Hopes and Facts in Evaluating the Performance of HPC-I/O on a Cloud Environment

Pilar Gomez-Sanchez
Computer Architecture and Operating Systems Department (CAOS)
Universitat Autònoma de Barcelona, Bellaterra (Barcelona), Spain.
Sandra Méndez
High Performance Systems Division
Leibniz Supercomputing Centre (LRZ), Garching (Munich), Germany.
Dolores Rexachs and Emilio Luque
Computer Architecture and Operating Systems Department (CAOS)
Universitat Autònoma de Barcelona, Bellaterra (Barcelona), Spain.

***Abstract*—Currently, there is an increasing interest about the cloud platform by the High Performance Computing (HPC) community, and the Parallel I/O for High Performance Systems is not an exception. In cloud platforms, the user takes into account not only the execution time but also the cost, because the cost can be one of the most important issue. In this paper, we propose a methodology to quickly evaluate the performance and cost of Virtual Clusters for parallel scientific application that uses parallel I/O. From the parallel application I/O model automatically extracted with our tool PAS2P-IO, we obtain the I/O requirements and then the user can select the Virtual Cluster that meets the application requirements. The application I/O model does not depend on the underlying I/O system. One of the main benefits of applying our methodology is that it is not necessary to execute the application to select the Virtual Cluster on cloud. Finally, costs and performance-cost ratio for the Virtual Clusters are provided to facilitate the decision making on the selection of resources on a cloud platform.**

***Keywords*-application I/O model, I/O system, Cloud Cluster, I/O phases, I/O access pattern, I/O configuration.**

I. Introduction

Nowadays, the interest about the cloud computing platform is increasing. The scientific community have interest about the cloud computing because some benefits of clouds are that users can acquire and release resources on-demand and they can configure and customize their own Virtual Cluster (VC) [1]. Parallel scientific applications that use parallel I/O can benefit of these platforms, because the user can create and configure the I/O system considering the application requirements which represents an advantage over the traditional HPC-IO systems.

However, in cloud environment, the number of parameters increases considerably. The instance selection for the cluster nodes is not trivial because an instance type comprises of a combination of CPU, memory, storage, and networking capacity. Furthermore, we can observe that the user needs to select components related to the cluster configuration such as the number of instances, the storage global capacity, file system type, etc.

In this paper, we propose a methodology to guide the user in the evaluation of Virtual Clusters (VCs) that will be configured, taking into account the application I/O requirements, and besides, it reduces the cost to do it. The methodology has different stages explained in section II.

There are several studies that evaluate the I/O system performance on cloud platform. Expósito et al. in [2] present an evaluation of the performance of I/O storage subsystem on Amazon EC2, using the local and distributed file system NFS. Juve et al. [3] evaluate the performance and the cost of three real scientific workflows on Amazon EC2, using different storage systems. They use the distributed file systems NFS and Gluster, and the parallel file system PVFS2. Liu et al. [1] evaluate different I/O configuration, using the distributed file system NFS and the parallel file system PVFS2. Furthermore, Liu et al. [4] present a tool, ACIC, to optimize the I/O system for HPC applications in cloud. ACIC allows them to recommend an optimized I/O system configuration according to the user's selected objective. In our work the VC configurations are evaluated taking into account the application I/O model. As our application I/O model does not depend on the underlying I/O system, it can be used on different VCs and cloud platforms.

We have applied our methodology for the NAS BT-IO [5] and S3D-IO [6] benchmarks in four VCs. Furthermore, we select the NFS and PVFS2 as global file systems. In this research, the cloud platform selected is Amazon EC2. The tool selected to create a VC on Amazon EC2 is StarCluster [7].

The rest of this article is organized as follows. Section II introduces our methodology. In Section III, we present the experimental results, in Section IV we review the experimental validation of this proposal. Finally, in the last section, we present the conclusions and future work.

II. Proposed Methodology

We propose a methodology for the performance and cost evaluation of the VC on the cloud environment. This is focused on the I/O requirements of the parallel scientific applications.

Below we explain each step with more details.

1) Application Parallel I/O Characterization: The I/O characteristics are represented by an I/O model. We trace the parallel application with PAS2P-IO [8] and the traces are analyzed to obtain the I/O model. This model is based on the I/O phases of the application. An I/O phase is a repetitive sequence of the same pattern for each file and for a set of processes of the parallel application. Phases represent the order of occurrence of the I/O events and the access order on the different files of the application. A detailed description of the process to extract the I/O model is presented in [9].

In this step, we obtain the following information for the phases for each file of the parallel application:

- $phases$, number of phases on each files of parallel application.
- $IdPh$ is the identifier of phase.
- np, number of processes that composes the phase.
- The I/O pattern of each phase is compose by the request size (rs), the type of operation (w/r) and the number of I/O operations ($\#iop$).

- Number of repetitions rep of the phase $IdPh$.
- Weight $weight(IdPh)$ for the phase $IdPh$. It represents the data transferred during a phase, it is expressed in Bytes and it is calculated by expression 1.

$$weight(IdPh) = np * rs * rep \tag{1}$$

- File Size: For the I/O files and output files, the size is calculated by expression 2 and for input files by expression 3, where phs_write is the number of phases with only write operations and phs_read is the number of phases with only read operations.

$$IOFSize = \sum_{p=1}^{phs\_write} weight_p \tag{2}$$

$$INFSize = \sum_{p=1}^{phs\_read} weight_p \tag{3}$$

- Access Mode (Strided, Sequential and Random).
- Access Type (Unique (a file per process) or Shared (a shared File between the processes)).

Beside, we obtain the number of files (nf) and the storage capacity required for the parallel application. The capacity is calculated by expression 4.

$$StorageCapacity_{app} = \sum_{i=1}^{nf} IOFSize_i + INFSize_i \tag{4}$$

The information obtained in this step represent the I/O requirements for the parallel application because we obtain the I/O characteristics from the I/O model.

2) Creation and Configuration of the Virtual Clusters: A VC is represented by the components shown in Table I.

Table I
COMPONENTS OF VIRTUAL CLUSTER

Parameters	Description
Instance type (*)	Number of cores, processor capacity, RAM memory size.
Number of instances(*)	
Number of I/O nodes (-)	Data servers and metadata server.
Storage type(+)	Temporal and/or persistent.
Device type temporal(+)	HDD or SSD.
Device type persistent(+)	HDD or SSD.
Capacity of temporal storage(+)	As minimum the storage capacity required (expression 4).
Capacity of persistent storage(-)	
Network performance (+)	Low, Moderate, High, Unknown.
I/O library (-)	MPI, NetCDF, pnetcdf, HDF5.
Local file system (+)	File system Linux ext3, ext4, xfs, etc.
Global file system (-)	Parallel, Distributed or Network File systems.
Stripe size (-)	Related by the parallel file system.

(*) the parameters which can be selected by the user, (-) the parameters that the user must configure manually, (+) the parameters that the user cannot change because they are by default depending on instance type.

We can create a VC quickly with StarCluster [7]. We apply the following considerations as a starting point on the selection of the components for a VC that meets the user requirements.

- Storage Capacity: it must guarantee as minimum the storage capacity required (exp. 4) by the application.
- File system Type: it depends on whether the access type is Shared or Unique. If the access type is Shared then the file system must be a global file system such as NFS, PVFS2 or Lustre. When the access type is Unique, it is possible to use the local file system to take advantage of the local disks attached to compute modes.
- Instances: the number of instances is a parameter determined by the number of processes required for the application I/O model and the compute. Instance type depends on the cost that the user can afford.
- Storage Type: it can be temporal (cost free) and/or persistent (cost by GB/month). Usually, the parallel scientific applications use temporal storage during the execution and only the data for postprocessing are saved on persistent storage.
- Network Performance: it can be either high, low, moderate or unknown. It is associated with the instance type. The workload on the network will depend on the number of processes, the request sizes and number of I/O operations.
- I/O library: it depends on the I/O library used by the application. In our case, we only use MPI-IO and POSIX-IO because the application I/O model is obtained at MPI library level.

The baseline software on a VC for each compute node depends on the Machine Image selected. Similar to physical HPC systems, in the HPC on cloud the Linux operating system is used more frequently, especially for the I/O software stack. The software for the parallel processing, such as MPI and global file system, must be installed and configured by the user. The cost will be the main restriction on the creation of a VC. The system components allow to take a decision considering the I/O requirements for the application.

3) Characterization of the Virtual Clusters: We use the IOzone [10] benchmark to obtain the average values for the transfer rate at local file system level. IOzone is a file system benchmark tool that generates and measures a variety of file operations. The benchmark obtains the average transfer rate for request sizes between the minimum and maximum. In this step, an instance can be discarded if, based on the user requirements it provides a low transfer rate.

Furthermore, IOzone can calculate the peak values for the global file system of the VCs. The peak values in this case is the sum of the values obtained on each I/O node of the global file system.

4) Performance Evaluation on the Virtual Clusters for the application I/O model: IOR [11] benchmark evaluates the performance at global file system level. IOR is designed to measure parallel file system I/O performance at both the POSIX and MPI-IO level. The IOR performs writes and reads to/from files under several sets of conditions and reports the resulting throughput rates.

We analyze the access patterns of the I/O model at phases level and proposed an IOR configuration based on the application I/O model, where the relevant parameters are the numbers of processes (np), the number of segments $(-s)$, block size $(-b)$ and transfer size $(-t)$. Table II shows input parameters for IOR based on the I/O model phase. The output of this process is the transfer rate expressed in MB/s, named BW_{CH}, and I/O time for application I/O model. The I/O model has been extracted executing the application once in the cluster. Then, the user can select the VC on cloud using the IOR customized with the I/O model, without reexecuting the application.

5) Cost Evaluation of the Virtual Clusters: Performance obtained using IOR for the application I/O model is used to calculate the cost. The total cost for a specific VC is composed of a variable cost ($cost\_var$) and a fixed cost ($cost\_fix$). This is computed by the expression 5.

$$cost\_tot = cost\_var + cost\_fix \tag{5}$$

The metric selected for IOR is the transfer rate, expressed in MB/s and named BW_{CH}. The variable cost estimated for the application I/O model is calculated by expression 6. The billing is per utilized hours.

$$cost\_var = \sum_{i=1}^{phases} cost(phase[i]) * num\_inst \tag{6}$$

Where num_inst is the number of instances used for the execution and the $cost(phase[i])$ is calculated by expression 7 , and $cost_{inst}$ is the cost per hour for the instance type $inst$.

$$cost(phase[i]) = \frac{weight_{(phase[i])}}{BW_{(CH)}(phase[i])}/3600 * cost_{inst} \tag{7}$$

6) Comparison of the Performance-Cost Ratio for the Virtual Clusters: The performance and the cost for the VCs are presented to the user to simplify the decision making. To compare the performance-cost of the different VCs, we use the results of equation 8.

$$perf\_cost_{ci} = \frac{performance_{ci}}{cost_{ci}} \tag{8}$$

Where ci represent a VC, and $i \in \{1..TotalVirtualClusters\}$.

To compare the Cluster ck and Cluster cj, ck has a higher performance-cost ratio than cj, if $perf\_cost_{ck}$ is greater than $perf\_cost_{cj}$.

III. Experimental Results

In this section, we present the performance evaluation and the cost analysis for two scientific application such as I/O kernels NAS BTIO and S3D-IO that present different I/O access patterns.
BT-IO and S3DIO have been traced using PAS2P-IO to extract their I/O models. This process was done in physical computer clusters Finisterrae of the Centre of Supercomputing of Galicia (CESGA) [12] and Supernova of the Wroclaw Centre for Networking and Supercomputing [13].

We have selected three instance types from Amazon EC2 [14], taking into account the I/O requirements of application (I/O model + resources requirements) and the price of instance. Table III shows the characteristics of the Amazon Instances considered in our experiments. Using the instances of Table III, we have created four Virtual Clusters (VCs); Table IV shows the components of the created VCs for our experiments.

A. Results for the NAS BT-IO

In this section we present the evaluation methodology for the NAS BT-IO Class B and C, using 4, 9, 16, 25 and 36 processes.

1) Application Parallel I/O Characterization: The I/O phases identification is applied to Block Tridiagonal(BT) application of NAS Parallel Benchmark suite (NPB) [5]. We have obtained the following meta-data of NAS BT-IO in the FULL subtype with our tool PAS2P-IO: Explicit offset, Blocking I/O operations, Collective operations, Strided access mode, Shared access type and a shared File accessed by all the MPI processes.

Figure 1 shows the I/O model for the BT-IO for the CLASS B using 16 processes. This behavior is observed for the rest of the classes for the different number of processes. Table V presents the I/O phases for the I/O model using 4, 9, 16, 25 and 36 processes for the classes B and C. Also, we present the storage capacity required by BT-IO for the different classes.

2) Creation and Configuration of the Virtual Clusters: Due to the BT-IO uses a Shared file, we have configured a global file system for the different experiments. For this, we select the network file system NFS and the parallel file system PVFS2. We have selected for BT-IO the VC 1 (executing BT-IO for Class B and C until 16 processes), VC 2 (using 17 nodes until 25 processes), VC 3 (using 11 nodes because these are enough to execute BT-IO for 36 processes Class C) and VC 4 (executing BT-IO Class B and C).

3) Characterization for the Virtual Cluster: Once we have created the VCs, we execute IOzone to evaluate the peak values for transfer rate provided by the ephemeral disk. We only evaluate the ephemerals used for the NFS and PVFS2 global file systems . We evaluate write and read operations for request sizes from 4k to 1GB. Table VI presents the peak values for the four VCs.

4) Performance Evaluation on the Virtual Clusters for the application I/O model: Table VII shows the input parameters to configure IOR from the I/O model phases of the BT-IO. From this process, we obtain the transfer rate (BW_{CH}) and execution time for the BT-IO model. These values are used to calculate the variable cost of expression (6) for the different Classes and number of processes on the four VCs. We show the results obtained for the VC 3.

5) Cost Evaluation of the Virtual Clusters: Figure 2 presents the cost evaluation for the four VCs for the IOR configured to BT-IO Class B and C. Comparing the cost between the VCs we observe that VC 2 is not a good choice, if we want to execute the BT-IO Class B or C with NFS, because the cost is higher than other VCs. However, for the Class C on the right picture on Figure 2, when we work with PVFS2, the cost decreases drastically on VC 4 and we can confirm that this configuration is the best respect to the other three.

6) Comparison of the Performance-Cost Ratio for the Virtual Clusters: Figure 3 shows the performance-cost ratio for the four VCs. This is obtained by the expression (8). We can observe for the Class B that the VC 3, left picture in Figure 3, is more efficient than VCs 1 and 2. The instance type used on VC 3 is compute-intensive and this is an advantage for this workload. Furthermore, it is using a SSD device for the NFS in contrast the other VCs that are using HDD. For a fairer comparison, we evaluate the Class C (right picture in Figure 3), it compares the VCs 3 and 4 that use the same instance type and the I/O device type, but different global file system. VC 4, in best cases, is 100 times more efficient than the VC 3. The user can select

Table II
INPUT PARAMETERS FOR IOR BASED ON THE APPLICATION I/O MODEL

Access Mode	Access Type(AT)	Param. for AT	Number of processes	Number of segment	Block size	Transfer size
Strided	UNIQUE	-F	np=np(IdPh)	-s=rep	-b=rs(IdPh)	-t=rs(IdPh)
Strided	SHARED		np=np(IdPh)	-s=rep	-b=rs(IdPh)	-t=rs(IdPh)
Sequential	UNIQUE	-F	np=np(IdPh)	-s=1	-b=weight(IdPh)	-t=rs(IdPh)
Sequential	SHARED		np=np(IdPh)	-s=1	-b=weight(IdPh)	-t=rs(IdPh)

Table III
CHARACTERISTICS OF THE AMAZON'S INSTANCES SELECTED

Instances	Processor	CPU	RAM (GB)	Storage(GB)	AWS Ireland ($ Per Hour)	AWS Virginia ($ Per Hour)
m1.small	Intel Xeon Family	1	1.7	1x160	0.047	0.044
m1.large	Intel Xeon Family	2	7.5	2x420 HHD	0.190	0.175
c3.xlarge	Intel Xeon E5-2680 v2 2.8 GHz	4	7.5	2x40 SSD	0.239	0.210

Table IV
DESCRIPTIVE CHARACTERISTICS OF THE VIRTUAL CLUSTERS CONFIGURED FOR THE EXPERIMENTS (STEP2)

I/O components	Virtual Cluster 1	Virtual Cluster 2	Virtual Cluster 3	Virtual Cluster 4
Instance Type	m1.small	m1.large	c3.xlarge	c3.xlarge
Number of Instances	17	17	11	11
Storage Type Temporal	Ephemeral	Ephemeral	Ephemeral	Ephemeral
Storage Type Persistent	EBS	EBS	EBS	EBS
Device Type Temporal	HDD	HDD	SSD	SSD
Device Type Persistent	HDD	HDD	HDD	HDD
Capacity of Temporal Storage	160GB	420GB	40GB	300GB
Capacity of Persistent Storage	8GB	8GB	8GB	16GB
Networking Performance	Low	Moderate	High	High
Number of data servers	1	1	1	8
Number of Metadata Server	1	1	1	1
File system Local	ext3	ext3	ext3	ext3
File system Global	NFS	NFS	NFS	PVFS2
Stripe Size	—	—	—	64KB
I/O library	mpich2, pnetcdf	mpich2, pnetcdf	mpich2, pnetcdf	mpich2, pnetcdf
EBS Fixed Cost EU($ per GB-month)	0.55	0.55	0.55	0.55
EBS Fixed Cost US-East($ per GB-month)	0.50	0.50	0.50	0.50

Table V
THE I/O PHASES FOR THE BT-IO MODEL USING 4, 9, 16, 25 AND 36 PROCESSES FOR THE CLASSES B AND C(STEP1)

IdPh	np	Operation	rs	rep	weight(IdPh) (rep*np*rs)	weight(app)	Storage Capacity required
Class B						3.2GB	
1 to 40	4	Write_at_all	10MB	1	$1 * 4 * 10$MB		1.6GB
41	4	Read_at_all	10MB	40	$40 * 4 * 10$MB		
1 to 40	9	Write_at_all	4.5MB	1	$1 * 9 * 4.5$MB		1.6GB
41	9	Read_at_all	4.5MB	40	$40 * 9 * 4.5$MB		
1 to 40	16	Write_at_all	2.5MB	1	$1 * 16 * 2.5$MB		1.6GB
41	16	Read_at_all	2.5MB	40	$40 * 16 * 2.5$MB		
1 to 40	25	Write_at_all	1.62MB	1	$1 * 25 * 1.62$MB		1.6GB
41	25	Read_at_all	1.62MB	40	$40 * 25 * 1.62$MB		
1 to 40	36	Write_at_all	1.12MB	1	$1 * 36 * 1.12$MB		1.6GB
41	36	Read_at_all	1.12MB	40	$40 * 36 * 1.12$MB		
Class C						12.9GB	
1 to 40	4	Write_at_all	40.55MB	1	$1 * 4 * 40.55$MB		6.4GB
41	4	Read_at_all	40.55MB	40	$40 * 4 * 40.55$MB		
1 to 40	9	Write_at_all	18MB	1	$1 * 9 * 18$MB		6.4GB
41	9	Read_at_all	18MB	40	$40 * 9 * 18$MB		
1 to 40	16	Write_at_all	10.14MB	1	$1 * 16 * 10.14$MB		6.4GB
41	16	Read_at_all	10.14MB	40	$40 * 16 * 10.14$MB		
1 to 40	25	Write_at_all	6.49MB	1	$1 * 25 * 6.49$MB		6.4GB
41	25	Read_at_all	6.49MB	40	$40 * 25 * 6.49$MB		
1 to 40	36	Write_at_all	4.50MB	1	$1 * 36 * 4.50$MB		6.4GB
41	36	Read_at_all	4.50MB	40	$40 * 36 * 4.50$MB		

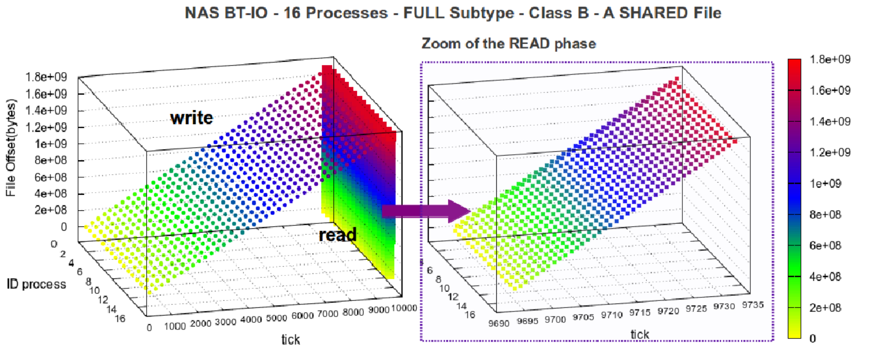

Figure 1. The left picture shows the I/O model for the application and the right picture shows a zoom on the read operations. It can be observed that write and read are done in the same file offset. The application uses a Shared file. Each MPI process performs a write operation every 122 communication events. This is done 40 times, and after, each process performs 40 read operations consecutively.

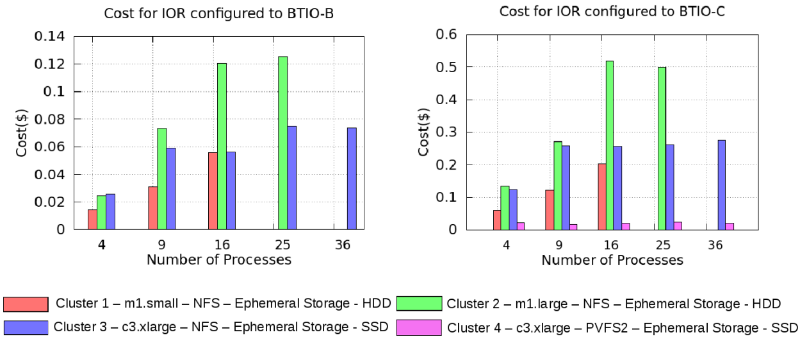

Figure 2. Cost Evaluation of the four Virtual Clusters using IOR configured for the BT-IO. The left picture corresponds to Class B and the right picture to Class C. Virtual Clusters with experiments without results are limited by storage capacity or the parallel degree required. Class B was not tested on Virtual Cluster 4 because the I/O workload is small for its I/O system.

Table VI
PEAK VALUES FOR THE VIRTUAL CLUSTERS OBTAINED FROM IOZONE (STEP3)

Cluster	rs	Write (MB/s)	Read (MB/s)	Time (sec)
1				33.6
	32KB	108	114	
	256KB	106	115	
2				43.6
	32KB	85	84	
	1GB	78	91	
3				24.1
	8MB	116	291	
	1GB	96	308	
4				24.1
	8MB	930	2,328	
	1GB	771	2,462	

Table VII
IOR INPUT PARAMETERS FROM THE I/O MODEL PHASES OF THE NAS BT-IO SUBTYPE FULL - COLLECTIVE OPERATIONS AND $s = rep = 40$. OUTPUTS FOR THE VIRTUAL CLUSTER 3. (STEP4)

np(IdPh)	b=rs(IdPh) (MB)	t=rs(IdPh) (MB)	BW_{CH} (MB/s)	Time (sec)
Class B				
4	10.0	10.0	104	15.5
9	4.5	4.5	91	17.7
16	2.5	2.5	96	16.9
25	1.6	1.6	73	22.6
36	1.1	1.1	74	22.2
Class C				
4	40.6	40.6	87	74.5
9	18.0	18.0	83	77.8
16	10.1	10.1	84	77.2
25	6.5	6.5	82	79.0
36	4.5	4.5	78	82.9

Table VIII
I/O PHASES FOR THE S3D-IO MODEL USING 8, 16 AND 32 PROCESSES FOR THE WORKLOAD 200X3 AND 400X3, OPERATION TYPE WRITE_ALL AND $rep = 1$ (STEP1)

IdPh	np	rs (MB)	weight(IdPh) (MB)	weight (app)	Storage Capacity
200x3			(rep*np*rs)	4.8GB	4.8GB
1 to 5	8	122	976		
1 to 5	16	61	976		
1 to 5	32	30.5	976		
400x3			(rep*np*rs)	39GB	39GB
1 to 5	8	977	7816		
1 to 5	16	488	7808		
1 to 5	32	244	7808		

Table IX
IOR INPUT PARAMETERS FROM THE I/O MODEL PHASES OF THE S3D-IO - COLLECTIVE OPERATIONS AND $s = rep = 1$. OUTPUTS FOR THE VIRTUAL CLUSTER 4. (STEP4)

np(IdPh)	b=rs(IdPh) (MB)	t=rs(IdPh) (MB)	BW_{CH} (MB/s)	Time (sec)
200x3				
8(2x2x2)	122	122	541	9.0
16(2x2x4)	61	61	474	10.4
32(2x4x4)	30.5	30.5	411	11.9
400x3				
8(2x2x2)	977	977	398	98.1
16(2x2x4)	488	488	431	90.6
32(2x4x4)	244	244	425	91.7

the VC 4 where the only extra work is the installation and configuration of PVFS2.

B. Results for the S3DIO

In this section we present the evaluation methodology for the S3DIO [6] for the workload 200x3 and 400x3, using 8, 16 and 32 processes.

1) Application Parallel I/O Characterization: S3D-IO uses parallel NetCDF for checkpointing. A checkpoint is performed at regular intervals, and its data consist of 8-byte three-dimensional arrays. We have obtained the following metadata for the S3D-IO with our tool PAS2P-IO: Collective write, Individual file pointer, Blocking I/O operations, Strided access mode, Shared access type and five Shared files accessed by all MPI processes.

Figure 4 shows the I/O model for the S3D-IO using 8 and 16 processes for the workload 200x200x200 (200x3). This behavior is observed for the rest of the workload for the different number of processes. Table VIII presents the I/O phases for the I/O model using 8, 16 and 32 processes for the workloads 200x3 and 400x3. Besides, we show the storage capacity required by the application.

2) Creation and Configuration of the Virtual Clusters: We have selected VCs 1 and 2, considering the number of cores available, to execute S3D-IO for the workload 200x3, 400x3 up to 16 processes. On VC 3 we run the application for the workload 200x3 up to 32 processes. On VC 4 due to its capacity and number of cores we can run for the workloads 200x3 and 400x3 up to 32 processes.

3) Characterization on the Virtual Cluster: The characterization has been presented in section III-A3.

4) Performance Evaluation on the Virtual Clusters for the application I/O model: Table IX shows the input parameters to configure IOR from the I/O model phases of the S3D-IO and the results obtained for the VC 4. Values obtained for the IOR configured for the S3DIO are used to calculate the costs on the four VCs.

5) Cost Evaluation of the Virtual Clusters: Figure 5 presents the cost evaluation for the four VCs for the IOR configured to S3DIO for the workloads 200x3 and 400x3. For the workload 200x3, left picture on Figure 5, we can observe that VC 4 is cheaper than the other VCs and VC 2 is more expensive than the other VCs for the 16 processes. However, VC 2 has a similar cost to VC 3 for 8 processes. For this workload the instance type c3.xlarge is not an advantage on comparison to m1.large because the decrease

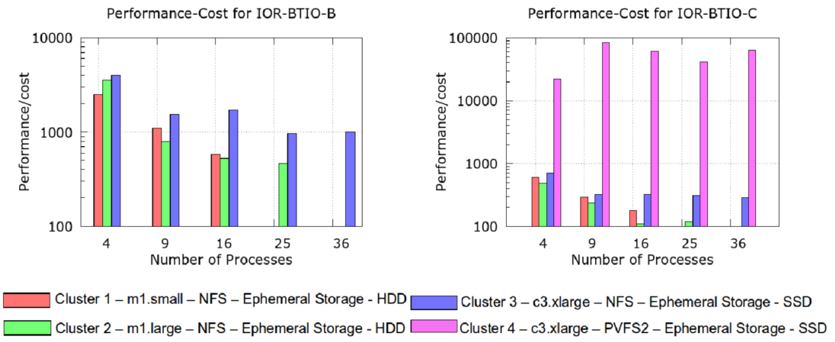

Figure 3. Performance-Cost ratio of the four Virtual Clusters using IOR configured for the BT-IO. The left picture corresponds to Class B and the right picture to Class C. Results are shown in logarithmic scale. Virtual Clusters with experiments without results are limited by storage capacity or the parallel degree required. Class B was not tested on Virtual Cluster 4 because the I/O workload is small for its I/O system.

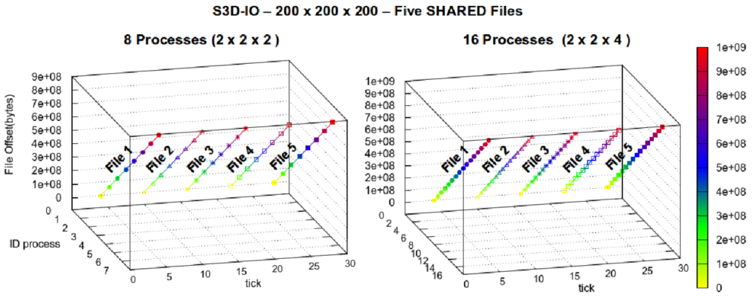

Figure 4. The left picture shows the I/O model for 8 processes with a workload 200x200x200. The application uses five shared Files. All MPI processes write once on the File 1, after all processes write on the File2, and so on. This access pattern is representing the checkpointing process for the S3D application. The same behavior is observed in the right picture for 16 processes.

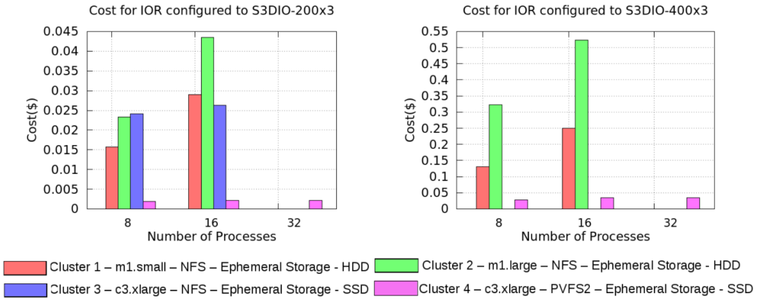

Figure 5. Cost Evaluation of the four Virtual Clusters using IOR configured for the S3DIO. The left picture corresponds to workload 200x3 and the right picture to workload 400x3. Virtual Clusters with experiments without results are limited by storage capacity or the parallel degree required.

in execute time does not compensate the cost associated to type and number of instances. For the workload 400x3, right picture on Figure 5, the VC 4 is cheaper than the rest of the VCs and the VC 2 is the most expensive.

6) Comparison of the Performance-Cost Ratio for the Virtual Clusters: Figure 6 presents the performance-cost ratio for the four VCs for the IOR configured to S3DIO for the workloads 200x3 and 400x3. We can observe for the workload 200x3, left picture in Figure 6, that the VC 4 is 100 times more efficient than the VC 1, 2 and 3 for 8 and 16 processes. For the workload 400x3, right picture in Figure 6, the VC 4 is around 100 times more efficient than the VC 1 and 2. In this case, VC 3 is not used because its storage capacity is not enough for the I/O workload. The VC 1 is more efficient than the VC 2 despite it uses the instance micro m1.small. Furthermore, the performance-cost ratio remains on similar values for the three number of processes evaluated both for the workload 200x3 and 400x3.

IV. Experimental Validation

In this section, we show the benefits of executing customized IOR with the application I/O model instead of executing the application on the cloud. Figure 7 presents a comparison of the time execution for the IOR and BT-IO for classes B and C using 4, 9, 16, 25, and 36 processes. The two left pictures correspond to the Class B and the two right correspond to the Class C. The same happens with S3DIO.

In Figure 7 we can observe that IOR time is smaller than BT-IO time for the two classes and for the different number of processes. Our methodology is suitable for applications with significant I/O, because when the I/O workload per process decreases the advantage to use IOR instead of run the real application, this is not significant, and the I/O leaves to have impact on application execution time. We can observe this situation using 25 and 36 for BT-IO Class B.

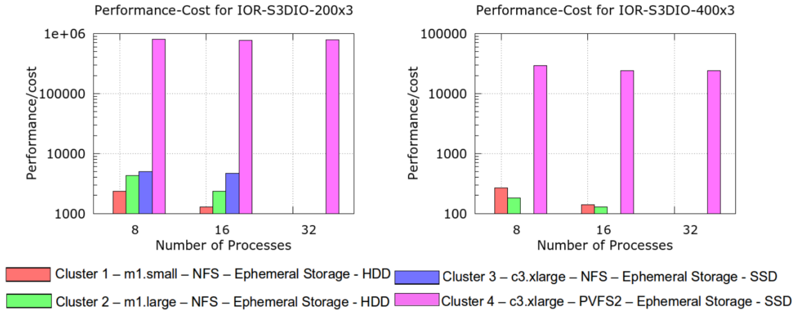

Figure 6. Performance-Cost ratio of the four Virtual Clusters using IOR configured for the S3DIO. The left picture corresponds to workload 200x3 and the right picture to workload 400x3. Results are shown in logarithmic scale. Virtual Clusters with experiments without results are limited by storage capacity or the parallel degree required.

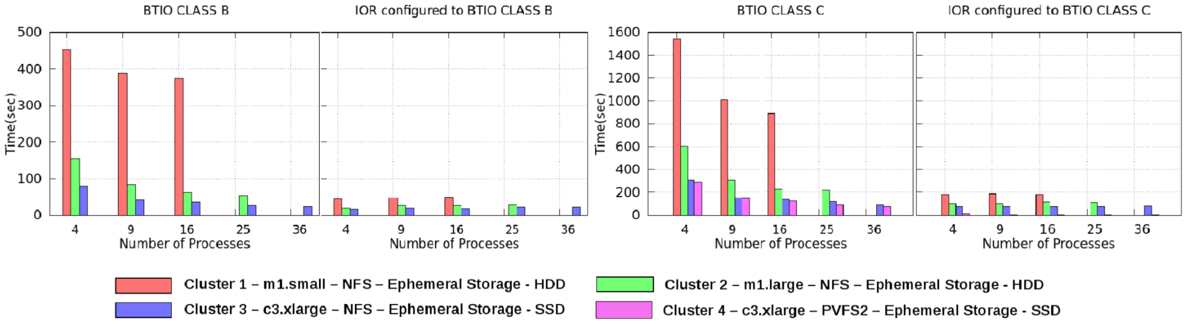

Figure 7. Comparison of the IOR Execution Time vs BT-IO Execution Time.

V. Conclusion

We have applied a methodology to evaluate performance and cost of Virtual Clusters (VCs) on a cloud environment, taking into account the application I/O model. Components for the VC that impact on the cost and performance have been selected for the scientific parallel applications that use parallel I/O. We give a list of the components for the VCs to facilitate the choice. From the parallel application I/O model, that is extracted once, we customize the IOR. So, we can change the parameters quickly and this allows to evaluate similar configurations without executing the application every time. We show that evaluate with IOR allows to reduce the execution time and the cost to evaluate the I/O performance for different VCs.

As future work, we will continue analyzing the impact of the different components for the VC configuration on cost and performance. Also, we will work with different file systems and other cloud platform to evaluate the applicability of our methodology.

Acknowledgment

This research has been supported by the MINECO (MICINN) Spain under contract TIN2011-24384. The research position of the PhD student P. Gomez has been funded by a research collaboration agreement, with the "Fundación Escuelas Universitarias Gimbernat".

Appreciation to The Centre of Supercomputing of Galicia (CESGA) under the Science and Technology Infrastructures program (in spanish ICTS) and The Wroclaw Centre for Networking and Supercomputing (WCSS), Poland.

References

[1] M. Liu, J. Zhai, Y. Zhai, X. Ma, and W. Chen, "One Optimized I/O Configuration Per HPC Application: Leveraging the Configurability of Cloud," in *Proceedings of the Second Asia-Pacific Workshop on Systems*. ACM, 2011, pp. 15:1–15:5.

[2] R. Expósito, G. 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," *Journal of Grid Computing*, vol. 11, no. 4, pp. 613–631.

[3] G. Juve, E. Deelman, G. B. Berriman, B. P. Berman, and P. Maechling, "An Evaluation of the Cost and Performance of Scientific Workflows on Amazon EC2," *J. Grid Comput.*, vol. 10, no. 1, pp. 5–21, Mar. 2012.

[4] M. Liu, Y. Jin, J. Zhai, Y. Zhai, Q. Shi, X. Ma, and W. Chen, "ACIC: Automatic Cloud I/O Configurator for HPC Applications," in *Proceedings of the Int. Conf. on High Performance Computing, Networking, Storage and Analysis*, ser. SC'13. ACM, 2013, pp. 38:1–38:12.

[5] P. Wong and R. F. V. D. Wijngaart, "Nas parallel benchmarks i/o version 2.4," Computer Sciences Corporation, NASA Advanced Supercomputing (NAS) Division, Tech. Rep., 2003.

[6] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C. S. Yoo, "Terascale direct numerical simulations of turbulent combustion using S3D," *Computational Science & Discovery*, vol. 2, no. 1, p. 015001, 2009.

[7] StarCluster. (2014) An Open Source Cluster-Computing Toolkit for Amazon's Elastic Compute Cloud (EC2). [Online]. Available: http://star.mit.edu/cluster/

[8] S. Méndez, J. Panadero, A. Wong, D. Rexachs, and E. Luque, "A New approach for Analyzing I/O in Parallel Scientific Applications," in *CACIC12*, 2012, pp. 337–346.

[9] S. Méndez, D. Rexachs, and E. Luque, "Modeling Parallel Scientific Applications through their Input/Output Phases," in *Cluster Computing Workshops, 2012 IEEE Int. Conf. on*, Sept 2012, pp. 7–15.

[10] W. D. Norcott. (2006) IOzone Filesystem Benchmark. [Online]. Available: http://www.iozone.org/

[11] W. Loewe, T. McLarty, and C. Morrone. (2012) IOR Benchmark. [Online]. Available: https://github.com/chaos/ior/blob/master/doc/USER_GUIDE

[12] CESGA. (2014) Finisterrae of the centre of supercomputing of galicia (CESGA). [Online]. Available: https://www.cesga.es

[13] WCSS. (2014) Supernova of the Wroclaw Centre for Networking and Supercomputing (WCSS). [Online]. Available: https://www.wcss.pl

[14] AWS-EC2. (2014) Amazon Elastic Compute Cloud, Instance Types. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html

Received: June 2014. Accepted: February 2015.