Load Balancing of Virtual Machine Resources in Cloud Using Modified Genetic Algorithm

Damandeep Kaur
Department of Computer Science Engineering
Sri Sai College of Engineering and Technology, Badhani, Punjab, Pathankot

Meenakshi Sharma
Head of Computer Science and Engineering
Sri Sai College of Engineering and Technology, Badhani, Punjab, Pathankot

Sukhbeer Singh
Department of Computer Science Engineering
Sri Sai College of Engineering and Technology, Badhani, Punjab, Pathankot

Abstract: Cloud Computing is one of the important areas which are used today. With the advent of the technology cloud is being exposed to large number of users. So security is becoming the issue in cloud computing. Mobile devices are using the cloud to store the data which cannot be stored in there devices due to limited storage. Wide variety of users is using the cloud with different intensions. The problem of security has to be managed properly within the cloud. In case of cloud computing data centers are used. If load is asserted on single data center then security and reliability both are compromised. So in order to resolve the problem concept of load balancing is used. Load balancing is the mechanism of dividing the workload among various computing resources such as computers, clusters etc. Load balancing will aim to provide highest throughput and may increase reliability.

Keywords: Cloud Computing, Mobile, Load Balancing. Security, Throughput, reliability

I.INTRODUCTION

The cloud computing is one of the most commonly used technology. The cloud is used in order take a backup of the data which is used in case of mobiles and other devices. As cloud is exposed to more and more users, the security is becoming a issue. The data center is the one which is going to provide the resources to the user with effective and efficient utilization, This is known as Load Balancing. In case of Load Balancing the load will be equally distributed among the large number of data centers. No data center will going to get partial load. If data center goes down it is possible to ensure that work is not going to be stopped. In cloud computing load balancing will ensure that one resource is not overwhelmed or underutilized. The architecture of cloud will involve the following layers

Cloud Service Model

Cloud computing is a delivery of computing where massively scalable IT-related capabilities are provided ―as a service across the internet to numerous external clients. This term effectively reflects the different facets of cloud computing paradigm which can be found at different infrastructure levels.

Cloud Computing is broadly classified into three services: ―”IaaS", "PaaS" and "SaaS". Cloud Computing have some different utility services.

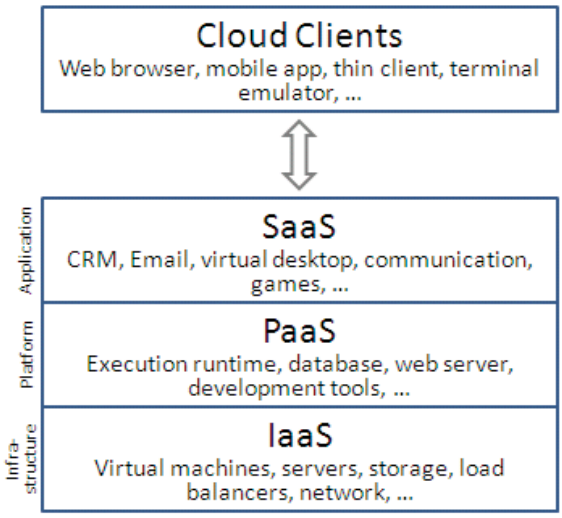

Figure 1.1 Cloud Service Models

The various service models are as follows:

IaaS (Infrastructure as a service) model: IaaS is the delivery of technology infrastructure as on demand scalable service. The main concept behind this model is virtualization where user have virtual desktop and consumes the resources like network, storage, virtualized servers, routers and so on, supplied by cloud service provider. Usage fees are calculated per CPU hour, data GB stored per hour, network bandwidth consumed, network infrastructure used per hour, value added services used, e.g., monitoring, auto-scaling etc. Examples: Storage services provided by AmazonS3, Amazon EBS. Cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Some applications of IaaS are email, video conferencing, games and Google docs.

PaaS (Platform as a service) model:

It refers to the environment that provides the runtime environment and software deployment framework. PaaS is a platform where software can be developed, tested and deployed. It means the entire life cycle of software can be operated on a PaaS. Examples: Google App Engine (GAE), Microsoft Azure, IBM SmartCloud, Amazon EC2, salesforce.com and jelastic.com and so on.

SaaS (Software as a service) model:

SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet. Through this service delivery model end users consume the software application services directly over network according to on-demand basis. For example, Gmail is a SaaS where Google is the provider and we are consumers.

Genetic Algorithm

GA is a method that optimizes a problem by iteratively trying to improve a candidates solution with concern to a given measure of quality service. Such techniques are commonly known as Norms as they make little or no assumptions about the problem being optimized and can finding very large spaces of candidates solutions.

DE optimizes a problem by maintaining traffic of candidate's solutions and creating new candidates solutions by combining existing ones , and then keeping whichever candidate's solution has the best score or fitness on the best possible solution for the problem at hand. In this way, the best possible solution for the problem is treated as a black box that merely provides a measure of quality.

A basic variation in the DE algorithm works by having traffic of candidates solutions (called agents). These Candidates are moved around in the finding-space by destination simple mathematical formulae to combine the positions of existing Candidates from the traffic. If the new position of an agent is an advancement it is accepted and forms part of the traffic, otherwise the new position is simply rejected. The process is iterated and by doing so it is hoped, but not sure, that a satisfactory solution will gradually be reached.

Formally, let $f : \mathbb{R}^n \to \mathbb{R}$ be the cost function which should be minimized or fitness function which should be maximized.
Variants of this algorithm are continually being developed in an effort to improve best possible solution for the performance. Several different schemes for performing crossover and mutation of Candidates are possible in the basic algorithm given above, see e.g. Price et al., Liu and Lampinen, Qin and Suganthan, Civicioglu and Brest et al.[1] There are also some work in making a hybrid best possible solution for the method destination DE combined with other optimizers.

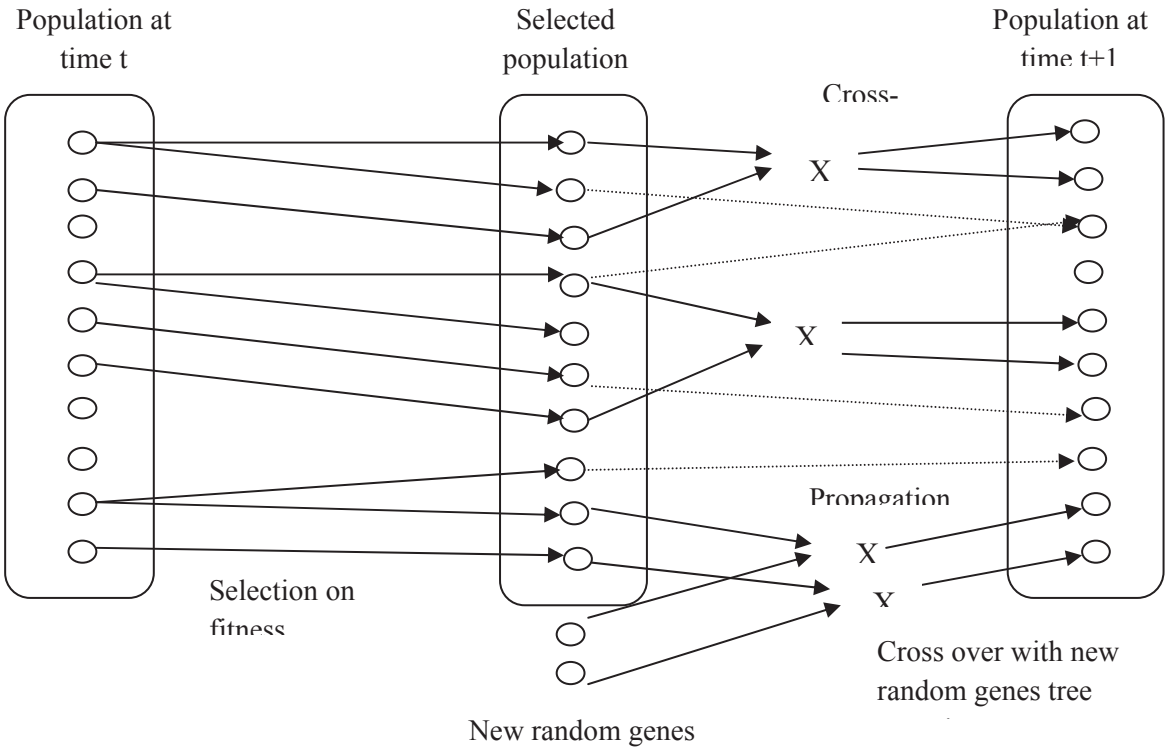

Figure 2. Pattern of selection of node

II. LITERATURE SURVEY

In this section, we provide necessary review of literature which includes graph flow problem history with optimization techniques and different mutation techniques implemented for optimization.
Rainer Storn (1997) [1] In this paper a new heuristic approach for minimizing nonlinear and non-differentiable continuous space functions is presented. As compare to previously existing global optimization method, this is a faster and have more certainty than others. This new technique requires few control variables and is robust enough.
Zhenyu Yang • Ke Tang • Xin Yao (2010) [5] In 2010 Zhenyu Yang and Ke Tang proposed that Differential evolution (DE) h as become a very powerful tool for global continuous optimization problems. Involvement of this parameter is important for these techniques to improve its performance. In this paper, they propose a generalized parameter adaptation scheme. Applying the scheme to DE results in a new generalized adaptive DE algorithm.
Janez Brest, Mirjam Sepesy,Mauˇcec (2010) [6] This paper proposes a self-adaptive differential evolution algorithm, called jDElscop, for solving large-scale optimization problems with continuous variables. The proposed algorithm employs three strategies and a population size reduction mechanism. The performance of the jDElscop algorithm is evaluated on a set of benchmark problems.
Wang, Shouzheng (2010) [10] the objective of this nonlinear optimization is minimization of system losses and improvement of voltage profiles in a power system. The proposed algorithm is implemented on the IEEE 14-bus system. To validate the effectiveness of the algorithm, the simulation results are compared with other optimization

algorithms'. This paper has presented and compared three algorithms based on swarm intelligence and evolutionary techniques for solving the reactive power optimization problem.
Zhao, Mengling, et al (2010) [11] this proposed algorithm can achieve global search and local search effectively by using of the differential evolution algorithm. For the constrained optimization problems, this paper presents a new comparison mechanism based on the concept of Pareto optimal solution.
Sun, Chengfu et. al (2012) [9] In order to increase the searching ability of DE, two modified differential evolution are merging by mechanism of quadratic approximation, Gaussian distributing, immune theory, differential evolution. First, the quadratic approximation is employed to better the performance of immune self-adaptive DE and the novel algorithm is named quadratic approximation. Also, the gaussian distributing is introduced into framework of ISDE to increase the variety of individual. The performance of proposed algorithm is tested by benchmark problems and compared with original DE and ISDE.
For instance Chandrasekaran K. et al. (2013) [4] design and implement genetic algorithm for scheduling strategy on virtual machine resources in cloud computing using its current state. The Proposed algorithm schedules VMs such that it achieves load balancing and there is less need of VM migrations.

Table 1. Summary of literature survey GA optimization and problems with different mutation techniques

Year	Authors	Paper name	Description
2009	Rammohan Mallipeddi ,Ponnuthurai,Nagaratnam Suganthan,	Differential evolution algorithm for with en sembles of paramete- rs and mutation and crossover strategries.	Differential Evolution (DE) has drawn much attention for solving numerical optimization problems. However, the performance of DE is responsive to the choice of the mutation and crossover strategies. In this paper DE with a coordination of mutation and crossover strategies and their associated control parameters known as EPSDE is implemented.[13]
2010	Yingying Yu, Yan Chen, Taoying Li	A new design of genetic algorithm for solving TSP.	In this paper, algorithm developed is able to obtain an optimal solution to TSP from a huge search space. To illustrate it more clearly, a program based on this algorithm has been implemented, which presents the changing process of the route iteration in a more intuitive way.[14]
2011	Pavlos S. Georgilakis	Differential Evolution Solution to TransmissionExpansion Planning Problem	An improved differential evolution (IDE) model is proposed for the solution of this new market-based TEP problem. The modifications of IDE in comparison to the simple differential evolution method are the following: (1) the scaling factor F is varying within some range, (2) an auxiliary set is employed to improve the diversity of

			the population[15].
2012	Omar Kettani, Faycal Ramdani,Benaissa Tadili	A quantum differential evolutionary algorithm for independent set problem	The Independent Set problem depicts to find a maximum cardinality subset of vertices of a given graph such that no two vertices are adjacent. In this paper, we propose a quantum evolutionary algorithm which uses a differential operator to update the quantum angles[16]
2012	Ma Jingyan, Zhang kehong	Research on TSP solution based on genetic algorithm of logistic equation.	The TSP is a NP problem. There are many genetic algorithm methods to solve the problem. The thesis put forward a new selective operator which can improve the genetic algorithm by heuristic information based on the changing characters of combinatorial information in TSP problem and the Logistic equation[17].
2013	Lingming Zhang Milos Gligoric Darko Marinov Sarfraz Khurshid	Operator based and random mutant selection: Better together	Mutation testing is a powerful methodology for evaluating the quality of a test suite. However, the methodology is also very costly. Selective mutation testing is a well-studied technique. Two common approaches are operator based mutant selection, which only generates mutants using a subset of mutation operators. This paper presents eight random sampling strategies defined on top of operator based mutant selection,[18]

III.PROPOSED SYSTEM

In the proposed system the genetic algorithm will be used in order to enhance the load balancing mechanism. The genetic algorithm will have number of phases associated with it. The proposed system will make the changes in the crossover function and concept of distance will be introduced in this case. The distance will be used as a parameter in order to select the VM which is closest to the current machine from where data is to be offloaded. The redundancy handling mechanism will also be used in order to reduce the cost associated with the system.

IV.RESULT AND EXPERIMENT

The implementation of the system will be conducted using the CloudSim. The cloudsim in the proposed system will be integrated with the NETBEANs software. The result of the simulation is as listed below

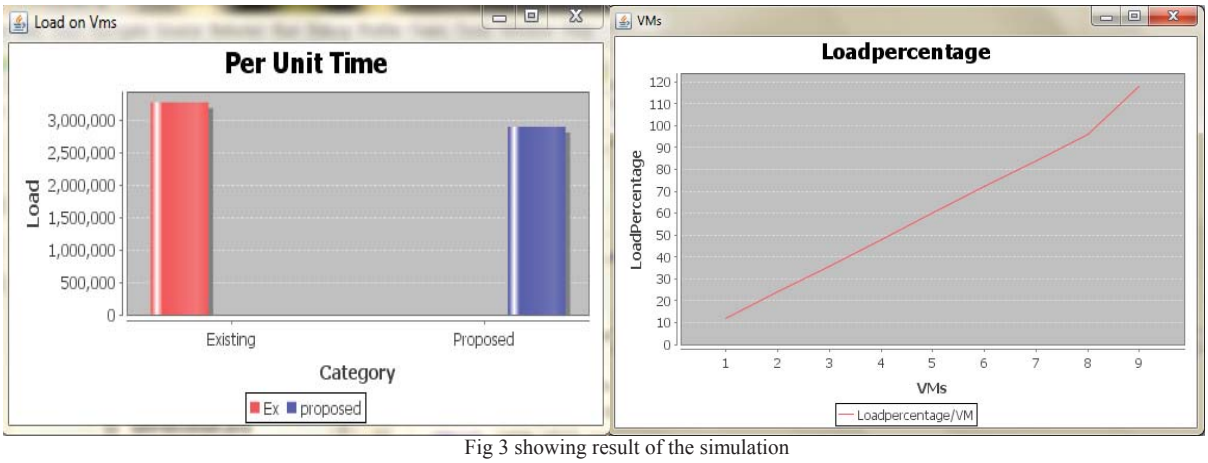

Fig 3 showing result of the simulation

V.CONCLUSION

From the above comparison it is clear that the GA can provide very effective load balancing mechanism. However the work which is done does not consider the network I/O , Storage I/O, Overall cost and Power consumption. In the future work all of the above said parameters must be consider.

REFERENCES

[1] Rainer Storn, November (1997) "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces" Journal of Global Optimization **11:** 341–359, 1997

[2] The back and H.P. Schwefel (1993) "An overview of evolutionary algorithms for parameter optimization. Evolutionary computation" 1(1):1-23; classic paper "unified the field".

[3] Matthieu Weber _ Ferrante Neri _ Ville Tirroren, "Shuffle Or Update Parallel Differential Evolution for Scale Optimization"(2008)

[4] Chandrasekaran K. and Usha Divakalra, "Load balancing of virtual machine resources in cloud using genetic algorithm", National Institute of Technology Karnataka, Surathkal(2013),pp. 156-168.

[5] Zhenyu Yang, Ke Tang, Xin Yao;Scalability of generalized adaptive differential evolution for large-scale continuous optimization, Soft Computing(2010)

[6] Janez Brest · Mirjam Sepesy Mau˘cec(2010);Self-adaptive Differential Evolution Algorithm using Population Size Reduction and Three Strategies

[7] S. Z. Zhao, P. N. Suganthan, *Snr Member, IEEE* and S. Das(2011);Self-adaptive Differential Evolution with Multi-trajectory Searchfor Large Scale Optimization

[8] A. LaTorre _ S. Muelas _ J.M. Pe~na(2011);A MOS-based Dynamic Memetic Differential Evolution Algorithm for Continuous Optimization ;A Scalability Test

[9] Sun, Chengfu, Haiyan Zhou, and Liqing Chen"Improved differential evolution algorithms." Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on. Vol. 3. IEEE, 2012.

[10] Wang, Shouzheng, Lixin Ma, and Dashuai Sun. "Hybrid differential evolution particle swarm optimization algorithm for reactive power optimization." Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific. IEEE, 2010

[11] Zhao, Mengling, et al. "Multi-objective Optimization Based Differential Evolution Constrained Optimization Algorithm." Intelligent Systems (GCIS), 2010 Second WRI Global Congress on. Vol. 1. IEEE, 2010

[12] Ye, Sishi, et al. "A hybrid adaptive coevolutionary differential evolution algorithm for large-scale optimization." Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 2014.

[13] Rahnamayan, Shahryar, Hamid R. Tizhoosh, and Magdy MA Salama. "Opposition-based differential evolution." Evolutionary Computation, IEEE Transactions on 12.1 (2008): 64-79.

[14] Noman, Nasimul, and Hitoshi Iba. "Accelerating differential evolution using an adaptive local search." Evolutionary Computation, IEEE Transactions on 12.1 (2008): 107-125.

[15] Tušar, Tea, and Bogdan Filipič. "Differential evolution versus genetic algorithms in multiobjective optimization." Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2007.

[16] Das, Swagatam, Ajith Abraham, and Amit Konar. "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives." Advances of Computational Intelligence in Industrial Systems. Springer Berlin Heidelberg, 2008. 1-38.

[17] Chao-Xue Wang, Chang-Hua Li, Hui Dong and Fan Zhang, 2013. "An Efficient Differential Evolution Algorithm For Function Optimization." Information Technology Journal, 12: 444-448

[18] A. K. Qin, V. L. Huang, and P. N. Suganthan," Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization." 1089-778X/$25.00 © 2008 IEEE