Computer and Information Science; Vol. 8, No. 2; 2015
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

System Level Benchmarks for the Cloud

Sanjay P. Ahuja[1] & Niharika Deval[1]

[1] School of Computing, University of North Florida, Jacksonville, Florida, USA

Correspondence: Sanjay P. Ahuja, School of Computing, University of North Florida, Jacksonville, Florida, USA. E-mail: sahuja@unf.edu

Received: March 4, 2015 Accepted: April 1, 2015 Online Published: April 26, 2015

doi:10.5539/cis.v8n2p58 URL: http://dx.doi.org/10.5539/cis.v8n2p58

Abstract

Cloud Computing has emerged as a transformational computing platform in which services are managed and delivered over the internet. Cloud computing enables organizations to outsource their infrastructure computing requirements with greater reliability and significant cost savings. Cloud computing services are provided on demand and are classified into three fundamental models: Infrastructure-as-a-Service (IaaS), Platform-as-a-service (PaaS) and Software-as-a-service (SaaS), based on level of abstraction. Given the flood of cloud providers off late, developers as well as end users are facing difficulty in choosing the right cloud provider or architecture that is best suited for their application requirements. It is very important for organizations to make a right decision before shifting their business applications on to the cloud to ensure that their applications once delivered on the cloud will be resilient and meet the performance and other service level agreements (SLAs) in the cloud environment. For this, the end users need a standard way of comparing and benchmarking the services offered by different providers. This paper discusses the concepts of benchmarking and its purpose. System-level benchmarks are used to measure the performance of overall system or subsystem. This paper surveys the system-level benchmarks for traditional (non-cloud) computing environment and makes recommendations for the system level benchmarks that can be used in cloud environments. The paper also makes future recommendations for proposing a standard benchmarking approach to determine performance variability and elasticity in the cloud environment.

Keywords: cloud computing, benchmarking, system-level benchmarks

1. Introduction

Cloud computing has fundamentally transformed the way the organizations around the globe approach the utilization and implementation of the technology by allowing them to carryout essential aspects of their business without having to deal with purchasing and maintaining infrastructure. There are many cloud providers offering their services across three basic service models-Infrastructure-as-a-service, Platform-as-a-service and Software-as-a-service.

Choosing a cloud provider from multitude of options is a tedious job for the developers because it is based on many key factors such as price, performance, reliability, dynamic scalability etc. Understanding these key metrics is very important for organizations or end users when determining appropriate provider. Moreover, public cloud providers do not reveal the details of implementation of the services provided by them. The most efficient and effective approach of assessing the performance is through benchmarking.

Benchmarking is the act of measuring the performance of a system or specific aspects of the performance and comparing the results to another system's results through unified procedure using specific metrics. Benchmarking often refers to concept of System under Test (SUT) (Folkerts et al, 2012) which is the set of components necessary to run the benchmark. A standard way of comparing the performance of the systems will help the organizations to make a right choice and utilize the service provided by the provider to its fullest potential. Benchmarking is necessary for organizations to identify, understand and adapt best practices to improve their own business.

Organizations would benefit from transparent and insightful comparisons of how different applications run on different clouds with different kinds of instances and configurations and make informed decisions about deploying their applications on cloud. With benchmarks users can evaluate the performance of various systems and compare the services of different cloud platforms in a vendor-neutral approach to assure that certain

guaranteed level of performance would be achieved before an organization moved their applications to the cloud provided by the service providers.

There are two types of benchmarks based on the level of performance measured. They are system-level benchmarks and component benchmarks. System level benchmarks evaluate the overall performance of a system that is executing real programs or applications. They evaluate each subsystem and indicate the effect of the subsystem on the overall performance. System level benchmarks are easy to understand and are useful in reviewing the hardware. Component level benchmarks, unlike system level benchmarks, test a specific component of the system. Component level benchmarks evaluate the performance of sub-system within a system. It takes a specific component into account and tests all its features to evaluate its efficiency.

Benchmarks for traditional computing evaluate the performance and price of a static system. Therefore these benchmarks are not always applicable in the cloud environment where scalability of resources is dynamic and performance varies accordingly (Binnig et al, 2009). As cloud technology is widely adopted, there is increasing need for the cloud benchmarks. This paper primarily focuses on system level benchmarks. It surveys the existing system level benchmarks in traditional as well as cloud environments and makes recommendations for benchmarks that can be applicable for cloud environment. The paper also addresses the need for benchmarking approaches to determine performance variability and elasticity capabilities in cloud environment and make some recommendations in these areas.

The remainder of this paper is organized as follows. Next section presents the related work in cloud benchmarking. Section III discusses the basic types of benchmarks in detail. Existing benchmarks in traditional computing are discussed in Section IV. Some of the system-level benchmarks in cloud computing are presented in the Section V. Finally, a conclusion of this paper and future works are presented in Sections VI.

2. Related Work

Ahuja et al in (Ahuja, 2013) presented system-level benchmarking on Amazon EC2 cloud services to measure the performance of CPU, memory and I/O. The paper focuses on three largest public clouds providers-Amazon EC2, Google App engine and Microsoft Windows Azure. The paper also focuses on the pricing model of all the three providers. The benchmarks STREAM, IOR, AND NPB-EP were used in the experiment to test memory, I/O and CPU performance respectively. These benchmarks were run on micro instance and small instances. The results clearly indicate that many factors contribute to the performance variability.

The authors in (Binnig et al, 2009) argue that traditional benchmarks are not sufficient to benchmark the cloud services because cloud computing has additional features such as elasticity, pay per usage and fault tolerance which are not addressed by traditional benchmarks. It also presents some suggestions on how to build benchmarks that are suited to the cloud environment. The paper also analyses TPC-W benchmark and discusses how this benchmark can be adopted to cloud environment. New metrics to analyze scalability, costs and fault tolerance are also discussed.

The authors in (Gillam et al, 2013) conducted a benchmarking project for comparing the performance of cloud computing systems. Three public cloud providers-Amazon EC2, Rackspace and IBM SmartCloud, and one private cloud provider – OpenStack, were chosen for this benchmarking project. Ubuntu 10.06 and RHEL6 operating systems were selected and generic instance types are chosen to run the benchmark. The selection of the benchmarks were used to test memory I/O, disk I/O, CPU, application and network performance for these cloud providers. Memory I/O was tested using STREAM benchmark. CPU capabilities were evaluated using LINPACK benchmark. Bonnie++ and IOZone were the benchmarks used to test Disk I/O. Application (compression) was tested using Bzip2. Iperf, MPPTEST and simple speed tests are used to evaluate the network performance. The results show that there is performance variation among different instances in different regions of the same offering. The paper concludes by identifying future work by extending the wider range of instances and more benchmarks to obtain better performance distribution.

3. Types of Benchmarks

Benchmarks are categorized into two types based on the measurement of levels of performance – component level benchmarks and system level benchmarks. Benchmarks can also be classified into two other categories when we consider benchmark composition. Those two categories are synthetic benchmarks and application benchmarks.

Component level benchmarks are used to test a particular component of a system running real applications/workload. These benchmarks focus on testing the performance of a subsystem instead of taking the whole system into account. Component level benchmarks can be implemented as synthetic benchmarks.

Synthetic benchmarks are the combination of basic operations that together result in measuring the performance of the system under test. For example, consider the Bonnie++ benchmark that measures the speed of a hard drive or file system by averaging the basic functions - Random Create, Random Delete, Sequential Create and Sequential Delete. The best examples of Synthetic benchmarks are Whetstone and Dhrystone. Synthetic benchmarks are of limited use to users because their only purpose is to yield a specific performance value and do not compute any real tasks. Another drawback of synthetic benchmarks is that they do not reflect the program behavior since they are not real programs.

System level benchmarks are used to test the overall performance of a system running real applications/workloads. System level benchmarks are used to compare the systems of different architectures. These benchmarks measure the entire system which includes operating systems, networks, compilers, database, I/O, processor and etc. An example of system level benchmarks includes the TPC benchmarks suite. Most application benchmarks are considered system-level benchmarks because application benchmarks measure the overall performance of a system and evaluate how each subsystem or component affects the overall performance. Application benchmarks are larger and complex to execute and are generally application specific.

4. Traditional System-Level Benchmarks

i. SysBench (Falko, 2012): This benchmark is used to compute CPU computing power and memory bandwidth in a single interface. It measures system performance by performing tests on CPU, file I/O and MySQL.

ii. SYSMark (Bapco, 2012): This benchmark measures and compares PC performance based on real world applications that runs a preset script. The scripts are based on the user driven workloads and usage models developed by application experts. It determines the computer performance on task switching and it also tests the multi-tasking when two or more applications are running real tasks at the same time.

iii. Winstone benchmark (Computer Business Review, 1992): This benchmark belongs to the family of PC Magazine benchmarks that measures overall performance of PC. Winstone benchmark is used to measure the overall performance of a PC running real Windows applications. The content creation Winstone 2002 version measures a PC's performance running 32-bit Windows based content creation applications on Windows 98, 2000, XP.

iv. PCMark04 (FutureMark, 2003): This benchmark is the first multitasking benchmark that features both system level and component level benchmarking. System level benchmarking of PCMark04 measures CPU's overall performance that includes tests for CPU, memory, graphics and hard disk.

v. SDM (System Development multitasking) Benchmark Suite: SDM Release 1 (SPEC, 1994) is a benchmark suit that contains two multi-tasking system level benchmarks for UNIX systems: 057.sdet and 061.kenbus1. These two benchmarks are used to test the system resources such as CPU, I/O, memory, operating system and many UNIX utilities. These benchmarks allows users to compare systems based on the behavior of a system's throughput as the load varies on the system and allows the developers to compare different hardware and software releases. It measures the performance by gradually increasing number of concurrent scripts of the benchmark which results in increase in the system's workload. SDM benchmarks performance depends on the configuration of system resources.

vi. 057.sdet benchmark is used to find the throughput of a system when multiple processes are concurrently executed where each process executes a script form script directory.061.kenbus1benchmark depends on the concept called think time, defined as a user's finite typing rate with pauses in between the operations.

vii. Netperf (Netperf.org, 2005): This benchmark is used to measure the various aspects of performance of network infrastructure. The network performance is measured in Gbps. It supports Berkley Sockets Interface and TCP or UDP for both IPv4 and IPv6. Unidirectional throughput i.e. bulk data transfer and end-end network latency tests are provided by netperf benchmark.

5. Traditional System-Level Benchmarks Used in the Cloud

This sections surveys system level benchmarks from traditional computing environments that have been used for benchmarking cloud based systems.

i. Unixbench Benchmark: The purpose of Unixbench benchmark (Unixbench, 1984) is to measure the overall performance of UNIX-like system at system-level. It measures the performance of the system

running single threaded and multi-threaded tasks. Unixbench runs series of individual tests, aggregates the scores and produces final indexed score which is the geometric mean of individual test scores. The testing factors that are included in the benchmark include Dhrystone, Whetstone, throughput, process creation, shell scripts, system call overhead etc. The results of this benchmark depend on hardware, operating system and compiler version. Higher Unixbench score indicates better performance. Cloud Spectator used Unixbench to compare public cloud IaaS instances (Cloud Spectator, 2014).

ii. Bonnie++: Bonnie++ is a disk I/O performance benchmark that conducts tests to derive performance relating data read and write speeds, maximum number of seeks per second, and maximum number of file metadata operations/second (Bonnie++, 2007). Read et al used Bonnie++ and IOZone for benchmarking disk IO performance in the cloud (Read, 2010).

iii. IOZone: This benchmark is used to test the file system performance to measures variety of file operations. It is known for its broad analysis of file system of vendor's computer platform. It is portable and also runs on many operating systems. It offers tests for following operations: Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read, pread ,mmap, aio_read, aio_write (IOZone, 2006).

iv. Cachebench: This benchmark suite evaluates the performance of memory hierarchy of computer systems, particularly multiple levels of cache present on or off the processor (Cachebench, 2006). It incorporates 8 benchmarks of which the first three benchmarks Cache Read, Cache Write and cache Read/Write/Modify provide information about the compiler. The remaining benchmarks are hand tuned Cache Read, hand tuned Write and hand tuned Cache Read/Write/Modify, memcpy() and memset(). This benchmark suite was used by Ostermann et al to evaluate cloud computing services for scientific computing (Ostermann, 2008).

v. IOR benchmark: Interleaved or Random (IOR) benchmark tests the performance of parallel file system using various interface and access patterns (IOR Benchmark, 2013). System performance is measured focusing on Parallel/Sequential read/write operations. The drawback of this benchmark it needs MPI installed on the system for process synchronization. This benchmark has been used by Ghoshal et al to evaluate the IO performance of virtualized cloud environments (Ghoshal, 2013).

vi. Blogbench: This is a file system benchmark for UNIX systems. It stresses the file system with multiple threads of random reads, writes, and rewrites (Denis, 2008). It mimics the behavior of a blog by creating blogs with content and pictures, modifying blog posts, adding comments to these blogs, and then reading the content of the blogs.It has been used to benchmark a virtual machine of an OpenStack cloud (OpenStack, 2014).

vii. Iperf: Iperf is a reliable benchmarking tool to measure maximum TCP bandwidth, allowing tuning of various parameters and UDP characteristics (Iperf, 2010). Iperf using TCP streams measures network throughput whereas Iperf in UDP measures packet loss, jitter, delay. Advantage of this benchmark is that the server can handle multiple connections at a time. The benchmark can be run for user specified time (Default run time is 10seconds) rather than set of amount of data to transfer. It has been used by Cloud Spectator to test the internal network throughput capability of largest cloud IaaS providers (Cloud Spectator, 2014)

viii. DBench: DBench is a popular open source benchmark that used to test the disk I/O performance (DBench, 2008). It uses file system calls to measure the disk performance. It generates only file system load. Throughput result is expressed in MB/sec. DBench benchmark has been used by Cloud Spectator group to analyze disk I/O performance of cloud providers in (Cloud Spectator, 2014).

6. Conclusions and Future Work

The paper highlights the need for benchmarking cloud systems and provides clear distinction between the types of benchmarks. Since cloud technology is increasingly being adopted there is consequent need of cloud benchmarking. As an initial step this paper recommends some traditional benchmarks which could possibly be used in a cloud environment at a system level. One of them is SYSMark benchmark suite that is used to analyze and test the performance of graphics and overall performance (SYSMark, 2014). AMD, NVIDIA and Intel started utilizing the SYSMark benchmark suite for testing purposes, but AMD & NVIDIA opted out of SYSMark 2012 disputing the scores doesn't reflect real world usage. However, even the latest version of SYSMark suite which is released in 2014 is used to measure the performance based on real world applications widely being used on variety of Intel CPUs. This benchmark can be used in Cloud environment to measure the

overall system performance of any server instance configured with Intel based CPUs or graphics card. Another benchmark is Sysbench benchmark suite (Timme, 2012) that can be used in the cloud environment to measure system performance by performing basic tests on CPU and file I/O performance of cloud providers across various instance configurations.

Several benchmarks have been designed for performance evaluation with regards to CPU, RAM, disk, storage and network. Cloud customers often base their deployment decision on these factors along with price and SLAs. But it is more essential to incorporate performance variability, scalability and elasticity aspects during vendor selection process. These factors need to be considered alongside performance evaluation and pricing factors.

Therefore the paper recommends two important areas for further research. First is measuring performance variability and second is analyzing elasticity in cloud platforms. From the existing research in (Iosup, 2012) and (Leitner, 2014) it is clear that many approaches have been proposed for determining performance variability of cloud systems. Some of those works defined mean, standard deviation, coefficient of variation (Schad, 2010) and max-min approaches (Talluri, 2003) to calculate performance variability over time. Additional research is needed into the methodology to determine performance variance across the various cloud service models.

Although there are numerous approaches to analyze and compare elasticity of various systems, there is lack of unified strategy that provides objective comparison of elastic capabilities of different offerings. Existing research works (Kupperberg, 2011), (Galante, 2012), (Jennings, 2014) often combine elasticity with its related terms - efficiency and scalability factors, without considering elasticity as an separate attribute (Herbst, 2013). There are few developed methodologies for elasticity metrics targeting SaaS platform and PaaS platform individually. Research is needed to quantify elastic capabilities of systems across all the three cloud service models.

References

Ahuja, S. P., Furman, T. F., Roslie, K. E., & Wheeler, J. T. (2013). Empirical Performance Assessment of Public Clouds Using System Level Benchmarks. *International Journal of Cloud Applications and Computing, 3*(4), 81-91. http://dx.doi.org/10.4018/ijcac.2013100106

Bapco.com (2012). *SYSmark 2012 Lite Features - BAPCo.* Retrieved November 11, 2014, from http://bapco.com/products/sysmark-2012-lite.

Binnig, C., Kossmann, D., Kraska, T., & Loesing, S. (2009). *How is the Weather tomorrow? Towards a Benchmark for the Cloud*. Proceedings of the Second International Workshop on Testing Database Systems, Providence, Rhode Island.

Bonnie++ (2007). Retrieved from http://www.coker.com.au/bonnie++/

Cachebench. (2009). *Cachebench Home Page*. Retrieved November 11, 2014, from http://icl.cs.utk.edu/projects/llcbench/cachebench.html

Cloud Spectator. (2014). Retrieved from http://cloudspectator.com/for-enterprises/

Code.google.com (1984). *Byte-unixbench - A Unix benchmark suite - Google Project Hosting.* Retrieved November 11, 2014 from https://code.google.com/p/byte-unixbench/

Computer Business Review (1992). Microsoft and Ziff-Davis Create Winstone Benchmark. *Computer Business Review.* Retrieved November 11, 2014, from http://www.cbronline.com/news/microsoft_and_ziff_davis_create_winstone_benchmark.

DBench (2008). *DBench benchmark*. Retrieved from https://dbench.samba.org

Denis, F. (2008). *Blogbench - manned.org*. Manned.org. Retrieved November 12, 2014, from http://manned.org/blogbench/fbdee406

Falko, T. (2012). *How To Benchmark Your System (CPU, File IO, MySQL) With sysbench.* HowtoForge - Linux Howtos and Tutorials. Retrieved November 11, 2014, from https://www.howtoforge.com/how-to-benchmark-your-system-cpu-file-io-mysql-with-sysbench

Folkerts, E., Alexandrov, A., Sachs, K., Iosup, A., Markl V., & Tosun. C. (2012). *Benchmarking in the Cloud: What it Should, Can, and Cannot Be*. TPC Technology Conference on Performance Evaluation and Benchmarking (TPCTC 2012), Istanbul, Turkey, 173-188.

FutureMark (2003). *PCMark Bechmark.* Retrieved November 11, 2014, from http://www.futuremark.com/benchmarks

Galante, G., & Bona, L. (2012). *A Survey on Cloud Computing Elasticity*. In Proceedings of the 2012

IEEE/ACM Fifth International Conference on Utility and Cloud Computing, UCC'12, 263–270, Washington, DC, USA, 2012. IEEE Computer Society.

Ghoshal, D., Canon, R., & Ramakrishnan, L. (2011). *IO Performance of Virtualized Cloud Environments*. Lawrence Berkeley National Laboratory: Lawrence Berkeley National Laboratory. LBNL Paper LBNL-5432E. Retrieved from http://datasys.cs.iit.edu/events/DataCloud-SC11/p08.pdf

Gillam, L., Li, B., Loughlin, J., & Tomar, A. (2013). Fair Benchmarking for Cloud Computing Systems. *Springer Open Access Journal of Cloud Computing*. http://dx.doi.org/10.1186/2192-113X-2-6

Herbst, N., Kounev, S., & Reussner, R. (2013). Elasticity in Cloud Computing: What it is, and What it is Not. Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24-28.

IOR Benchmark. (2013). *IOR*. Retrieved November 12, 2014, from https://www.nersc.gov/users/computational-systems/cori/nersc-8-procurement/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks/ior/

Iosup, A., Yigitbasi, N., & Epema, D. (2012) *On the Performance Variability of Production Cloud Services*, CCGRID, 2011, Cluster Computing and the Grid, IEEE International Symposium on, Cluster Computing and the Grid, IEEE International Symposium, 104-113. http://dx.doi.org/10.1109/CCGrid.2011.22.

IOzone. (2006). Retrieved from http://www.iozone.org/

Iperf (2010). *IPerf.* Retrieved from https://iperf.fr/

Jennings, B., & Stadler, R. (2014). Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management,* 1–53.

Kupperberg, M., Herbst, N., Kistowski, J., & Reussner, R. (2011). Defining and quantifying elasticity of resources in cloud computing and scalable platforms. *Tech. Rep.,* 2011. Retrieved from http://digbib.ubka.uni- karlsruhe.de/volltexte/1000023476

Leitner, P., & Cito, J. (2014). *Patterns in the Chaos – a Study of Performance Variation and Predictability in Public IaaS Clouds*. Retrieved from http://arxiv.org/pdf/1411.2429.pdf

Netperf.org (2005). *The Netperf Homepage*. Retrieved November 11, 2014 from http://www.netperf.org/netperf/

OpenStack (2014). *OpenStack Cloud Software*. Retrieved November 16, 2014, from https://review.openstack.org/#/c/97030/

Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., & Epema, D. (2008). *An Early Performance Analysis of Cloud Computing Services for Scientific Computing,* Delft University of Technology Parallel and Distributed Systems Report Series, report number PDS-2008-006. Retrieved December, 2008, from http://www.st.ewi.tudelft.nl/~iosup/PDS-2008-006.pdf

Read, J. (2010). *Disk IO Benchmarking in the Cloud*, Cloud Harmony, June 2010. Retrieved November 16, 2014, from http://blog.cloudharmony.com/2010/06/disk-io-benchmarking-in-cloud.html

Schad, J., Dittrich, J., & Quiane-Ruiz, J. (2010). *Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance. Proceedings of the VLDB Endowment, 3*(1), 2010.

Spec.org (1994). *SPEC SDM.* Retrieved November 11, 2014, from https://www.spec.org/sdm91/

SYSMark. (2014). *SYSMARK.* Retrieved from http://bapco.com/products/sysmark-2014

Talluri, S., & R. Narasimhan. (2003). Vendor Evaluation with Performance Variability: A Max–Min Approach. *European Journal of Operational Research, 146*, 543–552.

Timme, F. (2012). *Sysbench Benchmark*. Retrieved from http://www.howtoforge.com/how-to-benchmark-your-system-cpu-file-io-mysql-with-sysbench