Quantitative Characterization of Scientific Computing Clusters

Aurelio Vivas(✉) and Harold Castro

COMIT Reserach Group, Universidad de los Andes, Bogotá, Colombia
{aa.vivas,hcastro}@uniandes.edu.co

Abstract. Several forms of non-HPC clusters named cluster of workstations and cluster of virtual machines have become available in universities and research institutions as cost effective solutions for scientific computing. With the need to characterize the cluster computing systems that are traditionally used to run high-performance computing applications and those that are not, the terms tightly- and loosely- coupled clusters were adopted. However this qualitative characterization of clusters does not provide further characterization of non-HPC systems, and does not reveal real insights into their capacity to tackle many scientific applications. As a consequence, researchers who use these computing systems do not have the tools to make informed decisions about what type of applications better fits the capacity and capabilities of every kind of non-HPC cluster. In this work, we propose the cluster performance profile. This profile enables the quantitative characterization, initially, on non-HPC clusters in order to support decisions in the use and development of these clusters.

Keywords: Cluster computing · High-performance computing · Cluster overhead · Cluster coupling · Cluster benchmarking

1 Introduction

Although supercomputers have been considered as the de facto computer architectures for the execution of high-performance computing (HPC) applications, their operation and maintenance costs have increased at higher rates [9] when compared to other forms of cluster architectures. These increasing rates have led the scientific community into the development of emerging and low-cost non-HPC clusters such as cluster of workstations (COW), cluster of desktops (COD), and cluster of virtual machines (COV), which are commonly found in research laboratories and institutions.

In non-HPC clusters, researchers have sought to give answers to the following questions. First, *what kind of cluster better fits the needs of a given scientific application (HPC and non-HPC)*. Second, which is a common and recurring question made among scientists when considering the execution of scientific applications on these clusters, *what configuration (nodes, and processors) of these*

P. Navaux et al. (Eds.): CARLA 2022, CCIS 1660, pp. 47–62, 2022.
https://doi.org/10.1007/978-3-031-23821-5_4

clusters render the best performance for a given scientific application. In order to give answers to the above-mentioned research questions, we propose the **cluster performance profile**, this profile summarizes the performance obtained for a variety of scientific applications in strong scaling benchmarking experiments. From these performances, we derive further cluster specific metrics named cluster overhead and cluster coupling, based on a previously proposed methodology [14]. The cluster performance profile that includes the applications performance metrics, the cluster overhead, and cluster coupling enables the characterization of not only non-HPC, but also HPC clusters by quantifying their strengths and weaknesses when executing well-known scientific computing kernels such as the seven dwarfs of scientific computing. The seven dwarfs of scientific computing exhibit a variety of communication and computation patterns common to many scientific applications, they enable further characterization of the capacity of clusters under different kinds of workloads.

Research questions are addressed as follows in this research paper. Section 2 presents related work on characterizing scientific computing clusters and evaluating their capacity for the execution of scientific applications. Section 3 presents a background on cluster overhead, cluster coupling and the proposed cluster performance profile. Section 4 presents the performance evaluation of four clusters and the calculation of their cluster performance profiles. Section 5 discusses our findings on these cluster profiles and provides answers to the proposed research questions. Conclusions on the use of the cluster performance profile and future directions are drawn in Sect. 6.

2 Related Work

The quantification of the capacity of scientific computing clusters for the execution of high-performance computing applications is a common and ongoing research problem, named performance evaluation and benchmarking. These evaluations provide to some extent an estimate of the performance that computers can deliver for specific applications. On non-HPC clusters, related works have been conducted in virtual clusters built on top of containers or virtual machines over cloud [1,6,8,10,12], IoT [3,11], workstations [2], and desktops [4] infrastructures. Bare-metal cluster deployments on top of workstations [7] and desktops [13] have also been considered. Furthermore, those related works usually estimate the capacity in terms of the metrics delivered by the traditional compute-intensive benchmarks such as high-performance Linpack (HPL), NAS Parallel Benchmarks (NPB), or HPC Challenge Benchmark (HCB).

Most related works that estimate the capacity of virtual and containerized clusters concentrate their efforts in the determination of the computation and communication overhead these technologies pose on the performance of HPC applications. These works usually compare the capacity (given in FLOPS, latency, bandwidth, and related metrics) of the bare-metal host system with the same system when hosting containers or virtual machines. On the other hand, related works that estimate the capacity of bare-metal deployments, such as clusters of workstations and desktops use the capacity estimated in supercomputers as base line to determine if the system under study has a satisfactory capacity.

In this work we extend the state of the art in performance evaluation and benchmarking of scientific clusters in the following directions: (i) we extend the performance analysis commonly conducted in strong scaling analyses by including the cluster performance profile that comprises the traditional performance metric, and our previously proposed metrics (cluster overhead and cluster coupling [14]), finally (ii) we provide a quantitative characterization of four small size scientific computing clusters and demonstrate the validity of this characterization, the characterization being summarized in a cluster performance profile per cluster.

3 Background

Here we define the concepts that are relevant for the quantitative characterization of scientific computing clusters; namely, cluster overhead, cluster coupling, and the cluster performance profile.

3.1 Cluster Overhead and Coupling

Cluster distributed systems have been categorized as loosely- or tightly-coupled according to the storage, interconnection, processing technology and the components packing strategy employed in their development. In addition, the speed and reliability of the interconnection channel have been considered as the criteria for this classification [5]. Nevertheless, the loosely- and tightly-coupled classification does not provide quantitative information about how coupled these different computing systems are. Accordingly, in [14] we proposed a methodology to quantitatively estimate the coupling of clusters using a metric we called cluster overhead.

The **cluster overhead** is estimated by determining how similar a given cluster is to its tightly-coupled counterpart, assumed to be a single node. Figure 1 depicts the performance for a single node of a cluster ($P_h(1)$) and the performance for the same cluster with n computing nodes ($P_h(n)$), for the high-performance computing application building block h. This figure also depicts $P'_h(1)$ and $P'_h(n)$ serving as linear approximations of $P_h(1)$ and $P_h(n)$ respectively. Then, the cluster overhead with respect to h is given by the following formula.

$$\alpha = \measuredangle \overline{P'_h(1)}\,\overline{P'_h(n)} \tag{1}$$

Although computers' performance does not have linear behavior, linear approximations led us to derive properties of lines that allow further understanding of cluster's performance. For instance, in the formula, the angle (α) between the segments $\overline{P'_h(1)}$ and $\overline{P'_h(n)}$ stand as the performance loss also known as the cluster overhead, this measured in degrees. Note that there is an inverse relationship between the cluster overhead and coupling, as shown in Fig. 2. In Fig. 2, large values for α stand for large overhead, resulting in a poor similarity between a node and the whole cluster, this being rendered in a loose cluster coupleness.

On the other hand, small values for α stand for small overhead, resulting in a higher similarity between a node and the whole cluster, this being rendered in a tightly cluster coupleness. Then, the **cluster coupling** is defined by the following formula, where α stands for the cluster overhead.

$$c = \frac{1}{\alpha} \tag{2}$$

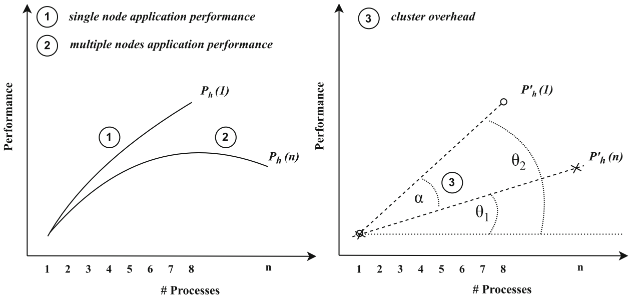
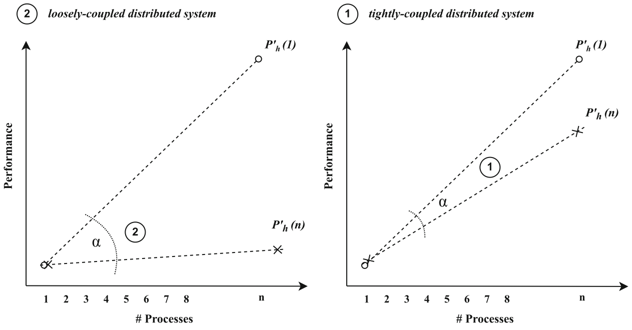


Fig. 1. Cluster overhead



2 loosely-coupled distributed system
1 tightly-coupled distributed system
Performance
$P'_h(1)$
$P'_h(n)$
α
1 2 3 4 5 6 7 8 n
# Processes

Fig. 2. Cluster coupleness

Note that the performance measure used for the calculation of coupling and overhead metrics variates according to the specific application being used for the measurement; for instance, NAS Parallel benchmarks use the MOP/s (millions of operations per second) as the standard to deliver benchmark performance, the type of operation OP variates according to the specific benchmark. Benchmarks such as FT, MG and CG use as an operation unit the Float Point whilst EP use Random Numbers Generated.

3.2 Cluster Performance Profile

Table 1 presents the cluster performance profile for an hypothetical cluster. This profile summarizes the properties of segments such as $\overline{P'_h(1)}$ and $\overline{P'_h(n)}$ that are of interest for clusters characterization. These properties are: the slope (m), the angle between a segment and the x-axis (θ), the cluster overhead (α) that is calculated with respect to another segment, and the coupling (c) which is derived from the cluster overhead. In the cluster performance profile, m and θ bring information about how fast the performance increases when considering a segment established between points $(p, P'_h(p))$ and $(p+1, P'_h(p+1))$, where p is the number of processors considered and $P'_h(p)$ is the performance of h when considering p processors of the cluster. If we think in terms of a line segment, m is the segment's slope and θ the angle formed between the x-axis and the segment, both m and θ are directly proportional. In a similar way, α describes how close the performance of a cluster being represented by a segment is to a tightly-coupled instance being represented by another segment.

The baseline for the cluster performance profile is a single node. Since this is supposed to be a tightly-coupled instance, when compared to itself, the cluster overhead is defined to equal zero, $\alpha = 0$. As the number of computational resources (processors, nodes) used in computing h increases, performance will tend to drop due to the parallel overhead. This also may affect the performance growth rate being estimated by m and θ, increases the cluster overhead α, and decreases the system coupling. However, although the above is the expected behavior, different computation and communication patterns might differ from this behavior for different settings of nodes and processors. These differences are intended to be caught in the cluster performance profile.

Table 1. Cluster performance profile

Application	Nodes	$(p, P_h(p))$		m	θ	α	c
h	1	$(1, P_h(1))$	$(4, P_h(4))$	m_1	θ_1	0	c_1
	2	$(1, P_h(1))$	$(8, P_h(8))$	m_2	θ_2	> 0	$< c_2$
	4	$(1, P'_h(1))$	$(16, P_h(16))$	m_3	θ_3	$>> 0$	$<< c_3$

4 Performance Evaluation

Five scientific computing dwarves (spectral methods, sparse linear algebra, unstructured meshes, structured meshes and monte carlo) represented in four NAS parallel benchmarks (FT, CG, MG and EP) are used to evaluate the performance of three non-HPC clusters C_{w_1}, C_{w_2}, C_{cov} and one HPC cluster C_{hpc} which is used for validation purpose. The resulting performance delivered by the applications in a strong scaling evaluation is then used to the elaboration of a cluster performance profile per cluster, according to the methodology proposed in Sect. 3.2. The findings on these profiles are further discussed in Sect. 5.

4.1 Experimental Setup

The experiment comprises a strong scaling performance evaluation on dedicated clusters C_{w_1}, C_{w_2}, C_{cov} and C_{hpc} whose technical details are shown in Table 2. C_{hpc} is the university high-performance computing system; C_{cov} is a Microsoft Hyper-V-based cluster of virtual machines whose cluster computing nodes were deployed on different servers at the university datacenter, although virtual machines are dedicated, the datacenter is not; C_{w_1} and C_{w_2} are clusters of workstations differing in their network bandwidth and latency capabilities. The FT, CG, MG, and EP benchmarks are executed over the only-MPI execution scheme. Here, we increased the number of MPI processes and the number of nodes. The Class C of NAS parallel benchmark problem sizes is considered and remains fixed through experiments. The MPI processes mappings considered for the experiment are described in Table 3. In Table 3, *ppn* stands for MPI processes per node and *tp* for the total number of processes in the cluster.

Table 2. Clusters specifications

Specs	C_{hpc}	C_{cov}	C_{w_1}	C_{w_2}
# of nodes	4	4	4	4
CPU(s)	28	4	4	4
CPU model	Intel(R) Xeon(R) CPU E5-2690 v4	Intel(R) Xeon(R) CPU E5-2680 v4	Intel(R) Core(TM) i7-4790	Intel(R) Core(TM) i7-4790
CPU clock speed	2.60 GHz	2.40 GHz	3.60 GHz	3.60 GHz
Thread(s) per core	2	1	2	2
Core(s) per socket	14	1	4	4
Socket(s)	2	4	1	1
NUMA node(s)	2	1	1	1
Network bandwidth	40 Gb/s 4x QDR	10 Gb/s	100 Mb/s	1 Gb/s
Network latency	1.3 us	~77 us	~380 us	~81 us

Table 3. Processes mappings

# of nodes	1	1	1	1	1	1	2	2	2	2	2	4	4	4	4	4
ppn	1	2	4	8	16	32	1	2	4	8	16	1	2	4	8	16
tp	1	2	4	8	16	32	2	4	8	16	32	4	8	16	32	64

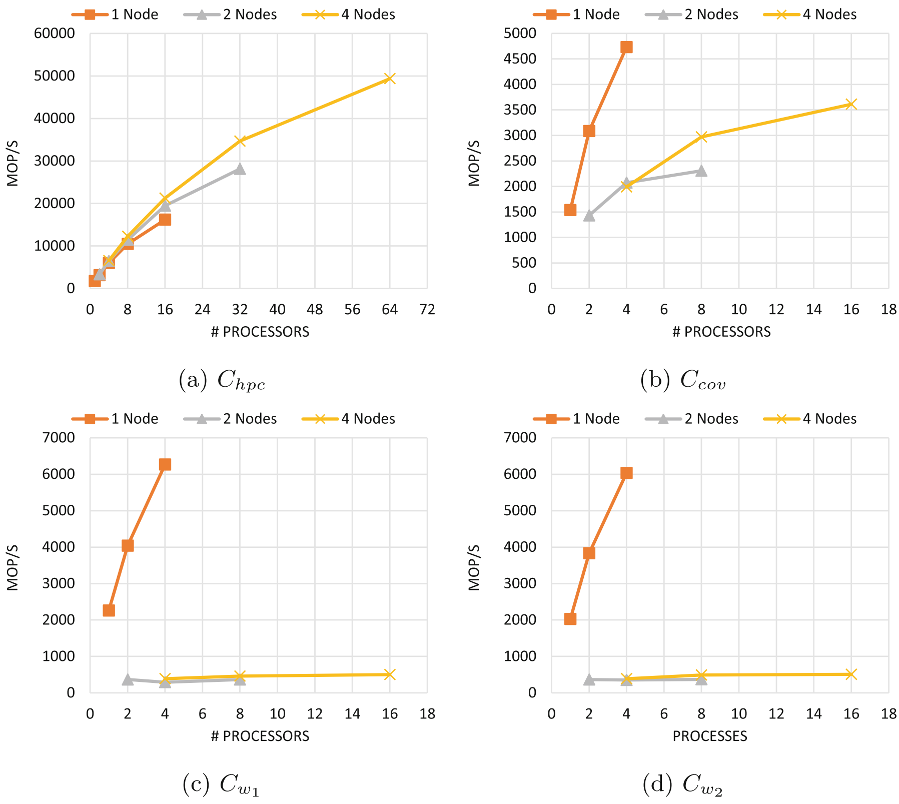
4.2 Threats to Validity

The performance exhibited by a cluster is susceptible to a countless number of software parameters. To name a few: the problem size, the parallelization scheme, the supporting numerical libraries and the algorithms. Given the actual difficulty in providing an accurate measure of overall computers performance, our estimation considers a simplified version of the problem. Here, we consider small-scale clusters, and well-known scientific computing kernels with fixed problem sizes and execution schemes; these in order to validate our methodology. Regarding the execution scheme, in the HPC cluster we do not use all the physical cores in order to be able to use only powers of 2 number of processors, then this must be considered when performing comparisons between clusters. Also although experiments were executed three times, we observe no significant variations that lead to errors in our estimations of performance. In this regard, results are well supported by theory.

Finally, note the performance is a compound metric that measures the application and computing system performance, not the computing system in isolation, the former is what we mean when referring to the performance of the cluster.

4.3 Results

Figures 3, 4, 5 and 6 describe the performance achieved for clusters C_{hpc}, C_{cov}, C_{w_1} and C_{w_2} in the FT, CG, MG and EP benchmarks, respectively. These figures compare the performance achieved in the clusters for the different settings of nodes and processors considered for the computations.



(a) C_{hpc}

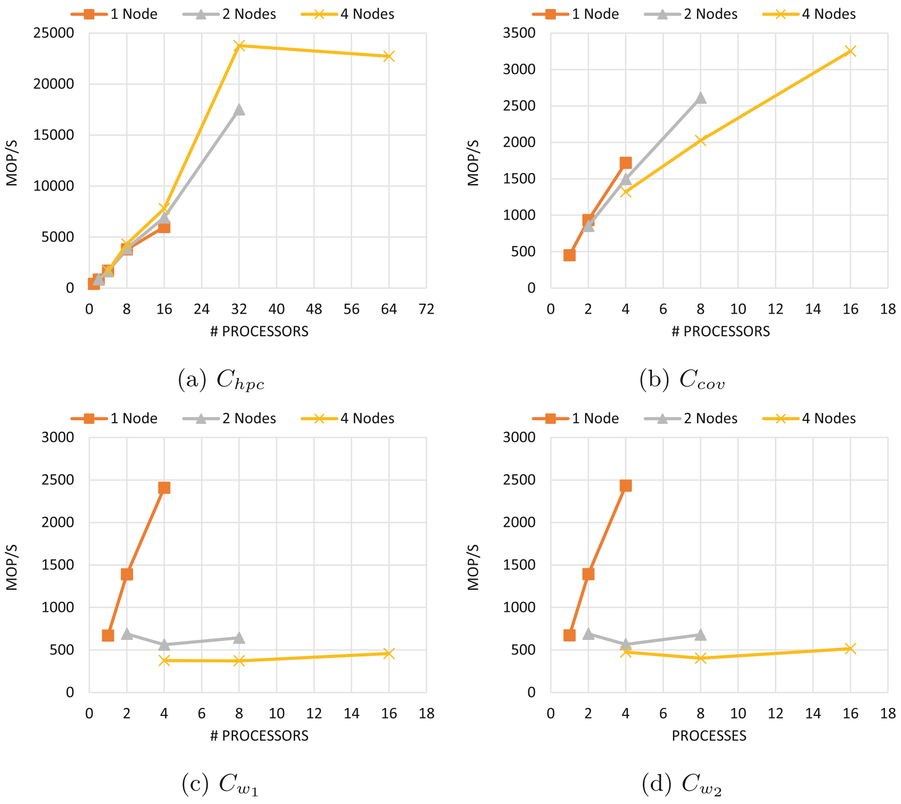
(b) C_{cov}

(c) C_{w_1}

(d) C_{w_2}

Fig. 3. FT - fast Fourier Transform - spectral methods

In FT, Fig. 3, C_{hpc} achieves strong scalability when compared to the other clusters, as expected. However, this cluster exhibits a particular performance behavior which is the divergence in the performance seen on different settings of nodes from 16 processors. This divergence is explained by the NUMA memory architecture commonly present in high-performance computing cluster nodes. Here the performance attained for one, two, and four nodes is similar until the number of processors used per node cross the processors' capacity of a single socket or NUMA domain. This suggests intra-node communication issues being rendered in the node performance degradation. Note that these issues are solved when we distributed 16 processes in two or four nodes, that is, by considering $2n * 8p$ and $4n * 4p$ settings, respectively. Regarding the non-HPC computing systems, although C_{cov}, C_{w_1} and C_{w_2} achieve strong single node scaling, C_{cov} outperform the multi-node performance of C_{w_1} and C_{w_2}. Finally, even though C_{cov} exhibits better inter-node communication capabilities, better overall performance is achieved for a single node in either C_{w_1} or C_{w_2} clusters.



(a) C_{hpc}

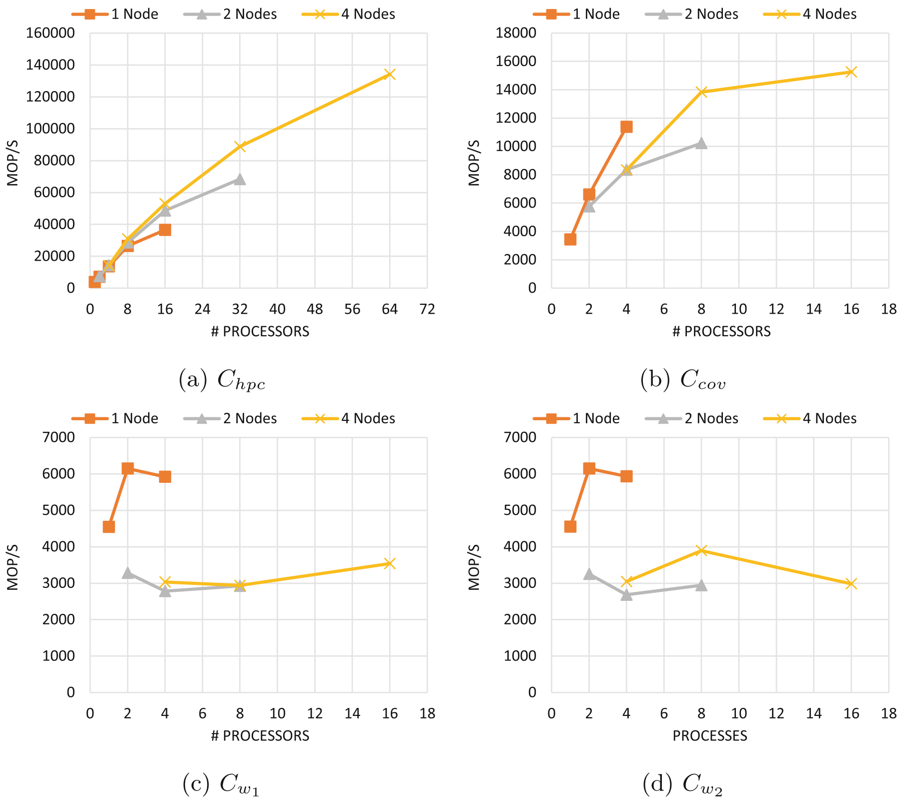
(b) C_{cov}

(c) C_{w_1}

(d) C_{w_2}

Fig. 4. CG - Conjugate Gradient - sparse linear algebra and unstructured meshes

In CG, Fig. 4, C_{hpc} also demonstrates an upward trend in performance when increasing the number of nodes and processors; however, the effect of the NUMA memory architecture substantially hurts the performance of this computing kernel. For example, for 32 processors, the $4n * 8p$ nodes-processors setting outperforms the $2n * 16$ setting, since in the former setting, the eight processors per node do not cross the boundaries of a single node socket. The above-mentioned NUMA effect is also seen for 64 processors, but here the increasing performance tendency dramatically drops. Concerning the non-HPC computing systems, all sustain scalable performance for a single node; however, C_{cov} exceeds the multi-node performance of its counterparts. In addition, C_{cov} achieves the best overall performance in four nodes. If we compare the maximum performance achieved in the whole C_{cov} cluster and the maximum performance achieved in a single C_{hpc} node, C_{cov} just reaches 54.29% of the C_{hpc} single node capacity. Moreover, if we compare the maximum performance achieved in C_{w_2} against the one achieved in C_{cov}, a single C_{w_2} node reaches 74.79% of the whole C_{cov} cluster capacity.



(a) C_{hpc}

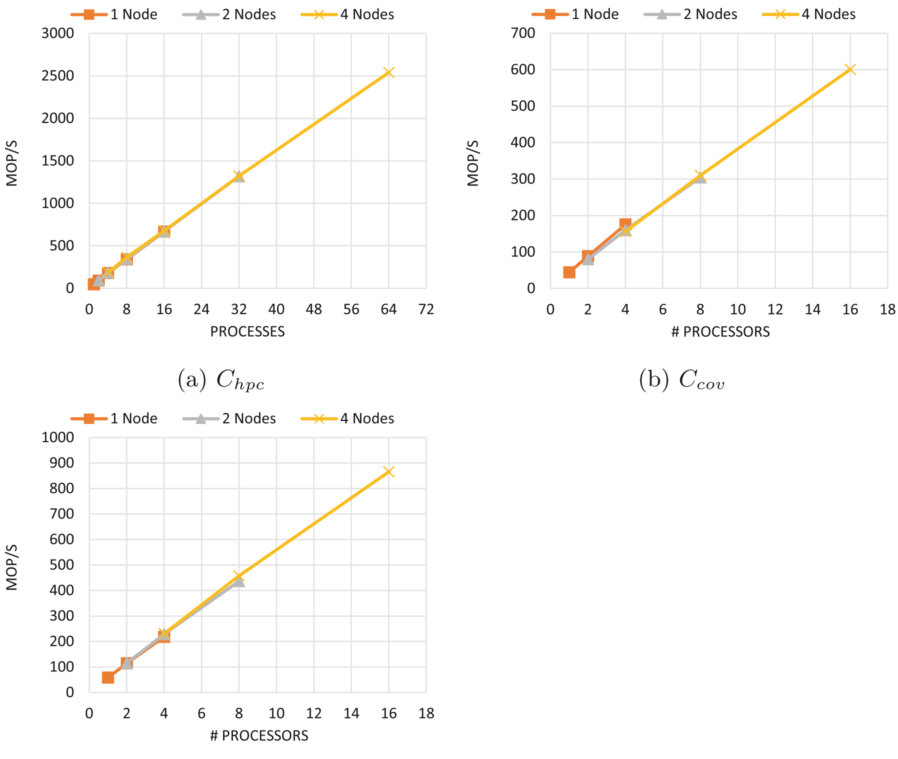
(b) C_{cov}

(c) C_{w_1}

(d) C_{w_2}

Fig. 5. MG - Multi Grid - structured meshes

In MG, Fig. 5, C_{hpc} demonstrates the same upward scaling pattern seen in the FT computing kernel. This similarity suggests the same communication and computation pattern. Both kernels solve the Poisson equation and employ short- and long-distance communication operations, but MG, unlike FT, is a memory-intensive kernel [14]. In the non-HPC computing systems, this computing kernel reports poor scalability in C_{cov}, and no scalability is seen in the C_{w_1} and C_{w_2} clusters. In particular, the lack of scalability in C_{w_1} and C_{w_2} might be attributed to the memory-intensive nature of MG, suggesting a memory bandwidth issue. Conclusively, for the MG computing kernel, C_{cov} outperform clusters C_{w_1} and C_{w_2} in all node settings, namely one, two and four nodes. Finally, when compared to C_{hpc}, the whole C_{cov} cluster reaches only 41.75% of a single C_{hpc} node capacity.



(a) C_{hpc}

(b) C_{cov}

(c) C_{w_1}

Fig. 6. EP - Embarrassingly Parallel - Monte Carlo

In EP, Fig. 6, clusters C_{hpc}, C_{cov}, and C_{w_1} demonstrate strong scalability as expected. Note that C_{w_2} was not considered since it will achieve roughly the same performance of C_{w_1} as computing nodes are the same, but variate in the inter-nodes interconnection. When considering the non-HPC computing systems, the overall performance achieved in cluster C_{w_1} outperforms the performance achieved in C_{cov}. Here, C_{cov} just reaches 69.40% of the overall performance of C_{w1}.

4.4 Clusters Performance Profiles

Tables 4, 5, 6 and 7 provide four performance-derived metrics for the clusters under study; namely *slope*, θ, α, and *coupling*. These metrics characterize the clusters under study with respect to the four fundamental computing kernels: FT, MG, CG, and EP. Note that unlike the previous section where application performance was analyzed per node and per processing unit, here the performance is described for the clusters as whole entities.

Table 4 depicts the characterization computed for the C_{hpc} cluster. The *slope*, as mentioned in Sect. 3, describes the rate of growth in the performance delivered by an application for a given processors and computing nodes setting. A downward trend is expected in this rate as the number of computing nodes and processors climb due to the parallel overhead. This pattern is being exhibited by FT, MG, and EP. CG, on the contrary, describes a particular behavior. Here the rate attained when considering a two-nodes C_{hpc} cluster is higher than the one achieved for one and four nodes C_{hpc} clusters. This is explained by the increasing performance tendency seen in two nodes, Fig. 4a, that is contrary to that observed for one and four nodes. Note that the *slope* also tells us the performance cost of increasing the number of computational resources (processors and nodes). In the case of C_{hpc}, the increase in cost tends to be linear in all applications except for CG.

C_{hpc} proof consistency in the cluster overhead and coupling when considering the expectation that the overhead might tend to increase as more computing resources are added for the computation; as a consequence, the resulting coupling might decrease at the same extent. Note that this expectation is consistent in all applications except for CG which proof a negative -0.0501 cluster overhead. This means that instead of having overhead, the application is exhibiting a performance rate (*slope*) that surpasses the single node rate. Since the trend seen in two nodes C_{hpc} cluster surpass the one exhibited by a single node that was considered as the tightly coupled instance of the cluster for the computation of coupling, -19.9600 can't represent the coupling of the two nodes C_{hpc} cluster; because a single node is not any more a tightly coupled instance of the cluster. Another particular behavior seen in C_{hpc} is the one exhibited for FT. Here the performance rate in FT decreases significantly from two to four nodes; this suggests scalability issues of FT for many computing nodes. Lastly, in general, C_{hpc} exhibits a cluster overhead close to zero and higher degree of coupling, being this a characteristic of high-performance computing systems. However, cluster overhead and coupling will really make sense when we are able to compare these values with the ones obtained from other clusters.

Table 5 depicts the characterization computed for the C_{cov} cluster. Here, the *slope* decreases significantly describing a polynomial behavior, except for CG

Table 4. Cluster performance profile C_{hpc}

Kernel	**Nodes**	$(p, P'(p))$	$(p+1, P'(p+1))$	*slope*	θ	α	*coupling*
FT	**1**	(1,1762.41)	(16,16184.33)	961.4613	89.9404	0.0000	–
	2	(1,1762.41)	(32,28131.62)	850.6197	89.9326	0.0078	128.7781
	4	(1,1762.41)	(64,19124.67)	275.5914	89.7921	0.1483	6.7427
MG	**1**	(1,3810.23)	(16,36541.11)	2182.0587	89.9737	0.0000	–
	2	(1,3810.23)	(32,68435.4)	2084.6829	89.9725	0.0012	815.3292
	4	(1,3810.23)	(64,134120.4)	2068.4154	89.9723	0.0014	693.1671
CG	**1**	(1,406.55)	(16,5991.34)	372.3193	89.8461	0.0000	–
	2	(1,406.55)	(32,17519.91)	552.0439	89.8962	−0.0501	−19.9600
	4	(1,406.55)	(64,22721.68)	354.2084	89.8382	0.0079	127.0909
EP	**1**	(1,46.94)	(16,666.79)	41.3233	88.6137	0.0000	–
	2	(1,46.94)	(32,1318.02)	41.0026	88.6029	0.0108	92.2507
	4	(1,46.94)	(64,2540.23)	39.5760	88.5526	0.0612	16.3457

where there is a linear decrease in *slope*. Regarding cluster overhead, this cluster keeps consistency in the expectation that the overhead might tend to increase as more computing resources are added for the computation; as a consequence, there is also a consistency in the resulting coupling. Note that the values for cluster overhead and coupling on this cluster can be compared to the ones achieved in C_{hpc}; however, when doing this comparison, we need to consider that C_{hpc} was evaluated up to 16 processors while C_{cov} only considered four. If we consider C_{hpc} as a base line, we can conclude that C_{cov} exhibits the characteristics of a loosely coupled computing system; these are higher cluster overhead and lower coupleness.

Table 5. Cluster performance profile C_{cov}

Kernel	**Nodes**	$(p, P'(p))$	$(p+1, P'(p+1))$	*slope*	θ	α	*coupling*
FT	**1**	(1,1535.95)	(4,4731.63)	1065.2267	89.9462	0.0000	–
	2	(1,1535.95)	(8,2307.81)	110.2657	89.4804	0.4658	2.1468
	4	(1,1535.95)	(16,3611.28)	138.3553	89.5859	0.3603	2.7753
MG	**1**	(1,3428.22)	(4,11388.69)	2653.4900	89.9784	0.0000	–
	2	(1,3428.22)	(8,10229.9)	971.6686	89.9410	0.0374	26.7567
	4	(1,3428.22)	(16,15256.62)	788.5600	89.9273	0.0511	19.5825
CG	**1**	(1,449.58)	(4,1718.61)	423.0100	89.8646	0.0000	–
	2	(1,449.58)	(8,2611.97)	308.9129	89.8145	0.0500	19.9891
	4	(1,449.58)	(16,3252.4)	186.8547	89.6934	0.1712	5.8417
EP	**1**	(1,44.35)	(4,175.16)	43.6033	88.6862	0.0000	–
	2	(1,44.35)	(8,304.24)	37.1271	88.4571	0.2291	4.3656
	4	(1,44.35)	(16,600.7)	37.0900	88.4556	0.2306	4.3363

Table 6 show the performance derived metrics computed for the C_{w_1} cluster. Although we expected a downward trend in *slope* as we increased the number of computing nodes and processors, C_{w_1} depicts an unexpected trend. We observe negative *slopes* for the communication intensive computing kernels, this being the rates of performance degradation. We also observe that for the FT and MG computing kernels the performance degradation rate is greater in the two nodes setting than in the four nodes setting, whereas in GC the major degradation rate takes place, as expected, in the four nodes setting. We attributed the higher degradation in the two nodes setting to the poor network capacity of this cluster, this degradation seems to be compensated by the amount of computing processors used in the four nodes cluster setting. The equivalent behavior seen in FT and MG obeys the similarity these computing kernels have in terms of computation and communication. Finally, the negative *slope* renders higher cluster overheads, thus poor cluster coupling.

Table 6. Cluster performance profile C_{w_1}

Kernel	**Nodes**	$(p, P'(p))$	$(p+1, P'(p+1))$	*slope*	θ	α	*coupling*
FT	**1**	(1,2261.71)	(4,6264.74)	1334.3433	89.9571	0.0000	–
	2	(1,2261.71)	(8,363.13)	−271.2257	−89.7888	179.7458	0.0056
	4	(1,2261.71)	(16,500.53)	−117.4120	−89.5120	179.4691	0.0056
MG	**1**	(1,4544.48)	(4,5920.05)	458.5233	89.8750	0.0000	–
	2	(1,4544.48)	(8,2924.32)	−231.4514	−89.7525	179.6275	0.0056
	4	(1,4544.48)	(16,3539.47)	−67.0007	−89.1449	179.0200	0.0056
CG	**1**	(1,670.57)	(4,2408.03)	579.1533	89.9011	0.0000	–
	2	(1,670.57)	(8,644.73)	−3.6914	−74.8425	164.7436	0.0061
	4	(1,670.57)	(16,458.49)	−14.1387	−85.9543	175.8554	0.0057
EP	**1**	(1,57.69)	(4,217.92)	53.4100	88.9274	0.0000	–
	2	(1,57.69)	(8,436.51)	54.1171	88.9414	−0.0140	−71.3638
	4	(1,57.69)	(16,865.55)	53.8573	88.9363	−0.0089	−112.2702

Table 7 contains the performance derived metrics computed for the C_{w_2} cluster. This cluster demonstrates the same performance degradation rate pattern

Table 7. Cluster performance profile C_{w_2}

Kernel	**Nodes**	$(p, P'(p))$	$(p+1, P'(p+1))$	*slope*	θ	α	*coupling*
FT	**1**	(1,2023.55)	(4,6031.51)	1335.9867	89.9571	0.0000	–
	2	(1,2023.55)	(8,368.31)	−236.4629	−89.7577	179.7148	0.0056
	4	(1,2023.55)	(16,508.38)	−101.0113	−89.4328	179.3899	0.0056
MG	**1**	(1,4550.05)	(4,5934.79)	461.5800	89.8759	0.0000	–
	2	(1,4550.05)	(8,2940.89)	−229.8800	−89.7508	179.6266	0.0056
	4	(1,4550.05)	(16,2982.86)	−104.4793	−89.4516	179.3275	0.0056
CG	**1**	(1,671.32)	(4,2432.56)	587.0800	89.9024	0.0000	–
	2	(1,671.32)	(8,679.92)	1.2286	50.8560	39.0464	0.0256
	4	(1,671.32)	(16,515.7)	−10.3747	−84.4943	174.3968	0.0057

seen in C_{w_1}, since clusters computing nodes are the same except for the nodes interconnection. Although the negative *slopes*, cluster overhead and cluster coupling metrics are similar on C_{w_1} and C_{w_2} for FT and MG, there are subtle improvements in these metrics due to the improvement in the interconnection network in the C_{w_2} cluster. In addition, the network enhancement improved the coupling of the cluster in terms of the CG computing kernel.

5 Discussion

A scalability analysis and the one focused on cluster overhead and coupling metrics, were conducted on four scientific computing clusters in order to quantitatively characterize their capacity to execute scientific applications. The scalability analysis suggests memory bandwidth issues in C_{w_1} and C_{w_2} that might prevent the scalability of memory intensive computing kernels such as MG. In addition, the analysis demonstrates the negative effect of the NUMA memory architecture, established in C_{hpc} cluster nodes, that can slightly affect the scalability of scientific applications exhibiting computing patterns similar to FT and MG; being these spectral methods and structured meshes. But the NUMA architecture substantially hurts the scalability of sparse linear algebra and unstructured meshes computations such as the ones exhibited in CG for the C_{hpc} cluster. Finally, when considering the non-HPC clusters, C_{w_1} and C_{w_2} demonstrate best overall performance for FT in one node and EP in four nodes whilst C_{cov} demonstrate best overall performance in CG and MG in the four nodes setting.

The cluster performance profile reveals numerous cluster specific behaviors that might be considered, first, for the selection of clusters for specific scientific applications, and, second, to guide architectural design decisions on the development of these clusters. In the first matter, when considering the non-HPC clusters, under the experimental conditions, the workstations based clusters C_{w_1} and C_{w_2} best fits the needs of FT and EP workloads, for the given problem size and the only-MPI parallelization scheme. On the other hand, the virtual machines based cluster C_{cov} best fits the need of CG and MG workloads.

In the second matter, for instance, the performance rate of most computing kernels executed in C_{hpc} decreases linearly when increasing the number of computing resources. In contrast, C_{cov} exhibits polynomial decrease in this rate and C_{w_1} and C_{w_2} exhibit negative performance rates. Note that these performance rates can be improved by cluster designers by considering enhancements, for example, in the nodes interconnection as demonstrated when improving the interconnection of C_{w_1} in C_{w_2}.

6 Conclusion

In this work we proposed the cluster performance profile, this profile comprises performance related metrics for specific computing kernels and cluster specific metrics derived from the performance exhibited on these computing kernels. This profile was introduced to support both researchers running scientific applications on HPC and non-HPC clusters and cluster designers mainly developing low cost scientific computing clusters.

In this regard, the profile delivers two main benefits for researchers; first, serve as a guide for non HPC experts to determine what kind of scientific applications would strong scale, in a given cluster, when increasing the number of computing resources and which might not; second, as this profile is based on well-known building blocks seen in many scientific applications, it can be considered as a first glance when determining the appropriate cluster for the execution of applications developed by the combination of these building blocks. In addition, the cluster performance profile also serves as a guideline for cluster designers that will be able to perform improvements led by metrics such as the cluster overhead and coupling on scientific computing clusters.

Future work may involve the use of the cluster performance profile in large scale scientific computing clusters to fulfill two main objectives; first, determine the validity of the proposed quantitative characterization in this type of clusters; then, determine how this profile constructed on well-known building blocks executed on large scale clusters can anticipate the performance that could be achieved from applications made by the combination of well-known scientific computing building blocks.

References

1. Aljamal, R., El-Mousa, A., Jubair, F.: Benchmarking Microsoft Azure virtual machines for the use of HPC applications. In: 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 382–387. IEEE (2020)
2. Beserra, D., et al.: Performance evaluation of hypervisors for HPC applications. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 846–851. IEEE (2015)
3. Beserra, D., Pinheiro, M.K., Souveyet, C., Steffenel, L.A., Moreno, E.D.: Performance evaluation of OS-level virtualization solutions for HPC purposes on SoC-based systems. In: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), pp. 363–370. IEEE (2017)
4. Chavarriaga, J., Gómez, C.E., Bonilla, D.C., Castro, H.E.: Capacity of desktop clouds for running HPC applications: a revisited analysis. In: Florez, H., Leon, M., Diaz-Nafria, J.M., Belli, S. (eds.) ICAI 2019. CCIS, vol. 1051, pp. 257–268. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32475-9_19
5. Ebbers, M., Hastings, C., Nuttal, M., Reichenberg, M.: Introduction to the new mainframe: networking. Copyright IBM Corp (2006)
6. 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 (2010)
7. Huse, L.P., Bugge, H.: High-end computing on SHV workstations connected with high performance network. In: Sørevik, T., Manne, F., Gebremedhin, A.H., Moe, R. (eds.) PARA 2000. LNCS, vol. 1947, pp. 324–332. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-70734-4_38
8. Mehrotra, P., et al.: Performance evaluation of amazon elastic compute cloud for NASA high-performance computing applications. Concurr. Comput. Pract. Exp. **28**(4), 1041–1055 (2016)

9. Muraña, J., Nesmachnow, S.: Simulation and evaluation of multicriteria planning heuristics for demand response in datacenters. Simulation 00375497211020083 (2021)
10. Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., Epema, D.: A performance analysis of EC2 cloud computing services for scientific computing. In: Avresky, D.R., Diaz, M., Bode, A., Ciciani, B., Dekel, E. (eds.) CloudComp 2009. LNICSSTE, vol. 34, pp. 115–131. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12636-9_9
11. Rajovic, N., Carpenter, P.M., Gelado, I., Puzovic, N., Ramirez, A., Valero, M.: Supercomputing with commodity CPUs: are mobile SoCs ready for HPC? In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1–12 (2013)
12. Saini, S., et al.: An application-based performance evaluation of NASA's nebula cloud computing platform. In: 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, pp. 336–343. IEEE (2012)
13. Setiawan, I., Murdyantoro, E.: Commodity cluster using single system image based on Linux/Kerrighed for high-performance computing. In: 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), pp. 367–372. IEEE (2016)
14. Vivas, A., Castro, H.: Estimating the overhead and coupling of scientific computing clusters. Simulation 00375497211064198 (2021)