2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015)
Dehradun, India, 4-5 September 2015

RAS: A novel approach for Dynamic Resource Allocation

Abhirup khanna* and Sarishma†

* CSE specialization in Cloud Computing and Virtualization,UPES, Dehradun, Uttarakhand abhirupkhanna@yahoo.com
†CSE specialization in Cloud Computing and Virtualization,UPES, Dehradun, Uttarakhand sarishmasingh@gmail.com

***Abstract*—With ever changing user requirements the ways of computing have changed. Presently, cloud can be seen as a computing model that can cope with these changes as it gave access to unlimited amount of resources. But with time this myth has changed, leading to situations even when the cloud also falls short of resources. It thus becomes evident that appropriate resource allocation needs to done in order to sustain effective functioning of the cloud environment. To overcome this problem the concept of Resource Allocation System (RAS) was introduced in order to provision and maintain resources in an optimized manner. In this paper, we present our own resource allocation system that works on the principles of dynamic resource allocation. We propose an algorithm that states the working of RAS and defines how resources are allocated dynamically. The algorithm is implemented using the CloudSim simulator. Finally, we conclude by depicting the working of our algorithm and display the simulated results.**

Keywords:resource allocation system; dynamic resource allocation; cloud computing; CloudSim.

I. Introduction

Since the inception of cloud computing significant rise can be seen in the area of distributed computing. The present day model of distributed or ubiquitous computing allows various heterogeneous devices to be connected with one another that leads to sharing of resources. Cloud computing can be considered an enabler for distributed computing along with a part of it. Cloud works on the principle of having resources hosted at remote geographical where in any end user can have access to them. Any amount of resource can be added to the data centre in order to enhance its resource pool. These resources are then utilized by the tasks and applications running at the cloud end. Cloud computing has offered new ways of resource utilization by optimizing resources held by idle systems thereby reducing the overall cost of task execution. One of the key features of cloud computing is pay per use, which could only become a reality after successful optimization of idle resources. Such idle resources when properly automated form a part of resource pool of CSP.

Resource pool of a cloud service provider is large but the real problem is with the organization, provisioning, scheduling and optimization of the resources present in it. With time, different strategies and scheduling algorithms have come into picture which can be used to schedule the resources, such as round robin algorithm, min-min and max-min algorithm, genetic algorithms, etc. By implementing these algorithms, cloud providers are able to develop environments where they can use these resources in the most optimal way possible. Resources can be categorized as real or virtual resources. The physical resources which are present in real world are termed as real resources whereas those resources which exist virtually that is in form of virtual machines are virtual resources. It is inconvenient to transform real resources every time the user asks for it but the converse is possible when talking about virtual resources. They can be spawned anytime catering to varying user requirements on a real time basis[7].

It is the work of the cloud service provider to ensure QoS thereby allocating both real and virtual resources to end users in an optimized manner. This problem intensifies even more when we are dealing with multiple data centers located throughout the globe. To overcome this problem, a new system was introduced known as Resource Allocation System (RAS). The functionalities of a RAS are very much similar to a human brain. It coordinates with every layer present in the cloud environment and allocates resources accordingly[14]. It provides an efficient way of allocating resources so as to meet the minimum requirements of every application running on cloud. RAS also facilitates the user by providing a way to track down the utilization of resources by its VM.

In this paper, we redefine the working of RAS and propose are very own resource allocation system build on the concepts of Dynamic Resource Allocation. The rest of the paper is categorized as follows: Section II elucidates the related work in the field of resource allocation for a cloud environment. We explain our proposed resource allocation system in Section III and its simulation in detail in Section IV. Finally, Section V exhibits the results after simulation.

II. RELATED WORK

Over time researchers have formulated various resource allocation strategies in order to allocate resources in the most efficient manner. Continuous work is still going on so as to generate more optimized and efficient systems where resources can be provisioned and scheduled in the best possible way. In this section, we have presented some of the research works pertaining to this area.

- **Genetic algorithm[1]:** For a given resource scheduling scenario with multiple constraints, genetic algorithm focuses on presenting a time bound representation scheme so as to generate the most efficient solution. A set of heuristics technique is applied to the problem such as mutation, cross-over technique etc. These heuristics work best when we deal with a large problem set for resource allocation. In case of smaller problem set, it may not make as efficient solution as it is supposed to make. For a given set of problems, it generates a set of possible solutions i.e. solutions which can be used, termed as population. Heuristics and genetic operators are applied to filter out the population iteratively until we find the best solution. There are conditions when iterations will stop such as end point reached, manually inspected solution is selected, result quality degrading over multiple iterations, computation-time bound reached or a combination of above reasons. Genetic algorithm is best suited for solving search and optimization problems. It is also used when branch and bound techniques fail to produce results.

- **Idle systems[2]:** In this present scenario, the infrastructure layer of the cloud needs to incorporate better techniques for resource allocation in order to cater to the requirements of every end user. One such solution to this problem can be the use of idle systems to become a part of the cloud environment. These idle systems can be machines from universities, homes and even enterprises. The compute and storage capabilities of such systems can be made available in case of over utilization of cloud resources. The idle systems can join or leave the cloud environment on an ad hoc basis. The architecture involving the use of idle systems has three major components namely; Cloud Computing platform, Harvesting middleware and Resource allocation Agent.

- **Workflow algorithm[3]:** When talking of workflow algorithm resources to be allocated and the problem pertaining to resource allocation are considered as a set which is known as workflow. Using a directed acyclic graph, this algorithm allocates resources to the task by mapping them to their corresponding resources. The graph is made in such a way that every node represents a task and the edges symbolize the dependency among various tasks. By evaluating this graph, the solution is obtained for a workflow. At large scale, workflows can be solved by using a workflow management system which can solve multiple workflows simultaneously. Research is still going on in order to make the algorithm more efficient as presently it doesn't consider the execution time of tasks which is a drawback.

- **Load Balance Min-Min[4]:** When we talk about cloud computing, resource allocation plays a major role when it comes to the execution of complex tasks in minimum possible time. There are tasks or jobs which demand higher number of resources for their executions as their computation level is very high. For such scenarios, Opportunistic Load Balancing (OLB) is used which aims at distributing tasks over resources so as to make every node busy. There can be cases where multiple tasks are allotted to some nodes which are already very busy. In such cases, Load Balance Min-Min algorithm is used which aims to balance the load of unbalanced nodes. The minimum execution time of each task to allocated to the cloud environment. These two algorithms when combined together can decrease overall execution time of a task and can also achieve load balancing in a multiple level cloud environment.

- **Agent-based adaptive resource allocation[5]:** Each request coming from an end user task is mapped to an appropriate data center which is chosen from a set of available distributed data centers. This scheme resulted in fast allocation of resources and decreased execution response time thereby fulfilling the requirements of the end user. Infrastructure-as-a-Service model was used by the cloud service provider where consumer requests were fulfilled by delivering the virtual machines to consumers from a data center. The motive of adaptive resource allocation scheme is to find an appropriate data center for the execution of consumer request. The two major factors which are considered in this are the geographical location the data center with respect to consumer and second is the present workload on the data center. Agent based test bed written in Java was implemented to get the results. This test bed was implemented using Java with Java Agent Development Framework (JADE).

III. PROPOSEDWORK

In this section we talk about our Resource Allocation System (RAS). The system which we propose is based upon the principles of dynamic resource allocation in which resources are allocated on the fly as per the requirements of the end user. The RAS model we propose has the following three functionalities:

- Discovery of resources: It is the work of the RAS to ensure that the required amount of resources are present in the common resource pool before allocating them to respective VMs.

- Monitoring of resources: It is the work of the RAS to monitor resource usage of every VM with respect to the applications running on it. It maintains a record of every VM running on a particular server. The RAS keeps track of every resource that is allocated in terms of its location, time of usage and its availability.

- Dynamic allocation: The aim behind creating this RAS is to foster dynamic allocation of resources. It is the role of the RAS to dynamically spawn resources from the common resource pool and provide them to the VMs. It is the RAS which takes care of the addition and subtraction of resources from the common resource pool.

Working of RAS: The working of RAS could be explained through the illustration of the algorithm that forms the core for it. The algorithm depicts the concepts of dynamic resource allocation stating how resources are allocated in a dynamic fashion. Following is the proposed algorithm. RV=> Resources allocated to a VM
RC=> Resources allocated to a Cloudlet
RP=> Resources present in the common resource pool.

RV=Σ RC
Tv=> Timer for a VM

```
if(RV < RC)
print("sending of VM")
RP=RP - RV
print("VM created")
print("cloudlets")
print("resources allocated")
print(RV)
print("resources left")
print(RP)
else(RV > RC)
print("out of resources")
if(VM is shutting down || Tv == 0)
RP=RP+RV
print("resources left")
print(RP)
else(Cloudlet is shutting down)
RP=RP +RC
print("resources left")
print(RP)
if(RV1 < RP && RV2 < RP && RV1<RV2)
RP=RP - RV1
print("resources allocated")
print(RV)
print("resources left")
print(RP)
else(RV1==RV2 && RV1 < RP && RV2 < RP)
print("wait for random time")
allocate resources to the VM having least waiting time
```

IV. IMPLIMENTATION &SIMULATION

The above mentioned algorithm is implemented on the CloudSim framework. In CloudSim there are various predefined classes that provide a simulation environment for cloud computing. It is a java bases simulation tool and can be implemented either with Eclipse or NetBeans IDE. For our proposed algorithm we have used the eclipse IDE. To run CloudSim in eclipse, we first need to download the eclipse IDE and install it. After successful installation of eclipse download the latest CloudSim package, extract it and import it in eclipse. Talking of our proposed work we have created our own classes in CloudSim and have portrayed our algorithm in form of java code. The following are the screenshots that depict the working of our algorithm on CloudSim framework.

The above screenshot exhibits the working of our resource allocation system. It shows how resources are allocated to respective VMs from a common resource pool. The screenshot depicts creation of a Host (Server) and two subsequent VMs on it. Each VM has a set of Cloudlets running on them. Each VM is up and running for a specific period of time. Every time a VM requests for a resource the allocated resources get subtracted from the common resource pool. Talking of VM 0 it has three cloudlets running on it and resource being allocated to it are 5GB of storage space, 1GB of RAM and 500 bps of bandwidth. Similarly resources are added to the common resource pool when some Cloudlet or VM stops its execution. The same can be seen when Cloudlet 2 of VM 1 is stopped. Addition of resources to the common resource pool may also

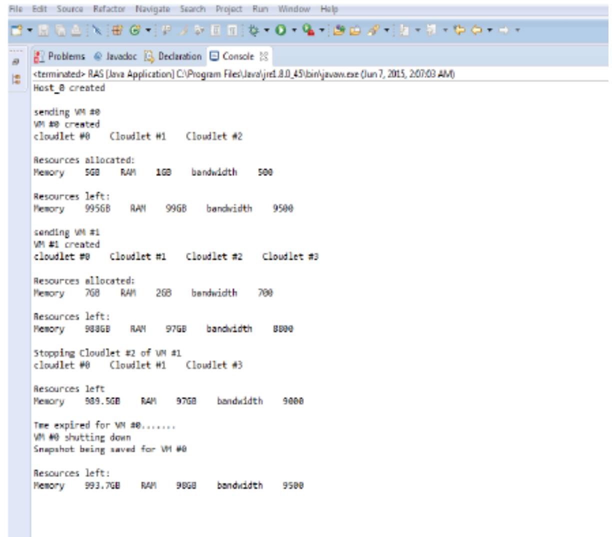

Fig. 1. screenshot of resource allocation system

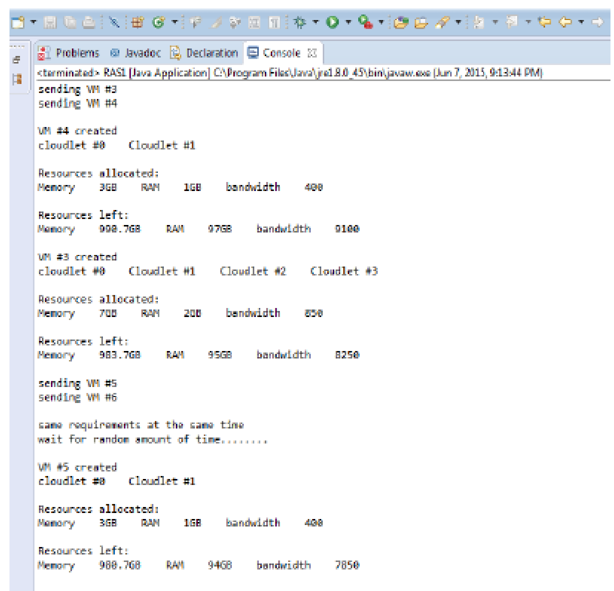

Fig. 2. conditions to resource allocation

take place when the time period of a VM expires. Certain amount of storage space is still utilized for storing the snapshot of that VM.

The above screenshot explains two conditions pertaining to resource allocation:

- Firstly, when two VMs having different resource requirements approach the RAS at the same time. , when two VMs having same resource requirements approach the RAS at the same time.

In the first case when requirements of the two VMs are different then the VM having minimum resource requirements among the two is preferred by the resource allocation system. The same can be seen when VM #3 is allocated resources prior to VM #4 despite the fact that they both had requested

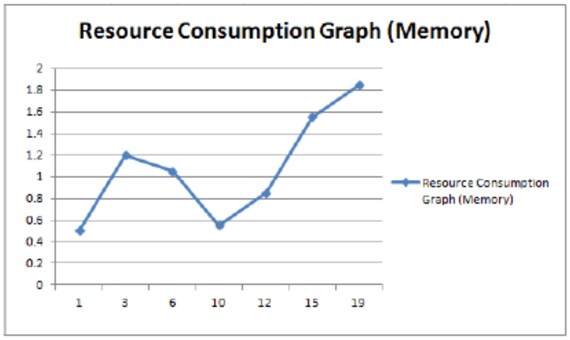

Fig. 3. resource consumption (memory)

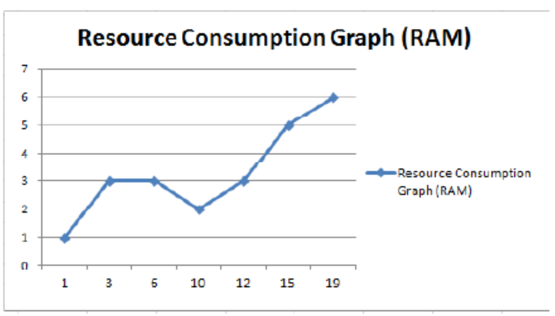

Fig. 4. resource consumption (RAM)

for resources at the same time. Talking of the second case when the requirements of both the VMs are same, the RAS would allot a random waiting time to both the VMs. Every VM will wait for its turn to ask for the resources according to its waiting time. The VM having minimum waiting time will be first allocated resources followed by the subsequent VM. The same can be seen when VM #5 is allocated resources prior to VM #6 as it possessed minimum waiting time out of the two.

V. RESULTS

After successful implementation of the algorithm on CloudSim the following screenshots depict the simulated results.

The above graph shows increase in resource consumption (memory) of a server with respect to time. The X axis represents time (minutes) whereas the Y axis represents the percentage of resources consumed. Resource consumption of a server starts at time t=1 min and keeps on changing till it reaches t=19 min. Every rise and fall depicted in the graph is due to starting or stopping of a Cloudlet or a VM. Rise in percentage of resource consumption can be seen when VM #0, VM #1, VM #2, VM #3 and VM #4 are created. Where as a drop in the percentage can be observed when Cloudlet #2 of VM #1 and VM # 0 are stopped.

The above graph shows increase in resource consumption (RAM) of a server with respect to time. The X axis represents time (minutes) whereas the Y axis represents the percentage of resources consumed. Resource consumption of a server starts at time t=1 min and keeps on changing till it reaches t=19 min. Every rise and fall depicted in the graph is due to starting or stopping of a Cloudlet or a VM. Rise in percentage of resource consumption can be seen when VM #0, VM #1, VM #2, VM #3 and VM #4 are created. Where as a drop in the percentage can be observed when VM # 0 is stopped.

VI. CONCLUSION

Cloud computing is a computing model that has changes the worlds outlook towards accessibility of resources. With cloud, users could have access to endless resources from anywhere and everywhere. But with increase in number of cloud users this concept started to fade resulting in lack of resources at the cloud end. The problem was addressed by the inception of a resource allocation system that allowed resources to be allocated in an optimized manner. In this paper, we lay emphasis on this concept and present our own Resource Allocation System. The RAS works on the principles of dynamic resource allocation. An algorithm is also presented that illustrates the working of RAS. We implement and simulate this algorithm on CloudSim which is a cloud framework. At the end we have the simulated results which are analyzed and well explained through graphical representations.

REFERENCES

[1] Ahmed, A., Alsammak, A. K. A Generic resources allocation approach for better Cloud Computing IaaS Services.

[2] Beloglazov, A., Abawajy, J., Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, 28(5), 755-768.

[3] Hoenisch, P., Schulte, S., Dustdar, S. (2013, December). Workflow scheduling and resource allocation for cloud-based execution of elastic processes. In Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on (pp. 1-8). IEEE.

[4] Wang, S. C., Yan, K. Q., Liao, W. P., Wang, S. S. (2010, July). Towards a load balancing in a three-level cloud computing network. In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on (Vol. 1, pp. 108-113). IEEE.

[5] Jung, G., Sim, K. M. (2011, September). Agent-based adaptive resource allocation on the cloud computing environment. In Parallel Processing Workshops (ICPPW), 2011 40th International Conference on (pp. 345-351). IEEE.

[6] Buyya, R., Ranjan, R., Calheiros, R. N. (2009, June). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In High Performance Computing Simulation, 2009. HPCS'09. International Conference on (pp. 1-11). IEEE.

[7] Dearle, A. Macdonald,A. Fernandes. An Approach to Ad hocCloud Computing. Graham Kirby, s.l.: University of St Andrews,2010.

[8] Greenberg, A., Hamilton, J., Maltz, D. A. and Patel, P. The Cost of a Cloud: Research Problems in Data Center Networks.s.l. :ACM SIGCOMM Computer Communication Review, 2009.10.1145/1496091.1496103.

[9] E.Williams. Energy Intensity of Computer Manufacturing: Hybrid Assessment Combining Process and Economic Input-Output Methods.s.l.: Environmental Science Technology, 2004. 6166-6174.

[10] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, M. Tsugawa. Chicago,IL. Science Clouds: Early Experiences in Cloud Computing for Scientific Applications.: Cloud Computing and Its Applications, 2008.

[11] D. Huang, X. Zhang, M. Kang, and J. Luo, Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication, in Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering, 2010.

[12] H. Raj, R. Nathuji, A. Singh, and P. England, Resource management for isolation enhanced cloud services, in Proceedings of ACM workshop on Cloud computing security, 2009, pp. 77V84.

[13] Lai, K., Rasmusson, L., Adar, E., Sorkin, S., Zhang, L., Huberman, B. A. (2004). Tycoon: An implementation of a distributed, market-based resource allocation system. arXiv preprint cs/0412038.

[14] Buyya, R., Yeo, C. S., Venugopal, S. (2008, September). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on (pp. 5-13). Ieee.

[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.