A RESEARCH SURVEY ON VIRTUAL MACHINES FOR EFFICIENT COMPUTING IN OPERATING SYSTEMS

Y. Subba Reddy1,V Venkata Ramana2,G. Rama Subba Reddy3,Dr. Pandurangan Ravi4

1, 2, 3 Associate Professors, Department of CSE, CBIT, Proddatur, Y.S.R, A.P, (India)

4 Principal, CBIT, Proddatur, Y.S.R, A.P, (India)

ABSTRACT

Studies to virtualization old back to 1970's, while researches targeted on multiplexing solutions in the mainframe. Nevertheless, because of the fall of components value and as well the passing of modern multi-tasking program, virtualization researches fall to its value. Recently, the revitalization of virtualization technological innovation has customized Enterprises' IT technique from "one program one machine" to multiplexing applications/services into one device that is defined as Virtualization 1.0. Service, merging is an affiliate level on-going pattern and is actuated by higher protection and reliability that virtualization offered.This literary work study talks about the styles of growth of exclusive device, such as genuine virtualization, para-virtualization, and binary interpretation, pre-virtualization and hardware-assisted virtualization. Within the latter part, we often analyze the security and efficiency sides and difficulties introduced by a virtual machine monitor (VMM).

Keywords : VMM, OS, Vitualization

I. INTRODUCTION

The source of virtualization technological innovation old back to 19 70's [1], during which the Exclusive device Observe (VMM) came into being as a thin program system-abstraction part between components and program. VMM allowed dividing a components system into one or a lot of virtual machines during which unlimited in function methods (OS) might run on great of it as if it absolutely was running on great of local components. Although VMM has shown its advantage in multiplexing components among several programs, the growth of recent multi-tasking OS and also the fall of the components price [2] searched the world of VMM that was acquainted be run on CPUs.

In the 19 90's, Stanford School Scientists researched the prospective of virtual device to overcome the complications that in function methods, growth was not within the same speed with components growth, such as, extremely multiprocessing device (MPP). The inquiry discovered that researchers might make use of virtualization technology innovation and modify current in function methods to take pleasure from efficiency gain while not vital adjustment [6].

To boot, later studies found virtualization may well be employed in areas like great quality, protection and management disadvantage.

Nowadays, the blossom of VMM reveals a brand new chance, server/service merging, with the company. With the increasing great high quality of the program, in that some could be fragile and insecure, company assumes a "one-application-one-machine" viewpoint for tight support encapsulation to take guardianship of an efficient efficiency. Nevertheless, this can bring low usage rate, around 20%. VMM provides a vehicle for multiplexing, solitude, protection and migration.

Though the detailed delights illustrate a favorable probability of VMM, these enhanced efficiency and an options area unit descends from studies that generate over several component virtualization complications, together with CPU virtualization, Memory virtualization and I/O virtualization. Various researchers and VMM providers strike the matter by totally different methods. We are starting to speak about a variety of the methods, such as, genuine virtualization, para-virtualization, and binary interpretation, pre-virtualization and hardware-assisted virtualization. Within the later a part of the textbook, we are failing to talk about on the forthcoming complications, like prospective strike on virtual device, specifically virtual device dependent rootkit (VMBR) [19].

The rest of the paper is organized as follows. We usually protect linked work in section II. Observing this, we usually review the category of VMM and current VM products in section III. In section IV, we usually talk about the program and components, virtualization methods in VMM and the way the methods set out over an assortment of the efficiency expense. In IV, we usually talk around the protection issue in VMM and studies victimization VMM to increase security. Finally, we usually determine with conclusion in section VI.

II. RELATED WORK

This literary work study follows latest study documents, "Virtual Machine Monitors: Current Technology and Upcoming Trends" [2] in 2005 and "Survey of System Virtualization Technique" [5] in 2004. We'd require to upgrade the [2] and [5] on latest virtualization studies together with components progression, like Apple Natural Hill State [14], AMD SVM [15], and rule strategy, like Binary Interpretation [13] and Pre-virtualization [16].

III. BACKGROUND

Program virtualization permits several application systems (OS) to run on primary of a slice of components by components dividing or via the hypervisor, also called (Virtual Machine Observe, VMM). The components dividing strategy subdivide physical server introduction parts, each can run an OS. Hypervisor strategy places a portion of code between application and component system and permit several visitor OS to run on primary of it. The benefits over components dividing square measure resource powerful alliance and discussing among OS as hypervisor has full management over actual hardware.

A. PRODUCT AND PROJECT

In this section, we are mentioning the list of VMM projects

Name	Host CPU	Guest CPU	Host OS(s)	Guest OS(s)	Technique used	License
Bochs	Any, just need to be compiled	x86, x86-64	Windows, Windows Mobile, Linux, IRIX, AIX, FreeBSD, OpenBSD, BeOS, Mac OS X	DOS, Windows, *BSD, Linux	Emulation	LGPL
Zones	x86, x86-64, SPARC	x86	Solaris 10	Solaris (8, 9 or 10), Linux	OS-level virtualization	CDDL
Cooperative Linux	x86	x86	Windows 2000, XP, 2003, Vista[1]	Linux	Porting	GPL v2
DOSBox	any	x86	Linux, Windows, Mac OS X, BeOS, FreeBSD, OpenBSD, Solaris, QNX, IRIX, MorphOS, AmigaOS	Internally emulated DOS shell. Classic PC booter games and unofficially, Windows 1.0 to 3.11	Emulation	GPL
DOSEMU	x86, X86-64	x86	Linux	DOS	Emulation	GPL v2
GXemul	any	ARM, MIPS, M88K, PowerPC, SuperH	Unix-like	NetBSD, OpenBSD, Linux, Ultrix, Sprite	Emulation	BSD
Hercules	any	z/Architecture	Unix-like	Linux on zSeries, z/OS, z/VM, z/VSE, OS/360, DOS/360, DOS/VS, MVS, VM/370, TSS/370.	Emulation	QPL
Hyper-V	x64 + hardware-assisted virtualization (Intel VT or AMD-V)	x86, x86-64	Windows 2008 w/Hyper-V Role, Windows Hyper-V Server	Supported Drivers for Windows 2000, Windows 2003, Windows 2008, Windows XP, Windows Vista, Linux (SUSE 10 Released, More Announced))	Hypervisor Type I	Proprietary (free of charge with Windows Server 2008)
JPC (Virtual Machine)	Any running the Java Virtual Machine	x86	Java Virtual Machine	DOS	Emulation	GPL v2
KVM	x86 , x86-64, s390,PowerPC	same as host	Linux	Linux, Windows, FreeBSD, Solaris	Para-virtualization + In-kernel virtualization	GPL v2

LinuxOnLinux	Itanium	compatible	Linux	Linux	Paravirtulization + Hypervisor Type II	GPL
Linux-VServer	x86, X86-64, Alpha, PowerPC/64, PA-RISC/64, SPARC/64, ARM, S/390, SH/66, MIPS	compatible	Linux	Various Linux distributions	OS-level virtulization	GPL v2
Logical Domains	UltraSPARC T1, UltraSPARC T2	compatible	Solaris 10	Solaris, Linux (Ubuntu Server 7.10), and FreeBSD (experimental)	Paravirtualization	**?**
Mac-on-Linux	PowerPC	PowerPC	Linux	Mac OS X, Mac OS 7.5.2 to 9.2.2, Linux	Full-virtulization	GPL
Mac-on-Mac	PowerPC	PowerPC	Mac OS X	Mac OS X, Mac OS 7.5.2 to 9.2.2, Linux	Full-virtulization	GPL
OpenVZ	Intel x86, X86-64, IA-64, PowerPC64, SPARC/64	Same as host	Linux	Various Linux distributions	OS-level virtulization	GPL
Oracle VM	Intel x86, x86-64, Intel VT-x	Intel x86, x86-64, Intel VT-x	none (bare metal install)	Microsoft Windows, Oracle Enterprise Linux, Red Hat Enterprise Linux	Paravirtulization + HW virtulization	Free, Commercial
Parallels Desktop for Mac	Intel x86, Intel VT-x	Intel x86	Mac OS X (Intel)	Windows, Linux, FreeBSD, OS/2, eComStation, MS-DOS, Solaris	Hypervisor Type II	Proprietary
Parallels Workstation	x86, Intel VT-x	x86	Windows, Linux	Windows, Linux, FreeBSD, OS/2, eComStation, MS-DOS, Solaris	Hypervisor Type II	Proprietary
PowerVM	POWER4, POWER5, POWER6, PowerPC 970	POWER4, POWER5, POWER6, PowerPC 970, X86 (PowerVM-Lx86)	hardware / firmware, no host OS	Linux-POWERPC, Linux-X86, AIX, i5/OS, IBM i	Hypervisor Type II	Proprietary
QEMU	x86, X86-64, IA-64, PowerPC, Alpha, SPARC 32 and 64, ARM, S/390, M68k	x86, X86-64, ARM, SPARC 32 and 64, PowerPC, MIPS	Windows, Linux, Mac OS X, Solaris, FreeBSD, OpenBSD, BeOS	Changes regularly [6]	Emulation	GPL/LGPL
QEMU w/ kqemu module	Intel x86, X86-64	Same as host	Linux, FreeBSD, OpenBSD, Solaris, Windows	Changes regularly [7]	In-kernel virtualization	GPL/LGPL
QEMU w/ qvm86 module	x86	x86	Linux, NetBSD, Windows	Changes regularly	In-kernel virtualization	GPL
RTS Hypervisor	x86	x86	none: bare metal installation	Windows XP, XP-Embedded, Linux, VxWorks, Windows CE, ETS, OS-9 and	Hypervisor type I	Proprietary

				proprietary OS		
Sun xVM Server	x86-64, SPARC	(Same as host)	none (bare metal with Solaris as special guest providing backend drivers to guests)	Windows XP & 2003 Server (x86-64 only), Linux, Solaris	Paravirtulization + Porting + HW virtualization	GPL v3
User Mode Linux	x86, x86-64, PowerPC	(Same as host)	Linux	Linux	Porting	GPL v2
Sun xVM VirtualBox	x86, x86-64	x86, (x86-64 only on VirtualBox 2 with hardware virtualization)	Windows, Linux, Mac OS X (Intel), Solaris, eComStation	DOS, Windows, Linux, OS/2, FreeBSD, Solaris	Full-virtulization	GPL v2; full version
Virtual PC 2007	x86, x86-64	x86	Windows Vista (Business, Enterprise, Ultimate), XP Pro, XP Tablet PC Edition	DOS, Windows, OS/2, Linux(Suse, Xubuntu), OpenSolaris(Belenix)	Full-virtualization	Proprietary (free of charge) from Jul 2006)
Virtual PC 7 for Mac	PowerPC	x86	Mac OS X	Windows, OS/2, Linux	Full-virtualization	Proprietary
VMware ESX Server	X86, x86-64	x86, x86-64	none (bare metal install)	Windows, Linux, Netware, Solaris, FreeBSD, ...	Hypervisor Type I	Proprietary
VMware ESXi Server	x86, x86-64	x86, x86-64	none (bare metal install) (embedded)	Windows, Linux, Netware, Solaris, FreeBSD, ...	Hypervisor Type I	Proprietary (Free)
VMware Fusion	x86, x86-64	x86, x86-64	Mac OS X (Intel)	Windows, Linux, Netware, Solaris, others	Hypervisor Type II	Proprietary
VMware Server	x86, x86-64	x86, x86-64	Windows, Linux	DOS, Windows, Linux, FreeBSD, Netware, Solaris, Virtual appliances	Hypervisor Type II	Proprietary (Free)
VMware Workstation 6.0	x86, x86-64	x86, x86-64	Windows, Linux	DOS, Windows, Linux, FreeBSD, Netware, Solaris, Darwin, Virtual appliances	Hypervisor Type II	Proprietary
VMware Player 2.0	x86, x86-64	x86, x86-64	Windows, Linux	DOS, Windows, Linux, FreeBSD, Netware, Solaris, Darwin, Virtual appliances	Hypervisor Type II	Proprietary (Free)
Wind River hypervisor	x86, PowerPC	(Same as host)	bare metal	Linux, VxWorks, bare metal virtual board	Hypervisor Type I	Proprietary
Wind River VxWorks MILS Platform	PowerPC	(Same as host)	bare metal	VxWorks, bare metal virtual board	Paravirtualization	Proprietary
XEN	x86, X86-64, (PowerPC and IA-64 ports in progress)	(Same as host)	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP & 2003 Server (needs vers. 3.0 and an Intel VT (Vanderpool) or AMD-V (Pacifica)-capable CPU), Plan 9	Paravirtulization + Hypervisor Type II	GPL

z/VM	z/Architecture	z/Architecture (the new one)	with or without OS, with different deep	Linux on zSeries, z/OS, z/VSE, z/TPF, z/VM, VM/CMS, MUSIC/SP, OpenSolaris for System z, and predecessors	Full-virtualization + Hypervisor Type I	Proprietary One time Charge + Maintained support
LPARs	z/Architecture	z/Architecture	Intrinsic feature of System z mainframes	Linux on zSeries, z/OS, z/VSE, z/TPF, z/VM, VM/CMS, MUSIC/SP, and predecessors	HW Hypervisor	Intrinsic feature of System z mainframes

B. ADVANTAGES FROM VIRTUALIZATION

Uniform view of underlying hardware

VMM will offer a homogenous read of underlying hardware despite vendors and design by techniques like commercialism typical interface. Through VMM, hardware is viewed as a pool of resource, wherever directors don't give birth to be compelled to beware of the coupling of hardware and system computer code. Discotheque [10] paper has shown VMM will just about convert underlying large multiple process (MPP) machine to a traditional interface that goods OS will utilize.

Encapsulation

VMM will give finish encapsulation of exclusive device rule declares, so VM is revoked and set off again with none loss of land. Via VM revocation and recommencement, VM is moved to different actual device and still perform on the rainfly.

Easy to replicate

As VM will be considered as a conventional computer file once it's not corporal penalties, the high top quality of the VM will be really like a conventional computer file, wherever it will be simply tracked and passed on via totally different media. The top qualification provides comfort in putting in a stop (by stone current VM at a specific point) and rollback (by running stored VM at a selected checkpoint).

Another advantage is customer PC environment will be portable and executed on any VMM, so giving customer additional comfort.

Isolation

VMM will provide solitude of programs, wherever several programs is operating on several VMs on "one program on one VM" foundation. VMM provides the liability and protection needed to validate accident in any VM won't have an impact on substitute operating VMs. so programs is multiplexed in an extremely a lot of efficient way.

Utilization and cost saving

Device that scarifies efficient for higher usage or one program on one device to validate liability. VMM provides a mean to multiplex program with high liability and thence will increase server usage rate.

By combining several services into one device, the need for buying components is reduced and businesses will get pleasure from cost saving from reduced components cost and future management and energy invoice and therefore reduced Total Cost of ownership (TCO).

IV. IMPLEMENTATION ISSUES IN VMM

Though the advantages square measure wide seen in Section three, the implementation of VMM faces completely different challenges. To start out with, Popek and Goldberg [3] have expressed a virtual machine monitor have to be compelled to satisfy 3 essential characteristics.

Equivalence property

Any system that operates on great of VMM ought to execute indistinguishable as if the system is run straight on great of local components and therefore the system can have the freedom to having access to blessed guidelines that it plans. Equivalence property provides 2 exclusions, moment, that VMM involvement can extend system performance time and source availableness that system might have to be present at for distributed source to be out there.

Efficiency property

All simple guidelines ought to be asleep by the components straight, with no involvement in the least on the aspect of VMM. It's significant because the value of training emulation can carry

great expense from interpretation and introduction. If a larger aspect of guidelines got to be copied, the efficiency cannot coordinate to those running on local ingredients.

Resource control property

It is not possible for any system to deliver an impact on sources, e.g., storage and VMM has complete control over actual source.

In this section, we are starting to speak about several methods to obtain in section III qualities precise and their tradeoffs, together with efficiency, interface and convenience.

A. PURE VIRTUALIZATION

Some functions are designed with the intent to be virtualizable, as well as IBM/370 [32]. If the unique benefit method for VMM isn't reinforced in components, genuine virtualization usually de-privileges visitor OS to work on less blessed stage, whereas VMM is run in most benefit stage, as proven in determine three. As Popek and cartoonist requirement is met in these components, immediate efficiency has already protected the equivalence. Thus, unmodified OS will run on great of VMM as if it's working on unique components. Nevertheless, the efficiency, expense will be great thanks to change between beneficial ways.

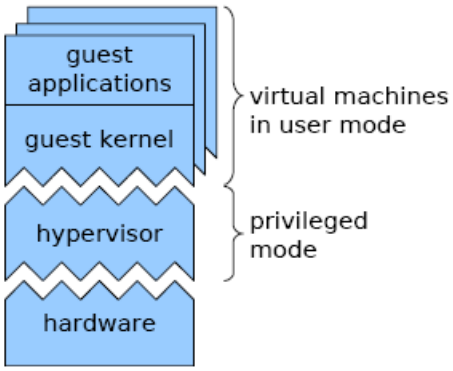

Figure 1: Pure Virtualization System

B. BINARY TRANSLATION

Binary Translation is applied in hypervisor together with VMWare EXS server. The idea behind Binary Translation [33], [13] is to interchange code that can't be directly dead, including, i) PC-relative addressing, ii) Direct management flow, iii) Indirect management flow, iv) Privileged directions. Underneath this theme, computer code blocks are translated on the fly into compiled code fragment (CCF). Performance potency is achieved as most of the code area unit safe and

may run identically, whereas privileged directions will be replaced by hyper call that avoid privileged instruction entice. Therefore VMM with Binary translation can run quick than classical VMM. Adaptive binary translation eliminates the amount of entice by following.

i) Assume instruction to be innocent till tried guilty

ii) During execution, VMM find instruction that entice oftentimes and adapt their translation

iii) Retranslate non-IDENT (identical) to avoid the trap

iv) Patch the first IDENT translation with a forwarding jump to the new translation

Thus entice are often avoided.

Binary Translation doesn't need guest OS kernel to be changed. At identical time, it maintains low overhead that matches Popek and Goldberg's potency property. However, as practicality required to be equivalent, the interpretation method is difficult.

C. HARDWARE ASSISTED VIRTUALIZATION

To facilitate the virtualization in processor, Intel and AMD have wanted to eliminate the requirement for watching and translation of directions. To the present finish, new directions are additional. A brand new management structure, Virtual Machine management Structure (VMCS) [14], has additionally created its look.

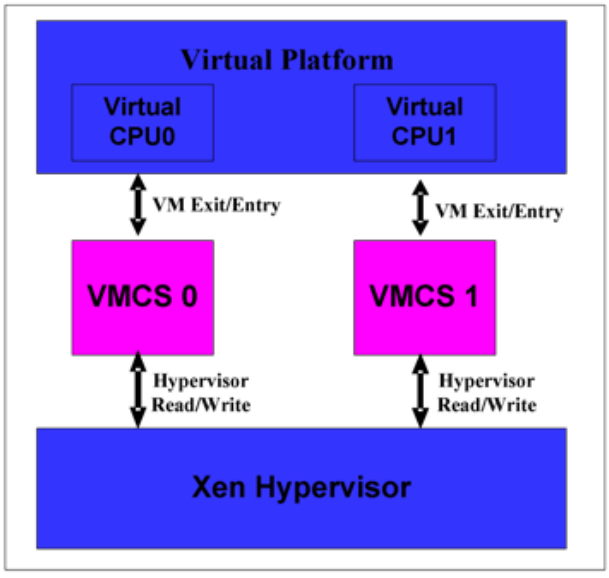

Figure 2: VMCS Representation in XEN

The program will gain a lot of management by running on the lower privilege level. On 32-bit x86 design, there are a 4 privilege levels (called rings), namely 0, 1, 2 and 3 (But, rings one and a pair of are detailed in x86-64 as a result of they not serve) The OS is meant to run on the ring 0, that has the very best level of management. Applications run in the ring, and that they can't modify their level of execution. And an application can't stop the OS whereas the OS could conceive to stop an application.

In a virtualized setting, we've one or many OS. The hypervisor that controls them needs a better privilege level than the OS as a result of it controls the physical resources and client OS. Because the hypervisor is that the ring zero, guest OS should be stationed along a better ring: 1 or 3. Still, the OS area unit designed to work on the ring 0. Additionally, work on a hoop below the applications ensures management of these applications.

To resolve this downside, an Intel American state technology introduced a replacement technique of execution, called VMX. It's a beginning stage. This level corresponds to what we have a tendency to assume rings that area unit beneath the ring zero, and a traditional level, appreciate the recent rings from 0 to 3. The hypervisor runs in VMX root level that induces a lot of management. The guest operating systems run on the ring zero, that area unit on VMX traditional level. Thus, this manner, VMM and guest OS work every on the privilege level that they're contrived to work along. Then no a lot of changes area unit required for the Guest OS, e.g. binary translation.

The new directions enable the processor to modify to a replacement mode of execution, guest mode execution. This mode has additionally four totally different stages of privileges. Thus, it deletes the issues related to the implementation of guest OS in a very totally different privilege level. Once it's required, the processor switches to the guest mode execution. To do so, the VMM ought to set the management bits that area unit accountable to change to the present mode, those its area unit keeps within the VMCS. At the time of shift between the OSs, the state of the guest OS is kept within the VMCS and also the state of the opposite guest OS is loaded from the VMCS.

Intel and AMD claim that this way they increase the focal ratio of the hypervisor. Nevertheless, trials conducted by VMware tend to indicate that the gain isn't thus obvious. The gain is way clearer with Parallels digital computer, explicit some testers. As an example, the quantity of cycles of calculation utilized by VM entry directions, VM exit, VMCS read, write VMCS has been divided by 2 between the Pentium four and Core two pair.

V. SECURITY CONCERN

While experts experiencing the advantages introduced by virtualization from multiplexing services, developing stop and rollback, powerful migration, pre-configured application

submission and several separated examining surroundings, scientists is constantly on the open up possibilities and difficulties in virtualization technology.

Opportunity

Researches together with [24], [37], [38], use virtual machines to hear instruction, such as, LiveWire [24]. LiveWire is associate degree Intrusion Detection System (IDS) that runs on prime of VMM as shown in figure 3. The analysis pinpoints implementing IDS with VMM has benefits over typical approaches that IDS resides in host system or network. The paper assure VMM to be troublesome to attack compare with trendy OS because of

i) Simple and unnatural Interface

ii) Simple Protection Mode

iii) Simple design and tiny code size

Thus, Virtual Machine contemplation (VMI) will offer higher isolation compare with host system approach in order that malware will hardly attack IDS. It additionally offers higher visibility compare with network approach in order that IDS will examine hardware and guest OS computer code state and event via VMM.

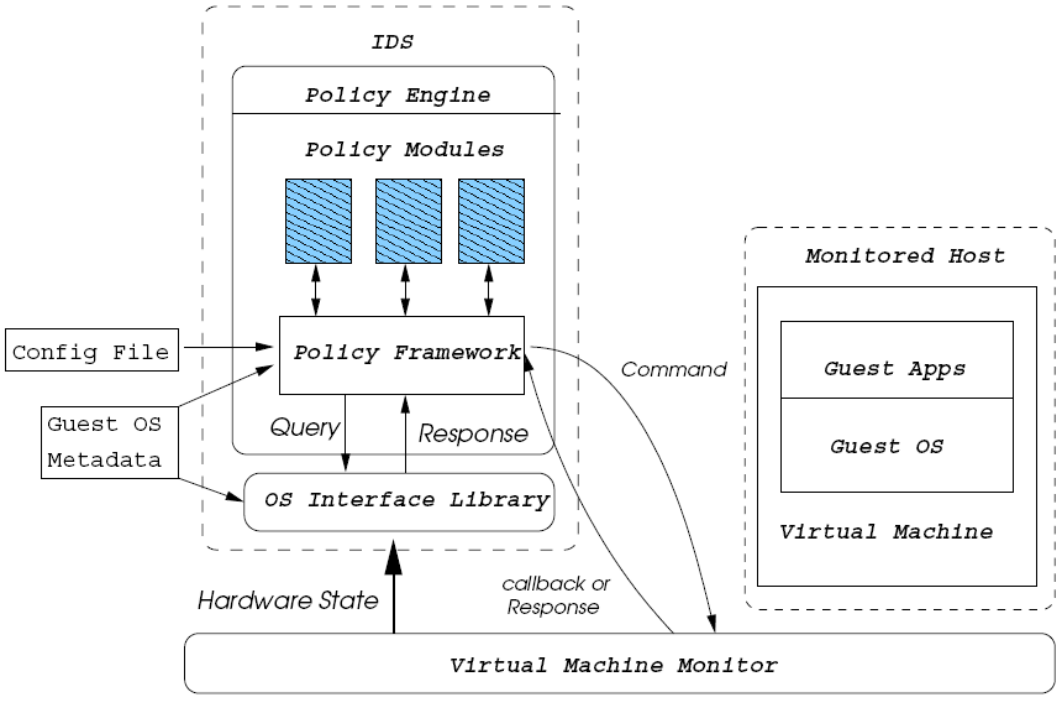

Figure 3: High-Level View of LiveWire VMI based IDS architecture.

In further to new style of IDS, users is benefitted from running totally different security level application in several VMs. as an example, user will use one separated VM for security group action, e.g. on-line banking, whereas another VM for net water sport. VMM isolation property secures the banking VM from web water sport VM even it's rattled by malware. Terra [25] has place one step any by implementing trusty VMM that gives 2 styles of VM, i) Open VM, performs like different VM, ii) Closed VM, offers further security measures as well as, root secure, attestation and trusty path.

VI. CONCLUSION

In this literature analysis, we've got said 2 survey papers in 1994 and 1995 [2] [5]. We have a tendency to extend the survey by that specialize in performance and security aspects of virtual machine monitor. The performance side leads United States of America to hide researches on computer code and hardware technique for virtualization and pinpoints however overhead is lowered by techniques together with Binary Translation [33] and Para-virtualization [11] [12] and newer hardware technology [13] [14] [15]. The protective side leads United States of America to researches on mistreatment VMM on intrusion detection [24] [37] [38] and secure computing [25]. Lastly, we have a tendency to conjointly explore potential attack and detection of virtual machine primarily based rootkits (VMBR) [26] [39] [40].

References

[1] R.P. Goldberg, “Survey of Virtual Machine Research”, 1974

[2] M Rosenblum, T Garfinkel, “Virtual Machine Monitors: Current Technology and Future Trends”- Computer, 2005 - ieeexplore.ieee.org

[3] Gerald J. Popek and Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. Communications of the ACM, 17(7):412–421, 1974.

[4] JE Smith, R Nair , “The architecture of virtual machines”, Computer, 2005 - ieeexplore.ieee.org

[5] R Rose, “Survey of system virtualization techniques”, Retrieved March, 2004 - robertwrose.com

[6] Poul-Henning Kamp Robert N. M. Watson, “Jails: Confining the omnipotent root”.

[7] Khairil Yusof ,“FreeBSD Jails: Lightweight Virtualization”.

[8] Octave Orgeron (Sun Microsystem) “An introduction to Logical Domains.”

[9] Menno Lageman, “Sun Client Solutions : Solaris Containers — What They Are and How to Use Them”

[10] E Bugnion, S Devine, K Govil, M Rosenblum , "Disco: Running commodity operating systems on scalable multiprocessors", ACM Transactions on Computer Systems, 1997 - portal.acm.org

[11] P Barham, B Dragovic, K Fraser, S Hand, T Harris , "Xen and the art of virtualization", ACM SIGOPS Operating Systems Review, 2003 - portal.acm.org

[12] Andrew Whitaker, Marianne Shaw, and Steven D. Gribble. Scale and performance in the denali isolation kernel. ACM SIGOPS Operating Systems Review, 36(SI):195–209, 2002.

[13] K Adams, O Agesen"A comparison of software and hardware techniques for x86 virtualization", 2006 - portal.acm.org

[14] Rich Uhlig, Gil Neiger, Dion Rodgers, Amy L. Santoni, Fernando C. M. Martins, Andrew V. Anderson, Steven M. Bennett, Alain Kagi, Felix H. Leung, and Larry Smith. Intel virtualization technology. IEEE Computer, 38(5):48–56, 2005.

[15] Advanced Micro Devices, Inc. AMD64 Virtualization Codenamed Pacifica Technology, Secure Virtual Machine Architecture Reference Manual. Austin, TX, USA, 2006.

[16] J LeVasseur, V Uhlig, B Leslie, M Chapman, G Heiser, "Pre-virtualization: uniting two worlds", Proceedings of the twentieth ACM symposium on Operating systems principles

[17] J LeVasseur, V Uhlig, B Leslie, M Chapman, G Heiser, "Pre-virtualization: software layering for virtual machine"

[18] Logical Domains 1.0.3 Admin Guide

[19] Ashley Saulsbury, "UltraSPARC Virtual Machine Specification"

[20] J. P. Casazza, M. Greenfield, and K. Shi, "Redefining Server Performance Characterization for Virtualization Benchmarking", Intel Technology Journal, 2006, 10(3):243-252.

[21] D. Gupta, R. Gardner, and L. Cherkasova, "XenMon: QoS monitoring and performance profiling tool", HP Labs Technical Report, HPL-2005-187, October 2005.

[22] Menon, JR Santos, Y Turner, GJ Janakiraman, Willy Zwaenepoel , "Diagnosing performance overheads in the xen virtual machine environment", usenix.org

[23] Diwaker Gupta (UC San Diego), Lucy Cherkasova (HP Labs), Rob Gardner (HP Labs), Amin Vahdat (UC San Diego), "Performance Isolation in XEN".

[24] T Garfinkel, M Rosenblum , "A virtual machine introspection based architecture for intrusion detection", Proc. Network and Distributed Systems Security Symposium, 2003 - eprints.kfupm.edu.sa

[25] T Garfinkel, B Pfaff, J Chow, M Rosenblum, D Boneh , "Terra: A virtual machine-based platform for trusted computing" - portal.acm.org

[26] ST King, PM Chen , "SubVirt: Implementing malware with virtual machines" - 2006 IEEE Symposium on Security and Privacy, 2006 - ieeexplore.ieee.org

[27] GW Dunlap, ST King, S Cinar, MA Basrai, PM Chen , "ReVirt: Enabling intrusion analysis through virtual-machine logging and replay",- ACM SIGOPS Operating Systems Review, 2002 - portal.acm.org

[28] C SE, Y RESPONSE , "A Testing Methodology for Rootkit Removal Effectiveness", symantec.com

[29] T Garfinkel, M Rosenblum, "When Virtual is harder than Real: Security Challenges in Virtual Machine Based Computing Environments"

[30] J. S. Robin, C. E. Irvine, "Analysis of the Intel Pentium Ability to support a Secure Virtual Machine Monitor", USENUX Security Symposium, pp 129-144, 2000

[31] R. Goldberg. Architectural Principles for Virtual Computer Systems. Ph.D. thesis, Harvard University, Cambridge, MA, 1972.

[32] R. J. Creasy, "The Origin of the Vm/370 Time-Sharing System", IBM Journal of Research and Development, vol 25, no 5, p 483, 1981

[33] R. Sites er al., "Binary Translation", Comm, ACM, Feb 1993, pp 69-81

[34] Tony Shoumack, "BEGINNERS GUIDE TO LDOMS: UNDERSTANDING AND DEPLOYING LOGICAL DOMAINS"

[35] T Garfinkel, K Adams, A Warfield, J Franklin , "Compatibility is not transparency: VMM detection myths and realities" - Proceedings of the 11th Workshop on Hot Topics in Operating System - usenix.org

[36] K Skapinetz, "Virtualisation as a blackhat tool", Network Security, Oct 2007

[37] KG Anagnostakis, S Sidiroglou, P Akritidis, K Xinidis, E Markatos, "Detecting targeted attacks using shadow honeypots", - usenix.org

[38] A Joshi, ST King, GW Dunlap, PM Chen , "Detecting past and present intrusions through vulnerability-specific predicates", - Proceedings of the twentieth ACM symposium on Operating System Principles, pp 91-104, 2005 - portal.acm.org

[39] Blue Pill Project, www.bluepillproject.org, Invisible Things Lab

[40] J. Rutkowska, "Red Pill or how to detect VMM using (almost) one CPU instruction", 2005

[41] P. Ferrie, N. Lawason, T. Ptacek, "Don't Tell Joanna, The Virtualised Rootkit Is Dead" In Proceedings of Black Hat USA 2007,

[42] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla. Pioneer: Verifying integrity and guaranteeing execution of code on legacy platforms. In Proceedings of ACM Symposium on Operating Systems Principles (SOSP), pages 1–16, 2005

[43] Y.-M. Wang, D. Beck, B. Vo, R. Roussev, and C. Verbowski. Detecting Stealth Software with Strider GhostBuster. In Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN), June 2005.

[44] M Carbone, D Zamboni, W Lee, "Taming Virtualization", - IEEE Security & Privacy, 2008 - ieeexplore.ieee.org