Performance Evaluation of Multiple Cloud Data Centers Allocations for HPC

Eduardo Roloff[1], Emmanuell D. Carreño[1], Jimmy K. M. Valverde-Sánchez[1], Matthias Diener[1], Matheus da Silva Serpa[1], Guillaume Houzeaux[2], Lucas M. Schnorr[1], Nicolas Maillard[1], Luciano Paschoal Gaspary[1], and Philippe Navaux[1]

[1] Informatics Institute,
Federal University of Rio Grande do Sul - UFRGS
Porto Alegre, Brazil
{eroloff,edcarreno,jkmvsanchez,mdiener,msserpa,schnorr,nicolas,paschoal,navaux}@inf.ufrgs.br

[2] Dpt. Computer Applications in Science and Engineering
Barcelona Supercomputing Center (BSC-CNS)
Barcelona, Spain
{guillaume.houzeaux}@bsc.es

Abstract. This paper evaluates the behavior of the Microsoft Azure G5 cloud instance type over multiple Data Centers. The purpose is to identify if there are major differences between them and to help the users choose the best option for their needs. Our results show that there are differences in the network level for the same instance type in different locations and inside the same location at different times. The network performance causes interference in the applications level, as we could verify in our results.

Keywords: Cloud Computing, HPC, Azure, MPI, NAS

1 Introduction

Cloud Computing offers an interesting alternative for High Performance Computing (HPC) applications, due to the pay-per-use cost model and the elasticity [7] to provide any amount of resources in little time. However, due to the virtualized environment, there are some aspects of the Cloud that still remain as a barrier for the large adoption of Cloud Computing by the HPC community. It is clear that the CPU virtualization is not a problem, because the CPU performance in the Cloud is the same as in a traditional machine. Memory accesses and disk I/O are in an earlier stage of development to be used in the cloud, but they do not represent a big issue at this time. The main bottleneck of Cloud Computing is the network performance, a very important aspect for HPC.

In this paper, we provide an extensive evaluation of the network performance in the Microsoft Azure public Cloud. Since MPI is an important standard for HPC communication [10], we evaluate its performance using three different communications patterns: Single Transfer, Parallel Transfer and Collective Communications. We used the same type of virtual machine (VM) instance among four different Azure Data Centers and used the machines at different times, during working hours and during the night. The purpose was to verify if the time of the day of each allocation causes a performance impact in the machines and if they have different performance levels among different Data Centers. We used a traditional cluster as a baseline for comparison purposes.

Our results shown that the execution during the night has lower performance than when executed during the day. We also conclude that there a slightly difference in the application performance when compared the execution times between the different Data Centers.

2 Motivation

The Cloud Computing model offers an interesting alternative as an environment for HPC applications, due to the pay-per-use cost model and the elasticity of resources. The public Cloud could provide any amount of resources in little time, without upfront costs. Theoretically, when the user sends his applications and data to the cloud, they could be stored anywhere on earth, the user does not have control over this. Moreover, the major cloud providers give the user the option on which Data Center location the application and data will be stored. This is necessary because there are some situations where the user needs to know and decide where his application is, due to regulations or data confidentiality.

However, the same VM instances could present different performances when executing in different locations. This could be caused by the different behavior of the users of the Data Center, more or less load, or even by the Data Center configuration itself. There is a lack of research that compares the same VM instances among the same provider.

Our proposal is to provide a comparison among different Data Centers to verify if they present significant differences when executing the same application using the same type of VM instance. This is important to help the user that could execute his application anywhere as well as could help the user with location restrictions. We intend to help the users to choose the machines and locations with the best performance among all available in Microsoft Azure.

3 Methodology

This section describes the hardware and software environments as well as the MPI and NAS benchmarks that were used in our evaluation. The scientific HPC application used is explained as well.

3.1 Cluster and Cloud Environments

We performed experiments on one traditional cluster system as well as four Data Center locations of Microsoft Azure using the G5 VM instance. The G5 instance is a VM with 32 cores, composed of a E5-2698v3 CPU running at

Table 1. Configuration of the cluster and cloud environments used in the experiments.

Machine name	Processor model	Freq.	Cores per instance	Network	Location	Price/hour ($) for all instances
Econome	E5-2660	2.2 GHz	16	10 Gbit/s	France	—
G5	E5-2698 v3	2.3 GHz	32	—	4 DCs	69.52

2.3 GHz with 448 GB of RAM, there is no precise information about the network interconnection. The traditional cluster is the *econome* machine from *GRID 5000* and is composed of two 8-core processors, the network interconnection is 10 Gbit Ethernet.

In all environments, we create systems with 128 cores in total to maintain a comparable baseline. The total number of nodes were four, for the G5 machines, and 8 for the econome cluster. The locations of Microsoft Azure used were: West Europe (WEU), West USA (WUS), East USA (EUS) and Southeast Asia (SAS). To the best of our knowledge, all systems are running without Hyper-Threading. All environments use Intel processors of recent generations, at least the Sandy-Bridge family.

Table 1 contains an overview of the machines used in the evaluation. Although main memory sizes vary between different instance sizes, all amounts were sufficient for our experiments and are therefore not mentioned in the table.

All the tests were executed using two allocations in the cloud to compare the differences among the day. We allocate the machines and executed the tests around 2 AM and 2 PM on business days. The cluster was evaluated just once, because it consists of isolated machines that did not show significant variability during the day.

3.2 Intel MPI Benchmarks

We use the Intel MPI Benchmark communication tests. This benchmark allows us to measure the performance of the most important MPI functions. There are three classes of benchmarks named single transfer, parallel transfer and collective benchmarks. We have selected the PingPong benchmark of the single transfer class, this benchmark entails just two process into communication. The Sendrecv of the parallel transfer class was used, this is based on the MPI_Sendrecv function. For the Sendrecv, each process of a periodic communication chain sends a message to its right neighbor and receives one from its left neighbor. For the collective benchmark, the Reduce and AllToAll were used, the first based on the MPI_Reduce function performs a reduction operation on all processes, and the second based on the MPI_AllToAll function which is a data movement operation, where each process sends data from all to all processes [4]. Each one of the experiments was performed with different message sizes, 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, and 32768 bytes.

3.3 NAS

The NAS Parallel Benchmarks (NPB) are a set of benchmarks developed to help evaluate the performance of parallel environments. The benchmarks are derived from computational fluid dynamics (CFD) applications and consist of nine applications with different needs. They cover all major aspects of parallel systems. We used the MPI version of NAS.

3.4 Alya

Alya is a simulation code for multi-physics problems, based on a variational multi-scale finite element method for unstructured meshes. It is used in areas, such as wind energy, aerospace, oil and gas, biomechanics and biomedical research, environment and automotive industry, among others. Developed at Barcelona Supercomputing Center, written in Fortran 90/95 combining MPI and OpenMP. Parallelization of the work is mainly performed using MPI, the original mesh is partitioned into sub-meshes that are executed for MPI processes [8, 9].

4 Results

We classified the results of our experiments into two parts. The first subsection has the MPI results, to analyze the network performance. The second subsection has the applications results, then we could analyze the performance of the NAS benchmarks and the application Alya.

4.1 MPI Benchmarks

The MPI results are divided into three different groups: Single Transfer, Parallel Transfer, and Collective Communications. The single transfer results shows the measured performance between two nodes. The parallel transfer shows the results of all the nodes communicating at the same time. Finally, the collective communications results show the behavior of MPI collective operations. These three groups cover the majority of communication patterns used in HPC applications.

Single Transfer Single Transfer tests are communication between two different processes, all other processes in the cluster wait. We executed the PingPong test from the Intel MPI benchmarks, running each process in a different machine. The purpose was to identify the network performance of a point-to-point communication without interference of other communications. We present results for both Latency and Bandwidth of this test.

Figure 1 shows the latency results of the PingPong test. The line in the lower part of the Figure show the cluster results, as we can see there is practically no latency when varying the package size from 0 bytes to 32 KB. In the other hand,

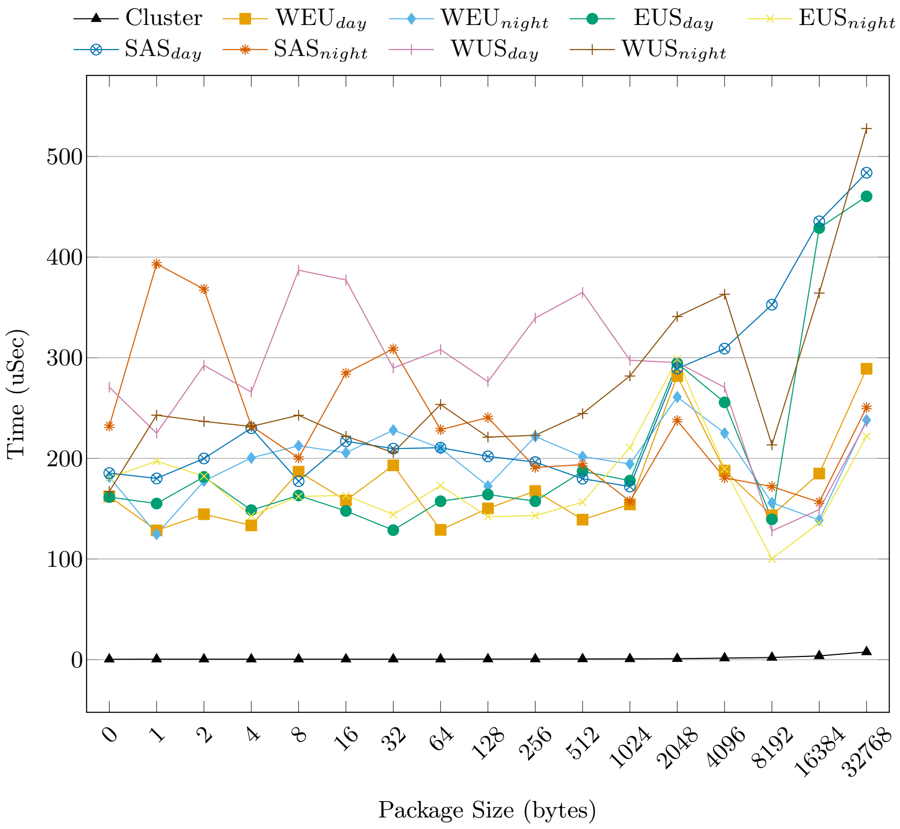

Fig. 1. Latency Results for PingPong benchmark.

we could conclude that latency in the cloud are less predictable, because there is no such clear tendency for all cloud. In some cases, when the package size was increased, the latency in the cloud decreased and we expect the opposite, using the cluster results as the baseline. Almost all the clouds have a spike when the package size was changed to 2KB, this could mean that in the cloud infrastructure exists some kind of network optimization for smaller packages. Most of the cloud instances showed the same pattern, with acceptable variability. However the EUS day, SAS day and WUS night executions exhibited some undesirable high latency for the 16 KB and 32 KB packages sizes.

Figure 2 shows the bandwidth results for the PingPong test in logarithmic scale. We could observe that in this case, the cloud instances have the same pattern with little variation between them. The growth of the bandwidth usage is following the pattern of the cluster as well. However, the instances were able to achieve a bandwidth (for package size of 32 KB) of just 140Mb/sec and the cluster achieved a bandwidth of 4,248Mb/sec. This points out the network bottleneck of the cloud compared to physical clusters. Despite performance itself,

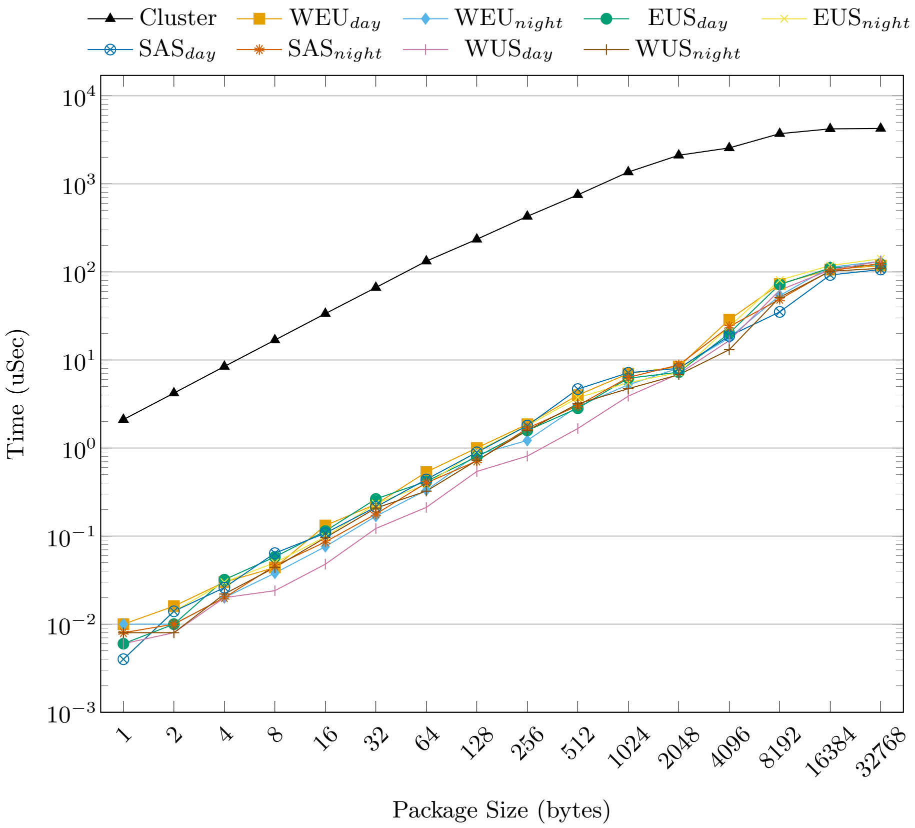

Fig. 2. Bandwidth Results for PingPong benchmark.

the user could use the predictable pattern of the cloud network bandwidth to create an application to take advantage of this characteristic.

Parallel Transfer Parallel Transfer tests measure the communication between more than two processes, in our case we used one process per node. We executed the SendRecv test from the Intel MPI benchmarks. With this test, we are able to identify the network performance when the network has a much higher utilization rate than on the Single Transfer test. We show both Latency and Bandwidth results for SendRecv test.

Figure 3 shows the latency results for the SendRecv test. It is possible to verify that the cluster latency is slightly different from the PingPong test, because in this test the latency has a small increase when the package size increases. This behavior could mean a level of network contention, the reason could be that this test performs a lot more concurrent communication in the network. On the other hand, the cloud results are better than in the PingPong test, they showed less

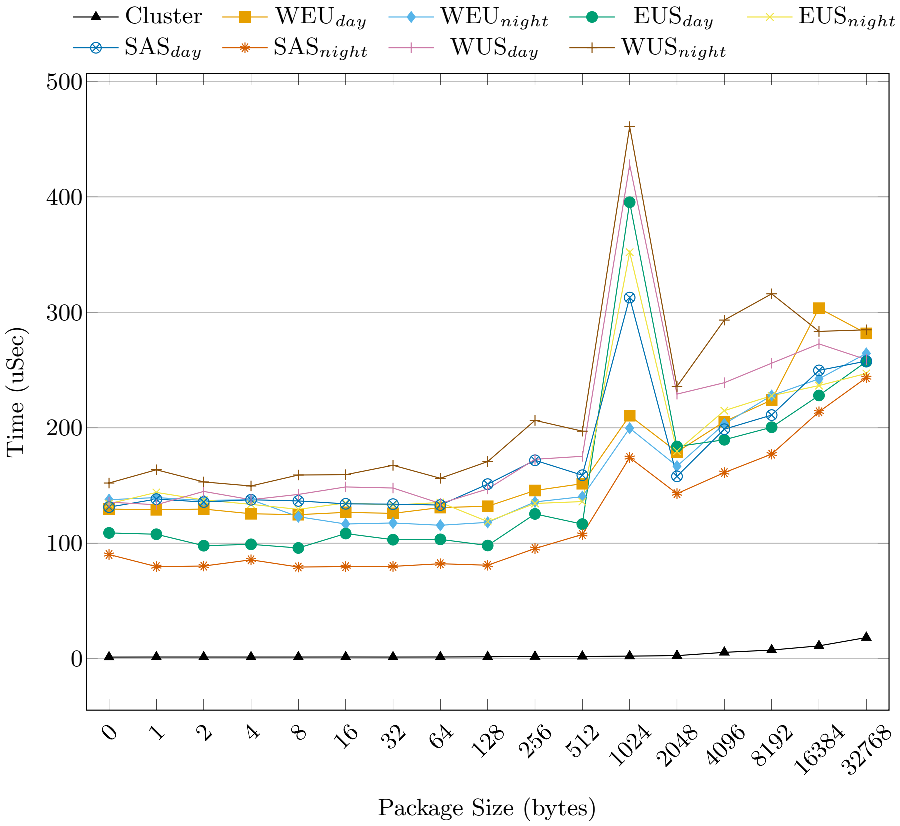

Fig. 3. Latency Results for SendRecv benchmark.

latency and a more predictable behavior, all the cloud instance allocations displayed the same pattern. It is interesting to note that all of the cloud allocations present a spike when the package size is 1KB, and then all of them return to the standard pattern. This could be explained for a possible SDN network configuration. The network is configured to handle a certain number of bytes at same time, for optimization, and when this number is reached the switches need to go to the controller to get a new configuration. This took same time, then the latency increases a little and in the next interaction, with the new configuration, the latency returns to the normal behavior.

Figure 4 shows the bandwidth results of the SendRecv test in logarithmic scale. As in the PingPong test, the cloud instances have the same pattern of increasing the bandwidth when the package size increases. We could observe that we have a small decrease of the bandwidth when the package reaches 1 KB. This remarks the explanation of the latency behavior with the same package size. The bandwidth achieved by the cluster was 3,451Mb/sec when the cloud allocations were around 250 Mb/sec for a 32 KB package. Comparing these numbers with

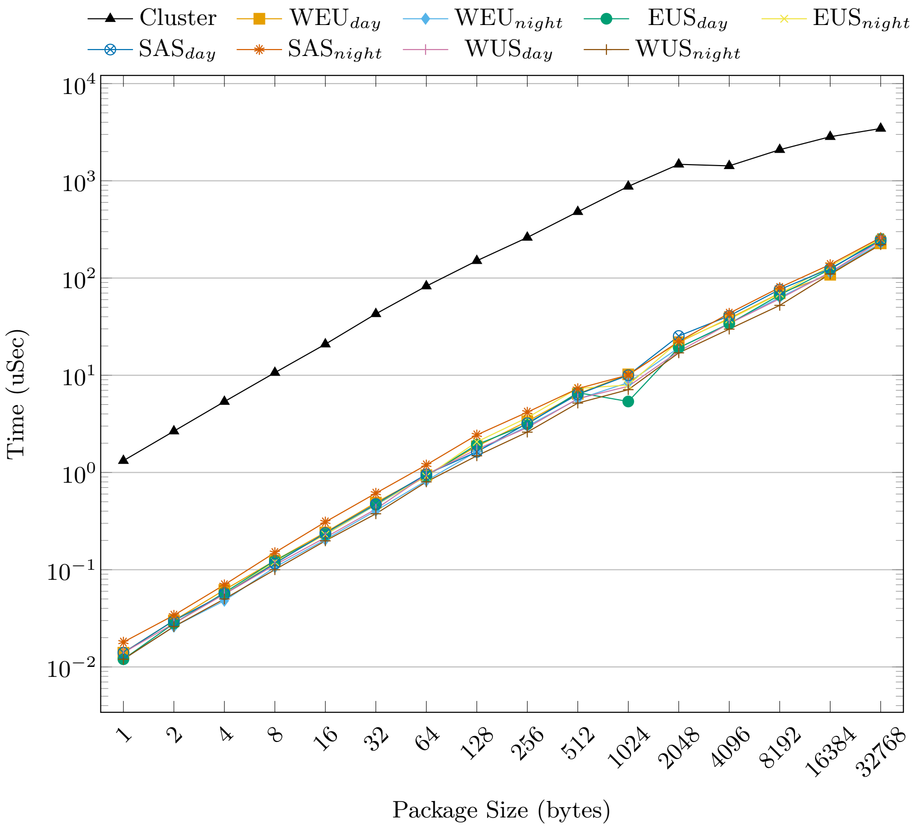

Fig. 4. Bandwidth Results for SendRecv benchmark.

the PingPong test, we could observe that the cluster achieved a lower bandwidth in this test and the cloud allocations attained a higher bandwidth. This indicates that the cloud network scales better than the cluster network when there is more communication in the network.

Both the predictable latency behavior and the bandwidth increase of the cloud allocations could benefit the user when configuring his application to be executed in the cloud.

Collective Communications The Collective Communications tests are designed to measure the performance of the MPI collective operations. There are several collective operations in the MPI standard, due to space restrictions we present the results of the Reduce and AlltoAll tests from Intel MPI Benchmarks.

Figure 5 shows the results of the Reduce test, it measures the performance of the MPI_Reduce operation. The results are displayed in time, showing the average time of an operation. We could observe that the cluster has a time for

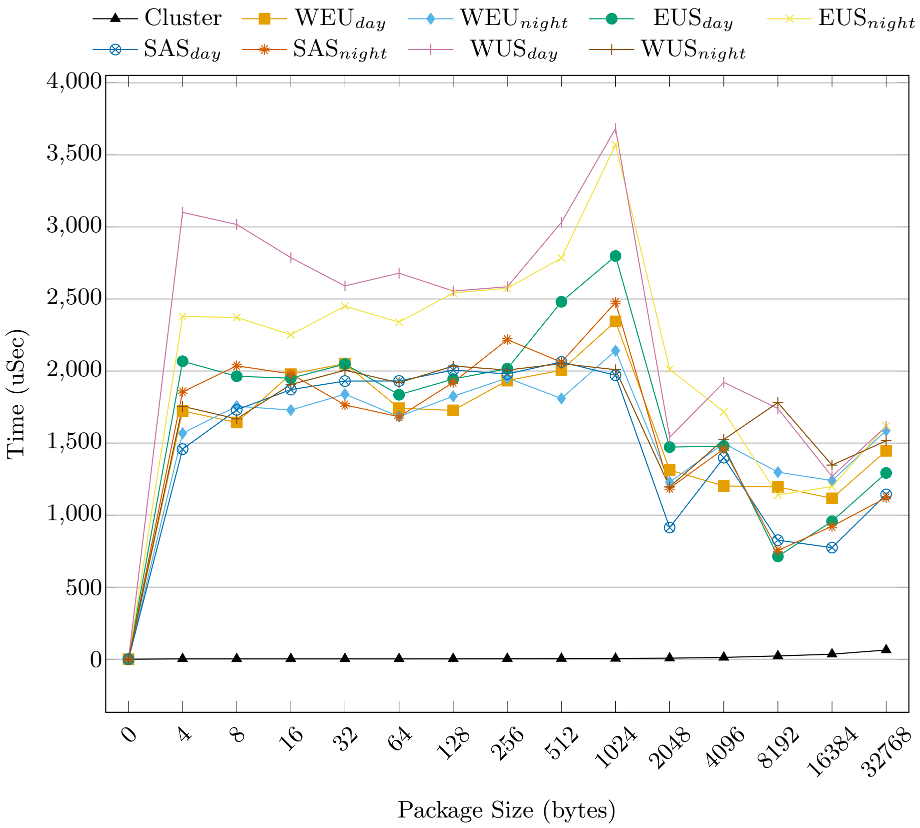

Fig. 5. Results for Reduce benchmark.

this operation close to zero and the cloud allocations present a higher time. The cloud allocations have the same behavior with decreasing the time when the package size reaches 2 KB. The WUS during the day presented slightly lower performance then the other instances, but this did not impede the usage of this Data Center. According to the results we obtained, it is difficult to recommend using the MPI_Reduce function often in applications in the cloud, because the execution time of the application will be affected.

Figure 6 shows the results of the AlltoAll test, the vertical axis shows the time to execute the operation. In this test, all the processes send a message to all other processes and receive a message from all the other processes. The test was performed varying the package size. The time needed for the cluster to perform this operation is very short. The cloud allocations are very predictable and showed a good performance as well. Using a package size from 4 Bytes up to 8 KB, the time for all cloud allocations is around 200 uSec, that is acceptable. The package size has a key role in this operation.

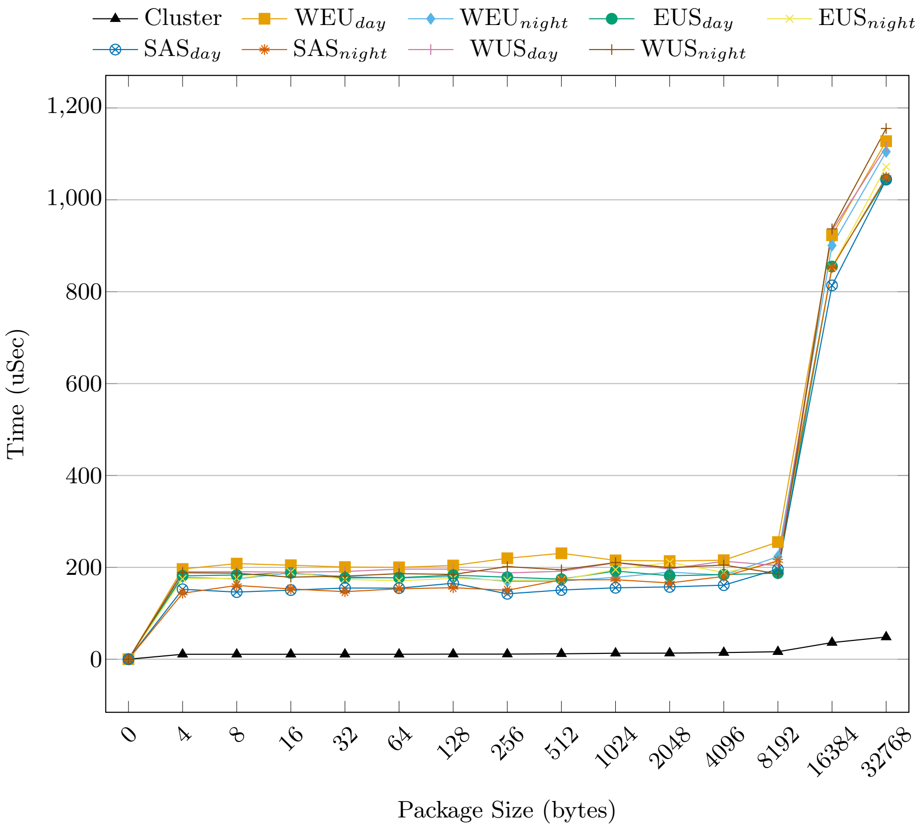

Fig. 6. Results for All to All benchmark.

For package sizes up to 8 KB, both the cluster and the cloud allocations presented the same behavior, with a constant time. When the package was increased from 8 KB to 16 KB, the time in the cluster was increased 3 times and the cloud allocations increased the time by 4 times. The reason for this could be the TCP frame used in the network or some aspect of the MPI implementation. Despite the reason, it is clear that this operation presents good performance until a certain package size. If the application uses several MPI_AlltoAll operations, it is necessary that the user measures the performance of this operation in his network to optimize the application performance by adjusting the package size.

4.2 Applications

We used both NAS benchmarks with the sizes B and C, that represents medium input sizes, and Alya application to measure the performance in the Cloud allocations against the physical cluster execution.

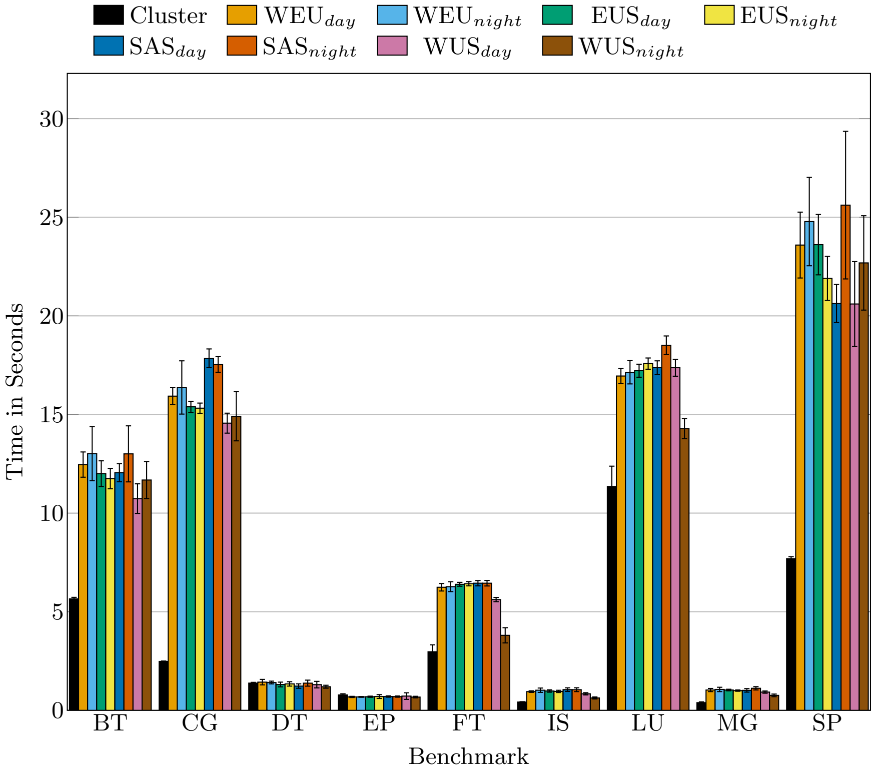

Fig. 7. Performance Results for NAS Class B.

NAS

Figure 7 shows the performance results of NAS-MPI benchmark class B, for multiple nodes on a cluster and four Microsoft Azure data centers. Cluster was faster than Azure's data centers in most cases. The cluster was faster for all the benchmarks, except for DT and EP. These two benchmarks have little communication and they are CPU-bound, as the CPU of the cloud instances is faster then in the cloud.

Comparing the execution during day and night, we did not observe much variability between these experiments. We can conclude that there is no difference between executing HPC applications during day or night in the Azure Cloud. Additionally, we did not observe a huge variation among the four different data centers. We could conclude again that a user could use any data center that he wants, or needs, without significant performance loss.

Figure 8 shows the results for NAS class C. The behavior was practically the same as in the class B results, without big changes.

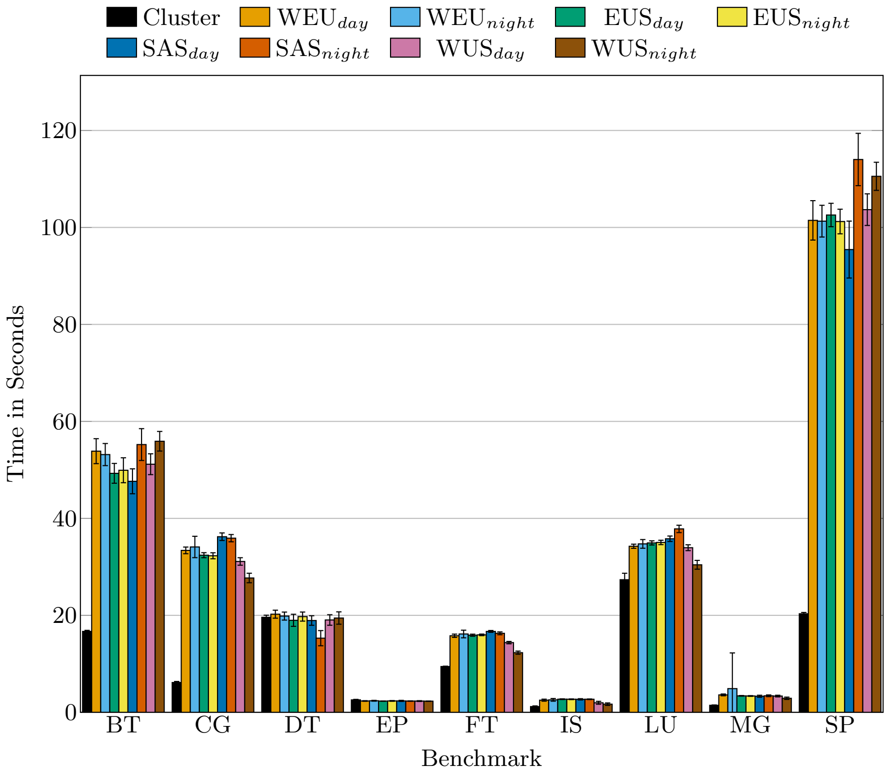

Fig. 8. Performance Results for NAS Class C.

The conclusion that we have is that the G5 instances of the Azure Cloud present an excepted performance degradation according to the size of the problem. Also, we could conclude that the main bottleneck of these instances, and possibly in the whole provider, is the network interconnection. This is supported by the network results and the knowledge of the NAS applications. The applications with little communication, DT and EP, presented better performance in the cloud and all the other presented a performance loss in the cloud, because they all depend on the network performance in different levels.

Alya

Figure 9 illustrates the results of the Alya application among four Azure Data Centers. Due to the NAS results, we decided to not execute Alya during night and day, because the differences between them are low. The results present low variability among the four Azure locations, showing that a real HPC application with a heterogeneous behavior does not depend of the Data Center configuration.

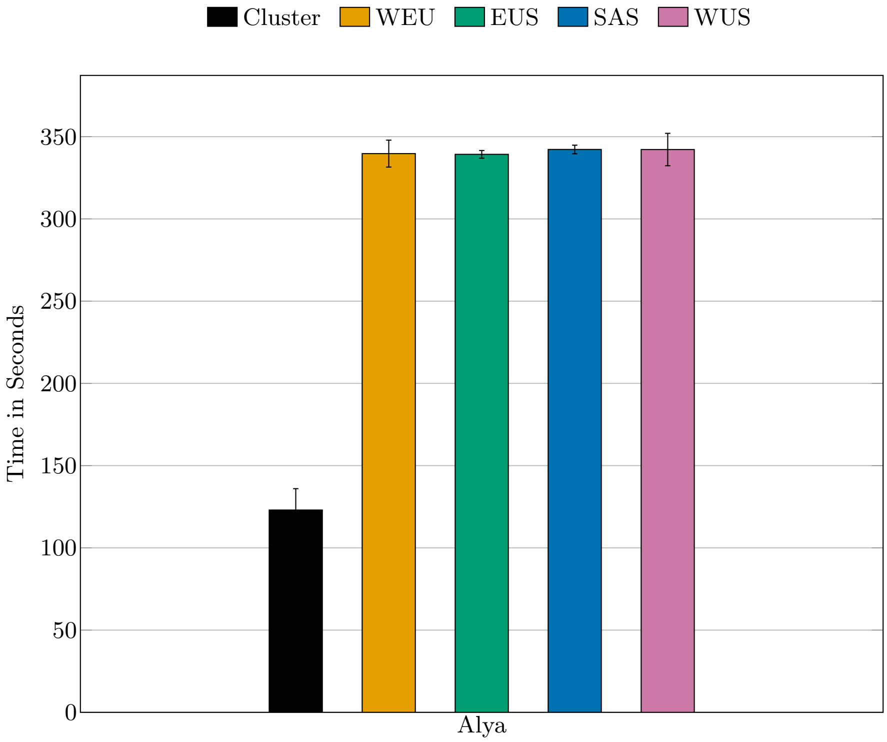

Fig. 9. Performance Results for Alya

Compared with the cluster results, the clouds presented 2 times performance loss. This was expected and it is similar to the NAS performance results.

5 Related Work

Marathe et al. and Awad et al. [6, 1] compare a virtualized cloud cluster against a physical cluster. However, the authors do not provide a comprehensive evaluation of public clouds, because they only used a single Data Center and do not provide a evaluation of the behavior of the different locations of the same provider. Since scientists may have the need to execute their applications in their country, due to legal restrictions, an evaluation of multiple locations is necessary.

The work of He et al. and Iosúp et al [3, 5] provide a comparison between three public clouds and compare the results against a physical machine. However, the authors compared aspects of the machines and does not provide a comparison with focus in HPC needs.

Ekanayake and Fox [2] compared several different applications with a focus on communication patterns. They observed that the applications with more communication presented more degradation when executed in the cloud, which echoes our analysis of the network performance. Our work provides a deeper analysis, because we explore the possibilities inside the cloud providers Data Centers, using the same VM among Data Centers and using two different allocations for each one.

6 Conclusions and Future Work

With our results, we could notice that the network is still the main bottleneck in the Cloud. We saw that there is little variability between the executions during day and night in the same Data Center, with slowdowns during the night execution. Among different Data Centers, we did not observe much variation between them. Regarding to the real HPC application, Alya, we observed that the variation is low among the four Data Centers and it performed well on all of them.

As future work, we intend to compare more aspects of the machines, such as disk I/O and memory bandwidth, that are important components of HPC environments.

Acknowledgments. This research received funding from the EU H2020 Programme and from MCTI/RNP-Brazil under the HPC4E project, grant agreement no. 689772. Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). Additional funding was provided by CAPES and Microsoft.

References

1. Awad, O.M.O., Artoli, A.M.A., Ahmed, A.H.A.: Cloud computing versus in-house clusters: a comparative study. In: Computer Applications and Information Systems (WCCAIS), 2014 World Congress on. pp. 1–6 (Jan 2014)
2. Ekanayake, J., Fox, G.: Cloud Computing: First International Conference, CloudComp 2009 Munich, Germany, October 19–21, 2009 Revised Selected Papers, chap. High Performance Parallel Computing with Clouds and Cloud Technologies, pp. 20–38. Springer Berlin Heidelberg, Berlin, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-12636-9_2
3. He, Q., Zhou, S., Kobler, B., Duffy, D., McGlynn, T.: Case study for running hpc applications in public clouds. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. pp. 395–401. HPDC '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1851476.1851535
4. Intel MPI Benchmarks: User Guide and Methodology Description (2014)
5. Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T., Epema, D.: Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Transactions on Parallel and Distributed Systems 22(6), 931–945 (June 2011)

6. Marathe, A., Harris, R., Lowenthal, D.K., de Supinski, B.R., Rountree, B., Schulz, M., Yuan, X.: A comparative study of high-performance computing on the cloud. In: Proceedings of the 22Nd International Symposium on High-performance Parallel and Distributed Computing. pp. 239–250. HPDC '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2462902.2462919
7. d. R. Righi, R., Rodrigues, V.F., da Costa, C.A., Galante, G., de Bona, L.C.E., Ferreto, T.: Autoelastic: Automatic resource elasticity for high performance applications in the cloud. IEEE Transactions on Cloud Computing 4(1), 6–19 (Jan 2016)
8. Vázquez, M., Houzeaux, G., Rubio, F., Simarro, C.: Alya Multiphysics Simulations on Intel's Xeon Phi Accelerators, pp. 248–254. Springer Berlin Heidelberg, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-662-45483-1_18
9. Vázquez, M., Houzeaux, G., Koric, S., Artigues, A., Aguado-Sierra, J., Arís, R., Mira, D., Calmet, H., Cucchietti, F., Owen, H., Taha, A., Burness, E.D., Cela, J.M., Valero, M.: Alya: Multiphysics engineering simulation toward exascale. Journal of Computational Science 14, 15 – 27 (2016), http://www.sciencedirect.com/science/article/pii/S1877750315300521, the Route to Exascale: Novel Mathematical Methods, Scalable Algorithms and Computational Science Skills
10. Zounmevo, J.A., Kimpe, D., Ross, R., Afsahi, A.: Using mpi in high-performance computing services. In: Proceedings of the 20th European MPI Users' Group Meeting. pp. 43–48. EuroMPI '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2488551.2488556