ALLOCATION OF RESOURCES DYNAMICALLY USING VIRTUAL MACHINES IN CLOUD AREA

DR.S.PREM KUMAR*

M.JYOTHI**

S.RAGHU RAM***

Abstract:

Cloud computing is a technology which is providing emerging resources to the customers. Cloud computing provides the resources to the end customers based on their needs. The emerging growth of the cloud computing in the market takes place with an advanced technology which is a virtualization technology. In this project I am going to use this virtualization technology to allocate the data center resources based in the usage of application. It also supports green computing by rearranging the number of servers which are in use. The "skewness" concept has been introduced in order to measure the unevenness in the multidimensional utilization of a server. We can improve overall utilization of the server resources by minimizing the skewness.

Index terms: Cloud computing, resource management, virtualization, green computing

* *Professor& HOD, Dept of CSE, G.PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY, KURNOOL, AP, INDIA.*

** *M. Tech Student, Dept of CSE, G.PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY, KURNOOL, AP, INDIA.*

*** *Assistance Professor, Dept of CSE, G.PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY, KURNOOL, AP, INDIA.*

Introduction:

The physical property and also the lack of direct capital investment offered by cloud computing is appealing to several businesses. There's lots of dialogue on the advantages and costs of the cloud model and on a way to move bequest applications onto the cloud platform. Here we have a tendency to study a different problem: however will a cloud service supplier best multiplex its virtual resources onto the physical hardware? This is vital as a result of abundant of the touted gains within the cloud model come back from such multiplexing. Studies have found that servers in several existing information centers are usually severely underutilized owing to over-provisioning for the height demand [1], [2]. The cloud model is anticipated to form such practice extra by giving automatic proportion and down in response to load variation. Besides reducing the hardware value, it conjointly saves on electricity that contributes to a major portion of the operational expenses in giant data centers.

Virtual machine monitors (VMMs) like Xen give a mechanism for mapping virtual machines (VMs) to physical resources [3]. This mapping is basically hidden from the cloud users. Users with the Amazon EC2 service [4], as an example, don't recognize wherever their VM instances run. it's up to the cloud supplier to form positive the underlying physical machines (PMs) have enough resources to meet their desires. VM live migration technology makes it potential to vary the mapping between VMs and PMs whereas applications are running [5], [6]. However, a policy issue remains as a way to decide the mapping adaptively in order that the resource demands of VMs are met while the quantity of PMs used is decreased . This is challenging once the resource wants of VMs square measure heterogeneous due to the varied set of applications they run and vary with time because the workloads grow and shrink. The capacity of PMs may be heterogenous as a result of multiple generations of hardware exist in an exceedingly knowledge center. We aim to realize two goals in our algorithm:

Overload avoidance:

The capability of a PM ought to be sufficient to satisfy the resource wants of all VMs running on that. Otherwise, the PM is full and can cause degraded performance of its VMs. Inexperienced computing. The amount of PMs used ought to be minimized as long as they will still satisfy the requirements of all VMs. Idle PMs are often turned off to save lots of energy.

There is associate inherent trade-off between the two goals within the face of fixing resource wants of VMs. For overload avoidance, we should always keep the employment of PMs low to reduce the chance of overload just in case the resource wants of VMs increase later.

For inexperienced computing, we should always keep the utilization of PMs fairly high to create economical use of their energy. In this paper, we tend to gift the planning and implementation of an automatic resource management system that achieves a good balance between the two goals. We tend to build the following contributions:

- We tend to develop a resource allocation system that may avoid overload within the system effectively whereas minimizing the amount of servers used.
- We tend to introduce the idea of "skewness" to live the uneven utilization of a server. By minimizing skewness,
- We are able to improve the utilization of servers within the face of flat resource constraints.
- We tend to style a load prediction formula that may capture the longer term resource usages of application accurately while not trying within the VMs.The algorithm will capture the rising trend of resource usage patterns and facilitate scale back the position churn significantly.

Important concepts for dynamic resource allocation:

Skewness algorithm:

We introduce the conception of "skewness" to live the unevenness within the multi-dimensional resource utilization of a server. By minimizing lopsidedness, we will mix differing types of workloads nicely and improve the utilization of server resources. Let n be the amount of resources we have a tendency to take into account and RI be the utilization of the i-th resource. We have a tendency to outline the resource lopsidedness of a server p as

$$skewness(p) = \sqrt{\sum_{i=1}^{n} (\frac{r_i}{\overline{r}} - 1)^2}$$

Our rule executes sporadically to judge the resource allocation standing supported the expected future resource demands of VMs. we have a tendency to outline a server as a hot spot if the use of any of its resources is on top of a hot threshold. we have a tendency to outline the temperature of a hot spot p because the sq. add of its resource utilization on the far side the hot threshold:

$$temperature(p) = \sum_{r \in R} (r - r_t)^2$$

Wherever R is that the set of overloaded resources in server p and r_t is that the hot threshold for resource r.

We have a tendency to outline a server as cold spot if the utilizations of all its resources square measure below a cold threshold. This indicates that the server is usually idle and a possible candidate to show off to avoid wasting energy. Finally, we have a tendency to outline the hot threshold to be. A level of resource utilization that's sufficiently high to justify having the server running but not therefore high on risk turning into a hot spot within the face of temporary fluctuation of application resource demands.

Hot and cold spots:

Our formula executes sporadically to judge the resource allocation standing supported the expected future resource demands of VMs. we tend to outline a server as a hot spot if the use of any of its resources is higher than a hot threshold. this means that the server is full and hence some VMs running thereon ought to be migrated away. We outline the temperature of a hot spot p because the sq. add of its resource utilization on the far side the recent threshold: where R is that the set of full resources in server p and rt is the hot threshold for resource r. (Note that solely overloaded resources square measure thought-about within the calculation.) The temperature of a hot spot reflects its degree of overload. If a server isn't a hot spot, its temperature is zero. We outline a server as a chilly spot if the utilizations of all itsresources square measure below a chilly threshold. this means that theserver is generally idle and a possible candidate to show off tosave energy. However, we tend to do thus only if the commonresource utilization of all actively used servers (i.e., APMs)in the system is below a inexperienced computing threshold. A serveris actively used if it's a minimum of one VM running. Otherwise,it is inactive. Finally, we tend to outline the nice and cozy threshold to be

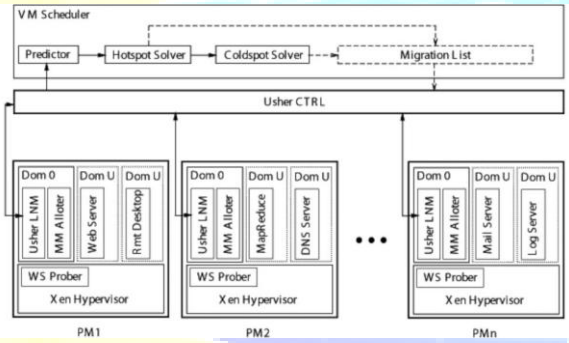
alevel of resource utilization that's sufficiently high to justifyhaving the server running however not thus high on riskbecoming a hot spot within the face of temporary fluctuation ofapplication resource demands.Different types of resources will have completely different thresholds.

For example, we will outline the recent thresholds for CPUand memory resources to be ninety and eighty percent, severally.Thus a server may be a hot spot if either its CPU usage is higher than90 percent or its memory usage is higher than eighty percent.

Green computing: When the resource utilization of active servers is just too low,some of them will be turned off to save lots of energy. This ishandled in our inexperienced computing formula. The challengehere is to scale back the quantity of active servers throughout lowload while not sacrificing performance either currently or within thefuture. We want to avoid oscillation within the system.Our inexperienced computing formula is invoked once theaverage utilizations of all resources on active servers square measurebelow the inexperienced computing threshold. we have a tendency to type the list ofcold spots within the system supported the ascending order oftheir memory size. Since we want to migrate away all itsVMs before we will close up Associate in Nursing underutilized server, wedefine the memory size of a chilly spot because the mixturememory size of all VMs running on that. Recall that our modelassumes all VMs connect with shared back-end storage.Hence, the value of a VM live migration is set largelyby its memory footprint. Section seven within the supplementary fileexplains why the memory may be a sensible live full. Wetry to eliminate the cold spot with very cheap price initial.For a chilly spot p, we have a tendency to check if we will migrate all its VMssomewhere else. for every VM on p, we have a tendency to try and realize adestination server to accommodate it. The resource utilizationsof the server when acceptive the VM should be below thewarm threshold. Whereas we will save energy by consolidatingunderutilized servers, overdoing it's going to produce hot spots inthe future. the nice and cozy threshold is intended to stop that.If multiple servers satisfy the on top of criterion, we have a tendency to like onethat is not a current cold spot. this is often as a result of increasingload on a chilly spot reduces the probability that it will beeliminated. However, we are going to settle for a chilly spot because thedestination server if necessary. All things being equal, weselect a destination server whose imbalance will be reducedthe most by acceptive this VM. If we will realize destinationservers for all VMs on a chilly spot, we have a tendency to record the sequenceof migrations and update the

anticipated load of connectedservers. Otherwise, we have a tendency to don't migrate any of its VMs. Thelist of cold spots is additionally updated as a result of a number of them couldno longer be cold attributable to the planned VM migrations within theabove method.The on top of consolidation adds further load onto the connectedservers. This is often not as serious a haul as within the hot spotmitigation case as a result of inexperienced computing is initiated solelywhen the load within the system is low. Withal, we wantto sure the additional load attributable to server consolidation. Werestrict the quantity of cold spots which will be eliminated ineach run of the formula to be no over a particularpercentage of active servers within the system. This is often referred to as theconsolidation limit.

System Architecture:



The architecture of the system is presented in Fig. 1. Each PM runs the Xen hypervisor (VMM) which supports a privileged domain 0 and one or more domain U. Each VM in domain U encapsulates one or more applications such as Web server, remote desktop, DNS, Mail, Map / Reduce, etc. We assume all PMs Sharebackend storage. The multiplexing of VMs to PMs is managed using the. Usher framework. The main logic of our system is implemented as a set of plug-ins to usher. Each node runs an Usher local node manager (LNM) on domain 0 which collects the usage statistics of resources for each VM on that node. The CPU and network usage can be calculated by monitoring the scheduling events in Xen. The memory usage within a VM, however, is not visible to the hypervisor. One approach is to infer memory shortage of a VM by observing its swap activities. Unfortunately, the guest OS is required to install a separate swap partition. Furthermore, it may be too late to adjust the memory allocation by the time swapping occurs. Instead we implemented a working set prober (WS Prober) on each hypervisor to

estimate the working set sizes of VMs running on it. We use the random page sampling technique as in the VMware ESX Server.

Memory resource management in VMware ESX server:

Prior to talking regarding however ESX manages memory for virtual machines, it's helpful to initial perceive however the applying, guest operating system, hypervisor, and virtual machine manage memory at their several layers.

1)An application starts and uses the interfaces provided by the operating system to explicitly allocate or de-allocate the virtualmemory throughout the execution.

2)In a non-virtual environment the operating system assumes it owns all physical memory in the system. The hardware does not provide interfaces for the software package to expressly "allocate" or "free" physical memory. The software package establishes the definitions of "allocated" or "free" physical memory.

Different operating systems have different implementations to realize this abstraction. One example is that the software package maintains associate degree "allocated" list and a "free" list, therefore whether or not or not a physical page is free depends on that list the page presently resides in.

Conclusion:

In this project the management of resources have been successfully designed, implemented and evaluated for cloud computing services. The multiplexing to virtual machines to physical machines takes place based on application demands.Skewness concept is used in order to combine the virtual machines with different resource characteristics so that the capacities of the servers are utilized. The implemented algorithm achieves both overload avoidance and green computing.

References:

[1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," technical report, Univ. of California, Berkeley, Feb. 2009.

[2] L. Siegele, "Let It Rise: A Special Report on Corporate IT," The Economist, vol. 389, pp. 3-16, Oct. 2008.

[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of
Virtualization," Proc. ACM Symp. Operating Systems Principles(SOSP '03), Oct. 2003.

[4] "Amazon elastic compute cloud (Amazon EC2),"http://aws.amazon.com/ec2/, 2012.

[5] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I.Pratt, and A. Warfield, "Live Migration of Virtual Machines,"Proc. Symp. Networked Systems Design and Implementation(NSDI '05), May 2005.

[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast TransparentMigration for Virtual Machines," Proc. USENIX Ann. Technical
Conf., 2005.

[7] M. McNett, D. Gupta, A. Vahdat, and G.M. Voelker, "Usher: AnExtensible Framework for Managing Clusters of Virtual Machines,"
Proc. Large Installation System Administration Conf.(LISA '07), Nov. 2007.

[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Boxand Gray-Box Strategies for Virtual Machine Migration," Proc.
Symp. Networked Systems Design and Implementation (NSDI '07),
Apr. 2007.

[9] C.A. Waldspurger, "Memory Resource Management in VMwareESX Server," Proc. Symp. Operating Systems Design and Implementation(OSDI '02), Aug. 2002.

[10] G. Chen, H. Wenbo, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao,"Energy-Aware Server Provisioning and Load Dispatching forConnection-Intensive Internet Services," Proc. USENIX Symp.Networked Systems Design and Implementation (NSDI '08), Apr. 2008.