RESEARCH ARTICLE OPEN ACCESS

Improving Resource Allocation in Virtualized Cloud Environment

Nisha Sharma[1], Mamta Dhanda[2]
Department of Computer Science and Engineering,
JMIT, Radaur
Haryana-India

ABSTRACT
Cloud computing offers utility-oriented IT services to users worldwide. Based on a pay-as-you-go model, it enables hosting of pervasive applications from consumer, scientific, and business domains. However, data centers hosting Cloud applications consume huge amounts of electrical energy, contributing to high operational costs and carbon footprints to the environment. The basic principle of cloud computing is that user data is not stored locally but is stored in the data center of internet. To gain the maximum degree of the benefits, the services offered in terms of resources should be allocated optimally to the applications running in the cloud. The following section discusses the significance of resource allocation. This research work was focused on the design and implementation of an automated resource management system that achieves a good balance. Main objective of this work is implementation of a resource allocation policy that can avoid overload in the system effectively while minimizing the number of servers.

Keywords:- Cloud Computing, Resource allocation, Resource Optimization, Cloud Sim Toolkit.

I. INTRODUCTION

Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital out lays in hardware to deploy their service or the human expense to operate it. They need not be concerned about over provisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or under provisioning for one that becomes wildly popular, thus missing potential customers and revenue. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1000 servers for one hour costs no more than using one server for 1000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT.

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The data center hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal data centers of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. People can be users or providers of SaaS, or users or providers of Utility Computing. We focus on SaaS Providers (Cloud Users) and Cloud Providers, which have received less attention than SaaS Users.

Figure 1: Cloud Platform on the web

II. ESSENTIAL CHARACTERISTICS:

***On demand self-service*:-**A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access: Capabilities are available over the network and accessed through standard mechanisms that

promote use by heterogeneous thin or thick client platforms (e.g. mobile phones, tablets, laptops, and workstations).

Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. .

Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service: Cloud systems automatically control and optimize resource use by leveraging metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

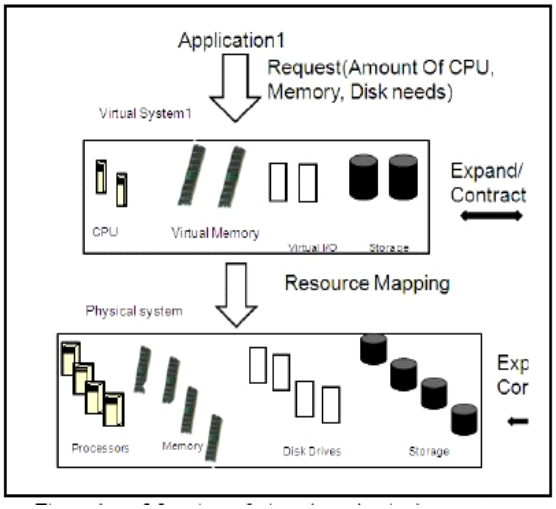

Fig 2: Mapping of physical to Virtual

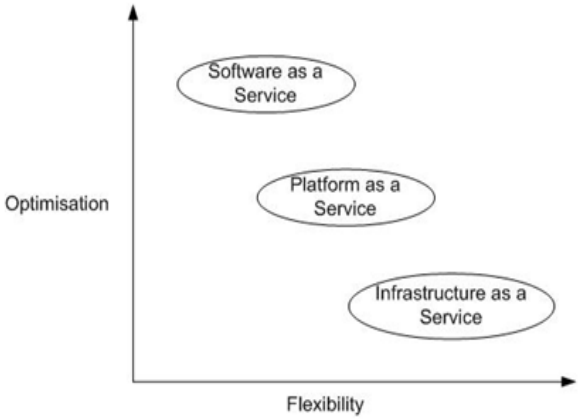

Fig 2 – Software, Platform and Infrastructure Services

Virtualization

In a virtuallized cloud computing environment, customers may never know exactly Where their data is stored. In fact, data may be stored across multiple data centers in an effort to improve reliability, increase performance and provide redundancies. This geographic dispersion may make it more difficult to ascertain legal jurisdiction if disputes arise

Virtual Machine.

As discussed earlier, a host can simultaneously instantiate multiple VMs and allocate cores based on predefined processor sharing policies (space-shared, time-shared). Every VM component has access to a component that stores the characteristics related to a VM, such as memory, processor, storage, and the VM's internal scheduling policy, which is extended from the abstract component called VM Scheduling.

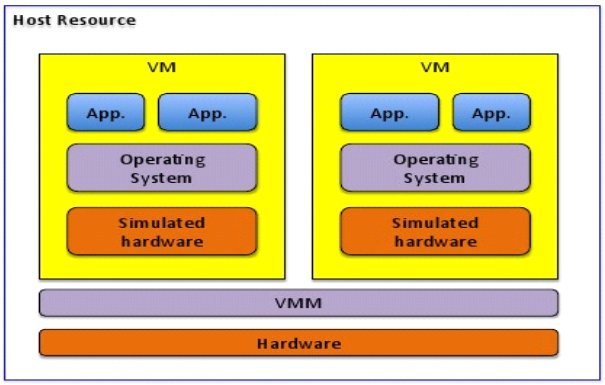

Figure3: Virtual Machine Architecture

A system which can automatically scale its infrastructure resources is designed in. The system composed of a virtual network of virtual machines capable of live migration across multi- domain physical infrastructure. By using dynamic availability of infrastructure resources and dynamic application demand, a virtual computation environment is

able to automatically relocate itself across the infrastructure and scale its resources.

III. PRESENT WORK

Cloud computing offers utility oriented IT services to users globally. It is Based on a pay as you go model, it enable hosting of pervasive applications from customer, scientific, and business domains. The data centers hosting Cloud applications consume large amounts of electrical energy, contributing to high operational costs and carbon footprints to the environment. The basic principle of cloud computing is that user data is not stored locally but is stored in the data center of internet.

Our main object is to develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. We have successfully designed a Resource algorithm that can capture the resource usages of applications accurately without looking inside the Virtual Machines . This work is focus on the design and implementation of an automated resource management system that achieves a good balance between the two goals. Two goals are overload avoidance and reduction of Physical Machines used.

Overload avoidance: The capacity of a PM should be sufficient to satisfy the resource needs of all Virtual Machines running on it. Otherwise, the Physical Machines is overloaded and can lead to degraded performance of its Virtual Machines.
Reduction of Physical Machine:Our main object is to develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. We have successfully designed a Resource algorithm that can capture the resource usages of applications accurately without looking inside the Virtual Machines.
Overload avoidance: The capacity of a PM should be sufficient to satisfy the resource needs of all Virtual Machines running on it. Otherwise, the Physical Machines is overloaded and can lead to degraded performance of its Virtual Machines.
Reduction of Physical Machine: The number of Physical Machines used should be minimized as long as they can still satisfy the needs of all Virtual Machines . Idle Physical Machines can be turned off to save energy.
Number of Physical Machines used should be minimized as long as they can still satisfy the needs of all Virtual Machines . Idle Physical Machines can be turned off to save energy.
After measuring the uneven utilization of resources server over cloud system in the process and implementing the developed algorithm we have improved the overall utilization of servers in the face of multidimensional resource constraints.

Future research may include the extension of our adaptive resource allocation approach to QoS features, such as timeliness, accuracy and security. We can also extend the work in providing Allocation policies for Elastic Clouds such as Amazon EC2, Rackspace, etc.

Resource Allocation Table:

The following table depicts the Resource allocation of queue based resource allocation to manage the cloud systems for successfully terminated machines.

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Finish Time
9	SUCCESS	3	16	120	120.2
21	SUCCESS	3	16	120	120.2
33	SUCCESS	3	16	120	120.2
1	SUCCESS	2	2	160	160.2
13	SUCCESS	2	2	160	160.2

Fig: Result Table:

Interpretation of Results:

Finish Time

Fig below depicts the finish time required for set of virtual machines to execute various task with cloudlet size increasing., as cloudlet size increases more and more VMs get allocated for fixed set of resources and finish time also increased, however the proposed algorithm works even better for given a set of cloudlets and VMs over a set of data centers.

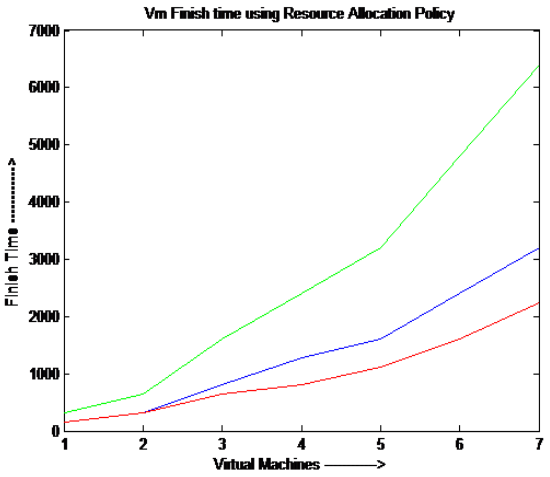

Fig : Finish time of against other algorithms

VM Resource Requirements

Cloudsim offers several ways to adjust a virtual machine's system resources. It can adjust how much memory is allocated to a virtual machine, and you can adjust settings that control how physical CPU resources are allocated to virtual machines. These allocations depend upon the underlying Allocation policy.

Allocating memory

A virtual machine's memory allocation is part of a virtual machine's configuration. When resource allocation policy creates a virtual machine, a specified amount of memory for the virtual machine is used. The maximum amount of memory you can assign to a virtual machine is gigabytes (GB); for x86 machines. however, for XEN hypervisor x64 can utilize 2^64 bytes of memory, depending on the available physical memory.

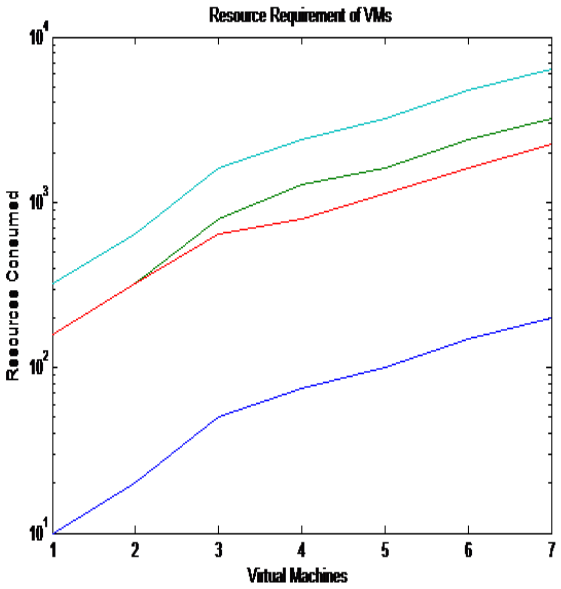

Fig 5.3 Resource Requirement of VMs for proposed algorithm (blue) against other algorithms

IV. PROPOSED WORK

An optimal RAS should avoid the following criteria as follows:

. Resource contention situation arises when two applications try to access the same resource at the same time.

. Scarcity of resources arises when there are limited resources.

.Resource fragmentation situation arises when the resources are isolated. There will be enough resources but not able to allocate to the needed application.

. Over-provisioning of resources arises when the application gets surplus resources than the demanded one

. Under-provisioning of resources occurs when the application is assigned with fewer numbers of resources than the demand.

V. CONCLUSION

This work was focused on the design and implementation of an automated resource management system that achieves a good balance between the two goals. Two goals are overload avoidance and reduction of Physical Machines used.

Our main object was to develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. We have successfully designed a Resource algorithm that can capture the resource usages of applications accurately without looking inside the Virtual Machines . The algorithm can capture the rising trend of resource usage patterns and help reduce the placement churn significantly.

REFERENCES

[1] N.Krishnaveni, G.Sivakumar,"Survey on Dynamic Resource Allocation Strategy in Cloud Computing environment", Dept. of CSE Erode Sengunthar Engineering College Thudupathi,India, International Journal of Computer Applications Technology and Research, Vol. 2, Issue 6, pp. 731 - 737,2013.

[2]Shabnam Khan,"A survey on scheduling based resource allocation in cloud computing", Computer Science and Engineering Dept., Sobhasaria Engineering College, Sikar,Rajasthan, India. International Journal For Technological Research In Engineering, Vol. 1, Issue. 1, Sep – 2013.

[3] VaghelaAnkita,"A survey on various resource allocation policies in cloud computing environment", Department of Computer Engineering, Alpha College of Engineering and Technology, Gujarat, India, Vol. 2, Issue 5, pp. 760 – 763.

[4] Anshul Rai, Ranjita Bhagwan, Saikat Guha,"GeneralizedResource Allocation for the Cloud", Microsoft Research India.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of Cloud computing environments for scaling of application services," in Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'10), ser. Lecture Notes in Computer Science, vol. 6081. Busan: Springer, May 2010, pp. 13–31.

[6] "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon. com/ec2.

[7] J. Varia, "Best practices in architecting Cloud applications in the AWS Cloud," in Cloud Computing: Principles and Paradigms, R. Buyya, J. Broberg, and A. Goscinski, Eds. Wiley Press, 2011, ch. 18, pp. 459–490.

[8] Q. Zhang, E. Gurses, R. Boutaba, and J. Xiao, "Dynamic resource alloca- ¨ tion for spot markets in Clouds," in Proceedings of the 2nd Workshop on Hot Topics in

Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE '11). Boston: USENIX, Mar. 2011.
[9] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or let's be friends," Computer Networks, vol. 53, no. 17, pp. 2905 – 2922, 2009, virtualized Data Centers.
[10] ´I. Goiri, F. Julia, J. Fit ` o, M. Mac ´ ´ias, and J. Guitart, "Resource-level QoS metric for CPU-based guarantees in Cloud providers," in Economics of Grids, Clouds, Systems, and Services, ser. Lecture Notes in Computer Science, J. Altmann and O. Rana, Eds. Springer Berlin / Heidelberg, 2010, vol. 6296, pp. 34–47