



# **TPS- A Novel Approach to Increase the Number of Guest Virtual Machines by Reducing Physical Memory Usage**

Suresh, Anusha C S, Savitri A M

Dept. of CNE, Oxford College of Engineering, Bangalore, Karnataka, India

Dept. of CSE, University BDT College of Engineering, Davanagere, Karnataka, India

Dept. of CSE, GSSIT, Bangalore, Karnataka, India

**ABSTRACT:** Improving memory utilization is important for improving the efficiency of a cloud datacenter by increasing the number of usable VMs. Memory over commitment is a common technique for this purpose. Transparent Page Sharing (TPS) is a technique to improve the utilization by sharing identical memory pages to reduce the total memory consumption. For a cloud datacenter, we might expect TPS will reduce memory usage because VMs often execute the same OS and middleware and thus they may have many identical pages. However, TPS is less effective for Java - based middleware because the Java VM finds it difficult to manage the layouts of internal data structures that depend on the execution of Java programs. This paper presents detailed breakdowns of the memory usage of KVM guest VMs executing a Web application server. Then we propose increasing the amount of page sharing by VM.

**KEYWORDS:** Memory usage analysis; Memory reduction; Transparent Page Sharing; Virtualization; KVM

## **I. INTRODUCTION**

In a modern datacenter, server machines are virtualized as virtual machines (VMs) to improve the utilization of physical servers and to reduce the server management costs. In this environment, increasing the number of VMs on each piece of hardware is a key factor for higher cost effectiveness. For example, a cloud computing datacenter often uses a separate guest VM for each user to ensure security and isolate those resources from the other users. This kind of configuration runs many small VMs. Even small reductions of the resources used by each VM are important to improve the efficiency of the entire data center, especially if more VMs can run on each real machine. To increase the number of VMs in a machine, efficient use of memory is quite important, because the cost to move allocated memory from one VM to another is high, since the data in memory must be moved. Memory over-commitment is a key feature of hypervisors to improve memory utilization by allocating virtualized physical memory (guest memory) to guest VMs so that the total memory appears larger than the physical memory of the physical host machine. However, the performance is often degraded when the amount of actually used guest memory exceeds the amount of host physical memory, because the hypervisor starts paging out some of the guest memory to disk.

There are three approaches to reduce paging in a hypervisor.

1. One is to dynamically reduce the amount of guest memory and let the guest OS page out memory as needed [1].
2. The second approach is to improve the speed of the page-out and page-in operations by paging to RAM instead of disk [2] [3].
3. The third approach is to reduce the need for host physical memory by sharing a single host physical page among multiple guest memory pages if their content is identical, even though they are pages for different guest VMs. The third approach is called Transparent Page Sharing (TPS) [1]. The hypervisor detects the identical pages by scanning the memory in the background and shares them in a copy-on-write manner. This operation is transparent to the guest VMs because the sharing takes place via the address translation in the hypervisor. This technique is implemented in many hypervisors, such as VMware, using the Difference Engine [2] for Xen [4, 5], using KSM for KVM [6], and Power VM [7, 8].



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

For a cloud computing datacenter, VMs often execute the same OS and middleware and thus they may have many identical pages since many users will use standard disk images provided by the cloud service provider. Also, for a Platform as a Service (PaaS) datacenter, the service provider can select the OS and middleware to be used in the VMs, and they usually use the same software stack to reduce management costs. In these environments, we believe that TPS can effectively reduce the memory requirements. However, certain characteristics of TPS generally render it ineffective for Java-based middleware. A program suitable for TPS should have many pages that contain the same data on every execution. The identical pages should also be read-only for long enough periods to compensate for the overhead of scanning them.

## Advantages of VM

### 1. Run Old Applications

Got an application that won't play nice in Windows 7 or Vista, but works fine in XP or an even earlier version of Windows, Just grab an old Windows CD and install it within a virtual machine (VM). Then install your app. VMware Player features Unity mode, which allows applications running in the virtual machine to appear as if they're running natively on the host computer. They have their own taskbar buttons and their own program windows, making for a seamless experience. For this to work, however, you'll need to install the VMware Tools program on the virtualized operating system. You're usually prompted to do this after installation of the OS has finished.

### 2. Access Virus-Infected Data

Ever been sent a file that your antivirus program has flagged, but which contains important data you just have to view? Most virtualization software includes snapshot functionality, which means you can create a "saved state" of the virtual OS and its entire hard disk. It's a little like travelling back in time. You could create a snapshot in the virtual machine, open the infected file within the VM to access the data and, if the virus causes chaos, simply click to restore the VM snapshot. Hey presto--a clean virtualized computer.

### 3. Browse in Complete Safety

Why not install Windows on VMware Player, then install Firefox, and run it in Unity mode so it appears to run natively on the host computer? Essentially Firefox will be running in what's known as a sandbox, meaning that should it (or one of its plugins) get hacked while you're online, there'll be no absolutely no risk to your actual operating system. You could create a snapshot once everything's been configured in the virtual machine in order to get things back up and running quickly, should anything go wrong.

### 4. Test Software, Upgrades, or New Configurations

The virus testing technique above isn't limited to malware. You could use your virtual computer to test new software, updates, or even new configurations of software before you roll them out for real on your main OS. Some server administrators use virtualization to create a copy of an existing installation of an operating system, plus its data, which they then run virtualized and play with to see if configuration changes or updates will cause any harm. If you manage workstation computers and want to be sure a Windows update is OK before rolling it out, you could do the same thing, just test it in a virtualized machine first.

### 5. Run Linux on Top of Windows (or vice-versa)

Want to give Linux a try but can't face repartitioning your computer's hard disk? Provided that it would ordinarily install on your computer, you can run just about any operating system inside a virtual machine, including most Linux distros. Linux and Mac users have been using virtualization for years to run Windows on top of their chosen OS in this way. If you run a Linux machine for mail or Web services, as examples, having a desktop version of Linux for occasional use will also make it easier to communicate with the server. There's no need to install PuTTY on Windows to communicate via secure shell (SSH), for example, because Linux has that kind of thing built-in.

### 6. Back Up an Entire Operating System

Because the virtual OS is entirely contained within a series of files, backing it up is as simple as backing up any other files. It's the same with virtualized server installations too. If you're running a virtual machine on a server to host your mail server, and it's brought down by a hack attack, then bringing things back to working order is as simple as restoring



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

the backup files (assuming the vulnerability that allowed the hack is quickly addressed once things are up and running, of course).

Bear in mind that creating a copy of a VM creates legal issues. Backing up should be fine, but if you create a copy of a VM installation to give to a friend, for example, then you'll be contravening copyright laws (assuming they apply, as with Microsoft, but not always with Linux).

## 7. Create a Personal Cloud Computer

If you're out of the office, there's no need to take your laptop with you. Just leave it running (with power saving turned off!), take your mobile phone or tablet computer instead, and access the laptop via a Remote Desktop Protocol (RDP) connection over the Internet. This will let you access the same desktop environment you're used to, although there'll be no fancy graphics.

## 8. Run Headless For Web Development

Most virtualization software allows the virtual machine to run headless, which is to say, without displaying a desktop (or other user-interface). Essentially, the virtual PC runs in the background although accepts all other kinds of connections, such as networking. For people creating websites, this offers the possibility of running their very own private web server for testing purposes.

## 9. Make a Backup of Your Server for Emergencies

Amazon's Elastic Cloud Computing (EC2) service allows you to copy across any existing virtualized Windows 2008 Server installation for use on EC2 (eventually all kinds of server installs will be supported, such as Linux).

Creating periodic backups of existing server installs in this way could provide vital redundancy if a catastrophe happens to your existing server. After the hurricane hits, leaving your physical servers in a whirl of dust, all you'd have to do is boot up the EC2 image, reconfigure things slightly to take into account the different IP addresses, and then continue as usual.

## 10. Reuse Old Hardware

By installing Citrix XenDesktop on your Windows server, you can turn old, less powerful computers into thin clients, wiping out the need for a workstation IT upgrade budget. The clients access their personal desktop spaces on the server and there's little noticeable difference compared with running the operating system and applications locally. XenDesktop includes clever technology to avoid common thin-client pitfalls, such as the fact videos and animations don't play well, by shifting some of the processing work to the client computer.

XenDesktop also allows your workers to access their desktops from home, provided the server is configured to be publicly accessible and they have the right client software installed. You can even use mobile phones to connect to the desktop environments.

## Disadvantages of Virtual Machine

### 1. Magnified physical failures

Imagine you have ten important servers running on one physical host and its RAID controller runs amok, wiping out all of your hard disks. Don't say that this is not very likely, as we have already had two or three incidents from malfunctioning RAID controllers from well-known brands. There are several ways to compensate for this downside.

### 2. Degraded performance

There is no doubt that virtualization requires extra hardware resources. The problem is that it is almost impossible to estimate in advance how many extra resources will be needed. I know that there are capacity planning guides and tools but from my experience every piece of software behaves differently in a virtualized environment. We have applications that are quite modest as long as they run on a physical server, but when they were virtualized their resource requirement multiplied.

You can't do much if you have such applications. In our case, we had no choice but leave them on physical servers. Hence, the only solution to this problem is to thoroughly test each application with the virtualization solution of your choices.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

### 3. New skills

It sounds so easy – install a virtualization solution and then just deploy your servers as you are used to. Not really! Many things are different in a virtual environment. I will give you just one example. When we installed our first server virtualization solution, I instructed our administrators to test some of their servers in the virtual environment. After a week or so, an administrator told me that he could not test his server because there was no more RAM available on the host. I was quite surprised, as this server has enough capacity for 10 VMs.

When I logged on, only 3 VMs were actually running. What happened? Some of his fellow administrators had assigned the same amount of RAM to the virtual servers as their physical servers had required. It took me quite some time to convince them to change their working habits. When you buy a new physical server, it is common practice to equip this server with as much memory as your budget allows. This makes sense, as it takes time to order new memory modules and add them to the server. Even if you do not require it now, you will most likely require more RAM very soon.

Of course, this situation is different in a virtual environment. I assign blame to myself, as we should have discussed things in advance. I should have told the administrators that they first need to figure out how much RAM their servers really need using a performance monitoring tool. If their server requires more RAM later, it is not a big deal to assign more. I chose this simple example because it demonstrates that you have to do some rethinking when you work in a virtual environment. The fact that several administrators share one physical server causes problems that didn't previously exist. Of course, it is also necessary to acquire many new technical skills.

### 4. Complex root cause analysis

Virtualizing a server certainly implies big changes to the whole system. A new layer of complexity is added and can cause new problems. However, the main difficulty is that if something doesn't work as it is supposed to, it can require considerable extra efforts to find the cause of the problem. I have another example for this downside of server virtualization.

We installed a SUSE Linux server under Virtual Server 2005. Everything worked fine at first. But, then the admin reported that his SSH sessions often got disconnected. We had another Linux machine running on the same host which didn't show this behaviour, so we thought it must be a configuration problem on the Linux system. However, this skilled Linux administrator wasn't able to find the problem's cause. I then had the idea to move this virtual machine to another Virtual Server host – and the problem was gone. So did Virtual Server or Linux cause the problem? Well, I can't tell you. We never figured it out.

### 5. New management tools

Virtualization also has advantages, such as easier migration, cloning or snapshots. However, you can only take advantage of these new capabilities if you have the proper tools. Often, the tools that come with a virtualization solution are not enough, only supporting basic management tasks. This means that you need additional utilities, which cost both money and time. I am not only talking about such tools as VMware Virtual Centre or Microsoft Virtual Machine Manager (VMM).

Another important field is backup, or more precisely, disaster recovery. Of course, you can use your current backup software to secure your virtualized servers. However, one of the advantages of server virtualization is that disaster recovery becomes much easier and faster provided you have a backup solution that is able to perform live backups of the virtual machines and not just of the virtualized servers running in these VMs.

The problem lies in that there are no real standards when it comes to virtual server management. But, there are standards for server management in general. For example, there are many backup tools that allow you to secure your Windows, UNIX and Mac machines, but it is difficult to find a disaster recovery solution that supports all the various virtualization solutions out there. All in all, this means that your zoo of management tools will grow, meaning more work for you.

### 6. Virtual machine sprawl

Even though virtual server management can get quite complex, installing a new virtual machine is a piece of cake. You need a new server? Just clone your master image to a new VM and you are done within a few seconds. The problem is that the number of servers might grow faster than the number of admin who are supposed to manage them. It is good that even virtual servers have physical limits. As soon as you reach the limit of your virtual capacity, the virtual machine sprawl will naturally stop.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

## II. RELATED WORK

Ballooning is a technique to reduce paging in a hypervisor by dynamically reducing the amount of memory available to a guest OS. The guest OS may reduce its memory usage more efficiently than the hypervisor because it has more information about the usage of its memory pages. For example, it can reduce memory by shrinking its disk cache rather than by paging-out pages. However, this approach requires a resource manager that can decide on the size of each guest VM, and the effectiveness of ballooning depends on the heuristics used by the manager to set the sizes. For a hypervisor that does not provide such a manager, such as KVM, we cannot use ballooning unless we install a separate manager.

TPS, paging to RAM, and paging to disk. For paging to RAM approach, they devised two techniques. One stores entire pages to RAM after compressing them. The other calculates and stores in RAM only a set of differences from a similar page in the physical memory, a latter technique that is also called *subpage sharing*. The paging to RAM approach can improve the overall performance by reducing the overhead of paging in and out because the overhead to compress or calculate differences is much smaller than that to access the disk, as long as the reduction of the paging overhead outweighs the increased paging. A weakness of this approach compared to TPS is that every access to a compressed or sub-page-shared page requires restoring the full page, while there is no overhead for reading TPS-shared pages. TPS is quite good for reducing the memory usage of read-only parts of the Java class areas, because there is no runtime overhead for reading the class metadata. Active Memory Expansion [9] in Power VM [10] also implements memory page compression in a production hypervisor, though it does not support sub-page sharing.

Memory Buddies [11] tries to increase page sharing by collocating guest VMs that execute similar workloads, expecting those VMs to have more similar pages than the guest VMs executing different workloads. Their approach improved the sharing for native programs, but the amount of shareable memory was small for their Java application (SPECjbb). They only noted that the sharing was limited because SPECjbb is a memory-intensive program and most of its memory is soon overwritten. This is true for the Java heap, but they did not analyze the VM native memory area.

For Software as a Service (SaaS) datacenter, multitenancy is another approach for reducing physical memory usage. It is an approach that runs only a single instance of the middleware and uses separate instances for each of applications that run in the shared middleware. Java has the Application Isolation API (JSR 121) [12] to help implement a multi tenant server. The advantage of this approach over the VM-based one is smaller memory consumption because the middleware and the OS is shared among all of the applications. The disadvantage is that a misbehaving application may affect other applications running with the same middleware. For example, if one application uses up all memory available to the server process, other applications may crash because of memory shortages. Another example would be if one application crashes upon access to an invalid memory region, so the entire service process can crash.

These resource and fault isolation problems have been studied for many years. Multi-User Virtual Machine (MVM) [13, 14] is an efficient implementation of the Application Isolation API and it also has limited support for memory quota enforcement and fault isolation. It counts usage of the Java heap for each application to enforce memory quotas, and it isolates Java code from crashes in user-provided JNI methods by executing them in a separate process. J-RAF2 [15, 16] implemented enforcement of CPU quotas by using byte code modifications.

## III. PROPOSED ALGORITHM

Virtual machines provides a complete system platform which supports the execution of a complete operating system and multiple OS environments can co-exist on the same computer, in strong isolation from each other, due to this advantage users wants different platforms in a single hardware but increase in the VMs directly proportional to decrease in host system performance, This paper focuses on breaking down the memory usage of Guest VMs to balance the total memory usage of all platforms in the host machine, by this system performance can be maintained.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

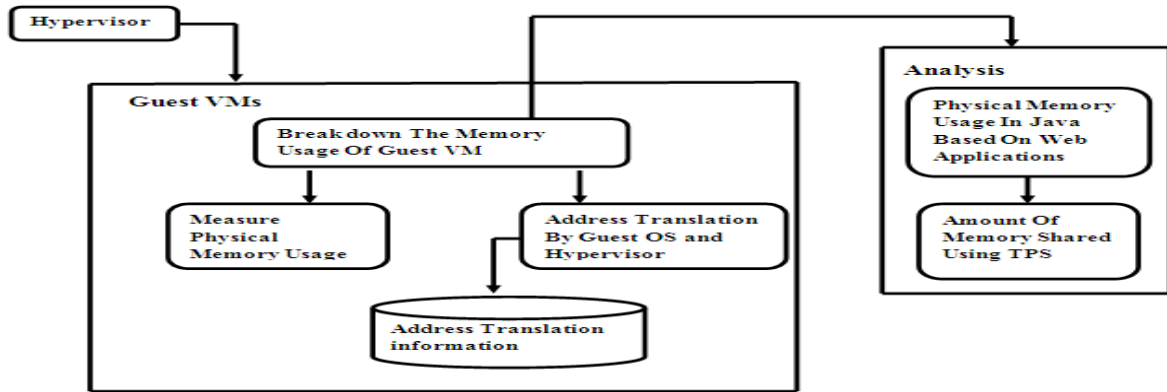
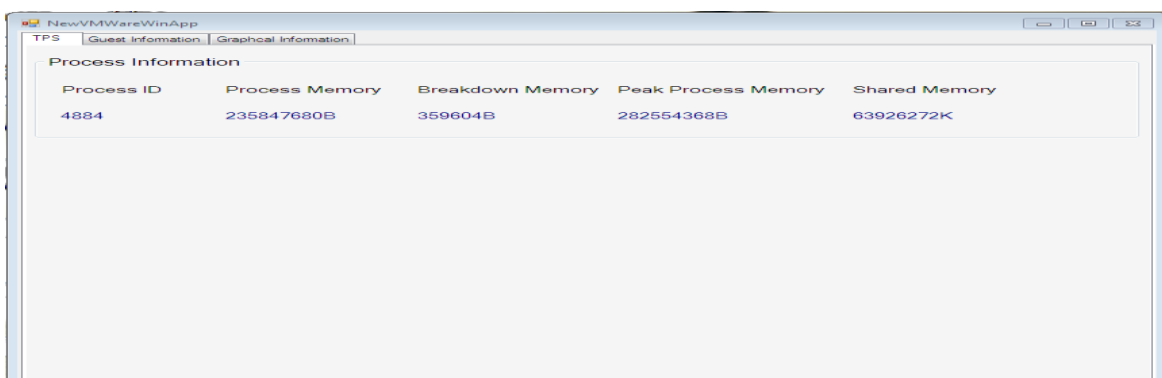


Fig 1: Proposed System

## IV. PSEUDO CODE

- Step 1: Measuring physical memory usage of guest VMs by TPS.
- Step 2: Breaking down the memory usage of guest VMs by TPS.
- Step 3: Memory Analysis, which is used to check
  - if (memory utilizing by VMs process < previous usage i.e. amount of memory saved by TPS)
  - {
  - Checking the Host system performance whether it is acceptable one or not
  - Else
  - Breakdown of memory usage carried out again.
- Step 4: Go to step 2
- Step 5: End

## V. SIMULATION RESULTS



Process ID	Process Memory	Breakdown Memory	Peak Process Memory	Shared Memory
4884	235847680B	359604B	282554368B	63926272K

Fig 2: Breakdown and Sharing of Memory after Connecting VM to Host Machine

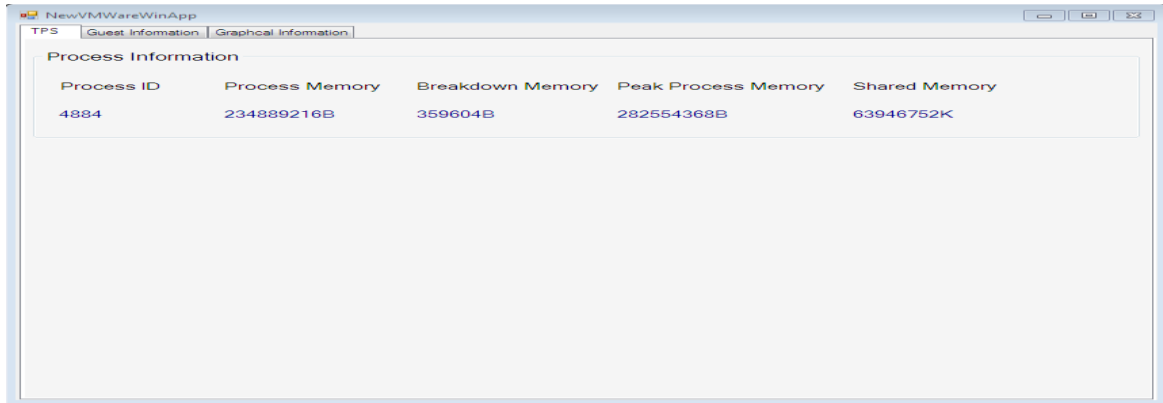
Fig shows the detailed breakdown memory and sharing of memory when Host machine is connected to virtual machine i.e. Guest machine, the figure 2 shows the particular process and respective process memory breakdown memory and shared memory.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014



Process ID	Process Memory	Breakdown Memory	Peak Process Memory	Shared Memory
4884	234889216B	359604B	282554368B	63946752K

Fig 3: Changes in Process Memory and Shared Memory after Adding Process to Guest Machine

The fig 3 shows the changes in the process memory and shared after adding some process in the virtual machine or else by executing some process in the virtual machine.

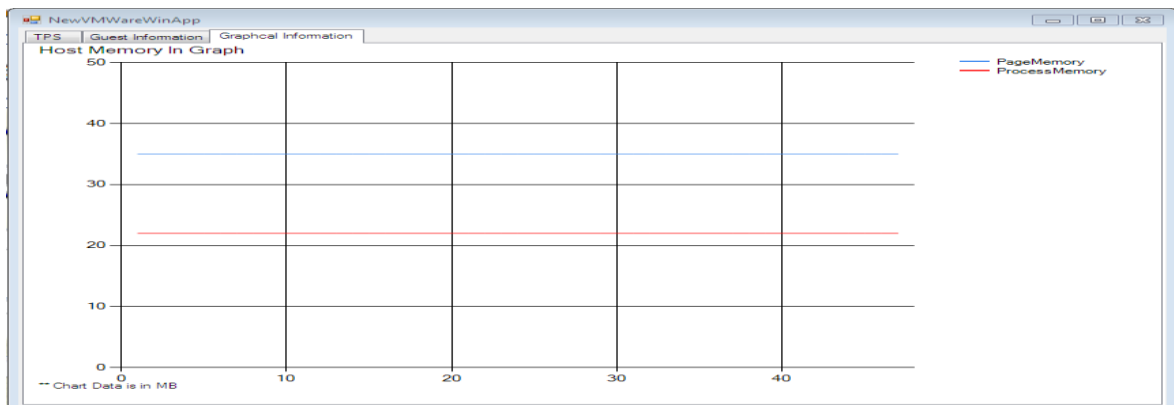


Fig 4: Graphical Analysis before adding process to VM

The fig 4 shows graphical analysis before adding process to virtual machine.

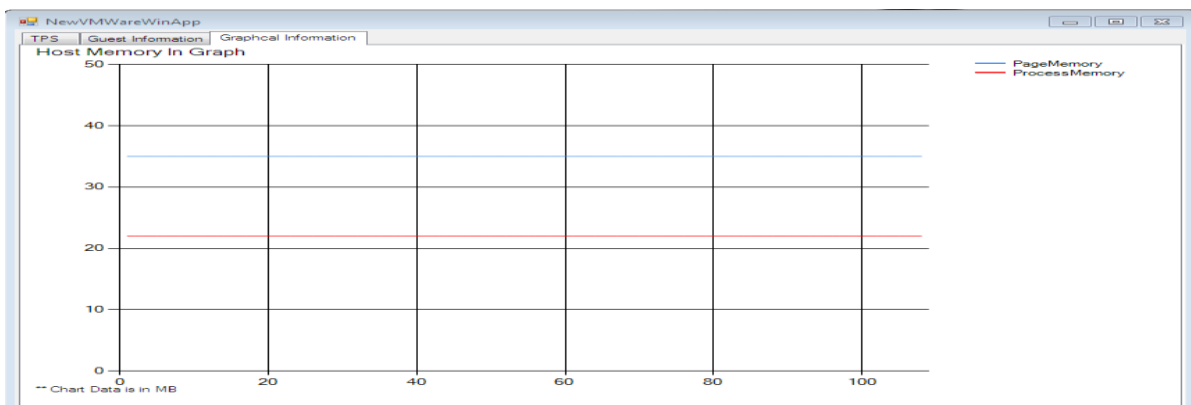


Fig 5: Graphical Analysis after adding process to VM

The fig 5 shows variations in the graph after adding some process to virtual machine.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

## VI. CONCLUSION AND FUTURE WORK

In this results the memory management in host machine and guest machine, breakdown memory can be analysed. After adding some process in the virtual machine changes in process memory and shared memory can be detected.

The memory usage of guest VMs in the host machine could always a path for degradation in the performance of the host machine, this project effectively analyses the memory usage of both guest machine and Host machine then Breakdowns the memory usage of guest machine and This approach reduced the physical memory usage for the class area by up to 89.6% by using TPS. The total memory usage of each of four guest VMs was also reduced by 9%. By this approach, reducing the physical memory usage increased the number of guest VMs running with acceptable performance.

## REFERENCES

1. Carl A. Waldspurger. Memory Resource Management in VMware ESX Server. (OSDI '02). 2002..
2. Diwaker Gupta, Sangmin Lee, Michael Vrible, Stefan Savage, Alex C. Snoeren, George Varghese, Geoffrey M. Voelker, and Amin Vahdat. Difference Engine: Harnessing Memory Redundancy in Virtual Machines. 8th USENIX (OSDI '08), pp. 309-322. 2008..
3. David Hepkin. Active Memory Expansion: Overview and Usage Guide. February 9, 2010..
4. Arcangeli, Izik Eidus, and Chris Wright. Increasing memory density by using KSM. In Proceedings of the Linux Symposium 2009, pp. 19-28..
5. Kernel Samepage Merging. <http://www.linux-kvm.org/page/KSM>
6. KVM. [http://www.linux-kvm.org/page/Main\\_PageD](http://www.linux-kvm.org/page/Main_PageD).
7. Rodrigo Ceron, Rafael Folco, Breno Leitao, and Humberto Tsubamoto. Power Systems Memory Deduplication. REDP-4827-00.
8. Stuart Devenish, Ingo Dimmer, Rafael Folco, Mark Roy, Stephane Saleur, Oliver Stadler, and Naoya Takizawa. IBM PowerVM Virtualization Introduction and Configuration. SG24-7940-04.
9. David Hepkin. Active Memory Expansion: Overview and Usage Guide. February 9, 2010. POW03037-USEN-00. [http://www.ibm.com/systems/power/hardware/whitepapers/am\\_exp.html](http://www.ibm.com/systems/power/hardware/whitepapers/am_exp.html)
10. Stuart Devenish, Ingo Dimmer, Rafael Folco, Mark Roy, Stephane Saleur, Oliver Stadler, and Naoya Takizawa. IBM PowerVM Virtualization Introduction and Configuration. SG24-7940-04.
11. Timothy Wood, Gabriel Tarasuk-Levin, Prashant Shenoy, Peter Desnoyers, Emmanuel Cecchet, and Mark D. Corner. Memory Buddies: Exploiting Page Sharing for Smart Colocation in Virtualized Data Centers. In Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE '09), pp. 31-40. 2009.
12. Oracle Corporation and/or its affiliates. JSR-000121 Application Isolation API Specification. <http://jcp.org/aboutJava/communityprocess/final/jsr121/index.html>
13. Grzegorz Czajkowski and Laurent Daynès. Multitasking without compromise: a virtual machine evolution. In Proceedings of the 16<sup>th</sup> ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA '01), pp. 125-138. 2001.
14. Grzegorz Czajkowski, Laurent Daynès and Nathaniel Nystrom. Code Sharing among Virtual Machines. In Proceedings of 16th European Conference (ECOOP '02), pp. 155-177. 2002.
15. Walter Binder and Jarle Hulaas. A Portable CPU-Management Framework for Java. In *IEEE Internet Computing*, Vol. 8, No. 5, September/October 2004, pp. 74-83. 2004.
16. Jarle Hulaas and Walter Binder. Program Transformations for Portable CPU Accounting and Control in Java. In Proceedings of 2004 ACM SIGPLAN Symposium on Partial Evaluation and Program Manipulation (PEPM'04), pp. 169-177. 2004.

## BIOGRAPHY

**Suresh** Perceiving Master of Technology (MTech) in Computer Networks Engineering Department, Oxford College Of Engineering, Bangalore, Karnataka India.

**Anusha C S** Perceiving Master of Technology (MTech) in Computer Science and Engineering Department, University BDT College Of Engineering, Davanagere, Karnataka India.

**Savitri A M** Perceiving Master of Technology (MTech) in Computer Science and Engineering Department, GSSIT, Bangalore, Karnataka India.