Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.999 – 1005

RESEARCH ARTICLE

Avoiding Overload by Using Virtual Machine in Cloud Data Centre

[1]Indumathi.S
PG Student
[2]Mr. P. Ranjithkumar M.E
Assistant Professor
[1,2]Department of Computer Science and Engineering
Sri Subramanya College Of Engineering and Technology, Palani

Abstract— Cloud data centres improve CPU utilization of their servers (physical machines or PMs) through Virtualization (virtual machines or VMs). Over virtualized and under virtualized PMs suffer performance degradation and power dissipation respectively. In this paper, system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. Thus minimizes the hardware cost and saves the energy and used when a system under a heavy load thus improves the system performance. The concept of "skewness" is introduced to measure the unevenness in the multidimensional resource utilization of a server. By minimizing skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources. Finally minimizes the overload in a server by mitigate the process from high load (i.e. hot spot) server into normal load server (i.e. warm state) with the help of mitigation list. Physical Machine does not allow using a server in case of overload. Thus avoid the failure of server under high load.

Keywords: Cloud Computing, Resource Management, Virtualization, Green Computing

I. INTRODUCTION

Virtualization is a key technology underlying cloud computing platforms [5, 12], where applications encapsulated within virtual machines are dynamically mapped onto a pool of physical servers. Virtual machines provide several benefits in a cloud computing environment, including increased physical resource utilization via resource multiplexing, as well as flexibility and easy scale up/ scale-down through migration and fast restarts. In modern virtualization based compute clouds, applications share the underlying hardware by running in isolated Virtual Machines (VMs). Virtual machine monitors (VMMs) provide a mechanism for mapping virtual machines (VMs) to physical resources [3]. This mapping is largely hidden from the cloud users. Besides reducing the hardware cost, it also saves on electricity which contributes to a significant portion of the operational expenses in large data centers. VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running [5], [6].

However, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. This is challenging when the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink. The capacity of PMs can also be heterogeneous because multiple generations of hardware co-exist in a data center. For overload avoidance, the utilization of PMs should keep

too low to reduce the possibility of overload in case the resource needs of VMs increase later. For green computing, the utilization of PMs should keep reasonably high to make efficient use of their energy.

II. RELATED WORK

Automatic scaling of Web applications was previously studied in [10] [11] for data center environments. In MUSE [12], each server has replicas of all web applications running in the system. The dispatch algorithm in a frontend L7-switch makes sure requests are reasonably served while minimizing the number of under-utilized servers. Work [12] uses network flow algorithms to allocate the load of an application among its running instances. For connection oriented Internet services like Windows Live Messenger, work [10] presents an integrated approach for load dispatching and server provisioning. All works above do not use virtual machines and require the applications be structured in a multi-tier architecture with load balancing provided through an front-end dispatcher. In contrast, our work targets Amazon EC2-style environment where it places no restriction on what and how applications are constructed inside the VMs. A VM is treated like a black box. Resource management is done only at the granularity of whole VMs.

VM live migration is a widely used technique for dynamic Resource allocation in a virtualized environment [8] [10] [12]. The Proposed work also belongs to this category. Sandpiper combines multi-dimensional load information into a single Volume metric [8]. It sorts the list of PMs based on their volumes and the VMs in each PM in their volume-to-size ratio (VSR).

The HARMONY system applies virtualization technology across multiple resource layers [20]. It uses VM and data migration to mitigate hot spots not just on the servers, but also on network devices and the storage nodes as well. It introduces the Extended Vector Product *(EVP)* as an indicator of imbalance in resource utilization.

Dynamic placement of virtual servers to minimize SLA violations is studied in [9]. They model it as a bin packing Problem and use the well-known first-fit approximation algorithm to calculate the VM to PM layout periodically. That Algorithm, however, is designed mostly for off-line use. It is Likely to incur a large number of migrations when applied in on-line environment where the resource needs of VMs change dynamically.

Many efforts have been made to curtail energy consumption in data centers. Hardware based approaches include novel thermal design for lower cooling power, or adopting power-proportional and low-power hardware. Work [10] uses Dynamic Voltage and Frequency Scaling (DVFS) to adjust CPU power according to its load. We do not use DVFS For green computing.

III. SYSTEM ARCHITECTURE

A. Existing System

Previously Virtual machine monitors (VMMs) provide a mechanism for mapping virtual machines (VMs) to physical resources. This mapping is largely hidden from the cloud users. Users do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running. The capacity of PMs can also be heterogeneous because multiple generations of hardware coexist in a data center.

In this when a user send a request to the server the VM start searching for a results in PM which is responsible for allocate a resource based on the current application demand. This mapping is hidden from the users; it should not support the simillar applications, so it will affect the system with various application demands. This also uses a more number of servers but the utilization of a server is not efficient that causes the performance and thus increase the hardware cost.Fig.1 shows architecture of an existing system.

The Major drawbacks of this system are

- How to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized.
- When the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink.

Some other drawbacks are such as

- Waste of electricity
- Task overloaded
- Increased hardware cost

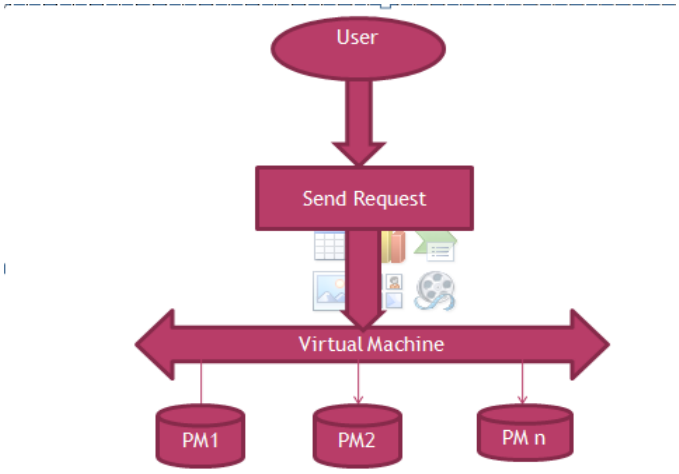

Fig.1 System Architecture of Existing System

In our scheme, shown in fig.2 server states can be further classified into three categories based on their load in a server by assigning threshold value to sever based on the number of process running in a server: cold, hot, and warm. Initially a process in a cold state, if the sever consumes process more than its hot threshold value, it indicates server in a hot state no other process could not enter into a server thus avoids overload in an server. If the server consumes average threshold value then it indicates sever in a warm state it will able to process some more requests. If the sever consumes cold threshold then the server is an cold state, the server process the fewer requests, therefore the fewer requests are migrated to an hot state or warm state servers in a manner that will not cause an overload in an sever, then turn off the server in an cold state to save the energy for future use. In some other cases it may be in an idle state that is no other process in server, then it will turned off to support the concept of green computing.

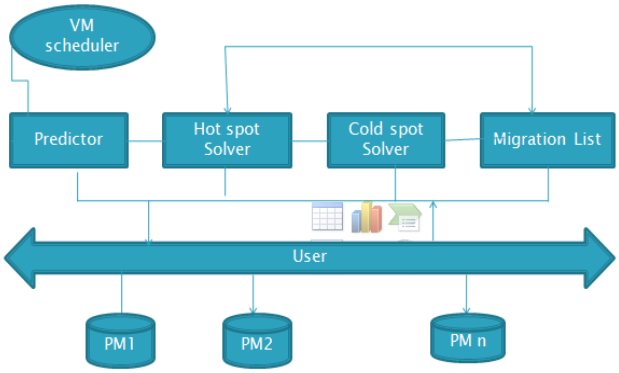

Fig.2. Architecture of Proposed System

B. Proposed System

In Proposed System the design and implementation of automated resource management system is presented to achieve a good balance between two goals such as

- **Overload Avoidance**
  The capacity of a PM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs.
- **Green Computing**

  The number of PMs used should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy.

IV. VM SCHEDULER AND PREDICTOR

A. VM Scheduler

VM Scheduler run and invoked periodically receives from the user, the resource demand history of VMs, the capacity and the load history of PMs (personal machine), and the current layout of VMs on PMs. It schedules the number of thread process running in a system, based on this the state of the process varies like hot state, cold state, warm state. Then it can forward the request to predictor, which predicts the state of process based on the threshold value. When multiple systems are connected together it schedules the process that is running on different systems. This will help the user to view which process running in a particular system.

B. Predictor

The predictor predicts the future resource demands of VMs and the future load of PMs based on past statistics. The load of a PM can be computed by aggregating the resource usage of its VMs. The local node manager at each node first attempts to satisfy the new demands locally by adjusting the resource allocation of VMs sharing the same VMM. Physical Machine can change the CPU allocation among the VMs by adjusting their weights in its CPU scheduler. Instead on make an prediction based on the external behavior of VMs, calculate the Exponentially weighted moving average (EWMA)

$$E(t) = \alpha * E(t-1) + (1-\alpha) * O(t),\ 0 \leq \alpha \leq 1$$

Where E (t) and O (t) are the estimated and the observed load at time t, respectively. α reflects a tradeoff between stability and responsiveness. To predict the future load on server measure the load every minute and predict the load in next time, Predictive value is modeled as a linear function of its past observations.

V. HOT AND COLD SPOT SOLVER AND MIGRATION LIST

A. Hot and Cold Spot Solver

The hot spot solver in our VM Scheduler detects if the resource utilization of any PM is above the hot threshold (i.e., a hot spot). If so, some VMs running on them will be migrated away to reduce their load. Then it can give the request to cold spot solver. The Physical Memory does not allow using the server, thus avoiding overload in a server. A server as a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away. The temperature of hot spots reflects its degree of overload in a server.

The cold spot solver checks if the average utilization of actively used PMs (APMs) is below the green computing threshold. If so, some of those PMs could potentially be turned off to save energy. It identifies the set of PMs whose utilization is below the cold threshold (i.e., cold spots) and then attempts to migrate away all their VMs then it forward request to migration list. A server is actively used if it has at least one VM running. Otherwise, it is inactive.

B. Migration List

When migration list can receive the request from cold spot solver and it can compiles list of VMs and migration list can passes it response to the user control for execution. If the minimum number of process is allocated to a particular server, then that process is migrated to a hot state server in a manner that will not cause an overload in an server. Also mitigate if a server allocated to more number of process beyond the hot state, then that process migrated to cold state. By mitigate the process from hot to cold and cold to hot the energy of server is saved and thus will reduce the hardware cost. The list of hot spots is sorted in descending temperature order.

VI. SKEWNESS ALGORITHM AND GREEN COMPUTING

A. Skewness Algorithm

This algorithm is mainly used to quantify the unevenness in the utilization of multiple resources on server

$$\text{Skewness}(R) = 1/n\ (X_o/x - 1) + (X_1/x - 1) + \ldots (X_n/x - 1)$$

Where n be a number of resources, and X be the average utilization of all resources for server R and X_0 be the utilization of 0^{th} server and so on. This can combine different kinds of workload and improve the overall utilization server resources.

B. Comparison with Vector Dot Algorithm

Vector Dot is the scheduling algorithm used in HARMONY [3] virtualization system, whose architecture is to this algorithm. It optimizes utilization for three types of resources including physical servers as well as data centre network bandwidth and I/O bandwidth of the centralized storage system. Instead of Virtual Machine migration it uses Virtual storage migration. The

skewness algorithm is similar to vector dot by minimizing the skewness. However Vector Dot emphasizes on load balancing. Skewness explicitly takes care of the structure of the residual resources to make them as useful as possible. Compared to Vector Dot, Skewness support load balance as well as green computing.

C. Green Computing

The resource utilization of active server is too low then the server is turned off to save the energy. The major problem in this to reduce the number of active server when the load of server is too low without any performance degradation either current or future.

This algorithm is invoked when the utilization of servers are below the green computing threshold. Then sort the list of cold spot based on ascending order of the memory size. Before going to turn off the underutilized server necessary to migrate away all of VMs. After accepting the VM must below the warm threshold, while saving the energy by consolidating underutilized server, overdoing it may be create hot spots in future, the warm threshold is designed to prevent them. The above consolidation adds extra workload on related server this is not a major problem because green computing is initiated only when the system load is too low. Eliminate the cold spots in the system only when the average load of all active servers (APMs) is below the green computing threshold. Otherwise, leave those cold spots there as potential destination machines for future offloading

D. Effects of Load on a Server

Load on a server varies, initially when the number of requesting resource process is in cold state then the load on server is like a fig.3 .Later the requesting resource process is in moderate load like warm state then load on server is like on fig. 4, When more number of requests on server that is it reaches above the hot threshold then the load on server is like in fig.5.

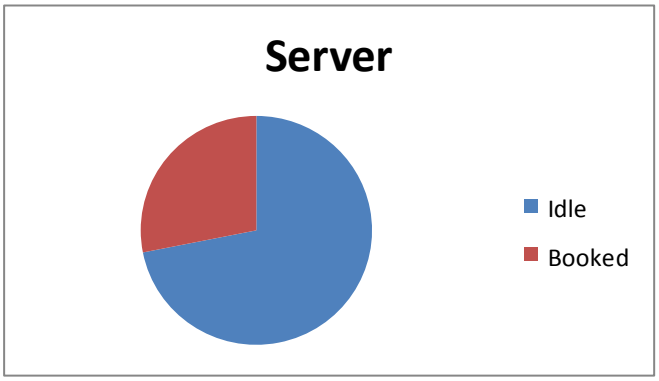

Fig.3. Initially load on server

VII. CONCLUSION

The design, implementation, and evaluation of a resource management system are presented for cloud computing services. Our system multiplexes virtual to physical resources adaptively based on the changing application demand. The skewness metric is used to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. This algorithm achieves both overload avoidance and green computing for systems with multi resource constraints. Thus resource allocation on Server under high load is avoided. Failure or crashes of server is limited. The overall utilization of resources is improved without performance degradation.

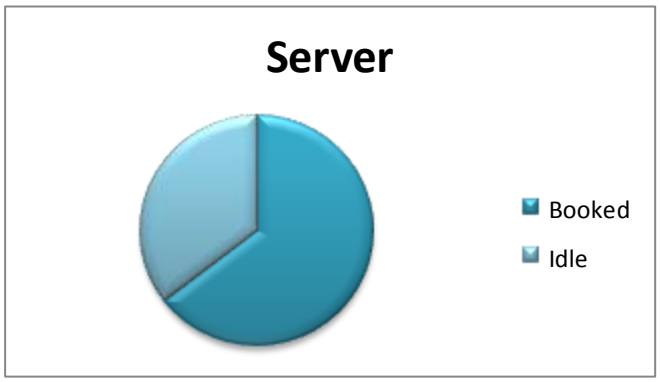

Fig. 4 Normal load on server

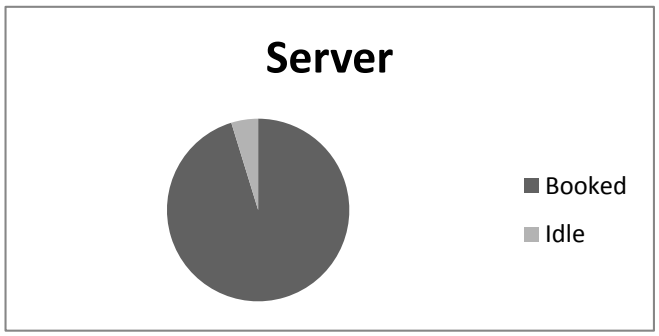

Fig. 5 High load on server

VIII. FUTURE WORK

The predictor predicts the future resource demands of VMs and the future load of PMs based on past statistics. The load of a PM can be computed by aggregating the resource usage of its VMs. In Future the resource utilization of servers are visualized in a graphical form in a cloud environment using cloud tool "EyeOS". User can easily identify the load on server from the graph and thus avoid overload on server. Thus reduce the average decision time regarding resource allocation decisions.

REFERENCES

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

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

[3] 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.

[4] Sheng Di, Cho-Li Wang, "Dynamic Optimization of Multiattribute Resource Allocation in Self-Organizing Clouds," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 3, pp. 464,478, 2013. doi: 10. 1109/TPDS. 2012. 144

[5] J. Sonneck and A. Chandra, "Virtual Putty: Reshaping the PhysicalFootprint of Virtual Machines," Proc. Int'l Hot Cloud Workshop in Conjunction with USENIX Ann. Technical Conf., 2009.

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

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

[8] R. Goldberg. "Survey of Virtual Machine Research," *IEEE Computer*, 7(6), June 1974.

[9] X. Meng et al., "Efficient Resource Provisioning in Compute Clouds via vm Multiplexing," Proc. IEEE Seventh Int'l Conf. Autonomic Computing (ICAC '10), pp. 11-20 2010.

[10] R. Nathuji and K. Schwan, "Virtual power: coordinated power management in virtualized enterprise systems," in *Proc. of the ACM SIGOPS symposium on Operating systems principles (SOSP'07)*, 2007.
[11] T. Sandholm and K. Lai, "Mapreduce optimization using regulated dynamic prioritization," in *Proc. of the international joint conference on Measurement and modeling of computer systems (SIGMETRICS'09)*, 2009.
[12] A. Singh, M. Korupolu, and D. Mohapatra, "Server storage virtualization: integration and load balancing in data centers," in *Proc. of the ACM/IEEE conference on Supercomputing*, 2008.

