A performance evaluation of Azure and Nimbus clouds for scientific applications

— **Source link** ↗

Radu Tudoran, Alexandru Costan, Gabriel Antoniu, Luc Bougé

Institutions: French Institute for Research in Computer Science and Automation, École normale supérieure de Cachan

Related papers:

- CloudCmp: comparing public cloud providers

- Early observations on the performance of Windows Azure

- An Evaluation of the Cost and Performance of Scientific Workflows on Amazon EC2

- Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud

- Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing

A Performance Evaluation of Azure and Nimbus Clouds for Scientific Applications

Radu Tudoran, Alexandru Costan, Gabriel Antoniu, Luc Bougé

A Performance Evaluation of Azure and Nimbus Clouds for Scientific Applications

Radu Tudoran[1], Alexandru Costan[1], Gabriel Antoniu[1], Luc Bougé[2]
[1]INRIA Rennes - Bretagne Atlantique, [2]ENS Cachan - Antenne de Bretagne
{radu.tudoran, alexandru.costan, gabriel.antoniu}@inria.fr, luc.bouge@irisa.fr

ABSTRACT

The emergence of cloud computing brought the opportunity to use large-scale computational infrastructures for a broad spectrum of scientific applications. As more and more cloud providers and technologies appear, scientists are faced with an increasingly difficult problem of evaluating various offerings, like public and private clouds, and deciding which model to use for their applications' needs. In this paper, we make a performance evaluation of two public and private cloud platforms for scientific computing workloads. We compare the Azure and Nimbus clouds, considering all the primary needs of scientific applications (computation power, storage, data transfers and costs). The evaluation is done using both synthetic benchmarks and a real-life application. Our results show that Nimbus incurs less varaibility and has increased support for data intensive applications, while Azure deploys faster and has a lower cost.

1. INTRODUCTION

Cloud computing has recently emerged as a viable alternative to the acquisition and management of hardware or software: it allows resources to be dynamically provisioned on a fine-grained, elastic basis, in a scalable way. As this model relieves users from the resource management burden, more and more applications are being ported on clouds. With multiple cloud providers and technologies emerging, developers are faced with an increasing difficulty to evaluate and compare the various offerings and to decide which model or platform to use for their applications. Considering cloud service delivery, the options are threefold. *Private clouds* are built using in-house resources from a local cluster or data center: this option favors a high degree of control on resources, but requires a certain management overhead. In contrast, *public clouds* are not restricted within an organization: multiple users typically share resources and environments for application development and deployment. These services are proposed by commercial providers using a pay-per-use model. *Hybrid clouds* combine private and public cloud infrastructures, possibly from several providers.

In all their flavors, clouds prove to be a cheap and customizable alternative to supercomputers and, moreover, they have been proven much more reliable than grids. Because of these advantages, cloud computing is gaining increasing attention for scientific applications, which typically consist of a very large number of tasks that process massive datasets to extract meaningful information. However, most current commercial clouds have been built to support web and small database workloads, which are very different from typical scientific computing workloads. Moreover, the use of virtualization and resource time sharing may introduce substantial performance penalties for data-intensive or computation-intensive scientific applications. Under these circumstances, an important step in the process of choosing a specific cloud model is to extensively *characterize the performance* of the provided services and resources.

In this paper, we evaluate the performance of public and private cloud platforms for scientific computing workloads. We assess the benefits and the drawbacks of the two models, the constraints they impose to the targeted applications and the costs they incur. Motivating this work is a typical researcher's dilemma when choosing the appropriate platform for large scale scientific experimentation. One option would be to deploy an open-source cloud on the local cluster in order to be able to fully customize the environment where the applications are running, since usually users don't have complete control on the physical nodes in the cluster. Faced with the burden of managing the open-source cloud, the alternative is represented by the public commercial clouds. Several previous studies have compared supercomputers and public clouds showing an increasing improvement in the scalability and performance of cloud-based HPC systems [5]. However, Linpack numbers do not always translate into application performance, which depends on the composition and pattern of computation, communication and I/O within each individual parallel application.

We make a subtler evaluation, by considering besides performance, other factors that impact scientific applications: ease of management, cost and performance stability. We evaluate whether the Azure commercial cloud is a feasible and cost effective alternative for offloading computation and storage resources from an in house maintained Nimbus cloud, for scientific applications. To this end, we characterize both platforms using several metrics: *data access performance, computation speed, variability* and *cost models*. We rely on synthetic benchmarks to assess the network efficiency when using several communication protocols and

models (e.g. TCP, RPC). For evaluating the computation power we use a real life genetic and neuro-imaging application. In addition, we make an evaluation of the transfer time of data from users' local storage to the cloud storage and from there to the computation instances, and vice versa, as this is an inherent step in any data-intensive application.

The contributions of this paper are:

- An overview of the issues related to running scientific applications on public and private clouds.
- A comparison between the performance of Azure, a public commercial cloud, and Nimbus, an open source cloud built on top of a private infrastructure. We make a broad-range performance analysis, that in contrast with the existing work, takes into considerations all primary needs of scientific applications (computation power, storage and data transfers). To our knowledge, no prior studies evaluated the moving of the initial and final data between the local hosts and the cloud compute and storage nodes. The evaluation is done using both synthetic benchmarks and a real-life application.
- An analysis of the cost of running applications on the Azure and Nimbus clouds.

The rest of the paper is structured as follows. Section 2 discusses the related research and the novelty of this contribution with respect to the existing work. Section 3 presents the public and private cloud models. In Section 4 we introduce the metrics used for the performance and cost evaluation presented in Section 5. Section 6 concludes this paper.

2. RELATED WORK

Much previous research has investigated the feasibility of using clouds for scientific applications. The great majority of these studies evaluated public clouds, focusing on tightly coupled, MPI-style applications. Our work differs from these in several ways. First, we consider both public and private clouds in order to assess the performance trade-offs, scalability issues and the main limiting factors for applications relying on these two solutions. Second, our study is not limited to tightly-coupled applications; the analysis we perform also considers the loosely-coupled parallel ones, with their specific requirements. Third, most previous efforts provided a limited set of performance metrics with no in-depth analysis of the detected behavior; they focused mainly on micro benchmarks such as NAS [3] or Linpack [12]. In comparison, our work makes a comprehensive examination of issues relevant to the scientific community, including performance, cost, scalability and cloud variability, illustrated by a real-world application.

There has been a recent spur of research activity in assessing the *performance of virtualized resources*, in public cloud computing environments (with a focus on Amazon EC2). In [8] the authors quantitatively examined the performance of a set of benchmarks designed to represent a typical HPC workload run on Amazon EC2. Results showed a strong correlation between the percentage of time an application spends communicating and its overall performance on EC2. However, no indication is given on the potential sources of the communication problems, although multi-tenancy and node heterogeneity could impact on them. A recent study [16] on the new EC2 Cluster Compute Instances confirms that even the use of the 10-Gigabit Ethernet interconnection, rather than the traditional Gigabit one, still dramatically affects applications handling a lot of small messages. However, the authors notice that for many typical HPC workloads, the cloud performance relative to that using in-house InfiniBand-connected clusters justifies the option of going for clouds as cost-effective, unless the inhouse cluster achieves a rather high utilization level, which is often not the case. Also close to our work are the early observations on Windows Azure [6], which include a performance evaluation of its compute and storage resources. In contrast to this body of previous work, ours is broader in scope: besides the commercial public clouds, usually used as black boxes, we are interested in the performance of private clouds, deployed on owned and controllable infrastructures. We compare the performance of Windows Azure with a private, Nimbus-based, cloud by considering a broader set of metrics (e.g. cloud variability, cost analysis).

A few prior efforts have investigated *data sharing and storage* issues in cloud environments. Experiments with several storage and file systems deployed on the Amazon EC2 clouds are compared in [9] to assess their suitability for executing task workflows. As expected, the cost closely follows performance. Palankar et al. evaluate Amazon S3's ability to provide storage support to large-scale science projects from a cost, availability and performance perspective [13]. However, their experiments are limited to the performance of downloading data, which is only relevant for a subset of cloud applications. In [11], the authors use an open source IaaS cloud, Eucalyptus, running on private infrastructures to examine the performance of the Gfarm filesystem. Evaluations with micro benchmarks and with MapReduce applications showed scalable I/O performance when compared to other file systems. To our best knowledge, our work is the first to evaluate and compare the I/O performance of both public and private platforms; we complement the existing studies focused mainly on throughput evaluation with considerations on staging in and out the data.

3. CLOUD DEPLOYMENT MODELS

Scientists adopt different deployment models based on their particular requirements. This casts the differences between private and public clouds into a clear perspective. The adopted model can be sometimes more complex, as a user owning a private cloud can acquire public cloud resources to form a hybrid cloud; two or more private clouds could also interact for common goals and thereby form a federated cloud. In this section we give a brief overview of public and private clouds and their impact on scientific applications.

3.1 Public clouds

The public clouds provide compute and storage resources on demand, using a pay-as-you-go model. They offer wide availability and economies of scale that only very large data center operators can achieve. Several commercial offerings exist from providers like Microsoft, Amazon or Google. Building on their elasticity, scientists have started to migrate applications in the public clouds. These allow users to customize the environment and replicate experiments through the use of VM images, while proving an efficient collaboration tool that enables the global sharing of information.

The compute and storage nodes are usually distinct, as the cloud storage is a stand-alone system providing persistency to applications. The storage is accessed using HTTP based protocols and consists of binary large objects with sizes in

the orders of GBs, that are usually structured in a higher level namespace (buckets, containers). The main issues for the scientific applications arise from the impact of the data transfer protocols on the throughput, size limitations on the data objects, the coarse-grain access and the little support for heavy concurrency (e.g. no or little versioning support).

While commercially available public clouds could be used for scientific computing, they were not designed with such applications in mind. Several studies presented in the previous section showed that many public cloud platforms have slow network connections between the VMs, which often becomes a bottleneck for running some scientific applications. Also, the multitenancy model of the cloud implies that applications from different users can share the same physical machine and the network infrastructure, leading to performance variability.

Table 1: Azure virtual machines (VM) specifications

VM Size	CPU Cores	Memory	Disk
ExtraSmall	shared	768 MB	20 GB
Small	1	1.75 GB	225 GB
Medium	2	3.5 GB	490 GB
Large	4	7 GB	1000 GB
ExtraLarge	8	14 GB	2040 GB

In this paper, we choose a Platform-as-a-Service public cloud, Microsoft's Azure. It allows scientists to install applications, customize the computing environment and it provides a comprehensive API for building applications: role based workers, message queues, tables, blobs etc. The computing service of Azure is composed of two types of nodes: web and worker roles, typically used by scientific applications as a master-slave / replicated workers model. The provided resources are described in Table 1: there is a one to one relationship between the virtual cores and the physical cores and an Extra Large VM acquires the whole physical machine. The storage service of Azure is composed of 3 systems: BLOBs, Tables and Queues. This work will concentrate on evaluating the Azure BLOBs as this is the default mechanism to store large sets of scientific data.

3.2 Private clouds

Open-source cloud frameworks such as Eucalyptus, Open-Nebula and Nimbus [10] are designed to allow an organization to set up a private group of machines as their own cloud. Private clouds are typically deployed behind a firewall where access is controlled and the target users are known. Hence, the primary distinction between private and public clouds may be more of a relative distinction concerning the ownership of resources and the ability to enforce security policies, etc., versus delegating those responsibilities to a second party. For scientific applications, users can create arbitrary size clusters of cloud servers. Private clouds provide an integrated set of tools that deliver the power and versatility of infrastructure clouds to scientists. Their implementation specifically targets features of interest to the scientific community such as batch schedulers, best-effort allocations, increased security through credentials, etc.

We chose Nimbus as a representative private cloud since it was designed to address the needs of the scientific community in multiple contexts, ranging from high-energy physics to bioinformatics. Nimbus is an open-source Infrastructure-as-a-Service cloud platform, which enables clients to lease a set of virtualized computational resources and to configure customized environments. The architecture of a Nimbus cloud is based on several modular components, on top of which new modules are added to enable easy cluster configuration, interfaces to other IaaS Clouds or building PaaS clouds. The Workspace Service provides an implementation of a compute cloud allowing users to lease computational resources by deploying VMs. A complementary tool, Cumulus, provides an implementation of a quota-based storage cloud, allowing multiples implementations and playing the role of the front end to the Nimbus storage repository for VM images. Nimbus is an interesting option for scientists since it provides functionality well beyond homemade scripts: it offers a structure and APIs for launching, configuring, and monitoring applications.

4. PERFORMANCE METRICS

We have selected several metrics, structured according to the execution phases of scientific applications. A first step consists of deploying the customized environment and fetching the initial data. In a second phase, the application is executed, so we are interested in the computation performance and the efficiency of the network transfers. We then give some considerations about the estimated costs.

4.1 Staging data

Before actually executing an application in a cloud, one has to acquire the nodes, set the environment and prepare the data to be fed into the application. One time consuming step consists of copying the data from the local storage (exterior to the cloud) to the cloud storage and from there to the computation nodes (and vice-versa for the final results). Our first metric measures these times to move data to and from the cloud. Such a metric is relevant for data intensive applications like the ones that have to index huge collections of documents or process massive data sets.

4.2 Tracing applications' performance

This can be done by investigating two main components. On one hand, the *computation power* is influenced by the CPU and memory usage. On the other hand, the *data transfers* between the VM nodes are affected by the underlying network and the existing traffic.

Computation speed. To assess the performance differences between Azure and Nimbus, we used A-Brain, a reference real-life application, that joins neuro-imaging and genetic analysis [14]. The application compares brain regions of MRI images with genes for finding significant links between the two. This application is representative for a large class of scientific workloads that split an initial large domain into subdomains, perform the computation (e.g., matrix operations) and combine the final results (e.g., image processing, weather simulations, etc). A-Brain reads data from the local storage into memory, stressing the caching mechanisms of the machine, performs floating point operations on large matrixes and then writes the results back to the local storage. We track readings from the CPU, memory usage and virtual disk I/O to have a clear picture on how the application performs on Azure and Nimbus. We further take into account how the multitenancy model influences the computations and investigate if there is a way around it.

Data transfers. It is interesting to observe the impact of the network performance, considering that in private settings user applications typically run in isolation, while on public clouds they share the network infrastructure. We use a synthetic benchmark in which we transfer large sets of

random data between the cloud computing instances. Since TCP is the default underlying protocol for data transfers we examine its performance in both settings. We also noticed that many scientific applications rely on RPC to implement their functionalities, and therefore evaluated the overhead introduced by this communication paradigm in clouds. Again, we take into account the sustained network performance during long time intervals, since variability can impact the overall performance of applications [16].

4.3 Cost

Computing the costs of resources in public clouds is straightforward using the lists of pricing for the charged services (hours of computation, storage capacity in time, transfer of data etc.). However, when it comes to computing the costs in private clouds things are more complex. From the scientist's perspective, the cost is zero, as he usually has free access to the local cluster. To make a fair comparison with the public clouds, in this paper we consider the infrastructure owner's perspective. The cost of the hardware (servers, racks, network switches etc.), the energy cost (consumed power and cooling) and the human costs for the management make up a total cost from which one can derive a "virtual" price per hour, taking into account some reference period and the utilization level of the platform.

5. EVALUATION AND RESULTS

In this section we evaluate the Azure and Nimbus clouds with respect to the previous metrics. We then provide a cost-estimation of running scientific applications in a private cloud and compare it with the costs of commercial clouds.

5.1 Experimental setup

For our evaluation we used 40 nodes in the Azure cloud, half of them with Small instances and the others with ExtraLarge instances (occupying the entire physical machine). We chose these two types of instances in order to better understand the multitenancy model of Azure. The interconnection network between the machines is Gigabit Ethernet. We also created a Nimbus cloud on top of the Grid5000 infrastructure and executed the synthetic benchmarks and the ABrain application inside the VMs deployed in the cloud. We used 40 nodes belonging to the *griffon* cluster, as it is equipped with a large amount of memory and has 8 cores per node (similar to the ExtraLarge machines used in Azure). These nodes provide an efficient support for virtualization and each of them can host several VMs while being used as Workspace control agents. To install and deploy Nimbus, we used a set of Ruby scripts that take advantage of the Grid5000 API to deploy one's own customized operating system image. Intracluster communication is done through a Gigabit Ethernet network as well.

5.2 Cloud deployment and data fetching

A first step for running applications in the clouds consists of setting the environment. For public clouds this means that a certain number of resources are leased and the targeted application is deployed on them. In Azure, we observed 10-20 minutes startup times, which is consistent with the observations of [6]. For private clouds, two steps are required. First, the cloud environment is deployed on the local infrastructure; the observed time for this operation with Nimbus on Grid5000 is about 15 minutes. Next, the VM

Table 2: Upload/Download time (seconds) from local to cloud storage (L-Local, AB- AzureBlobs, C - Cumulus)

Size GB	Azure				Nimbus			
	L→AB		AB→L		L→C		C→L	
	μ	σ	μ	σ	μ	σ	μ	σ
0.125	16,41	0,378	12,43	0,456	12,06	0,094	12,93	0,340
0.25	31,57	0,167	31,57	0,139	24,32	0,154	25,41	0,445
0.5	63,84	0,358	82,52	2,963	48,31	0,623	51,86	0,654
1	129,90	1,788	90,00	5,137	92,30	0,309	107,44	0,454
10	1288,1	2,237	1176,5	5,318	769,59	2,121	854,25	3,245
25	3281,7	44,48	3284,7	5,734	2043,43	11,232	2247,32	14,211
100	13207	336,4	10409	13,26	8152,43	102,12	8798,12	95,214

images containing the targeted environment are uploaded to the Cumulus repository. Finally, the VMs are provisioned within the cloud; this takes several minutes (e.g., 10 minutes for a small cluster with 10 machines and a 2GB VM image).

The second step consists of moving the data to be processed from the users' local storage into the cloud. To do this, data is first transfered to some cloud accessible storage (e.g., the cloud storage service if available), which is a one time operation. Then, for each run of the application, data is transfered from this storage to the computing nodes. Once the processing is finished, the data follows the reverse path. The cloud storage services were the AzureBLOBs in Azure and the Cumulus repository in Nimbus.

In a first series of experiments we transferred data sets of increasing sizes between the local storage and the cloud storage. We repeated each experiment 20 times and computed the mean (μ) and the standard deviation (σ) for the transfer times in Table 2. In the case of Nimbus this time is the sum of two components: transferring data to the grid frontend and from there to Cumulus, since Grid5000 nodes are not directly accessible from the exterior. We noticed slightly smaller times for the private cloud, which is expected due to proximity. The standard deviation also shows an increased upload stability of the private infrastructure, normal because of the isolation. The transfer times are in the same order of magnitude for both platforms, which is due to several limiting factors. For both, the read/write rate of the disks on the local machine represents a bound for the transfer rates. For AzureBLOBs, the rate to download the data is higher than the rate for uploading, as opposed to the private infrastructure. AzureBLOBs use strong consistency [4], so the write operation completes only after all replicas (3) are written. Cumulus is not using implicit replication, so it finishes the write operation faster.

Next, we measured the times to transfer data between the cloud storage and the computing nodes (Table 3). We noticed that uploading data to the compute machines is faster on the private infrastructure, due to the higher, unshared bandwidth available. Regarding the two types of VMs tested in Azure we see a gain in performance and stability for ExtraLarge VMs for downloads only. We argue that this is due to the exclusive access to the network interface and the additional memory available that allows larger network buffers to manage incoming data and.

5.3 Computing performance

In Figure 1 we show the evolution of the running times for the neuro-imaging application over several time intervals, during different hours and days within a week. For each time interval we tracked several thousands samples, one minute apart, by running the application on a fixed dataset stored locally. As it can be seen, the nodes from Nimbus deliver the best performance. This is due to the fact that public clouds

Table 3: Upload/Download times (seconds) from cloud storage to the computing nodes (S - Small, EL - ExtraLarge)

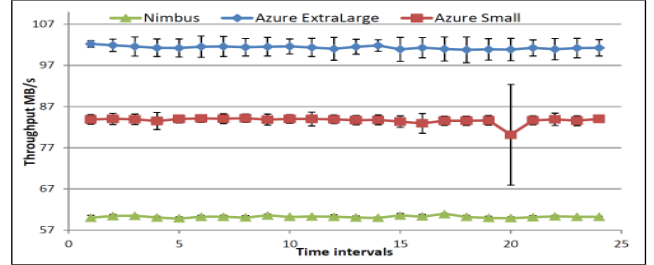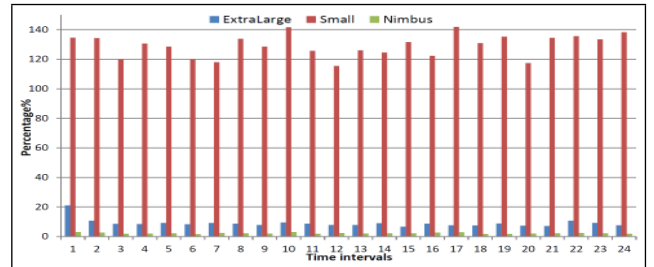| Size MB	Azure								Nimbus			
	S→AzureBlobs		AzureBlobs→S		EL→AzureBlobs		AzureBlobs→EL		VM→Cumulus		Cumulus→VM	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
128	17,20	0,099	13,18	0,056	17,04	0,656	12,91	0,656	12,62	0,318	14,13	0,198
256	34,87	0,587	26,92	0,532	33,65	0,273	25,54	0,132	29,29	0,502	29,17	0,152
512	71,47	3,210	63,68	3,872	68,10	0,631	66,32	3,774	62,17	1,490	64,08	0,225
1024	147,34	1,305	121,79	3,685	135,79	0,837	114,93	1,547	131,75	0,856	133,94	0,954
10240	1439,01	0,584	1224,84	10,95	1395,81	17,59	1036,21	1,426	1222,32	6,325	1322,54	4,365
25600	3585,12	51,64	3085,75	8,655	3505,59	55,33	2563,18	5,243	3010,32	11,32	3102,32	8,547
102400	14087,1	200,3	12715,5	23,02	14161,8	48,54	10233,7	16,17	12023,32	24,21	12435,21	23,21
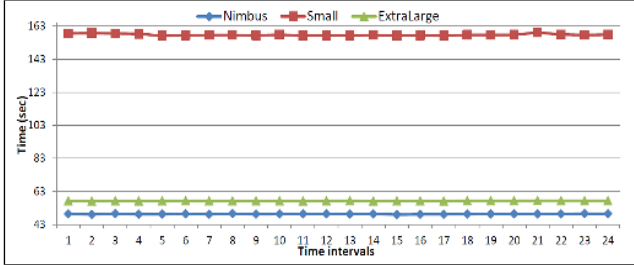
Figure 1: Evolution of the application execution in time
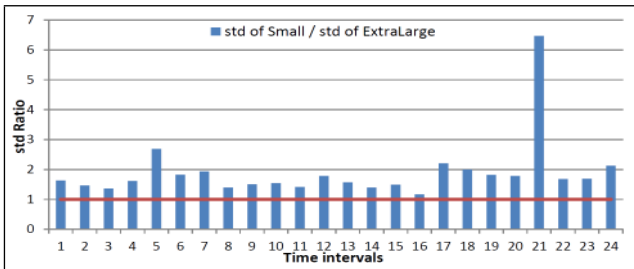
Figure 2: Evolution of the standard deviation of the Small instances with respect to the ExtraLarge ones

Figure 3: Evolution of the TCP throughput in time for transferring 512MB

Figure 4: Coefficient of variation for RPC(HTTP)

typically rely on commodity hardware, which is less efficient for scientific computation than the high end hardware used in private infrastructures like Grid5000. The standard deviation of all samples is 0,242 for Nimbus, 0,523 for ExtraLarge and 1,207 for Small, confirming the stability of Nimbus.

For better understanding the variability due to multitenancy in Azure we propose the analysis in Figure 2. We assume that there exists a "default" variability that is due to the OS and to the virtualization mechanisms. Since the ExtraLarge VM occupies the entire physical machine and all computations are done locally, we consider that the standard deviation measured for the ExtraLarge VMs approximates this default variability. Hence, in Figure 2 we represent the ration between the standard deviation measured on the Small instances with respect to the reference one. The goal of this analysis is to see the fairness of sharing the physical nodes among VMs belonging to different users. A value of the ratio close to 1 (in red) would mean that either the neighboring applications do not affect our performances, or that there are no neighboring applications, as we can see for the 16th time interval. A high value for the ratio would indicate that the performance of the instance is affected by external factors, as is the case for the 5th and 21th interval. The reason for observing both small and big values is that the samples (2880) were done in a time span of approximately 1 week, one minute apart, a time span that proved sufficiently large to capture various situations. The interferences that affect the instances are caused by the local disk I/O or by memory accesses (we don't include the CPU since there is a 1 to 1 relation between virtual CPUs and physical cores). Such an analysis could be used as a hint for sched-

ulers when distributing the workload among the instances.

5.4 Intra-cloud network

Our next experiments examine the TCP throughput as a measure of path capacity because TCP is the dominant traffic type of cloud applications. Path capacity impacts data transfer throughput and congestion events can lead to errors or delayed responses. We have done several thousands transfers of 512MB of data, with one minute delay between them, within a period of a week and we grouped them into 24 time intervals. In Figure 3 we present the throughput of the TCP communication and its variation in time. The public cloud delivers an almost double performance although with a higher variability. Nimbus is penalized by the network virtualization approach implemented for deploying cloud clusters. However, it should be taken into account that when it comes to private infrastructures, users might have the option of choosing a better interconnectivity (10 Gigabit Ethernet, InfiniBand), which is not available in public clouds like Azure. Naturally, not sharing the network card significantly increases the throughput of the ExtraLarge instances.

The next series of experiments evaluate the overhead introduced by RPC with HTTP as the underlying protocol; this communication paradigm is used by a large class of scientific applications. We transferred repeatedly (1440 times) 128MB of data between several nodes, with a 1 minute pause between successive transfers. In order to assess the incurred variability we compute the coefficient of variation $cv = \frac{std}{mean} * 100$, depicted in Figure 4. Not surprisingly, Nimbus proved to be very stable for such transfers. On the other hand, due to the limitations imposed by Microsoft
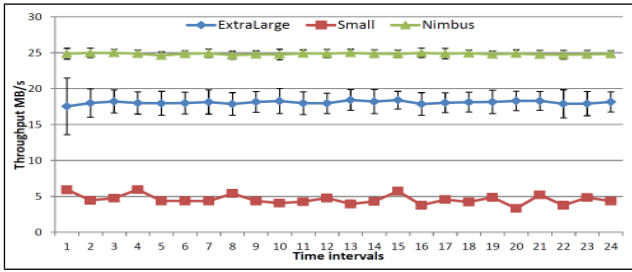
Figure 5: Evolution of the RPC (HTTP) throughput in time for transferring 128MB of data

mentioned earlier, the Small VMs exhibit a huge variability, which is not the case for the ExtraLarge ones.

In Figure 5 we illustrate the throughput for the RPC communication. Due to the magnitude of the standard deviation for the Small machines, we chose not to represent it. Unlike for TCP, Nimbus delivers up to 39% better performance for such type of communication. This confirms the existence of some software control on the HTTP traffic in Azure, for fairness reasons. As HTTP is used to access the Azure storage service, unrestricted data transfers from VMs to AzureBlobs could temporary slow down neighboring applications.

5.5 Cost analysis

In order to compare the costs of running applications on the two platforms, we estimate the price per hour of computation in a private infrastructure and compare it to the price listings of Azure. The cost of an owned infrastructure has several components: hardware infrastructure, human resources and electrical power. It is hard to obtain exact values of these for a private infrastructure like Grid5000. However, we derive an estimate based on official values of the hardware and personel costs in conjunction with the history of utilization [2]. The Grid5000 report [1] estimated the total cost (without electricity) of the platform for 3 years at 15 million euros. The total number of cores is 7469 with an average utilization of 85% per year. Hence, the cost of one hour of computation for a core can be computed as follows:

$$Cost_{infrastructure} = \frac{15000000}{3*24*365*0.85*7469} = 0.0899 \; euros/h$$

For computing the electricity cost we used Hamilton's methodology [7] and the work of Wang et al [15]:

$$Cost_{electricity} = p * P_{raw} * PUE \; euros/h,$$

where p is the electricity cost, and we take as reference the value of 0.11 euros / kWh. P_{raw} is the energy consumption of one core per hour and it will be substituted with 0.046 kWh as reported by Wang. For PUE, a metric for measuring the efficiency of power distribution of equipments, we consider the value 1.7 which is taken from the Hamilton's parameter setting. Thus, the power cost per hour is: $Cost_{electricity} = 0.0086 \; euros/h$, resulting in a total cost per hour of: $Cost_{total} = 0.0985 \; euros/h$. Commercial cloud pricing for a computation hour is straightforward. Microsoft Azure charges 0.0852 $euros/hour$, which is 13.5% cheaper.

6. CONCLUSIONS

In this paper we systematically compared the performance and cost of two public and private clouds (Azure and Nimbus) along several dimensions that matter to scientific users. We observe dramatic performance and cost variations across platforms in their virtual instances, storage services, and network transfers.

As most users, scientists prefer the ownership of virtual resources, since this reduces the uncertainty concerning access when needed. Private clouds like Nimbus incurred less variability and showed a better computing performance due to the underlying hardware. Arguably, they are the first option for data intensive applications since they incur lower data staging times, due to proximity, and have good support for RPC. On the other hand, public clouds like Azure have decreased costs and deployment times and support efficient TCP data transfers. We believe that this work represents a significant step towards enabling an informed cloud model selection for the emerging scientific HPC applications.

7. ACKNOWLEDGEMENTS

8. REFERENCES

[1] Grid5000 Plate-forme de recherche experimentale en informatique. Technical report. www-sop.inria.fr/aci/grid/public/Library/rapport-grid5000-V3.pdf.

[2] Grid5000 Platform Status. https://www.grid5000.fr/mediawiki/index.php/Status.

[3] NAS. http://www.nas.nasa.gov/publications/npb.html.

[4] B. Calder, J. Wang, A. Ogus, and et al. Windows azure storage: a highly available cloud storage service with strong consistency. In *SOSP '11*, pages 143–157, 2011.

[5] Q. He, S. Zhou, B. Kobler, D. Duffy, and T. McGlynn. Case study for running hpc applications in public clouds. In *HPDC '10*, pages 395–401, 2010.

[6] Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey. Early observations on the performance of windows azure. In *HPDC '10*, pages 367–376, 2010.

[7] J. Hamilton. http://perspectives.mvdirona.com/2008/11/28/CostOfPowerInLargeScaleDataCenters.aspx.

[8] K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. Wasserman, and N. Wright. Performance analysis of high performance computing applications on the amazon web services cloud. In *CloudCom 2010*, pages 159 –168, 2010.

[9] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B. Berman, and P. Maechling. Data sharing options for scientific workflows on amazon ec2. In *SC'10*, pages 1 –9, 2010.

[10] K. Keahey and T. Freeman. Science clouds: Early experiences in cloud computing for scientific applications. In *Cloud Computing and Its Applications 2008*, 2008.

[11] K. Kobayashi, S. Mikami, H. Kimura, and O. Tatebe. The gfarm file system on compute clouds. In *IPDPSW 2011*, pages 1034 –1041, 2011.

[12] J. Napper and P. Bientinesi. Can cloud computing reach the top500? In *Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop*, pages 17–20, 2009.

[13] M. R. Palankar, A. Iamnitchi, and M. Ripeanu. Amazon s3 for science grids:a viable solution? In *Proc. of the 2008 Intl. W. on Data-aware Distributed Comp.*, pages 55–64, 2008.

[14] J.-B. Poline, C. Lalanne, A. Tenenhaus, E. Duchesnay, B. Thirion, and V. Frouin. Imaging genetics: bio-informatics and bio-statistics challenges. In *19th International Conference on Computational Statistics*, Paris, France, Aug. 2010.

[15] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou. Distributed systems meet economics: pricing in the cloud. In *HotCloud'10*, 2010.

[16] Y. Zhai, M. Liu, J. Zhai, X. Ma, and W. Chen. Cloud versus in-house cluster: evaluating amazon cluster compute instances for running mpi applications. In *SC '11*, pages 11:1–11:10, 2011.