AN EFFICIENT SERVICE ALLOCATION & VM MIGRATION IN CLOUD ENVIRONMENT

Puneet Dahiya

Department of Computer Science & Engineering
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Murthal, Sonepat-131039
punit13dahiya@gmail.com

Jitender Kumar
Department of Computer Science & Engineering
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Murthal, Sonepat-131039
jitenderkbhardwaj@gmail.com

Abstract

A cloud environment is the popular shareable computing environments where large number of clients/users are connected to the common cloud computing environment to access the resources and the services. The presented work is focused on the concept of effective resource allocation, de-allocation and reallocation in a cloud environment. To present the concept, we have taken a cloud environment with multiple clouds along with multiple virtual machines. These all clouds are assigned by a specific priority. Now as the user request arrive, it performs the request to the priority cloud under its requirements in terms of memory & processor capabilities. When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting. We can also some migration work so that if a cloud is under utilize then its services are migrated to nearest cloud having sufficient utilization. Hence the work provides efficient allocation, de-allocation and reallocation of cloud services with minimum need for migration of service from one cloud to another.

***Keywords*— Resource Allocation, Resource Scheduling, Resource Migration**

1.INTRODUCTION

Cloud computing [1] is a technique to provide and maintain data and applications with the help of internet and remote servers. It provides facility of centralize memory, storage, applications and processing for powerful computing. Through its centralized computing facilities the cloud computing allows users and organizations to use applications and other services without local storage and local location. Cloud computing provides Information Technology (IT) as a service resource.

Cloud environment helps us to create, configure and utilize application remotely. A user only needs to connect the Internet to avail the facility of cloud computing at anytime and at any location. The cloud computing services can be available either through private or public networks. All types of popular services such as mailing, chatting, conferencing are now available through cloud computing.

Cloud computing divided into of two parts or ends called the front part (end) and the back part (end). “The front end includes client’s devices and applications that are required to access cloud. The back end refers to the cloud itself. The whole cloud is administered by a central server that is used to monitor client’s demands”.

The presented work is focused on the concept of effective resource allocation, de-allocation and reallocation in a cloud environment. We can also some migration work so that if a cloud is under utilize then its services are migrated to nearest cloud having sufficient utilization. Hence the work provides efficient allocation, de-allocation and reallocation of cloud services with minimum need for migration of service from one cloud to another.

2. RESOURCE SCHEDULING & ALLOCATION

Two main actors involve in cloud computing are cloud providers and the cloud users. Clouds providers i.e., cloud itself establish the cloud data centers and various resources to be used by cloud users. Cloud users i.e., end users can actually use the cloud resources & pay according to their usage. The basic communication (interaction) between cloud providers and cloud users can be easily understood using Figure 1 below [3].

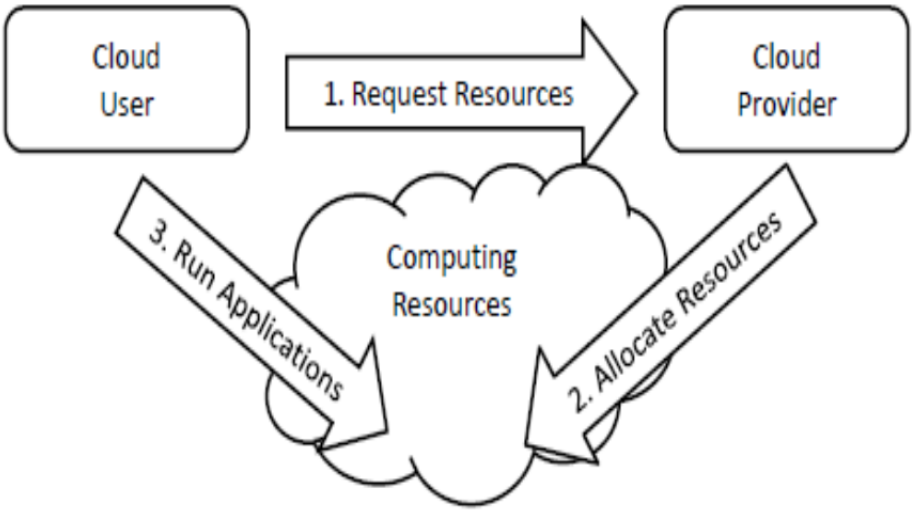

Figure 1: Interaction between cloud actors

The interaction steps are listed below:

1. The cloud user initiates a request for any specific resources to the cloud provider.
2. On receiving the request the cloud provider checks for the availability of specific resource.
3. If resource exists then assign the resource to the requesting user.
4. The user now utilizes the services of assigned resources to perform any specific task or application.
5. When no more service is required then the user releases the resource, pay for the resource & closes the connection.
6. The provider now schedule and allocate the resource to other requesting clients. [5, 6].

One interesting aspect of the cloud computing environment is that these actors or say players are generally from different organization and regions with their own need & interests. "The main goal of cloud providers is to generate as much revenue as possible with minimum investment on cloud infrastructure." To achieve this objective the cloud providers host multiple virtual machines to be used by multiple clients to attain maximum profit.

Energy efficient Cloud resources allocation consists in identifying and assigning resources to each incoming user request in such a way, that the user requirements are met, that the least possible number of resources is used and that data center energy efficiency is optimized. Figure 2 shows the resource allocation and scheduling scheme for cloud computing.

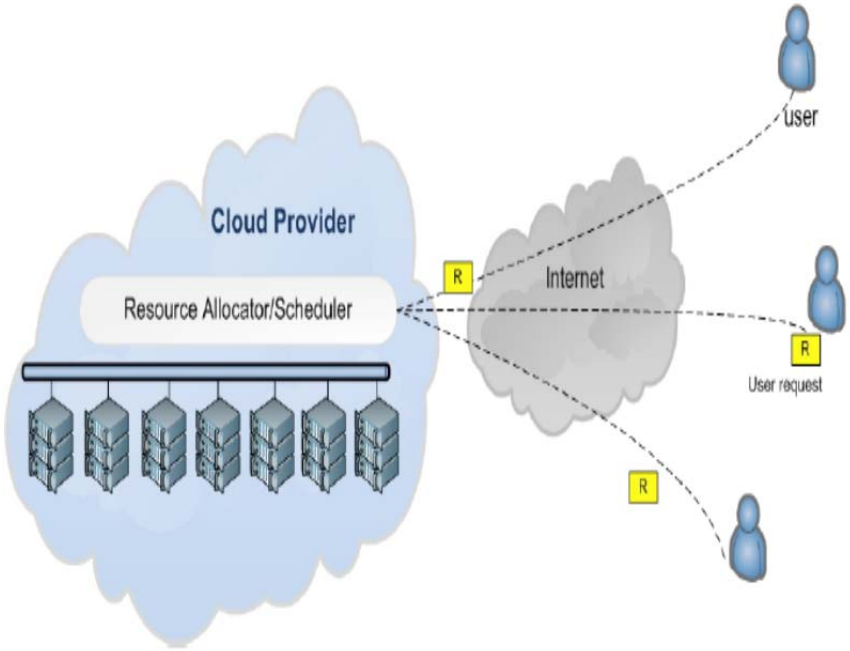

Figure 2: Resource Allocation & scheduling in Cloud Computing

2.1 Resource Allocation

In [8] authors wrote "Resource allocation involves deciding what, how many, where, and when to make the resource available to the user. Typically, users decide the type and amount of the resource containers to request then providers place the requested resource containers onto nodes in their datacenters. To run the application efficiently, the type of resource container need to be well matched to the workload characteristics, and the amount should be sufficient to meet the constraints i.e., job must be completed before its deadline. In an elastic environment like the Cloud where users can request or return resources dynamically, it is also important to consider when to make such adjustments."

2.2 Job Scheduling

In [9] authors wrote "Once the resource containers are given to the user, the application makes a scheduling decision. In many cases, the application consists of multiple jobs to which the allocated resources are given. The job scheduler is responsible for assigning preferred resources to a particular job so that the overall computing resources are utilized effectively. The application also has to make sure each job is given adequate amount of resources, or its fair share. Such a scheduling decision becomes more complex if the environment is heterogeneous."

3. PROPOSED MODEL

The number cloud services on the Internet are increasing in rapid rate. The basic purpose of all of the cloud services is to provide efficient, effective & varied types of services to their clients. The presented work is focused on the concept of effective resource allocation, de-allocation and reallocation in a cloud environment. To present the concept, our work initialize cloud data centers with multiple cloud servers that provide facility of different types of resources. Now as the user request arrive, it performs the request to the priority cloud under its requirements in terms of memory & processor capabilities. As the particular cloud will get the request, it will search for the number of requested processors. If the numbers of processors are available with the current cloud, the resources will be allocated to that particular client. But if the sufficient numbers of processors are not available then the search will be performed for the next particular cloud to perform the resource allocation.

When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting. We can also some migration work so that if a cloud is under utilize then its services are migrated to nearest cloud having sufficient utilization. Hence the work provides efficient allocation, de-allocation and reallocation of cloud services with minimum need for migration of service from one cloud to another.

Four main activities involve in our work are listed below:

• As soon as user requests arrive schedule them on the cloud servers

• Depending upon the capabilities of cloud servers allocate the requests to particular server.
• Deallocate the processes when no services are required by the client.
• Process Migration in overload conditions

Figure 3 shows the middle layer along with cloud servers and multiple clients.

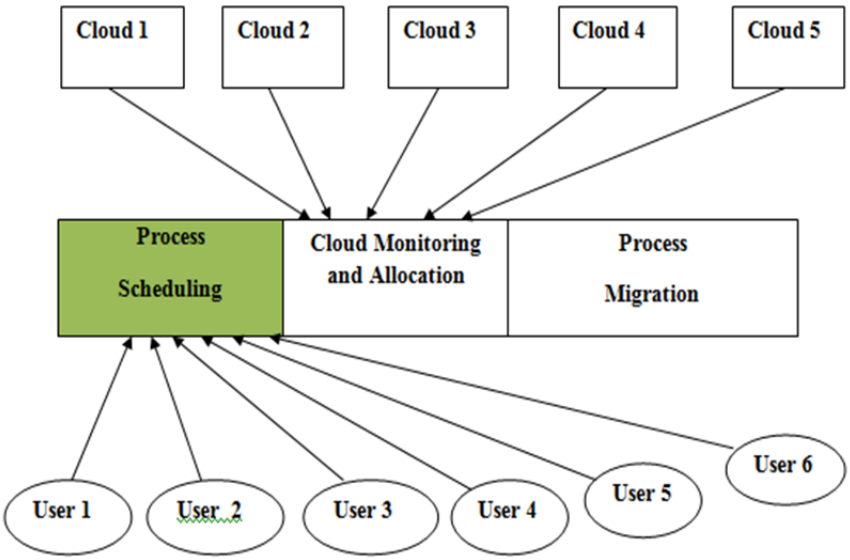

Fig 3: Process Scheduling, Allocation and Migration

Whenever user sends application demands requesting for datacenter resources than each and every request is encapsulated in virtual machine (vm) requests with the help of virtualization to the middle layer. For each vm request in domain V, server from the resource pool of server nodes is allocated with the help of vm scheduler. The vm scheduler calculates the utilization of each running server and finds the most suitable server for placing the vm request according the algorithm proposed and updates the utilization after each placement of vm request. It also ensures that upper threshold (maximum number of services) must not be crossed. The queue manager at vm scheduler maintains a queue of allocated vm requests for each server. No other request can placed on the server until these allocated queues of requests keep executing. On deallocation vm scheduler releases the resources occupied by the client. It checks for job migration from one server to another when server utilization is below minimum threshold level [12, 16].

3.1 Proposed Algorithm

In this paper, we propose the solution of efficient allocation & deallocation of cloud services. The main objectives of our paper are:

• Minimize the number of servers used
• Maximize the resource utilization until upper threshold of utilization is reached
• Optimize the resource allocation strategy by migrating the low utilization server processes to another nearby server.

For achieving the above goals two thresholds: upper threshold and minimum threshold have been defined. When utilization of all the running reaches upper threshold i.e., all the services/resources of a server are consumed then we start the new server for mapping the requests.

Following three algorithms explain the basic concepts of our proposed work on efficient service allocation, de-allocation and migration concepts.

Algorithm: **Allocation Process**

1. Initialize & Start the servers on Cloud Datacenters
2. Accept ClientId & No. of CPUs requested.
3. Start the allocation process for client ClientId on First Server.
4. If remaining Capacity of server is more than the requested CPUs then
(i) Place the request of ClientId on the server.
(ii) Update the utilization Percentage and remaining capacity of server.
(iii) Set allocated status to true for the client ClientId.
(iv) Exit.
Else
(i) Start the allocation process for client ClientId on Next Server.
(ii) Goto Step 4.
[end of If]
5. If no next server available then
Write: “Datacenter is Out of Server”.
6. Exit.

Algorithm: **De-allocation Process**

1. Accept ClientId & No. of CPUs to deallocated.
2. If allocated status is false for clientId then
Write: “No Allocation is performed by the client” & Exit.
3. If No of CPUs more than the allocated CPUs then
Write: “Deallocation of CPUs must be less than or equal to allocated CPUs” & Exit.
4. Perform deallocation from the allotted server.
5. Update the utilization Percentage and remaining capacity of server.
6. Exit

Algorithm: **Migration Process**

1. Set i := 1.
2. Repeat while I <= MaxServer
3. If UtilizationPer of Server server[i] < 40 then
(i) Set sid1 := server[i].
(ii) Set no. of CPUs for migration
(iii) break
[end of If]
[end of Loop]
4. if(I = MaxServer) then Write: “No server is under Utilized” & Exit.
5. Set i := 1.
6. Repeat while I <= MaxServer
7. If UtilizationPer of Server server[i] >= 40 then
(i) Set sid2 := server[i].
(ii) break.
[end of If]
[end of Loop]
8. if(I = MaxServer) then Write: “No server has the required Utilization” & Exit.
9. Update the remaining capacity of server ids sid1 & sid2
10. Update the utilization value of server ids sid1 & sid2
11. Write: "Resource of server id sid1 is migrating to server id sid2.
12. Exit.

4. RESULTS & DISCUSSIONS

We compare result of proposed algorithm with the work of other previous researchers. The metric used for comparison is the number of servers started with respect to number of virtual machines requested. We compare our work with following algorithms:

- Greedy First-fit, "it assigns a VM request to the first scanned physical server that satisfies the demands of all resources for that specific request" [13, 14].
- Min-min, "the VM request with the lowest CPU capacity or lowest minimum completion time requirement is assigned first" [15].
- Load balancing technique, "in this technique requests are fulfilled after checking the minimum and maximum threshold values. If threshold values are satisfied then allocation is performed " [16].

Figure 4 below shows the comparative chart of existing techniques and proposed algorithm for efficient service allocation and deallocation on cloud computing.

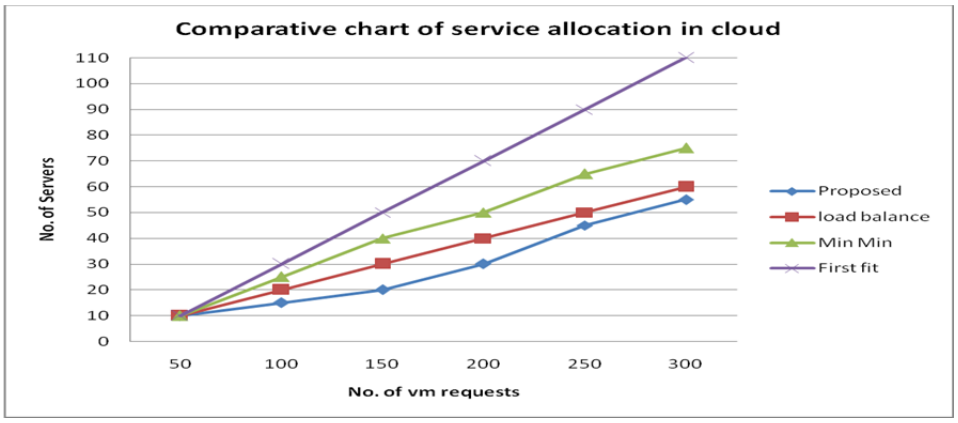

Figure 4: Simulation Results

As seen from the simulation result shown in figure 4 above, it is clear that the proposed algorithm is able to produce better results when compared to first fit, min-min and load balancing algorithm. From the results we can say that the numbers of running servers are being reduced with help of our algorithm thereby avoiding the overload and overheating. The energy consumption is also reduced as less number of servers are started by the proposed algorithm. The total operational cost is also reduced.

5. CONCLUSION

Cloud computing is a technique to provide and maintain data and applications with the help of internet and remote servers. The presented work is focused on the concept of effective resource allocation, de-allocation and reallocation in a cloud environment. As the particular cloud will get the request, it will search for the number of requested processors. If the numbers of processors are available with the current cloud, the resources will be allocated to that particular client. But if the sufficient numbers of processors are not available then the search will be performed for the next particular cloud to perform the resource allocation. When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting. We can also some migration work so that if a cloud is under utilize then its services are migrated to nearest cloud having sufficient utilization. Hence the work provides efficient allocation, de-allocation and reallocation of cloud services with minimum need for migration of service from one cloud to another.

REFERENCES

[1] B. Hayes, "Cloud Computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, Jul. 2008.
[2] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, no. 145, p. 7, 2011.
[3] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, "Cloud Computing, A Practical approach"
[4] D. Warneke, O. Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp 985 - 997, June 2011, DOI: http://doi.ieee computersociety.org/10.1109/TPDS.2011.65.
[5] V. Vinothina, Dr. R. Sridaran, Dr. Padmavathi Ganapathi, "Resource Allocation Strategies in Cloud Computing", International Journal of Advanced Computer Science and Applications [IJACSA], Vol. 3, No.6, 2012. ISSN: 2158-107X (Print), DOI: 10.14569/issn.2156-5570.

[6] Sowmya Koneru, V N Rajesh Uddandi, Satheesh Kavuri, "Resource Allocation Method using Scheduling methods for Parallel Data Processing in Cloud", International Journal of Computer Science and Information Technologies[IJCSIT], Vol. 3(4), 2012, pp 4625 - 4628 4625, ISSN: 0975-9646.
[7] Gihun Jungand, Kwang Mong Sim, "Location-Aware Dynamic Resource Allocation Model for Cloud Computing Environment", International Conference on Information and Computer applications (ICICA 2012), pp 37 - 41, IPCSIT, Vol. 24, IACSIT Press, Singapore.
[8] Thangaraj P, Soundarrajan S, Mythili A, "Resource allocation policy for IaaS in Cloud computing", International Journal of Computer Science and Management Research, Vol 2, Issue 2, pp 1645 - 1649, February 2013, ISSN 2278-733X.
[9] Mohd Hairy Mohamaddiah, Azizol Abdullah, Shamala Subramaniam, and Masnida Hussin, "A Survey on Resource Allocation and Monitoring in Cloud Computing", International Journal of Machine Learning and Computing, Vol. 4, No. 1, February 2014.
[10] Pratik P. Pandya, Hitesh A. Bheda, "Dynamic Resource Allocation Techniques in Cloud Computing", International Journal of Advance Research in Computer Science and Management Studies Volume 2, Issue 1, January 2014.
[11] Nguyen Trung Hieu, Mario Di Francesco, and Antti YlaJaaski, "A Virtual Machine Placement Algorithm for Balanced Resource Utilization in Cloud Data Centers", 2014 IEEE International Conference on Cloud Computing.
[12] B. Rajasekar, S. K. Manigandan, "An Efficient Resource Allocation Strategies in Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 2, February 2015.
[13] Jin, D. Pan, J. Xu, and N. Pissinou, "Efficient VM placement with multiple deterministic and stochastic resources in data centers" in IEEE Global Communications Conference (GLOBECOM*)*, 2012, pp. 2505–2510.
[14] Li, Bo, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong. "Enacloud: An energy-saving application live placement approach for cloud computing environments." In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, pp. 17-24. IEEE, 2009**.**
[15] Kokilavani, T., Dr George Amalarethinam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing", International Journal of Computer Applications 20, no. 2 (2011): 43-49.
[16] Sumita Bose, Jitender Kumar, "An Energy Aware Cloud Load Balancing Technique using Dynamic Placement of Virtualized Resources", Advances in Computer Science and Information Technology (ACSIT) Volume 2, Number 7; April – June, 2015 pp 81 – 86.