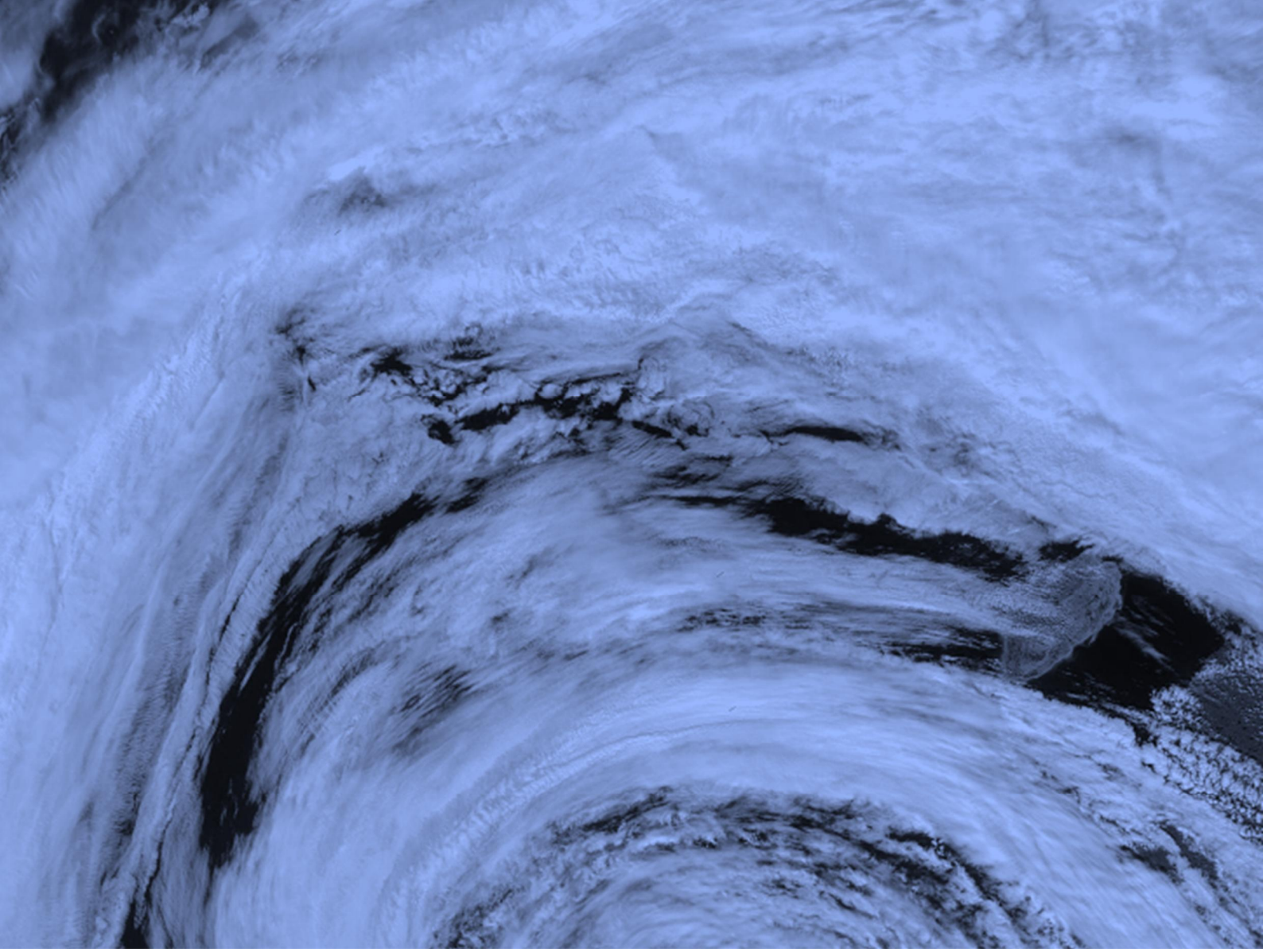
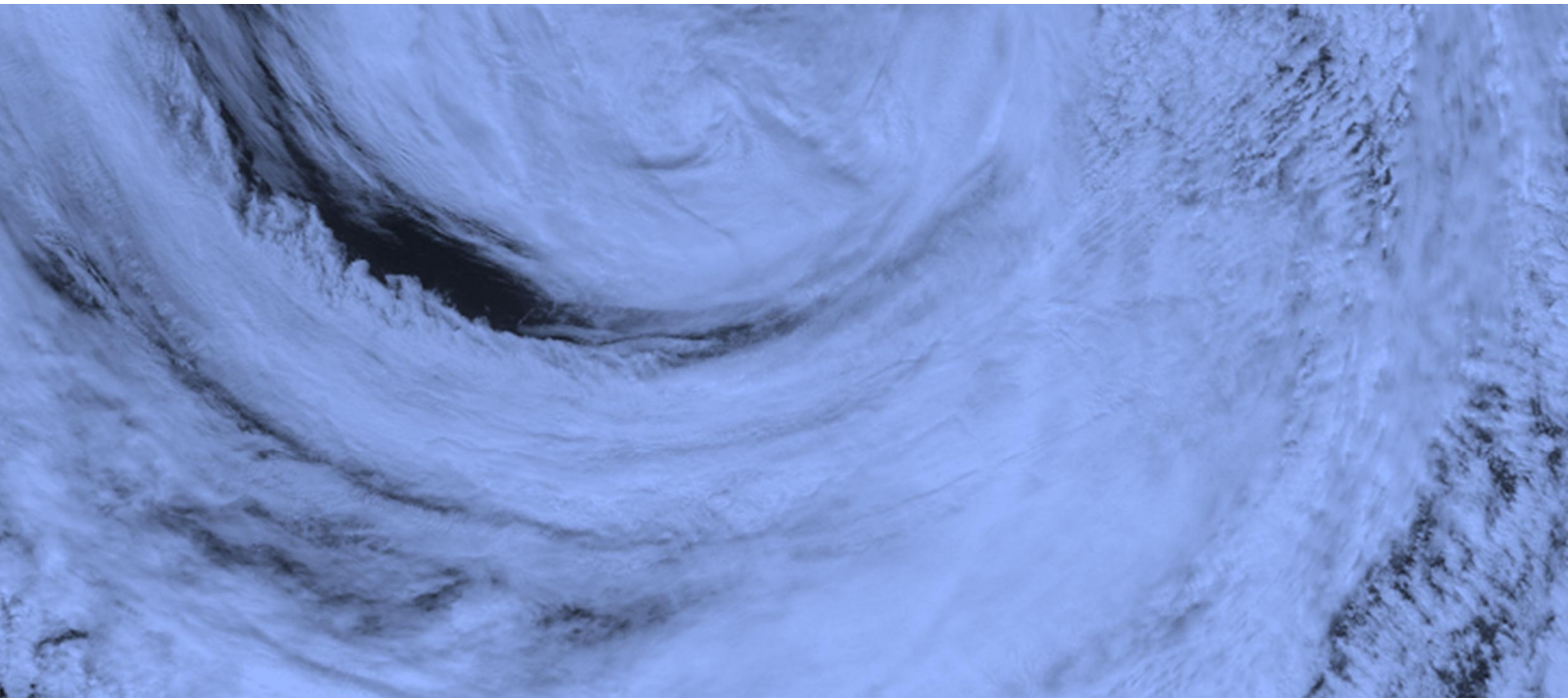


High-performance computing in the cloud?



High-performance computing in the cloud?

Table of Contents

Cover image: NASA Worldview

Introduction

In the last few years cloud computing has become a popular model for offering compute resources in the commercial sector. Infrastructure as a Service (IaaS) clouds enable users to easily scale the compute capacity and enable them to run custom software stacks. These characteristics have also raised interest in the field of scientific computing.

This report is a quantitative assessment of the competitiveness of commercial IaaS clouds and CSC's computing resources, which comprise bare-metal supercomputers, clusters and cloud resources. The comparison is done in terms of price-performance based on specifications, as well as actual benchmarks for various workloads. We show here that the large commercial cloud providers are not, for the majority of scientific workloads, competitive in terms of price nor application performance. For a subset of scientific workloads they work well, and hence CSC is also actively developing a private high performance cloud offering.

Compute capacity for scientific computing

Traditionally high-performance computing (HPC) resources comprise bare-metal supercomputers, and clusters, where computational work (jobs) are queued and executed when there are sufficient resources on the machine. All applications and libraries need to be compiled for the operating system that is installed on the machine

In IaaS clouds capacity is offered using virtualization technology. On each node a hypervisor runs multiple virtual machines (VMs, or "instances") on virtual operating platforms. On these VMs one can install Linux or Windows, and a complete custom stack of software. A VM can efficiently utilize the central processing unit (CPU) and main memory, but accessing external devices such as disks, graphics processing units (GPUs) and network interfaces may incur significant overhead since the hypervisor translates the accesses in software. Recently a number of hardware based approaches have been developed to reduce overhead: Single Root I/O Virtualization (SR-IOV) that supports access from multiple guest VMs, and PCI passthrough that supports one VM per node.

In scientific computing even small overhead may be unacceptable, and so-called software containers have become an option to virtualization. A container uses Linux cgroups and kernel namespaces to enable containers to run a fully custom stack of software including system tools, system libraries, runtimes, and code on top of the Linux host kernel. This enables packaging an application into a container that will run on any Linux system that supports containers. The overhead from containers is small because they run natively on top of the Linux kernel.

Commercial cloud providers

In commercial IaaS clouds the focus has been on hardware that is useful for running standard business and web applications. These are a match also with scientific workflows where the individual instances do not need to be tightly coupled with a high-performance interconnect. Typically, the cloud platforms offer different flavours of nodes, optimizing for compute performance, I/O performance, or the amount of memory. The most common interconnect is 10 GB Ethernet. The main commercial cloud providers are[1]: **Amazon web services (AWS) EC2**, **Microsoft Azure**, **IBM's Spectrum Computing, Google Cloud Platform** and **Fujitsu Cloud Service K5.**

There are also cloud platforms and "computing as a service" providers that are focused on HPC and performance. The typical hallmark of these is dedicated infrastructure which comprises fast nodes and Infiniband interconnects supporting RDMA. Typically these are not traditional clouds but rather bare-metal clusters around which ease-of-user services such as pre-installed applications and consulting services have been developed. In this regard they are similar to traditional supercomputing centers for science. A few companies in this category are **Bull extreme factory**, **Nimbix**, **Penguin Computing On Demand** (POD) and **Sabalcore.**

[1] https://www.srgresearch.com/articles/aws-remains-dominant-despite-microsoft-and-google-growth-surges

Additionally some companies provide a software product with which external clouds such as EC2 can be used more easily or more efficiently. Examples from this category are: **CycleCloud** and **Alces Flight**.

CSC - IT Center for Science

CSC supports researchers and academic institutions by providing a comprehensive set of computing services. CSC's computing platforms are located in Kajaani, Finland, in one of the most energy-efficient datacenters in the world.

Sisu supercomputer is a Cray XC40 system with over 40,000 cores, supporting large-scale parallel computation which requires exceptionally powerful supercomputing resources, from several hundreds to up to thousands of cores. It features a collection of preinstalled applications and development tools which support massive parallelism.

Taito cluster is aimed for general-purpose technical and scientific computing, from single-core (sequential, or serial) jobs to small parallel jobs of a few hundred cores. It features a large collection of preinstalled applications and development tools, special large-memory nodes for tasks requiring a large memory footprint, and specialized compute nodes with NVidia Tesla GPUs.

cPouta and ePouta are CSC's cloud computing platforms that offer high performance computing platforms via the IaaS model. ePouta is a secure private cloud for sensitive data, such as genomic data.

Specifications and pricing

Cloud pricing models

Cloud providers typically provide a pricing scheme where the price per CPU hour or node hour is dependent on multiple factors: CPU performance, memory size, disk performance, are the resources reserved for a fixed-term, and are the resources reliable or can the instances be terminated if other higher paying customers require more resources. For example for AWS EC2 there are essentially three pricing models[2] for compute resources

1. **On-demand:** No fixed costs, only pay for dedicated use of node. Most expensive option per hour.
2. **Reserved instances:** Nodes are reserved for the customer, and the customer pays for the capacity even if they are not used.
3. **Spot Instances:** Market driven pricing, where instances are terminated when price goes above the bid set by the customer.

Furthermore cloud providers charge for data storage and transfer. As datasets grow and become more complicated the role of the storage subsystem becomes increasingly important. A single storage model is not enough for complicated data storage and analysis requirements and therefore service providers have different solutions with different performance characteristics. We can identify three main categories of storage that are:

1. Persistent local storage.
2. Object or database storage.
3. Cold storage that is accessed rarely.

Data transfer cost is dependent on the type of the access and also on the geographical region. Variety of different storage solutions gives users flexibility and also an opportunity to optimize the cost of data storage and access. On the other hand, users must have predictable needs in order to be able to fully optimize the costs. In most cases transferring data between different storage models or services incurs extra cost.

CSC - IT Center for Science pricing model

The pricing for CSC's resources[3] used here for comparison is the published prices for academic customers. The price is calculated based on the actual costs incurred by the operations and management of the systems, their amortization and infrastructure costs (datacenter, networking, storage etc.), costs for service development and a small risk buffer.

[2] https://aws.amazon.com/ec2/pricing/

[3] https://research.csc.fi/pricing-of-computing-services

Price-performance comparison

For comparing the price competitiveness of cloud resources in term of compute performance we have used instances that provide the closest possible match to CSCs regular compute nodes in Sisu supercomputer and Taito cluster. We normalize the comparison to correspond to a node with a theoretical performance of 1 TFlops. The prices are converted to Euros using an exchange rate of 1 Euro = 1.11 US Dollar (May 31, 2016). The prices offered by the cloud vendors are typically not stable, but vary over time. For AWS EC2 spot prices we use the median value for May 15th - June 30th, while the Google cloud platform pricing for preemptible instances represents a typical price for that time period. Other prices have been collected in May 2016.

Table 1. Pricing for compute intensive nodes

System	Pricing model	€ per normalized node hour	€ per GB hour
CSC - Sisu		0.529[4]	0.0083
AWS EC2 - c4.8xlarge	spot	0.493	0.0055
	on-demand	2.29[5]	0.027
	reserved	1.04	0.013
Google Cloud Platform - n1-highcpu-32	spot	0.543	0.010
	on-demand	1.861	0.038
	reserved	1.297	0.0265
Nimbix	on-demand	6.551	0.0170
	Subscription	4.330	0.0113
Sabalcore	Academic price	1.516	0.0118

On AWS EC2, even the most cost-effective compute intensive instances, where the instance is reserved for 3 years and paid upfront, are twice as expensive as CSC servers (Table 1). Spot prices on AWS and preemptable instances on Google cloud are marginally cheaper (10%) than the node pricing at CSC, but the spot instances have a much lower reliability than CSC resources. To utilize these the workflow has to be able to run the jobs on unreliable instances which can be terminated at any time. This further increases the complexity of using the resources, and decreases the fraction of workflows that can be adapted to it. An example of such workflows is globus genomics[6]. The HPC centric cloud providers (Nimbi, Sabalcore) are more than three times as expensive.

[4] https://research.csc.fi/pricing-of-computing-services

[5] https://aws.amazon.com/ec2/pricing/

[6] https://www.globus.org/genomics

In data intensive computing the amount of memory can oftentimes become a limiting factor. In the table below we compare the cost with the amount of memory used, and normalize the price by reporting the cost per GiB per hour. While the exact capacity differs, we can identify 3 fairly comparable scenarios:

1. **Normal**: 60 - 128 GB / node
2. **Large**: 160 - 256 GB / node
3. **Huge**: 1500 - 2000 GB / node

The normal case is covered by the previous section on compute intensive nodes. In Table 2 the data intensive large memory nodes are detailed.

Table 2. Pricing for data intensive nodes

System	Memory size	Pricing model	€ per normalized node hour	€ per GB hour
Taito	L		0.529	0.00206
Taito	H		0.491	0.00059
AWS - r3.8xlarge	L	spot	1.404	0.00166
		on-demand	8.34	0.0109
		reserved	3.41	0.0044
AWS - x1.32xlarge	H	spot	0.680	0.00074
		on-demand	6.123	0.0074
		reserved	1.712	0.0021

When focusing on the price per GB the story is similar, Spot prices on AWS EC2 are competitive, with 20% lower price for large memory sizes and 25% higher prices per GB for huge memory sizes. The more typical on-demand and reserved instances are already significantly more expensive. For example the on-demand price per GB for large memory size instances is 430% higher, and for huge memory size instances 1154% higher.

Benchmarks

For the benchmarking we set up a small test cluster to AWS EC2 using a modified version of Elasticluster[7]. We added support to Amazon Linux 16.03 and modified the provisioning so that all instances are added to same placement group in order to get the best network performance. We chose Amazon Linux because it supports the Single Root I/O Virtualization (SR-IOV) without any additional kernel configurations. Additionally we also manually set up a cluster by installing openMPI on Amazon Linux 16.03 instances (AWS in the figures).

Message-passing performance

Message Passing Interface (MPI) is the de facto standard parallelization strategy in scientific computing, and nearly all parallel scientific applications have implemented their intra-node communication using it. We evaluated the performance of the MPI communication on different platforms using the Intel MPI Benchmarks suite[8] version 4.1.

First, we measured the network characteristics dictating the overall MPI performance: ping-pong latency and the message bandwidth. The measurements in the AWS EC2 on c4.x8large instances are compared to those carried out in a Cray XC40 supercomputer (Sisu) at CSC. Furthermore, as the overall performance of many parallel applications depend on the performance of MPI collective operations (rather than raw bandwidth), we benchmarked also the MPI_Alltoall routine, which is the bottleneck communication operation in, among others, scientific algorithms featuring spectral methods (e.g. Fast Fourier Transforms).

The Figure 1 illustrates the MPI ping-pong latency (smaller latency equals to better communication performance) as a function of the message size, measured between two MPI tasks placed on different nodes. Note the logarithmic scale on both axis. Cray's Aries network appears to be at least 100 times lower latency than EC2 or cPouta with all message sizes, even more pronouncedly so with small messages where latencies define the observed performance. Already based on the latencies it looks quite impossible to efficiently run typical MPI applications across multiple nodes on any of the available cloud computing platforms featuring an Ethernet interconnect.

Next, we measured the point-to-point communication bandwidth (higher is better) available for two MPI processes located on different nodes, presented in Figure 2. Note again the logarithmic scales. The interconnect of the Cray XC is able to provide 20,000 MB/s bandwidth with certain message sizes, whereas the highest bandwidth of the AWS EC2 or Alces Flight is 430 MB/s. cPouta's 40 Gb Ethernet gives higher bandwidth for large messages, reaching almost 1000 MB/s at the maximum.

[7] http://gc3-uzh-ch.github.io/elasticluster/

[8] https://software.intel.com/en-us/articles/intel-mpi-benchmarks

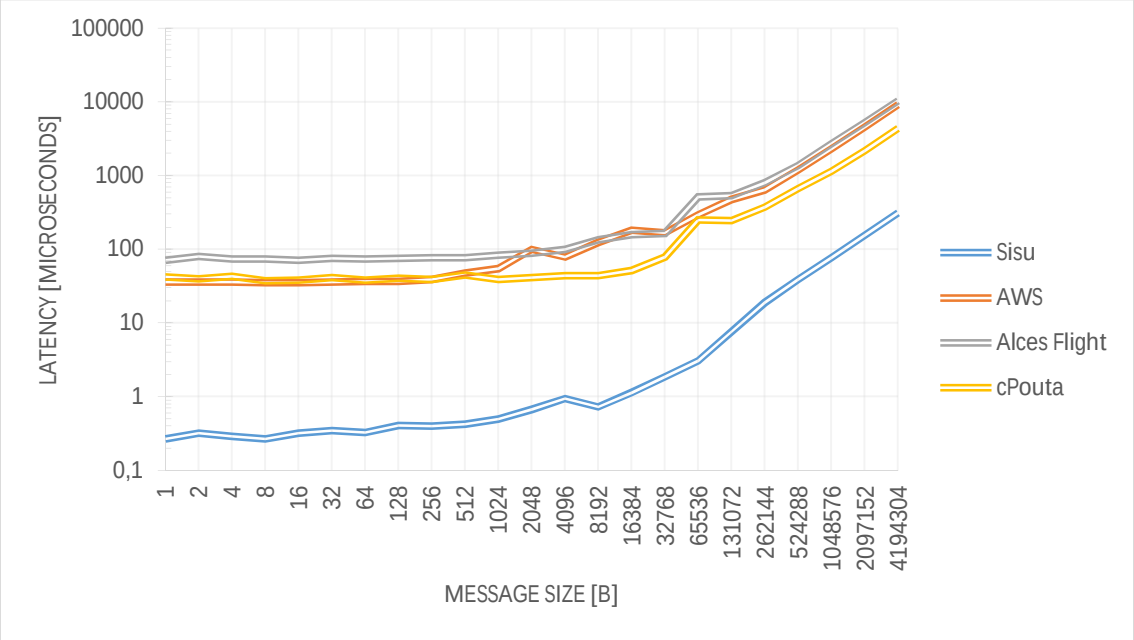


Figure 1. MPI ping-pong latency as a function of the message size.

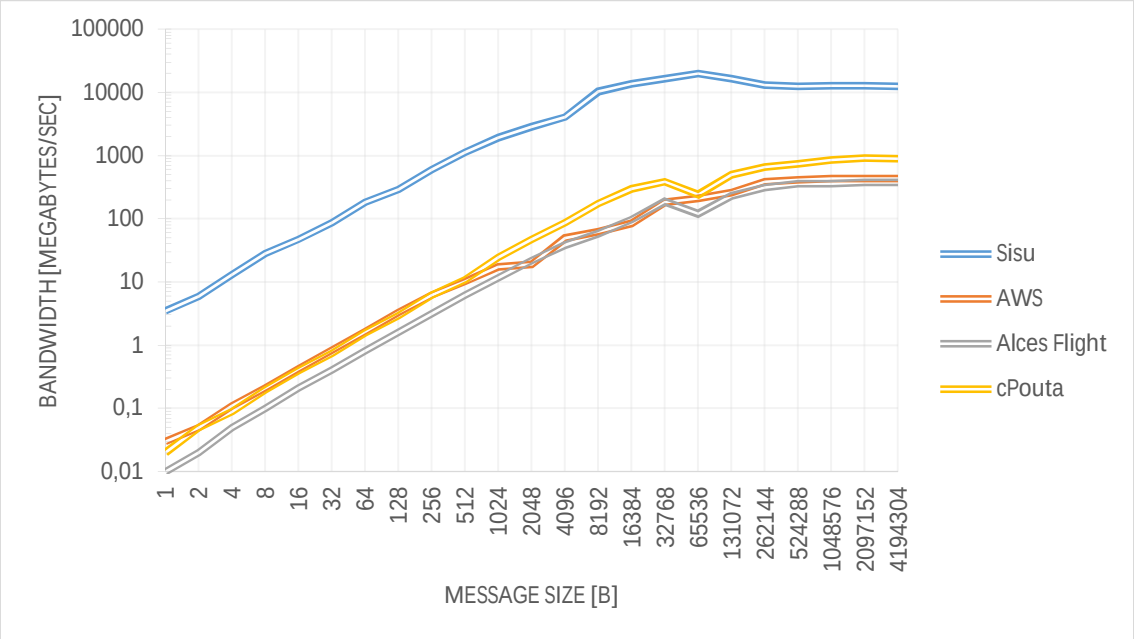


Figure 2. MPI ping-pong bandwidth as a function of the message size.

While the ping-pong latency and bandwidth measure more the capability of the interconnect than the MPI library (implementation) itself, MPI collective operations are heavily dependent on the implementation - a collective can be slow even on a fast interconnect if the implementation does not fully utilize the network capabilities. The measurements of the average throughput time of MPI_Alltoall as a function of message (size of the data sent to all other tasks, by each task) are presented in Figure 3.

These were measured with 144 MPI tasks on 8 nodes. The MPI_Alltoall operation seems to be systematically some 10 times faster on the Cray XC40 than on the cloud platforms for messages up to 128 byte in size. With larger messages than that, something on the EC2 and cPouta both makes the difference to bounce up. With 1024 byte and larger messages the operation becomes so slow on EC2 that the run aborts because of a network timeout. Based on these measurements, it appears impossible to execute efficiently MPI applications dependent on the MPI_Alltoall operation on the cloud platforms.

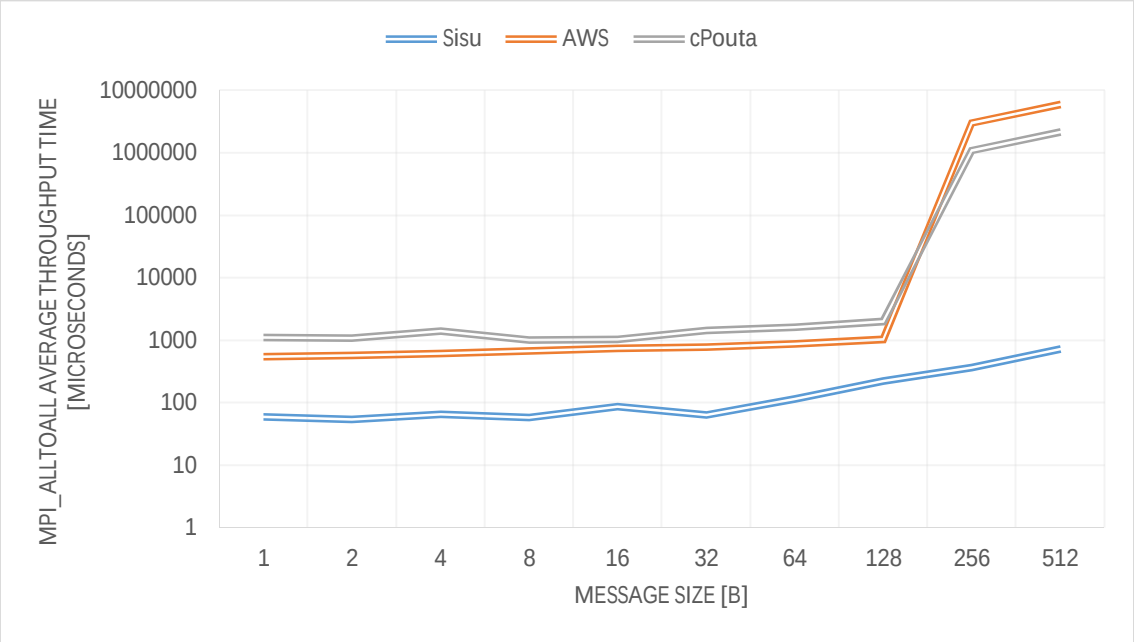


Figure 3. Performance of MPI_Alltoall as a function of the message size.

Storage performance

The storage performance was tested using the FIO benchmark with the sequential and random reads and write rates (operations/s) presented in Table 3 and the read/write bandwidths presented in Table 4. Sequential operations used 4MB block size and random operations used 4kB block size.

Table 3. Results of the FIO benchmark.

Storage	Read IOPS	Write IOPS
Pouta io.*	148261	135698
AWS io1	18153	17175
AWS i2.8xlarge	222182	184422

AWS's storage-intensive instances give the best IOPS performance but are also fairly expensive. The I/O intensive volumes (io1) in Amazon EBS service give a considerably poorer performance and are capped at 20000 IOPS. Overall the performance is quite good in all cases compared to regular spinning disk instances which typically range from 100s to few 1000s of IOPS.

Table 4. High throughput (high bandwidth) storage option benchmarks (single node) [9]

Storage	Read BW (MiB/s)	Write BW (MiB/s)
CSC Lustre file system	1794	1790
Sisu DataWarp	7530	1637
Pouta std	1353	498
AWS st1	538	541

The bandwidth was measured using different filesystems which are optimized for bandwidth intensive operations. In these cases the shared Lustre filesystem is notably faster than the cloud options. The specialized DataWarp nodes in Sisu provide extremely high bandwidth especially for read operations, thanks in part to the very high-speed Aries interconnect network.

It should be noted that in the tests a single server was reading and writing data. With parallel computations oftentimes multiple nodes are used for file operations. In these cases at least DataWarp and Lustre can scale to >10GiB/s.

Application performance: Gromacs

In addition to the synthetic MPI benchmark we also ran molecular dynamics code Gromacs[10] as a representative test case of an application that requires a good interconnect. Gromacs is one of the most used codes at CSC, and on HPC systems worldwide. In 2015, 78 million core hours were used on running Gromacs on CSC computing platforms, equaling to 1.7 M€ as the total value of the computing time.

The test case was taken from the Unified European Applications Benchmark Suite (UEABS)[11] (test case A), and tested using Gromacs version 5.1.1. Figure 4 shows the scientific throughput of Gromacs (ns/day, i.e. how much the simulation progresses in time in a given wall-clock time), higher number equaling better performance. On Sisu, one is able to achieve some 180 ns/day performance with 16 compute nodes. On the cloud platforms, no speedup is obtained by adding more nodes after one full node, and one is inherently limited to 20 ns/day scale, no matter how many nodes are being employed. This can be attributed to the poor MPI_Alltoall performance discussed earlier. The lack of parallel scalability is a serious issue, since typically one needs good

[9] The parameters used for fio were: --ioengine=libaio --iodepth=16 --direct=1 --size=80G --numjobs=8 --runtime=240 --group_reporting

[10] http://www.gromacs.org/

[11] http://www.prace-ri.eu/ueabs/

enough performance to be able to simulate the system for long enough within a reasonable amount of time.

Gromacs features also a GPU (CUDA) implementation of the main computational kernel. We compared the Gromacs performance on the K80 GPU nodes in CSC's Infiniband cluster Taito (Taito GPU on the figure) with the AWS GPU instances (AWS G2). The observed ns/day rates are presented in the same Figure 4 as the CPU results discussed earlier. Gromacs uses single-precision floating point operations with GPUs so the case is well-suited for the GPUs available on AWS. Any codes requiring double precision are infeasible to run on the AWS GPUs, due to the very low double precision peak performance. Again, issues related to the insufficient interconnect hit the AWS GPU results: running on two nodes is two times slower than running on a single GPU instance. On an Infiniband-based GPU cluster, the GPU version is able to harness the computing power of a few GPU nodes.

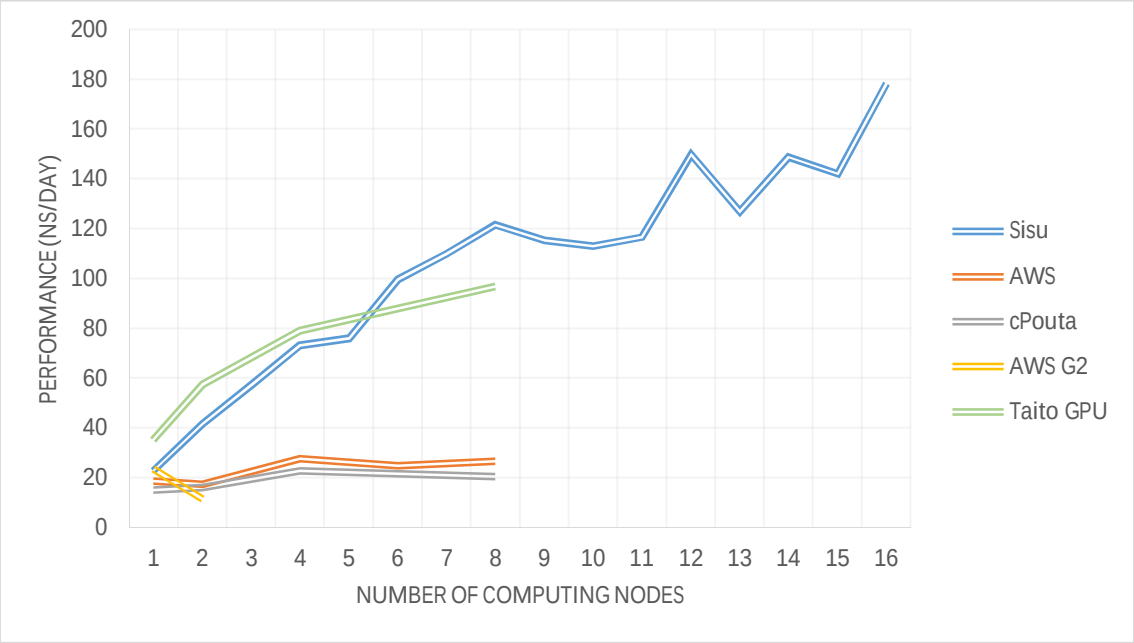


Figure 4. Performance Gromacs 5.1.1 as a function of the number of compute nodes.

Application performance: Biobench2

In addition to the benchmarks that stress the communication layer we also ran the Biobench2 benchmark suite[12]. This suite includes a selection of widely-used bioinformatics applications together with input datasets. We excluded BLAST from the suite because of the size of the reference dataset (40GB). The applications are listed in Table 5. Most of these applications require good I/O bandwidth and they also do a very high number of I/O operations.

Amazon EC2 results were computed using c3.2xlarge instances with provisioned 3600 IOPS volume of size 120GB. Comparison results on Taito were computed using Taito cluster nodes that have a raid array of local disks and Sandy Bridge CPUs. Pouta tests were run on the Pouta I/O

[12] https://wiki.hpcc.msu.edu/display/Bioinfo/Bioinformatics+Benchmarking

instances. Comparison of results is given in Table 5. Here the EC2 is faster than Taito nodes in most tests by 10-30%, with an exception in HMMER. This is due to the slightly faster and newer CPU (2.8GHz Ivy Bridge vs 2.7GHz Sandy Bridge) and newer and faster SSD storage. The Pouta nodes with their high performance SSDs are in most cases (ClustalW2, HMMER, QuEST, Velvet) faster than AWS EC2. In this kind of application load the performance is determined by the node (CPU, memory, SSD) performance and the cloud instances provide an appealing yet less cost-efficient alternative.

Table 5. List of bioinformatics applications that were used for benchmarking, and performance on EC2, Taito and cPouta

			Execution time (s)		
Application	Version	Description	**EC2**	**Taito**	**cPouta**
BEDTools	2.12.0	A flexible suite of utilities for for comparing genomic features.	5.1	5.6	6.1
ClustalW2	2.1	General purpose multiple sequence alignment program for DNA of proteins.	176	197	157
HMMER	3.0	Used for searching sequence databases for homologs of protein sequences, and for making protein sequence alignments.	47	34	38
MUMmer	3.22	A system for rapidly aligning entire genomes.	82	97	88
QuEST	2.42	Statistical software for analysis of ChIP-Seq data and peak calling.	742	974	698
Velvet	1.1.05	A popular sequence assembler for very short reads.	63	70	59

Conclusions and discussion

Existing commercial clouds are not suitable for parallel HPC workloads.

The benchmarks here, and elsewhere[13], show that parallel applications featuring anything more complex than embarrassingly parallel communication patterns cannot efficiently employ more than one computing node. This means that the vast majority of the scientific workload being run at CSC cannot be efficiently executed on a cloud platform; 87% of CPU cycles at CSC were spent in jobs using more than 32 cores in 2015.

The main reason for this lack of performance is that the large commercial cloud infrastructures are built on Ethernet interconnects, which have considerably higher latency and lower bandwidth that Infiniband (used in Taito) or proprietary interconnects such as Cray Aries (used in Sisu). Some performance gains can be obtained by accessing the network card directly (SR-IOV or PCI passthrough). However, these do not completely solve the performance issues[14].

Additionally, the largest simulations on Sisu utilize well over 10,000 cores. Running these huge simulations in a cloud will not be feasible to run due to overhead from virtualization, even if there would be cloud services offering high performance networks. This means that for the high-end workloads there are no alternatives to supercomputing centers and infrastructures.

Commercial clouds are not price-competitive.

In the price-performance comparison it was shown that the only instances that have a comparable price-performance ratio to CSC is spot priced AWS EC2 instances, and preemptable instances on Google cloud. Reserved and on-demand instances are significantly more expensive. The big drawback in spot pricing is that the price for these instances fluctuates, and they may not be available at a reasonable cost and can be terminated at any time. This limits the the fraction of workflows that can be adapted to it. Also, as discussed above parallel applications can in any case not be efficiently run on these resources.

For parallel applications one may turn to the HPC centric cloud providers listed earlier. They use Infiniband, and while we have not benchmarked their performance we expect them to provide good performance for medium sized parallel jobs using up to one thousand cores. Looking at pricing shows that these HPC cloud providers are even more expensive, with more than three times higher cost per node hour (Table 1.)

In addition to the costs related to compute resources there will also be charges for data transfers and storage. Depending on the case at hand these can range from being very small, to being very large. These may also lead to a lock-in effect, where it is difficult to change to another computing platform since retrieving data from deep storage such as AWS Glacier and data transfer costs can make it too expensive to move the data.

---

[13] G. K. Lockwood, M. Tatineni, and R. Wagner, "SR-IOV: Performance Benefits for Virtualized Interconnects," in *XSEDE '14*, 2014, pp. 1–7.

[14] J. Zhang, X. Lu, and D. K. Panda, "Performance Characterization of Hypervisor-and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters," 2016, pp. 1777–1784.

The main reasons for the competitiveness of CSC's resources can primarily be attributed to the following facts:

1. There is little margin of profit in the pricing.
2. Since the system is accessed through a queuing system designed for scientific computing it can maintain very high level of utilization. In a cloud environment a large fraction of the resources are idle, to be able to cope with spikes in demand.
3. The infrastructure is optimized to serve the particular needs of the Finnish scientific computing community. This means that investments, in-house development and user support are sharply focused on this task instead of trying to provide something for everyone, which adds overhead.

Looking toward the future infrastructure at CSC it can be noted that as system size grows, the amount of personnel needed to administer and operate it stays fairly fixed. As CSC pricing is based on actual costs, a larger system will have a lower per core price. An infrastructure that is twice as large, the price per core hour is further reduced by approximately 13%.

In-house cloud and containers offer great opportunities for CSC and CSC customers.

CSC is one of the world's leading HPC centers in providing cloud services: Production cloud services have been running since 2010, there is strong technological expertise in-house, and cloud computing has a central role in CSC's strategy. These resources will be further developed and expanded.

Many CSC customers using the cPouta cloud are working on big data workloads and bioinformatics. For these workloads cloud is a suitable solution, since the interconnect performance is typically not a major concern. The BioBench2 benchmark also shows (Table 5) that the cloud provides good performance for bioinformatics.

Sensitive data, such as patient data, typically cannot be processed in commercial public clouds due to legislation. Even setting up a private cloud that can meet the strict compliance requirements to process such data is not trivial. CSC has developed ePouta to explicitly meet this need. Even with less sensitive data there may be data governance and information security issues if the cloud infrastructure is located abroad.[15] All of CSC's cloud resources are located in Finland.

The emergence of container technologies, most notably Docker, has the potential to enable also domain specific computing environments on clusters and supercomputers with much reduced overhead[16] compared to virtualization. It also enables efficient scheduling of tasks even when the system is highly loaded using HPC batch job queue systems such as SLURM. There are two notable projects in this space, Singularity[17] and Shifter[18], which are already running in production at major HPC sites and they are being also evaluated at CSC at the time of writing.

---

[15] Cloud Computing Benefits, risks and recommendations for information security: https://resilience.enisa.europa.eu/cloud-security-and-resilience/publications/cloud-computing-benefits-risks-and-recommendations-for-information-security

[16] J. Zhang, X. Lu, and D. K. Panda, "Performance Characterization of Hypervisor-and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters," 2016, pp. 1777–1784.

[17] http://singularity.lbl.gov/

[18] http://www.nersc.gov/research-and-development/user-defined-images/

Author details and acknowledgements

This report is commissioned and published by CSC – IT Center for Science Ltd, 2016.

Corresponding author

Dr. Sebastian von Alfthan, Senior Application Specialist
CSC – IT Center for Science
Sebastian.von.alfthan@csc.fi, mobile +358 40 588 8688
www.csc.fi

Working group at CSC

Dr. Pekka Manninen, Dr. Jussi Heikonen, Dr. Sami Ilvonen, Olli-Pekka Lehto