ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 11(15), DOI: 10.17485/ijst/2018/v11i15/118729, April 2018*

Load Balancing using Cluster and Heuristic Algorithms in Cloud Domain

Rajani S. Sajjan and Rekha Y. Biradar*
Department of Computer Science and Engineering, V.V.P.I.E.T., Solapur – 413004, Maharashtra, India;
rajanisajjan78@gmail.com, rbiradar20@gmail.com

Abstract

Objectives: Enterprises are using cloud services to improve the performance as well as to reduce the overall cost. As access to cloud resources increases, the load on servers increases exponentially. In enhancing the usage and utilization of cloud services, load balancing is essential. In the proposed system, cluster and heuristic algorithms are combined to reduce the makespan of tasks and to balance the load in cloud environment. **Methods/Statistical Analysis:** Initially in cloud environment, traditional load balancing algorithms are used to balance the load on servers. The solution or result obtained by such algorithms is not the perfect one. Glaring algorithms are required for load balancing as well as to obtain most favorable solution or solution which is most appropriate for the given problem. Heuristic algorithms significantly balance the milestone on the servers by giving a more optimal solution quickly and efficiently. **Findings:** Proposed system simulates the load balancing system that partitions the virtual machines and schedules the task using heuristic algorithms by considering task requirements. **Application:** This paper summarizes the survey of various heuristic algorithms. We have proposed the task based approach towards load balancing (TB-LB) in cloud environment that uses k-means clustering approach to organize Virtual Machines (VMs) into groups and it combines the features of three heuristic algorithms to increase the speed and to keep the makespan small as possible.

Keywords: Cloud Computing, Clustering, Heuristic Algorithms, Load Balancing Algorithms

1. Introduction

Today, modern technologies use cloud system to provide improved performance. Cloud systems provide high availability of computing resources at high speed and are more cost-effective. A cloud system is combines the multiple servers and for execution or processing of tasks it preserves the communication between servers. The cloud delivers on-demand computing resources over the Internet on a metered basis[1,2]. Cloud computing provides flexibility, efficiency, availability and scalability at low cost. It delivers cloud services in three forms such as software, platform and infrastructure using four types of clouds namely private, community, public and hybrid. Organizations are growing day by day[1,2]. Many organizations are using cloud to reduce their infrastructure cost. So the load on server increases exponentially. This load may be distributed unevenly. So some nodes are under loaded and some are heavily loaded. Imbalanced load further reduces the system performance[3]. Load balancing leaves its influence on response time, execution time, scalability, throughput as well as performance of the system[3].

To balance the load on servers, to enhance the quality of service (QoS), and to produce best running time at reasonable cost job scheduling is essential in a cloud environment[4]. To improve the performance of the cloud system, scheduling is significant factor. Traditional load balancing algorithms are handy and straight forward. These algorithms are inappropriate to handle difficult scheduling problems in order to produce the desired results or solution[5] as the results produced by such algorithms are far from optimal solution.

Today, many of the researchers are attracted towards heuristic algorithms to improve scheduling performance in the cloud system. Heuristic algorithm plays a very

*Author for correspondence

important role in scheduling to balance the load on cloud computing Systems. Heuristic algorithms provide several practical based frameworks in discovering an ideal solution which is sufficient but it is not the perfect or best[5]. In this paper, section 2 describes load balancing. Section 3 and 4 describes heuristics algorithms and its analysis. Section 5 describes proposed system and section 6 gives conclusion.

2. Load Balancing

Load balancing has very much impact on the system performance. It is the process of allocating the workload equally among all the nodes. Proper load balancing aims to achieve high resource utilization, improved throughput, faster response, improved system performance, etc. It also minimizes carbon emissions caused due to the improper utilization of resources[3]. There are various load balancing techniques. Figure 1 shows the categories of load balancing algorithms. It can be static or dynamic. Static load balancing algorithms are used in correlative and unchangeable environment, while dynamic load balancing algorithms are used in miscellaneous and changing environment[3]. Any change in system state is not considered by static algorithm. Hence it is not flexible. Round robin, weighted round robin, min-min, min-max, etc. are examples of static load balancing algorithms. Any change in system state is considered by dynamic algorithms. Hence it is flexible. Dynamic algorithms can be distributed or non-distributed. Honey bee foraging behavior, throttled, ESCE (Equally Spread Current Execution), Ant colony, etc., are examples of dynamic load balancing algorithms[3].

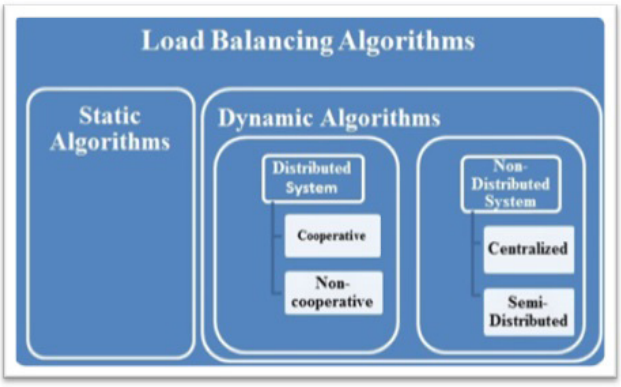

Figure 1. Categories of Load Balancing Algorithms.

In distributed System, the task of load balancing is distributed among all the nodes. If all nodes work together to perform tasks of load balancing then it is known as cooperative distributed system. If every node works individually to perform task of load balancing then it is known as non-cooperative distributed system[3]. In centralized non-distributed system, load balancing of the entire system is performed by central node and all other nodes interact with the central node. In semi distributed system, the complete system is divided into groups and one central node is selected for each group which performs tasks of load balancing of that group[2]. Load balancing requires optimum or near optimum solution. A heuristic algorithm aims to provide solutions which are near to optimum.

3. Heuristics Algorithms

Heuristic algorithms are used to discover a solution among the set of all possible solutions. These algorithms do not give assurance that the obtained solution will be the best. The search space for given problem is set of all possible solutions. The solution is approximate and not the most accurate. Sometimes these algorithms may provide the accurate result. The results produced by these algorithms are near to the actual[6]. The heuristic algorithms suggest approximate solutions for optimization problem. Many real world problems or issues are referred as an optimization problem. The purpose of the heuristic algorithm is to discover an acceptable solution as possible for all occurrences of the given problem. As the speed of the processing to get the solution is as important as the quality of the solution for heuristic algorithms. A good heuristic algorithm provides the solution which is near to optimal with reasonable computational effort[6].

Balancing of load in cloud domain is very complex real world problem that cannot be easily solved. The load can be balanced efficiently through Heuristic algorithms. Meta heuristic algorithms uses search space of given problems solution and discovers the appropriate solution. Hyper-heuristics search within a search space of heuristics. Hyper-heuristics algorithms have been applied to variety of problems, such as the travelling salesman problem, job scheduling, vehicle routing problem, timetable, and many more. Hyper-Heuristics algorithm combines the features of two or more Heuristics algorithms to provide good results.

Heuristic algorithms such as genetic algorithm, ant colony optimization, simulated annealing, particle swarm optimization and artificial bee colony optimization. The results given by heuristics based algorithms are within the reasonable time and are optimal[7].

3.1 Genetic Algorithm (GA)

Genetic algorithms are encouraged from Darwin's theory about evolution. GA is population-based heuristic algorithm used to solve complex optimization problems. The algorithm is loaded with a set of solutions generated randomly called a population. A population is generated based on old solution which is selected according to their fitness. Newly generated population will be better than the old population. This is repeated until the best solution[8]. New population can be generated by repeating following three operations: selection, crossover and mutation[8].

- **Selection:** Two good populations are selected according to their fitness[8].
- **Crossover:** This operation is performed to exchange parts of two parents to generate two better children using one or two crossover points[8].
- **Mutation:** After the crossover operation mutation operation is performed. Selected bits are inverted or exchanged[8]. Some of the bits are thrown randomly.

Following are the some areas where genetic algorithms can be applied:
1) Scheduling
2) Combinational optimization
3) Data Mining
4) Image Processing

Advantages:
1) Used when the search space is large
2) Chances of getting optimal solution are more
3) Better load balancing
4) Minimizes the makespan

3.2 Simulated Annealing (SA)

Simulated annealing is also heuristic algorithm used to find a good solution for complex optimization problems. When the objective is to minimize or maximize something, then SA will provide a solution which nears to optimum. This algorithm is stimulated from the heat treatment applied in material science to increase the size and to reduce the hardness by cooling[9]. SA generates a random trial point for each of the iteration and then shifts it. The algorithm compares the new with current point. SA makes the new point as the next point, if is better than the current point. Depending on the result of acceptance function, SA accepts or rejects a worse point[10]. Accepting worse solutions is decreased slowly through heating and cooling effect[9]. Combinational optimization problems can be solved by using SA.

Following are the some areas where simulated annealing can be applied:
1) Scheduling
2) Combinational optimization
3) Data Mining
4) Chemical Engineering
5) Image Processing

Advantages:
1) Used when search the space is large and discrete
2) Provides better results

3.3 Particle Swarm Optimization (PSO)

PSO is population-based global search and swarm based intelligence heuristic algorithm used to solve tasks scheduling problems in cloud computing. This algorithm is inspired from social behavior of a flock of birds. PSO is initialized with a population of random solutions, and then it searches for optima by updating generations[11,12]. The particles follow the current optimum particles while flying through the problem space to get an optimized solution. The potential solutions received are termed as particles. Types of PSO are multi-objective PSO, bi-objective PSO, hybrid PSO, etc.[12]. Depending upon the best position and best particle of population, every particle can adjust its flying path. This algorithm rapidly gathers comprehensive least possible solution and provides a reasonably good solution[12].

Following are some areas where particle swarm optimization can be used[12]:
1) Environmental Engineering
2) Scheduling
3) Combinational optimization
4) Data Mining
5) Image Processing
6) Chemical Engineering

Advantages:

1) Search speed is fast.
2) Provides better results
3) Minimizes the response time
4) Provides faster convergence

4. Literature Survey

In [13], author has proposed a method in which Genetic Algorithm is used for load balancing. The population initialization is based on priority of request. Priority of request is calculated from the length of the job. The simulation of proposed system is done in Cloud Analyst.

In [14], author has proposed a method in which Genetic Algorithm is used for load balancing. The population reduction method is used. Original population of resources is divided into equal groups. Then Tournament Selection method is applied on every group to select best resource from that group and new population is generated. The simulation of proposed system is done in CloudSim.

In [15], author has proposed a method based on Multi-Population Genetic Algorithm (MPGA). In this algorithm, min-min and min-max algorithm is used to generate initial population. The immigration and elite reservation operations are used in MPGA. The simulation of proposed system is done in MATLAB.

In [16], author has proposed algorithm using particle swarms based to minimize the makespan. Positions of particles are initialized randomly. The solution of problem is termed as particle and dimension of particles is number of tasks. The proposed system is simulated using CloudSim.

In [17], author has proposed an Improved Particle Swarm Optimization (IPSO) algorithm. This algorithm read just s position of particle and velocity of particle for updating its fitness value.

In [18], author has proposed an Improved Adaptive Heuristic Algorithm (IAHA). This algorithm makes tasks prioritization based on graph topology. Hence, reduction in completion time of application. It is based on adaptive crossover rate and mutation rate to perform crossover and mutation operation. IAHA is able to provide optimized solution.

In [19], author has proposed a virtual machine placement method based on Improved Simulated Annealing algorithm (ISA). Here two models are such as resource utilization model of servers and dynamic placement model for virtual machines are presented. The simulation of proposed system is done in CloudSim 3.0.

In [20], author has proposed an algorithm to discover the best resource allocation in cloud domain. Proposed system named as Simulated Annealing Load Balancing (SALB). This algorithm minimizes the degree of imbalance. The simulation of proposed system is done in CloudSim.

Table 1 shows the analysis based on response time provided by GA using PRM[14] and PSO[16]. The simulation of both algorithms listed in Table 1 is done using 50 virtual machines and number of task 100. Similarly, Table 2 shows the analysis based on degree of load balance in IPSO[17] and IAHA[18]. The simulation of both algorithms listed in Table 2 is done using 100 tasks.

Table 1. Analysis based on response time

	GA using PRM[14]	PSO[16]
No. of Virtual Machines	50	50
No. of tasks	100	100
Response Time in millisecond	561.55	147

Table 2. Analysis based on degree of load balance

	IPSO[17]	IAHA[18]
No. of tasks	100	100
Degree of load balance	0.4	2.7

5. Proposed System

The proposed system task based approach towards load balancing (TB-LB) in a cloud environment combines features of three heuristic algorithms such as genetic algorithm, simulated annealing, and particle swarm optimization to make the smaller makespan. It organizes the available VMs into groups using clustering approach. The clustering approach used in proposed system is k-means. As per the jobs requirement, the global load scheduler selects the suitable/best cluster and sends the job to the selected cluster. Cluster load balancer selects the available VM by using a proposed load balancing algorithm and assigns the job the

selected VM for execution. Figure 2 shows the architecture of the proposed system. Proposed TB-LB system takes the advantages three heuristic algorithms.

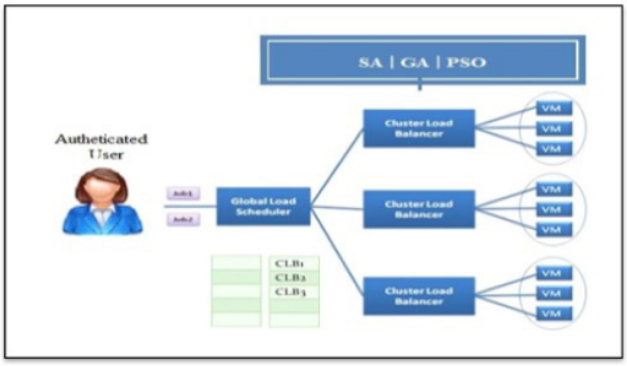

Figure 2. Proposed (TB-LB) System Architecture.

After successful login, VMs list is initialized with VM properties such as Central Processing Unit (CPU), memory, and bandwidth. VMs are clustered. Clusters are parallel and distributed system. Clusters are governed under the supervision of single administrative domain. Clustering is a task of organizing set of VMs into groups in such a way that VMs belong to a group (cluster) are similar in some way. In k-means clustering, the available set of data is divided into k number of groups. It is a partitioning method. Using this algorithm, the proposed system divides and allocates the available VMs to k number of groups. The number of groups is equal to k. To assign VMs to the k clusters Euclidean Distance (EUD) formula is used. Two formulas are used to perform the task of VM allocation to a particular group[21].

To calculate distance of VM i with cluster center j

$$EUD(VMi)(Cj) = \sqrt{(CPUi - CPUj)^2 + (Memi - Memj)^2 + (BWi - BWj)^2}$$

To calculate new mean of each cluster when a new VM is allocated to a cluster

$$CPUj = (CPUi + CPUj)/2$$
$$Memj = (Memi + Memj)/2$$
$$BWj = (BWi + BWj)/2$$

Here, CPU represents the Central Processing Unit, Mem represents the Memory and BW represents the Bandwidth. And 'i' represents the VM that needs to be clustered and 'j' represents the VM as cluster center. Through clustering we can implement virtualization. In the proposed system, we have clustered virtual machines.

Advantages of clustering of virtual machines:

1) There is no need of extra hardware.
2) Control and monitoring is easier.
3) Troubleshooting is also easier.
4) Flexibility
5) High availability
6) Cost efficient
7) Can be used in cloud based simulation

A login generally requires the user to enter two pieces of information, first a user name and then a password. Authenticated user is allowed to request the jobs for execution. All cluster information is stored at Global Load scheduler. As per the Jobs requirement (memory, CPU, bandwidth, etc.), Global Load scheduler selects the suitable/best cluster and sends the job to the selected cluster. Cluster Load Balancer maintains a table that stores information about VMs in that cluster. Now Cluster Load Balancer schedules the jobs by using proposed load balancing algorithm. Due to clustering technique, less time is required to find the suitable VM. In the proposed system, we have changed the way of using the heuristic algorithm by assigning priority to it. Figure 3 shows the algorithm for proposed system. For each of the iteration, one algorithm is picked as heuristic algorithm from the pool of heuristics algorithms. Improvement detection function checks the improvement in makespan. Diversity detection function automatically decides when to change the algorithm and selects algorithm from pool. Perturbation function optimizes the results produced by each of these algorithms.

Algorithm 1: Algorithm for proposed TB-LB

Input: Input VM, Input Task
Output: The Best So For Makespan (BSFMK) as the final solution

```
Login using username and password
Initialize VM list
Cluster n VMs into K clusters using K-means clustering
Input Task
Initialize the population Z={z1,z2,...,zn}
DO
    IF i = 0
        Schedule task with randomly selected a hueristic algorithm from algorithm list containing SA, GA and
    PSO
    ELSE
            Schedule task with algorithm having highest priority
            Update the population of solutions Z by using the selected algorithm.
    Compute F1=Improvement Detection(Z).
    Compute F2=Diversity Detection(Z).
    Compute F3=Pertub(Z)
    IF Pertub(Z) = TRUE
        Assign lowest priority to current algorithm
WHILE i ≤ 50
END
```

Figure 3. Proposed (TB-LB) System Architecture.

The hardware and software requirements of the proposed system are listed as follows:

1) Hardware Requirements:

Memory	:	2GB or above
Processor	:	Intel Core i3/i5/i7 CPU

2) Software Requirements:

Operating system	:	Windows 7 or above
System type	:	32 or 64 bit operating system.
Language	:	Java
Database	:	XAMPP server for MySQL
IDE	:	NetBeans 8.2

3) Cloud Environment:
Cloud Simulation (CloudSim 3.0) with NetBeans 8.2

Implementation of the proposed system is in progress and in the near future we will present the system. Implementation of proposed system is based on cloud based simulation, using CloudSim. The simulation using CloudSim supports the techniques for implementing cloud computing infrastructures and services. This simulation also supports for the execution of tasks. Cloud based simulation allows cloud users to test their services free of cost in a repeatable and controllable environment. Through cloud simulation, we can improve the design and operation of complex systems. Analysis of performance could be easy through the cloud simulation. CloudSim is open source software and uses Java as a programming language.

6. Conclusion

We have proposed a load balancing algorithm based on clustering and which combines the features of three heuristic algorithms to keep the makespan to the smallest possible amount. By adopting clustering technique, proposed system reduces the time required to search suitable VM and enhances the performance of the cloud system.

7. References

1. Sajjan RS, Yashwantrao BR, Harshal T. Enterprise architecture and services in cloud computing: A survey. International Journal of Computer Sciences and Engineering. 2016; 4(4):28–34.
2. Mell P, Grance T. The NIST definition of cloud computing. National Institute of Standards and Technology. Special Publication 800-145; 2011. p. 1–7.
3. Sajjan RS, Yashwantrao BR. Load balancing and its algorithms in cloud computing: A survey. International Journal of Computer Sciences and Engineering. 2017; 5 (1):95–100.
4. Kaur G, Kaur S. Improved hyper-heuristic scheduling with load-balancing and RASA for Cloud computing systems. International Journal of Grid and Distributed Computing. 2016; 9(1):13–24. Crossref.
5. Deepa K, Prabhu S, Sengottaiyan N. A hyper-heuristic method for scheduling the jobs in cloud environment. Informatics Engineering, an International Journal. 2016; 4(1):23–31.
6. Mart R, Reinelt G. The Linear ordering problem exact and heuristic methods in combinatorial optimization. 2011; 175:17–40. Crossref.
7. Mangla P, Goyal SK. A study of meta-heuristic based techniques of load balancing in cloud computing environment. International Journal in Management and Social Science. 2016; 4(4):194–7.
8. Genetic Algorithm [Internet]. [cited 2018 Mar 05]. Available from: https://en.wikipedia.org/wiki/Genetic_algorithm.
9. Sliwko L, Getov V. A meta-heuristic load balancer for cloud computing systems. IEEE 39th Annual International Computers Software & Applications Conference; 2015. p. 121–6. Crossref.
10. Global optimization [Internet]. [cited 2018 Feb 08]. Available from: https://en.wikipedia.org/wiki/Global_optimization.
11. Masdari M, Salehi F, Jalali M, Bidaki M. A survey of PSO-based scheduling algorithms in cloud computing. Journal of Network and Systems Management. 2017; 25(1):122–58. Crossref.
12. Pandey S, Wu L, Guru SM, Buyya R. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. 24th IEEE International Conference on Advanced Information Networking and Applications; 2010. p. 400–7. Crossref.
13. Makasarwala HA, Hazari P. Using genetic algorithm for load balancing in cloud computing. 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). Ploiesti; 2016. p. 1–6. Crossref.
14. Patel RR, Patel SJ, Patel DS, Desai TT. Improved GA using population reduction for load balancing in cloud computing. International Conference on Advances in Computing, Communications and Informatics (ICACCI). Jaipur; 2016. p. 2372–4. Crossref.
15. Wang B, Li J. Load balancing task scheduling based on Multi-Population Genetic Algorithm in cloud computing. 35th Chinese Control Conference (CCC). Chengdu; 2016. p. 5261–6. Crossref. PMCid:PMC5228194.

16. Acharya J, Mehta M, Saini B. Particle swarm optimization based load balancing in cloud computing. International Conference on Communication and Electronics Systems (ICCES). Coimbatore; 2016. p. 1–4. Crossref.
17. Pan K, Chen J. Load balancing in cloud computing environment based on an improved particle swarm optimization. 6th IEEE International Conference on Software Engineering and Service Science (ICSESS). Beijing; 2015. p. 595–8. Crossref.
18. Zhang Y, Li Y. An improved adaptive workflow scheduling algorithm in cloud environments. Third International Conference on Advanced Cloud and Big Data. Yangzhou; 2015. p. 112–16. Crossref.
19. Su N, Shi A, Chen C, Chen E, Wang Y. Research on virtual machine placement in the cloud based on improved simulated annealing algorithm. World Automation Congress (WAC). Rio Grande; 2016. p. 1–7.
20. Fan Z, Shen H, Wu Y, Li Y. Simulated-annealing load balancing for resource allocation in cloud environments. International Conference on Parallel and Distributed Computing, Applications and Technologies. Taipei; 2013. p. 1–6. Crossref.
21. Kapoor S, Chetna D. Cluster based load balancing in cloud computing. Eighth International Conference on Contemporary Computing (IC3); 2015. p. 76–81.