ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 4, April 2018

Genetic Algorithm based Load balancer Resource Allocation in Intercloud Environment

[1] N.Krishnaveni, [2] Dr.S.Nagaprasad, [3] Dr.Manju Khari

[1] Research Scholar-JJT University, Rajasthan

[2] Faculty of Computer Science, Dept. Of Computer Science, S.R.R.Govt.Arts & Science College, Karimnagar, Telangana State

[3] Computer Science Department, Ambedkar Institute of Advanced Communication Technologies and Research, Delhi.

Abstract: - **Load balancer using Genetic Algorithm plays a crucial role in the performance of Intercloud platform. Resource allocation needs accurate and thoughtful precision in performance, as incoming applications could be impatient and erratic, resources will be limited and time constraint will be steep. Establishing trade off between demand and supply using available resources is tough. As the rate of arrival of application is sporadic, deadlines are stern and resources are restricted in Intercloud, it calls for an efficient resource allocation algorithm. Genetic Algorithm-based resource allocation shows better result than traditionally used FIFO. Decision variables considered are making span time and throughput.**

***Keywords*:** **Genetic Algorithms (GA), FIFO. Broker, SLA (Service Level Agreement).**

I. INTRODUCTION

Intercloud computing paradigm has emerged as a futuristic technology to facilitate the customers across all domains with Infrastructure as a service (IaaS), Platforms a service (PaaS) and Software as a Service (SaaS). The cloud customers can subscribe instances as per their requirement and pay for the services on the basis of the usage. Intercloud is defined as practice of interconnecting and interoperating cloud computing environments of multiple service providers for the load balancing objective and accommodating spikes in demand. Cloud providers integrate resources to serve customers better without over-provisioning of resources. The sudden confounds in the cloud user's requirement can be managed in un-interruptible manner, by associating multiple cloud providers. Intercloud is beneficial to parties, customers and the service providers involved in the federation. The customers can choose any Intercloud federation of choice, instead of selecting a handful of 'global' cloud provider sin the market. The participating cloud provider can connect to the federal market place that enables each participant to buy, lease and sell the computing capacity on demand. The Intercloud component called cloud broker is the gateway which register the incoming applications. Figure 1.Representscloud broker architecture on which federation works. It may differ organization-wise. The cloud providers sometimes have spare capacity in the form of physical machine or VMs which can be monetized by submitting it to the cloud broker of Intercloud federation for others to reserve, lease, creating an additional mode ofrevenue.Load balancing in Intercloud computing is divided into two important tasks, first is resource allocation and second is task scheduling in the distributed cloud environment an efficient load balancer ensures that:

- Resources are easily available on demand.
- Efficient utilization of resources under condition of high/low workload.
- Consolidation of resources

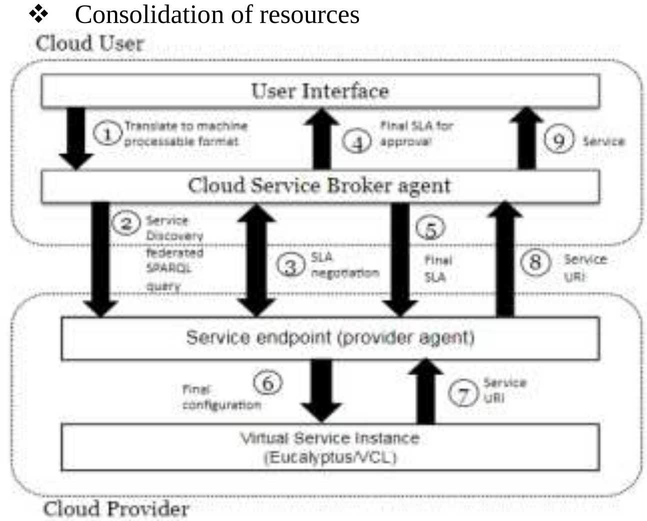


Figure 1.Cloud broker architecture

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

GA-based load balancer is used to perform the above mentioned objectives. The main task of the broker in Intercloud is to maintain the balance between the demand of the incoming applications and the supply of the available resources. In Intercloud, cloud vendors can associate in cloud federation using brokering of resources using the information stored in cloud repository. The cloud providers associated in the federation broadcasts their requirement or resources to spare to cloud broker iteratively. Historical performance of cloud providers and its constituting components are stored.

II. INTERCLOUD AND ITS COMPONENTS:

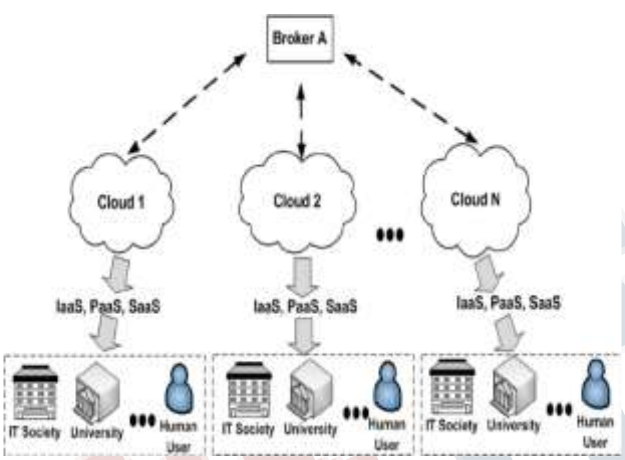


Figure 2: Intercloud Architecture

Intercloud is an e-integrated infrastructure having extended computational capability for utilization of resources and storage capacity. It is like a seamless functionality mesh where each node is a cloud or its component represented in Figure 2. It illustrates the interrelationship between servers, physical machines and the VMs. The interoperation between multiple clouds needs to be in compliance among its member. It is a powerful platform, which can be real rewarding tasks when its strength is exercised efficiently. Here application is a service to be executed by Intercloud for any user/enterprise. Each application is divided into small tasks. Each task can be executed by one or multiple clouds. The objective is an optimum utilization of resources in accordance with SLA by using GA-based resource allocation. Presently; most of the research work on load balancing in Intercloud is based on dynamic scheduling and migration of VMs [1, 2]. Moving to Intercloud is the next best logical option for betterment of both providers and users perspective in order to attain optimum resource allocation in the cloud environment.
Storage service architecture:

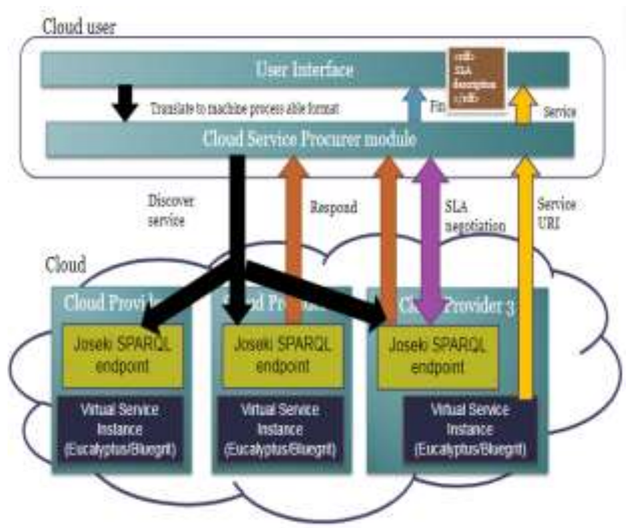


Figure 3: Storage Service Architecture

In the above figure it demonstrates the storage service, how the user stores the data among cloud servers without violating the SLA (Service Level Agreement).

III. SYSTEM MODEL:

In the Intercloud environment the resource allocation of VMs mainly considers the granularity of incoming applications. This chapter provides a resource allocation strategy to enable effective load balancing. Load balancing is achieved using genetic algorithm-based resource allocation. Genetic algorithm (GA) is a population based heuristic optimization algorithm [3, 4]. It is divided into few phases to attain the best solution. It instructs the searching criteria and adjusts the searching direction automatically using genetic operators like crossover and mutation. GA works better when search space is large.

IV. PROPOSED SOLUTION:

Developed from Darwin's survival strategy and instructed by genetic operators, GA is a competitive optimization tool. Each possible solution is termed as a chromo some. Several chromosomes are populated which constitute the population of the generation. New population is generated when genetic operators act on the chromosomes of the previous generation. Encoding is the process of designing chromosome in such a manner that the tasks and the resources are encoded in one chromosome. Each chromo some is represented as a 2 ×M matrix, where, M is the chromosome length. The first row of the matrix represents the requested task, and the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

second row of the matrix corresponds to the cloud where the application is executed. Here applications are registered, assigned a unique id, and then the cloud broker maps the tasks to available resources using GA. The length of chromosome depends on the number of tasks in the batch. If 25 tasks are collected in a batch then the chromosome is 2 × 25.For simulations, 300-400 applications are generated from different users. The length of application, tasks, and server capacity are expressed in MIPS. The phases of GA are repeated iteratively till convergence criteria are met [5, 6, 7].

Steps of Genetic Algorithm are as follows:

Encoding - Chromosomes are generated by random allocation of tasks to VMs.
Fitness Evaluation - Fitness value of each chromosome is calculated in the population.
Selection - Two chromosomes are selected from the population pool as parent chromo some.
Crossover - Offspring's are generated from parent chromosome using cross over operator and crossover probability.
Mutation - New chromosome is obtained considering the mutation probability and changing allocation randomly.
Encoding of the problem plays a vital role. Based on the nature of the problem, encoding is done to map the problem statement as a chromosome. A number of tasks which needed to be allocated
resources in a batch is represented as a chromosome.
In the first generation number of tasks in a batch is randomly allocated to available VMs. Fitness value of each chromosome is calculated. Then the selection procedure is used to decide two selected chromosomes as parents. Elitism selection is considered. In elitism selection; the chromosome having highest fitness value is selected for the next generation mating pool.

Initial Population: The first phase of genetic algorithm involves encoding of the chromosome. The number of chromosomes in the initial population is equal to the population size. The chromosomes are selected from the initial population and genetic operators are applied on them to form the chromosomes for next generation.

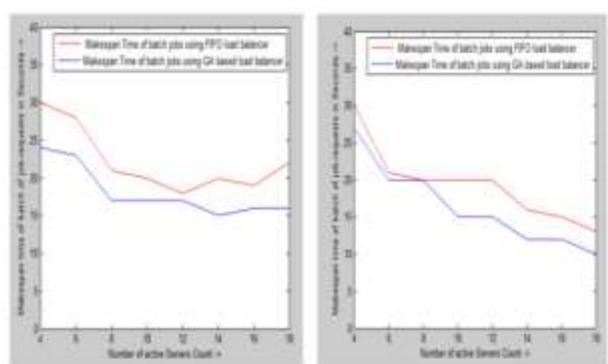
Fitness Evaluation: The fitness value of any chromosome depicts the productivity of individual chromosome. The probability of contribution of the chromosome in the next generation depends on the fitness value.

Selection: The phase of GA used to select the pair of the chromosomes for genetic operators is called selection. This intermediate phase is important in Darwin's theory of survival. There is various selection procedures practiced to select the chromo some such as Roulette wheel, Boltzmann strategy, Tournament selection and Rank selection. Elitism selection process is considered. The best fit chromosome is moved to the next generation as it is so that the chromosome having highest fitness value is not lost in the process of GA [9, 8].

Crossover: The crossover operation is done by selecting two parent chromo some from the population and then creating a pair of child chromosome by alternating and re-joining the parts of the parent's chromosome.
Mutation: The genetic operator which boosts fitness value of chromosomes is termed as mutation. It is the vital operation which addresses premature convergence to a large extent. The value of mutation probability is usually set to considerably low value.

V. RESULTS:

GA is used to attain optimal trade off between rate of arrival of applications and the number of active physical machines. Initialization is done randomly to avoid premature convergence and GA operators are selected by rigorous simulation. The model suggests the best VM-task pair. It also suggests the number of active server for better resource allocation. By gathering all of the conceivably optimal solutions, a designer can decide on the number of physical and VMs required for smooth functioning in Intercloud environment.
In all cases, GA load balancer is more efficient than FIFO. The strength of our model is that it improvises the resource allocation procedure in fast and efficient way. The rate of emptiness of queue is high (the same batch of job-request takes comparatively lesser time when GA based load balancer is used) in the case of GA than FIFO.



Rate of arrival=30 Rate of arrival=40

Figure 4: Results of comparing GA and FIFO

VI. CONCLUSION:

Load balancer using GA plays a crucial role in the performance of Intercloud platform. Resource allocation needs accurate and thoughtful precision in performance, as incoming applications could be impatient and erratic, resources will be limited and time constraint will be steep. Establishing trade off between demand and supply using available resources is tough. As the rate of arrival of application is sporadic, deadlines are stern and resources are restricted in Intercloud, it calls for an efficient resource allocation algorithm. GA-based resource allocation shows better result than traditionally used FIFO. Decision variables considered are making span time and throughput.

References:

[1]. S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu, and A. Abraham, "Hybrid job scheduling algorithm for cloud computing environment," in Proceedings of the Fifth International Conference on Innovations in [2]. Bio-Inspired Computing and Applications IBICA, pp. 43–52, Springer, 2014.
[3]. H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in Proceedings of the 20th international symposium on High performance distributed computing, pp. 171–182, ACM, 2011
[4]. S. Chen, J. Wu, and Z. Lu, "A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness," in IEEE 12th International Conference on Computer and Information Technology (CIT), pp. 177–184, IEEE, 2012.
[5]. N. A. Mehdi, A. Mamat, H. Ibrahim, and S. K. Subramaniam, "Impatient task mapping in elastic cloud using genetic algorithm," Journal of Computer Science, vol. 7, no. 6, pp. 877–883, 2011.
[6]. K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A genetic algorithm (ga) based load balancing strategy for cloud computing," Procedia Technology, vol. 10, pp. 340–347, 2013.
[7]. P. Chand and J. R. Mohanty, "Multi objective genetic approach for solving vehicle routing problem with time window," in Trends in Computer Science, Engineering and Information Technology, pp. 336–343, Springer, 2011.
[8]. P. Chand and J. R. Mohanty, "Environmental multi objective uncertain transport trail model using variant of predator prey evolutionary strategy," International Journal of Applied Decision Sciences, vol. 8, no. 1, pp. 21–51, 2014.
[9]. S. A. Hamad and F. A. Omara, "Genetic-based task scheduling algorithm in cloud computing environment," International Journal of advanced Computer science and applications, vol. 7, no. 4, pp. 550–556, 2016.
[10]. T. T. Tejaswi, M. Azharuddin, and P. K. Jana, "A ga based approach for task scheduling in multi-cloud environment," Journal of Supercomputing, Springer, 2015.