IM-VDE: In-Memory Virtual Desktop Environment A performance comparison of a new Virtual Desktop paradigm

Dai Hoang Tran
Department of
Computer Engineering
Kyung Hee University
Email: dai.tran@khu.ac.kr

Tien-Dung Nguyen
Department of
Computer Engineering
Kyung Hee University
Email: ntiendung@khu.ac.kr

Eui-Nam Huh
Department of
Computer Engineering
Kyung Hee University
Email: johnhuh@khu.ac.kr

Choong Seon Hong
Department of
Computer Engineering
Kyung Hee University
Email: cshong@khu.ac.kr

Abstract—In-memory Computing (IMC) is the new trend for enabling high-performance computation and fast data processing. It is currently being used for large enterprises, e-commerce shops who need real-time interactions, low latency responses and instant results. Given the enhancement of the IMC, we apply this new paradigm to the Virtual Desktop Environment (VDE), and look into the performance differences in comparison with traditional VDE. The end results shows positive feedback, but there are trade-offs we need to concern for the In-memory Virtual Desktop Environment.

I. INTRODUCTION

With the advance of technology, modern business information system nowadays consumes and processes huge amount of data in their daily operations. Not only that, but the end-users also have higher expectation in the applications they interact with, such as immediate responses and smooth experience in their interaction. Thus, the need for a better mechanism to handle massive amount of data in short period of time and meet real-time constraints is a must in nowadays computing system, and one of the solution is IMC. IMC refers to a technique that, instead of using big, complex data format that stored in slowed access disk storage, the data now resides directly in memory for quick random access. The random access speed to memory is about hundred thousand times faster than mechanical hard drive. The gap is so large, that not only using IMC solves the problem of speeding up processing time, it opens many other computation areas that using mechanical hard disk storage just cannot afford.

In recent years, IMC is gaining popular in the IT industry, thanks to hardware's cost decrease in producing memory components. This has enabled many software vendors to begin to offer their software products utilize the IMC approach. This new computing paradigm still evolves slowly, and most of the IMC innovations have their target on the database system, that usually incurs lot of input/output (I/O) access to database stored on the hard disk. SAP HANA [1] is one popular business data processing software for large enterprise, that uses IMC approach, and is described as a new platform that dramatically accelerates analytic, business processes, sentiment data processing, and predictive capabilities. Another interesting research topic of IMC is detailed in [5], in which the authors built a Version Control System (VCS) that runs

in-memory in the cloud, called Protego. Their system design approach is considered to be flexible, automation with scalable characteristic of the cloud environment.

Following the trend of IMC, one particular computing service we look at is the Virtual Desktop Service (VDS), that normally runs on top of a Virtual Desktop Infrastructure (VDI). Nowadays, an enterprise VDI system is analogy to a cloud platform, that contains multiple physical servers run on virtualization hypervisor such as Xen [2] or VMware vSphere [3]. Then, virtual desktop machines (VDMs) can be created by these virtualization platforms. Each VDM is a complete desktop environment, its resources including Central-Processing-Unit (CPU), memory and hard-drive are virtualized from physical hardware, this makes resource scaling and provisioning simpler. End-users connect to those VDMs by using thin client terminals, that usually have pre-installed remote control software to connect to using protocol such as Virtual Network Computing (VNC) or Remote Desktop Protocol (RDP). We call this environment Virtual Desktop Environment (VDE). Even-thought traditional VDE offers sufficient performance to normal users, power users who need fast computational machine and smoothness in their interactions still find traditional VDE does not give them the satisfactory. And usually the bottle neck comes from the slow access speed of physical hard disk. It is much slower in I/O access than memory flash such as DRAM (Dynamic Random Access Memory). In this paper, we propose using a different approach for providing VDE that put the whole environment into DRAM, and we call this new VDE approach the "In-Memory Virtual Desktop Environment" (IM-VDE).

In this research, we conducted a series of benchmarks that compare the performance of VDMs running in tradiational VDE and in IM-VDE, on both server-side and client-side perspective. On the server-side, most of the tests are to compare the performance gap between a VDM running with IMC approach and the other one running in normal hard disk. On the client-side, we focus on the delivery of image quality and smoothness in frame-rate that end-users would perceived. All of the tests make use of many open-source benchmarking software.

The remaining of the paper is organized as follows: chapter two shows related work within IMC field, chapter three shows
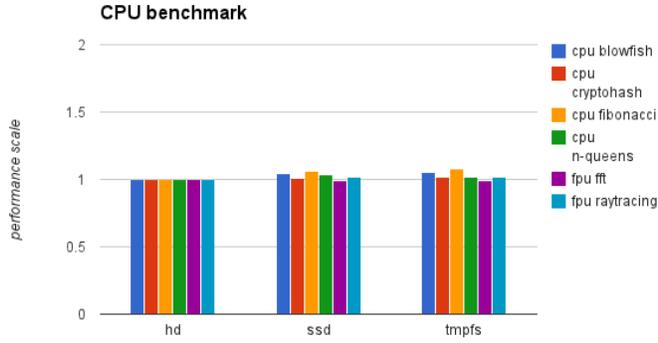
Fig. 1. HardInfo benchmark

Fig. 2. Sysbench benchmark

our approach of the experiments. Chapter four elaborates the results and the issues that we face within our experiment on IM-VDE, what are the drawbacks that need to be resolved, and we draw our conclusion in chapter five.

II. IN-MEMORY VDE APPROACH

A. Server-side performance benchmark

In preparation for the benchmarking, we use a physical server with hardware specifications as follow: Intel i5-2500 CPU with 4 cores, 16 gigabytes of DRAM memory, one SSD, one mechanical hard drive (HD)and the host OS is Linux distribution "Ubuntu Saucy 13.10". Next step, we prepare three VDMs using "VMware Player" software with the guest OS is also "Ubuntu Saucy 13.10". We then distribute those VDMs to reside on different file-systems, one stays in SSD, one stays in HD and the last one resides in TmpFS. Once the set-up is done, we conduct our benchmarking using state-of-the-art open-source software to accomplish this task. The first task is using a benchmark tool call "HardInfo: system profiler and benchmark" [4]. "HardInfo" can gather information about the OS environment, and do several benchmarks that mostly related to CPU-bound performance. By performing each test several time, we take the average results and compare the difference. The benchmark's result is presented in Figure 1. Beside "HardInfo", we use another open-source tool called "SysBench", which is a modular, cross-platform and multi-threaded benchmark tool. One particular salient feature of this tool is the focus on database benchmark under intensive load. This will give us some perspectives in real life practical environment. Similar to aforementioned benchmarking, we calculate the average results of several tests and compare with other VDMs. The benchmarking result is illustrated in Figure 2.

B. User-perceived performance benchmark

Benchmarking the VDE performance from the end-user perspective is not a simple task. It does not work by simply running a few benchmark software to know the results. In the VDE, user interaction with the remote graphical terminal is accomplished by using different remote access protocols such as VNC or RDP, which are commonly used in Linux and
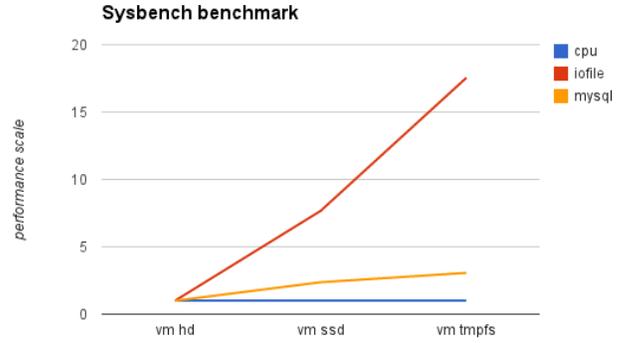
Windows OS respectively. Hence, to measure the performance of VDMs when users are using them, we mainly concern how quick and smoothness the rendering of remote server's graphical environment to client terminal. Another factor to consider is the type of end-user who operates the client. An office worker who mainly works with word processing application can tolerate the performance of VDM better than a power user such as a programmer or designer, where the smoothness and continuous flow of interactive responses are very important to them. Lastly, the network condition also plays a crucial role in the VDE, but in our test, we assume that the network condition is stable, since what we focus on is how the performance of a VDM running in-memory differs with a VDM running on HD. Thus with a stable network, it can help us to see the differences better.

With these regards, we have decided to conduct the tests as follows: we use the VNC protocol for the remote graphical solution, since it is open-source and platform independent. Then, for each VDM on each file-system, we respectively remotely connect to the machine, run automate script to start up many popular applications sequentially, and finally play the classical "Big Buck Bunny" video. We then record a video of the interactive session on the client-side. After that, we combine those recorded videos by putting into a side-by-side comparison video-clip, and gives it to a group of volunteers who will then rate them in scale from one to ten. By using this method, we can really reflect the end-user perspective on how they perceive the performance. The survey result is shown in Figure 3.

III. EVALUATION

In this section, we give our evaluation results regarding the experiments we conducted, and discuss some of the challenges, problems we found during the testing.

A. Results

From Figure 1, we conclude that in term of CPU-bound benchmark, the IM-VDM has the same performance as a normal hard disk VDM (HD-VDM). It is understandable since the synthetic benchmarks of the CPU do not relied on other hardware components such as memory and/or persistent
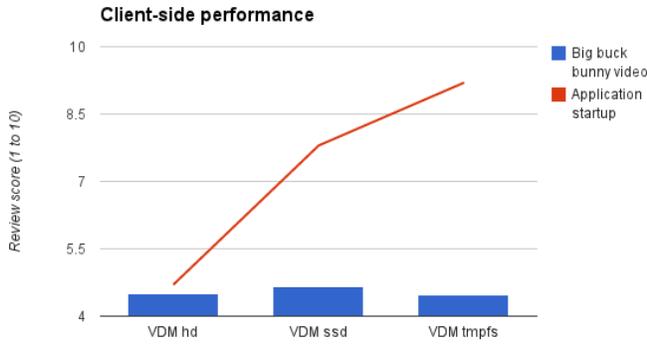
Fig. 3.   Survey result

device. And given two identical CPUs, those VDMs should yield similar performing results. But when we look at Figure 2, a big gap with order of magnitudes by tenth between IM-VDM and HD-VDM is clearly seen. The I/O access speed gives a huge speed up, thus this is one of the main benefit we get from IM-VDM, that is the access speed of data. Not only it speeds up the processing power of the applications, it also provides a better computational mechanism to solve problems faster.

For the client-side, the survey result in Figure 3 shows some differences in user-perceived performance. During the process of booting up multiple applications, thanks to speedy I/O access, applications start almost instantly, therfore we have high-score ratings from the volunteers. On the other hand, for the streaming of "Big buck bunny" video, which is a CPU-bound task, shows equally average results.

B. Challenges

Even though utilizing VDM to run in-memory give us positive boost in performance, there are challenges and issues we encountered that should be addressed. Firstly, as soon as we start the benchmarking, we have disk space problem with IM-VDM. Since we have limite amount of DRAM in our testbed, the IM-VDM cannot affort to have large storage size. This is the common issue for IMC approach. To overcome this, we need to provisioning lot of DRAM memory, this leads the another problem, which is cost effective. We know the price for DRAM memory has been reduced significantly over years, but the price of HD also reduces a lot. Given the price per gigabytes of nowadays commodity hardware, a DRAM flash for desktop would cost $10/GB, while HD's cost is only $0.08/GB.

Not only we have disadvantage in storage space, using TmpFS also wastes lot of memory resources, as we encoun-tered the phenomenon we called "double memory usage". Because the OS treats the TmpFS memory block as normal disk block, it still loads the data into main memory one more time to process that data instead of accessing the memory block directly in the TmpFS region. This in some case will cost twice the amount of memory usage for one program.

Last but not least, we have fault-tolerance and persistent

issue. Since DRAM is volatile, in case of unwanted events happen such as electricity lost, the VDM will lost all the memory storage. Hence, we need a proper backup mechanism for in-memory file-system as also concern in [6]

IV. CONCLUSION

After thorough benchmarking of our IM-VDE testbed, it is clear that IM-VDE provide positive benchmark results. But there are many concerns that we need to take into consideration when using IM-VDE. The first one is the performance-gain in trade-off with the storage shortage because the cost of DRAM is still quite high in comparison with HD, but it provides significant boost up in term of computing power. The second concern is the type of end-user for the VDE. If he/she runs tasks that depend on CPU but not much I/O accesses involve, the positive effects are often unnoticeable. Currently, IM-VDE is still a relativity new Virtual Desktop Environment paradigm, there are still many challenges but we believe IM-VDE will set the new trend.

V. ACKNOWLEDGMENT

REFERENCES

[1] "Harness the Power of SAP HANA In-Memory Computing." In-Memory Technology. http://www.sap.com/pc/tech/in-memory-computing-hana.html

[2] The Xen Project, the Powerful Open Source Industry Standard for Virtualization." The Xen Project, the Powerful Open Source Industry Standard for Virtualization. http://www.xenproject.org

[3] "VSphere Hypervisor." Free VMware , Free Virtualization (ESXi). http://www.vmware.com/products/vsphere-hypervisor

[4] "HardInfo." : System Profiler and Benchmark. http://hardinfo.berlios.de

[5] Gioachin, F.; Qianhui Liang; Yuxia Yao; Bu-Sung Lee, "Protego: In-Memory Version Control System in the Cloud," Software Engineering Conference (APSEC), 2012 19th Asia-Pacific , vol.1, no., pp.232,239, 4-7 Dec. 2012

[6] Hyunjun Kim, Joonwook Ahn, Sungtae Ryu, Jungsik Choi, and Hwansoo Han. 2013. In-memory file system for non-volatile memory. In Proceed-ings of the 2013 Research in Adaptive and Convergent Systems (RACS '13). ACM, New York, NY, USA, 479-484.