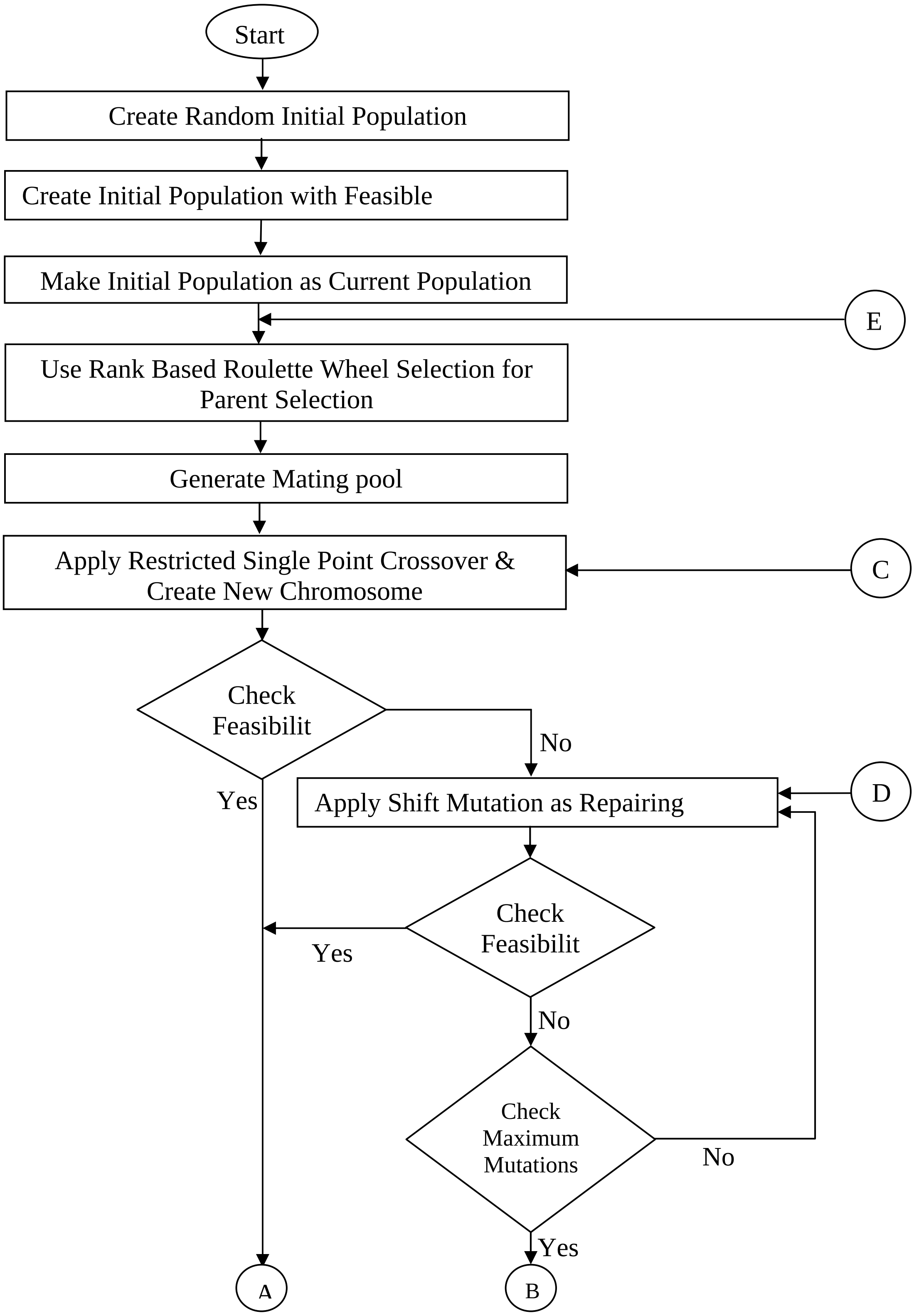
Start
Create Random Initial Population
Create Initial Population with Feasible
Make Initial Population as Current Population
E
Use Rank Based Roulette Wheel Selection for Parent Selection
Generate Mating pool
Apply Restricted Single Point Crossover & Create New Chromosome
C
Check Feasibilit
No
Yes
Apply Shift Mutation as Repairing
D
Check Feasibilit
Yes
No
Check Maximum Mutations
No
Yes
A
B

Fig. 11(a) Workflow of Proposed Algorithm

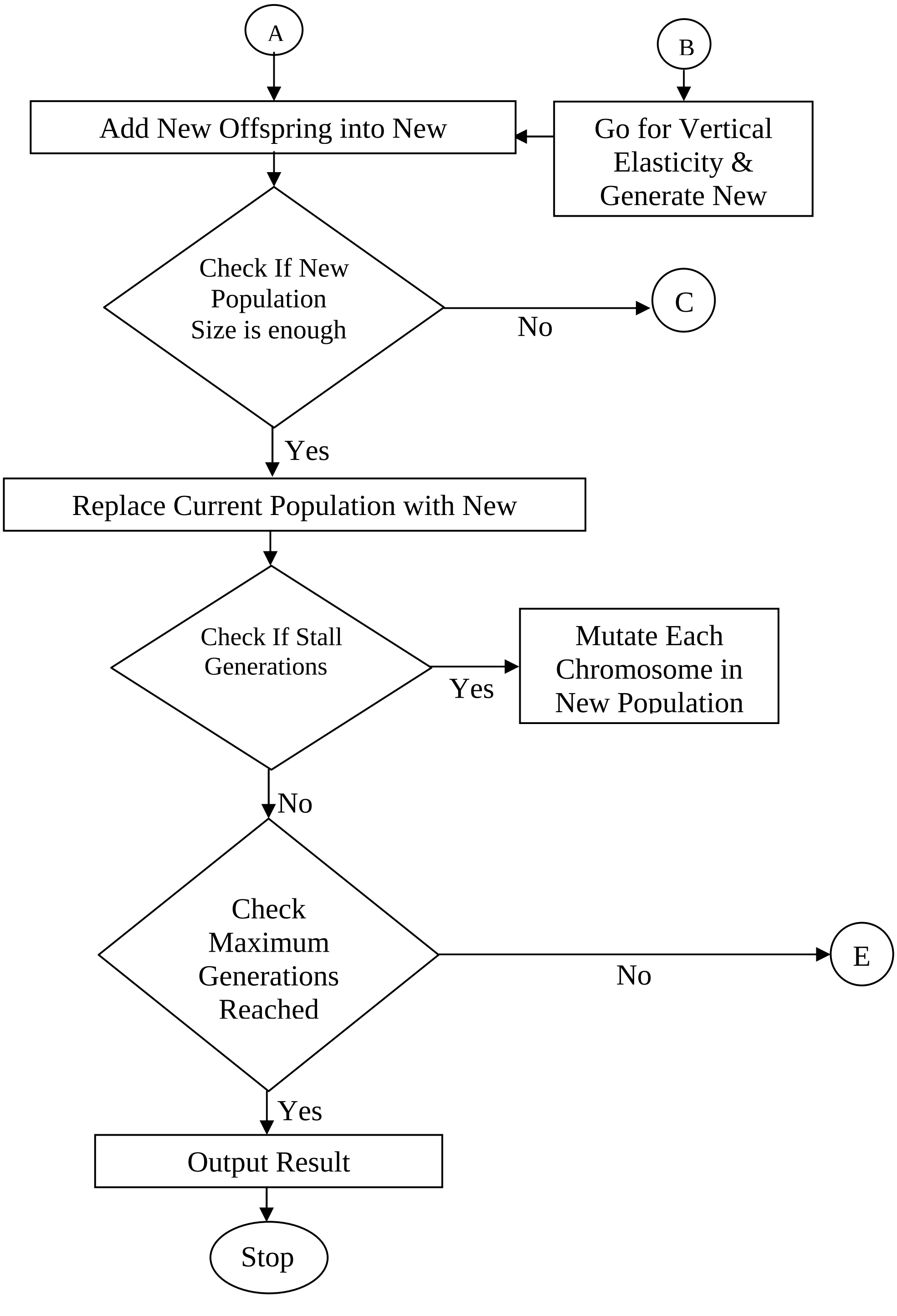
A
B
Add New Offspring into New
Go for Vertical Elasticity & Generate New
Check If New Population Size is enough
No
C
Yes
Replace Current Population with New
Check If Stall Generations
Yes
Mutate Each Chromosome in New Population
No
Check Maximum Generations Reached
No
E
Yes
Output Result
Stop

Fig. 11(b) Workflow of Proposed Algorithm

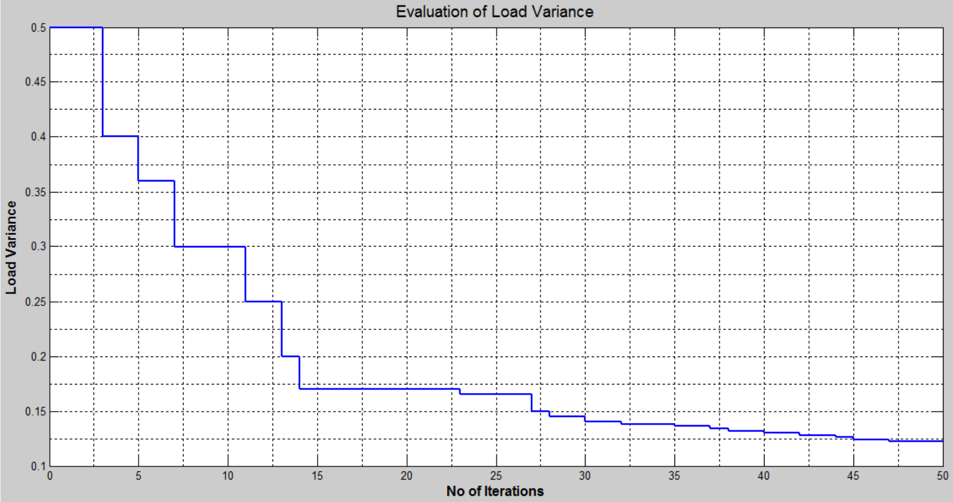

Fig. 12 Evaluation of Load Variance

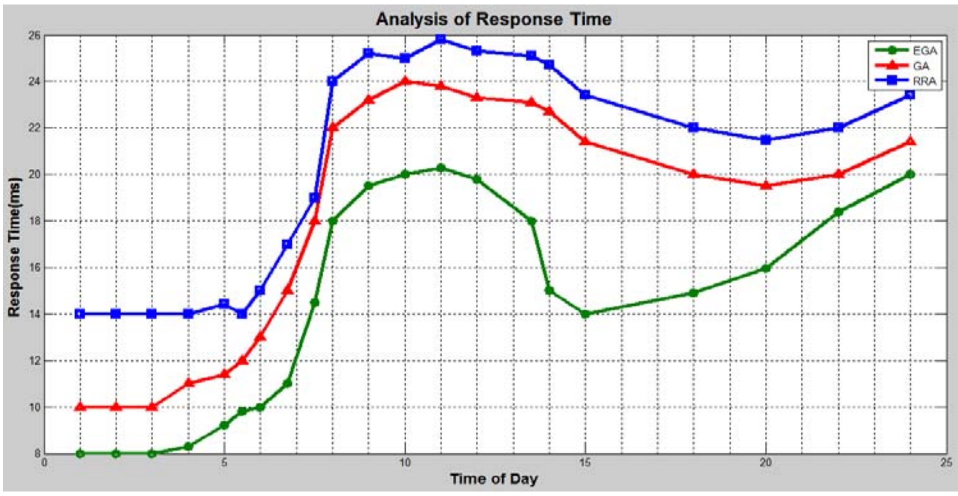

Fig. 13 Analysis of Response Time

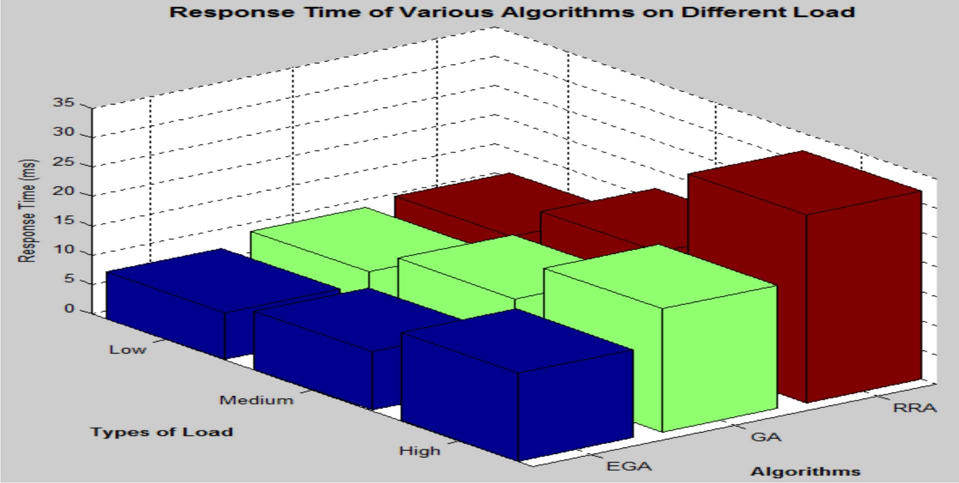

Fig. 14 Response Time of Various Algorithms on Different Load

VI. Conclusion

Nowadays, Most of the organizations are shifting towards cloud computing environment for their different services requirements in order to find an alternative solution for managing on-demand requirement of users. The organizations use rental resources instead of buying additional resources. In this paper, enhanced genetic algorithm approach is proposed for improving resource allocation in cloud computing environment. This approach reduces the execution time of algorithm and effectively handles vertical elasticity by adding resources in cloud server. The proposed approach has been evaluated by comparing with existing algorithms namely simple genetic algorithm and round robin. Moreover, the experimental results show that the performance of the proposed algorithm is not reduced even in high workload and the algorithm effectively allocates the resources. Finally, this algorithm is also focused on how vertical elasticity is effectively handled in cloud computing environment while allocating resources. The simulation results demonstrated that the proposed approach significantly ameliorates the resource allocation time which affords execution time than existing techniques. Future direction of this work is to experiment with horizontal elasticity in order to provide effective resource allocation in Cloud Computing environment.

REFERENCES

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, “Xen and the art of virtualization”, in: Proceedings of the 19th ACM Symposium on Operating Systems Principles, SOSP 2003, Bolton Landing, NY, USA, 2003.

[2] P. Mell, T. Grance, “The NIST Definition of Cloud Computing”, National Institute of Standards and Technology, Information Technology Laboratory, Technical Report Version 15, 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, “A view of cloud computing”, Communications of the ACM, Volume 53, Issue 4, pp. 50–58, 2009.

[4] P. Mell, T. Grance, “The NIST Definition of Cloud Computing”, Tech. rep., U.S. National Institute of Standards and Technology (NIST), Special Publication 800-145, 2011.

[5] M. Durairaj, A. Manimaran, "A Study on Securing Cloud Environment from DDoS Attack to Preserve Data Availability", The International Journal of Science & Technoledge, ISSN 2321 – 919X, Volume 3, Issue 2, February 2015.

[6] M. Durairaj, P. Kannan, “A study on Virtualization Techniques and Challenges in Cloud Computing”, International Journal of Scientific &Technology Research, Volume 3, Issue 11, pp.147-151, 2014.

[7] A. Ranjini, B.S.E.Zoraida, “Analysis of Selection Schemes for Solving Job Shop Scheduling Problem Using Genetic Algorithm”, IJRET: International Journal of Research in Engineering and Technology, Volume: 02, Issue: 11, Nov-2013.

[8] M. Durairaj, P. Kannan, “A Novel Approach for Elastic Application Partitioning in Mobile Cloud”, IEEE - ICAET-4th International Conference on Advances In Engineering & Technology, IEEE Xplore, 2014.

[9] A.Ranjini, B.S.E.Zoraida, “Preemptive Appliances Scheduling in Smart Home Using Genetic Algorithm”, Artificial Intelligence and Evolutionary Algorithms in Engineering Systems Advances in Intelligent Systems and Computing, Volume 324, pp 387-393, Springer, 2015.

[10] A. Vouk, “Cloud computing- issues, research and implementations”, in Proc. of Information Technology Interfaces, 2008, pp. 31-40.

[11] T. V. T. Duy, Y. Sato and Y. Inoguchi, “Improving accuracy of host load predictions on computing grids by artificial neural network”, Proceeding of IEEE International Symposium on Parallel and Distributed Processing, pp. 23-29, 2009.

[12] J. Gu, J. Hu, T. Zhao, G. Sun, “A new resource scheduling strategy based on genetic algorithm in cloud computing environment”, J. Comput. 7, pp.42–52, 2012.

[13] Yang Xu, Lei Wu, Liying Guo, Zheng Chen,Lai Yang, Zhongzhi Shi, “An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing”, in Proc. of AI for Data Center Management and Cloud Computing: Papers, from the 2011 AAAI Workshop (WS-11-08), pp. 27–32, 2008.

[14] J Gomoluch, M Schroeder., “Market-Based resource allocation for Grid Computing: a model and simulation”, In: M Endler, D Schmidt, editors. Proceedings of the first international workshop on middleware for Grid Computing (MGC 2003). Rio de Janeiro: Springer-Verlag, Vol. 6, No. 5, pp. 211-218, 2009.

[15] S. Chaisiri, B.-S. Lee, D. Niyato, Optimal virtual machine placement across multiple cloud providers, in: Proceedings of the 2009 4th IEEE Asia-Pacific Services Computing Conference, pp. 103–110, 2009.

[16] R.K. Grewal, P.K. Pateriya, “A rule-based approach for effective resource provisioning in hybrid cloud environment”, Int. J. Comput. Sci. Inform. Volume 1, Issue 4, pp.101–106, 2012.

[17] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, A. Ghalsasi, “Cloud computing the business perspective”, Decis. Support Syst. Volume 51, Issue 1, pp.176–189, 2011.

[18] A.-F. Antonescu, T. Braun, “Modeling and simulation of concurrent workload processing in cloud-distributed enterprise information systems”, in: ACM SIGCOMM Workshop on Distributed Cloud Computing, DCC 2014, 2014.

[19] N. Roy, A. Dubey, A. Gokhale, “Efficient autoscaling in the cloud using predictive models for workload forecasting”, in: Cloud Computing (CLOUD), 2011 IEEE International Conference on, IEEE, 2011, pp. 500–507.

[20] Alexandru-Florian Antonescu, Torsten Braunb, “Simulation of SLA-based VM-scaling algorithms for cloud-distributed applications”, Future Generation Computer Systems, 2015.

[21] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), Elsevier, Procedia Technology 10, 2013, pp. 340 – 347.

[22] T. EpoMofolo, R. Suchithra, “Heuristic Based Resource Allocation Using Virtual Machine Migration: A Cloud Computing Perspective”, International Refereed Journal of Engineering and Science (IRJES), vol. 2, May 2013, pp.40-45.

[23] Rajkumar Buyya , Rajiv Ranjan and Rodrigo N. Calheiros, “Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: challenges and opportunities”, in: Int. Conf. on High Performance Computing & Simulation, 2009, HPCS’09, IEEE, 2009, pp. 1–11.

[24] V.H. Nguyen, F. Dang Tran, J.-M. Menaud, “Autonomic virtual resource management for service hosting platforms”, In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD ’09. IEEE Computer Society, Washington, DC, USA, 2009, pp. 1–8.

[25] Jayasinghe, D., Pu, C., Eilam, T., Steinder, M., Whally, I., Snible, E., “Improving performance and availability of services hosted on iaas clouds with structural constraint-aware virtual machine placement”, In: 2011 IEEE International Conference on Services Computing (SCC), July, 2011, pp. 72–79.

[26] A. di Costanzo, M.D. de Assunção, R. Buyya, “Harnessing cloud technologies for a virtualized distributed computing infrastructure”, IEEE Internet Computing, 13, 2009, pp. 24–33.

[27] Rodrigo N. Calheiros, Adel Nadjaran Toosi, Christian Vecchiola, Rajkumar Buyya, “A coordinator for scaling elastic applications across multiple clouds”, Future Generation Computer Systems, 28, 2012, pp.1350-1362.

[28] M. D. de Assuncao, A. di Costanzo, and R. Buyya, “Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters,” in HPDC, 2009, pp. 141–150.

[29] Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A, “Cloud computing—The business perspective” , Decision Support Systems, 51, 2011, pp. 176-89.