Fuzzy Based VM Allocation Policy for Load Balancing

CS218 Topics in Cloud Computing

Term Project 2

Piyush Bajaj

Saketh Saxena

Dr. Melody Moh

CONTENTS

Abstract

Cloud computing is a computing model which ensures delivery of computing services such as servers, storage, databases, networking and software over the internet and is commonly referred to as the cloud. Cloud computing brings advantages such as cost, flexibility and availability of service to the users. These advantages have increased the demand for cloud services which brings along a lot of technical issues such as availability of resources and scalability. Cloud Load Balancing is a technique to divide the load among different available resources and equalize it across virtual machines to achieve high availability and increase throughput. In our research project, we have explored two new algorithms - Clustering based Load Balancing using Bayes clustering (LB-BC) and Load Balancing Algorithm for virtual Cluster using Fuzzy Clustering (LB-VC-FC). The LB-VC-FC algorithm in comparison to the other algorithms returns the most efficient results for balancing comprehensive loads across multiple dimensions (CPU, memory, and bandwidth) of the virtual clusters. Using the techniques described in the LB-VC-FC algorithm we have extended the VM allocation policy for the datacentre which is a worst fit policy since it considers only the free processing elements(PE) of the hosts and allocates the VM to the host with the most free Pes and implemented two new algorithms to achieve load balancing. The first algorithm is an advanced VM allocation policy which considers the CPU utilization, memory utilization and the bandwidth utilization of the hosts and allocates the VM to the host with the first host which has greater available resources than the ones requested and in case all hosts are busy, it finds the host with the maximum available resources and allocates the VM to the host. The second allocation policy uses fuzzy logic to compute the fuzzy sets with host membership values for each VM. The membership function is then maximized to get the most optimal host to allocate the VM to. Based on our experiments the advanced VM allocation policy optimizes resource utilization in terms of number of hosts used and reduces the multidimensional load and the fuzzy logic based policy further optimizes the comprehensive load but returns similar resource utilization results as the simple allocation policy.

1. Introduction

Cloud computing is a computing model which enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage devices, applications and services which can be rapidly provisioned and released with minimal management effort and service provider interaction [2]. Cloud computing brings advantages such as virtualization of resources, cost, scalability, flexibility and high availability of service to users [6]. These advantages have driven numerous organizations to port their applications and services, with a huge user base, to the cloud environment which inadvertently has resulted in a high demand for cloud services. Since the resources in an open cloud computing environment are typically shared, spikes in demand lead to technical issues relating to maintaining availability of resources, providing scalability and ultimately resulting in deadlocks.

One way of avoiding this situation is via load balancing, Cloud Load Balancing is a technique to divide the load among different available resources and equalize it across virtual machines to achieve efficiency and increases throughput [6]. It helps in avoiding deadlocks by optimal utilization of resources and increases the system performance. The main challenge in load balancing is to distribute dynamic workload across multiple nodes to ensure no single node is overwhelmed [6]. The goal of load balancing is to reduce resource consumption which will further lead to reducing energy consumption and ensure scalability, avoid bottlenecks, and over-provisioning. Thus, having an efficient load balancing technique in place is extremely important, to meet the growing demand of cloud computing resources.

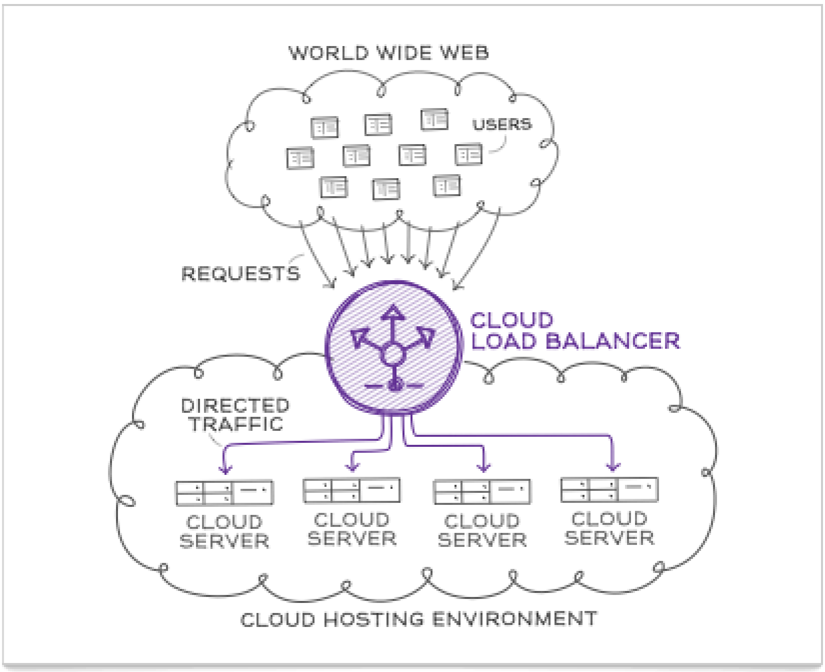


Figure 1: Cloud Load Balancing Schematic Representation [5]

A lot of work has already been done for load balancing in the cloud and, typical load balancing techniques include task scheduling algorithms, load balancing policies, and client-side load balancing [5]. As part of our research project we have explored some of existing load balancing techniques such as workload and client aware policy (WCAP)[5], opportunistic load balancing[1], Load Balancing Min-Min[10], Join-Idle-Queue[7] and Honeybee Foraging Behaviour [5]. We have also explored some new approaches to load balancing in the cloud which employ the use of clustering techniques and heuristics to optimize load balancing such as Load Balancing using Bayes Clustering [10] and Load Balancing using Fuzzy Clustering in Virtual Clusters (LB-VC-FC) [5]. Using the techniques described in LB-VC-FC pertaining to fuzzy logic and formulation of the membership function we have extended the native simple virtual machine allocation policy and implemented two new VM allocation policies to achieve load balancing in the datacentre. The VM allocation policy considers the host with the most number of free processing elements in the datacentre and allocates the VM to the host, this is considered to be a worst fit policy since it does not consider the memory utilization, CPU utilization and memory utilization loads of the host. We have endeavoured to mitigate these shortcomings in the first algorithm which is an advanced VM allocation algorithm which considers the comprehensive loads pertaining to CPU, memory and bandwidth of the hosts and selects the first host which has more resources available than the requested ones, in case none of the hosts have the required resources the policy allocates the VM to the one with the maximum available resources at that time. The second algorithm we have implemented is a fuzzy logic based VM allocation policy. In this we compute a set of membership values for each VM for all the hosts and pick the host with the maximum membership value. The feature set for the membership function is comprised of the available memory resources, CPU resources and fuzzy resources along with an added weight of number of free processing elements for the host. We elucidate these two algorithms and the comparative results for these algorithms in the following sections. Section 2 explains in detail the project description including the existing algorithm and the proposed enhancements, section 3 details the experimental setup, section 4 deals with the project design, section 5 demonstrates the system architecture, section 6 shows the performance and evaluation results and sections 7 concludes this report and gives the future direction for our enhancements.

2. Project Description

As stated earlier, we have extended the simple VM allocation policy of the datacentre. We would like to present an overview of the VM allocation policy and then describe our proposed enhancements.

Simple VM Allocation Policy for Cloud Load Balancing

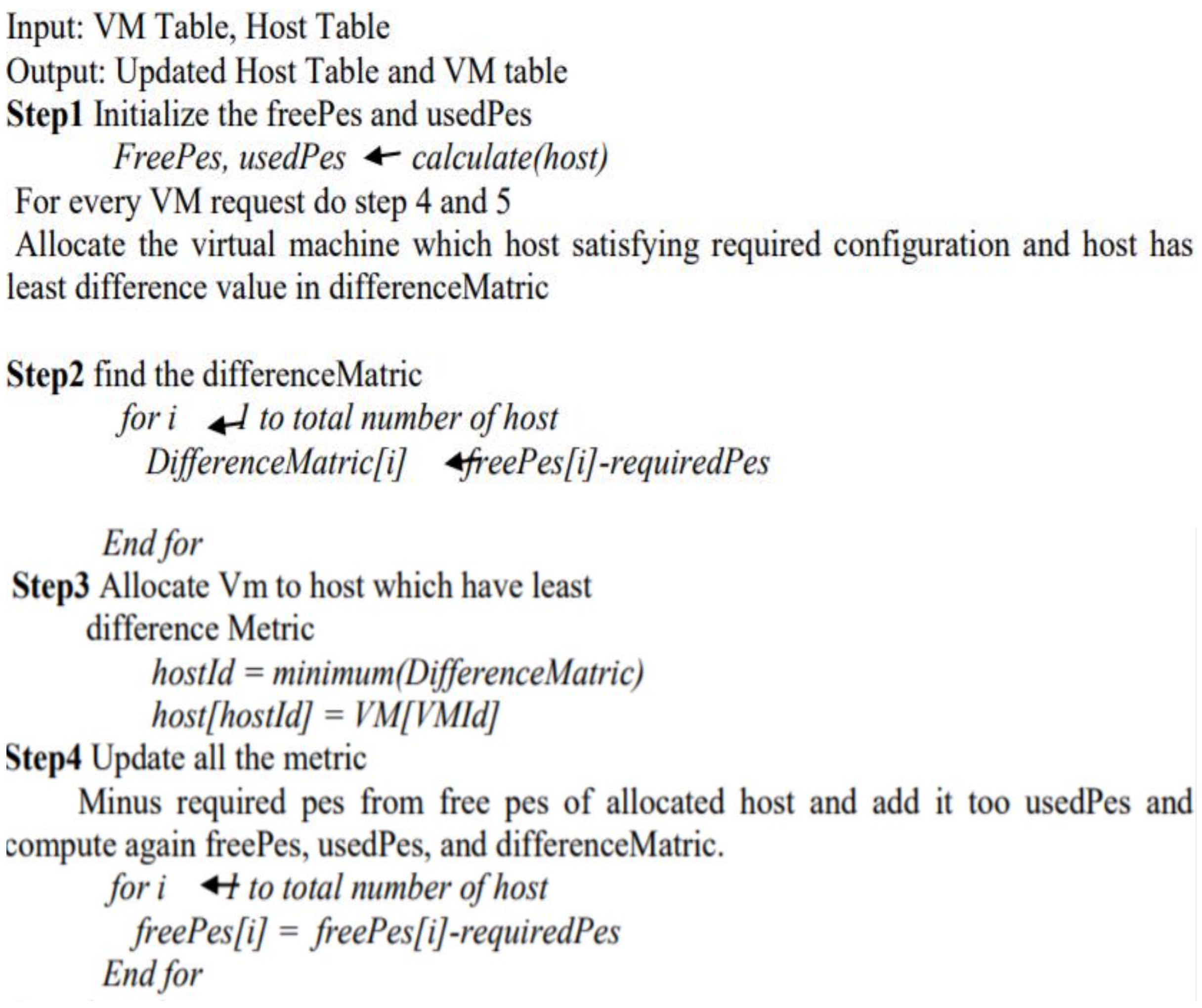
Cloud Computing provides way to allocate several virtual machines to run on physical machine. This allows more number of application and services run at same time to achieve more productivity and high profit rate for cloud service providers. To get the benefits, proper allocation method must be used to reduce migration of machines, avoid overload on single physical machine, less energy consumption etc.

In a cloud computing environment lots of workload needs to be maintained and though the virtualization is dependent or limited by actual hard ware resources VM allocation policy plays a vital role in cloud computing life cycle. It is crucial for the cloud data center to balance the loads of different computing nodes and to maintain high utilization efficiency of system resources. The objective of the Simple VM Allocation Policy is to utilize the resource by making all the processing elements (PE) for a host, occupied. It chooses the host with less PEs that are in use as the host for a VM. It is therefore not a best fit policy, allocating VMs into the host with most available Pes, not considering other resource loads such as memory and bandwidth.

Most of the algorithms however, consider only the loads on the system processing unit to separate the work load, loads of other system resources are ignored which causes poor resource utilization, may lead to overloading and greater task completion time.

Algorithm for simple VM Allocation Policy

The algorithm for the simple VM allocation policy is as follows:



```
Input: VM Table, Host Table
Output: Updated Host Table and VM table
Step1 Initialize the freePes and usedPes
        FreePes, usedPes ← calculate(host)
 For every VM request do step 4 and 5
 Allocate the virtual machine which host satisfying required configuration and host has
least difference value in differenceMatric

Step2 find the differenceMatric
        for i ←1 to total number of host
          DifferenceMatric[i] ←freePes[i]-requiredPes

        End for
Step3 Allocate Vm to host which have least
      difference Metric
          hostId = minimum(DifferenceMatric)
          host[hostId] = VM[VMId]
Step4 Update all the metric
      Minus required pes from free pes of allocated host and add it too usedPes and
compute again freePes, usedPes, and differenceMatric.
        for i ←1 to total number of host
          freePes[i] = freePes[i]-requiredPes
        End for
```

UsedPes stores the number of used processing elements for the particular host.

DifferenceMatric is one dimensional metric that store different between required pes and free pes for every host in data center.

FreePes is the list that stores the free space or more specifically free pes (Processing Elements) for the particular host.

VM table is table for storing host with key value. The map between each VM and its allocated host. The map key is a VM UID and the value is the allocated host for that VM.

Proposed Enhancements

I. Advanced VM Allocation Policy considering CPU, memory and bandwidth utilization

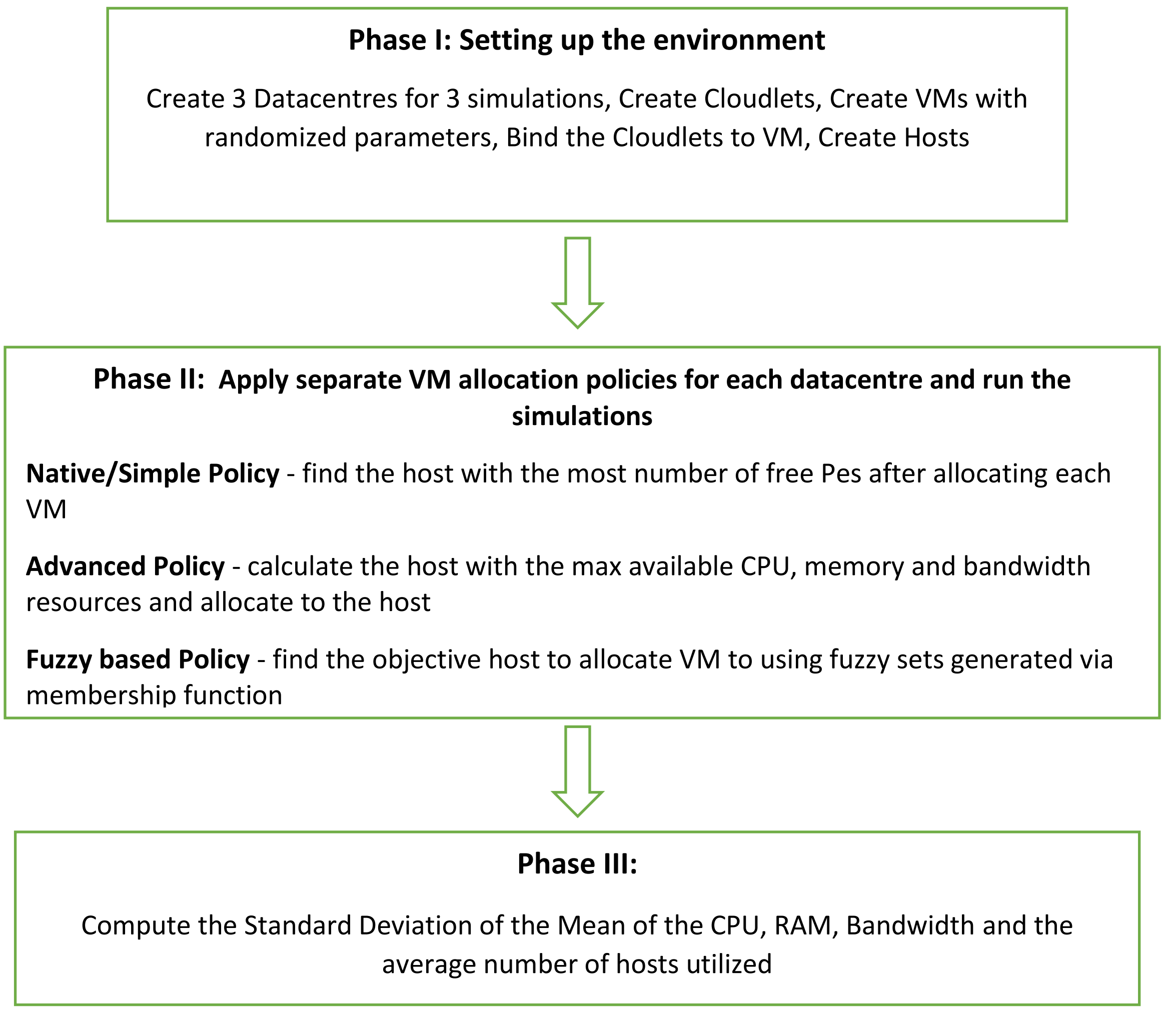
We have implemented an advanced VM allocation policy as an alternative to the simple VM allocation policy. The advanced policy considers the availability of all the current host resources i.e. the current available CPU in mips, the current available bandwidth and the current available memory of the hosts and finds the host which has all three of these resources in surplus or equal to the resources required by a particular VM. In situations where none of the hosts has enough resources for allocating an incoming VM then a suboptimal solution is selected where the host with the maximum available resources is allocated to the VM. This is a partial of Best Fit Host policy where initially the best host is selected to maximize throughput and obtain high resource utilization.

II. VM Allocation Policy using Fuzzy Sets

The advanced VM allocation policy is further enhanced to determine the most suitable objective host for a virtual machine. In this policy we combine the fuzzy logic proposed in LB-VC-FC algorithm and the simple VM allocation policy to obtain the most optimal node to allocate VM to. The concept of fuzzy sets is used by generating membership values for a set of available hosts in the datacentre against each VM. Therefore, each VM has a fuzzy set of host membership values mapped to it. The membership value is calculated by using a Euclidean distance measure to obtain a membership value in the range 0 to 1. We consider host having a value of 0 to be the least optimal host and 1 as the most optimal host or a particular VM. The feature set considered for the Euclidean distance is a set of the available CPU in mips, available RAM and available bandwidth of a particular host. The Euclidean distance is computed between each host and the VM to be allocated and thus the fuzzy set is obtained. Next the membership values are added an additional weight by using the number of free processing elements of the hosts and maximizing this value to obtain a further host with the most available resource this allows for tie breaking or reallocation in case of failure of allocation. This case occurs when the requested resources are too high and the available resources are too low.

3. Project Design

The workflow of our project is illustrated in Figure 2 below.



Phase I: Setting up the environment

Create 3 Datacentres for 3 simulations, Create Cloudlets, Create VMs with randomized parameters, Bind the Cloudlets to VM, Create Hosts

Phase II: Apply separate VM allocation policies for each datacentre and run the simulations

Native/Simple Policy - find the host with the most number of free Pes after allocating each VM

Advanced Policy - calculate the host with the max available CPU, memory and bandwidth resources and allocate to the host

Fuzzy based Policy - find the objective host to allocate VM to using fuzzy sets generated via membership function

Phase III:

Compute the Standard Deviation of the Mean of the CPU, RAM, Bandwidth and the average number of hosts utilized

Figure 2: Project Workflow

The first phase, involves setting up the simulation environment by creating 3 datacentres for running 3 different simulations using the same VM parameters which are randomly generated for each run of the program. The second phase, deals with assigning the respective allocation policy for each data centre and computing the actual allocations while the simulation is started. Once the simulation ends, we calculate the standard deviation of the mean of all the hosts' load/utilization values over the period of the simulation and then compute the standard deviation for all the values and the execution ends.

4. System architecture

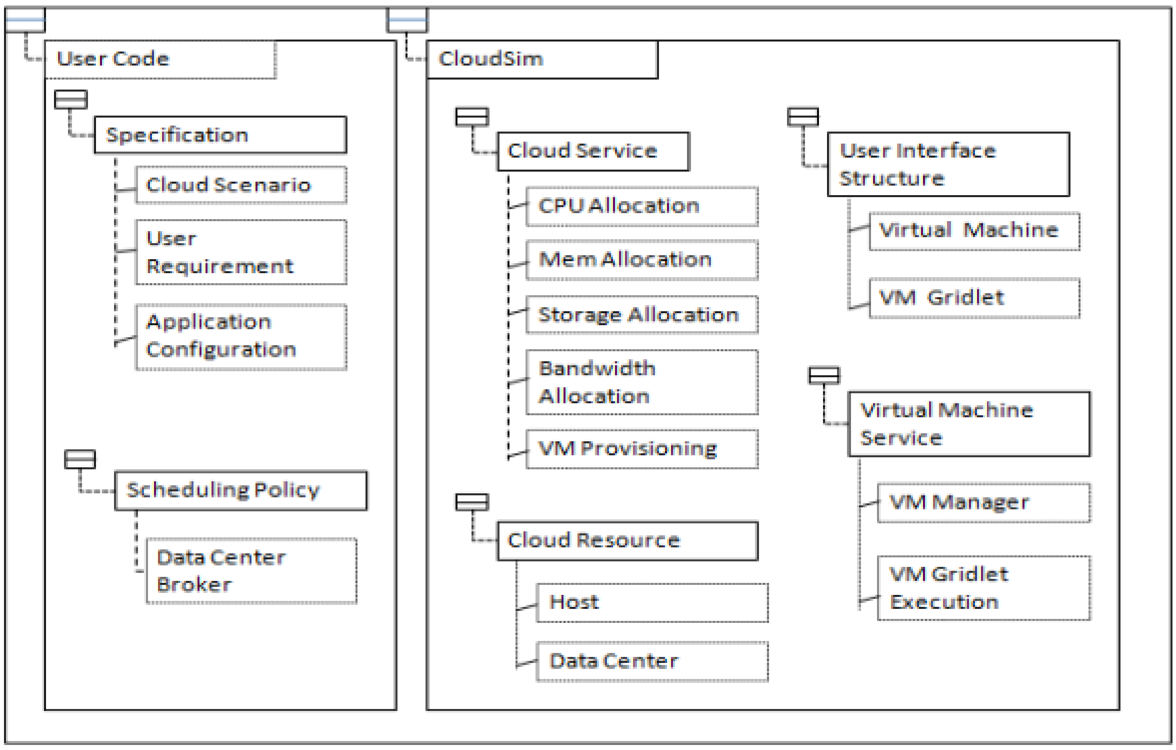


Figure 3. System Architecture in CloudSim [3]

The architecture of the system is modelled in CloudSim where the simulation environment is set up as described in the project design section. The CloudSim architecture is based on GridBus middleware environment to simulate a cloud computing environment to perform research and experiments as shown in figure 3. We have added two new VM allocation policies which we implemented as external classes in the CloudSim environment which we are importing in the datacentres. Three datacentres are created one for simulating each VM allocation policy. We have also implemented a special class to store the initial randomly generated VM parameters to recreate for each datacentre. Finally, the simulation is run on CloudSim to obtain standard deviation of the mean resource utilization values.

5. Experimental Setup

The experimental environment used in our project is in CloudSim 3.0.3 [9] and the configuration is as follows:

- 3 datacentres running each running one of the three allocation policies respectively
- 25 physical nodes in each datacentre
- 500 virtual machines in each data centre (The VM parameters are randomized to simulate variable VM sizes and requested resources and the same set of VMs are used to study simulations of the three policies)
- 1000 cloudlets in each datacentre
- The server is configured with x86 architecture, CentOS operating system.
- Virtualization hypervisor Xen

Overview of CloudSim

CloudSim is a popular cloud computing environment simulation tool, used to allow modelling, simulation and testing of cloud computing infrastructures and application services. The modelling and simulation that is done through CloudSim is faultless. Suppose an application package has been developed and a developer or researcher wants to check its performance in a controlled environment, then they can use CloudSim as a widespread framework [3]. CloudSim framework is built on top of GridSim framework. The architecture of CloudSim is as shown in Figure 4 below.

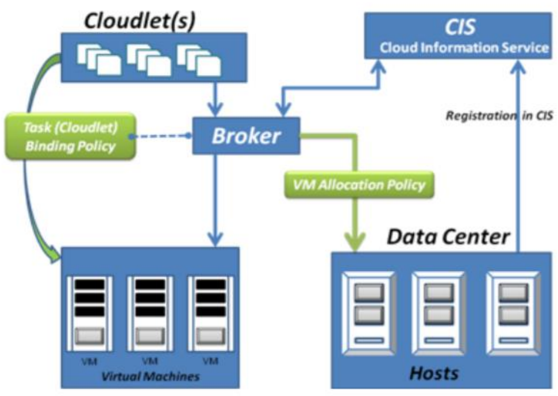


Figure 4: Architecture of CloudSim[8]

6. Performance Evaluation & Results

For evaluating the effectiveness of the above proposed approach, we conducted experiment and by simulating the algorithms in CloudSim.

Simple VM allocation policy allocates VM on to the least utilized hosts, while Advanced level VM allocation policy allocates VM based on the all the available resources of the host as requested by the VM. On the other hand, Fuzzy based VM allocation policy improvises the previous algorithm and considers Euclidean distance of the requested resources of the VMs and available resources of the hosts to find the most objective host to allocate a particular VM.

Calculation of Standard Deviation of mean of resource usage statistics for multiple runs:

We are extracting the mean CPU, bandwidth and memory utilization values after allocation of each VM and the computing the standard deviation values of all the means for each algorithm. By comparing the standard deviation values of the three algorithms we can study the performance of the algorithms where a higher SD values means that the resource utilization was quite varied and a lower one means that the utilization for the resource was within a small range.

For the purpose of this project since the VM parameters are generated randomly in order to validate our results we have computed the standard deviation values for multiple runs of the simulation. The next part of the report will be explaining the results obtained

Experimental Results:

To standardize and validate our results we simulated the algorithms 30 times and the following diagrams [Figures 5-8] show the comparison of the Standard Deviation of the mean utilization of CPU, memory and network bandwidth between Simple/Native, Advanced and Fuzzy Based VM Allocation Policy.

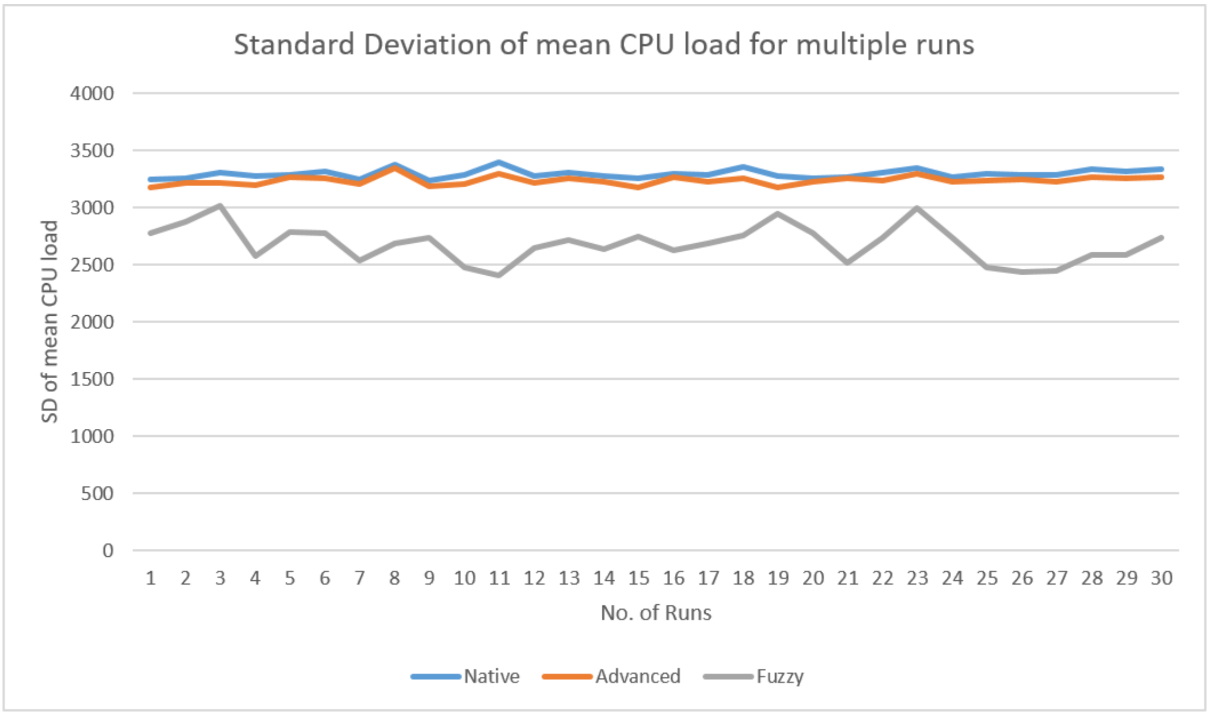


Figure 5. Comparison of Standard Deviation of mean CPU utilization rate for Simple, Advanced and Fuzzy based VM Allocation Policy

Figure 5 above, shows the comparison of the S.D. values for CPU utilization for multiple runs. The results indicate that the Fuzzy based algorithm has the lowest standard deviation value and the range is generally restricted. The advanced allocation policy has slightly lower values showing suboptimal CPU utilization and the simple VM allocation policy has the highest standard deviation value.

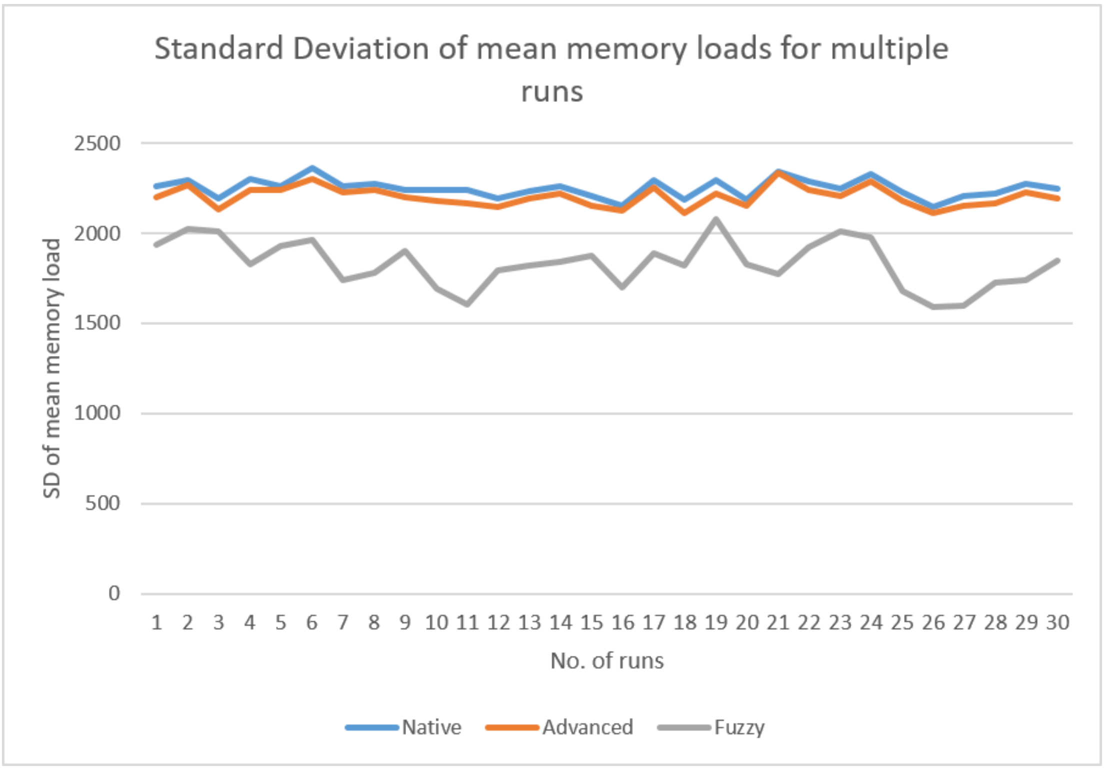


Figure 6. Comparison of Standard Deviation of mean memory utilization rate for Simple, Advanced and Fuzzy based VM Allocation Policy

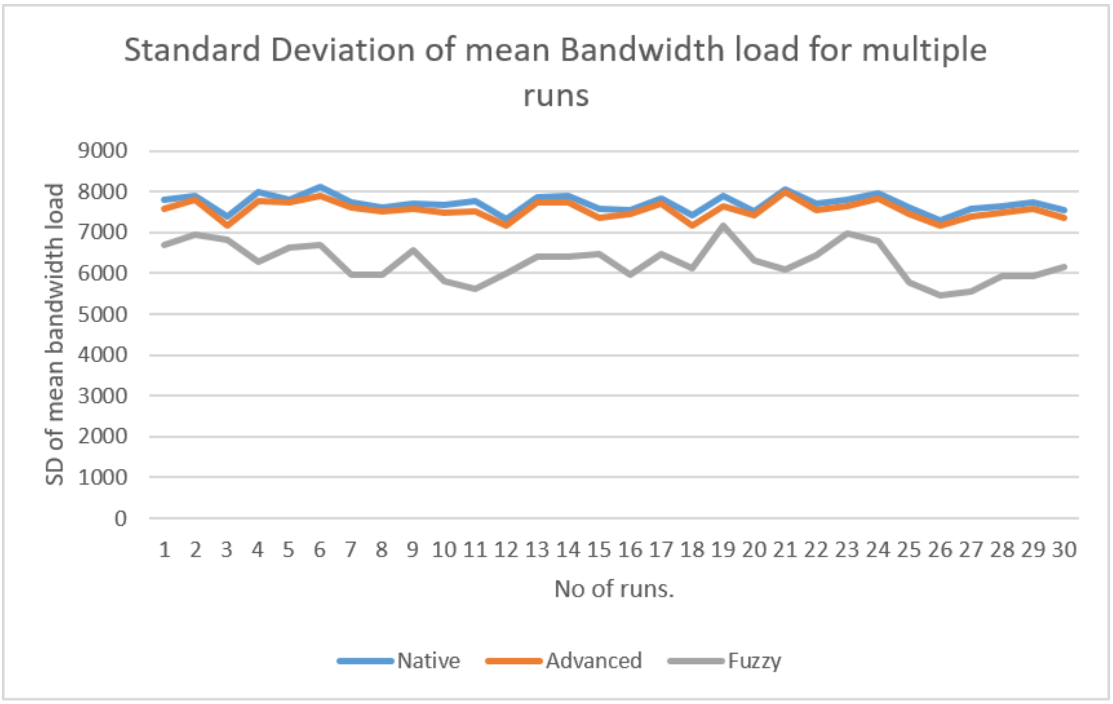


Figure 7. Comparison of Standard Deviation of mean Bandwidth utilization rate for Simple, Advanced and Fuzzy based VM Allocation Policy

The mean of the bandwidth for each host in the data center was calculated. The Standard Deviation (SD) of those Mean of bandwidth load were calculated. This simulation was executed 30 times to get the desired change of variation in output values. It can be observed from Figure 6. that the SD of the means of bandwidth load is lowest for the Fuzzy based VM allocation policy than the other 2 allocation policy.

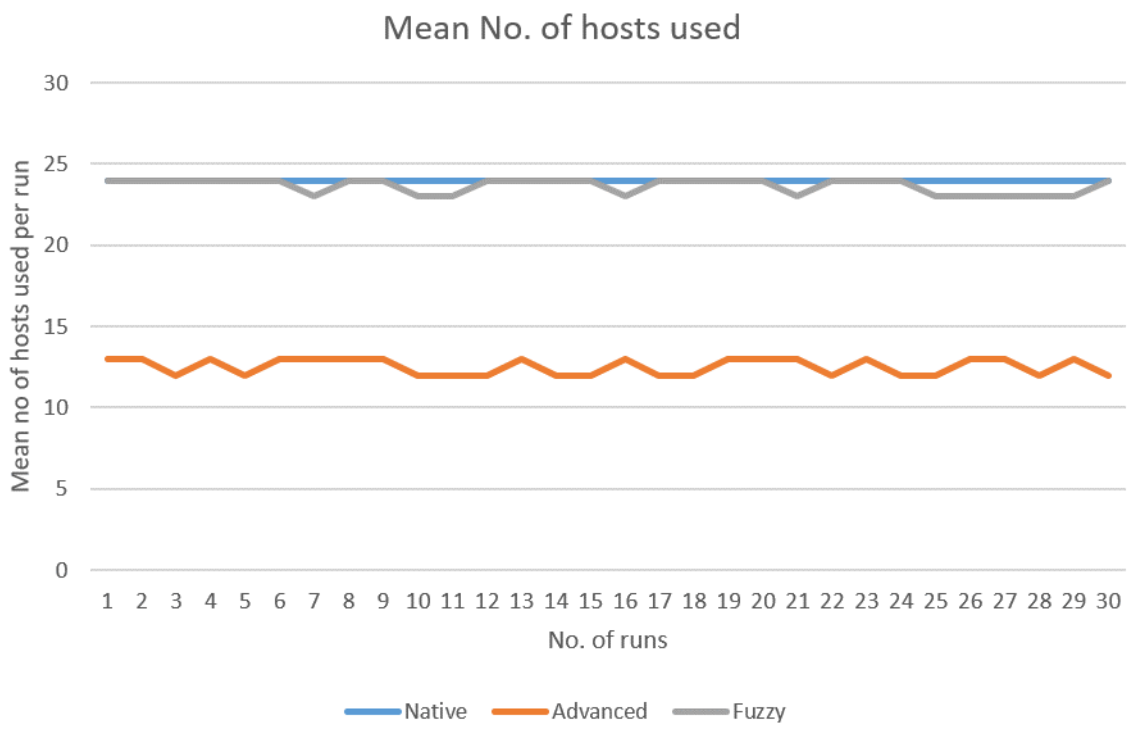


Figure 8. Comparison of number of hosts utilized for Simple, Advanced and Fuzzy based VM Allocation Policy

It can be observed from Figure 8 for all the VM allocation, the mean number of hosts utilized returned by the Advanced VM allocation policy is far less than the simple and Fuzzy based VM allocation policy.

The following charts [Figures 9-11] show the comparison of the CPU, memory and network bandwidth loads in a single run among Simple, Advanced and Fuzzy Based VM Allocation Policy over simulation time.

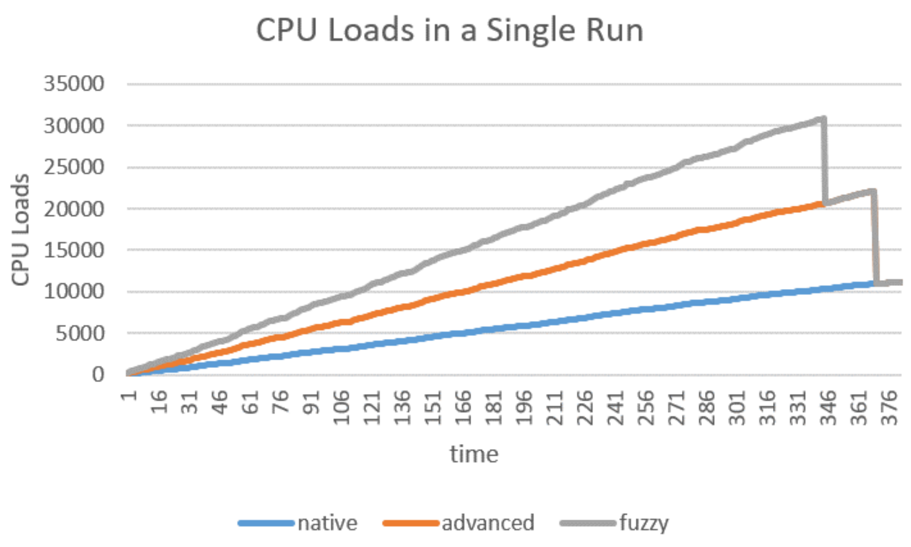


Figure 9. Comparison of CPU loads in a single run over time for Simple, Advanced and Fuzzy based VM Allocation Policy

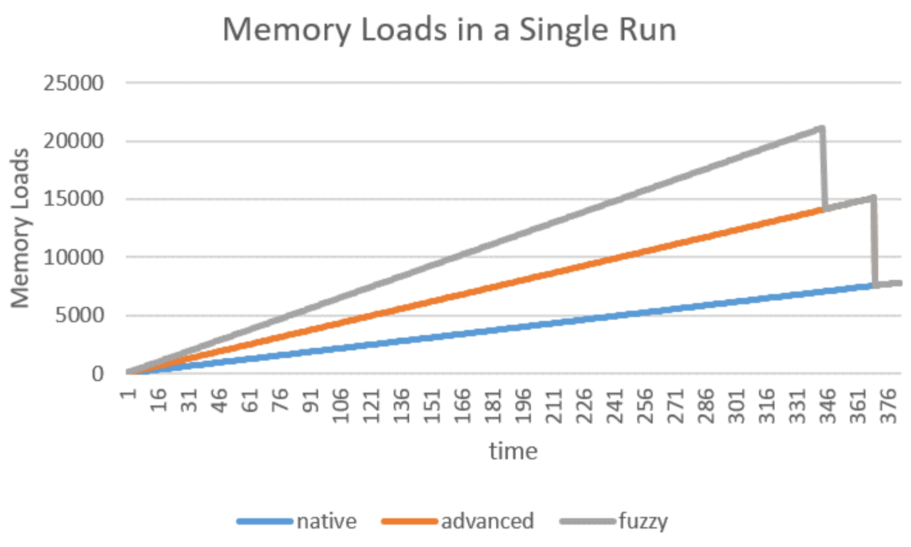


Figure 10. Comparison of Memory loads in a single run over time for Simple, Advanced and Fuzzy based VM Allocation Policy

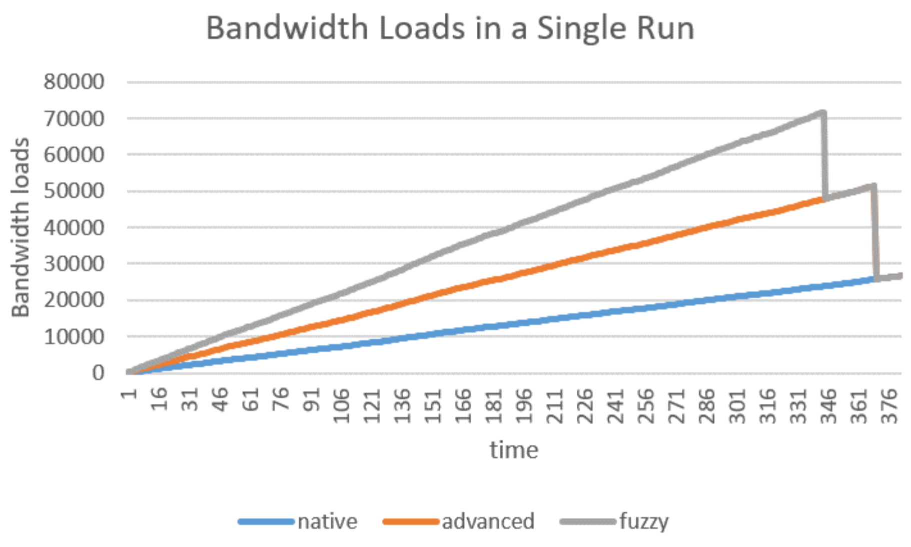


Figure 11. Comparison of bandwidth loads in a single run over time for Simple, Advanced and Fuzzy based VM Allocation Policy

It is observed through analysis of the above graph and the data generated that the Fuzzy based VM allocation policy executes at a higher resource utilization rate by spreading the VMs across all hosts by optimally performing VM allocation and thus, reduces the completion time of the simulation as compared to the advanced and the native/simple VM allocation policy essentially balancing the load most effectively to give better output.

7. Conclusion and Future Work

In summary, the simple VM allocation algorithm allocates the virtual machines to the host of the data center which have amount of free pes close to the amount of pes required by virtual machine. The advanced VM allocation policy allocates the VMs to the host considering all the resources for the hosts and the VMs. Finally, this algorithm is improved by computing the membership values based on the Euclidean distance of the CPU, memory and bandwidth resources of the VMs and the hosts as the feature sets.

It can be concluded from results of implementation that proposed advanced VM allocation policy works efficiently when it comes to resource utilization. It can also be observed that the total number of hosts utilized are less for this policy while reducing the overall resource load as compared to the native simple VM allocation policy. It is also shown through the experimental analysis that the fuzzy sets based VM allocation policy does not affect the optimize resource utilization but greatly optimizes the over load of the datacentre. For further enhancements and experimentation the proposed VM allocation policies can be integrated to handle dynamic workload of the hosts and VMs.

8. References

[1] Al Nuaimi, Klaithem, et al. "A survey of load balancing in cloud computing: Challenges and algorithms." Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on. IEEE, 2012.

[2] Bhavya, V. V., K. P. Rejina, and A. S. Mahesh. "An Intensification of Honey Bee Foraging Load Balancing Algorithm in Cloud Computing."

[3] Calheiros, Rodrigo N., et al. "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services." arXiv preprint arXiv:0903.2525 (2009).

[4] http://www.globaldots.com/knowledge-base/cloud-loadbalancing/cloud_balancer/

[5] Huang, Weihua, et al. "Load balancing algorithm for virtual cluster using fuzzy clustering."Computer and Communications (ICCC), 2016 2nd IEEE International Conference on. IEEE, 2016.

[6] Kokilavani, T., and DI George Amalarethinam. "Load balanced min-min algorithm for static meta-task scheduling in grid computing." International Journal of Computer Applications 20.2 (2011): 43-49.

[7] Lu, Yi, et al. "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services." Performance Evaluation 68.11 (2011): 1056-1071.

[8] Parikh, Kushang, et al. "Virtual machine allocation policy in cloud computing using cloudsim in java." International Journal of Grid and Distributed Computing 8.1 (2015): 145-158.

[9] Wickremasinghe, Bhathiya, Rodrigo N. Calheiros, and Rajkumar Buyya. "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications." Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. IEEE, 2010.

[10] Zhao, Jia, et al. "A heuristic clustering-based task deployment approach for load balancing using bayes theorem in cloud environment." IEEE Transactions on Parallel and Distributed Systems 27.2 (2016): 305-316.