

Dynamic memory Allocation using ballooning and virtualization in cloud computing

V Holy Angel Jenitha¹, R.Veeramani²

¹(PG Student, CSE Department, SRM University, India)

²(Assistant Professor IT Department, SRM University, India)

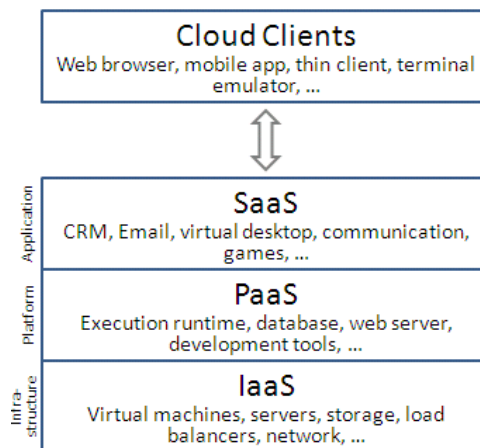
Abstract: Cloud computing has changed the way in which computer resources are used and shared. Once we register with cloud service provider we can access the software or hardware resources without the need to purchase our own product. Virtualization plays a vital role to maximize resource usage. In this paper, we define a system that dynamically release the unused memory pages and thereby can improve the performance of the overall system. We can maximize the memory usage when ballooning is used effectively with gray box analysis data. We can predict the memory needs by applying mathematical formula on working set samples. Once prediction is done, balloon list can be created to release the unused memory. Skewness algorithm is used to efficiently distribute load among the virtual machines.

Keywords: Ballooning, Cloud computing, Skewness, Usher, Virtualization

I. INTRODUCTION

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services [1]. It offers three service models: Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

- SaaS – Allows cloud users to access software applications .Google docs is an example
- PaaS – provides developer high-level integrated environment to design, build, run, test, deploy and update the applications created by client using development language and tool provided by the service providers. Skytap cloud is an example.
- IaaS - provides infrastructure to lease the processing power, storage, network and other basic computing resources, with which users can develop, run and store application. Amazon EC2 is an example



This paper is for Infrastructure clouds using the virtualization to maximize the resource usage. Virtual machine monitors are used to map physical machines (PM) with virtual machines (VM). Cloud users are not aware of this mapping and are implemented by the cloud providers. Cloud providers should ensure that the physical resources are properly managed to meet the user needs. Network resources will be released if there is no need for connection whereas memory behaves differently. Whether the system is in use by user or not, there will be some background processes running and consuming considerable amount of memory. We need to ensure that needed pages are available in main memory to ensure good performance.

When a Virtual machine starts its execution hypervisor allocate memory of value specified in the configuration file. At any instant of time, all the memory will not be used by the Virtual machine [3]. There will

be situation in which only few pages are reference and while others are unused. Also there can be other possibility of frequent page fault and performance degradation of few virtual machine instances. This paper present a dynamic memory allocation system that improves the performance of virtual machines by increasing/decreasing the memory usage based on the process running on it.

The goals are as follows

- **Overload Avoidance:** The memory allocated to virtual machine is properly managed to ensure good performance
- **Green computing by ballooning:** Release the used memory area of virtual machine to hypervisor for allocating to another machine that is in need of memory

II. SYSTEM OVERVIEW

The system architecture is given in fig 1. The architecture is based on Xen hypervisor which is a open source bare metal hypervisor available as open source.

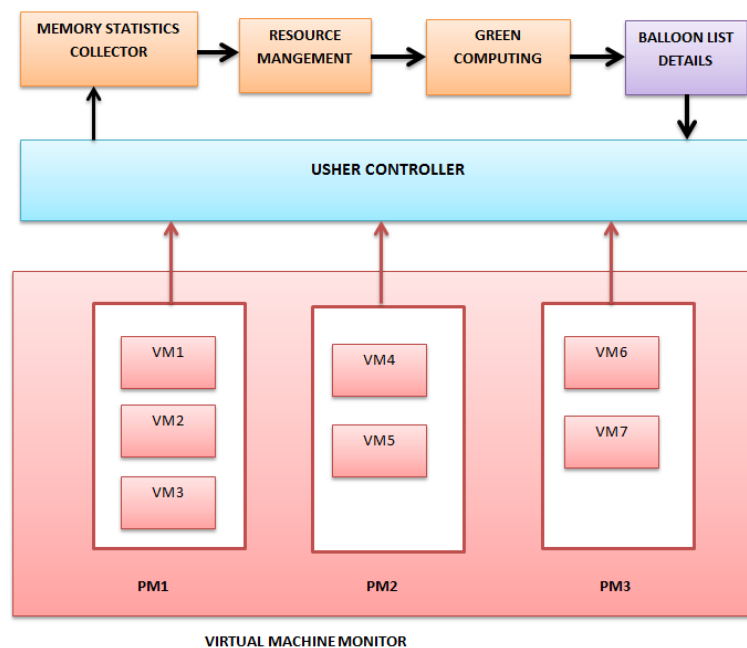


Fig 1 – System Architecture

Physical Machine (PM) runs the Xen hypervisor (Virtual Machine Monitor) which supports a privileged domain 0 and one or more domain U [2]. Domain U of each and every Virtual Machine (VM) encompasses different applications such as Web server, remote virtual desktop, Domain Naming Service, Mail, Map etc. Our assumption is all the PMs Share the same backend storage. Mapping of VMs to PMs is managed using the Usher framework [3]. The only place to implement our logic is to add our program logic as plugin to Usher frame work. Usher Local Node Manager (LNM) runs on each node of domain 0. It gathers the usage details of resources for each VM on that node. We need to know the memory usage of VMs which is not visible to the hypervisor. Different researchers proposed many ways to infer memory usage. One approach is to infer memory shortage of a VM by observing its swap activities [4] But this raise the need of separate swap partition in the guest OS. Best approach is to find out the working set size using Working set proper on each hypervisor. Random page sampling technique of VMware ESX Server [5] is used to choose page samples. The information collected at each PM are sent to User central controller (Usher CTRL) in which our system can be added as plug in. The scheduler program is called periodically (say every 5 minutes) and gathers the memory demand history of VMs, memory history of PMs and current global memory.

Future memory load of PM is the sum of memory usage of all the VMs running on it. Load prediction algorithm is based on mathematical model of moving average. At each node, LNM attempts to satisfy the new demands locally by adjusting the memory allocation of VMs sharing the same VMM. If memory is not available locally to satisfy we try to get from global pool. The Resource management module determines the skewness of memory usage and suggests the best VM to be allocated to next request. Green computing module continuously checks to ensure whether we can remove few pages from working set that are not used but still holding main

memory (Least recently used –LRU algorithm) . Balloon list are maintained to intimate the balloon driver on which VM, the host can request for memory release during shortage.

The various component of the proposed system are as follows.

- Memory Statistics Collector
- Resource Management
- Green Computing
- Balloon list details

The details of each component are given below

2.1 Memory Statistics Collector

We can have scripts that collect OS level details on memory usage of Guest OS, page fault and other information which is called gray box details [4]. The script is a simple shell that does not impact the performance. The details are collected and stored in file every five minute. This duration can be incremented /decremented based on demand by setting the appropriate sleep time for the script. We need to know the guest OS running in order to get the memory statistics and use the appropriate memory commands.

2.2 Resource Management

The different tasks carried in resource management are as follows

- Predict future memory demand using equation (1)
- Calculate the skewness using equation (2)
- Find the best VM to allocate memory for next request

We use exponentially weighted moving average (EWMA) to predict the future memory usage. Skewness is calculated to ensure that the resource utilization is even and avoid overload of any particular VM. With the calculate skewness value we can find the best VM to allocate the next request of user.

2.3 Green Computing

Green computing is the way to avoid unnecessary usage of any memory resources. Here we have threshold limits set for hot, warm and cold spots. If memory usage exceeds the threshold limit the performance will be affected and it will request more memory or go for page replacement. Our aim is to come up with list of VMs from which we can claim the unused memory.

2.4 Balloon List Details

Ballooning is the concept used in Xen to claim memory from guest OS when host is in need of memory using balloon drivers [6]. When host is in need of memory it will force memory pressure on all VM since it is not aware of the process happening inside Guest OS. Due to the pressure, Guest OS is forced to go for demand paging to move some pages to disk and free the memory. In our system, we provide the list of VMs on which host can claim memory so that it will not disturb the other VMs that are fully loaded with task. The balloon list has details of VM address and percentage of memory that is free. The balloon list is saved in database and updated each time when the update happens after each run of the script. The list is given to the balloon list driver when the host is having memory shortage and it need to force few VM to release the memory. Old system will induce memory pressure on all the available VMs where as our system insists the balloon drivers to claim the unused memory from the list of VMs that is having enough memory. This will ensure that loaded VMs are not disturbed and avoid too demand paging of them.

III. ALGORITHM AND EQUATIONS

Algorithms used in the proposed system are as follows

- Load Prediction algorithm
- Skewness algorithm

- Green Computing algorithm

3.1 Load Prediction Formula

We predict the future memory demand using exponentially weighted moving average (EWMA) using a TCP-like scheme:

$$E(t) = \alpha * E(t - 1) + (1 - \alpha) * O(t - 1) \quad (1)$$

$E(t)$ – Estimated memory usage at time t

$E(t-1)$ – Estimated memory usage of past time

$O(t-1)$ – Observed memory usage of past time

$$\alpha = \frac{O(t - 1)}{E(t - 1)}$$

α - Denote the tradeoff between the observed and estimated value

Future demands are forecasted to ensure the memory is properly managed and only unwanted memory area is released.

3.2 Skewness Calculation Formula

We can use skewness algorithm to ensure there is uniform utilization of memory across VMs

$$P = r/R - 1 \quad (2)$$

Where p = skewness to determine the usage level of memory by a VM

r – Memory usage of the current VM under evaluation

R – Average of memory usage by all VMS

If $p = 0$, the Memory utilization is normal

$P > 0$, then the memory usage is more compared to others VMS

$P < 0$, then the memory usage is less compared to other VMs

Example

VM1,2,3 area allocated 20 MB of memory each

VM1 has consumed 10 MB

VM2 has consumed 15 MB

VM3 has consumed 5 Mb

$$R = 30/3 = 10$$

$$\text{VM1's } p = 10/10 - 1 = 0$$

$$\text{VM2's } p = 15/10 - 1 > 0$$

$$\text{VM3's } p = 5/10 - 1 < 0$$

- New request can be allocated to VM3
- This ensures that none of the VMs are overloaded when compared with the usage of other VMs belonging to the same VMM.

3.3 Green computing algorithm

Threshold limits are set for hot, warm and cold spot values as 85, 60 and 25.

- Calculate the % of memory usage based on the predicted value of equation (1)
- Check against the threshold limit to check any VM falling under hot spot
- If the VM come under hot spot ,call page replacement algorithm

- If the VM fall under cold spot, add it to balloon list with % free details
- If the VM fall under warm temperature – No action
- When the VM is in cold spot we make entry in balloon list.

We have threshold set for release value – say 80% from free memory can be given. Balloon list will have VM ip address and free memory details

- Example: If the balloon list says VM1 has 10 MB free, then we will release 8MB (which is 80% of free memory) when memory pressure happens at host.

IV. CONCLUSION

The proposed system can be used by cloud service providers at data centers to manage the memory usage of OS related jobs. This can be used as an enhancement to balloon drivers' configuration and thereby can be used as an alternate to page replacement. Getting memory usage details is challenging as guest OS runs heterogeneous applications and each has different way to get usage. However when the guest OS environment is known, we can easily get memory usage from scripts. We do have some live VM migration techniques [7][8][9] that have more advantage over ballooning but they cost more than ballooning. Ballooning is a way to provide better memory utilization with low cost and providing additional information on which VM to increase/decrease memory yield high benefit to normal ballooning process.

REFERENCES

- [1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," technical report, Univ. of California, Berkeley, Feb.2009.
- [2] 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.
- [3] M. McNett, D. Gupta, A. Vahdat, and G.M. Voelker, "Usher: An Extensible Framework for Managing Clusters of Virtual Machines," Proc. Large Installation System Administration Conf.(LISA '07), Nov. 2007.
- [4] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Box and Gray-Box Strategies for Virtual Machine Migration," Proc. Symp. Networked Systems Design and Implementation (NSDI '07), Apr. 2007.
- [5] C.A. Waldspurger, "Memory Resource Management in VMware ESX Server," Proc. Symp. Operating Systems Design and Implementation (OSDI '02), Aug. 2002.
- [6] Dan Magenheimer, Oracle Corp. "Memory Overcommit... without the commitment" "Extended Abstract for Xen Summit, June 2008
- [7] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," Proc. IFIP/IEEE Int'l Symp. Integrated Network Management (IM '07), 2007.
- [8] 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.
- [9] Zhen Xiao, Weijia Song, and Qi Chen "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", Jun2013

BIOGRAPHY



V Holy Angel Jenitha is a M.Tech Student in the Department of Computer Science and Engineering, SRM University, Ramapuram, Chennai, Tamil Nadu, India. She has received the B.Tech degree in Information Technology from Panimalar Engineering College, Chennai (affiliated to Anna University) in the year 2008. Her areas of interest are cloud computing, networking and data warehousing. She has over 5 years of work experience in software industry.



R. Veeramani received the B.E degree in Computer Science and Engineering from University of Madras, Chennai, Tamil Nadu, India, in 1998 and M.E in Computer Science and Engineering from Annamalai University, Chidambaram, Tamil Nadu, India in 2006. He is currently an Assistant Professor at SRM University, Chennai, Tamil Nadu, India. He has more than 10 years' experience in teaching and development of e-commerce applications. His research interests are in Data Mining, Distributed Computing, Cloud Computing, Parallel System and Programming languages. In addition, he is member of Indian Science Congress Association.