Operating System Virtualization to support E-learning with Affective Intelligent Tutoring Systems

Chris Messom, Abdolhossein Sarrafzadeh, Anton Gerdelan, Martin Johnson, Jamshid Shanbehzadeh[1]
Institute of information and Mathematical Sciences, Massey Unv., Auckland, New Zealand
[1] Tarbiat Moalem University
{c.h.messom, h.a.sarrafzadeh, a.gerdelan, m.j.johnson}@massey.ac.nz

Abstract

This paper introduces an operating system virtual machine platform for deploying Affective Intelligent Tutoring Systems in an E-Learning environment. Managed E-learning environments are often web based and rely on a browser to minimize configuration on the client machine, however even in these scenarios it is not unusual to have very specific requirements of the client machine, browser type and version, Java™ virtual machine version and even operating system. In the case of Affective Intelligent Tutoring systems there are additional requirements associated with the affective sensors, camera, bio-mouse etc. An operating system virtual machine approach allows the software stack of the client to be pre-configured and distributed in one shot. This minimizes the client side configuration easing the adoption of the technology. This paper identifies some of the key performance issues associated with this approach.

1. Introduction

E-Learning is revolutionizing the delivery and management of learning at all levels of the curriculum. However progress has come at the cost of complexity in delivery platform that has to be managed by the School, College or University. The campus based computer facilities are often under the control of the IT services departments and they will normally provide custom configuration to support a given E-Learning application or tool. However when incompatibilities exist between the configuration requirements of different applications more aggressive approaches to management have to be taken. These include reinstalling system images to match the users and application currently in use which means the conflicting applications can not be used simultaneously. In addition most E-Learning enterprises allow access from home or mobile clients that are not under the control of the IT services, in this scenario minimizing conflicts is essential, but currently the only viable solution is to specify versions of the browsers, Java™ virtual machines and operating systems and minimize the number of applications and tools available to the clients. This paper introduces the operating system virtual machine model to overcome these problems.

The E-learning community is familiar with the concept of Java™ virtual machines for the execution of rich interactive content, however there is limited understanding of operating system (OS) virtual machines. OS virtual machines provide a complete operating system environment in a software container that runs on another operating system. The advantage of this approach is the ability to run many different types of operating system on one system without the need to reboot the machine as well as to provide isolated environments for applications that cannot be trusted. OS virtual machines have only recently been studied by the E-learning community in terms of providing hands on experience for students in the field of operating systems, networking and security providing a framework for simulating multiple machines on a single hardware platform [1, 2]. There has been little or no study of OS virtual machines as a platform for managing and delivery of E-learning solutions. The main reason for this is the web based delivery of most E-learning environments that try to minimize the configuration required on the student's client machine. Although most web based E-learning solutions would not require an OS virtual machine delivery platform, there are many client server and peer to peer based E-learning systems that would benefit from the ability to preconfigure OS virtual machine images with the necessary software for supporting the system [3-8]. In addition the ability to run multiple OS virtual machines on a single hardware system offers the ability to deliver systems that have conflicting configuration requirements.

978-1-4244-1841-1/08/$25.00 ©2008 IEEE

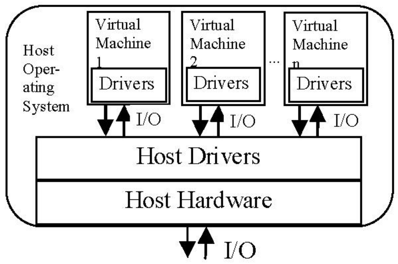

Figure 1. Virtualized guest operating systems on host

Affective Intelligent tutoring systems require additional hardware beyond the traditional pc, including webcam, microphone, speakers, bio-mouse and other sensors that can identify the affective state of the user. Interfacing to this hardware often requires specific drivers that need to be installed on the system, in addition the software that identifies users, facial expressions, gestures and affective state is often complex and needs to be installed on the student's pc in either client server or peer to peer mode. The burden of providing the hardware and software stack that can correctly interface to the affective intelligent tutoring system can be passed to the manager of the E-learning system by providing an OS virtual machine image with all the necessary software preconfigured for the student.

2. Operating System Virtualization

Operating system virtualization is becoming more popular for servers enabling one machine to easily host multiple operating systems (figure 1). All modern operating systems now support virtualization, the ability to run a guest operating system within the operating system. There are several different forms of operating system virtualization, but the most interesting in terms of performance is paravirtualization. Paravirtualization requires the guest operating system to be modified so that the host operating system can gracefully manage the guest operating systems. With modern processors adding hardware support for virtualization, the modifications required for paravirtualization have reduced significantly.

Although OS virtualization is very popular for servers, OS virtualization on the client side is only now becoming wide spread. There are several reasons for this, in particular the hardware requirements (large memory, high performing cpu to overcome virtualization overhead), but another reason is that users are only just becoming aware of advantages of virtualization. The most popular use of client side OS virtualization is in virtual appliances which package some functionality within a virtual machine. This means that the user does not need to install and configure multiple software applications and ensure compatibility, rather they install the whole virtual machine operating system with the software preinstalled and preconfigured. One virtual appliance that is particularly popular is the browser appliance that runs a browser in a separate virtual operating system, this browser cannot infect the host machine as it runs within the virtual machine and any viruses or Trojans planted via rouge code can be removed by closing the browser appliance and starting a new (uninfected) instance of the virtual machine [9-11]. Although browsers appliances and other virtual machine appliances can be used to promote security care must be taken that the virtual machine platform is not a vector for distributing malware [12].

3. Affective Intelligent Tutoring Systems

Alexander et al. [13] have pioneered the use of affect in intelligent tutoring systems. Although the use of affect in intelligent tutoring systems has been shown to not significantly improve student performance there is evidence that a persona effect can be induced if the intelligent tutoring system is able to respond with emotional intelligence. Affective intelligent systems can make use of multiple input and output devises to identify the emotional state of the users and to project an artificial emotional state of the intelligent tutor. These devices include standard devices such as webcams, microphones, speakers/ headphones as well as more specialized devices such as temperature, blood pressure, blood sugar, ecg and heartbeat sensors. Interfacing to these IO devises requires both low level drivers as well as high level application software for processing these inputs. Alexander et al's emotion detection system takes a webcam image and in real time identifies the affective state of the user based on facial expression (figure 2).

Figure 2. Affective state estimation using facial expressions

Installing and configuring the drivers and application software for these systems can be very tedious making a virtual machine image of a fully installed operating system an ideal distribution and delivery platform for affective intelligent tutoring systems. A key issue associated with using virtual machines for a distribution and delivery platform are the associated performance characteristics. The following sections investigate the cpu and IO performance of virtualized operating systems. The key IO devices for affective intelligent tutoring systems are the disk IO as well as usb IO since most off the shelf components use these protocols for interfacing with webcams, microphones, mouse etc. As also identified by Cherkasova et al [14] and Menon et al [15], this study shows that the IO performance is significantly degraded by virtualization but it is not degraded enough to impact the performance of the affective intelligent tutoring systems on modern computing platforms.

4. Virtual Machine Performance

4.1 CPU performance

Virtual machines have a reputation for good cpu performance. Barham et al. [16] show that applications running in a paravirtualized machine can run within 95% performance of the native host performance across a variety of application benchmarks. Currently a paravirtualized system requires guest operating systems that have been modified to support the virtualization and this modification may not exist for some intelligent tutoring machine platforms. In this scenario unmodified guests must be used with the associated performance impact. Table 1 shows the cpu performance of the native host system versus the fully virtualized guest system using the Linpack benchmark. The linpack benchmark is cpu intensive application that is frequently used to benchmark floating point performance.

Table 1: CPU scalability of OS virtualization using Linpack (average performance Gflops).

Problem Size	Host System (2 cores)	Virtual System (2 cores)	Ratio of Virtual to Host performance
1000	11.70	8.53	73%
2000	15.25	10.21	67%
4000	16.94	12.47	74%
5000	17.39	12.75	73%
10000	18.26	13.71	75%

As shown in table 1, when using a fully virtualized (rather than paravirtualized) guest operating systems there is a significant degradation in performance, the virtual machine performance being about 70% of native performance across a number of different problem sizes. Small problem sizes have a larger variance so the larger problem sizes in table 1 give a more reliable measure of performance, however in most intelligent tutoring systems quick response is required which only occurs if the computational load is small (this matches the small problem case). Although 70% performance ratio shows a significant difference between the virtual machine and host system it is still good enough for our affective intelligent tutoring systems. The new computer platforms tend to be more that twice as powerful computationally as systems that are only 2 years older, so the performance degradation introduced by the virtualization is easily offset with the performance improvement of the hardware.

4.2 Disk IO performance

Disk IO tends to be the slowest component of modern desktop computer systems. Most desktop systems will have just one hard disk and those with more normally do not have a large number of disks. Reading and writing to a local hard disk will be slow and is not recommended for tasks that require copying large files. In the case of distributing affective intelligent tutoring systems via virtual machine images, copying the disk images during deployment will be slow if copied to the local hard disk. It is recommended that in educational institution environments with Gigabit Ethernet connectivity to the desktop a storage area network (SAN) approach be adopted. SANs normally consist of multiple storage units that each contain a large number of fast hard disks that can be written to in parallel, significantly improving performance. Another area where disk IO performance will have an impact is in live migration of virtual machines [17, 18]. In a class room/ tutorial setting it may be appropriate for the affective intelligent tutor to follow the user as the user moves around the classroom for group work etc. When deployed on a virtual machine a live migration is the most appropriate method of moving the system, reducing down time which is critical in a classroom or tutorial situation.

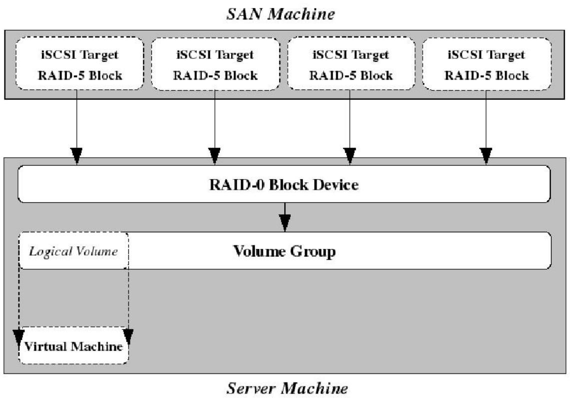

Figure 3. Test architecture for testing host and virtualization performance

Figure 3 shows the structure of the SAN used to test the IO performance for the host system and virtualized systems. The SANs export disk blocks that are then combined into a RAID block which can be exported to a virtual machine as a logical volume on a volume group. The RAID block can provide redundancy but we use it here to provide additional performance by striping writes across the disk block targets. Several different protocols exist for interacting with SANs include AoE (ATA over Ethernet) and iSCSI (internet Small Computer Systems Interface). Figures 4-5 show that both protocols have similar characteristics although AoE has some performance issues for character and block output over standard frame sizes which is not seen for jumbo frames (maximum transmission unit of 9000). For transfer of large files the block input and output are the key performance indicators as well as the seek performance (figure 5), seek performance will be particularly important when there are multiple users making use of the SANs which is the normal case in a corporate environment or an educational institution.

4.3 Virtual machine and Host system IO comparison

Table 2 shows that using the iSCSI protocol the block input on the host machine was 192 MB/s (MTU 1500) while the Virtual system could only sustain 50 MB/s. This represents a significant degradation in performance and shows that with the use of virtualized systems it may be more appropriate to transfer the disk image to the local system. However with the ease of management of a SAN based system, even this degraded performance is enough for the affective intelligent tutoring system, which once loaded does not need to read and write a large amount of data.

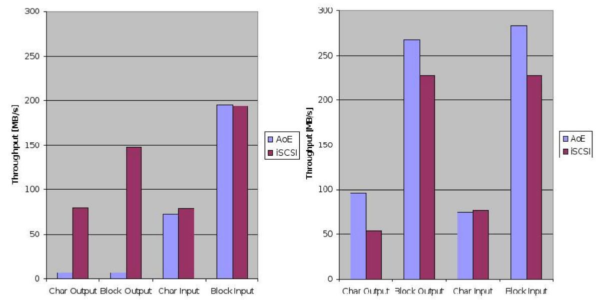

Figure 4 a. iSCSI vs AOE over SAN b. iSCSI vs AOE over SAN using jumbo frames

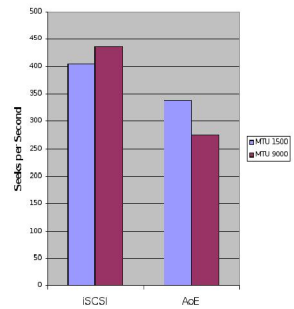

Figure 5. Seek performance of iSCSI vs AOE

4.4 USB IO performance

Standard USB 1.1 runs at 12Mb/s (and is backward compatible to 1.5Mb/s) while USB 2.0 supports up to 480Mp/s. USB 1.1 is fast enough for most applications such as reading and writing small files from USB memory as well as any external file backup system, however for applications such as high resolution video from a webcam (640x480 image size) and transferring large files USB 2.0 becomes essential. Accessing USB devises from a virtual machine adds additional overhead that reduces the effective throughput. Testing with a web camera streaming video (with an image size of 320x240) the virtualized system could sustain an average of 10 frames per second compared to the 15 frames per second of the host system. This degradation though significant is enough for the affective intelligent tutoring systems facial recognition system that does not require higher than 10 frames per second.

Table 2: Comparison of IO throughput and CPU efficiency between the virtual machine and host.

Machine	IO throughput	CPU usage	IO throughput per cpu util.
Host System	192 MB/s	14	14
Virtual System	50 MB/s	40	1.25

5. Conclusions

This paper has presented an operating system virtual machine based platform for the management and delivery of and affective intelligent tutoring based e-learning system. Although the OS virtual machine based system has significant overhead in terms of file and network IO we have shown that it is a viable platform for the delivery of client/ server or peer to peer based e-learning systems that would require significant configuration by the user if traditional approaches are adopted. In addition to hardware performance improvements, hardware support for virtualization is also progressing quickly meaning that over time the overhead associated with full virtualization and virtualized IO will be reduced.

6. References

[1] J. Hu, C. Meinel, M. Schmitt, "Tele-lab IT security: an architecture for interactive lessons for security education", *Proc. 35th SIGCSE tech. symp. on Comp. Sci. Edu.*, 2004.

[2] J. Hu, D. Cordel, and C. Meinel, "Virtual Laboratory for IT Security Education", *Proc. of the Conf. on Information Systems in E-Business and E-Government*, 2004.

[3] R. Keil-Slawik, T. Hampel, B. Essmann, B., "Re-conceptualizing learning environments: a framework for pervasive elearning", *Third IEEE Int. Conf. on Pervasive Computing and Comm. Workshops*, 2005, pp.322- 326.

[4] H Jin, Z Yin, X Yang, F Wang, J Ma, H Wang, J Yin, "APPLE: A Novel P2P based E-Learning Environment", *6th Int. Workshop on Distributed Computing*, 2004.

[5] Y. Lee, M. Junginger, J. Geller, "High Performance Publisher/Subscriber Communication For Adaptive, Collaborative Web-based Learning", *Journal of Distance Education Technologies*, 1(3), 2003, pp.14-27

[6] C.H. Messom, R.G. Craig, "Web Based Laboratory for Controlling Real Robot Systems", *The Proc. of the Biannual Conf. of the Distance Education Assoc. New Zealand*, 2002.

[7] G. Sen Gupta, A. Buhler, S. Demidenko, C.H. Messom, S.C. Mukhopadhyay, "A Multi-tiered Framework for Virtual Instrumentation System (VIS)", *IEEE Instrumentation and Measurement Technology Conference*, 2006, pp.976-981.

[8] R.G. Craig, C.H. Messom, "Replicating Reality - a better way to understanding", *Intl Conf on Computers in Education*, 2002, pp.1191-1195.

[9] J.N. Matthews, J.J. Herne, T.M. Deshane, P.A. Jablonski, L.R. Cherian, and M.T. McCabe, "Data Protection and Rapid Recovery From Attack With A Virtual Private File Server and Virtual Machine Appliances", *Proc. of the IASTED Int. Conf. on Communication, Network and Information Security*, 2005, pp.170–181.

[10] S. Santhanam, P. Elango, A. Arpaci-Dusseau, and M. Livny, "Deploying Virtual Machines as Sandboxes for the Grid", *2nd Workshop on Real, Large Distributed Sys.*, 2005.

[11] K. Fraser, S. Hand, R. Neugebauer, I. Pratt, A. Warfield, and M. Williamson, "Safe hardware access with the Xen virtual machine monitor", *1st Workshop on Op. Sys. & Arch. Support for the On-Demand IT Infrastructure*, 2004.

[12] S.T. King, P.M. Chen, Y-M. Wang, C. Verbowski, H.J. Wang and J.R. Lorch, "SubVirt: Implementing malware with virtual machines", *Proc. of the 2006 IEEE Symposium on Security and Privacy*, 2006, pp.314-327.

[13] S. Alexander, A. Sarrafzadeh, "Easy with Eve: A Functional Affective Tutoring System", *8th Int. Conf. on Intelligent Tutoring Systems*, 2006.

[14] L. Cherkasova and R. Gardner, "Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor", *Proc. of the USENIX Annual Tech. Conf*, 2005.

[15] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel, "Diagnosing Performance Overheads in the Xen Virtual Machine Environment", *First ACM/USENIX Conf. on Virtual Execution Environments*, 2005.

[16] 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. of the ACM Symp. on Operating System Principles*, 2003.

[17] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines", *Proc. of the 2nd ACM/USENIX Symp. on Networked Systems Design and Implementation*, 2005.

[18] F. Travostino, P. Daspit, L. Gommans, C. Jog, C. de Laat, J. Mambretti, I. Monga, B. van Oudenaarde, S. Raghunath, and P. Y. Wang, "Seamless live migration of virtual machines over the MAN/WAN", *Future Generation Computer Systems*, vol. 22, no. 8, 2006, pp.901–907.