Performance Analysis of Virtualized Environments using HPC Challenge Benchmark Suite and Analytic Hierarchy Process

Reza Bakhshayeshi
Islamic Azad University of Garmsar
Computer Engineering and IT Department
Garmsar, Iran
bakhshayeshi.reza@gmail.com

Mohammad Kazem Akbari
Amirkabir University of Technology
Computer Engineering and IT Department
Tehran, Iran
akbarif@aut.ac.ir

Morteza Sargolzaei Javan
Amirkabir University of Technology
Computer Engineering and IT Department
Tehran, Iran
msjavan@aut.ac.ir

***Abstract*—Steep improvement of cloud computing in recent years, persuaded experts admit it as a suitable and appropriate substitution for traditional computing methods. 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-driven computing 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 some virtualized environments, 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 then designate the best approach in general using Analytic Hierarchy Process.**

Keywords-cloud computing; virtualization; hypervisor; performance evaluation

I. 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. Fig. 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.

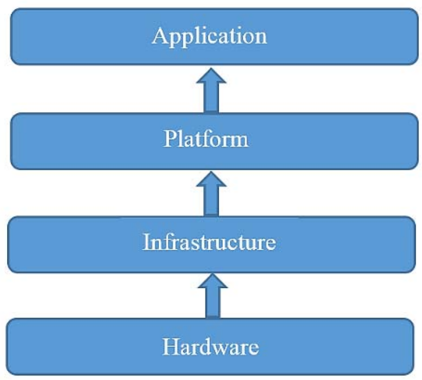

Figure 1. Cloud computing layered architecture.

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

978-1-4799-3351-8/14/$31.00 ©2014 IEEE

technologies, including VMware ESXi [3], KVM [4], Xen [5], Oracle VirtualBox [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], OpenVZ [11] and Oracle VM [12].

There have been previous works in this area, only to mention a few: Young et al. [13] analyzed the performance of Xen, KVM and VirtualBox using HPC Challenge benchmark, and they have suggested KVM as a preferable hypervisor. Luszczek et al. [14] evaluated the performance of HPC Challenge benchmark in several virtual environments, including VMware, KVM and VirtualBox 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 five different 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 obtained results and compares them. In section 5 we try to analyze obtained data using Analytic Hierarchy Process and finally the paper concludes in section 6.

II. 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 hypervisors.

Hypervisors are divided into two categories [15] (Fig. 2):

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 VirtualBox and VMware Workstation are examples of open source and commercial implementation of this type.

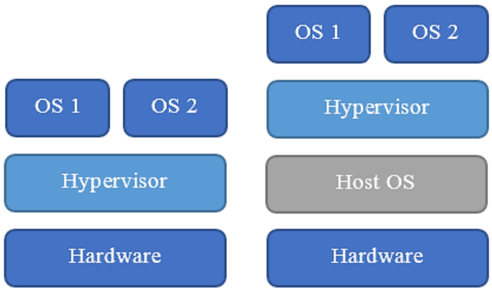

Figure 2. While bare-metal hypervisor runs directly on machine's hardware (left), hosted hypervisor runs on host's operating system (right).

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 (KVM) is a bare-metal hypervisor which enables Linux to perform as a virtual machine monitor by extending its capabilities [4].

Xen, very similar to KVM, is another bare-metal hypervisor which allows multiple operating systems to share hardware in a safe and resource managed way, without sacrificing either performance or functionality [5].

VMware ESXi, like KVM and Xen, is a type-1 virtual machine monitor, but unlike them, ESXi operates independently from any general-purpose operating system, therefore offering improved security, increased reliability, and simplified management. Its compact architecture designed for integration directly into virtualization-optimized server hardware, enabling quick installation, configuration, and deployment [3].

Oracle VirtualBox and VMware Workstation are cross-platform virtualization applications. From one side, they act as a computer within a computer and can be installed on Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. And from the other side, users can run multiple guest operating systems within the host operating system at the same time, each has its own applications while the original operating system environment remains intact and usable [6,7].

III. Performance Evaluation

In this section we evaluate the performance of virtualized environments created with either VMware ESXi 5, KVM on Ubuntu 3.2.0-23-generic kernel, Xen 4.1 or as bare metal hypervisors and Oracle VirtualBox 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 I.

TABLE I. 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 [16]	CPU performance	GFLOPS
DGEMM [17]	CPU performance	GFLOPS
STREAM [18]	Memory bandwidth	GB/s
PTRANS [19]	CPUs communication	GB/s
RandomAccess [19]	Memory updates	MUPS
FFT [19]	CPU performance	GFLOPS
Effective Bandwidth (bw., lat.) [20, 21]	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) [22] library, and the problem size. We use ATLAS [23] 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 [24]. For problem size we get benefit of the following recommended equation:

$$N = \sqrt{TM \div 8} \times 0.85 \qquad (1)$$

Which *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.

IV. Results

Each experiment executed 10 times so in total we had 150 executions (5 *hypervisors* × 3 *types of cluster* × 10 *times*). Here the results of 105 experiments (5 *hypervisors* × 3 *types of cluster* × 7 *benchmarks*) are going to present.

Fig. 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. Mainly there was a performance loss when number of instances increases due to performance overhead of creating new VM on the same machine.

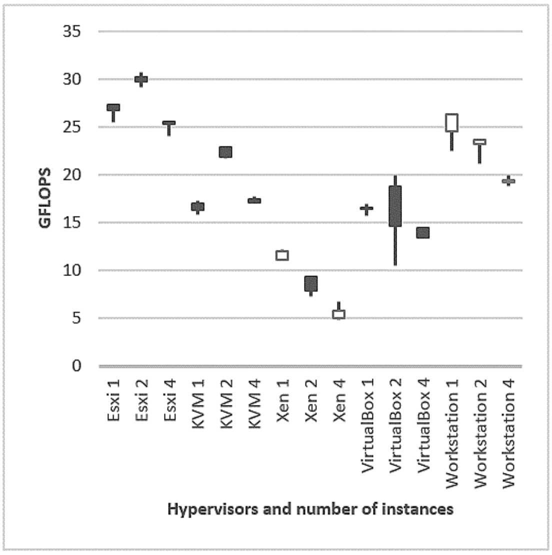

Figure 3. Comparison of *HPL* benchmark

Comparison of *PTRANS* benchmark is depicted in Fig. 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.

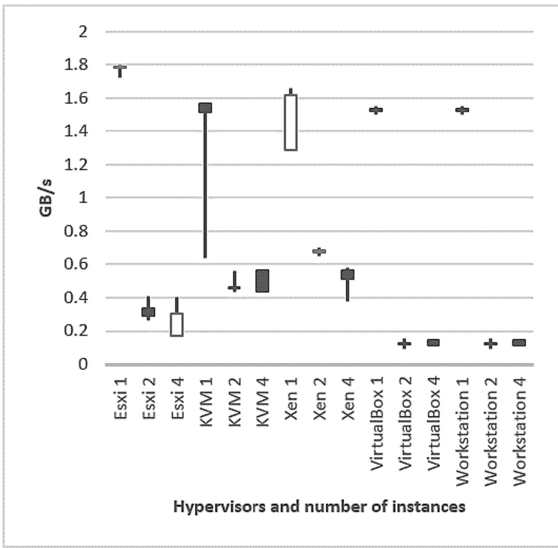

Figure 4. comparison of *PTRANS* benchmark

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

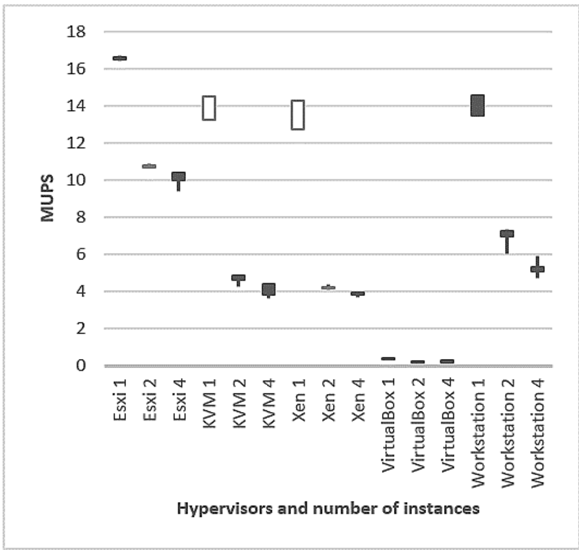

Figure 5. Comparison of *RandomAccess* benchmark

As shown in Fig. 6, VMware ESXi, KVM, Oracle VirtualBox 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.

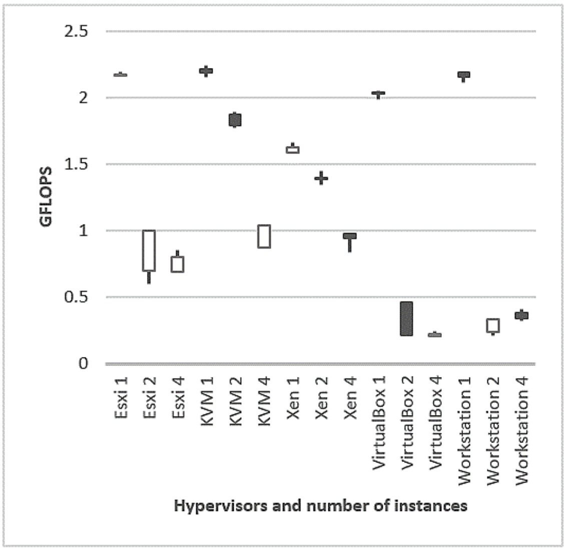

Figure 6. Comparison of *FFT* benchmark

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

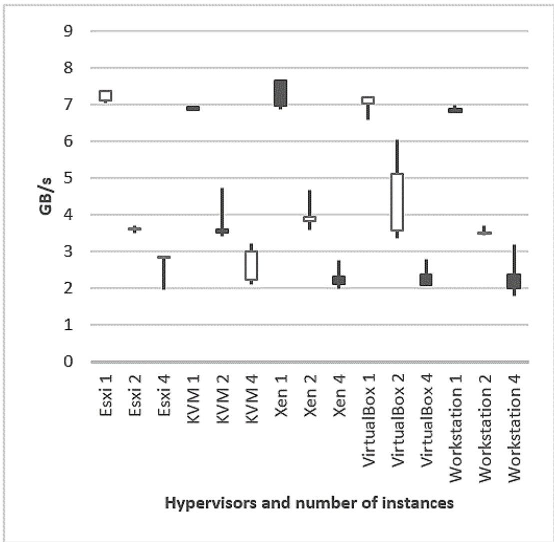

Figure 7. Comparison of *STREAM* benchmark

About *DGEMM* benchmark, as depicted in Fig. 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 Fig. 9 and Fig. 10, evaluated using *Effective Bandwidth (b_eff)* benchmark. KVM and Xen shown better bandwidth performance than others with 0.35 and 0.29 GB/s in 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.

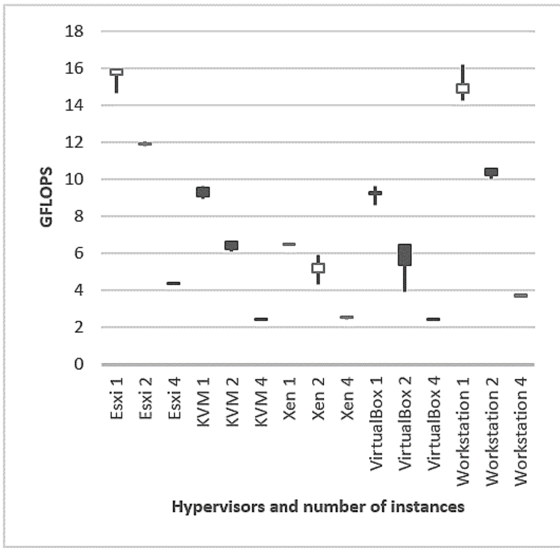

Figure 8. Comparison of *DGEMM* benchmark

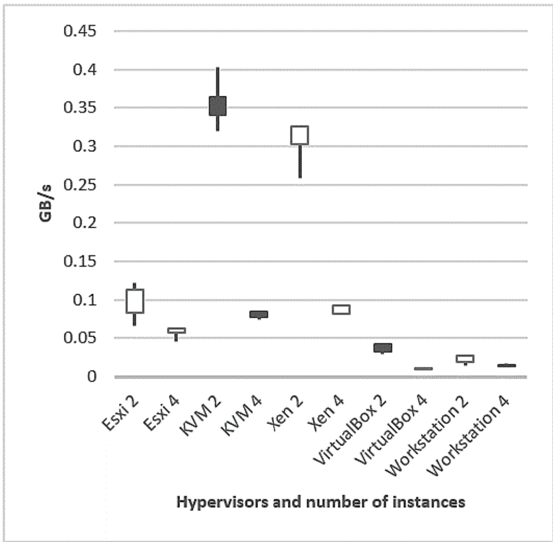

Figure 9. Comparison of Bandwidth

V. DISCUSSION

Table II 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.

As we can see, in general, VMware ESXi obtained best results, thus it can be a reliable solution for cloud and cluster infrastructure. However, when the workload is network bounded, KVM or Xen would be the preferable choice. These facts are also depicted in Fig. 11 and Fig. 12 using Analytic Hierarchy Process approach when considering benchmarks equal in pairwise comparisons. Analytic Hierarchy Process is a powerful decision making method in multi-criteria problems, it simplifies the problem into hierarchy through forming the

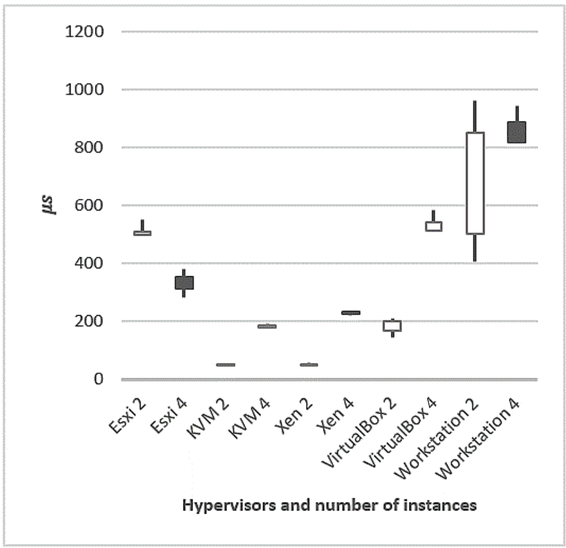

Figure 10. Comparison of Latency

comparison matrix to judge the weight [25]. This hierarchy made with help of MakeItRational AHP Software [26]. 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 its results in order to compare all cases. Fig. 12 shows the results using Analytic Hierarchy Process for 2 and 4 cluster cases including network bandwidth. Clearly, it can be concluded that KVM or Xen would be the preferable choice when the workload is network bounded.

TABLE II. 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
RandomAccess [19]	VMware ESXi	91.9
FFT [19]	VMware *ESXi* and Workstation, KVM, and VirtualBox	63.82
Effective Bandwidth (bw., lat.) [20, 21]	*KVM* and Xen	128.46, 155.77

VI. CONCLUSION

Steep improvement of cloud computing in recent years, persuaded experts admit it as a suitable and appropriate substitution for traditional computing methods. 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-driven computing requests of users in an efficient and cost effective manner. Apart from all these

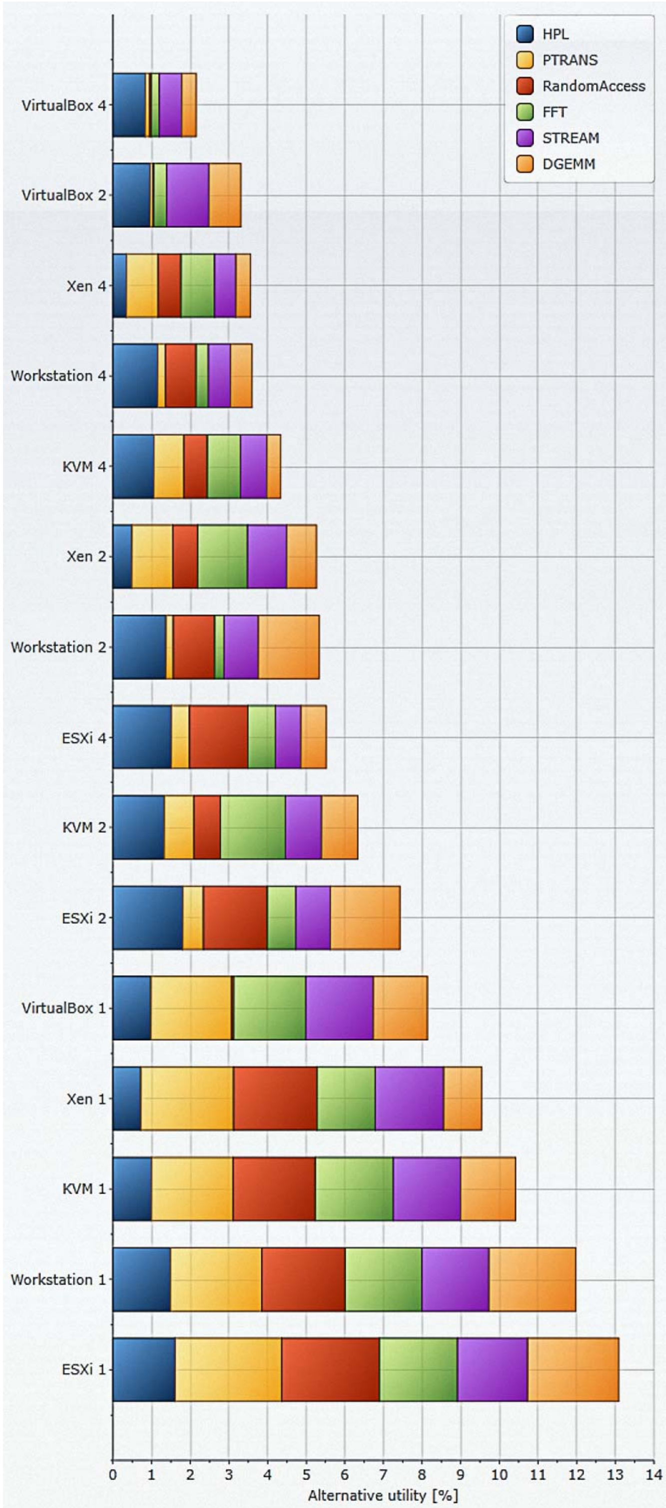

Figure 11. Benchmarks comparisons using Analytic Hierarchy Process. VMware ESXi gained highest total score.

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,

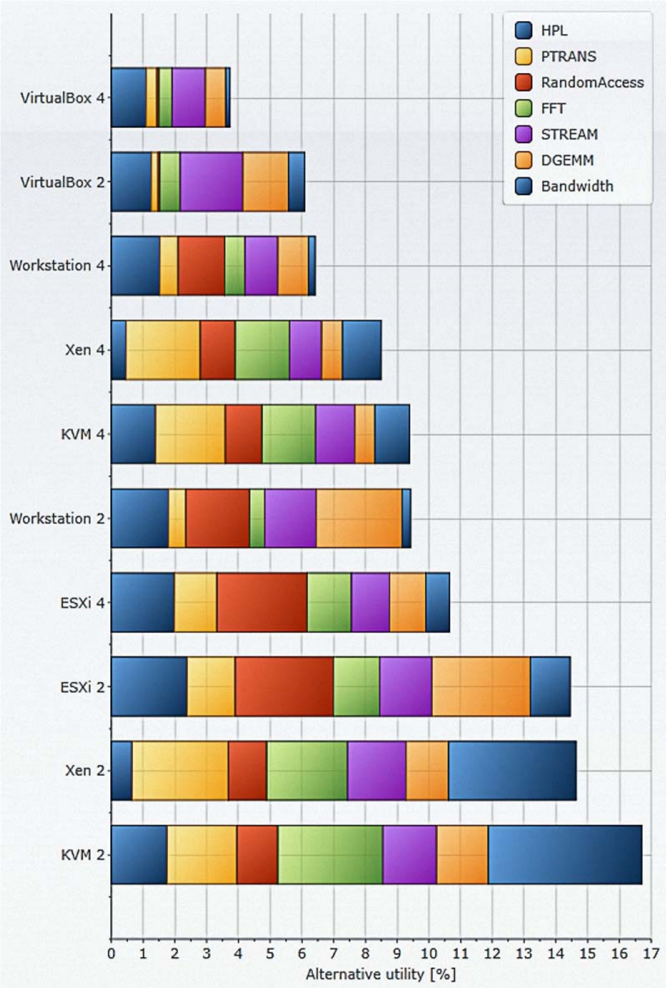

Figure 12. Benchmarks comparisons in cluster cases including bandwidth using Analytic Hierarchy Process. KVM obtained the highest total score in network bounded workloads.

Xen, Oracle VirtualBox, 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. As we concluded in previous section, in general, VMware ESXi obtained the best results, thus it can be a reliable solution for cloud and cluster infrastructure. 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, pp. 7-18, 2010.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," NIST special publication, vol. 800, p. 7, 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," in Proceedings of the Linux Symposium, 2007, pp. 225-230.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, et al., "Xen and the art of virtualization," ACM SIGOPS Operating Systems Review, vol. 37, pp. 164-177, 2003.

[6] J. Watson, "Virtualbox: bits and bytes masquerading as machines," Linux Journal, vol. 2008, p. 1, 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, et al., "The HPC Challenge (HPCC) benchmark suite," in Proceedings of the 2006 ACM/IEEE conference on Supercomputing, 2006, p. 213.

[9] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, et al., "Open MPI: Goals, concept, and design of a next generation MPI implementation," in Recent Advances in Parallel Virtual Machine and Message Passing Interface, ed: Springer, 2004, pp. 97-104.

[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, et al., "Quantifying the performance isolation properties of virtualization systems," in Proceedings of the 2007 workshop on Experimental computer science, 2007, p. 6.

[12] VM Oracle, "VirtualBox User Manual," ed: Oracle Corporation, 2011.

[13] 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," in Cloud Computing (CLOUD), 2011 IEEE International Conference on, 2011, pp. 9-16.

[14] P. Luszczek, E. Meek, S. Moore, D. Terpstra, V. M. Weaver, and J. Dongarra, "Evaluation of the HPC challenge benchmarks in virtualized environments," in Euro-Par 2011: Parallel Processing Workshops, 2012, pp. 436-445.

[15] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," Communications of the ACM, vol. 17, pp. 412-421, 1974.

[16] A. Petitet, "HPL-a portable implementation of the high-performance Linpack benchmark for distributed-memory computers," http://www.netlib.org/benchmark/hpl/, 2004.

[17] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, "A set of level 3 basic linear algebra subprograms," ACM Transactions on Mathematical Software (TOMS), vol. 16, pp. 1-17, 1990.

[18] J. D. McCalpin, "A survey of memory bandwidth and machine balance in current high performance computers," IEEE TCCA Newsletter, pp. 19-25, 1995.

[19] 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, pp. 207-228, 2000.

[20] A. E. Koniges, R. Rabenseifner, and K. Solchenbach, "Benchmark Design for Characterization of Balanced High-Performance Architectures," in IPDPS, 2001, p. 196.

[21] R. Rabenseifner, "Hybrid parallel programming on HPC platforms," in proceedings of the Fifth European Workshop on OpenMP, EWOMP, 2003, pp. 185-194.

[22] J. Dongarra, "Preface: Basic linear algebra subprograms technical (Blast) forum standard," International Journal of High Performance Computing Applications, vol. 16, pp. 115-115, 2002.

[23] R. C. Whaley and J. J. Dongarra, "Automatically tuned linear algebra software," in Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM), 1998, pp. 1-27.

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

[25] T. L. Saaty, What is the analytic hierarchy process?: Springer, 1988.

[26] Collaborative Decision Making Software - MakeItRational. Available: http://www.makeitrational.com