The CSI Journal on
Computer Science and Engineering
Vol. 12, No. 2 & 4 (b), 2014
Pages 29-35
Regular Paper

Performance Evaluation of Hypervisors using HPC Challenge Benchmark Suite

Reza Bakhshayeshi[1] Mohammad Kazem Akbari[2] Morteza Sargolzaei Javan[2]

[1]Faculty of Engineering, Islamic Azad University, Garmsar, Iran
[2]Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Abstract

In recent years, rapid improvement of cloud computing convinced everyone accept it as a powerful substitution. Nowadays, more and more organizations are getting used to create private clouds, on the other hand, public clouds must be robust enough to handle scientific computing and computing sensitive requests of users in an efficient and cost effective manner. Apart from all these necessities, it is intellectual to improve the cloud infrastructure performance by appropriate choices. In this paper we are going to evaluate the performance of various hypervisors, including VMware ESXi, KVM, Xen, Oracle VirtualBox, and VMware Workstation using HPC Challenge (HPCC) benchmark suite and Open MPI in order to represent solutions for virtualization layer of cloud computing architecture and clusters.

Keywords: Cloud Computing, Virtualization, Hypervisor, Performance Evaluation.

1. Introduction

With the fast improvement of processing and storage technologies alongside the success of the internet, computing resources have become cheaper and more commonly available than ever before. These technological trends lead to a new computational model called cloud computing, in which resources like processors and storage spaces could be allocated to users or de-allocated from theme easily with help of virtualization technology through the internet in an on demand way [1]. A precise definition of cloud computing which we believe covers all major aspects of a cloud is given by The National Institute of Standards and Technology (NIST) [2]: “Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

In General, cloud computing architecture composes of four layers [1]: hardware layer, infrastructure layer, platform layer, and application layer. Figure 1 depict the layered model of cloud computing architecture. The hardware layer deals with physical aspects of the cloud which are implemented in datacenters. The platform layer consists of operating systems and application frameworks which provides APIs for web applications. The application layer consists of cloud applications.

The Infrastructure layer, also known as virtualization layer, creates a pool of storage computing resources by partitioning the physical resources using virtualization technologies (hypervisors) like VMware ESXi, Xen, and KVM. Virtualization layer actually is the heart of a cloud infrastructure, since many key features of cloud like dynamic resource allocation, which leads to tremendous cost savings, are only available through virtualization technologies. In this paper we are going to evaluate the performance of some commonly used and preferably open source virtualization technologies (hypervisors), including VMware ESXi [3], KVM [4], Xen [5], Oracle Virtual Box [6], and VMware Workstation [7] with various measurements and metrics like processing power, memory updates capability and bandwidth, network bandwidth and latency using HPC Challenge (HPCC) [8] benchmark suite and Open MPI [9]. There are however, numerous other virtualization technologies also available, including Microsoft Hyper-V [10], Open VZ [11] and Oracle VM [12].

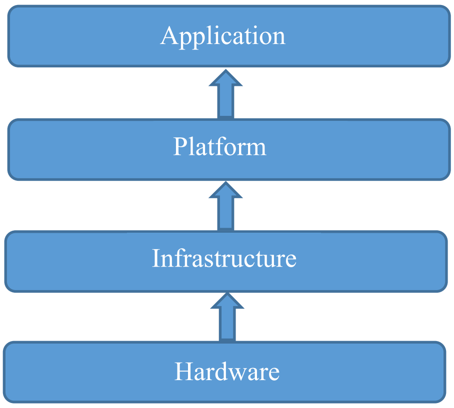

Figure 1. Cloud computing layered architecture

There have been previous works in this area, only to mention a few: V. Chaudhary et al. [13] evaluated the performance of VMware Server, Xen and Open VZ on x86 Fedora Core 5 with NAS Parallel Benchmarks (NPB) [14] for HPC and IOZone and Netperf to measure I/O in the absence of MPI benchmarks in order to gain insight into the potential performance bottlenecks that may impact distributed computations. They have evaluated the performance of single server and cluster of servers using the MPI implementation of NPB. They found that Open VZ is best for file systems. On serial NAS, most are close to native, and some even ran faster. For Open MP runs, Xen and Open VZ are close to real hardware, but VMware has large overhead. Also they have concluded Xen is best in networking (similar to our results).

Ying-Chuan Chen et al. [15] used HPC Challenge and Net PIPE [16] benchmarks to evaluate Xen full and partial virtualized and KVM. They found that performance of Xen PV platform is better than other investigated virtual platforms through FV of KVM and Xen. They suggested to build cluster or cloud based on PV technology of Xen. However, there is a troublesome disadvantage that the PV mechanism cannot support user to construct different virtual OS and Guest OS on host machine.

Young et al. [17] analyzed the performance of Xen, KVM and Virtual Box using HPC Challenge benchmark, and they have suggested KVM as a preferable hypervisor. Luszczek et al. [18] evaluated the performance of HPC Challenge benchmark in several virtual environments, including VMware, KVM and Virtual Box with purpose of evaluating the overheads of the different aspects of the system affected by virtualization.

The rest of this paper is organized as follows. Section 2 discusses hypervisors we have used. In section 3 we describe the method we have used to evaluate the performance of virtualized clusters. Section 4 illustrates the results and discusses them. Finally the paper concludes in section 5.

2. Virtualization and Hypervisors

Virtualization refers to the technique or approach of running many smaller virtual machines (VMs) on a computer, each having their own operating system and configuration. A hypervisor, or virtual machine monitor (VMM) is a program that runs on a host machine and creates VMs and allocates resources such as processor and memory to them. Hypervisor abstracts VMs from each other. In cloud computing architecture, allocating or de-allocating resources dynamically to users is only available with help of virtualization technologies.

Hypervisors are divided into two categories [19]:

1. Bare-metal (type 1) in which virtual machines directly run on host machine's hardware. KVM, Xen and VMware ESXi are examples of open source or free implementation of this type.

2. Hosted (type 2) in which virtual machines run on host's operating system. Oracle Virtual Box and VMware Workstation are examples of open source and commercial implementation of this type.

Figure 2 shows a general scheme of bare-metal and hosted hypervisor.

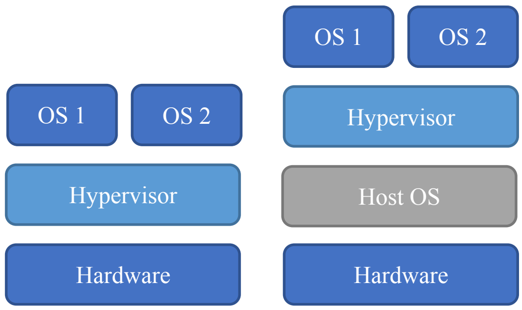

Figure 2. While bare-metal hypervisor runs directly on machine's hardware (left), hosted hypervisor runs on host's operating systems (right)

In the following we are going to present a brief definition of each bare-metal and hosted hypervisor which have been used in our experiments.

The Kernel-based Virtual Machine, or KVM, is a Linux subsystem which leverages virtualization extensions to add a virtual machine monitor (or hypervisor) capability to Linux [4].

Xen is a virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality. This is achieved by providing an idealized virtual machine abstraction [5].

VMware ESXi is the next-generation hypervisor, providing a new foundation for virtual infrastructure. ESXi operates independently from any general-purpose operating system, offering improved security, increased reliability, and simplified management. The compact architecture is designed for integration directly into virtualization-optimized server hardware, enabling rapid installation, configuration, and deployment [3].

Oracle Virtual Box is a cross-platform virtualization application. For one thing, it installs on existing Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. Secondly, it extends the capabilities of existing computer so that it can run multiple operating systems (inside multiple virtual machines) at the same time [6].

VMware Workstation functions as a computer within a computer (hosted), so that you can startup an entire operating system and run any programs for that operating system with it, while keeping your original operating system environment intact (and usable) [7].

3. Performance Evaluation

In this section we evaluate the performance of virtualized environments created with either VMware ESXi 5, KVM 1.0, Xen 4.1 or as bare metal hypervisors and Oracle Virtual Box 4.2 and VMware Workstation 9 as hosted hypervisors for comparison. We use The HPC Challenge benchmark (HPCC) [8] scientific computing benchmark suite. The HPC Challenge benchmark consists of basically 7 tests. The characteristics of used benchmarks are summarized in Table 1.

Table 1. Benchmarks used for virtualized environment performance evaluation. B, FLOP, U and PS stand for bytes, floating point operations, updates, and per second, respectively

Benchmark	Target	Unit
HPL [20]	CPU performance	GFLOPS
DGEMM [21]	CPU performance	GFLOPS
STREAM [22]	Memory bandwidth	GB/s
PTRANS [23]	CPUs communication	GB/s
Random Access [23]	Memory updates	MUPS
FFT [23]	CPU performance	GFLOPS
Effective Bandwidth b_{eff} (bw., lat.) [24, 25, 26]	Communication	GB/s, μs

We perform our experiments on homogenous virtualized environments build from Ubuntu Server 12.04 LTS image for single and multiple (2 and 4) instance(s). Each instance has 2 CPU cores and 1 GB of RAM. Our host operating system is Ubuntu Desktop 12.04 LTS when it is needed. Host machine has Intel Core i7 processor with 8 GB of RAM. We benefit from OpenMPI-1.6.4 [9] which is an open source MPI-2 implementation for parallel experiments. The benchmarks were compiled using GNU C/C++ 4.1.

Performance results of HPL benchmark depend on two factors: the Basic Linear Algebra Subprograms (BLAS) [27] library, and the problem size. We use ATLAS [28] library in our experiments. The ATLAS (Automatically Tuned Linear Algebra Software) project is an ongoing research effort focusing on applying empirical techniques in order to provide portable performance. At present, it provides C and Fortran77 interfaces to a portably efficient BLAS implementation, as well as a few routines from LAPACK [29]. For problem size we get benefit of the following recommended equation:

$$N = \sqrt{TM/8} \times 0.85 \quad (1)$$

where N is the problem size and TM is the total memory, including all nodes, in Byte. It reserves 15% of memory for operating system, which seems sufficient for Ubuntu Server 12.04 LTS.

4. Results

Each experiment executed 10 times so in total we had 150 executions ($5_{\text{hypervisors}} \times 3_{\text{typesofcluster}} \times 10_{\text{times}}$). Here the results of 105 experiments ($5_{\text{hypervisors}} \times 3_{\text{typesofcluster}} \times 7_{\text{benchmarks}}$) are going to present.

Figure 3 shows the HPL results. In general VMware ESXi with cluster of two nodes had the best results with 29.94 GFLOPS, on the other hand, Xen with 8.14 GFLOPS could reach only 27.93% of ESXi. Having two VMs for ESXi, KVM and Oracle Virtual Box had better results for them. Mainly there was a performance loss when number of instances rises, but this is expected behavior when new instances created on the same machine and as a result generate processing overhead on that machine.

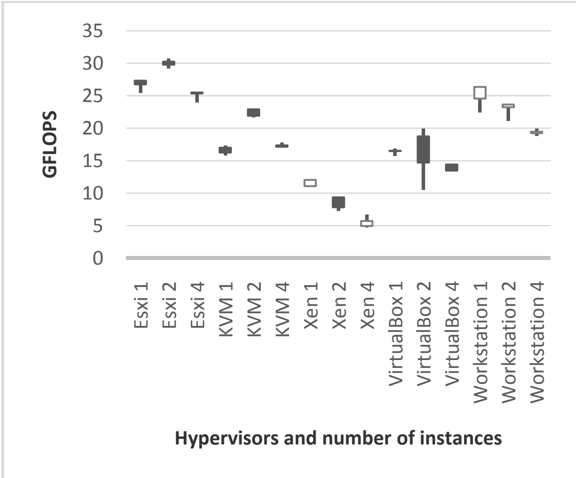

Figure 3. Comparison of HPL benchmark

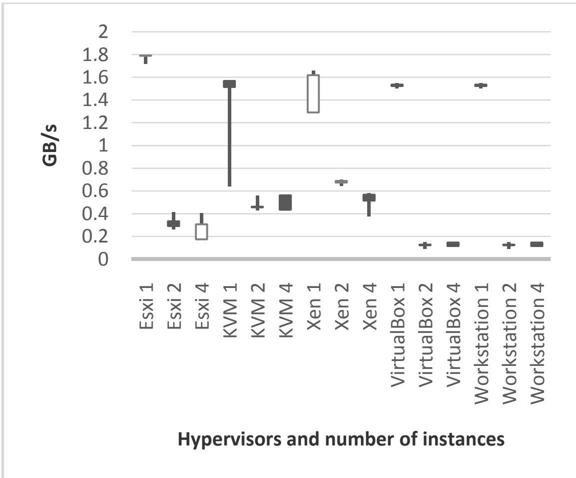

Figure 4. Comparison of PTRANS benchmark

Comparison of PTRANS benchmark is depicted in figure 4 Obviously adding more VMs decrease the gained bandwidth from two simultaneously working processors. So, having one VM will always reach the best result. VMware ESXi reached the highest with 1.78 GB/s. However other hypervisors also obtained near results to VMware ESXi in single instance case.

Figure 5 illustrates Random Access benchmark results. VMware ESXi with single VM reached the highest memory updates ratio with 16.55 MUPS. Oracle Virtual Box did not perfume well enough. It could reach only 0.41 MUPS in one VM case, which is only 2.47% of ESXi.

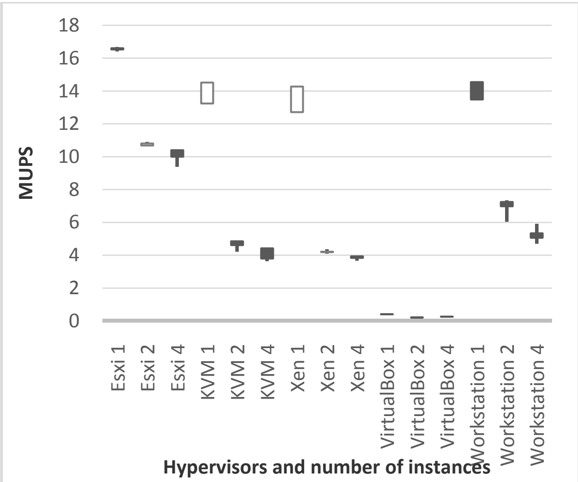

Figure 5. Comparison of Random Access benchmark

As shown in figure 6, VMware ESXi, KVM, Oracle Virtual Box and VMware Workstation all having close results for one VM at a time for FFT benchmark. They have all obtained FFT performance between 2 and 2.3 GFLOPS for single instance with little variations.

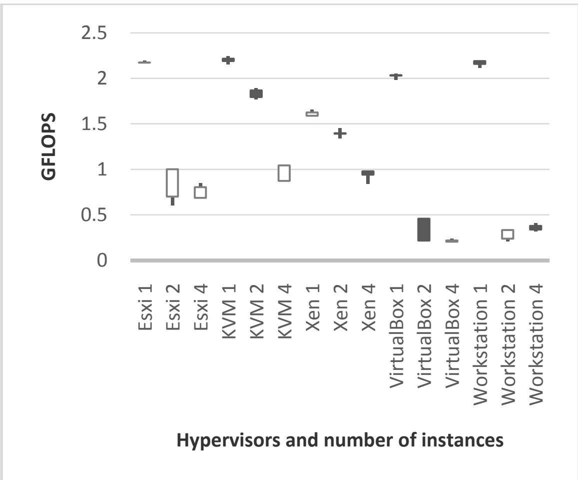

Figure 6. Comparison of FFT benchmark

For STREAM benchmark which is shown in figure 7 we obtained nearly same results for all hypervisors around 7 GB/s.

About DGEMM benchmark, as depicted in figure 8 both VMware ESXi and Workstation reached the highest performance with 15.6 and 15.11 GFLOPS respectively when there is one VM. As it can be seen, adding more VMs decreased the DGEMM performance for all hypervisors.

Network performance, including bandwidth and latency, as depicted in figure 9 and figure 10, evaluated using Effective Bandwidth (b_{eff}) benchmark. KVM and Xen shown better bandwidth performance than others with 0.35 and 0.29 GB/sin order, while ESXi could only reach 0.09 GB/s, which is only 25.71% of KVM. As we can guess, KVM and Xen performed better in latency and had lower time with 50.72 and 53.29 µs respectively. Note that running Effective Bandwidth (b_{eff}) benchmark on a single machine represents memory bandwidth and latency.

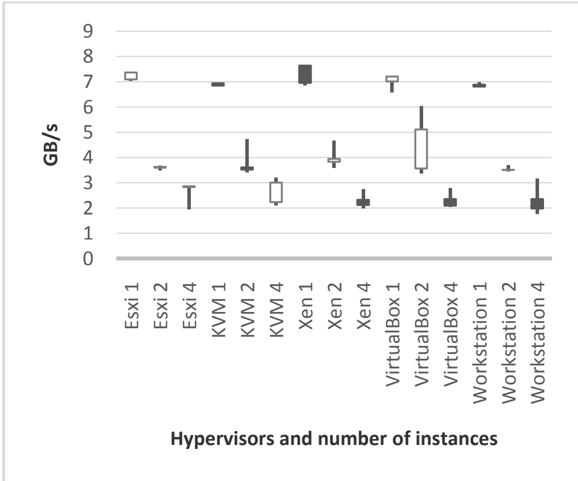

Figure 7. Comparison of *STREAM* benchmark

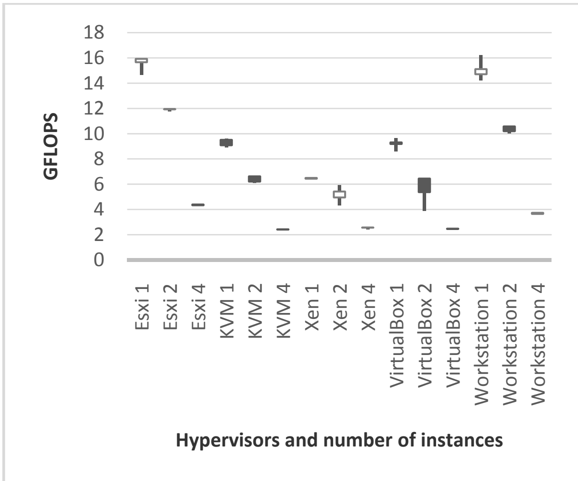

Figure 8. Comparison of *DGEMM* benchmark

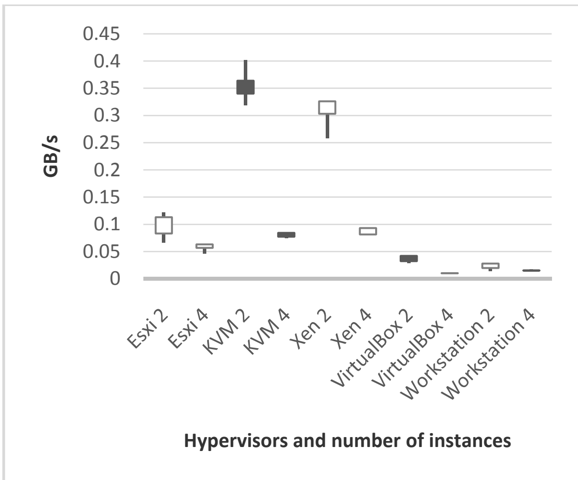

Figure 9. Comparison of bandwidth

Table 2 summarize the best achievements of hypervisors for each benchmark. Percentage difference calculated from average experiments results of best hypervisor and total average of other hypervisors in that specific benchmark.

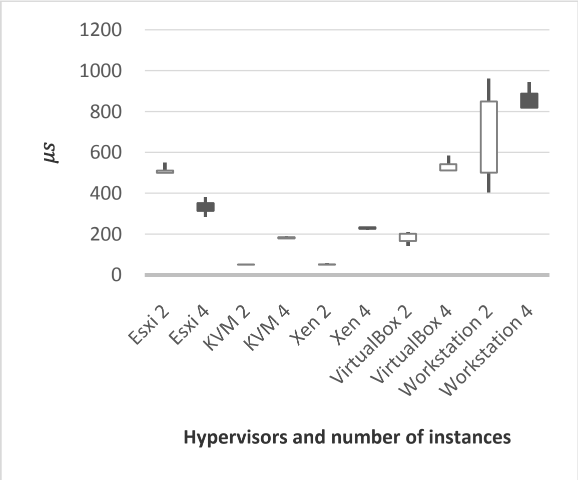

Figure 10. Comparison of latency

As we can see, in general, VMware ESXi obtained best results, thus it can be a reliable solution for cloud infrastructure. However, when the workload is network bounded, KVM or Xen would be the preferable choice. This fact is also depicted in figure 11 using Analytic Hierarchy Process approach when considering benchmarks equal in pairwise comparisons. AHP is a powerful decision making method in multi-criteria problems, it simplifies the problem into hierarchy through forming the comparison matrix to judge the weight [30]. This hierarchy made with help of Make It Rational AHP Software [31]. As we mentioned before, Effective Bandwidth (b_{eff}) benchmark refers to memory bandwidth and latency when it is running on a single instance, apart from that, it is meaningless to measure the network in a single VM, thus we omitted the Effective Bandwidth (b_{eff}) benchmark results in order to compare all cases. Since nature of benchmark designs are different, between HPL, DGEMM and FFT we obtained different results.

Table 2. Best achievements of hypervisors for each benchmark. Hypervisor with highest score highlighed italic in similar cases

Benchmark	Best Achievements	Percentage Difference
HPL [16]	VMware ESXi	52.14
DGEMM [17]	VMware ESXi and Workstation	79.1
STREAM [18]	All	-
PTRANS [19]	VMware ESXi	93.81
Random Access [19]	VMware ESXi	91.9
FFT [19]	VMware ESXi and Workstation, KVM, and Virtual Box	63.82
Effective Bandwidth b_{eff} (bw., lat.) [20, 21, 22]	KVM and Xen	128.46, 155.77

5. Conclusion

In recent years, rapid improvement of cloud computing convinced everyone accept it as a powerful substitution. Nowadays, more and more organizations are getting used to create private clouds, on the other hand, public clouds must be robust enough to handle scientific computing and computing sensitive requests of users in an efficient and cost effective manner. Apart from all these necessities, it is intellectual to improve the cloud infrastructure performance by appropriate choices. As we mentioned before, cloud computing layered architecture consist of virtualization layer (also called infrastructure layer). In this paper we compared some commonly used and preferably open source virtualization technologies, including VMware ESXi, KVM, Xen, Oracle Virtual Box, and VMware Workstation with various measurements and metrics like processing power, memory updates capability and bandwidth, network bandwidth and latency using HPC Challenge (HPCC) benchmark suite and Open MPI. Afterwards, we specified the best solution from the average of 10 times execution for each one.

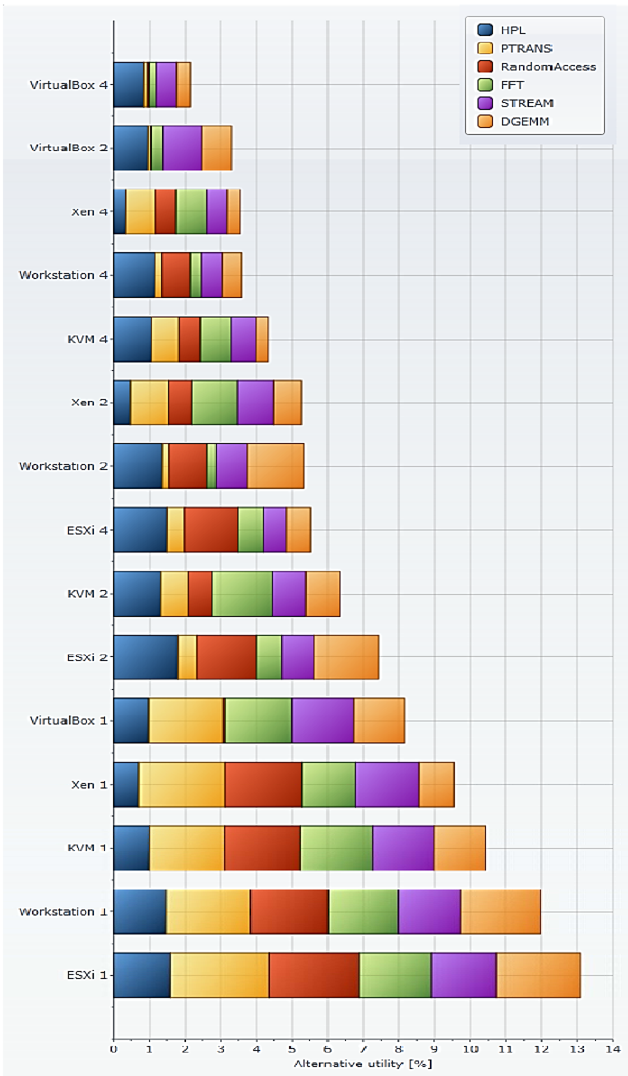

Figure 11. Benchmarks comparisons using Analytic Hierarchy Process when considering benchmarks equal in pairwise comparisons. ESXi gained highest total score

As we concluded in previous section, in general, VMware ESXi obtained the best results, thus it can be a reliable solution for cloud or cluster infrastructure in industry. However, when the workload is network bounded, KVM or Xen would be the preferable choice. Yet we need to investigate the results for CPU and memory bounded workloads as further studies.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-art and Research Challenges," *Journal of Internet Services and Applications,* vol. 1, no. 2, pp. 7-18, 2010.

[2] P. Mell, and T. Grance, "The NIST Definition of Cloud Computing," *NIST Special Publication,* vol. 80, no. 8, pp. 70-100, 2011.

[3] C. Chaubal, *The Architecture of Vmware Esxi*, VMware White Paper, 2008.

[4] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "KVM: The Linux Virtual Machine Monitor," *Proc.IEEE Intl Symp. on Linux*, pp. 225-230, 2007.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, and et. al., "Xen and The Art of Virtualization," *ACM SIGOPS Operating Systems Review,* vol. 37, no. 3, pp. 164-177, 2003.

[6] J. Watson, "Virtualbox: Bits and Bytes Masquerading as Machines," *Linux Journal*, vol. 2, no. 1, pp. 14-28, 2008.

[7] B. Ward, *The book of VMware: The Complete Guide to VMware Workstation*: No Starch Press, 2002.

[8] P. R. Luszczek, D. H. Bailey, J. J. Dongarra, J. Kepner, R. F. Lucas, R. Rabenseifner, and et. al., "The HPC Challenge (HPCC) Benchmark Suite," *Proc. ACM/IEEE Intl Conf. on Supercomputing*, pp. 213-219, 2006.

[9] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, and et. al., *Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation*, Recent Advances in Parallel Virtual Machine and Message Passing Interface, Springer, 2004.

[10] A. Velte, and T. Velte, *Microsoft Virtualization with Hyper-V*: McGraw-Hill, Inc., 2009.

[11] J. N. Matthews, W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, and et. al., "Quantifying the Performance Isolation Properties of Virtualization Systems," *Proc.IEEE Intl Workshop on Experimental Computer Science*, pp. 62-64, 2007.

[12] VM Oracle, *VirtualBox User Manual*, Oracle Corporation, 2011.

[13] V. Chaudhary, M. Cha, J. Walters, S. Guercio, and S. Gallo, "A Comparison of Virtualization Technologies for HPC," *Proc. IEEE Intl Conf. on Advanced Information Networking and Applications*, pp. 861-868, 2008.

[14] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, and et. al., "The Nas Parallel Benchmarks Summary and Preliminary Results," *Proc. ACM/IEEE Intl Conf. on Supercomputing*,pp. 158-165, 1991.

[15] Y. C. Chen, S. T. Wang, H. Y. Chang, T. M. Chen, and C. H. Li, "The Analysis for Virtualization Performance in Cluster and Cloud Computing," *Proc. IEEEIntl Conf. on Parallel and Distributed Processing Techniques and Applications*, pp. 352-358, 2011.

[16] Q. O. Snell, A. R. Mikler, and J. L. Gustafson, "Netpipe: A Network Protocol Independent Performance Evaluator," *Proc. IEEE Intl Conf. on Intelligent Information Management and Systems*, pp. 64-70, 1996.

[17] A. J. Younge, R. Henschel, J. T. Brown, G. von Laszewski, J. Qiu, and G. C. Fox, "Analysis of Virtualization Technologies for High Performance Computing Environments," *Proc. IEEE Intl Conf. on Cloud Computing*, pp. 9-16, 2011.

[18] P. Luszczek, J. J. Dongarra, D. Koester, R. Rabenseifner, B. Lucas, J. Kepner, and et. al., "Introduction to the HPC Challenge Benchmark Suite," *Lawrence Berkeley National Laboratory*, 2005.

[19] G. J. Popek, and R. P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures," *Communication Journal of ACM,* vol. 17, no. 2, pp. 412-421, 1974.

[20] A. Petitet, "HPL: APortable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers," *http://www.netlib.org/benchmark/hpl/,* 2004.

[21] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, "A Set of Level 3 Basic Linear Algebra Subprograms," *ACM Trans. on Mathematical Software*, vol. 16, no. 1, pp. 1-17, 1990.

[22] J. D. McCalpin, "A Survey of Memory Bandwidth and Machine Balance in Current High Performance Computers," *IEEE TCCA Newsletter*, pp. 19-25, 1995.

[23] D. Takahashi, and Y. Kanada, "High-Performance Radix-2, 3 and 5 Parallel 1-D Complex FFT Algorithms for Distributed-Memory Parallel Computers," *The Journal of Supercomputing*, vol. 15, no. 1, pp. 207-228, 2000.

[24] A. E. Koniges, R. Rabenseifner, and K. Solchenbach, "Benchmark Design for Characterization of Balanced High-Performance Architectures," *Proc.IEEE Intl Symp. Parallel and Distributed Processing*, pp. 196-208, 2001.

[25] R. Rabenseifner, and A. E. Koniges, *Effective Communication and File-i/o Bandwidth Benchmarks*,Recent Advances in Parallel Virtual Machine and Message Passing Interface, Springer, pp. 24-35, 2001.

[26] R. Rabenseifner, "Hybrid Parallel Programming on HPC Platforms," *Proc. Fifth European Workshop on OpenMP*, pp. 185-194, 2003.

[27] J. Dongarra, "Preface: Basic Linear Algebra Subprograms Technical (Blast) Forum Standard," *Intl Journal of High Performance Computing Applications*,vol. 16, no.1, pp. 115-115, 2002.

[28] R. C. Whaley, and J. J. Dongarra, "Automatically Tuned Linear Algebra Software," *Proc. ACM/IEEE Intl Conf. on Supercomputing*, pp. 1-27, 1998.

[29] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, and et. al., *LAPACK Users' guide*, Society for Industrial and Applied Mathematics Philadelphia, 1995.

[30] T. L. Saaty, "Analytic Hierarchy Process," *Encyclopedia of Biostatistics*, 2005.

[31] Collaborative Decision Making Software-Make Itrational, http://www.makeitrational.com, May 2008.

Reza Bakhshayeshi received MSc degree in Information Technology – Computer Networks from the Islamic Azad University of Garmsar, Iran. He is Cloud Computing Specialist at Cloud Research Center (CRC) of Amirkabir University of Iran. His research interests are in the areas of Cloud Computing, Distributed Systems and Monitoring.
E-mail: bakhshayeshi.reza@gmail.com

Mohammad Kazem Akbari received his Ph.D. in Computer Engineering in January 1995from the Case Western Reserve University, Cleveland, Ohio. He is now associate professor at the Amirkabir University of Tehran, Iran. His main research interests include Parallel Processing, Cloud Computing and VLSI Design.
E-mail: akbarif@aut.ac.ir

Morteza Sargolzaei Javan is ph.D. Candidate in Information Technology at the Amirkabir University of Tehran, Iran. He is Manager at Cloud Research Center (CRC). His research interests are in the areas of Cloud Computing and Big Data.
E-mail: msjavan@aut.ac.ir

Paper Handling Data:

Submitted: 21.05.2014
Received in revised form: 26.10.2014
Accepted: 05.12.2014
Corresponding author: Morteza Sargolzaei Javan, Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran.