American Journal of Intelligent Systems 2017, 7(3): 95-99
DOI: 10.5923/j.ajis.20170703.12

# Policy for Resource Allocation in Cloud Computing

**Amitha B.*, Shreenath Acharya**

Department of Computer Science, St. Joseph Engineering College, Mangalore, India

**Abstract** On-demand services such as resources, platform, infrastructure etc. can be provided to the users by using cloud computing environment, which in turn uses virtualization to provide the virtual copies of resources so that depending on users need, they can adjust the resources. In the proposed method, Hungarian algorithm is used to maintain load balance in the VM and Virtual machine allocation policy is used to improve the resource utilization by considering MIPS value and PE. Processing element value of host is used to decide host in which the VM should be created but if two or more hosts have same value the decision is taken based on MIPS value. The results obtained after simulations clearly portrays that proposed mechanism performs better than the considered existing system.

**Keywords** Virtualization, Hungarian algorithm, Virtual machine allocation policy algorithm, PE

## 1. Introduction

Cloud computing is used everywhere in the modern world. Benefits of Cloud computing such as scalability, processing speed, on-demand service, cost etc. makes it more popular along with that virtualization technique makes it more reliable, location independent to the user. Using virtualization we can increase/decrease the cloud capacity by providing virtual copies. We have many providers for cloud environment in that some are free/trial versions.

Now a days applications, services uses cloud environment as computing system, database and servers to complete their requirement. Since we can rent cloud system for days/months/years to complete their jobs many companies, developers are using this since it will avoid initial investment and overhead of maintenance. Variety of services are provided like SaaS, PaaS, IaaS but we focus on IaaS type of services.

There are different parameters, conditions which we need to take care in cloud computing to get optimized result .So in the proposed methods we take care of Load balancing and Resource utilization. Dynamically load is balanced by using Hungarian algorithm and it also reduces the time required for execution. Virtual machine allocation policy is used to improve resource utilization. In proposed methods we find assignments for task to cloudlets by using Hungarian algorithm and cloudlets to host by using Virtual machine allocation method.

Proposed methods are compared using CloudSim simulator tool which gives an idea of how this project works in real environment. So using this tool different possibilities are checked for the proposed methods which proves that proposed method is more efficient than existing method.

Rest of the paper is organized as follows: Section 2 presents the literature survey, Section 3 describes the problem statement, Section4 explains the proposed methods and finally Section5 tells about conclusion.

## 2. Literature Survey

Nagesh Hawanna, et al. have done research on resource utilization and load balancing by taking many attributes and SLA (service level agreement) to improve the utilization of resource and performance. Used Hungarian algorithm, virtual machine allocation technique to get mapped resource at less time. Results of existing system and proposed system are compared [1].

Weiwei Lin, et al. [2] has explained allocation of resources and they have proposed mapping method. In Existing method suspension of the application is required so authors proposed a new approach that assigns vm only if host has the required minimum resources. They proposed threshold-based dynamic resource allocation scheme to allocate resources depending on load changes (dynamic way). This helps in the resource utilization and to decrease the cost.

C. Valliyammai, et al [3] have discussed about resource allocation policies and classifications. Authors have mentioned that fluctuating workload needs to be managed. Challenges and issues in allocation of resources are discussed. Different allocation systems of cloud and the constraints, methodology used are compared.

Swathi Saxena, et al. [4] proposed a market-driven auction technique for demand based allocation of resources and based on payment capability the user is identified. Authors compared proposed technique with VCG auction mechanism

* Corresponding author:
amithabrao@gmail.com (Amitha B.)
Published online at http://journal.sapub.org/ajis

and have got optimized result.

S.Padmavathi, et al. [5] have focused on contention of resource between the tasks so for that, authors proposed priority based pre-emption policy to improve the utilization of resources and it provides less overhead in execution.

Kuan-Lung Huang, et al. concerns on load balancing when multiple vms run on host so they proposed DRA for optimization. They showed that vm will move to other machine if current machine has more load. The main objective was to support DRA, so authors have used Global Load balancing algorithm and results are verified by HPCC s/w [6].

Jiayin Li, et al. have explained adaptive resource allocation in cloud .To implement this they have proposed an allocation algorithm [7] for pre-emptable tasks. Using this technology they check recent task execution and they will adjust the resource allocation and also this can work in contention situation based on the updated information from adaptive method. Parallel processors are used to get high performance.

V. Vinothina, et al. has explained different challenges and resources allocation methods. This paper helps researchers and users to understand cloud computing. The authors [8] have discussed different methods with advantages and their disadvantages.

# 3. Problem Statement

Let us consider,

VM1, VM2, VM3 represents VMs
CLT1, CLT2, CLT3 represents Cloudlets
HT1, HT2, HT3 represents Hosts
CT1, CT2, CT3 represents Cost in rupees

The load balancing is carried out by considering the Cost c i.e.

CLT1, CLT2, CLT3 → VM1, VM2, VM3 only if CT1, CT2, CT3 is minimum.

Utilization of resource is calculated by considering Free_PE, MIPS.

Virtual Machine V1 is assigned to H only if HT1 (Free_PE) = MinFreePE ( ),

If HT1 (PE) = HT2 (PE) then

Consider HT (MIPS) based on that assign host to VM.

# 4. Proposed Methodology

### 4.1. Method Description for Hungarian Algorithm

During the creation of VM all the values are kept same for all the VMs, except MIPS

Step 1: Initialize CostMatrix

CostMatrix[ i ][ j ] = length / MIPS

**Table 1.** Parameters constant in VMs

| Parameters | Size | Ram | Bandwidth | Pes_No |
|---|---|---|---|---|
| Value | 10000 | 2048 | 1000 | 2 |

**Table 2.** MIPS values for VMs

| Parameters | VM1 | VM2 | VM3 | VM4 |
|---|---|---|---|---|
| MIPS | 25 | 50 | 25 | 50 |

During the creation of Cloudlet all the values are kept same for all the Cloudlets, except length.

**Table 3.** Parameters constant in cloudlets

| Parameters | FileSize | OutputSize |
|---|---|---|
| Value | 10000 | 2048 |

**Table 4.** Length values for cloudlets

| Parameters | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| length | 16000 | 12000 | 8000 | 4000 |

Step 1: Determine the CostMatrix and then apply Hungarian algorithm.

**Table 5.** CostMatrix

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 640 | 480 | 320 | 160 |
| VM2 | 320 | 240 | 160 | 80 |
| VM3 | 640 | 480 | 320 | 160 |
| VM4 | 320 | 240 | 160 | 80 |

We find min_ele_row and subtract all the element from this.

**Table 6.** CostMatrix value after subtracting min_ele_row

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 480 | 320 | 160 | 0 |
| VM2 | 240 | 160 | 80 | 0 |
| VM3 | 480 | 320 | 160 | 0 |
| VM4 | 240 | 160 | 80 | 0 |

We find min_ele_col and subtract all the element with min_ele_col.

**Table 7.** CostMatrix value after subtracting min_ele_col

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 240 | 160 | 80 | 0 |
| VM2 | 0 | 0 | 0 | 0 |
| VM3 | 240 | 160 | 80 | 0 |
| VM4 | 0 | 0 | 0 | 0 |

Step 3: Cover zeros by drawing a lines

**Table 8.** CostMatrix along with crossed element

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 240 | 160 | 80 | 0 |
| VM2 | 0 | 0 | 0 | 0 |
| VM3 | 240 | 160 | 80 | 0 |
| VM4 | 0 | 0 | 0 | 0 |

We have three lines which is less than 4, so find smallest element which is un-covered by drawing a line and subtract every element and then add smallest element to covered elements.

**Table 9.** CostMatrix values (after subtracting crossed element)

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 160 | 80 | 0 | -80 |
| VM2 | -80 | -80 | -80 | -80 |
| VM3 | 160 | 80 | 0 | -80 |
| VM4 | -80 | -80 | -80 | -80 |

**Table 10.** CostMatrix (after drawing lines)

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 160 | 80 | 0 | 0 |
| VM2 | 0 | 0 | 0 | 0 |
| VM3 | 160 | 80 | 0 | 0 |
| VM4 | 0 | 0 | 0 | 0 |

Step 4: Determine the mapping between tasks to VMs and calculate the total cost.

**Table 11.** Mapping values

| | CLT1 | CLT2 | CLT3 | CLT4 |
|---|---|---|---|---|
| VM1 | 0 | 0 | 0 | 0 |
| VM2 | 0 | 0 | 0 | 0 |
| VM3 | 0 | 0 | 0 | 0 |
| VM4 | 0 | 0 | 0 | 0 |

VM1 -> CLT 4
VM2 -> CLT 3
VM3 -> CLT 2
VM4 -> CLT 1
Now we check the CostMatrix value and we get
TotalCost = 320+240+320+160 = 1040 Rs.

### 4.2. Method Description for Virtual Machine Allocation Policy

Proposed system will give solution to the following problem i.e. if more than 2 hosts have Pesfree same then for the given task from which host we have to assign the PE in existing system we assign randomly but in proposed method we are assigning by considering other factors also so it gives efficient result.

Example:

Step 1: Enter the value for Free_Space, Req_Space for 3 host and 3 VM.

**Table 12.** Free_Space, Req_Space values for VM policy allocation

| | Host_1 | Host_2 | Host_3 |
|---|---|---|---|
| Free_Space | 50 | 70 | 30 |
| Req_Space | 10 | 25 | 20 |

Step 2: Calculate the diff_Matrix.

**Table 13.** diff_matrix values for VM policy allocation

| | Host_1 | Host_2 | Host_3 |
|---|---|---|---|
| diff_matrix | 40 | 45 | 10 |

Step 3: Assign the host to VM.

Minimum diff_matrix value will be assign to VM1. Sim ilarly find the next minimum value and assign to next VM. Finally we get

VM 1 → Host_3
VM 2 → Host_1
VM 3 → Host_2

Step 4: Find the Free_Space, Used_Space after assigning VM to host.

**Table 14.** Free_Space, Used_Space values for VM policy allocation

| | Host_1 | Host_2 | Host_3 |
|---|---|---|---|
| Free_Space | 40 | 25 | 10 |
| Used_Space | 10 | 45 | 20 |

This method is used to increase the utilization of resources by making all host busy instead of making few host idle.

### 4.3. Pseudo Code for Hungarian Algorithm

1. Initialize Cost_Matric[p] [q] = length / MIPS
2. Find the Reduce_Matric
3. **for** each row q = 1 to n
4.Cost_Matric[r][q]=Cost_Matric[r][q]–Min_Ele_row[p];
5. **end for**
6. **for** each col p = 1 to n
7.Cost_Matric[p][c]=Cost_Matric[p][c]–Min_Ele_col[p];
8. **end for**
9. Compute Line_Matric
10. line = find the Min_No_Of_Line ( )
11. **if**( line < VM )
12. **for** p = 1 to end
13. **for** q = 1 to end
14. **if** element is not covered then
15. Cost_Matric[p] [q] = Cost_Matric[p] [q] – Min_Ele;
16. **end if**
17. **end for**
18. **if** element is covered then
19. Cost_Matric[p][q] = Cost_Matric[p][q] + Min_Ele;
20. **end if**
21. **end for**
22. **end if**
23. FindMapping ( ) for all Cloudlets

### 4.4. Pseudo code for VM Allocation Policy

1. Initialize the Pefree and Peused
2. Pefree, Pesused←Calculate(hostvalues)
3. Repeat step 4, 5 for all vm request
4. Determine the DiffSpace
5. **for** p = 1 to No_of_Host
6. Diffspace[p] = Pefree[p] – Pereq
7. **end for**

8. Assign host to VM which has less DiffSpace
9. HostId = min(DiffSpace)
10. host[HostId ] = VM [VmId ]
11. Update Pefree, Peused and DiffSpace
12. **for** q = 1 to No_Of_Host
13. Pefree[q] = Pefree[q] – Pereq
14. **end for**

### 4.5. Flow of the Project

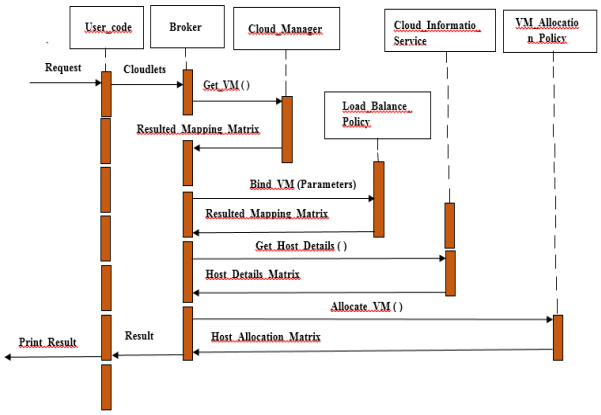


**Figure 1.** Flow of the project

Figure 1 shows how assignment of vm to cloudlet and host to vm is done using proposed methods.

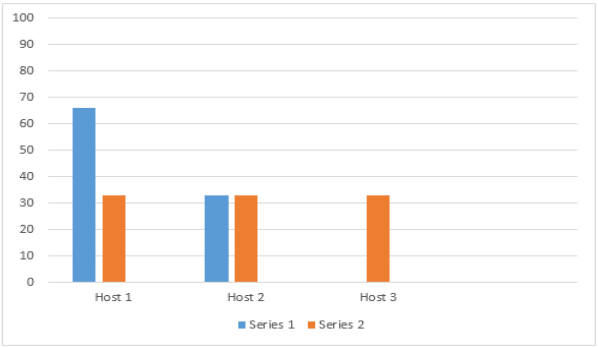


**Figure 2.** Resource Utilization using VM Allocation policy

In Figure 2 Orange color indicates resource utilization using proposed method and blue color indicates resource utilization without using Virtual machine allocation policy which shows resource utilization by making host busy.

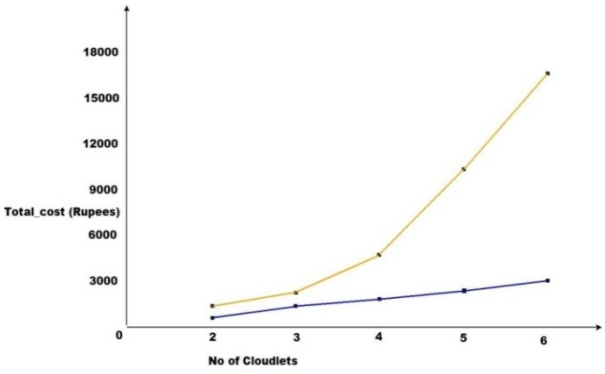


**Figure 3.** Representation of total cost

In Figure 3 Blue color indicates total cost using proposed method and orange color indicates without using Hungarian method.

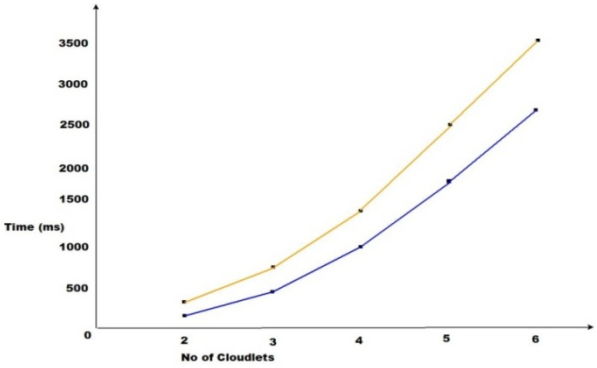


**Figure 4.** Representation of total time

In Figure 4 Blue color means time using proposed method and orange color means without using Hungarian method.

## 5. Conclusions

The main factors of consideration in cloud computing from the resource provisioning context are load balancing and utilization of resources. The proposed method uses Hungarian algorithm and the modified VM allocation policy. The results after comparison with the existing system proves Hungarian algorithm to be more efficient and cost effective. Modified VM allocation policy used to improve the resource utilization by considering processing element and MIPS of the host shows that it would be able to balance the load and improve the utilization of the resources by comparison with the existing VM.

The proposed system can be further improved by considering more parameters during resource allocation.

## REFERENCES

[1] Nagesh Hawanna, Kushang Parikh, Haleemap. k, Jayasubalakshmi R and N.Ch.S. Nlyengar, "Virtual Machine Allocated Policy in Cloud Computing Using Cloudsim in Java", Vol.8, No.1, pp.145-158, ISSN: 2005-4262, 2015.

[2] Weiwei Lin, James Z. Wang, Chen Liangc, Deyu Q, "A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing", Elsevier Ltd, 693-703, ISBN: 978-1-4799-8890-7, 2011, USA.

[3] S. ThamaraiSelvi, C. Valliyammai and V. Neelaya Dhatchayani, "Resource Allocation Issues and Challenges in Cloud Computing", IEEE International Conference on Recent Trends in Information Technology, ISBN: 978-1-4799-4989-2/14, 2014, Chennai.

[4] Narander Kumar, Swati Saxena, "A Preference-based Resource Allocation In Cloud Computing Systems", 3rd International Conference on Recent Trends in Computing, 104 – 111, Lucknow, 2015, doi: 10.1016/j.procs.2015.07.375.

[5] Saraswathi AT, Kalaashri. Y.RA, Dr.S. Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing", Elsevier, 30-36, 2015, Madurai, doi: 10.1016/j.procs.2015.03.180.

[6] Chao-Tung Yang, Hsiang-Yao Cheng and Kuan-Lung Huang, "A Dynamic Resource Allocation Model for Virtual Machine Management on Cloud", Springer-Verlag Berlin Heidelberg, pp: 581–590, 2011, Taiwan.

[7] Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen, Zhong Ming, "Adaptive Resource Allocation for Preemptble Jobs in Cloud Systems", 10th International Conference on Intelligent System Design and Application, ISBN: 978-1-4244-8136-1/10, 2011.

[8] Justin Y Shi, MoussaTaifi, Abdullah Khreishah, "Resource Planning for Parallel Processing in the Cloud", IEEE 13th International Conference on High Performance and Computing, 2011.

[9] http://azure.microsoft.com

[10] http://en.wikipedia.org/wiki/Cloud-computing