Experiences Applying Performance Evaluation to Select a Cloud Provider

CHARLES BOULHOSA RODAMILANS, ARTUR BARUCHI, EDSON TOSHIMI MIDORIKAWA
Department of Computer and Digital Systems Engineering
Polytechnic School – University of São Paulo
Avenida Prof. Luciano Gualberto, travessa 3, 158 – 05508-970
São Paulo – SP – Brazil
{rodamilans, artur.baruchi, emidorik}@usp.br

Abstract: - Due to commodity hardware and accessible virtualization technologies several cloud providers has arisen. The increase of cloud service providers available in the market makes the choice of one of them a hard task to the average customers. This paper presents the performance analysis of I/O storage, CPU and Memory for public cloud providers. This work has evaluated instances types with different storages types in Amazon EC2 and Rackspace providers. The IOzone, SPEC and NPB benchmarks have been used to evaluate each feature. The main findings are: (1) the external volumes can perform better than local disks and the network may not be the system bottleneck; (2) local disk can offers better cost-benefit than expensive storages; (3) the number of threads can improve the performance in local or external SSD volume; and (4) the same instance type of a cloud provider can offer different performance according to the site location. These results shows that a deep knowledge of the application behavior and the technologies used by cloud provider are the main keys to make the right choice about the service provider.

Key-Words: - Cloud Computing; Virtualization; Performance Evaluation; Benchmark; Performance Measures; Solid State Drive (SSD)

1 Introduction

Nowadays, Cloud Computing [1] has been adopted in many industries, government, and universities as a basic computational infrastructure. The technical reasons for this adoption are performance isolation, elasticity, on-demand services, scalability and availability [1]. The clouds has been used to storage and process large amount of data. Many public cloud providers offer instances and storage services [2] and it is difficult to choose what cloud provider to use, depending on which factor is more critical (e.g., performance, cost, security, or fault-tolerance). Many other aspects can be considered; for example, Terry et al. [3] discusses consistency aspects for cloud storage to be included in SLA contracts.

The characterization of a cloud provider is the first step to understand what application is more appropriate for this provider. Various aspects can be analyzed in cloud environment like computation, memory access and I/O disk [4]. These characteristics are directly related with applications characteristics (CPU bound, Memory bound and I/O bound). It is difficult to acquire hardware specifications in Cloud Computing because the providers make a hardware encapsulation. Cloud providers have many instances and storages' flavors [5][6]. Some technologies are incorporated in cloud services to performance improvement, e.g. Solid State Drive (SSD) [6]. All these characteristics have showed how hard is to choose a service provider.

In this context, there are several works of performance evaluation in Cloud Computing, e.g. High Performance Computing (HPC) [6][7][8] and I/O performance [6][9]. However, to the best of our knowledge, there is little research comparing performance characteristics among public cloud providers. In all these works, the benchmarks are used to help to understand the cloud performance aspects [4][10][11][12][13].

The large diversity of cloud providers offers a substantial variety of instances types and prices. It is hard to choose which instance to use and where the data will be stored to process in only one cloud. The difficulty increases with the availability of many cloud providers that offers many different instances and configuration types. This problem shows that is necessary a mechanism to compare the current variety of cloud providers.

The present paper aims to find a new way that improves the decision accuracy when choosing a service provider to a given application. This evaluation helps to understand and to find which instance configuration has a better performance. This research help to select which instance type, storage type and cloud provider to execute an application. This paper also explains the instance configuration impact in the application performance. Moreover, it permits to reduce the cost with the cloud provider choice.

This paper evaluates CPU, memory and I/O performance in Amazon EC2 [14]. The I/O performance analysis is also made in Rackspace [15]. Amazon EC2 and Rackspace are selected since they are the most popular Infrastructure-as-Service (IaaS) providers.

Two types of instances (small and large) and two types of storage (disk and block) are evaluated in each cloud provider. The SSD storage type is also analyzed. The IOzone benchmark is used to evaluate the storages using sequential (write and rewrite) and random (write and read) access modes operations. All disk operations are synchronous since this mode is used in database applications. The parallelism achieved by using multithreading is analyzed from 1 to 5 threads.

The instance and storage categorization in public cloud permits to select which cloud provider to use focusing on performance or costs. The choice of more expensive instance and more powerful hardware is no guarantee that the performance will be better. It is necessary to select the best cloud instance and configuration in several cloud providers depending on the prevalent type of operation in the user application [16] in order to achieve a better performance. The storage technologies also influence the I/O performance, in particular SSD storage can improve the data transfer performance and the local SATA drive can be the best choice in terms of cost. Finally, the number of threads can improve the performance without any additional costs to the user.

The evaluation has shown that the external block storage can present the best I/O performance compared with local storage, showing that network subsystem is not the bottleneck. But in some situations, local storage can offer better performance with minor costs than more expensive and powerful storages. Moreover, SSD disks performance can be improved increasing the number of threads, such as in local SSD storages in Amazon EC2 and SSD block devices in Rackspace.

Additional evaluation tests were performed in Amazon AWS aiming to evaluate and quantify the performance of CPU and Memory bound applications in different Virtual Machines (VM) setups available at Amazon EC2.

Main questions that guided this study are: (a) does the increased cost of larger instances is followed by a performance gain? (e.g. doubling the cost, will the performance double too?); and (b) how better an application will perform when executing in a larger instance?

This paper is organized as follows: Section 2 describes the related works. Section 3 presents an overview of Amazon EC2 and Rackspace providers, and its instances types and storage types. Section 4 describes the methodology and configuration applied in the experiments. Section 5 provides the evaluation analysis of I/O, CPU and memory performance and compare the instances, storages and investigate the influence of multithreading in Amazon and Rackspace with IOzone, SPEC, and NPB benchmarks. Section 6 presents our conclusions and future work.

2 Related Work

Several works have studied the performance of virtualized environments in Cloud Computing [17][18][9]. Usually, these papers analyze a specific aspect in only one environment or cloud provider. Hence, there is a gap in previous work about performance evaluation comparing different cloud providers. This section summarizes related work about performance aspects in Cloud Computing.

Various efforts have evaluated public clouds, especially in Amazon EC2 [17][18][9]. Previous works has used standard benchmark like IOzone [10], Linpack [11], NAS parallel benchmark (NPB) [12] and SPEC [13]. In this work we adopted the IOzone benchmark to evaluate disk performance, SPEC benchmark to the CPU and NPB benchmark to the memory.

Some previous works has shown that is possible the use of High Performance Computing (HPC) in a Cloud environment [6][7][8]. In HPC environments, it is important to guarantee a minimum performance ratio in all resources (CPU, memory, disk, network bandwidth, etc.). This paper investigates disk, CPU and memory performance.

Ghoshal *et al.* [9] compared I/O performance in two clouds providers, using IOR benchmark, but use only write and read I/O operation types and one public cloud. In this paper we evaluate many different access modes, including synchronous operations.

Expósito *et al.* [6] compared I/O performance in many storages at Amazon EC2 and the focus is only on Amazon EC2. This paper considers only sequential read and write operations in Amazon EC2, because the authors analyses scientific applications only. Here we consider more access modes (read, write, rewrite) and synchronous access found in database applications. Besides that we evaluate two cloud providers.

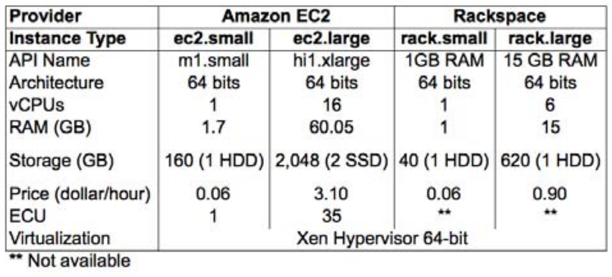
However, the most of previous works is focused in performance in only one public cloud provider, whereas there are very few previous works showing any results about choosing a public cloud provider under some performance requirements.

Comparing this paper to the previous works, we evaluate the I/O, CPU, and memory performance of Amazon EC2 and Rackspace. We evaluated the impact of the I/O performance with different instances types, number of threads and storage type in different Cloud Computing environments. The aspect of CPU and memory performance is also analyzed according to the computing power offered by available instance types. We show aspects on how to choose the public cloud provider taking into consideration certain performance requirements.

3 Overview of Cloud Providers Instances Types and Storage

This section shows an overview about Amazon EC2 and Rackspace instance types and storages. All information about the instances is summarized in Table 1. All providers use 64-bit Xen hypervisor.

Table 1: Description of chosen instances types.



Provider	Amazon EC2		Rackspace	
Instance Type	ec2.small	ec2.large	rack.small	rack.large
API Name	m1.small	hi1.xlarge	1GB RAM	15 GB RAM
Architecture	64 bits	64 bits	64 bits	64 bits
vCPUs	1	16	1	6
RAM (GB)	1.7	60.05	1	15
Storage (GB)	160 (1 HDD)	2,048 (2 SSD)	40 (1 HDD)	620 (1 HDD)
Price (dollar/hour)	0.06	3.10	0.06	0.90
ECU	1	35	**	**
Virtualization	Xen Hypervisor 64-bit			

** Not available

3.1 Instances Types Overview

Amazon EC2 and Rackspace offer a large variety of instances types, without any hardware relationship and this makes it difficult to compare instances between providers. Among all available instance types, we chose two distinct groups: small and large. The small group consists of instances with simple hardware (1 virtual CPU or vCPU), a small amount of disk storage, and has about the same prices in the providers. The large group contains the second best instance of each provider, with different vCPUs, storages, and prices.

Amazon EC2 offers many types of instances, from general-purpose to storage-optimized instances. The M1 Small instance (m1.small, named ec2.small in this article) has the following configuration: 1.7 GBytes of memory, 160 GBytes of local storage and 1 ECU (EC2 Compute Unit) processor. The Amazon WS informs that one processor with 1 ECU computing power provides the equivalent CPU capacity of an 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. This instance type costs U$ 0.06 per hour.

For small instances, the provider does not detail the specific processor architecture. According to /proc/cpuinfo, it informs that the processor has 1 physical core and is an Intel Xeon E5430 @ 2.66GHz processor.

In the large group, the High I/O Quadruple Extra Large instances (hi1.4large, named here ec2.large) was chosen; it has two SSD disks, with 1 TBytes each, as local disk storage. This instance is used to provide a very high I/O performance. Moreover, this instance has 8 cores (16 vCPUs with hyper threading) with 35 ECUs and 60.5 GBytes of memory. Each user can use only two simultaneous ec2.large instances because of Amazon limitation. This instance is much more expensive than ec2.small, and costs U$ 3.10 per hour.

Rackspace also has many instances and are called by the size of memory. The 1 GB RAM type (name here rack.small) is a resource with 40 GBytes of local storage, 1 vCPU and 1 GBytes of memory. This is a second simplest instance in Rackspace and costs U$ 0.06 per hour (the same price of ec2.small).

The other selected Rackspace instance is called 15 GB RAM (named here rack.large). It is a resource with 630 GBytes of local storage, 6 vCPUs and 15 GBytes of memory. It is the second best instance in Rackspace and costs U$ 0.90 (three and a half times cheaper than ec2.large). Rackspace doesn't have a CPU performance metric to compare its instances, like Amazon ECU.

Cost aspect

Amazon offers different purchasing options: (1) on-demand, the payment is made by hour of use and obtains immediately computation; (2) spot instances, it is made a bid on unused instances in Amazon EC2; (3) reserved instances, the user pays a fee for a time interval and gets a discount for one or three-year of use. The Rackspace has only on-demand instances, therefore it is used on-demand purchasing in this article, for comparison purposes.

Each instance will be allocated to users in non-dedicated manner (many VMs per physical node). Amazon has an additional dedicated allocation option with payment per hour, but the Rackspace has only this allocation with payment by month.

3.2 Storage Systems Overview

The selected providers offer the same storage types with different names. This paper adopts the following types of storage as described in Table 2: (1) disk storage, the user data is lost when the instance is released and it is usually stored in local disk device. It is also called non-persistent storage. Amazon’s local storage is called ephemeral or instance store and in Rackspace it is called disk; (2) block storage is a remote volume that can be attached to instance as block storage device. This volume is accessed through the network and user data is persistent. It is called Elastic Block Store (EBS) in Amazon and volume in Rackspace; (3) object storage manages data as object and the storage provides a web services interface for access the files. This storage is accessed by network. Simple Storage Service (S3) is the Amazon’ commercial name for object storage and is known as Cloud File for Rackspace. The object storage is not used in this work.

Table 2: Types of Storage Devices.

Type of Storage	Access Mode	Amazon Name	Rackspace Name
Disk	local	Ephemeral or Instance Store	Disk
Block	network	EBS	Volume
Object	network	S3	Cloud File

Table 3 presents the selected storage devices and Table 4 shows the storages costs. The prices are in dollar and were checked on October 2013. The storage price is an additional cost of use of the selected storage device. The (local) disk does not have any additional cost. Amazon permits EBS storage with minimum size of 1 GBytes and Rackspace with minimum size of 100 GBytes for Standard and SSD storage. This minimum size reflects the minimum costs. Amazon also offers the Provisioned Input/Output Operations Per Second (IOPS) storage type for EBS storage. This is a high performance storage for I/O intensive applications. IOPS value depends on the size of storage.

Table 3: Storages Description.

Provider	Storage Name	Description	Type of Storage
Amazon	ec2.root	Standard EBS in root device	block
	ec2.stand	Standard EBS	block
	ec2.iops	1000 IOPS	block
	ec2.ssd	Local SSD disk	disk
Rackspace	rack.local	Local HDD disk	disk
	rack.sata	Standard Volume	block
	rack.ssd	SSD Volume	block

The storage options in Amazon are: (1) ec2.root, the data and operating system is stored in the same Amazon block storage; (2) ec2.stand, the data is in one Amazon block storage and the operating system in other block storage; (3) ec2.iops, the provider assures a minimum IOPS performance for this storage. It can be guaranteed 1,000 IOPS for 100 GB volumes; and (4) ec2.ssd, this option use SSD disk for storage in Amazon. This disk can be used only in ec2.large instances.

Table 4: Storages Costs.

Provider	Storage Name	Storage Price (dollar/GB/month)	I/O Price (dollar)
Amazon	ec2.root	0.100	0.100 *
	ec2.stand	0.100	0.100 *
	ec2.iops	0.125	0.100 **
	ec2.ssd	0.000	0.000
Rackspace	rack.local	0.000	0.000
	rack.sata	0.150	0.000
	rack.ssd	0.700	0.000

* per 1 million I/O request
** per provisioned IOPS/month

The storage options in Rackspace are: (5) rack.local, the data are stored in local disk and this data is ephemeral; (6) rack.sata, the storage is external and the disk is SATA; (7) rack.ssd, this storage is external and it uses a SSD device for storage.

4 Experimental Configuration and Evaluation methodology

This section shows the experimental configuration of virtual machines on the cloud providers and the evaluation methodology adopted in the performance tests.

4.1 I/O Evaluation Methodology Description

The I/O performance evaluations of the Amazon and Rackspace providers have been made with the ec2.small, ec2.large, rack.small and rack.large instances types (Table 1) and ec2.root, ec2.stand, ec2.iops, ec2.ssd, rack.local, rack.sata and rack.ssd storages (Table 3).

This evaluation has used the IOzone benchmark of a local file system (ext3) on ephemeral disks and blocks devices, and the numerical results are a throughput in KBytes/sec.

The benchmark parameters used are -i0 -i2 -o -r -t, where (a) -i0, means running write and rewrite operations in sequential mode; (b) -i2, executes write and read operations in random mode; (c) -o sets synchronous operations; (d) -r, defines a record size in KBytes; (e) -s, the file size; and (f) -t, the number of threads or process.

The IOzone benchmark has been compiled in each instance type with GNU C 4.4.7 compiler without any additional flags to performance improvement.

The I/O evaluations were conducted by two distinct test cases: first, a sequential write test followed by a sequential rewrite test (-i0 option), and second, a random write test followed by a random read test (-i2 option). All operations were made in synchronous mode. The reason for these choices is because random accesses and synchronous operations are widely used in database environments.

The record size is 4 KBytes. It is the default option in most operation systems. The chosen file size is 3.4 GBytes for all experiments. In small instances evaluation, the main memory of virtual machines is stressed, and the file has at least the double of memory size. As ec2.small has more memory than rack.small, so 1.7 GBytes has been chosen as the basis to define the file size for the benchmark execution. The same file size has been used in the large instances evaluation. The aim is to know if a better instance really improves the performance observed by the application. The memory was not stressed in large instances evaluations.

The block storage size was setup to 100 GBytes because it is the minimum block size in Rackspace. This evaluation has used 1000 IOPS, because it is the maximum value for the chosen block size.

All Amazon experiments were made in the us-east-1 region (North Virginia), within the east-1a availability zone and Rackspace experiments were made in Chicago (ORD) region.

In the multithreading analysis, the number of threads was increased to evaluate performance parallelism in the instances and storage types. The number of vCPU in small instance is 1 and the number of threads was increased from 1 to 5 to verify if the performance would improve. During the experiments in large instances, the variation was the same, although more vCPUs were available.

The operating system selected was CentOS 6.4 for all experiments. The VM image that was used at Amazon was Official CentOS 6.4 x86_64 at AWS Marketplace (AMI ID ami-bf5021d6). The used Rackspace image was CentOS 6.4 default.

In this evaluation, the IOzone benchmark was executed five times and the standard deviation in small instance type experiments was calculated. The cache memory has been cleaned before running each test.

4.2 CPU and Memory Methodology Description

In order to evaluate the performance of Amazon EC2 instances in terms of CPU and memory resources, an experimental study was conducted as follows.

Table 5 presents the performed tests and Virtual Machines configurations in Amazon EC2 Cloud. All virtual machines were installed with CentOS 6.4 64-bit Linux. The kernel version used was 2.6.32-358.6.2 (image ID - ami-eb6b0182). Moreover, the virtual machines were installed in US EAST 1B Amazon Data Center, we used EBS as storage device. Following, the Table 5 summarizes the executed tests and VM configuration.

Table 5: Summary of VM configuration.

Instance Type	API Name	Memory (GBytes)	vCPU	ECU	CPU Benchmark	Memory Benchmark
cpu.mem. small	m1.small	1.7	1 (xeon 2.00 GHz)	1	SPEC CPU - twolf	Nasa Parallel Benchmark – BT (size B)
cpu.mem. medium	m1.medium	3.75	1 (xeon 2.00 GHz)	2		
cpu.mem. large	m1.large	7.5	2 (xeon 2.27 GHz)	4		

To evaluate the Amazon instances, it was used two benchmarks. One benchmark has a CPU Bound behavior (SPEC CPU); the other one has a Memory Bound behavior (NPB BT).

Among several benchmarks available in SPEC CPU, the twolf (TimberWolfSC) benchmark was chosen. This benchmark spends most of his executing time in internal loops and, as consequence, runs long memory access intervals causing many cache misses. On the other hand, the memory bound benchmark (NPB BT) executes several changes in the memory content during the run (and consumes a considerable amount of RAM memory, about 200MB). The NPB makes available several benchmark sizes to perform tests, but for this study it was chosen the B size (middle size – tridimensional matrix 102x102x102).

The metric used to evaluate the execution of benchmarks was execution time, in seconds. At the beginning of the execution it was collected the epoch timestamp (seconds since January 1st, 1970). The instances computational power was measured using a Perl program that computes the execution time of a 200 interactions, looping for 10 times. Inside the looping, a mathematical float point calculation is done. After that, with the collected information from the loops a new metric is computed (called N) that takes into account the processor clock. This metric shows the computational power (like Linux Bogomips), where greater values are better.

5 Performance Evaluation of I/O, CPU and Memory

This section presents an analysis of the performance results of the I/O storage, CPU and memory on two cloud-computing providers, Amazon EC2 and Rackspace, using the benchmarks described in section 4. The obtained results are used to guide the user seeking the best (or better) option to instantiate the virtual machines in these computational environments.

5.1 Evaluation of I/O Performance in Amazon EC2 and Rackspace

This section shows the evaluation of disk I/O performance in Amazon EC2 and Rackspace. We present the obtained performance results and a brief analysis of our findings in comparing both providers.

5.1.1 Evaluation of Small Instances

This section presents an analysis of the performance results of small instances in Amazon EC2 and Rackspace providers.

Sequential access mode

Fig. 1 presents the average bandwidth and standard deviation for disk I/O operations obtained for sequential write and rewrite tests and random write and read tests (measured in KBytes/sec).

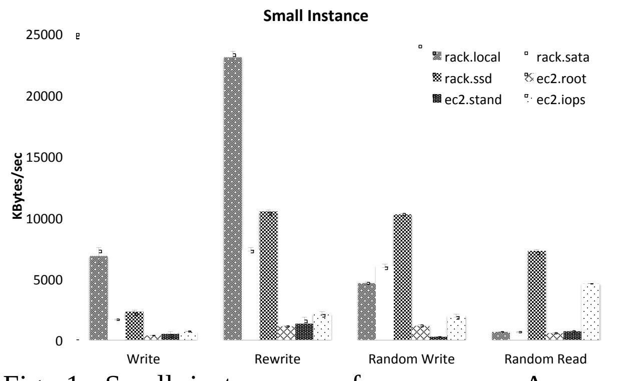


Fig. 1: Small instances performance on Amazon EC2 and Rackspace focusing in I/O operation type.

The more significant results in sequential write and rewrite operations performance were observed in Rackspace instances with local storage (rack.local). As rack.local has a local storage in Rackspace, it is not necessary additional payment to use it. rack.local had the best performance for write and rewrite operations and is the cheapest configuration. If the application updates frequently the storage, this instance and provider seems to be the right solution.

For write operation, the results of rack.local can obtain 197% of performance improvement over the second best provider configuration (rack.ssd). All the configurations performance is very similar in Amazon EC2 for write operation, ranging from 358 KBytes/sec to 682 KBytes/sec. The best Amazon configuration (ec2.iops) is 358% worse than the worst Rackspace configuration (rack.sata).

These results in the small instances can be explained, despite the fact that Amazon EC2 instances have more memory than Rackspace instances (1.7GB versus 1GB). The best results in Rackspace are probably due to the better memory and disk configuration (higher bandwidth between memory and disk, numbers of channels, disk technology and cache memory, among others). It is difficult to accurately report why this happens because the information about real resources in Cloud Computing is “cloudy”. It is not easy to get accurate information about the hardware. The hardware virtualization hides the real information about the hardware, e.g. number of memory channels or I/O bus.

As can be observed in sequential rewrite operation, the Rackspace also obtains better results than Amazon EC2 in all experiments.

The Rackspace with local storage (rack.local) is the best option again, as it gets up to 23,046 KBytes/sec, a 220% improvement compared to Rackspace with the SSD storage (rack.ssd) and 350% improvement compared to the best performance configuration in Amazon (ec2.iops).

These results confirm that Rackspace with local disk on small instances provide very high performance for sequential write and rewrite operations with synchronous I/O.

Table 3 shows the costs of instances and storages in both providers. Small instances in Rackspace and Amazon has the same costs (U$ 0.60/hour) and the local storage in Rackspace doesn’t need additional payment. These results demonstrate that local storage is better in terms of costs, too.

Random access mode

Performance results on random write and rewrite operations confirm one more time that Rackspace is better than Amazon for small instances.

For random write operations, the Rackspace with SSD external volume is able to get a 70% improvement compared with SATA external volume and it is able to quadruple performance of an Amazon with provisioned IOPS external volume.

Regarding the random read operation, the results are more insightful as Rackspace with SSD volume can obtain only 58% of the performance of Amazon with IOPS volume. However, the SSD volume is able to get a 940% improvement compared with others cases. In particular, in the Amazon scenario, IOPS volume is clearly the best performance, obtaining up to 557% performance improvement over other storages.

The local storage in Rackspace servers is a SATA HDD and an external storage can be SATA or SSD disk. The SSD external volume uses an iSCSI interface.

The access latency caused by the interconnection network to the external device has minor performance impact compared to the use of a local SATA disk. It happens because the local storage is shared with other users or VMs in the server (server consolidation), while the external storage in the data center is a specialized storage device appropriate for many simultaneous accesses.

Comments on the results

This evaluation about small instances has shown that Rackspace with local volume has better performance for sequential write and rewrite operations. Another advantage of using local disks is that its costs is free, whereas others volumes are charged. For random write and read operations, the best choice for performance is Rackspace with SSD external volume. That means, an additional cost can be used to obtain extra I/O performance. If the costs are a problem, the Rackspace using local disk is a good choice too.

5.1.2 Threads Evaluation of Small Instances

Today multithreaded applications are becoming very common and we evaluated parallel accesses to the storage devices in Cloud Computing environments.

Random write tests

Fig. 2 presents the maximum bandwidth obtained (measured in KBytes per seconds) for synchronous random write operations with variation of the number of accessing threads. The standard deviation is showed in each experiment. Small instances have only one vCPU in Amazon EC2 and Rackspace. In these experiments, the number of threads increases, ranging from 1 to 5, and all threads run in parallel.

For random write operation, the disk performance in rack.local appears to have no influence with the variation of number of threads. The observed bandwidth was almost the same, around 4,600 KBytes/sec. In Amazon EC2 with IOPS volume (ec2.iops), the increase in the number of threads also has little effect because the access to the volume is limited only by number of provisioned IOPS. In these experiments, Amazon EC2 ensures up to 1,000 IOPS for external volume and the page size used for random write operation is 4 KBytes.

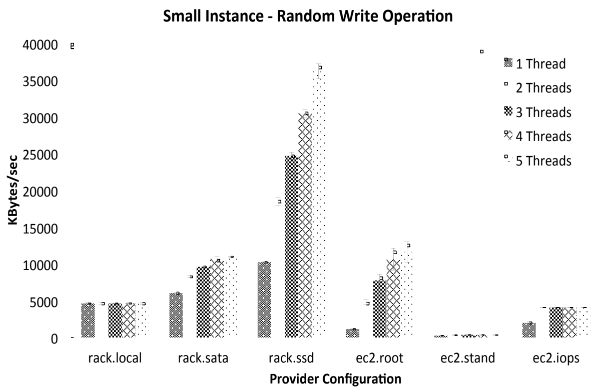


Fig. 2: Small instance performance with number of threads increasing for synchronous random write operation.

ec2.stand has the worst results of random write performance and the variation on the number of threads is not able to produce any benefit of performance.

The comparative analysis among different providers configurations shows that the increase of number of thread is able to get significant improvements of performance in rack.sata, ec2.ebs and rack.ssd experiments, with enhancements of 82%, 920% and 254%, respectively.

The best result of performance for random write operation is obtained by Rackspace with SSD external volume that showed a transfer rate of 36,000 KBytes per second with 5 threads.

These results show that Rackspace with external volume is better than local volume. This fact suggests that the local disk concurrency among cloud users is more significant to decrease performance of random write performance than the use of network to access an external volume.

The number of threads cannot be ignored and it is necessary to verify how many threads can increase the performance for the each specific application.

Random read tests

Fig. 3 presents the results for average bandwidth and standard deviations of small instances with synchronous random read operations and increases on the number of threads.

For the Amazon EC2 with EBS in root device (ec2.root), the experimental results has presented that increasing the number of threads do not improve the performance in random read operation, while the better performance in random write operation shows that the number of threads can influence the bandwidth on certain types of operations in the same provider configuration. This observation suggests that increasing the number of threads do not always improve the performance.

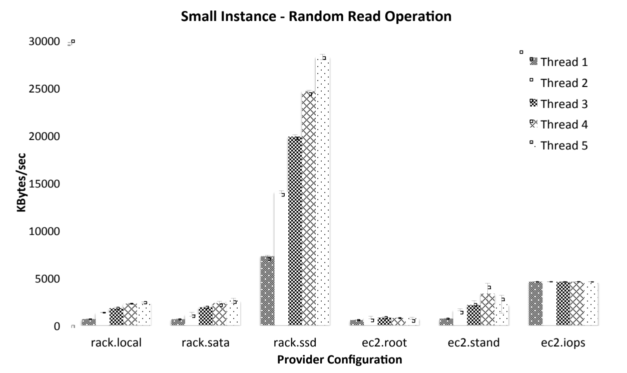


Fig. 3: Small instance performance with number of threads increasing for synchronous random read operation.

The ec2.stand results show that the number of threads can improve the performance, but it is limited. The increase of 4 to 5 threads decreases the performance in 35%. This result shows that an increase on the number of threads can decrease the performance.

The random read graph shows similar results for ec2.iops presented in previous random write graph. Again, these results confirm that the maximum provisioned IOPS is a performance limiting factor for all threads.

As can be observed, the Rackspace with SSD volume configuration is again the best performer. It achieves up to 28,000 KBytes/sec for random read operation with 5 threads. The SSD volume is able to obtain better performance than others, even when the number of threads increases.

Comments on the results

These evaluations have shown that Rackspace with external SSD volume is the best configuration for synchronous write and read random operation for small instances in these experiments. Increases in the number of threads can improve significantly the performance for this configuration. These evolutions confirm that the network is not the bottleneck in Rackspace for these operations. Then the use of external volume can be a good choice for improvements in the I/O performance.

The Amazon EC2 with provisioned IOPS configuration is a better choice than others configurations in Amazon provider. Increases in the number of threads cannot improve the performance when the I/O operations already are at the maximum IOPS ratio.

5.1.3 Evaluation of Large Instances

This section presents an analysis of I/O performance with the large instances group. With larger available memory size and better hardware configuration, these instances are expected to show higher bandwidth for disk accesses.

Write tests

Fig. 4 presents the performance for sequential write, sequential rewrite and random write operations, measured in KBytes/sec, for large instances with one thread. The configurations of storage in Rackspace use local disk, and external SATA and SSD volumes. The storage in Amazon EC2 has been configured only with a local SSD disk.

The results in Fig. 4 show that the Amazon EC2 with SSD obtains the best performance for all write operations experiments. In the random write operation, the SSD disk is able to achieve 31,517 KBytes/sec, which represents a 330% improvement compared with Rackspace SATA volume configuration.

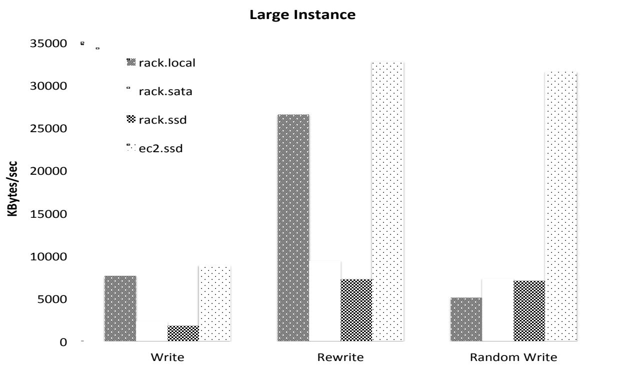


Fig. 4: Large instance performance on Amazon EC2 and Rackspace focusing in I/O Type.

The Rackspace configurations perform very similarly for write and rewrite operations in small instances, which the local disk is better than an external volume. Regarding random write operation, the external volumes in Rackspace is a little better than local disk. However, the SATA volume is able to get better performance than SSD in all the cases. It maybe happens because in these experiments the file size is smaller than the memory size. These results confirm that SATA external volume is better than SSD external volume for file size smaller than the main memory.

Read Tests

Fig. 5 shows the experimental read bandwidths for large instances, measured in KBytes/sec, for random read operation. All evaluated instances and storages presented similar results. The Rackspace SATA volume gets up 1,005,052.69 KBytes/sec, up to 2.5% improvement compared to other configurations. So we can conclude that as the local disk Rackspace is the cheaper option with the similar bandwidth results, an application that presents a random read access pattern can be allocated in a virtual machine with local disk in Rackspace.

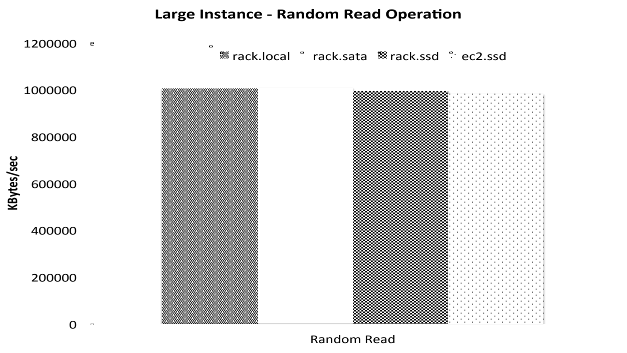


Fig. 5: Large instance performance on Amazon EC2 and Rackspace focusing in I/O type.

Comments on the results

The first conclusion that can be derived from this analysis is that the Amazon EC2 with local SSD disk provides better performance for write, rewrite and random write operations. The second conclusion is that random read operations have similar results for all devices configurations and the best choice is Rackspace with local disk because that configuration is cheaper. The costs analysis reveals that Rackspace with local disk is the best option in most cases.

5.1.4 Threads Evaluation of Large Instances

We repeat in this section the analysis of the influence of multithreading an application in terms of disk performance. For the sake of comparison with small instances, we limit the number of threads to five.

Fig. 6 shows the random write bandwidths obtained with the variation of the number of threads in the large instances. Again, the Rackspace with local disks doesn't present a significantly performance improvement as the number of threads increases and gets up to 5,094.15 KBytes/sec. On the other side, SSD disk in both providers presents a performance improvement. The Rackspace and Amazon EC2 configurations get a 242% and 204% improvement, respectively, comparing 1 thread and 5 threads. But the Amazon EC2 with SSD gets the best result, for 5 threads, with a throughput of 96,000 KBytes/sec. Once more, the Amazon with provisioned IOPS does not improve the performance as discussed in section 5.1.2. The Amazon EC2 with provisioned IOPS with 1 thread is not showed in the graph because its execution was aborted.

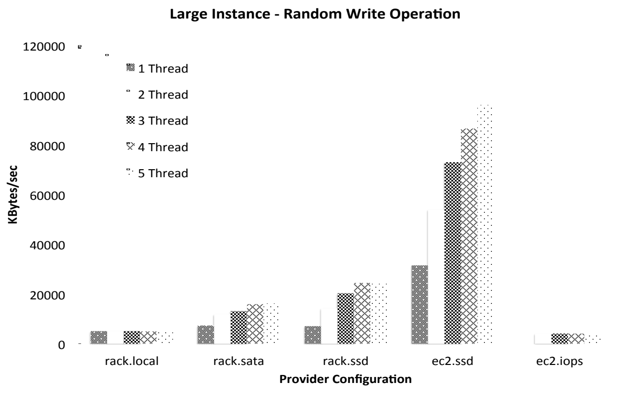


Fig. 6: Large instances performance with random write operation and variation of the number of threads.

Fig. 6 presents the random read operation results where the minimum bandwidth is approximately 1,000,000 KBytes/sec for most cases with one thread. Rackspace with local and external SATA storage achieve similar performance in most cases (except for 4 threads results), and obtains 298% performance improvement comparing with 1 thread and 4 threads. The best performance is obtained by the Amazon EC2 with local SSD disk that gets up to 4,600,000 KBytes/sec for 5 threads. This is an improvement of 370% compared with 1 thread.

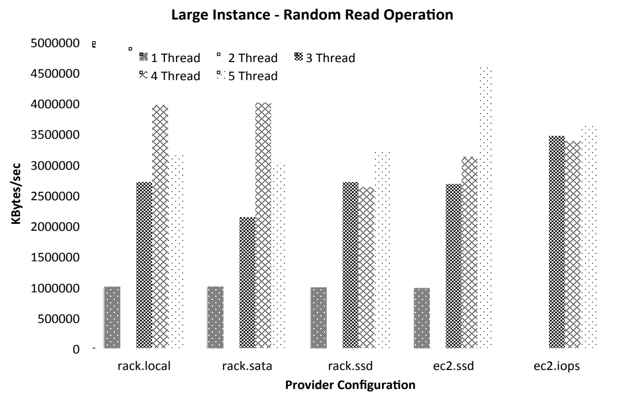


Fig. 7: Large instances performance with random read operation and variation of the number of threads.

Comments on the results

This evaluation has shown that Amazon EC2 with local SSD disk is a better choice for performance in random write and read operations for the large instances. These results also confirm that the Rackspace with local disk and Amazon EC2 with limited IOPS don’t get performance improvements as the number of threads increases. The best cost benefit choice for random read operation in large instances is presented by Local disk Rackspace instances with 4 threads.

5.1.5 Evaluation Review

Table 6 and Table 7 present the summary of our analyses for the best provider and storage configurations for I/O performance with 1 thread and many threads, respectively. These tables help to choose what provider and storage is better for sequential and random I/O operations.

The comparative analysis among different instances types shows that external volume devices can be better than local disks (Table 6). The local disk becomes the bottleneck because it is shared with multiples users and the network bandwidth can provide enough bandwidth as opposed to local disk access.

Table 6: Best provider and storage configuration for I/O performance for 1 thread.

Operation		1 Thread	
		Small	Large
Sequential	Write	Rackspace - local disk	Amazon EC2 - SSD disk
	Rewrite	Rackspace - local disk	Amazon EC2 - SSD disk
Random	Write	Rackspace - SSD block	Amazon EC2 - SSD disk
	Read	Rackspace - SSD block	Rackspace - local disk *

* similar performance, cheaper costs

Table 7: Best provider and storage configuration for I/O performance for many threads.

Operation		Many Threads	
		Small	Large
Random	Write	Rackspace - SSD block	Amazon EC2 - SSD disk
	Read	Rackspace - SSD block	Amazon EC2 - SSD disk

The analysis of the increase of the number of threads shows that: (a) expensive instances cannot be a good choice with one thread only cases, e.g. Rackspace with local disk with write and rewrite sequential operations for small instance and random read for large instance (Table 6); (b) Amazon EC2 with provisioned IOPS volume does not improve the performance for more than 2 threads because its limited IOPS ratio; (c) The increase of the number of threads is better for SSD device performance (Table 7); Rackspace with SSD volume gets better performance with increase of number of threads for small instances (although it has only one vCPU); Amazon EC2 with local SSD as storage has 16 vCPUs and gets better performance for large instances using many threads.

5.2 Evaluation of CPU and Memory on Amazon EC2

Based on the results for I/O operations on Cloud Computing providers, we also executed some performance studies to analyze the behavior of CPU and memory virtualization.

All the performed tests were run for 15 times. Due to small sample sizes, the Student-t distribution was used. The used confidence interval was 95%.

5.2.1 Execution time

In Fig. 8 it is presented the experimental results from SPEC CPU and NPB benchmarks, where it can be observed the performance of each tested instances in Amazon.

In Fig. 8 the cpu.mem.small instance has a poor performance in both workloads, as expected. However, the best instance was cpu.mem.medium. The cpu.mem.medium instance has an 89% performance improvement for BT Benchmark and 24% of improvement for the SPEC CPU benchmark, when comparing against the cpu.mem.large instance.

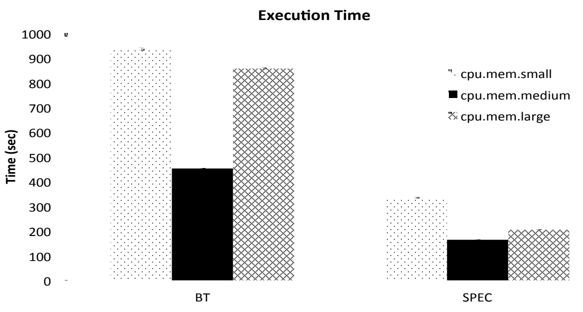


Fig. 8: Benchmark Average Execution time in Small and Large Instances.

5.2.2 Computational Difference

In Fig. 9 is shown the N metric result obtained in Perl Script benchmark, as discussed in section 4.2.

As expected, the N value is higher when the instance resources were bigger. However, the N value increase rate did not follow the cost rate. The computational difference between the cpu.mem.small and cpu.mem.medium instance is about 54%, however the cost increases 100% (twice the cost). The computational difference between the cpu.mem.medium and cpu.mem.large is 24%, and the cost increases 100%.

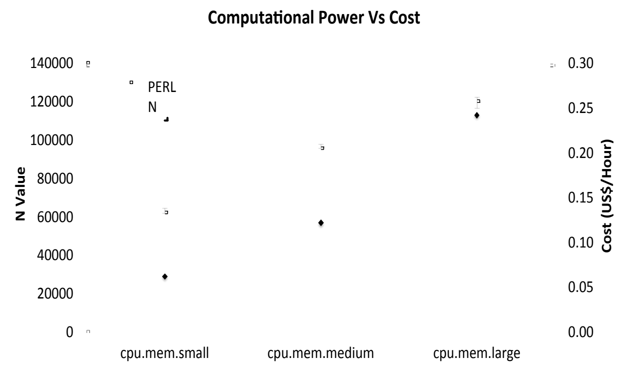


Fig. 9: N Value for all instances evaluated in Amazon and their costs.

5.2.3 Analysis and main findings

The poor performance presented by cpu.mem.large instance was a surprise in this study (it was better than cpu.mem.small instance, but worst than cpu.mem.large). Even the better processor (as observed in Fig. 9), the performance in both benchmarks was too low.

To evaluate this scenario, the main hypothesis was the influence of other Virtual Machines running in the same physical host. In order to evaluate this scenario, a Virtual Machine was installed in a different Amazon data center, placed in West.

Virtual Machines Interferences

Same tests were executed in the new Virtual Machine placed at Amazon WEST data center in order to evaluate the interference hypothesis. The results are presented in Fig. 10.

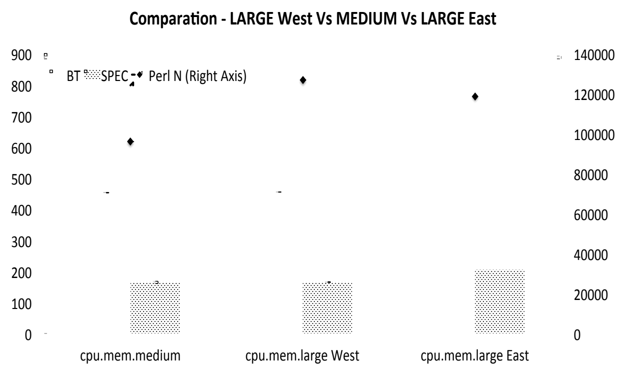


Fig. 10: Comparison against cpu.mem.large instance in DC West data center and cpu.mem.medium and cpu.mem.large Instance in EAST data center.

Using the same instance type, but in different data centers it is possible to obtain different performances. This result leads to a confirmation of the interference hypothesis.

These results show that the selection of an appropriate cloud provider is more difficult than choosing the more powerful or expensive one.

6 Conclusion

Cloud computing provides abstract infrastructure as a service which the consumer “pay as you go”. Also offers high I/O performance for HPC and commercial applications. Amazon and Rackspace are the main public cloud providers that offers infrastructure as a service.

This work has presented a performance evaluation analysis about the different I/O storages and instances types on the Amazon and Rackspace cloud providers. The different instances types and storages have been evaluated in both providers. The IOzone benchmark has been used in all experiments with sequential write and rewrite access modes, and random write and read access modes, using synchronous operations. The number of threads has been increased for evaluations of performance improvements.

The performance results have shown that the storages and a variation of the number of threads present significant differences in Amazon and Rackspace providers. This paper has also revealed that using the external block storage can provide more performance than local disk and the network is not the bottleneck performance, e.g. the study case of Rackspace Small instance with SSD block for random write and read operations with one thread. In addition, this paper has characterized what cloud provider can get better performance for each disk access operations and the evaluation revealed that local disk can offers better performance and cheaper costs for large instances, e.g. study case Rackspace large instance with local disk for random read operation with one thread. Finally, the analysis of the SSD storage has shown that numbers of threads improve significantly the performance of SSD disk in Amazon and SSD block in Rackspace.

From the CPU and memory analysis, we can say that not always the most expensive instance has the best performance. And even the same instance type can have different performance ratios depending on the selected datacenter location.

In general, the answers to our questions are: (a) not always a more expensive instance presents higher performance, and (b) the performance of a larger instance has to be considered case-by-case depending on the application’s resource usage pattern.

The next step of our research work is to explore in more details the network performance. That will help to understand in depth the network influence in the storage performance. One important future work is the analysis of the performance of a real application (e.g. openModeller [19], a biodiversity analysis application).

Acknowledgements

This work has been partially funded by National Council for Scientific and Technological Development – CNPq – Brazil under Project 140852/2009-3. The authors also wish to thank Prof. Dr. Liria Matsumoto Sato for her comments and suggestions on the earlier versions of this paper.

References:

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and others, “Above the clouds: A Berkeley view of cloud computing,” *EECS Dep. Univ. Calif. Berkeley Tech Rep UCBEECS-2009-28*, 2009.

[2] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, “Cloud monitoring: A survey.,” *Comput. Netw.*, vol. 57, no. 9, pp. 2093–2115, 2013.

[3] D. B. Terry, V. Prabhakaran, R. Kotla, M. Balakrishnan, M. K. Aguilera, and H. Abu-Libdeh, “Consistency-based service level agreements for cloud storage,” in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, pp. 309–324.

[4] V. Vedam and J. Vemulapati, “Demystifying Cloud Benchmarking Paradigm-An in Depth View,” in *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*, 2012, pp. 416–421.

[5] R. R. Expósito, G. L. Taboada, S. Ramos, J. Touriño, and R. Doallo, “Performance analysis of HPC applications in the cloud,” *Future Gener. Comput. Syst.*, 2012.

[6] R. R. Expósito, G. L. Taboada, S. Ramos, J. González-Domínguez, J. Touriño, and R. Doallo, “Analysis of I/O Performance on an Amazon EC2 Cluster Compute and High I/O Platform,” *J. Grid Comput.*, pp. 1–19, 2013.

[7] A. Marathe, R. Harris, D. K. Lowenthal, B. R. de Supinski, B. Rountree, M. Schulz, and X. Yuan, “A comparative study of high-performance computing on the cloud,” in *Proceedings of the 22nd international symposium on High-performance parallel and distributed computing*, 2013, pp. 239–250.

[8] E. Roloff, M. Diener, A. Carissimi, and P. O. Navaux, “High Performance Computing in the cloud: Deployment, performance and cost efficiency,” in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 2012, pp. 371–378.

[9] D. Ghoshal, R. S. Canon, and L. Ramakrishnan, “I/O performance of virtualized cloud environments,” in *Proceedings of the second international workshop on Data intensive computing in the clouds*, 2011, pp. 71–80.

[10] Iozone Filesystem Benchmark. Available: http://www.iozone.org/. [Accessed: 09-Nov-2013].

[11] J. J. Dongarra, P. Luszczek, and A. Petitet, “The LINPACK benchmark: past, present and future,” *Concurr. Comput. Pract. Exp.*, vol. 15, no. 9, pp. 803–820, 2003.

[12] NAS Parallel Benchmarks. Available: http://www.nas.nasa.gov/publications/npb.html. [Accessed: 09-Nov-2013].

[13] R. Giladi and N. Ahitav, “SPEC as a performance evaluation measure,” *Computer*, vol. 28, no. 8, pp. 33–42, 1995.

[14] Amazon Elastic Compute Cloud (EC2). Available: http://aws.amazon.com/ec2/. [Accessed: 09-Nov-2013].

[15] Rackspace. Available: http://www.rackspace.com/. [Accessed: 09-Nov-2013].

[16] S. Liu, X. Huang, H. Fu, and G. Yang, “Understanding Data Characteristics and Access Patterns in a Cloud Storage System,” in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, 2013, pp. 327–334.

[17] L. Ramakrishnan, R. S. Canon, K. Muriki, I. Sakrejda, and N. J. Wright, “Evaluating Interconnect and Virtualization Performance for High Performance Computing,” *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 2, pp. 55–60, 2012.

[18] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, “Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 6, pp. 931–945, 2011.

[19] M. E. de Souza Muñoz, R. De Giovanni, M. F. de Siqueira, T. Sutton, P. Brewer, R. S. Pereira, D. A. L. Canhos, and V. P. Canhos, “openModeller: a generic approach to species’ potential distribution modelling,” *GeoInformatica*, vol. 15, no. 1, pp. 111–135, 2011.