International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 16
June 2018

An Efficient Service Allocation, Deallocation & Migration in Cloud Environment

Nimanshi & Ms Mamta

[1]Department of Computer Science
South Point Institute of Technology and Management (SITM)
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat
[1]nimanshisehgal07@gmail.com

[2]Department of Computer Science
South Point Institute of Technology and Management (SITM)
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat
[2] mamtakalra21@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. All the machines are homogenous. 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**

I. INTRODUCTION

Cloud computing [1] is a technology that combines the internet and central remote servers to provide & maintain data and applications resources. 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 is a comprehensive solution that delivers Information Technology as a service.

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 t 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 comprises of 2 components —the front end and the back 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.

II. 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].

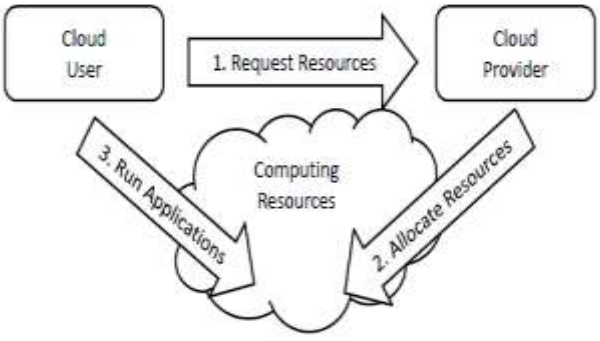

Figure 1: Interaction between cloud actors

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.

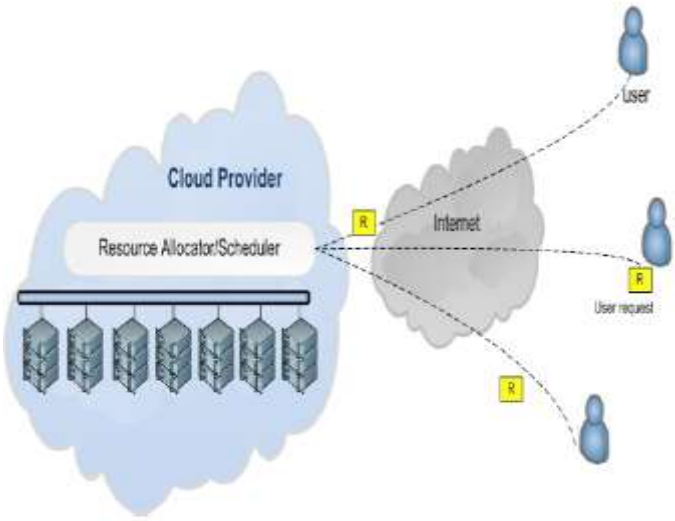

Figure 2: Resource Allocation & scheduling in Cloud Computing

A. Resource Allocation

Resource allocation [8, 9] 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.

B. Job Scheduling

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.

III. PROPOSED MODEL

There are an increasing number of Cloud Services available in the Internet. The basic purpose of all of the cloud services is to provide efficient, effective & varied types of services to their clients. In this present work we have defined a multiple cloud environment. Each cloud server is defined with certain limits in terms of memory and the CPU specifications. A middle layer is defined between the cloud servers and the client requests that will perform the allocation of the processes to different clouds in under load and over load conditions. Each user request contains some list of parameters. These parameters are the number of processor, process time, memory requirement, input output specifications etc. In the general case, the allocation of the processes is performed in a sequential order.

The middle layer will exist between the clouds and the clients. As the request will be performed by the user this request will be accepted by the middle layer and the analysis of the cloud servers is performed by this middle layer. The middle layer also called middleware performs following three activities:

• Scheduling the user requests

• Monitor the cloud servers for its capabilities and to perform the process allocation.

• 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.

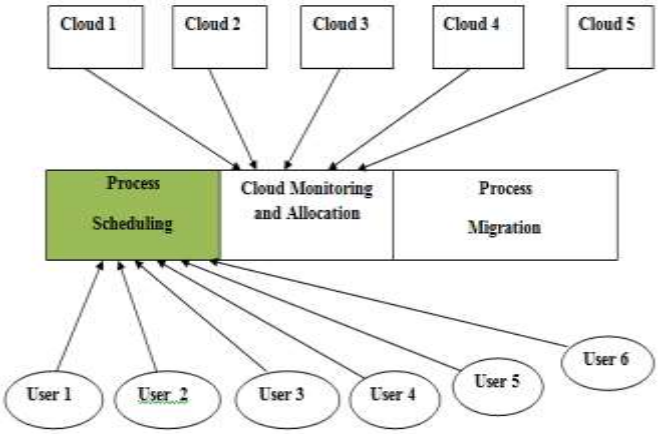

Fig 3: Process Scheduling, Allocation and Migration

IV. PROPOSED ALGORITHM

In this paper, we consider the problem 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 figures explain the algorithms for our proposed work on efficient service allocation, de-allocation and migration concepts. Figure 4 shows the initialization of servers and clients.

```
// Initialize all the available servers
1. For t ← 1 to MAX do
2. Serv[t].remainingCap ← 32
3. Serv[t].startedStatus ← false
3. Serv[t].util ← 0
4. Serv[t].serverId ← 100 + t
// end of for loop

// Initialize all the available clients
5. For t ← 1 to MAX do
6. Cl[t].userId ← t
7. Cl[t].allocatedStatus ← false
8. Cl[t].numCPU ← 0
9. Cl[t].serverId ← 0
// end of for loop

// starting 1st two servers
10. Start ← 2
11. For t ← 1 to start do
12. Serv[t].startedStatus ← true
13. Serv[t].util ← 0
// end of for loop
```

Figure 4: The initialization of servers and clients.

Figure 5 shows the algorithm for allocation process of our proposed work.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 16
June 2018

```
//allocation process
1. Satisfied Flag ←false;
2. For t ← 1 to start do
3. if(serv[t].ReaminingCap >= req and
                    serv[t].startedStatus = true)
4. Place the request on the server serv[t]
5. satisfiedFlag ← true
6. Update the utilization and remaining capacity
of server serv[t]
7. cl[clientId].servID ← serv[t].servID
8. cl[clientId].numCPU ← noOfCPU
9. cl[clientId].allocatedstatus ← true
10. break
 // end of for loop
11. If(satisfiedFlag = false)
12. For t ← 1 to MAX
13. if(serv[t].startedstatus = true)
14. continue;
15. else
16. serv[t].startedstatus ← true;
17. serv[t].util ← 0
18. start ← start + 1
19. allocation process
20. break
// end of if
// end of for
21.  if (t = MAX)
22. Print "Request cannot be satisfied because
data center is out of servers"
  // end of allocation
```

Figure 5: Algorithm for allocation process

Figure 6 shows the algorithm for de-allocation process of our proposed work.

```
// de-allocation process
 1. Accept clientId, noOfCPU
2. For t ← 1 to MAX do
3. if(clientId = cl[t].userID)
4. if(cl[t].allocatedstatus = true)
5. if(cl[t].numCPU >= ncpu)
6. update num of cpus
7. Identify the server id sid
8. if(cl[t].numCPU = 0)
9. cl[t].allocatedstatus ← false
10. cl[t].servID ← 0;
11.Update the utilization and remaining
capacity of server sid
12.  else
13. print "No. of cpu deallocated is more than
allocated"
14. else
15. Print "No cpu is allocated as yet"
16. break
// end of if
// end of for
```

Figure 6: Algorithm for de-allocation process

Figure 7 shows the algorithm for migration process (if required) of our proposed work.

```
// Migration module
1. For i  ← 1 to MAX do
2. if(serv[i].util < minThreshold and serv[i].util
> 0)
3. set server id sid1 for migration
4. Identify no of CPU & required utilization for
migration
5. break
// end of for
6. if(i=MAX)
7. Print "No server is underutilized"
8. exit
9. for I ← 1 to MAX do
10. if(serv[i].util >= 40 and 100-serv[i].util >=
rutil)
11. Set server id sid2 on which server id sid1
resources are to be transferred
12. break
13. if(i=MAX)
14. print "No server has the required
utilization"
15. exit
16. Update the remaining capacity of server id
sid1 & sid2
17. Update the utilization value of server id
sid1 & sid2
18. Print "Resource of server id sid1 is
migrating to server id sid2.
```

Figure 7: Algorithm for migration process

V. 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 8 below shows the comparative chart of existing techniques and proposed algorithm for efficient service allocation and deallocation on cloud computing.

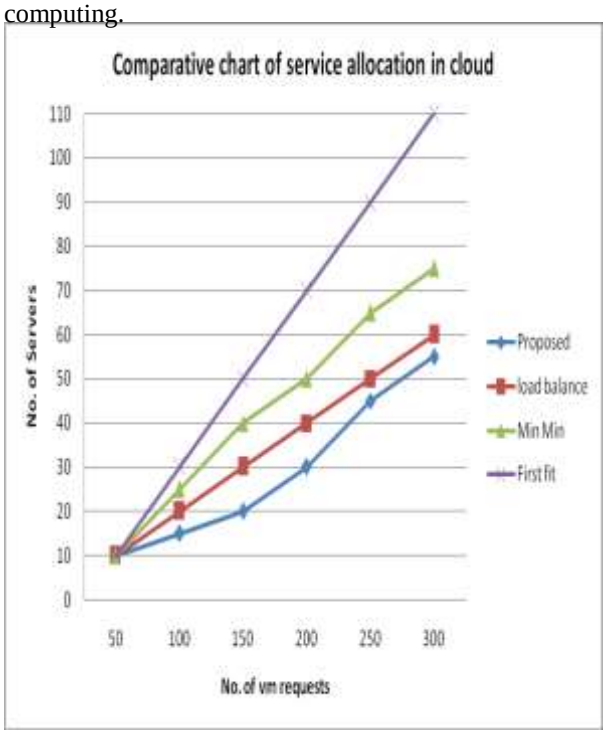

Figure 8: Simulation Results

V. CONCLUSION

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. 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.