OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

Virtualization Technology using Virtual Machines for Cloud Computing

T. Kamalakar Raju[1], A. Lavanya[2], Dr. M. Rajanikanth[2]

[1, 2] *Lecturer, Dept. of Computer Science, Andhra Loyola College, Vijayawada*

[3]*Lecturer, Dept. of Computer Science, Govt. Degree College, Movva*

Abstract: *Cloud computing is the delivery of computing and storage capacity as a service to a community of end users. The name "cloud computing" comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts services with a user's software, data and computation over a network. End users access cloud-based applications through a web browser or mobile application or a light-weight desktop while the business software and user's data are stored on servers at a remote location. Proponents claim that cloud computing environment allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT industry to more rapidly adjust resources to meet fluctuating and unpredictable business demand. In this paper, we present a system that uses virtualization technology to allocate the data center resources dynamically based on the application demands and support green computing by optimizing the number of servers in use. This method multiplexes virtual to physical resources adaptively based on the changing demand. We use the concept of skewness metric to combine virtual machines with different resource characteristics appropriately so that the capacities of servers are well utilized.*

Keywords: *Cloud, Hot spot, Physical machine, Skewness, Virtual machine.*

I. Introduction

Cloud computing emerges as a new computing technology which aims to provide customized, reliable and QoS (Quality of Service) guaranteed computing dynamic environments for end-users [1].Distributed processing, grid computing and parallel processing together emerged as cloud computing environment. The basic principle of cloud computing technology is that the user data is not stored locally but is stored in the data center of internet. The companies which provide cloud environment service could manage and maintain the operation of these data centers. The cloud users can access the stored data at any time by using the Application Programming Interface (API) provided by the cloud providers through any terminal equipment connected to the internet. Not only are the storage services provided but also both hardware and software services are available to the general public and business markets. The services provided by the service providers can be everything, from the infrastructure, platform or software resources. Each such cloud service(Figure 1) is respectively called as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS) [2].

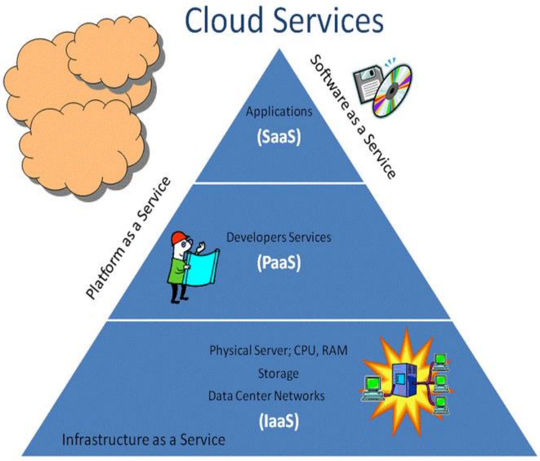

Fig. 1 Cloud Computing Services

There are numerous advantages of cloud computing technology, the most basic ones being the lower costs, re-provisioning of resources and remote accessibility. Cloud computing environment lowers cost by avoiding the capital expenditure by the company in renting the physical infrastructure from a third party provider. Due to the flexible nature of cloud computing technology, we can quickly access more resources from the cloud providers when we need to expand our business. The remote accessibility enables the cloud users to access the cloud services from anywhere at any time. To gain the maximum degree of the above mentioned benefits, the cloud services offered in terms of resources should be allocated optimally to the applications running in the cloud environment.

II. Related work

In [3], the authors proposed architecture, using the feedback control theory, for adaptive management of virtualized resources, which is based on Virtual Machine (VM). In this VM-based architecture all the hardware resources are pooled into common shared space in the cloud computing infrastructure so that hosted application can access the required resources as per there need to meet Service Level Objective (SLOs) of application. The adaptive manager use in this cloud architecture is multi-input multi-output (MIMO) resource manager, which includes three basic controllers: CPU controller, memory controller and I/O controller, its goal is regulate the multiple virtualized resources utilization to achieve SLOs of application by using the control inputs per-VM CPU, memory and I/O allocation.

In [4], the authors proposed a general two-layer architecture that uses the utility functions, adopted in the context of dynamic and autonomous resource allocation, which consists of the local agents and global arbiter. The responsibility of the local agents is to calculate utilities, for given current or forecasted workload and the range of resources, for each AE and results are transfer to global arbiter. Where, global arbiter computes near optimal configuration of the resources based on the results provided by the local agents. In [5], the authors proposed an adaptive resource allocation method for the cloud environment with preempt able tasks in which algorithms adjust the resource allocation adaptively based on the updated of the actual task executions. Adaptive list scheduling (ALS) and adaptive min-min scheduling (AMMS) algorithms are use for task scheduling process which includes static task scheduling, for static resource allocation, is generated offline. The online adaptive procedure is use for re-evaluating the remaining static resource allocation repeatedly with some predefined frequency.

The dynamic resource allocation based on the distributed multiple criteria decisions in computing cloud explain in [6]. In it, author contribution is two-fold, the first distributed architecture is adopted, in which the resource management is divided into independent tasks, each of which is performed by Autonomous Node Agents (NA) in ac cycle of three activities: (1) VMPlacement, in it suitable physical machine (PM) is found which is capable of running the given virtual machine and then assigned VM to that physical machine, (2) Monitoring, in it total resources use by hosted VM are monitored by NA, (3) In VM selection, if the local accommodation is not possible, a VM need to migrate at another PM and then process loops back to into placement. And second, using PROMETHEE method, NA carry out configuration in parallel by using multiple criteria decision analysis. This approach is potentially more feasible in large data centers than in the centralized approaches.

III. Proposed work

In this paper we develop a resource allocation method that can avoid overload in the cloud system effectively while minimizing the number of servers used. We introduce the concept of "skewness", which is used to measure the uneven utilization of a server. By minimizing the skewness, we can improve the overall utilization of the servers in the face of multi-dimensional resource constraints. We develop an effective load balancing algorithm using the Virtual Machine Monitoring to minimize or maximize different performance parameters.

A. System Overview

The architecture of the overall system is presented in Figure 2. Each physical machine (PM) runs the Xen hypervisor (VMM) which supports a privileged domain zero and one or more domain "U". Each VM in domain U encapsulates one or more applications such as the Web server, remote desktop, DNS, Map/Reduce, Mail, etc. We assume all PMs share a back- end storage. The multiplexing of the VMs to PMs is managed using the Usher framework [7]. The main logic of our cloud system is implemented as a set of plug-ins to Usher. Each node runs an Usher local node manager (LNM) on domain zero which collects the usage statistics of the resources for each VM on that node. The statistics collected at each PM are forwarded to the Usher central controller (Usher CTRL) where our virtual machine scheduler runs. The VM Scheduler is invoked periodically and then receives from the LNM the resource demand history of VMs, the capacity and the load history of the

PMs, and the current layout of VMs on PMs. The scheduler has several components, these include: The predictor predicts the future resource demands of VMs and the future load of PMs based on the past statistics. We compute the load of a physical machine by aggregating the resource usage of its VMs.

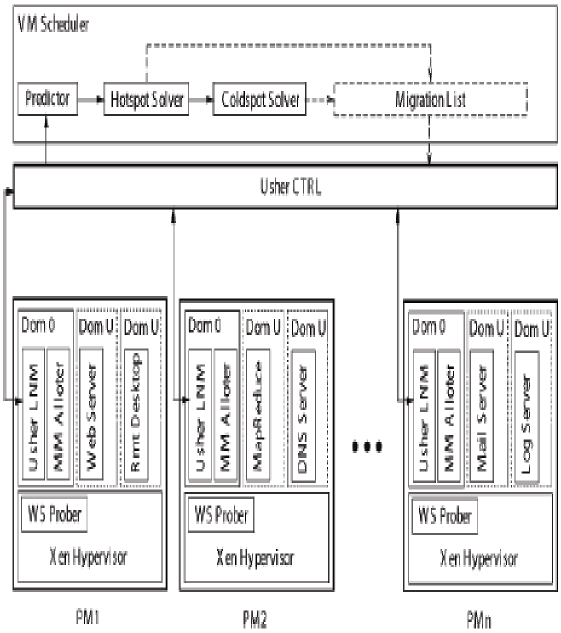

Fig. 2 System Architecture

The LNM at each node first attempts to satisfy the new demands locally by adjusting the resource allocation of virtual machines sharing the same VMM. The MM Alloter on domain zero of each node is responsible for adjusting the local memory allocation. The hot spot solver in our VM Scheduler detects if the resource utilization of any physical machine is above the hot threshold (i.e., a hot spot). The cold spot solver checks if the average utilization of an actively used PMs (APMs) is below some green computing threshold.

B. *Skewness Algorithm*

The skewness algorithm consists of three steps: hot spot mitigation, green computing, load balancing. Let “n” be the number of resources and “ri” be the utilization of the i-th resource. The resource skewness of a server “p” is defined as follows:

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

We use several adjustable thresholds that control tradeoff between performance and the green computing. The “hot threshold” defines the acceptable upper limit of the resource utilization. We define a server as a hot spot if the utilization of any of its cloud resources is above some hot threshold. We define the temperature of a hot spot “p” as the square sum of its resource utilization beyond the hot threshold:

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

Where R is the set of the overloaded resources in server p and rt is the hot threshold for resource r.

The temperature of a hot spot reflects its degree of system overload. If a server is not a hot spot, then its temperature is zero. The “cold threshold” denotes the acceptable lower limit of resource utilization. A server whose utilization of all system resources is under the cold threshold is defined as a cold spot. The “green computing” threshold defines the utilization level of all active physical machines, under which the system is considered power-inefficient therefore green computing operations get involved. Finally, the “warm threshold” defines the ideal level of the resource utilization that is sufficiently high to justify having the server running but not so high as to risk becoming a hot spot in the face of temporary fluctuation of the application resource demands.

C. Hot Spot Mitigation

For each scheduling round, the skewness algorithm takes two steps, hot spot mitigation and green computing, to calculate the migration list. In hot spot mitigation, we try to solve all hot spots in the descending order of the temperature. For each hot spot, we try to migrate away the virtual machine that can reduce the server's temperature the most. In those servers that can accommodate the virtual machine without becoming a hot spot, we choose a server with most skewness reduction by accepting this virtual machine as the migration destination. This does not necessarily eliminate the hot spot, but at least reduces the temperature. Hot spot mitigation step is finished after all hot spot are processed successfully. If the overall resource utilization of the active servers is lower than the green computing threshold, a green computing step is invoked.

D. Green Computing

In the green computing step, we try to solve cold spots in ascending order of the memory utilization, which representing the efforts taken to solve the cold spot. To resolve a cold spot, all of its virtual machines need to be migrated away. The destination of a virtual machine is decided in a way similar to that in the hot spot mitigation, but its resource utilization should be below the warm threshold after accepting the virtual machine. We also restrict the number of cold spots that can be eliminated in each run of the skewness algorithm to be no more than a certain percentage, for example 6%, of the active servers in the system. These arrangements are to avoid over consolidation that may incur hot spots later.

E. Load Balancing

The Load balancing algorithm (Figure 3) is divided into three parts: The first part is the initialization phase. In initialization phase, the expected response time of each virtual machine is to be found. In the second part, efficient virtual machine is found and in the last part, the ID of efficient virtual machine is returned.

Load Balancing Algorithm:

Step 1: For each virtual machine, find expected response time. The expected response time is found with the help of resource information program.

Step 2: When a request to allocate a new virtual machine from the Data Center Controller arrives, now find the most efficient VM (efficient VM having least loaded, minimum expected response time) for allocation.

Step 3: Return the identifier of the efficient virtual machine to the Datacenter Controller.

Step 4: Datacenter Controller identifies and notifies the new allocation

Step 5: Now update the allocation table increasing the allocations count for that virtual machine.

Step 6: When the virtual machine finishes processing the request, and then the Data Center Controller receives the Response. Data center controller notifies the efficient way for the VM de-allocation.

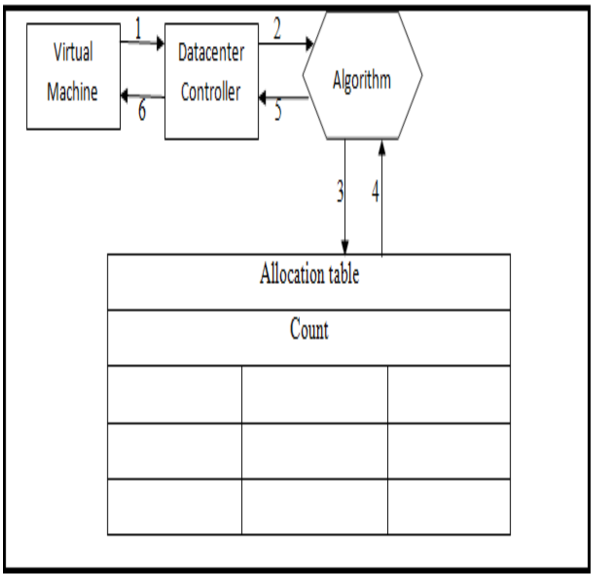

Fig. 3 Load Balancing

IV. Conclusion

Cloud computing technology emerges as a new computing paradigm which aims to provide customized, reliable and QoS (Quality of Service) guaranteed computing dynamic environments for the end users. In this paper, we develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. The capacity of a physical machine should be sufficient to satisfy the resource needs of all virtual machines running on it. Otherwise, the physical machine is overloaded and can lead to degraded performance of its virtual machines. We introduce the concept of “skewness” to measure the uneven utilization of the server. By minimizing the skewness, we can improve the overall utilization of the servers in the face of multi-dimensional resource constraints. The concept of the green computing is 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.

REFERENCES

[1] Lizhewang,JieTao,Kunze M.,Castellanos,A.C,Kramer,D.,Karl,w,”High Performance Computing and Communications”,IEEE International Conference HPCC,2008,pp.825-830.

[2] ZhixiongChen,JongP.Yoon,”International Conference on P2P, Parallel,Grid,Cloud and Internet Computing”,2010 IEEE:pp 250-257.

[3] “Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control,” in First International Conference on Information Science and Engineering, April 2010, pp. 99-102.

[4] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das, “Utility Functions in Autonomic Systems,” in ICAC ’04: Proceedings of the First International Conference on Autonomic Computing. IEEE Computer Society, pp. 70–77, 2004.

[5] Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen, Zhong Ming, “Adaptive Resource Allocation for Preempt able Jobs in Cloud Systems,” in 10th International Conference on Intelligent System Design and Application, Jan. 2011, pp. 31-36.

[6] Yazir Y.O., Matthews C., Farahbod R., Neville S., Guitouni A., Ganti S., Coady Y., “Dynamic resource allocation based on distributed multiple criteria decisions in computing cloud,” in 3rd International Conference on Cloud Computing, Aug. 2010, pp. 91-98.

[7] M. McNett, D. Gupta, A. Vahdat, and G. M. Voelker, “Usher: An extensible framework for managing clusters of virtual machines,” in Proc. of the Large Installation System Administration Conference (LISA’07), Nov. 2007.