Proceedings of 10th International Conference on Science and Engineering 2019,
7-8 December 2019, Yangon, Myanmar

ICSE2019-ICT-1

Efficient Virtual Machine Allocation Model for Cloud Data Center

Yu Mon Zaw[#1], Khine Moe Nwe[*2]

[# *] Faculty of Computer Science, University of Computer Studies, Yangon
Yangon, Myanmar

[1]yumonzaw@ucsy.edu.mm

[2]khinemoenwe@ucsy.edu.mm

Abstract -Cloud computing is the on-demand availability of computer system resources and services over the internet and cloud data centers play important role in fulfilling on-demand facilities such as servers, communication media and data storage facilities. In recent years, the dynamic allocation of virtual machines in cloud data centers has become one of the promising challenges. The result of under-utilization and wastage could cause the poor service and decreasing performance in data centers. Therefore, resource allocation policies play important role in allocating and managing the resources in cloud computing environment. This paper proposes a model of efficient Virtual Machine Allocation Policy with considering CPU and Memory (RAM) as attributes in mapping VMs to the most suitable Physical Machine in Cloud Data center. The proposed model uses Jump Search algorithm in order to receive fast speed allocation and the result is verified by simulating this model using CloudSim.

***Keywords* - VM allocation policy, Jump Search, Cloud Computing, Cloud Data center, dynamic allocation**

I. Introduction

Traditionally, businesses had a choice of building their own data center, using a hosting vendor or a managed service partner. But with the emergence of Cloud technology, the rise of Infrastructure as a Service (IaaS) from cloud providers makes a lot of changes in the traditional trend of data center infrastructure. The emergence of cloud computing technology has been altered the IT industry to become part of the modernized world and a large cloud data centers are serving the millions of demands over the internet for computing, storage and networking and application resources based on the pay-for-use model.

A data center usually contains a large number of Physical machines (PM) arranged on racks and packed densely in order to increase space utilization. In cloud computing, one of the key concepts for data center management is virtualization. The most important advantage of virtualization is the capacity to run many instances of operating system in a single PM thus making maximum use of the hardware capacities of the PM which helps immensely in saving money for energy and hardware costs. These individual instances of operating system are called Virtual Machines (VM). Thus, in cloud computing, users access the computing resources of a data center (DC) with the help of VM.

The fundamental infrastructure services (IaaS) is the provisioning of virtual machines (VMs). Virtualization refers to running multiple virtual computers, or virtual machines, inside a single physical machine (PM). In recent years, the problem of VM allocation in the cloud has been studied for efficient utilization of physical machine and quality of service (QoS) purpose. These studies result that the utilization of computing resources can be efficient and also reduce the number active PMs. Bin Packing Problem is a problem where objects with a given area must be packed into a finite number of bins with a given area such that the minimum amounts of bins are used. Also, it can enhance the allocation of VM [2]. In the proposed VM allocation model, the concept of a bin-packing problem is applied and VM allocation can be optimized with the Best-fit (BF) scheme. Jump Search algorithm is applied in finding the most fitted PM for a given VM according to two resource parameters CPU and Memory (RAM). The proposed VM allocation algorithm will maximize the effective utilization of physical servers and faster response mapping the VM to the servers in the cloud data center.

II. Related Works

Cloud resource management has become a key factor for the cloud data centers. The right placement of virtual machine (VM) on the right physical machines (PMs) is important issue in cloud computing environment. The improper placement of resources may result the overall resource wastefulness of an IaaS cloud.

In [1], the proposed Dynamic Resource Management Algorithm (DRMA) can save from resource underutilization at a particular period of time and the bin-packing algorithm is used to formalize the placement problem and developed by Best fit method. DRMA configured the CPU and memory as resource demand in cloud data centers.

In [3], Madnesh K. Gupta et al proposed a new VM placement algorithm called Resource-aware virtual machine placement algorithm (RVMP) for IaaS cloud. They minimize the power consumption by reducing number of active PMs with the use of a new technique called resource usage factor. This factor is used to place a VM on a suitable PM. By using the proposed model, a limited migration of VMs has occurred in order to minimize the unbalanced resources utilization.

In [4], Zhuo Tang et al. proposed a scheduling algorithm called virtual machine dynamic forecast scheduling (VM-DFS) to allocate virtual machines in a cloud computing environment. This algorithm is based on analysis of historical memory consumption and the most suitable physical machine can be selected to place a virtual machine according to future consumption forecast. The problem of VM allocation is carried out as a bin-packing problem and used by the first-fit decreasing scheme (FFD). According to its experimental results, it can minimize the number of physical machines used.

In [5], X.F. Liu et al. proposed an ACS-based approach, OEMACS, to assign m VMs to n PMs. The VMs were grouped depending on historical experience of packing the VMs together before using artificial ants to search for fittest place for hosting. The infeasible solution of the update was revised every search iteration to reduce the convergence time. The solution space was reduced while the iteration number grows. The solution is revised to turn infeasible solution to feasible one by ordering exchange process and migrate VMs on overloaded PMs which reduce the convergence time.

In [6], Bramantyo Adrian et al. conducted K-means Algorithm for VM Allocation. In this research, number of K cluster determined dynamically according to number of datacenter or according to number of host. Then after the clustering result obtained, broker perform the virtual machine allocation on host and datacenter according to the clustering result so in each host and datacenter has an identical virtual machine.

In [8], Khine Moe Nwe et al. proposed an efficient resource management for VM allocation in cloud data centers. They proposed a model which apply Binary Search algorithm in VM allocation in physical machines. This proposed model would be faster than existing mapping model because of the deployment of the binary search method.

III. Overview of The Proposed Model

The proposed model is developed by using CloudSim[7] . CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing environments. The model proposed by this paper is to dynamically allocate the Virtual Machines (VMs) into suitable active Physical Machines (PMs) or Hosts.

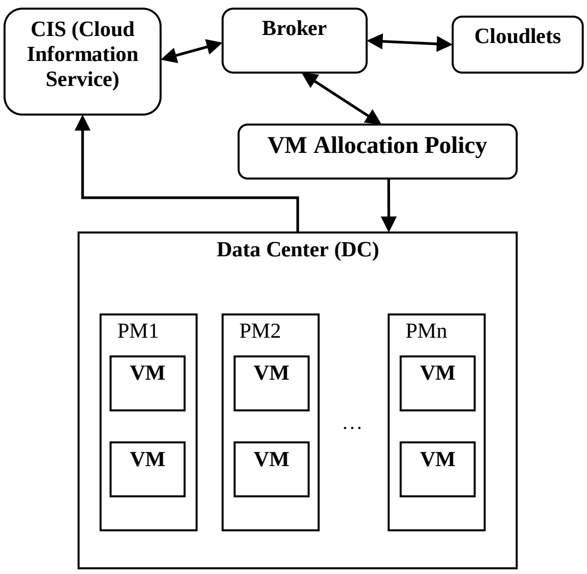

Fig 1. Overview of the proposed model

According to Figure 1, Data center (DC) is the resource provider—includes one or more Physical Machines (PMs). PM is the physical machine that allocates one or more VMs. Virtual machine (VM) is a machine on which the cloudlet will be executed. Cloud Information Service (CIS) has responsible for registering all the resources of data centers. Cloudlets are all tasks and applications that are executed on VMs. When a broker has the DC characteristics, it will submit VMs to the specific PM in the specific DC (in the proposed model, there will be only one DC) and then allocate the cloudlets (tasks) to specific VMs. Before broker submits VMs to PMs, it follows VM Allocation Policy where the Jump Search algorithm is deployed.

IV. Applying Jump Search Algorithm in VM Allocation Policy

If there are a huge amount of physical machines (PMs) in the data center, making decision of which PM to be choose to allocate correct VMs is more important. Making the right choice of PM can reduce resource wastefulness and can handle load balancing very well. The use of Jump Search can solve that situation.

The Jump Search is similar to Binary Search but instead of jumping both forward and backward, it will only jump forward. To apply Jump Search, the list of PMs and VMs must be sorted. In the proposed system, sorting PM list (PList) and VM list (VList) is in descending order.

The processing flow of Jump Search is as follows. In Jump Search, the interval square root of list-length is jumped ahead until reaching an element smaller than current element or end of the list. On every jump, the previous step is recorded. If an element smaller than the element we are searching for is encountered, we stop jumping. Then we run a Linear Search between the previous step and the current step. This makes the search space a lot smaller for Linear Search, and thus it becomes a viable option.

In this model, before allocating VMs by broker, the system collects and store the user requested incoming VMs in the virtual machine list (VList) at a regular time interval and waits for allocation in the list. CPU and memory (RAM) are considered as resource demand of the VM. Initially, all physical machine are assumed as actives i.e., Pactive. VList and PList are sorted in decreasing order.

The broker gets the first VM from the VM list (VList) and allocates on the most suitable PM by deploying the proposed efficient VM allocation algorithm. The detailed steps are presented in Figure. 2. The proposed VM allocation algorithm will be significantly faster than the ordinary allocation while mapping the VM to best suitable PM by deploying the Jump Search method.

Input : PList, VList,
Output : tHost, Pactive, VMSetp

where PList is PM list, VList is VM list
tHost is the target host to allocate specific VM
Pactive is currently active PM list
VMSetq is current VM list of current PMq
VMSetp is current VM list in for each PM
m is number of Virtual Machine
n is number of Physical Machine
c is number of active PM
s is number of non-active PM

BEGIN
Sort PList by CPU and Memory in descending order
Sort VList by CPU and Memory in descending order
PLength ← Size of PList

for each p ∈ PList
VMSetp ←Null;//initially no running VM set for each PM

```
endfor

for each vi ∈ VList, i ∈{1,2,..,m}
jumpStep ← Square root of PLength
 previousStep ← 0
jumpindex ←0
index←0

while(PList index (mem) > vi (mem) && PList index (CPU) > vi (CPU) )
then
        if (jumpindex >= PLength) then
             break;
        previousStep ← jumpindex
        jumpindex+= jumpStep
        index← minimum (jumpindex, PLength) -1
        end if
end while

while( previousStep != minimum( jumpindex, PLength)
) then
      if (PList previousStep (mem) <= vi (mem) || PList previousStep (CPU) <
      =  vi (CPU)) then
            break;
     else
        previousStep ++
    end if
end while
q← previousStep
tHost← PList q // target Host is now available

VMSetq ←VMSetq ∪ {vi}
tHostCPU ←tHostCPU – viCPU
tHost mem←tHost mem - vimem
if tHost is not belongs to Pactive then
       Pactive←    Pactive ∪ {tHost};
       c++
endif
endfor // end loop for vi
s=n-c;
END
```

Fig. 2 The proposed VM Allocation Algorithm

V. EXPERIMENT

A. Complexity Analysis

The time complexity of the proposed model is *O(sqrt(N))*where N is the number of PMs in data center since square root of PM Lists steps in each iteration is jumped. It is much faster than the ordinary sequential search one in the previous algorithms, but may be slower than a binary search. The advantage over the binary search is that a jump search only needs to jump backwards once, while a binary search can jump backwards up to log *n* times. So, this can be important if a jumping backwards takes significantly more time than jumping forward.

The proposed model is significantly faster than the previous models for mapping to a huge number of PMs. The more PMs contain in the model, the faster and better result in mapping will come out.

B. Running Time Analysis

The experiment is carried out by using CloudSim 3.0 with machine containing window 7 OS, 4.0 GB RAM and with core i5 processor and Eclipse IDE Oxygen version used. To improve the accuracy of experimental results, each group of experiment is run multiple times, and then calculate the average results.

For VMs, the simulations are conducted for four different combinations of RAM and CPU as follows:

1. 128 MB RAM, 8 CPU
2. 512 MB RAM, 4 CPU
3. 512 MB RAM, 4 CPU
4. 1024 MB RAM, 2 CPU

For PMs, the simulations are conducted for four different combinations of RAM and CPU as follows:

1. 256 GB RAM, 16 CPU
2. 512 MB RAM, 8 CPU
3. 128 GB RAM, 4 CPU
4. 1024 GB RAM, 4 CPU

First of all, the number of PMs available is fixed at 20 and the number of VMs is used in increasing order from 10 to 100. The result is shown in Table 1.

TABLE 1: FIXED PMs WITH DIFFERENT VMs

No: of PMs	**No: of VMs**	**Average Time Taken (mili sec)**
20	10	159.2
20	20	163.4
20	50	166.2
20	70	170.3
20	100	180.3

Next, a set of experiments keeping the number of VMs available fixed at 20 and increasing the number of PMs from 5 to 50 is carried out. The average processing time comparison result is shown in Table 2.

TABLE 2: FIXED VMS WITH DIFFERENT PMs

No: of VMs	**No: of PMs**	**Time Taken**
20	5	154.2
20	10	161.1
20	20	165.3
20	30	170.4
20	50	180.3

Based on the experimental results, the increase in execution time is not very big. The more PMs in data center are racked, the more efficient processing time result will be come out.

Table 3 and Figure 3 show the average VM allocation time comparison between ordinary Sequential Search based model, Binary Search based model and proposed Jump Search based model. In this comparison, the number of VMs is fixed at 20 and the number of PMs is increasing from 5 to 50.

TABLE 3: ORDINARY SEQUENTIAL SEARCH VS BINARY SEARCH VS JUMP SEARCH

No: of PMs	**Avg Time Taken (Sequential Search)**	**Avg Time Taken (Binary Search)**	**Avg Time Taken (Jump Search)**
5	157.6	153.5	154.2
10	170.2	161.2	161.1
20	180.3	165.2	165.3
30	191.6	172.3	170.4
50	206.9	183.5	180.3

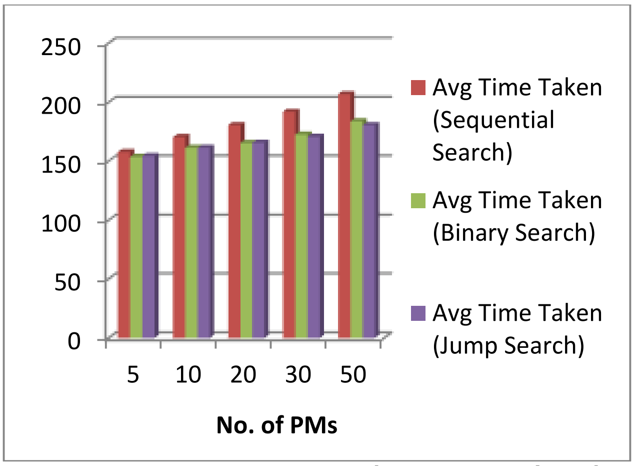

Fig 3. Processing Time Comparison between Sequential Search, Binary Search and Jump Search based models.

According to the results, there is no doubt that the allocation speed of the proposed algorithm is faster than that of ordinary sequential search based model. However, in comparing with Binary Search based model, the speed difference is not quite different. In fact, according to the result, we can adjudge that the more PMs the system tests, the better processing time will be available for the proposed system if we can reduce jumping backward times and increase forward step size. This is because jump search only needs to jump backwards once, while a binary search can jump backwards up to log *n* times. Moreover, the better utilization of PMs is also conducted.

In the future, there will be more comparison experiments carried out with other VM allocation algorithms and policies.

VI. Conclusions

In this paper, an efficient VM Allocation Model with proposed VM Allocation algorithm is presented. Two resources parameters i.e. CPU and memory (RAM) are configured in the proposed VM allocation problem. Jump Search based VM Allocation algorithm is contributed as the mapping method while finding out the most suitable PM to accommodate the VM. By using the proposed model, the faster rate of VM allocation and the more efficient PM usage will be resulted. In the future, this research will continue in developing and testing different VM Allocation Policies with different algorithms and will also make a comparison testing between them.

References

[1] Alexander Ngenzi, R.Selvarani Rangasamy, Suchithra R. Nair, “Dynamic Resource Management in Cloud Data Centers For Server Consolidation”, May 2015, DOI:10.3850/978-981-09-4426-1_004

[2] Coffman EG Jr, Garey MR, Johnson DS (1996) “Approximation algorithms for bin packing: a survey. Approximation algorithms for NP-hard problems”, pp 46–93.

[3] Madnesh K. Gupta, Tarachand Amgoth **, “**Resource-aware virtual machine placement algorithm for IaaS cloud”, The Journal of Supercomputing · July 2017, DOI: 10.1007/s11227-017-2112-9.

[4] Zhuo Tang, Yanqing Mo, Kenli Li and Keqin Li, “Dynamic forecast scheduling algorithm for virtual machine placement in cloud computing environment”, The Journal of Supercomputing, December 2014, DOI: 10.1007/s11227-014-1227-5.

[5] Liu, X.F., Zhan, Z.H., Deng, J.D., Li, Y., Gu, T. and Zhang, J., “An energy efficient ant colony system for virtual machine placement in cloud computing”, IEEE Trans. Evolut. Comput. 2016, 22, 113–128.

[6] Bramantyo Adrian and Lukman Heryawan , “Analysis of K-means algorithm for VM allocation in cloud computing”, 2015 International Conference on Data and Software Engineering (ICoDSE), 25-26 Nov. 2015.

[7] R. N. Calheiros, R. Ranjan and A. Belo, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”.

[8] Khine Moe Nwe, Mi Khine Oo and Khin Maung Htay, “Efficient Resource Management for Virtual Machine Allocation in Cloud Data Centers”, 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), 2018