Evaluation of the Performance of Tightly Coupled Parallel Solvers and MPI Communications in IaaS From the Public Cloud

Arturo Fernandez

Abstract—IaaS from the public cloud is becoming a new option for organizations in need of HPC capabilities. IaaS offers greater flexibility in hardware choices and operational costs. Furthermore, IaaS enables small organizations to access resources previously reserved for organizations with larger capitals. This article uses the HPCG benchmark to assess the performance of parallel solvers, which are critical in computational engineering, and microbenchmarks to measure collective MPI operations. The benchmarks encompass IaaS from five cloud vendors (AWS, Google Cloud Platform, Azure, Oracle CIoud Infrastructure, and Packet), and two architectures, ARM and x86_64. The benchmarks cover clusters with up to 4 500 cores and illustrate the benefits of higher network bandwidth when scaling up clusters. The results for some of the clusters are particularly promising as they exhibit good scalability and compare well to on-premises supercomputers. Additionally, the study includes a preliminary cost estimate based on on-demand prices for IaaS computational power and memory.

Index Terms—Benchmark testing, high performance computing, multicore processing, parallel architectures

1 Introduction

The advent of public and private clouds has shifted enterprise software such as databases or web hosting services from being mostly administered by on-premises hardware to cloud datacenters. A similar transition for scientific applications has lagged largely because of the technical challenges presented by this type of software, which requires the integration of many processes working in parallel with high memory demands and extensive input/output (I/O) operations. Accelerating this transition to cloud environments is highly desirable due to several motives. First, clouds offer greater flexibility in terms of number of cores, fluctuating workloads or short and long-term storage solutions. Cloud users can benefit from gaining access to the latest hardware much more quickly than on-premises users. There is also a financial motivation that includes the shift from a large upfront investment to capital expenditure along with reductions in the deployment and maintenance times associated with cluster operation. Lastly, an element that should not be overlooked is that the public cloud allows small organizations to access capabilities traditionally reserved to large businesses, national labs, and research universities.

A broad classification of scientific software relies in cataloguing it depending on how a large task is split into smaller processes advancing in parallel. When the parallel processes require exchanging information with others almost continuously, scientists use the term 'tightly coupled simulations.' When the processes can progress without the need to send or receive information from other processes, the terms commonly used to describe this scenario are either 'loosely coupled simulations' or High-Throughput Computing (HTC). The term High Performance Computing (HPC) is a bit ambiguous because it has different interpretations. In a broad sense, HPC simply indicates the performance of parallel processing in an efficient way and would encompass the abovementioned two scenarios. In a stricter or more academic sense, HPC refers only to 'tighly coupled simulations' and the present paper adopts this viewpoint. The distinction between the HTC and HPC paradigms is critical because the performance of the former depends mostly on the raw computational power, while the latency and bandwidth of the network connecting the parallel processes greatly affects the performance of the latter. One question that every HPC application must address is how to complete any exchange of information between parallel processes. Although several solutions have been proposed, a majority of HPC applicationss use MPI [1], [2] as the de facto API, which is also the situation in the present study. An important area of HPC is scientific software and, more specifically, engineering applications involving the discretization and solution of Partial Differential Equations (PDEs) describing physical systems. This discretization results in systems of equations with many millions of grid points and discrete variables, which require the use of efficient parallel solvers.

On-premises supercomputers have traditionally been the hardware of choice for state-of-the-art HPC. On the other hand, the interest here resides in Infrastructure as a Service (IaaS) offered by cloud service providers (CSPs) and publically available. The migration of HPC applications to new

• *The author is with ODYHPC, Murrysville, PA 15668 USA. E-mail: afernandez@odyhpc.com.*

Manuscript received 19 June 2019; revised 9 January 2021; accepted 14 January 2021. Date of publication 19 January 2021; date of current version 6 December 2022.
(Corresponding author: Arturo Fernandez)
Recommended for acceptance by G. Fox.
Digital Object Identifier no. 10.1109/TCC.2021.3052844

hardware benefits from understanding its strengths and weaknesess, which can be accomplished via benchmarks. The present work seeks to evaluate the performance of parallel solvers on IaaS with the aid of the HPCG (High-Performance Conjugate Gradient) benchmark [3] and MPI communications with the OSU microbenchmark suite or OMB [4]. As the number of CSPs offering IaaS is numerous nowadays, the present work limits its scope to a selection based on availability and technical criteria: Clustering must be feasible, and computational resources need to be available on a pay-as-you-go or on-demand basis in North America. Using this criterion, the selection includes four major CSPs: Amazon Web Services or AWS [5], Azure or AZ [6] from Microsoft Corporation, Google Cloud Platform or GCP [7], and Oracle Cloud Infrastructure or OCI [8], along with a relatively new company, Packet or PacN [9], founded in 2014 and that offers a variety of baremetal options. The benchmark results shown here reflect measurements performed in the 2nd quarter of 2020 with version 3.1 of HPCG and 5.6.2 of OMB.

2 Review of Related Works

Hardware benchmarking has become routine with specialized apps targeting different aspects of the underlying architecture for the intended use. The most common benchmark for HPC performance is the High Performance Linpack (HPL) benchmark reviewed by Dongarra *et al.* [10]. One of the reasons for this popularity is that it serves to score the semi-annual top500.org list [11] that ranks the fastest supercomputers in the world and is closely followed by the community. HPL involves the solution to a random system of linear equations represented by a dense matrix. The procedure to solve this system uses a two-dimensional block-cyclic data distribution and right-looking variant of the LU factorization with row partial-pivoting. Although HPL devises the raw power of the hardware being tested, it is less robust in evaluating the significance of communication latency and bandwidth critical in the numerical solution of PDEs. To overcome this deficiency, Dongarra *et al.* [3] introduced the HPCG benchmark, which solves a sparse matrix more representative of the discretized equations commonly found in computational engineering. The solution to the system involves a domain decomposition with a conjugate gradient method using an additive Schwarz preconditioner; additionally, a symmetric Gauss-Seidel sweep preconditioner is applied in each subdomain. These characteristics result in HPCG being much more memory-bound than HPL and having higher communication overhead. Hence, HPCG results are much lower than those measured with HPL, usually by about two orders of magnitude. Other teams have introduced other benchmarks such NASA's Parallel Benchmarks (NPB) [12] or the STREAM [13] suite.

Since the launch of AWS by Amazon in 2006, several groups have examined some aspects related to the performance of HPC software in different cloud platforms. An extensive discussion of the early state-of-the-art was the Magellan report [14] commissioned by the U.S. Department of Energy. The main purpose of this study was the assessment of cloud computing as an alternative platform for pplications from its Office of Science. The report recommended mid-level software along with applications with minimal communications and I/O as the most optimal candidates for migration to the cloud. Studies following the Magellan report include the works of Roloff *et al.* [15], Mauch *et al.* [16], Gupta *et al.* [17], Sadoghi *et al.* [18], Mohammed et Bazhirov [19], Kotas *et al.* [20] and Chang *et al.* [21]. Roloff *et al.* [15] used HPL, STREAM and NPB to evaluate the performance of AWS and AZ instances finding the overhead due to the hypervisor to have minimal impact on performance. Mauch *et al.* [16] also used HPL to evaluate AWS instances and proposed the use of Infiniband network connections to increase overall performance. Gupta *et al.* [17] used NPB along with 5 other apps to compare 6 different platforms including AWS. At the time of their study in 2014, they conjectured that clouds could be a complement but not a substitute to supercomputers, and that further research would be necessary to accomplish this goal. Sadoghi *et al.* [18] evaluated the performance of AWS instances and S3/EBS storage services related to memory, network, and I/O for a variety of conditions and workloads. Mohammadi and Bazhirov [19] used HPL to compare clusters composed by chosen instances from AWS, Azure, IBM Softlayer, and Rackspace. Their assessment indicated the AWS clusters composed with c4.8xlarge instances to exhibit the best HPL performance, and Azure to have the best scalability because of the Infiniband network. Kotas *et al.* [20] also used HPL and HPCG, among other apps, to compare the performance of clusters built with AWS c4.8xlarge and Azure H16r instances. Kotas *et al.* [20] noticed a lower performance of the Azure clusters despite their higher clock speed and network bandwith. Chang *et al.* [21] evaluated a potential migration of some of NASA's High-End Computing Capabilities (HECC) workload to public clouds by assessing the performance and cost of HECC's in-house resources versus AWS and Penguin-on-Demand (POD) clusters. Chang *et al.* [21] detailed better overall performance and scalability for HECC clusters than AWS clusters, which outperformed their POD counterparts. In their conclusions, Chang *et al.* [21] discussed the higher communication overhead from cloud clusters to cause loss performance. Recently, Netto *et al.* [22] introduced a taxonomy framework to classify public and hybrid clouds. In addition to the classical application of HPC in engineering, an emerging area is machine learning training. MLPerf [23] has incorporated a new suite targeting HPC training with the initial results posted in November of 2020 [24].

3 Results

The next subsections present the results divided into three sets of benchmarks depending on hardware size. CSPs offer units of minimum computational resources that receive different names such as instance, virtual machines or server. The present works uses the word 'instance' regardless of the vendor's notation. The most basic instance thus involves processing power from one or more cores, storage drive attached to it, and a virtual network facilitating the external connection to the instance and between instances. The first set of benchmarks corresponds to single instances with memories up to 256 GB and using 16 MPI ranks. It includes instances with and without active hyperthreading and would be the equivalent of workstations or small on-premises clusters. They are

relatively quick to set up and provide insight into the performance from the different CSPs' instance series. The second and third sets serve to evaluate IaaS able to perform larger computations and their distinction is based on computational power, mainly the available memory and number of cores. The second set encompasses configurations with a maximum core number of 128 and memories up to 2 TB. The third set involves clusters with memories beginning at 2 TB and a maximum core number of 4500, which enables running up to 9000 MPI ranks in hyperthreading mode. These configurations usually require a virtual cloud network insterconnecting the instances and able to support MPI communications. Due to the ample offer of IaaS, the study is naturally selective, and the third set builds on some of the best performers from the second set.

3.1 Evaluation of Single Instance Performance

HPC performance depends on several factors but the most outstanding ones are computational power, memory bandwidth, and network performance. For HPC purposes, computational power refers broadly to the ability of the processor to perform double-precision operations. The computational power of each instance depends thus on core count, clock frequency, and the theoretical number of double precision floating point operations per cycle (DP-FLOPs/cycle) for that specific processor family. The latter parameter ranges from 2 DP-FLOPs/cycle for ARM processors to 32 DP-FLOPs/cycle for the latest Intel Xeon Platinum and Gold Scalable processors.

The first tests encompass only single instances with the number of MPI ranks set to 16 for both single and hyperthreaded benchmarks. This is a modest figure, but it provides a baseline and explores hyperthreading significance. Table 1 lists the instance general characteristics. The first 12 instances, identified with capital letters, use 16 cores and each MPI rank is threaded in a single core. The other 12 instances, identified with little letters, have 8 cores and hyperthreading is active. Among the selected instances, the only one based on ARM architecture is AWS Graviton. The other instances are built on x86_64 architecture using either AMD or Intel processors. Additionally, Table 1 lists the DP-FLOPs/cycle for each processor and hypervisors for each instance. In this first group, the only instances without a hypervisor are the Packet baremetal instances (B, E, J, K).

The present benchmarks use version 3.1 of HPCG, which results in higher results than version 3.0. The compilation of the apps uses the highest level of optimization for speed (e.g., -O3) and builds to each specific architecture (e.g., using flags such as -march and -mtune). However, they do not link with optimized libraries potentially able to increase the outcome. This decision is intentional so that variations in readings are due to the underlying hardware and not the HPCG implementation. The present benchmarks use the default grid of 104x104x104 per MPI rank, which roughly roughly requires 0.75 GB of memory per rank. Some preliminary tests involved variations in the problem size, but the benchmarks exhibited homogeneous results except for small problems (e.g 32x32x32 per MPI rank), which also led to short computational times. After installing the apps and their dependencies, the instance benchmarks involve repeating the measurements between 3 and 6 times to assess their consistency. As it is common with most benchmarks, the results can exhibit some variability depending on software and hardware characteristics. From the hardware point-of-view, IaaS frequently uses shared resources with the potential to affect performance. The only instances from this group guaranteed to work in isolation are the Packet baremetal instances, but the performance from the other instances can theoretically fluctuate over time depending on how much resource sharing is taking place at the time. In practice, and during the completion of the present tests, the HPCG results fluctuate within 2 percent and the graphs reflect the highest measured value. Hereafter, and to be able to analyze the findings, a figure of merit is built by dividing the HPCG result by the number of cores. The units for this figure of merit is ((GFLOP/s)/core). Fig. 1 shows the HPCG figure of merit for single instances. Instance A based on ARM architecture exhibits the weakest performance. The benchmarks reflect weaker performance from AMD cores than their Intel counterparts at similar clock speeds (b vs. l or F vs. G) partially because of their lower DP-FLOPs/cycle. The best performers come from the Intel Xeon Platinum family of processors that outperform other Intel Xeon processors due to their higher DP-FLOPs/cycle. Fig. 1 also facilitates an evaluation of the impact of hyperthreading on HPC performance, which has been a topic of discusssion among system architects and users. When measuring the performance per core, hyperthreaded jobs generally exhibit better performance than singlethreaded ones as they take advantage of the underlying infrastructure. For example, the instances 'c', 'j' and 'l' have the same processor and configuration as instances 'C', 'G' and 'H,' respectively, except that the former use hypethreading. When compared versus the singlethreaded benchmarks, the performance gain is only 3.3 percent for the AMD-based instance, but it increases to 29.8, and 57.3 percent for the Intel-based instances.

TABLE 1
Instance Characteristics

Id	CSP	Instance	GB	CPU family	DP-FLOPs/cycle	HVisor
A	AWS	a1.4xlarge	16	AWS Graviton (2.3 GHz)	2	Nitro
B	PacN	c2.medium	64	AMD EPYC 7401 (2.0 GHz)	8	N/A
C	AZ	D32as.v4	128	AMD EPYC 7452 (2.35 GHz)	8	MS
D	OCI	VM.E3.Flex	128	AMD EPYC 7742 (2.25 GHz)	8	KVM
E	PacN	c3.medium	64	AMD EPYC 7402P (2.8 GHz)	16	N/A
F	AWS	c5.9xlarge	72	Intel X. Platinum 8124 (3.0 GHz)	32	Nitro
G	AZ	F32s.v2	112	Intel X. Platinum 8167 (2.7GHz)	16	MS
H	AZ	H16	112	Intel X. E5-2667 v3 (3.2 GHz)	16	MS
I	GCP	N1.highcpu.16	28	Intel X. GCP-custom (2.3 GHz)	16	KVM
J	PacN	m1.xlarge	256	Intel X. E5-2650 v4 (2.2 GHz)	16	N/A
K	PacN	m2.xlarge	384	Intel X. Gold 5120 (2.2 GHz)	32	N/A
L	OCI	VM.2.16	240	Intel X. Platinum 8167 (2 GHz)	32	KVM
a	AWS	m5a.4xlarge	64	AMD EPYC 7571 (2.52 GHz)	8	Nitro
b	OCI	VM.E2.8	64	AMD EPYC 7551 (2.0 GHz)	8	KVM
c	AZ	D16as.v4	64	AMD EPYC 7452 (2.35 GHz)	16	MS
d	OCI	VM.E3.Flex	64	AMD EPYC 7742 (2.25 GHz)	16	KVM
e	GCP	N2D.16	64	AMD EPYC 7B12 (2.25 GHz)	16	KVM
f	AWS	c5.4xlarge	32	Intel X. Platinum 8124 (3.0 GHz)	32	Nitro
g	AWS	m5.4xlarge	64	Intel X. Platinum 8175 (2.5 GHz)	32	Nitro
h	AWS	r5.4xlarge	128	Intel X. Platinum 8175 (2.5 GHz)	32	Nitro
i	AZ	D16sv3	64	Intel X. E5-2623 v3 (2.44 GHz)	16	MS
j	AZ	F16s.v2	112	Intel X. Platinum 8167 (2.7 GHz)	16	MS
k	GCP	c2.16	64	Intel X. GCP-custom (3.1 GHz)	32	KVM
l	OCI	VM.2.8	120	Intel X. Platinum 8167 (2.0GHz)	32	KVM

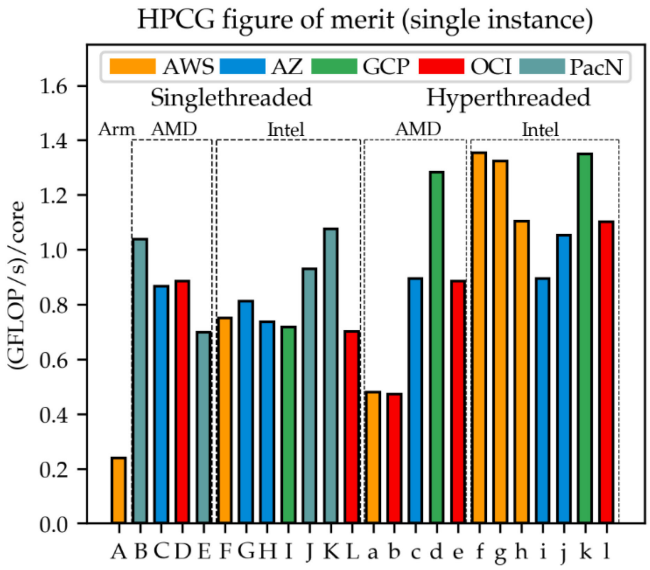

Fig. 1. HPCG figure of merit for single instances on IaaS from the public cloud. Table 1 lists instance characteristics.

Fig. 1 also suggests that computational power cannot predict performance by itself as network communications factor onto the overall efficiency even for single instances. To isolate this effect, microbenchmarks of MPI communications using the OMB suite follow. Due to the myriad of MPI operations, and because of the relatively low number of MPI ranks for these single instances, the first choice for communication microbenchmarks is the MPI_Alltoall operation where all the ranks exchange packets with all the others. Fig. 2 shows the latency times from this operation versus the packet size. In this diagram, the lowest value is dictated by the own network latency, while the latency times for the largest packets depend mainly on the network bandwidth. The latencies are very consistent with the values for singlethreaded instances ranging between 4 and 18 μs. For the 1 MB packet size, the ARM-based instance (A) exhibits the highest latency of 0.27 s, while the other instances exhibit latencies between 0.08 and 0.26 s.

In addition to performance, cost is another fundamental element in assessing any potential migration to the public cloud. To evaluate cost in a preliminary fashion, it is feasible to build an estimate based on the normalized cost for each instance and the HPCG figure of merit:

$$Cost\ [\$/GFLOP(HPCG)] = \frac{\left[\frac{\$}{s}\right]_{instance}\left[\frac{instance}{core\ number}\right]}{GFLOP\ (HPCG)/s}. \quad (1)$$

Fig. 3 shows cost estimates based on on-demand prices in the US (East region where available). Because CSPs usually offer large reductions for reserved jobs or even spot prices, the on-demand price is not necessarily the same price paid by the end-user and Fig. 3 should therefore be only used as a reference. It is incumbent for any organization to evaluate the total cost of ownership. The cost estimate portrays a large cost variability among the different microprocessors and vendors. Fig. 3 also shows hyperthreaded instances to be usually, but not always, more inexpensive. The relative cost between instances driven by AMD and by Intel is similar although the former occupyies the two most economical options.

3.2 Evaluation of Configurations With Up to 128 Cores and 2 TB of Memory

The next group of benchmarks target configurations composed by up to 128 cores and 2 TB of memory. This size makes them suitable for solving many problems in engineering. For example, for a configuration with 96 cores, 2 GB of

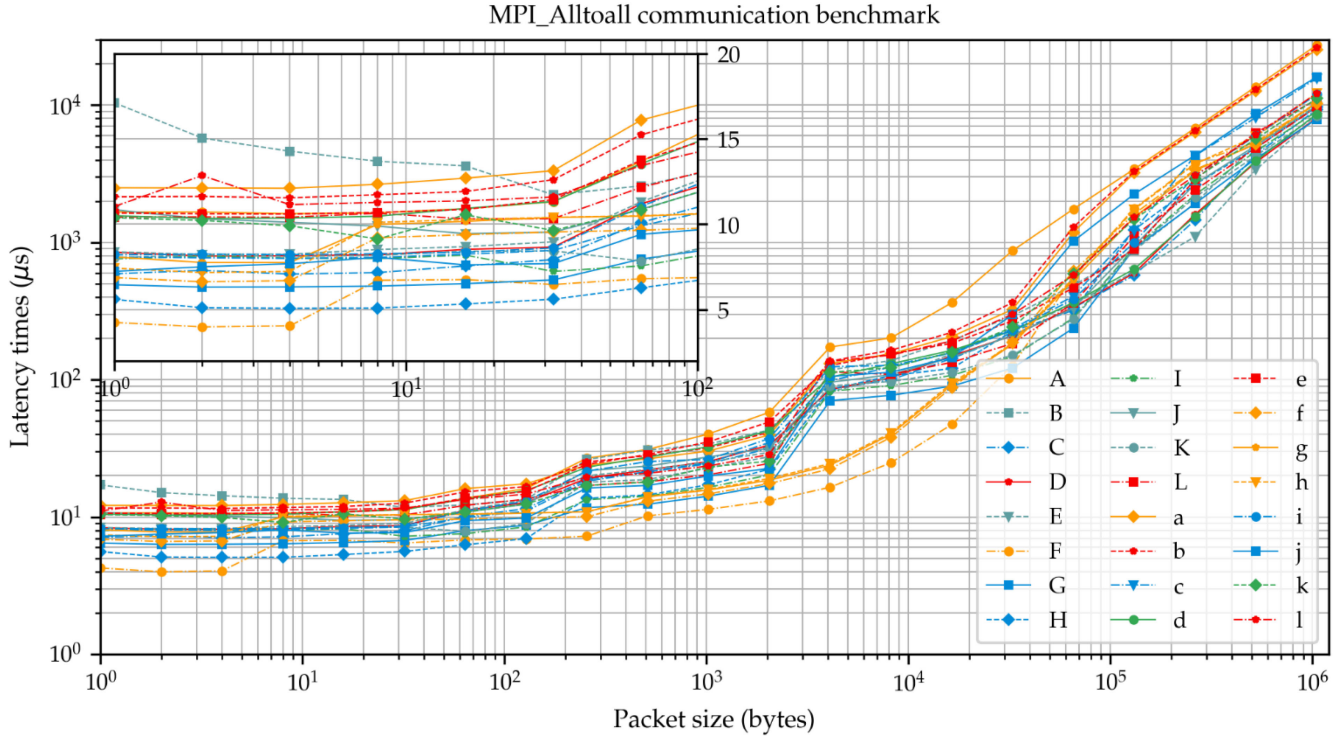

Fig. 2. Latency times from an MPI_Alltoall operation for single instances on IaaS from the public cloud.

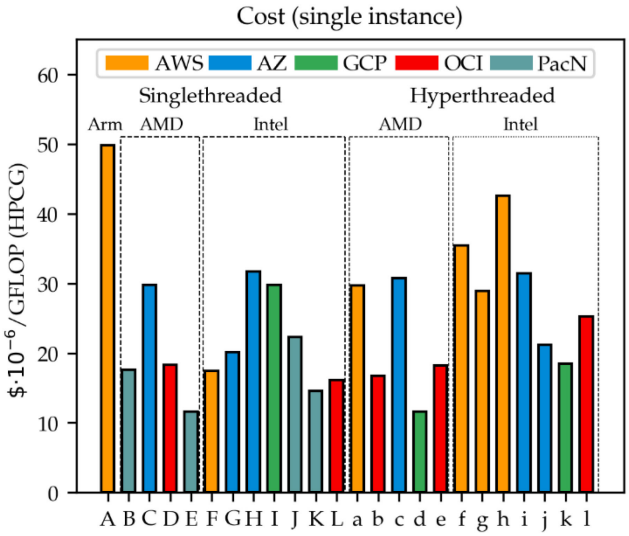

Fig. 3. Cost in $\$\cdot 10^{-6}$/GFLOP(HPCG) for instances on IaaS from the public cloud. Table 1 lists the instance characteristics.

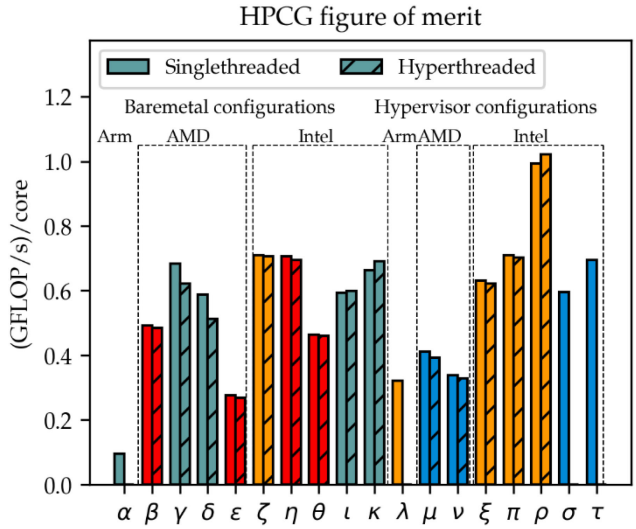

Fig. 4. HPCG figure of merit for the configurations from IaaS in the public cloud listed in Table 2.

memory per core, and considering a problem with 30 variables, this setup would suffice to perform simulations in 3D grids with the number of cells in the range of 10^8. These figures vary, of course, depending on available memory and number of variables. Table 2 lists the configuration characteristics including the cloud vendor, number of cores, total memory in Gigabytes, maximum network bandwidth, and processor family. In this set, two configuratins (α and λ) are based on ARM architecture. The former comes as a single instance using a ThunderX processor launched in 2016 by Cavium. The latter is powered by the new Graviton-2 developed by Annapurna Labs, and made publically available by AWS at the end the second quarter of 2020. The other configurations include a baremetal instance (ε) from OCI with 128 cores, and small clusters composed by between 2 to 6 instances based on x86_64 architecture from Intel and AMD processors. In total, 14 of these systems permit hyperthreading with the exceptions being the ARM and Azure H-series configurations. The first 10 configurations are baremetal, while the other 8 work with hypervisors.

TABLE 2
Configuration Characteristics

Id	CSP	Instance	Cores	Mem	Gbps	Processor family
α	PacN	c1.large.arm	96	128	10	Cavium ThunderX (2.4 GHz)
β	OCI	BM.E2.64	64	512	25	AMD EPYC 7551 (2 GHz)
γ	PacN	c2.medium	72	192	10	AMD EPYC 7401p (2.2 GHz)
δ	PacN	c3.medium	72	192	10	AMD EPYC 7402P (2.8 GHz)
ε	OCI	BM.E3.128	128	2048	50	AMD EPYC 7742P (2.25 GHz)
ζ	AWS	c5n.metal	72	384	100	Intel Xeon Platinum 8124 (3 GHz)
η	OCI	BM.HPC2.36	72	768	100	Intel Xeon Gold 6154 (3 GHz)
θ	OCI	BM.2.52	52	768	50	Intel Xeon Platinum 8167M (2 GHz)
ι	PacN	m1.xlarge	72	768	10	Intel Xeon E5-2640v4 (2.2 GHz)
κ	PacN	m2.xlarge	84	1152	10	Intel Xeon Gold 5120 (2.2 GHz)
λ	AWS	c6g.16xlarge	128	256	25	AWS Graviton 2 (2.3 GHz)
μ	AWS	c5a.24xlarge	96	384	20	AMD EPYC 7R32 (1.75 GHz)
ν	AZ	D64as v4	96	768	10	AMD EPYC 7452 (2.35 GHz)
ξ	AWS	c5.18xlarge	72	384	25	Intel Xeon Platinum 8124 (3 GHz)
π	AWS	c5n.18xlarge	72	384	100	Intel Xeon Platinum 8124 (3 GHz)
ρ	AWS	z1.12xlarge	96	1536	25	Intel Xeon Platinum 8151 (3.4 GHz)
σ	AZ	H16	96	672	10	Intel Xeon E5-2667 v3 (3.2 GHz)
τ	AZ	HC44rs	88	704	100	Intel Xeon Platinum 8168 (2.7 GHz)

Fig. 4 shows the HPCG figure of merit. When available, the singlethreaded and hyperthreaded results are shown side by side to ease their visual comparison. The overall performance depends on several factors, and the current benchmarks facilitate the examination of processor, network bandwidth, hyperthreading, and hypervisor significance. The weakest performance is realized by α using the ThunderX processor. However, the performance from the new AWS-Graviton 2 (λ) is much closer to the x86_64 configurations and, in some cases, it even produces a similar figure of merit. Although a general evaluation points to Intel performing somewhat better than AMD, a closer look reveals some subtle differences. Most bennchmarks with Intel processors exhibit a relatively uniform figure of merit, ranging between 0.6 and 0.7 GFLOP/s per core, with the exceptions being θ at the low end, and ρ at the high end. The latter is powered with Intel Xeon Scalable technology running at a higher clock speed (3.4 GHz) than the others, which reflects on the HPCG results. Other clusters (ε, η, and τ) with a figure of merit within 2 percent of 0.7 GFLOP/s per core are powered with Intel Xeon Platinum processors. The results for AMD are less consistent. Among the configurations using the 1st generation of EPYC processors, γ outperforms β partially because of its higher clock speed. The benchmarks with the EPYC-2 processor exhibit a relatively low figure of merit with δ as the only configuration surpassing 0.5 GFLOP/s per core. The memory hierarchy found in the EPYC-2 chiplet results in a relatively low memory bandwidth when all the cores are exchanging data between themselves. This response harms memory-bound apps or benchmarks such as HPCG. Fig. 4 also attests that processor family and clock speed are, however, not the only factors determining peak performance as network bandwidth has a strong impact on it. This is noticed from both a general assessment and a direct comparison. For example, the clusters ξ and π use the same processor, but the network bandwidth has increased from 25 to 100 Gbps. This variation

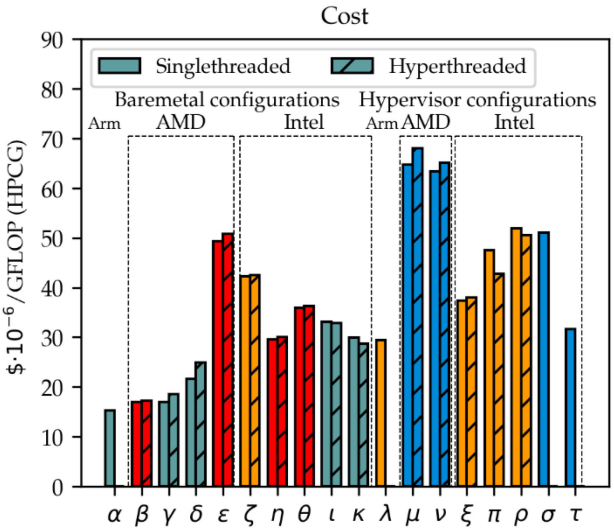

Fig. 5. Cost estimate for the configurations from IaaS in the public cloud listed in Table 2.

results in performance increases of 12.4 and 12.7 percent for single and hyperthreaded benchmarks, respectively. In general, the performance of clusters operating with a hypervisor compares similarly to baremetal clusters. A case in point is the performance of clusters ζ and π, which only differ in the latter being hypervisor enabled. The performance deteriorations are 0.015 percent (singlethreaded) and 0.68 percent (hyperthreaded). Other tests (not shown here) with AWS-z1d.metal instances resulted in similar outcomes when compared versus cluster ρ. These results point out two conditions: (i) cluster performance depends more strongly on other factors such as processor or network bandwidth than any overhead caused by a hypervisor, which agrees with previous observations by Roloff *et al.* [15]; (ii) the clusters using hypervisor do not exhibit any ill-effect from potential shared resources even though these instances are not rented as isolated ones.

Fig. 5 shows the estimated cost based on on-demand prices for computational power and memory. Again, this is only an estimate as the actual price paid by the end-user will depend on several variables not considered here. The 3 lowest costs correspond to the Cavium and AMD EPYC-1 configurations because of their low prices. However, they have limitations on their peak performance and memory capabilities. The costs for EPYC-2 configurations fluctuate along with their performances and the best performer, δ, is also the most economical option. Intel configurations sit at the high level of cost estimate, particularly those with the best performance. Lastly, the configuration with the new Graviton processor provides a moderate cost with the fifth lowest estimate in the present study.

Measurements of collective MPI communications complement the HPCG benchmarks. The first measurement is an MPI_Alltoall operation between all the ranks. When singlethreaded, this operation completes successfully for all configurations as the results in Fig. 6 testify. For small packets, the lowest latencies are exhibited by single instances, but this latency increases for clusters where these small packets need to be exchanged across instances. For intermediate packet sizes, the latency times exhibit fluctuations for some clusters (e.g., ξ or σ) but, overall, they exhibit growth. The latency times for the largest packets span two orders of magnitude and, more interestingly, the small cluster composed by 2 AWS c5n.18xlarge (π) exhibit a lower latency than the single instances. These measurements show the significance of network bandwidth. Clusters ξ and π facilitate again a direct assessment as the latency times for the smallest packets are around 30 percent higher for the cluster with lower latency; a figure that increases with packet size. The impact of hypervisor presence is again less significant.

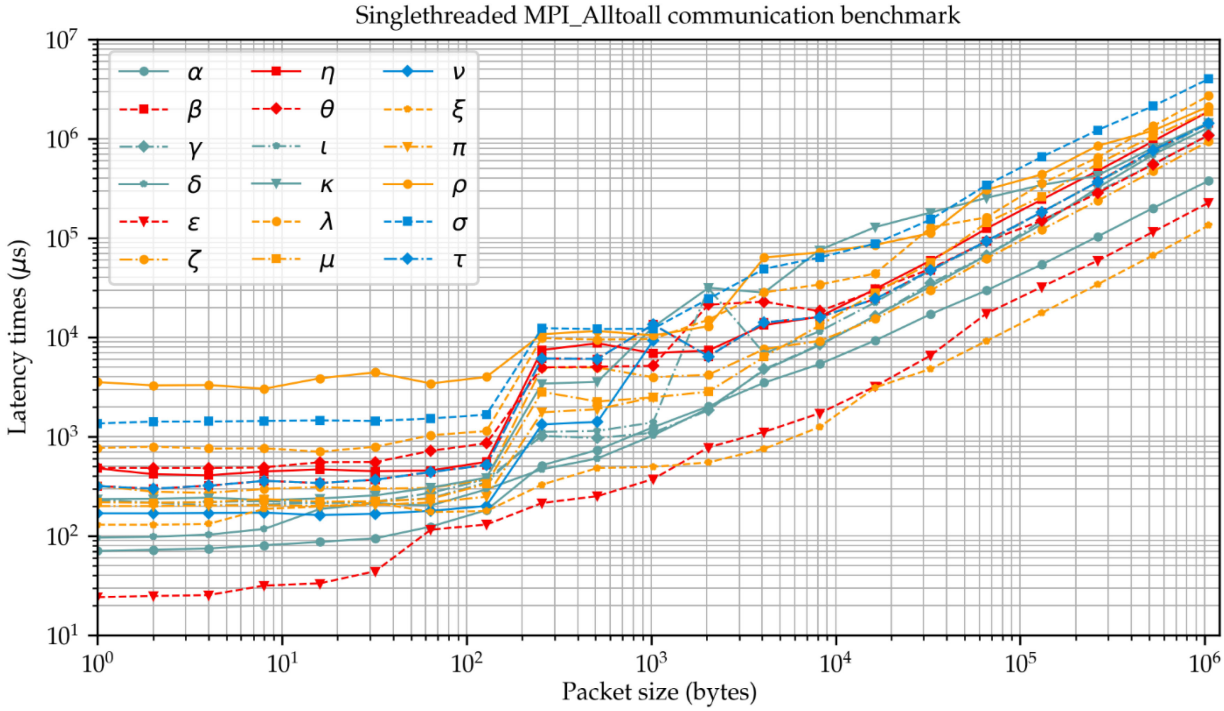

Fig. 6. Latency times from singlethreaded MPI_Alltoall operations for the configurations listed in Table 2.

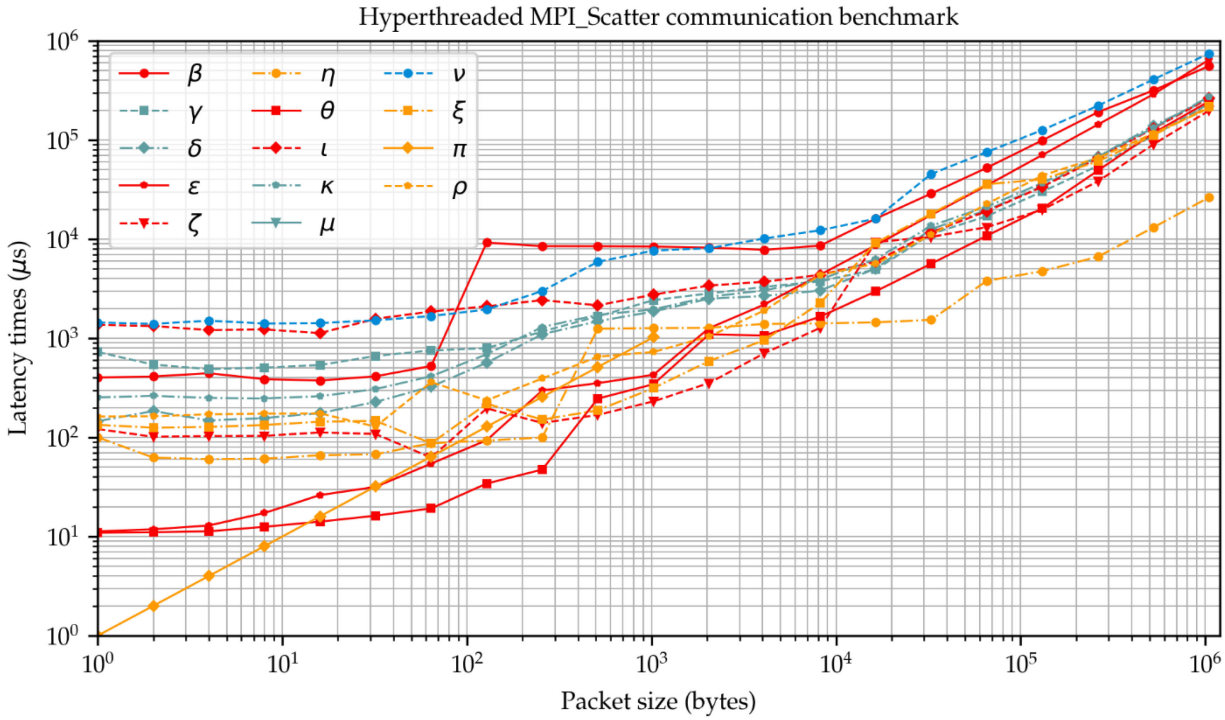

Fig. 7. Latency times from hyperthreaded MPI_Scatter operations for the configurations listed in Table 2.

A direct comparison between clusters ζ and π shows differences within 1 percent. Attempting to repeat the same MPI_Alltoall operation in hyperthreading mode commonly results in network saturation and the operation fails to complete in many of the configurations. Instead, and to quantify MPI communications when hyperthreaded is active, Fig. 7 shows the latencies from an MPI_Scatter operation. The latency diagram from this operation shows a more gradual growth of the latency times for single instances, while clusters exhibit a relatively uniform behavior for small packets, a transitional zone that sometimes exhibits a plateau, and a steady growth for the largest packets when latency times depend mainly on the network bandwidth.

3.3 Evaluation of Cluster Performance

The last set of benchmarks involves increasingly larger clusters with the the core count reaching 4500. The minimum memory of these clusters is 2 TB, which allows them to accommodate large workloads. The results from Section 3.2 facilitate the selection of the instances composing these clusters. The final selection consists of the following instances: m2.xlarge.x86 and c3.medium.x86 from Packet, BM.E2.64 and BM.HPC2.36 from OCI, and c5a.24xlarge, c6g.16xlarge, c5n.18xlarge and z1d.12xlarge from AWS. The network bandwidth ranges from 10 to 100 Gbps. The main purpose of this section is to assess scalability, and to provide a preliminary prediction on performance for clusters composed by hundreds and thousands of cores. To provide a baseline, the reference values are taken from the computations with a single instance. The scalability parameter for each cluster is defined as:

$$Sc = \frac{[HPCG/core]_{cluster}}{[HPCG/core]_{single\ instance}}. \quad (2)$$

Fig. 8 shows Sc as a function of the number of cores. These benchmarks illustrate the significance of network features. The Packet clusters, which have the lowest network bandwidth at 10 Gbps, exhibit a decreasing scalability as more instances are added to the clusters. In this situation, hyperthreaded jobs generally realize less decrease in performance than singlethreaded ones. The OCI clusters confirm this tendency as the network bandwidth of 25 Gbps also results in performance loss. For AWS, the clusters composed by c5a.24xlarge and z1d.12xlarge instances see also a decrease in scalability as the core count grows. This decline is already significant when the number of cores is several hundreds. The scalability for the clusters composed by c6g.16xlarge instances exbibits a more moderate deterioration as, after an initial decline, the scalability parameter settles down to a value close to 90 percent with a substantially more gradual decrease as more instances are added. The last AWS clusters (c5n.18xlarge) highlight the enhancement in scalability when increasing the network bandwidth to 100 Gbps. In this situation, the scalability remains higher than 97 and 98 percent for singlethreaded and hyperthreaded jobs even for clusters with 4500 cores. Furthermore, these instances enjoy not only a 100 Gbps bandwidth, but they use AWS proprietary Elastic Fabric Adapter (EFA) as a network interface. These results illustrate the efficiency of this technology when handling intensive workloads across thousands of cores. Although not shown here graphically, a cost estimate for the clusters is easily derived by combining the data from Figs. 5 and 8.

The large size of the clusters discussed in this section renders some communication operations such as MPI_Alltoall particularly challenging or even unfeasible. As the variety of MPI operations and range of cluster size is wide, and to provide an example of communication patterns, Fig. 8 shows the latencies from the MPI_Scatter operation for clusters

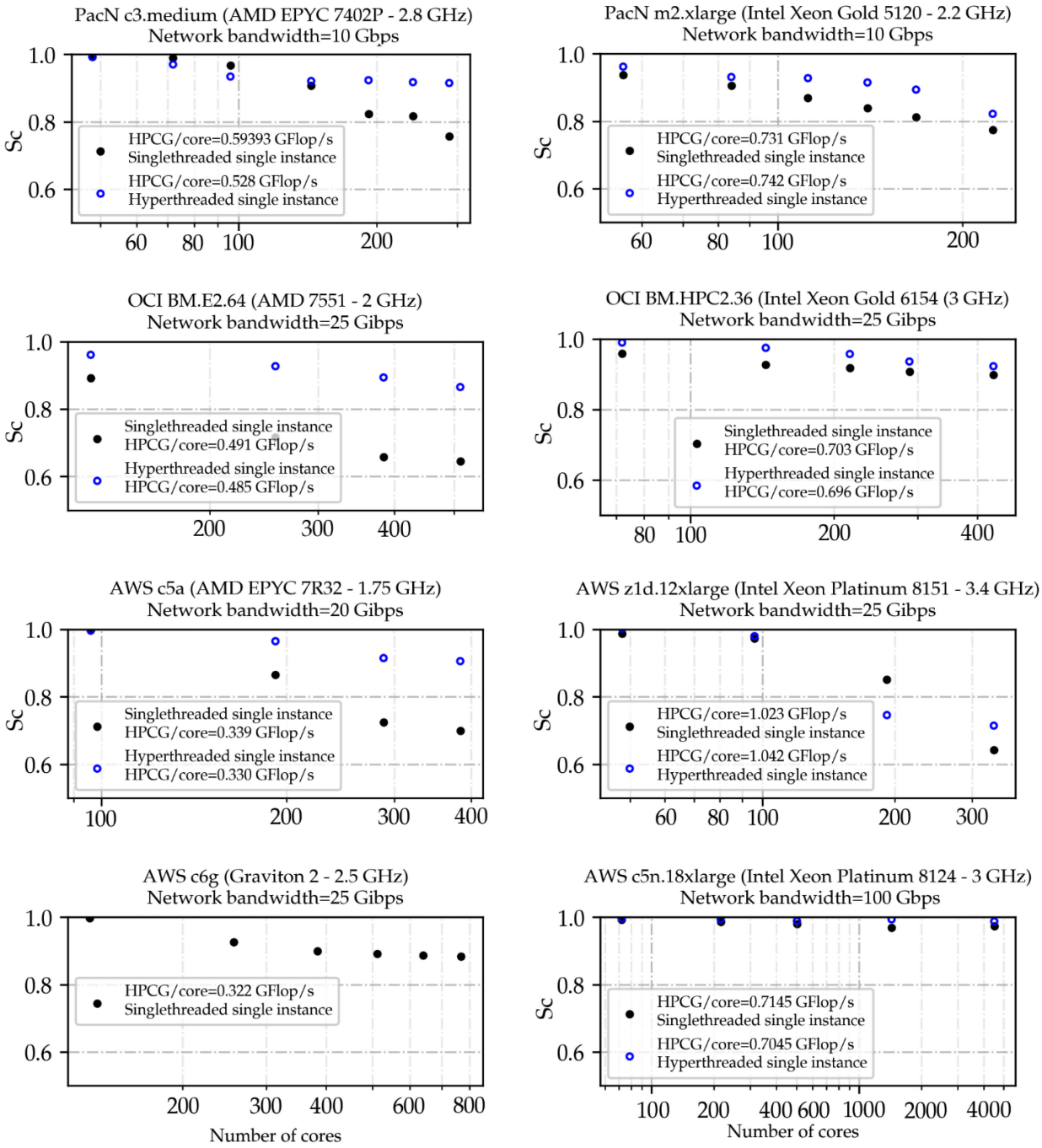

Fig. 8. Scalability for clusters composed by IaaS from the public cloud as a function of the number of cores.

with up to 512 cores. The latency diagram exhibits again 3 zones depending on the packet size. A feature for several os these clusters is that the transitional zone exhibits a plateau with the latency times being almost constant for packet sizes between 0.5 and 30 kbytes. This effect also occurs for β in Section 3.2, but it becomes bolder as the size of the clusters increases. The measurements of MPI_Scatter also reflect an increase of up to 400 percent in latencies when switching from singlethreaded to hyperthreaded mode.

4 Analysis

The results from the previous subsections, particularly those for clusters, facilitate a more generic analysis from which decision makers might be able to gain some insight. The HPCG benchmarks show that the performance and cost of clusters depend on several factors including architecture, processor family, and the price per instance set by CSPs. To facilitate an analysis of performance, it is useful to evaluate how these results compare versus on-premises hardware, especially versus supercomputers currently performing some of the most advanced scientific and engineering simulations. To do so, it is feasible to use the official HPCG list [25] a reference. As of November of 2020, this list includes 73 supercomputers that can be trimmed down by ruling out those with accelerators and vector processors. These 45 supercomputers, built between 2013 and 2020, have a core number spanning from 31524 to over 10 million. Fig. 10

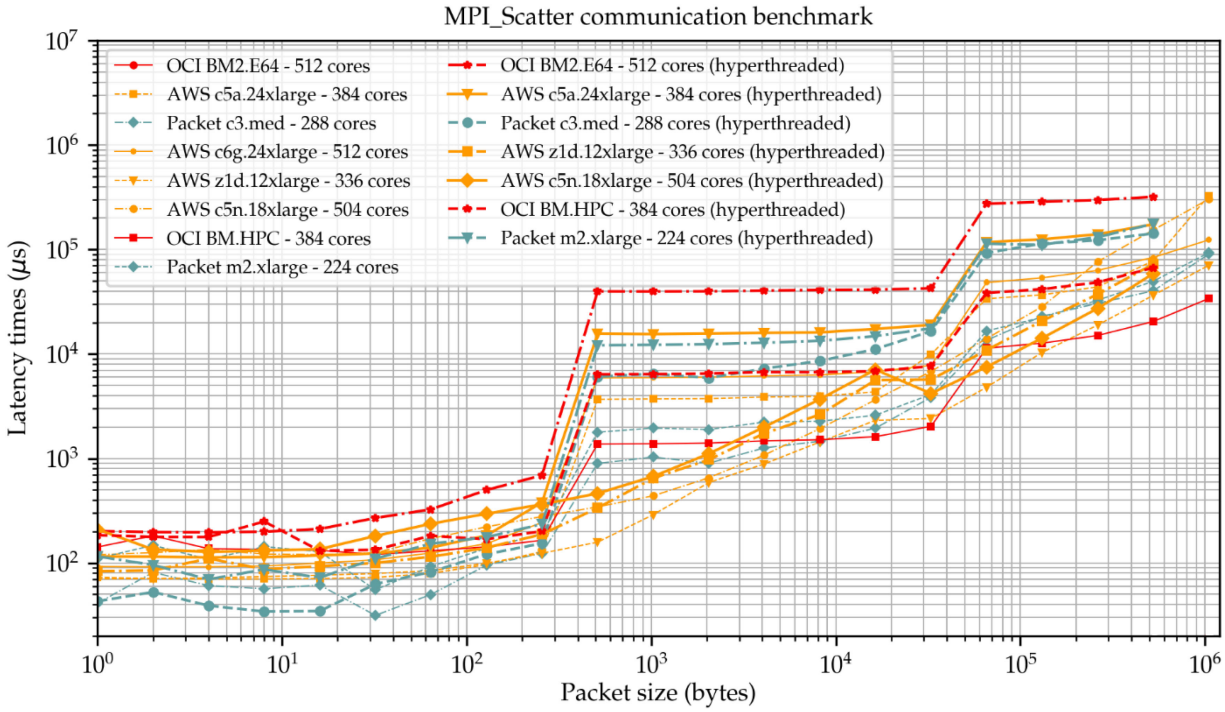

Fig. 9. Latency times from MPI_Scatter operations for clusters composed by IaaS from the public cloud.

shows a histogram with their HPCG figure of merit. Out of these 45 supercomputers, 4 of them use ARM architecture, 39 use x86_64, and the other 2 are special processors (Sparc & Sunway). A significant change over the last year is that the 3 supercomputers with a performance higher than 2 [(GFLOP/s)/core] are powered with the new Fujitsu A64FX processor. In addition to its high memory bandwidth, this processor is the only available one that incorporates the SVE extension to the Aarch64 ISA standard. The other Arm supercomputer uses Marvell ThundexX2 and measures at 0.63 [(GFLOP/s)/core]. The performance of ARM IaaS has traditionally lagged similar x86_64 configurations. However, its newer Neoverse N1 generation, represented here by AWS-Graviton 2, has seen a significant increase in performance. Major efforts are currently undergoing in the development of new ARM processors for datacenters, and specifically for cloud environments. Three keys that will impact the future success of ARM processors related to HPC performance are the growth in memory bandwidth for many core processors, the ability to increase the rate of double-precision operations per cycle, and the inclusion of SVE in all future Neoverse V1 and N2 processors. Cluster performance based on x86_64 architecture varies depending on several factors. Clusters using AMD processors exhibit a moderate HPCG figure of merit, but the results for this metric are in the same range as the EPYC-2 supercomputer currently listed in the HPCG website. The two best performers of the present study come from AWS using Intel processors. The figure of merit for small clusters with AWS-z1d.12xlarge instances would fit at the high end of the supercomputers powered by Intel, but scalability might hamper performance for large clusters. A more relevant comparison is with clusters composed by AWS-c5n.18xlarge instances, which have exhibited excellent scalability and their figure of merit would fall in the 64th percentile of the HPCG list.

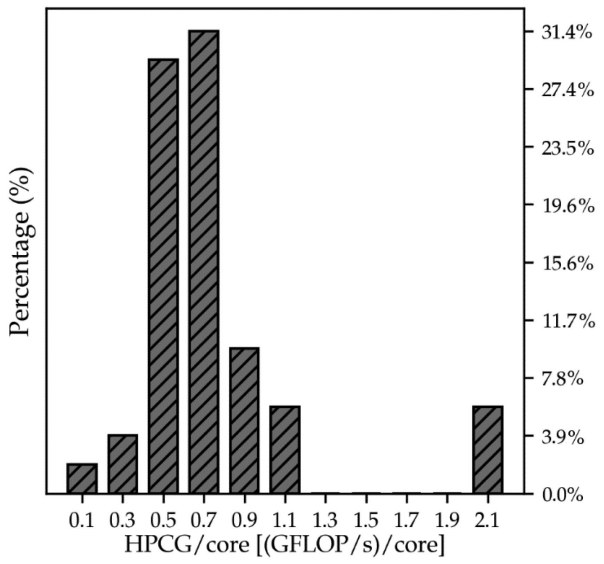

Fig. 10. Distribution of the HPCG figure of merit for the computers listed at www.top500.org/lists/hpcg/.

A different subject is an economical analysis where drawing general conclusions is more difficult as cost depends on several factors not considered here. The cost estimates presented here have shown some variability. For example, the best performers also sit at the high end of the cost estimate even after normalizing with performance. A potential compromise between cost and performance seems to be the use of instances powered by ARM processors (e.g., AWS c6g.16xlarge) as their cost and performance per core are expected continue decreasing and increasing, respectively, in the next generations. Ultimately, decision makers seeking to migrate HPC workloads to public IaaS will have to consider all these factors and find a compromise between performance and cost.

5 Conclusion

The present work has examined the performance of IaaS from several cloud vendors using the HPCG and OMB benchmarks. The focus has been on computational power derived from traditional processors as the evaluation of IaaS with accelerators is the subject of a separate work. Although the final evaluation of any app ultimately requires its own benchmarks, the present results should serve as a yardstick for organizations interested in the public cloud. The current IaaS offering is already able to replace small and medium on-premises clusters while providing a lot more flexibility in the number of cores, fluctuating workloads, storage solutions and other strengths generally associated with the public cloud. One of the main concerns in the cloud has been the ability to efficiently run on thousands of cores. The benchmarks presented here sit at the lower end of this limit, but they show IaaS to be able to handle this type of workloads.

References

[1] J. J. Dongarra, S. W. Otto, M. Snir, and D. Walker, "An introduction to the MPI standard," Univ. Tennessee, Knoxville TN, USA, Tech. Rep. CS-95-274, Jan. 1995.

[2] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming With the Message-Passing Interface*. Cambridge, MA, USA: MIT Press, 1999.

[3] J. J. Dongarra and M. A. Heroux, "Toward a new metric for ranking high performance computing systems," Sandia Nat. Lab. Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. SAND2013-4744, 2013.

[4] Accessed: Jan. 28, 2021. [Online]. Available: http://mvapich.cse.ohio-state.edu/benchmarks.

[5] Accessed: Jan. 28, 2021. [Online]. Available: https://aws.amazon.com

[6] Accessed: Jan. 28, 2021. [Online]. Available: https://azure.microsoft.com/en-us

[7] Accessed: Jan. 28, 2021. [Online]. Available: https://cloud.google.com

[8] Accessed: Jan. 28, 2021. [Online]. Available: https://www.oracle.com/cloud/

[9] Accessed: Jan. 28, 2021. [Online]. Available: https://www.packet.com

[10] J. Dongarra, P. Luszczek, and A. Petitet, "The LINPACK benchmark: Past, present and future," *Concurrency Comput., Pract. Experience*, vol. 15 no. 9, pp. 803–820, 2003.

[11] Accessed: Jan. 28, 2021. [Online]. Available: www.top500.org

[12] D. Bailey, J. Barton, T. Lasinski, and H. Simon, "The NAS parallel benchmarks," NASA Ames Res. Center Tech. Rep. RNR-91-002, 1991.

[13] J. Dongarra and P. Luszczek, "Introduction to the HPC challenge benchmark suite," Innovative Comput. Lab., Univ. Tennessee, Tech. Rep.ICL-UT-05-01, 2005.

[14] K. Yelick, S. Coghlan, B. Draney, and R. S. Canon, "The magellan report on cloud computing for science," Lawrence Berkeley Nat. Lab. (LBNL), U.S. Dept. Energy, Berkeley, CA, USA, Tech. rep. LBNL-5376E, 2011.

[15] E. Roloff, M. Diener, A. Carissimi, and P. O. A. Navaux, "High performance computing in the cloud: Deployment, performance and cost efficiency," in *Proc. 4th IEEE Int. Conf. Cloud Comput. Technol. Sci. Proc.*, 2012, pp. 371–378.

[16] V. Mauch, M. Kunze, and M. Hillenbrand, "High performance cloud computing. *Future Gener. Comput. Syst.*, vol. 29, pp. 1408–1416, 2013.

[17] A. Gupta *et al.*, "Evaluating and improving the performance and scheduling of HPC applications in cloud," *IEEE Trans. Cloud Comput.*, vol. 4, no. 3, pp. 307–321, Third Quarter 2014.

[18] I. Sadooghi *et al.*, "Understanding the performance and potential of cloud computing for scientific applications," *IEEE Trans. Cloud Comput.*, vol. 5, no. 2. pp. 358–371, Second Quarter 2015.

[19] M. Mohammadi and T. Bazhirov, "Comparative benchmarking of cloud computing vendors with high performance linpack," in *Proc. 2 nd Int. Conf. High Perform. Compilation Comput. Commun.*, 2018, pp. 1–5.

[20] C. Kotas. T. Naughton and N. Imam, "A comparison of amazon web services and micorsoft azure cloud services for high performance computing," in *Proc. IEEE Int. Conf. Consum. Electron.*, 2018, pp. 1–4.

[21] S. Chang *et al.*, "Evaluating the suitability of commercial clouds for NASA's high performance computing applications: A trade study," NASA Adv. Supercomputing (NAS) Division Tech. Rep. NAS-2018-01, 2018.

[22] M. A. S. Netto, R. N. Calheiros, E. R. Rodrigues, R. L. F. Cunha, and R. Buyya, "HPC cloud for scientific and business applications: Taxonomy, vision, and research challenges," *ACM Comput. Surveys*, vol. 51, no. 1, 2018, Art. no. 8.

[23] P. Mattson *et al.*, "MLPerf training benchmark," 2019, *arXiv:1910.01500*.

[24] [Online]. Available: https://mlperf.org/training-results-0-7

[25] Accessed: Jan. 28, 2021. [Online]. Available: https://www.top500.org/lists/hpcg/

Arturo Fernandez received the BS and PhD degrees from the Polytechnic University of Madrid. He was a postdoctorate researcher at WPI working in large-scale modeling of multiphase flow. He has consulted for EDA, worked for the Hall Process Improvement Department of Alcoa, and currently leads the Cloud Department at Odyhpc.com. He has lectured at WPI, CUA, and NCAT. His current research interests include the solution of PDEs commonly found in engineering and, more particularly, in computational fluid dynamics. His research projects have encompassed a variety of interdisciplinary topics including the implementation of HPC to perform direct numerical simulation of complex fluid flow, numerical solutions of the Navier-Stokes and Maxwell equations, modeling of electrolytic cells, and LES of turbulent flow with focus on marine renewable energy.