Dynamic virtual machine allocation policy in cloud computing complying with service level agreement using CloudSim

Dynamic virtual machine allocation policy in cloud computing complying with service level agreement using CloudSim

Parikh Aneri, Sumathy S*,
School of Information Technology and Engineering, VIT University, Vellore-632014, India.

*Email: ssumathy@vit.ac.in

Abstract. Cloud computing provides services over the internet and provides application resources and data to the users based on their demand. Base of the Cloud Computing is consumer provider model. Cloud provider provides resources which consumer can access using cloud computing model in order to build their application based on their demand. Cloud data center is a bulk of resources on shared pool architecture for cloud user to access. Virtualization is the heart of the Cloud computing model, it provides virtual machine as per application specific configuration and those applications are free to choose their own configuration. On one hand, there is huge number of resources and on other hand it has to serve huge number of requests effectively. Therefore, resource allocation policy and scheduling policy play very important role in allocation and managing resources in this cloud computing model. This paper proposes the load balancing policy using Hungarian algorithm. Hungarian Algorithm provides dynamic load balancing policy with a monitor component. Monitor component helps to increase cloud resource utilization by managing the Hungarian algorithm by monitoring its state and altering its state based on artificial intelligent. CloudSim used in this proposal is an extensible toolkit and it simulates cloud computing environment.

1. Introduction

Cloud computing provides various services such as infrastructure as a service, platform as service, and software as a service. These services are based on a pay-as-you-use model to cloud customers. It has favored transforming the large IT industry, making software service more attractive and easy to consume. From the user's perspective, cloud computing is allows to use and deploy their application from anywhere and anytime based on their demands. To provide these types of services continuously based on customer demand, various technologies such as virtualization, clustering and application server are deployed. Virtualization is used in the cloud computing for virtualizing operating system, storage system and data center. These applications may have distinctive setup and diverse engineering like informal communication, web facilitating, content conveyance and constant information preparing. Virtualization innovation is the heart of distributed computing lifecycle and it is constrained to physically accessible assets.

Henceforth, usage of all assets assumes a critical part in dealing with the distributed computing life cycle proficiently keeping in mind the end goal to convey these tremendous applications effectively and productively. Stack adjusting is required to deal with the heap powerfully in distributed computing condition and subsequently, it frames a basic piece of distributed computing life cycle. Cloud computing satisfies huge requirements by using the virtual machines (VMs). Resources are applied to VM based on requirements for the application. VM is used to deploy this application in cloud computing environment. Such applications may have different types of requirements, configuration and architecture such as web hosting, real-time data processing, social networking, etc. As virtualization is the centroid of cloud computing life cycle, utilization of these resources requires virtual machine which would effectively utilize all the resources.

The proposed system also provides dynamic load balancing. It contains many parameters such as execution time, memory utilization, etc. Dynamic load balancing needs dynamic resource allocation policy with cost optimization, which is realized using Hungarian algorithm. CloudSim, simulation toolkit that enables simulation of cloud computing environment is used.

2. CloudSim Architecture

Cloud is a kind of parallel framework which comprises of an accumulation of interconnected and virtualized Personal Computers. The proposed System for Dynamic load balancing and dynamic resource allocation for Cloud Computing Environment is deployed using CloudSim. The major entities of CloudSim, is described in figure1.

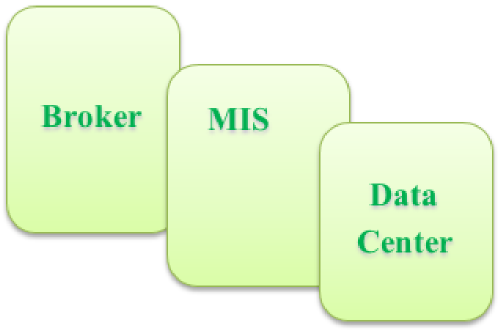


Figure 1. Entities in CloudSim

CloudSim is a simulation toolkit that provides simulation of cloud computing environments using virtual machines which need to be managed. It provides data center to simulate and deploy the application. It also provides mapping for utilizing the resources. In CloudSim, there are major entities like Data Center, Management Information System, and Cloud broker[15, 16].

In CloudSim simulator, cloud user requests the provider with different types of configurations like, Number of Processors, Memory, I/O file Size, Storage etc. All these requests are tasks which are managed by CloudBroker. In this model, all resources use single shared memory as a single pool technique. The datacenter has physical resources which contains different types of hardware configuration. There is a Management Information System (MIS) which acts like data repository. Cloud Broker is allowed to access data centers which are registered with MIS. CloudSim generates number of Virtual Machines and assigns task to them to simulate the data center. Here, task is denoted as a Job. Cloudlet contains list of different configuration based on cloud user request to deploy their application, so that Cloud Broker maps the cloudlet with Virtual machine and assigns the job with which Virtual Machine fulfills all the requirements.

CloudSim allows modeling different policies like scheduling policy [13] , distributed networking policy and so on. These polices are used in binding cloudlet to Virtual Machine. Flow diagram in figure 2 gives the clear relationship between these entities.

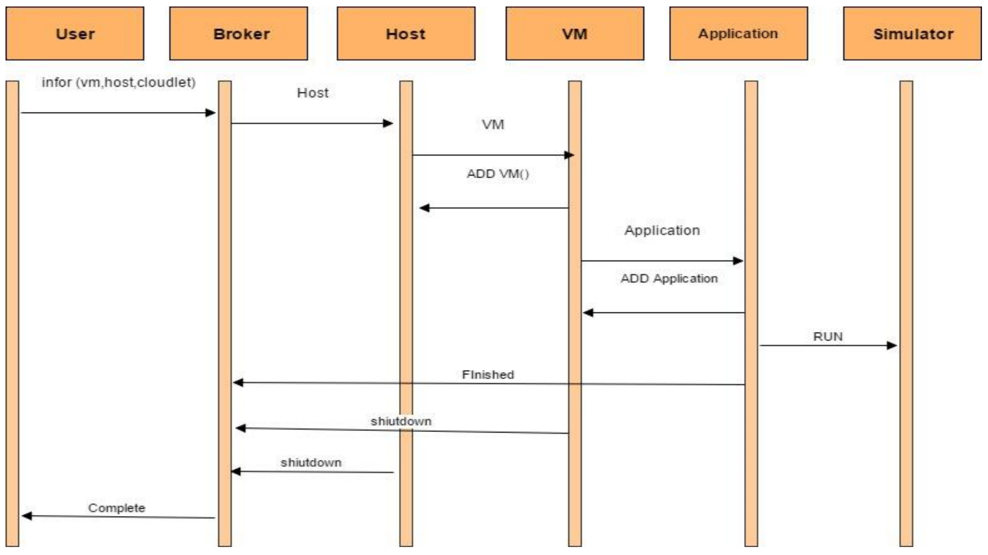


Figure 2. Flow Diagram CloudSim architecture

The user, broker, host, VM, application and the simulator entities are indicated. Cloud broker manages all the requests of users. The broker checks the specification which is needed to execute the task and also checks for the host which is free. Subsequently, the required configuration for the specific application is generated on applying to the virtual machine which generates the results.

3. Literature Survey

Endo, P. T et al [3], describes the challenges which create major effect on resource allocation in cloud computing. Focus is more on resource management policy and not on resource allocation method. Qiang Li, Qinfen Hao, Limin, Xiao , Zhoujun Li. et al [1], describe the architecture using feedback control theory and virtualization. All hardware resources are stored in a single place in memory in a virtual machine which has sharing architecture and cloud customer can request as per Service Layer agreements (SLA). CPU memory and controllers are used. The main purpose of the paper is to control different virtualized valuable usage to secure SLA agreements by using various control input for every VM. Problem with this type of architecture which is based on virtual machine architecture is on how to handle resource requests for every application within the time limit based on workload.

Li, J. Y. et al [2], describes an adaptive resource allocation algorithm for resource allocation. An adaptive min-min scheduling is used which is applicable for only static allocation. Majumdar, S. [5], describes a two-layer architecture for resource allocation policy in virtualization. There are mainly two types such as dynamic and static which depends on a global agent and a local agent. The agent computes current request and transfers it to a global agent. Global agent's main role is to optimize the current request and approve the configuration. This scenario between these two agents is very important for this architecture and any changes to the agent should be changed.

Shei, J. Y. et al [4], describe resource scheduling based on cloud computing. They describe resource planning in multiple dimensions like space or time. Author has used EC2 technique which is developed by Amazon for cloud computing environment. Oiza, B et al [6], describes the concept of CloudSim , which is a simulation tool kit that is used in cloud computing as simulation. Author describes about cloud broker strategy using different parameters. Chao-Tung Yang et al [18], discussed various methods to decrease the unpredictability of access to cloud assets by the clients. They quantified the execution of live movement of virtual machines with various particulars and broke down the information. Virtual Migration problem which accrued in tradition virtualization concept is discussed and have proposed a system to overcome with a dynamic virtualization technique.

Geetha ,V. et al [7], provide a survey on various classes of existing cloudlet scheduling algorithms and give a review on different classes of existing cloudlet planning calculations. Cloudlet behaves as a workload in cloud computing. Planning arrangement can be connected at two levels in cloud condition: PaaS layer and Service Level Agreement (SLA). They describe various policies related to virtual machine management for describing a software framework which mainly is used for simulation in cloud computing. In this strategy, data center resources are used for simulation. Duemitresecu,C. E. L. and Fostera, I. R. [8], describe a simulation technique termed GangSim. GangSim is used as a simulation technique for grid computing environment with a research between local and community resource allocation.

Muni Sekhar ,V. et al [11], described cloud computing to provide on demand service to shared network. So, the service should be customized with Quality of service. Quality of service is fulfilled with non-functional specifications like security, mobility and virtualization using OCSA algorithm. To provide guaranteed QoS, this model analyzes through DREAD. Peng Yang. et al [12], propose a virtual resource scheduling model for distributed communication network leveraged by the research on resource virtualization technology and the Software Defined Network (SDN).

4. Proposed Framework

.

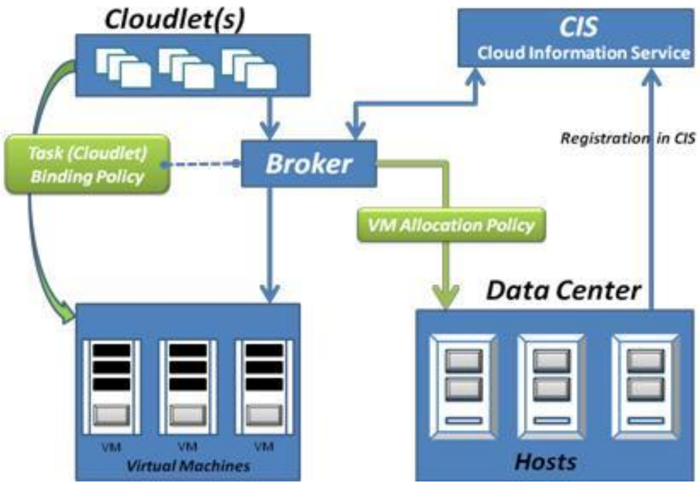


Figure 3. Resource allocation policy in cloudSim Environment

Focus of the proposed work is on a connection between Virtual Machine and cloudlet through the mapping that can be balanced between cloud request of users and Virtual Machine allocation strategy. There is a need to handle cloud request when requested by the user. To fulfill all these requests there is a need for resources but hardware resources are in limited number. Resource management is a big issue in cloud computing. Resource allocation policy in order to assign resources to suitable task considering minimal cost and maximum performance is a challenge. So, Virtual machine allocation policy is very important in this scenario. To solve this in less time and achieve effective performance, Hungarian algorithm is applied [14, 17].

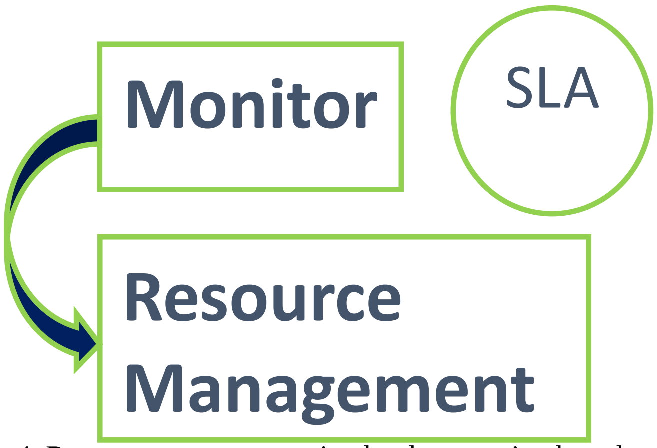


Figure 4. Resource management in cloud computing based on SLA.

As seen in filgure 4, a monitor process keeps track of resouce management policy periodically and analyze the current status of the Application and Virtual Machines. A new matrix of the resources and virtual machine is created and the resouce management policy is applied again on the newly created matrix. This paper proposes use of the Hungarian Algorithm for resouce managament policy in the field of the cloud computing. Monitor optiomizes the resources further along with the resouce management policy based on user SLA. For example, at time t1, virtual_machine_A does job for Application_A, virtual_machine_B for Appliacation_B and virtual_machine_C for Application_C. Virtual_machine_A has free procesing elements but Application_B and Application_C require more number of the processing elements at this stage. Now, at time t2, virtual_machine_A can also work for Application_B along with Application_A. Monitor identifies the task and resource mangement policy is applied again on current status so as to free virual_machine_B.

5. Hungarian Algorithm

Notation: Here, cloud matrix C_M, where m × n matrix with m as the number of VM and n is number of cloudlets. L_M is line matrix that covers all zeros in R_M. Here, R_M is the reduced matrix.

Step 1: Initialize C_M.

C_M[i][j] = C_P / VMparameter

First, it will check if the matrix is a square matrix or not. If number of virtual machines and number of cloudlets are not equal, then there is need to add dummy cloudlets with values of zero.

Step 2: Compute R_M.

For every row, j=1 to n
C_M[row][j] = C_M[row][j] – Min_Element_row [i];

For every column, I =1 to n
C_M[i][column]=C_M[i][column]– Min_Element_col[i];

Step 3: Compute L_M

If number of line < VM

C_M[i][j] = C_M[i][j] – min; if elements are covered by two lines
C_M[i][j] = C_M[i][j] + min;

If the numbers of lines which are covered by zero are not equal to a number of VM's, then find the minimum value of the unCross_Element and subtract it from all unCross_Elements. And the point where lines intersect each other in R_M, that point is increased by one.

Step 4: Mapping

Find possible arrangements.

Step 5: Stop

VM (Virtual Machine) Allocation Policy

Used Elements store the number of used processing elements for particular host. Diff_Matrix is a one dimensional matrix that stores different processing elements required and available processing element for every host in data center.

Algorithm for VM (Virtual Machine) Allocation

Step1 Cloud Matrix Initialization

Available_Elements and Used_Elements

Available_Elements, Used_Elements $\leftarrow$ calculate host

Step 2 find the Diff_Matrix

For i=0 to number of total host
Diff_Matrix[i] Available_Element[i]-Used_Elements

End for.

Step 3 Assign VM to least Diff_Matrix

Host_Id = min(Diff_Martic)
Host(Host_Id) =Vm(Vm_Id)

Step 4 Update all the matrix.

Calculate the Available Elements which are allocated to host and add it to Used_Elements and again calculate Available_Elements, Used_Elements and Diff_Matrix.

Here, for [i] is total number of host

Available_Element[i]=Available_Elements[i]- Used_Elements[i].

End for.

6. Illustrations and Results

Assume three virtual machines created in cloud with the following data.

	Virtual machine 1	Virtual machine 2	Virtual machine 3
Vmid	0	1	2
MIPS	250	500	250
Size	10000	10000	10000
Ram	2048	2048	2048
Bandwidth	1000	1000	1000
pesNmuber	2	3	2

	Cloudlet 1	Cloudlet 2	Cloudlet 3
Id	0	1	2
Length	40000	80000	120000
File Size	300	10000	10000
Output Size	300	2048	2048

Step 1: Initialize C_M.

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	80	160
VM 2	0	0	0
VM 3	0	80	160

Step 2: Compute the L_M

The number of lines is not equivalent to number of VM and minimum uncrossed components. Here 80 is the minimum component and crossing point position is R_M [1][0]. On subtracting minimum element and augmenting by one in all crossing point in R_M, outcome is created.

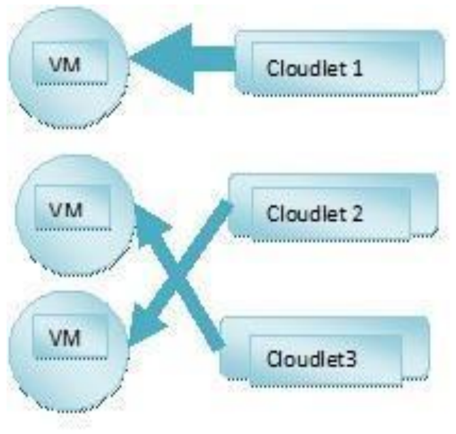
	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	0	80
VM 2	1	0	0
VM 3	0	0	80

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	0	80
VM 2	1	0	0
VM 3	0	0	80

Step 3: Find the Mapping

There are numerous potential outcomes for mapping and discovering each of these conceivable outcomes in NP finished class. Figure 5 indicates the Virtual Machine Mapping.

Line number	
Virtual machine 1	Cloudlet 2
Virtual machine 2	Cloudlet 3
Virtual machine 3	Cloudlet 1



Virtual Machine allocation policy:

	Host 1	Host2	Host3	Host 4	Host 5
Id	0	1	2	3	4
MIPS	250	250	250	500	250
Ram	2048	2048	2048	2048	2048
Storage	1M	1M	1M	1M	1M
Bandwidth	1000	1000	1000	1000	1000
pesNmuber	20	5	10	20	10

Step1 Initialize the Available_Elements and Used_Elements

	Host1	Host2	Host3	Host4	Host5
Available_ Elements	30	50	15	20	10
	Host1	Host2	Host3	Host4	Host5
Used_Ele ments	0	0	0	0	0

Step2 Find the Diff_Matric

	Host1	Host2	Host3	Host4	Host5
Diff_Matric	10	30	0	10	0
Step3 Allocate Vm to host which have least					
Hence vm1 will allocated to host 3					
Step4 Update all the metric					
	Host1	Host2	Host3	Host4	Host5
Available_ Elements	30	50	5	20	10
	Host1	Host2	Host3	Host4	Host5
Used_Ele ments	0	0	10	0	0

So, VitualMachine-2 will assign to host-3 and VirtualMachine-3 will assign to host-5 and VirtualMachine-4 will assign to host-4 and VirtualMachine-5 will assign to host-1.
The results are as follows.

	Host 1	Host 2	Host 3	Host 4	Host 5
Available_ Elements	30	50	15	20	10
	Host 1	Host 2	Host 3	Host 4	Host 5
Used_Elem ents	0	0	0	0	0

7. Conclusion

CloudSim provides simulation and modeling in cloud computing. In cloud computing, cloudBroker has huge number of tasks and huge number of resources, it uses least number of the resources to provide virtualization. CloudBroker can assign tasks to virtual machine which has the same specification as per user request. So, in cloud computing, resource allocation task plays a vital role. Hungarian algorithm maintained a fair distribution flow between Virtual machines and Cloudlets. Therefore, it increases the throughput of computing environment. Monitoring component impact a lot on managing the workload and resource utilization, it helped current resource utilization and assign pending task to other virtual machine such that all task can be completed with least number of the virtual machines. Thus, it increases the cloud resource utilization. In future, the proposed system can be extended and implemented with parallelism to reduce processing time and thereby the response time.

References

[1] Li, Q., Hao, Q., Xiao, L., & Li, Z. (2009, December). Adaptive management of virtualized resources in cloud computing using feedback control. *1st IEEE International Conference on Information Science and Engineering (ICISE),* pp. 99-102.

[2] Li, J., Qiu, M., Niu, J. W., Chen, Y., & Ming, Z. (2010, November). Adaptive resource allocation for preemptable jobs in cloud systems. *10th IEEE International Conference on Intelligent Systems Design andApplications (ISDA),* pp. 31-36.

[3] Endo, P. T., de Almeida Palhares, A. V., Pereira, N. N., Goncalves, G. E., Sadok, D., Kelner, J. & Mangs, J. E. (2011). Resource allocation for distributed cloud: concepts and research challenges, *IEEE network*, *25*(4), pp. 42-46.

[4] Shi, J. Y., Taifi, M., & Khreishah, A. (2011, September). Resource planning for parallel processing in the cloud. *13th IEEE International Conference High Performance Computing and Communications (HPCC),* pp. 828-833.

[5] Majumdar, S. (2011). Resource Management on cloud: Handling uncertainties in Parameters and Policies. CSI Communication, 22, pp.16-19.

[6] Limbani, D., & Oza, B. (2012) A proposed service broker strategy in cloud analyst for cost-effective data center selection. *International Journal of Engineering Research and Applications, India*, *2*(1), pp. 793-797.

[7] Geetha, V., Devi, R. A., Ilavenil, T., Begum, S. M., & Revathi, S. (2016, February), Performance comparison of cloudlet scheduling policies. *IEEE International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS),* pp.1-7.

[8] Dumitrescu, C. L., & Foster, I. (2005, May). GangSim: a simulator for grid scheduling studies. *IEEE International Symposium on Cluster Computing and the Grid, 2005. CCGrid 2005,* Vol. 2, pp. 1151-1158.

[9] Legrand, A., Marchal, L. & Casanova, H. (2003, May). Scheduling distributed applications: the Simgrid simulation framework. *In 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings. CCGrid,* pp. 138-145.

[10] Buyya, R., & Murshed, M. (2002). Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and computation: practice and experience*, *14*(13-15), pp. 1175-1220.

[11] Sekhar, V. M., Kumar, M. R., Rao, K. V., & Rao, N. S. (2013, August), Guaranteed Quality of Service in Cloud Ready Application. *IEEE International Symposium on Computational and Business Intelligence (ISCBI),*pp. 24 – 28.

[12] Yang, P., Ma, Z., Song, Y., Zhang, Q., Liu, G., & Hou, X. (2016, October). Research on the application of virtual resource scheduling technology in distribution communication network. *2nd IEEE International Conference on Cloud Computing and Internet of Things (CCIOT),* pp. 68-72.

[13] Frincu, M. E., & Craciun, C. (2011, December). Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments.*Fourth IEEE International Conference on Utility and Cloud Computing (UCC),* pp. 267-274.

[14] Tordsson, J. , Montero ,R. S., Vozmediano ,R. M. and Llorente, I. M. (2011). Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers, *Future Generation Computer Systems.*, Vol. 28, No. 2,. pp. 358–367.
[15] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, *41*(1), 23-50.
[16] Buyya, R., Ranjan, R., & Calheiros, R. N. (2009, June). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. *IEEE International Conference on High Performance Computing & Simulation, HPCS'09.* pp. 1-11.
[17] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, *2*(1-2), pp.83-97.
[18] C. T. Yang, Y. T. Liu, J. C. Liu, C. L. Chuang and F. C. Jiang, "Implementation of a Cloud IaaS with Dynamic Resource Allocation Method Using OpenStack," *International Conference on Parallel and Distributed Computing, Applications and Technologies, Taipei*, 2013, pp. 71-78.