Nepal Journal of Science and Technology Vol. 15, No.2 (2014) 141-150

Virtualization Tools and Techniques: Survey

Anum Masood, Muhammad Sharif, Mussarat Yasmin and Mudassar Raza
Department of Computer Science, COMSATS Institute of Information Technology, Pakistan
e-mail:mussaratyasmin@comsats.edu.pk

Abstract
Virtualization is a technique in which the user required services run remotely in a ubiquitous environment which gives scalable resources. Virtualization is being used in cloud computing for load balancing and aggregation of cloud resources. Virtualization provides higher hardware utilization. It is also being used for partitioning of computational resources and hence supports sharing of resources. Virtualization has different types such as Native virtualization, Full virtualization, Operating system level virtualization and Para virtualization. Other than these there is Resources virtualization, Desktop virtualization, Server virtualization, Data centres virtualization and application virtualization. The resources virtualization is implemented in different forms such as the Full virtualization, Native virtualization, Para virtualization, Operating system (OS) layer virtualization or Hosted virtualization. Virtual machines and Virtual machine monitors (VMMs) have been developed to offer better energy efficient solutions to the virtualization problems. Virtualization tools like OpenVz, Xen, VmWare etc. are widely used in the computing industry.

Key words: cloud computing, resources virtualization, server virtualization, virtualization

Introduction
Virtualization technique is used in cloud computing as it provides virtualized resources in ubiquitous computing cloud (Hewitt 2008). By service virtualization the servers are combined together for the energy efficiency in data centres. For the partitioning of computational resources and hardware sharing virtualization technique is widely used (Hlavacs *et al.* 2008, Koomey 2006, Sharif *et al.* 2012). Virtualization is used at application level and also on the server level (Berl & de Meer 2009, View 2007).

Servers consume 70% of power even at low utilization. In such cases virtualization plays an important role in minimizing the power consumption (Berl *et al.* 2009). Virtual machines (VM) are used for the energy efficiency in computing devices because VM can be executed on just one hardware platform and hence lessens the overall hardware requirement. This in turn reduces the energy needed to cool the hardware when it is excessively used (Rivoire *et al.* 2007).

Virtualization at system level means that the virtual machines (VMs) are used for the virtualization of resources like memory, processors, storage and the peripheral input output devices.

The concept of virtual machine was first presented in mid of 1960s. The initial virtual machine was designed for hardware platform of IBM CP-40 and IBM M44/44X. Early uses of virtualization were consolidation of servers, data recovery and testing of operating system kernel (Popek & Goldberg 1974, Rosenblum & Garfinkel 2005). Virtualization is most commonly used when there is the requirement of running different operating systems (OS) on a single computer hardware resource.

A virtual machine monitor (VMM) also known as hypervisor is basically software which gives abstraction of the virtual machine (VM) to guest operating systems executing under the virtual machine monitor.

In the same way, VMM guarantees the system availability even if one guest operating system domain fails. VMM makes virtualization possible as it helps in loading multiple operating systems simultaneously (Rose 2004).

VMM or hypervisor is used for handling resources which are being shared among the operating systems and also helps in holding the incoming interrupt requests. A VMM is preferred as any operating system may run on a given hardware platform. without any modification (Chen & Noble 2001).

Not all processors are virtualized as some have unprivileged instructions. In such cases VMM is used as it handles these instructions by dynamic recompilation technique which finds unprivileged instructions on runtime and traps them into VMM (Waldspurger 2002). This method is called full virtualization. Full Virtualization does not require operating system source code changes. When the guest operating system source code is to be modified by VMM routines then it is called Para virtualization (Rosenblum & Garfinkel 2005).

There are other forms of virtualization like native virtualization, operating system level virtualization etc. (King *et al.* 2003).

Methodology

Virtualization is a technique which is used in the data centre environment for energy efficiency. The general structure of virtualization is in layers. These layers are independent from each other (Crosby & Brown 2006).

Virtualization is also a method of abstraction of applications and the supporting hardware for these applications (Menascé 2005). Mainly there are six layers namely access layer, application layer, processing layer, storage layer, network layer, management layer and security layer (Dobrilovic & Stojanov 2006).

The main domains which come under virtualization effects are people working in data centres managing team, experts interacting over a distance and processes related to the configuration management and are converted into virtual processes. These processes can run on the same hardware platform irrespective of their own design requirement (Dan 2007).

Figure 1 shows the various layers of virtualization.

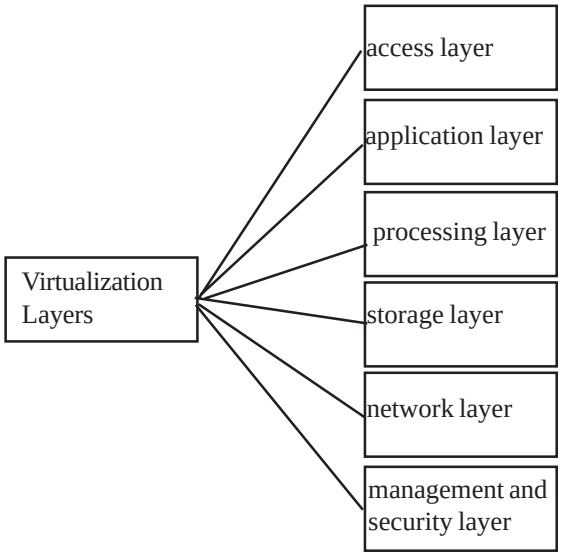

Fig.1. Layers of Virtualization (Ferstay 2006)

There are two implications of virtualization. One is by the separation of page tables of operating system kernel and user processes (Leslie *et al.* 2005). Page Table separation makes sure that memory is protected. In case of single space address method the address spaces of Linux processes and L4 tasks including Linux servers do not overlap, so they do not enter memory space of each other. Therefore, the memory is protected by the separation of address spaces (van Schaik & Heiser 2007). There are seven operating modes, six of which are privileged modes of 3ARM and only one unprivileged mode. The 4ARM11and MP Core have physically indexed physically tagged cache and virtually indexed physically tagged cache respectively (Wiggins *et al.* 2003). For processes the size of address space is 32MB only which is undesirable. Another issue is that the operating systems do not have single address space so they have to be modified (ARM 2003). For ARM most common virtual machine monitors are Virtual Logix and Trango.

Mainly two types of architectures are available for the virtualization software. They are bare metal and hosted architectures. There are some commonalities between these two architectures such as they can both be used in test applications as well as in engineering applications which are completely understood before designing (Engler & Kaashoek 1995).

Model virtualization is a new concept in which the model for virtualization is composed keeping in mind the required properties.

Basic virtualization model properties are given below:

1. Interoperability
   The virtual models are considered as normal models. These virtual models can be exchanged between various standard tools of modelling.
2. Synchronization
   The contributing models as well as the virtual models have same element instances by the help of sharing so the updates are automatically transferred from contributing model to virtual model and vice versa.
3. Faster creation time
   No copying or backup of information is needed and as the element instances are synchronized so they can be executed simultaneously.
4. Less memory usage
   Virtual models do not contain actual data, so no extra memory is used for storage which minimizes the memory usage (Nejabati *et al.* 2011, Uddin & Rahman 2012).

   Few of the virtualization types are briefly explained below.

Operating System Virtualization

Operating system virtualization has become the main component of IT infrastructure because of the reason that the end users interact with this type of virtualization (Bavier *et al.* 2004). Operating system virtualization is found in commonly used operating systems such as Red Hat Enterprise Linux, Windows Vista or Windows 7. These operating systems can run on single hardware platform simultaneously (King *et al.* 2005, Kolyshkin 2006).

Application Virtualization

Application virtualization resembles the concept of thin client (Uhlig *et al.* 2005). One of the examples of application virtualization is a situation in which an individual is using Microsoft Office on his PC while the running state of the application as well as the personal data are both being managed and stored on virtual software (Devine *et al.* 2002).

The PC computer provides the hardware support (RAM, main memory and CPU processing) for the Microsoft Office application but the application is not actually stored on the hardware and files are not saved on the secondary storage i.e., hard disk of the PC (Devine *et al.* 2002). Application virtualization differs depending on the running application on the local computer and management application logic which is remotely running on a virtual computer (Daniels 2009, Matthews *et al.* 2007).

Storage Virtualization

The concept of virtualization is also introduced in the storage technologies. The storage virtualization can be classified into two main categories: Block virtualization and File virtualization (Vaughan-Nichols 2006). Block virtualization is known by Storage Area Network (SAN) and Network Attached Storage (NAS) technologies. These are the distributed storage networks which appear to be a single physical device instead of multiple devices (Feldhaus *et al.* 2012).

Hardware Virtualization

Hardware virtualization based on Intel VT as well as the AMD-V technologies have been introduced as virtualization in the x86 processor architecture. Virtualization hardware support decreased the need of Para virtualization of guest operating systems i.e., not to completely change the operating system. Xen vendor Virtual Iron announced that Xen infrastructure supports only full virtualization using AMD-V and Intel VT processors but do not support Para virtualization (Stahl & Anand 2009).

Multiprocessor Virtualization

This type of virtualization is used when the host platform is multiprocessor with shared memory. The main aspect is partitioning of the system into numerous multiprocessors. It is done by distributing resources of the system. There are two types of partitions: physical and logical.

In *physical partitioning*, the physical resources used by a single virtual system are put out of place from other resources that are being used by other virtual systems (Barham *et al.* 2003). This means that there is a degree of isolation such that neither software nor hardware problems of one partitioned virtual system affect the programs in other virtual systems (Soltesz *et al.* 2007).

In *logical partitioning*, the hardware resources are time multiplexed among the partitions. This in turn improves the resource utilization of the system but the consequences are that few of the advantages of isolation of hardware are lost (Soltesz *et al.* 2007). Both logical and physical partitioning techniques use special software support which is based on hardware modifications mostly related to partitioning (Mergen *et al.* 2006).

Whole System Virtual Machines

In virtual machines, the guests and the host system software use same ISA which is also being used by the underlying hardware. There are few cases in which the guest and the host systems do not use the same ISA like the two commonly used desktop computers Windows PC and Apple PowerPC based systems use diverse ISAs (and various operating systems) (Habib 2008).

Whole system virtual machines deal with different ISAs by the virtualization of software which comprises of operating system and applications. As the ISAs are different, the virtual machine must emulate both the operating system and application code. Such type of virtual machine is Virtual PC, 8 in which a Windows operating system runs on a Macintosh platform (Adams & Agesen 2006, Bugnion *et al.* 1997).

Hosted Virtualization

The hosted virtualization uses operating system to manage the resources within the emulation layer which allows the VMs to run with applications in the same operating system. It is called the type-2 virtualization (Heiser 2008). Virtual machine is the physical machine's abstraction of the CPU, memory management, I/O etc. in which the guest operating system is virtualized and guest boots (Menon *et al.* 2006). The hosted virtualization manages physical resources of hardware platform. Virtual machine is implemented by the hosted virtualization layer within a regular operating system through the VMM (Smith & Nair 2005). Operating system manages memory allocation, resources and the CPU processing (Anderson *et al.* 2005, Hwang *et al.* 2008).

Operating System Level Virtualization

Operating system virtualization is different from full and Para virtualization in a sense that virtualization does not depend on the hypervisor (Virtual machine) in this type of virtualization. Instead of using the virtual machine monitor, the operating system is updated to isolate different instances of operating system within any system hardware of a single host machine while the guest portion of operating system is known as virtual private servers (VPS) (Swift *et al.* 2003). All the virtual private servers share a single kernel (Egi *et al.* 2007). The limitation is that if the kernel crashes then all the virtual private servers are crashed. The only benefit in using the single kernel is that the resources consumption is less as compared to the resources for multiple kernels (Chaudhary *et al.* 2008).

Native Virtualization

The native virtualization controls the hardware platform for virtualization within CPU itself to help in the virtualization of hardware. Native virtualization allows multiple operating systems to run on a single hardware (host processor) with any modification (Border 2007). The processor is not emulated by this type of native virtualization. It is different from full virtualization technique because in full virtualization it is possible that the operating system may execute on a virtual processor but with poor performance. In x86 there are a total of 64 series processors which are of two types: Intel and AMD (Nagarajan *et al.* 2007). These processors support virtualization by the help of virtualization extensions such as the AMD-V and Intel-VT. The x86's 64 processors helping the virtualization process is quite recent technique but it is becoming common (Armstrong *et al.* 2005, Matthews *et al.* 2007).

Virtualization Tools

Virtualization tools have immensely been the topic of research as they help in achieving virtualization in many types of hardware. The commonly used tools of virtualization are mainly related to the VMware products which are VMware Vcenter Converter and VMware Workstation (Hayden 2004). Few of them are open source tools i.e., they are freely available for anyone except the VMware Server which is free but not open source (the code in not accessible for all). This is due to the reason that it has two hardware platforms i.e., Linux and Windows (Gavrilovska *et al.* 2007, Ray & Schultz 2009) while virtualization tools such as the Microsoft Virtual PC or any other tool is mostly limited to only one operating system or hardware platform which means that they are restricted only to their own types of softwares. It is important to

consider that there are some similarities between the real environment and virtual environment which indirectly depend on the technique used for virtualization (Chowdhury&Boutaba 2010). Some of the popular virtualization tools used in various computing fields is as follows:

1. Virtual Network User Mode Linux (VNUML)
VNUML (Barham *et al.* 2003) is an open source and is available to all the users for free download. VNUML is basically a virtualization tool used for multiple virtual systems of Linux operating system. These virtual systems are known as guests which run their applications along with Linux operating system of the original system which is refer to as host.

2. Virtual Box
Virtual Box is used for implementation of virtual machines on the physical computers and servers. It also does full virtualization in the host computer which means that without any modification in the operating system the guest operating system is executed on the host computer (Geiselhart *et al.* 2003).

3. VMware Server
It is a source free virtualization tool for Linux as well as Windows operating system (Cox 2007). VMware Server is based on the full virtualization i.e., the physical desktop computer to run more than one virtual machine of varying operating system called guests on it.

4. Qemu
Qemu is used for execution of virtualization in the operating systems like Linux and Windows both. It is a popular open source (R.&M. 2007) emulator that provides fast emulation by the help of dynamic translation. It has many useful commands for the management of VM.

5. Xen
Xen is also an open source tool for virtualization used widely for Para virtualization in the host PC and guest computers (Bavier *et al.* 2006).

6. VMware
VMware is a VM (virtual machine) platform which helps in execution of unmodified operating system on the host or user level application hardware . Operating system which is being executed with VMware may get crashed, reinstalled, rebooted or crashed without any effect on the application running on the host computer.

VMware gives the separation of guest operating system from the real host operating system so that if the guest operating system fails then the physical hardware or the host machine does not suffer from the consequences (Fuertes & de Vergara 2007).

VMware is used to produce an illusion of standard Personal Computer hardware inside the virtual machine. Therefore the VMware is used to execute several unmodified operating systems at the same time on the single hardware machine by executing the operating system in the virtual machine of specific operating system.

Instead of indirect running of code on the hardware as in the case of software simulator, virtual machine executes the code directly on the physical hardware without any application for the interpretation of code.

7. EMF Tool
EMF virtualization tool is an eclipse based plug in on EMF basis to hold the transparent usage of virtual models all of which are based on EMF. For the creation of a virtual model using the EMF tool, the users have to provide contributing models along with Meta models for the virtualization. Following three elements are the basics of any virtual model formed by EMF tool.

1. Composition Meta model
It is used for the specification of virtual model concepts. The user may define it or it can be the amalgamation of various separate composition processes.

2. Correspondence Model
It is mostly defined along with the AMW^2 tool. This correspondence model contains all virtual links which are related in the contributing elements and identify in which manner they are to be composed.

3. Virtual Model
It is a file which specifies the physical location of all hardware resources which are to be used in the virtual composition process.

8. Virtual EMF
Virtual EMF is virtualization model composition tool. The specification of this tool is that it allows overcoming the limitations of virtual models such as the virtual models are unable to support concrete data although they are easily accessed. They help in manipulating the original data contained in other models of EMF, so this tool is also built on Eclipse/EMF1. Table 1 gives a comparison of virtualization tools on the basis of different factors.

Table1. Comparison of virtualization tools based on different factors

Virtualization Tool	Availability	Purpose	Mode of Virtualization
VMWare	Commercial	Gives better product for managing virtual infrastructure.	Full Virtualization
Xen	Open Source	For virtual machine (VM) migration.	Para Virtualization
QEMU	Open Source	For heterogeneous range of hardware architectures. May be used as emulator.	Native Virtualization
VNUML (Virtual Network User Mode Linux)	Open Source	Multiple virtual Linux operating systems (guests) to be executed as applications within a normal Linux operating system (host).	Full Virtualization
UML (User Mode Linux)	Open Source	For Linux system support.	Para Virtualization
Virtual Box	Commercial	Commercial version to support remote desktop protocol.	Native Virtualization
VMware Workstation	Open Source	Run under open source operating systems (OS).	Full Virtualization
VMware Vcenter Converter	Open Source	Run under open source operating systems (OS).	Full Virtualization
OpenVZ	Open Source	For partitioning of resources efficiently. Container Based.	Operating System Level Virtualization
Bochs	Open Source	For guest operating system debugging.	Emulator
VMware Server	Currently free (not open source)	Runs both on Windows and Linux platforms.	Full Virtualization
Microsoft Virtual PC	Commercial	Virtualization tool limited to Microsoft softwares.	Full Virtualization
Parallels	Commercial	It is a Machine Virtualization Tool which can execute on more than one instances of another operating system on standard hardware platform.	Full Virtualization
KVM (Kernel-based Virtual Machine)	Open Source	For Linux servers. CPU support for virtualization	Full Virtualization
VServer	Open Source	For isolating multiple Linux server environments driven by the single hardware platform.	Para Virtualization
LXC (Linux Container)	Open Source	For Linux hardware machine. Container Based.	Para Virtualization

Results and Discussion

Virtualization technique is used in cloud computing as it provides virtualized resources in ubiquitous computing cloud. By using virtualization the servers are combined together for the energy efficiency in data centres. For the partition of computational resources and hardware sharing virtualization technique is widely used. Virtualization is used at application level as well as on server level. Virtualization is also a method of abstraction of applications and the supporting hardware for these applications. There are many types of virtualization; few of them are discussed briefly in this survey paper.

Virtualization tools are a wide topic of research since they are used in the virtualization of hardware machines. The commonly used tools of virtualization are mainly related to the VMware products. Few of them are open source tools i.e., they are freely available for anyone except the VMware Server which is free but not open source. On the other hand, virtualization tools are limited to only one type of operating system or hardware platform which means that they are restricted only to their own types of softwares. Others are used for multiple operating systems and hardware machines. It is important to mention that some similarities exist between the real environment and virtual environment having indirect effect on the technique utilized for virtualization. A comparison of these tools is done in this paper to help in understanding the aspects on the basis of which various virtualization tools differ from each other.

References

KVM (Kernel-based Virtual Machine). Retrieved 19th May, 2013, 2013, from www.linux-kvm.org/ý.

LXC Linux Container. Retrieved JAN 2014, 2014, from www.lxc.sourceforge.net/.

Microsoft Windows Virtual PC. Retrieved JA 2014, 2014, from http://www.microsoft.com/windows/virtual-pc/.OpenVZ Wiki. Retrieved JAN 2014, 2014, from http://wiki.openvz.org/Main_Page/.

Parallels. Retrieved JAN 2014, 2014, from www.parallels.com/.

The User-Mode Linux Kernel. Retrieved JAN 2014, 2014, from http://user-mode-linux.sourceforge.net/.

Virtual Box. Retrieved JAN 2014, 2014, from http://www.virtualbox.org/.

VMwareServer. Retrieved JAN 2014, 2014, from http://www.vmware.com/products/server/.

VServer. Retrieved JAN 2014, 2014, from www.linux-vserver.org/.

IBM Project Big Green." Retrieved Jan 23, 2014, 2014, from http://www-03.ibm.com/press/us/en/photo/21514.wss.

Adams, K. and O. Agesen. 2006. A comparison of software and hardware techniques for x86 virtualization. In: *ACM SIGOPS Operating Systems Review*, pp. 2-13.

Anderson, T., L. Peterson, S. Shenker and J. Turner. 2005. Overcoming the Internet impasse through virtualization. *Computer* **38:** 34-41.

ARM, A. E.-S. 2003. *Technical Reference Manual, 2001-2003, ARM Limited*.

Armstrong, W. J., R. L. Arndt, D. C. Boutcher, R. G. Kovacs, D. Larson, K. A. Lucke, N. Nayar and R. Swanberg. 2005. Advanced virtualization capabilities of POWER5 systems. *IBM Journal of Research and Development* **49:** 523-532.

Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield. 2003. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review* **37:** 164-177.

Bavier, A., N. Feamster, M. Huang, L. Peterson and J. Rexford. 2006. In VINI veritas: realistic and controlled network experimentation. In: *ACM SIGCOMM Computer Communication Review*, pp. 3-14.

Bavier, A. C., M. Bowman, B. N. Chun, D. E. Culler, S. Karlin, S. Muir, L. L. Peterson, T. Roscoe, T. Spalink and M. Wawrzoniak. 2004. Operating Systems Support for Planetary-Scale Network Services. In: *NSDI*, pp. 19-19.

Berl, A. and H. de Meer. 2009. An energy-efficient distributed office environment. In: *Emerging Network Intelligence, 2009 First International Conference on*, pp. 117-122.

Berl, A., R. Weidlich, M. Schrank, H. Hlavacs and H. De Meer. 2009. Network virtualization in future home environments. Integrated Management of Systems, Services, Processes and People in IT.In *Springer* 177-190.

Border, C. 2007. The development and deployment of a multi-user, remote access virtualization system for networking, security, and system administration classes. In: *ACM SIGCSE Bulletin*, pp. 576-580.

Bugnion, E., S. Devine, K. Govil and M. Rosenblum. 1997. Disco: Running commodity operating systems on scalable multiprocessors. *ACM Transactions on Computer Systems (TOCS)* **15:** 412-447.

Chaudhary, V., M. Cha, J. Walters, S. Guercio and S. Gallo. 2008. A comparison of virtualization technologies for HPC. In: *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pp. 861-868.

Chen, P. M. and B. D. Noble. 2001. When virtual is better than real [operating system relocation to virtual machines]. In:*Proceedings of the Eighth Workshop on Hot Topics in Operating Systems*. pp. 133-138.

Chowdhury, N. and R. Boutaba. 2010. A survey of network virtualization. *Computer Networks* **54:** 862-876.

Cox, T. E. N. A. L. 2007. Towards an Operating Platform for Network Control Management. In: *Workshop on Programmable Routers for the Extensive Services of Tomorrow (PRESTO)*, , Princeton University, Princeton, NJ pp.

Crosby, S. and D. Brown. 2006. The virtualization reality. *Queue* **4:** 34-41.

Dan, K. 2007. "Virtualization is More than Virtual Machine Software." from http://www.kusnetzky.net/publications/ImpactPapers/20070829_Virtualization_is_more_than_VM.pdf.

Daniels, J. 2009. Server virtualization architecture and implementation. *Crossroads* **16:** 8-12.

Devine, S. W., E. Bugnion and M. Rosenblum. 2002. *Virtualization system including a virtual machine monitor for a computer with a segmented architecture*, *Google Patents*.

Dobrilovic, D. and Z. Stojanov. 2006. Using virtualization software in operating systems course. In: *International Conferenceon Information Technology: Research and Education, ITRE'06*. pp. 222-226.

Egi, N., A. Greenhalgh, M. Handley, M. Hoerdt, L. Mathy and T. Schooley. 2007. Evaluating xen for router virtualization. In: *Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN 2007*. pp. 1256-1261.

Engler, D. R. and M. F. Kaashoek. 1995. Exokernel: An operating system architecture for application-level resource management. In: *ACM SIGOPS Operating Systems Review*, pp. 251-266.

Feldhaus, F., S. Freitag and C. El Amrani. 2012. State-of-the-Art Technologies for Large-Scale Computing. *Ch* **1:** 1-17.

Ferstay, D. R. 2006. *Fast secure virtualization for the arm platform*, *The University of British Columbia*.

Fuertes, W. M. and J. E. L. de Vergara. 2007. A quantitative comparison of virtual network environments based on performance measurements. In: *Proceedings of the 14th HP Software University Association Workshop, Garching, Munich, Germany*, pp. 8-11.

Gavrilovska, A., S. Kumar, H. Raj, K. Schwan, V. Gupta, R. Nathuji, R. Niranjan, A. Ranadive and P. Saraiya. 2007. High-performance hypervisor architectures: Virtualization in hpc systems. In: *Workshop on System-level Virtualization for HPC (HPCVirt)*.

http://www.csm.ornl.gov/srt/hpcvirt07/papers/paper10.pdf

Geiselhart, G., L. Dupin, D. George, R. van der Heij, J. Langer, G. Norris, D. Robbins, B. Robinson, G. Sansoni and S. Thoss. 2003. Linux on IBM Eserver zSeries and S/390.

Habib, I. 2008. Virtualization with kvm. *Linux Journal* **2008:** 8.

Hayden, M. G. 2004. *Distributed network storage system with virtualization, Google Patents.*

Heiser, G. 2008. The role of virtualization in embedded systems. In: *Proceedings of the 1st workshop on Isolation and integration in embedded systems*, pp. 11-16.

Hewitt, C. 2008. ORGs for scalable, robust, privacy-friendly client cloud computing. *Internet Computing, IEEE* **12:** 96-99.

Hlavacs, H., R. Weidlich, K. A. Hummel, A. M. Houyou, A. Berl and H. De Meer. 2008. Distributed energy efficiency in future home environments. *annals of telecommunications-annales des télécommunications* **63:** 473-485.

Hwang, J.-Y., S.-B. Suh, S.-K. Heo, C.-J. Park, J.-M. Ryu, S.-Y. Park and C.-R. Kim. 2008. Xen on ARM: System virtualization using Xen hypervisor for ARM-based secure mobile phones. In: *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pp. 257-261.

King, S. T., G. W. Dunlap and P. M. Chen. 2003. Operating System Support for Virtual Machines. In: *USENIX Annual Technical Conference, General Track*, pp. 71-84.

King, S. T., G. W. Dunlap and P. M. Chen. 2005. Debugging operating systems with time-traveling virtual machines. In: *Proceedings of the annual conference on USENIX Annual Technical Conference*, pp. 1-1.

Kolyshkin, K. 2006. Virtualization in linux. *White paper, OpenVZ*

Koomey, J. 2006. Energystar, server energy measurement protocol, version 1.0. In: *Following energy efficiency server benchmark technical workshop, Santa Clara, CA*, pp.

Leslie, B., C. van Schaik and G. Heiser. 2005. Wombat: A portable user-mode Linux for embedded systems. In: *Proceedings of the 6th Linux. Conf. Au, Canberra*, pp.

Matthews, J. N., W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, M. McCabe and J. Owens. 2007. Quantifying the performance isolation properties of virtualization systems. In: *Proceedings of the 2007 workshop on Experimental computer science*, pp. 6.

Menascé, D. A. 2005. Virtualization: Concepts, applications, and performance modeling. In: *Int. CMG Conference*, pp. 407-414.

Menon, A., A. L. Cox and W. Zwaenepoel. 2006. Optimizing network virtualization in Xen. In: *USENIX Annual Technical Conference*, pp. 15-28.

Mergen, M. F., V. Uhlig, O. Krieger and J. Xenidis. 2006. Virtualization for high-performance computing. *ACM SIGOPS Operating Systems Review* **40:** 8-11.

Nagarajan, A. B., F. Mueller, C. Engelmann and S. L. Scott. 2007. Proactive fault tolerance for HPC with Xen virtualization. In: *Proceedings of the 21st annual international conference on Supercomputing*, pp. 23-32.

Nejabati, R., E. Escalona, S. Peng and D. Simeonidou. 2011. Optical network virtualization. In: *Optical Network Design and Modeling (ONDM), 2011 15th International Conference on*, pp. 1-5.

Popek, G. J. and R. P. Goldberg. 1974. Formal requirements for virtualizable third generation architectures. *Communications of the ACM* **17:** 412-421.

R., P. and C. M. 2007. *Case Study: Nationwide Uses Linux and High-Power Virtualization for Web Presence*. Gartner RAS Core Research Note G-00148213.

Ray, E. and E. Schultz. 2009. Virtualization security. In: *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, pp. 42.

Rivoire, S., M. A. Shah, P. Ranganathan and C. Kozyrakis. 2007. JouleSort: a balanced energy-efficiency benchmark. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 365-376.

Rose, R. 2004. Survey of system virtualization techniques. *Collections*

Rosenblum, M. and T. Garfinkel. 2005. Virtual machine monitors: Current technology and future trends. *Computer* **38:** 39-47.

Sharif, M., N. M. Butt, M. Raza and M. Arshad. 2012. Distributed Virtual Disk Storage System. *Control Theory and Informatics* **2:** 17-23.

Smith, J. E. and R. Nair. 2005. The architecture of virtual machines. *Computer* **38:** 32-38.

Soltesz, S., H. Pötzl, M. E. Fiuczynski, A. Bavier and L. Peterson. 2007. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In: *ACM SIGOPS Operating Systems Review*, pp. 275-287.

Stahl, E. and M. Anand. 2009. *A comparison of PowerVM and x86-based virtualization performance, Technical Report WP101574, 2009. URL: ftp:// dispsd-40-www3. boulder. ibm. com/common/ssi/sa/ wh/n/pow03029usen/POW03029USEN. PDF*.

Swift, M. M., B. N. Bershad and H. M. Levy. 2003. Improving the reliability of commodity operating systems. *ACM SIGOPS Operating Systems Review* **37:** 207-222.

Uddin, M. and A. A. Rahman. 2012. Virtualization implementation model for cost effective & efficient data centers. *arXiv preprint arXiv:1206.0988*

Uhlig, R., G. Neiger, D. Rodgers, A. L. Santoni, F. C. Martins, A. V. Anderson, S. M. Bennett, A. Kagi, F. H. Leung and L. Smith. 2005. Intel virtualization technology. *Computer* **38:** 48-56.

van Schaik, C. and G. Heiser. 2007. High-performance microkernels and virtualisation on ARM and segmented architectures. In: *Proceedings of the 1st International Workshop on Microkernels for Embedded Systems, Sydney, Australia*, pp.

Vaughan-Nichols, S. J. 2006. New approach to virtualization is a lightweight. *Computer* **39:** 12-14.

View, I. V. 2007. "Virtualization can help power efficiency." from http://www.03.ibm.com/systems/ virtualization/view/011607.html.

Waldspurger, C. A. 2002. Memory resource management in VMware ESX server. *ACM SIGOPS Operating Systems Review* **36:** 181-194.

Wiggins, A., H. Tuch, V. Uhlig and G. Heiser. 2003. Implementation of fast address-space switching and TLB sharing on the StrongARM processor. In: *Proceedings of 8th Asia Pacific Computer Systems Architecture Conference*. Japan, Springer Verlag. pp. 352-364.