Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 93 (2016) 3 – 11

Procedia Computer Science

6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India

Breaking HPC Barriers with the 56GbE Cloud

Muhammad Atif[a], Rika Kobayashi[a], Benjamin J Menadue[a], Ching Yeh Lin[a], Matthew Sanderson[a], Allan Williams[a]

[a]*National Computational Infrastructure, The Australian National University, Canberra 2601, Australia*

Abstract

With the widespread adoption of cloud computing, high-performance computing (HPC) is no longer limited to organisations with the funds and manpower necessary to house and run a supercomputer. However, the performance of large-scale scientific applications in the cloud has in the past been constrained by latency and bandwidth. The main reasons for these constraints are the design decisions of cloud providers, primarily focusing on high-density applications such as web services and data hosting.

In this paper, we provide an overview of a high performance OpenStack cloud implementation at the National Computational Infrastructure (NCI). This cloud is targeted at high-performance scientific applications, and enables scientists to build their own clusters when their demands and software stacks conflict with traditional bare-metal HPC environments. In this paper, we present the architecture of our 56 GbE cloud and a preliminary set of HPC benchmark results against the more traditional cloud and native InfiniBand HPC environments.

Three different network interconnects and configurations were tested as part of the Cloud deployment. These were 10G Ethernet, 56G Fat-tree Ethernet and native FDR Full Fat-tree InfiniBand (IB). In this paper, these three solutions are discussed from the viewpoint of on-demand HPC clusters focusing on bandwidth, latency and security. A detailed analysis of these metrics in the context of micro-benchmarks and scientific applications is presented, including the affects of using TCP and RDMA on scientific applications.

Peer-review under responsibility of the Organizing Committee of ICACC 2016

Keywords: Scientific Applications, Cloud Computing, High Performance Computing, High Performance Ethernet, RDMA over Ethernet, InfiniBand

1. Introduction

Cloud computing is increasingly being considered by the scientific community as an alternative platform for high-performance computing (HPC) due to its advantages of cost-effectiveness and flexibility. Typical HPC centres are traditionally based around a peak supercomputer, purpose-built for high-performance applications. However such supercomputers cannot always accommodate all of the scientific community's needs, chiefly for capacity and con-

* Muhammad Atif. Tel.: +61 02 61255031.
E-mail address: muhammad.atif@anu.edu.au

Peer-review under responsibility of the Organizing Committee of ICACC 2016
doi:10.1016/j.procs.2016.07.174

figuration reasons. As a result HPC clouds are gaining popularity in such circumstances. This has been recognised by commercial cloud providers, with many of these now offering specialist HPC instances, among them Amazon[1], Google Compute Engine[2] and Microsoft Azure[3]. Early evaluations of such commercial cloud solutions found their performance to be poor with respect to typical HPC systems[4,5,6]. Nevertheless, there was a recognition that the advantages of virtualised clusters, including instant and on-demand availability, could sufficiently offset the performance limitations to warrant their further investigation. Similarly, private clouds are rapidly gaining popularity at HPC centres as a complementary resource to support applications, workloads and pipelines not well suited to the peak computer. NCI has therefore deployed its cloud infrastructure on RDMA capable 56Gb network infrastructure, fast SSDs and processors typically used in supercomputers. In this paper, timing and scalability results are presented for the OSU MPI micro-benchmarks[7], NASA Advanced Supercomputing Parallel Benchmark suite (NPB)[8], and computational science applications from computational chemistry and computational physics domains. Where applicable, instrumented performance data obtained using IPM[9,10] are presented and detailed performance analysis is performed. By use of a 56Gb network, noticeable improvements have been observed in scalability for parallel applications on the cloud.

The rest of this paper is organised as follows. The motivations for the experimental study are given in Section 2 Related work is discussed in Section 3, which is followed by experimental set-up in Section 5. Evaluation results together with scalability of scientific applications are presented in Section 6, which lead to our conclusions and future work in Section 7.

2. Motivation

The biggest motivations for running HPC in the Cloud are user requirements that cannot be satisfied on a supercomputer, for reasons ranging from security to application performance on a parallel filesystem. For example, workloads generating small random reads and writes are not typically suited to the Lustre filesystem[11].

The supercomputing cluster is typically a highly contended resource, and users are often subjected to limiting usage quotas. Some user workloads may not make good use of the cluster, for example those whose communication requirements could have been satisfied by a cluster with a commodity network such as 10GigE, or which require software that cannot be supported on the supercomputer. Jobs dedicated to debugging and validation also typically do not require the supercomputing cluster unless interconnect performance is in question. Even when a job is not well suited to the cloud, in times of high demand, the use of a cloud as an alternative site may result in a shorter turnaround. In such situations, the users' jobs could better be run on a cheaper private cloud, or even a public cloud.

This however requires that the same comprehensive software stack can be easily replicated on the cloud environments, so that the availability of the cloud facilities comes with little extra effort to the user (and ideally would be transparent).

Having the ability to package up a standard HPC working environment into VMs gives HPC centers the ability to cloud-burst[12,13,14,15,16] as a means of responding to peak demand, when local resources are saturated, or when it is simply more cost-effective to do so.

In order to achieve this goal, we first investigate the feasibility of packaging the environment and the performance impacts of doing so for various workloads. This paper serves as a preliminary study in this respect. The second stage will involve development of supporting infrastructure.

3. Related Work

A considerable body of research exists into the use of virtualisation and/or cloud computing in HPC environments. Researchers have tested virtualisation in several scenarios in order to make a case for HPC in the cloud environment. Some related research work is discussed briefly in this section.

Ramakrishnan et. al.[17] present their experiences with a cross-site science cloud running at the Argonne Leadership Computing Facility and at the NERSC Facility. They deployed test-beds for running diverse software stacks aimed at exploring the suitability of cloud technologies for DoE science users. Some salient findings are that (a) scientific applications have requirements which demand tailored solutions (e.g. parallel file-systems, legacy datasets and pre-tuned

software libraries), requiring clouds designed for science needs; (b) scientific applications with minimal communications and I/O make the best fit for cloud deployment and (c) clouds require significant end-user programming and system administrator support.

Strazdins et. al.[4] compared the performance of Amazon EC2 Cloud, a private cloud based on VMWare ESX 4.1 and a supercomputer hosted at the NCI national facility in Australia. Benchmarks used were OSU MPI micro-benchmarks, the NAS Parallel benchmarks and two large scientific application codes (the UK Met Office's MetUM global climate model[18] and the Chaste multi-scale computational biology code[19]). The work showed the performance of the interconnect, especially in the case of communications-bound applications using small message sizes. The key finding was that communications-bound parallel applications, especially those using short messages, were at a disadvantage on the cloud infrastructure. It was also observed that IO intensive applications performed poorly due to the lack of infrastructure to properly support distributed filesystems such as Lustre.

He et. al.[5] found that most clouds they evaluated were optimised for business/enterprise applications. They noted the importance of a good system interconnect and the ease-of-use afforded by on-demand resources for the scientific community.

Ostermann et. al[20] assess whether commercial cloud resources are sufficient for scientific computing needs. In this work they use micro-benchmarks (lmbench[21], bonnie, CacheBench[22]) and HPCC[23] kernels). Their main finding is that the performance and reliability of commercial clouds are not sufficient to support scientific computing needs.

Jackson et. al.[24] present results from porting a complex astronomy pipeline, for detecting supernovae events onto Amazon EC2. They were able to encapsulate complex software dependencies and note that the EC2-like environments being more complex present a very different resource environment in comparison to a traditional HPC center e.g. images not booting up correctly, performance perturbations arising from co-scheduling with other EC2 users.

Jackson et. al.[10] conduct a thorough performance analysis using applications from the NERSC benchmarking framework. The Community Atmospheric Model from the CCSM code developed by NCAR was one of the benchmarks that were run. They find there is a strong correlation between communication time and overall performance on EC2 resources i.e. performance suffers the most for applications with greater global communication patterns. They also find there is significant performance variability between runs.

Evangelinos et. al[6] presented a detailed comparison of the MPI implementations LAM/MPI, MPICH, OpenMPI and GridMPI to test Amazon's HPC cloud. A custom application for the atmosphere-ocean climate model and the NAS parallel benchmarks were utilised to evaluate the system. It was concluded that the performance of Amazon's Xen based cloud is below the level seen at dedicated supercomputer centers. However, performance was comparable to low-cost cluster systems. Significant performance deficiency arose from the messaging performance, which for a communication intensive application was 50% slower compared to similar 'non-cloud' compute infrastructure.

While there have been a number of papers investigating cloud environments for HPC, this paper is different in that (a) we believe this to be the only study based on 56G Ethernet interconnect as opposed to 10G Ethernet used by other cloud providers (b) the study uses Remote Direct Memory Access (RDMA) over Ethernet for the virtualised cluster (c) our cloud incorporates lessons learned from our previous research[4] and incorporates the same processors used in the flagship supercomputer and SSD drives in RAID-0 configuration to give high I/O operations (IOPS) (d) we present a detailed performance analysis which includes complex memory intensive applications.

4. Design of the 56GbE Cloud

The 56GbE Cloud platform is based on the Kilo release of RDO[25]. RDO is an open source community effort for using and deploying OpenStack on Red Hat Enterprise Linux and distributions derived from it (such as CentOS, Scientific Linux and others)[25]. The cloud is based on Centos 6 and uses Single Root I/O Virtualisation (SR-IOV) to expose the 56G Mellanox Ethernet hardware devices (HCAs) directly to guest Virtual Machines as SR-IOV Virtual Functions (VFs). With this capability enabled, network traffic originating from or destined to a guest no longer passes through the hypervisor kernel. This eliminates a major performance bottleneck, significantly reducing latency and increasing bandwidth. Benchmarks show guest-to-guest performance closely approaching the theoretical maximum line rate.

In order to permit guests direct access to the network while preserving security, 56GbE-Cloud leverages another key capability of Mellanox SR-IOV capable hardware: VLAN tagging in HCA firmware. The Mellanox 56G Ethernet

Table 1: Description of the Experimental Platforms used in this paper. FDR-SC is a large supercomputer, 56GbE-Cloud is a private in-house cloud resource, the 10G-Cloud is a private cloud based on 10 Gigabit Ethernet. A 'node' in the table refers to physical hardware. VMM: Virtual Machine Manager.

Platform		FDR-SC	56GbE-Cloud	10GbE-Cloud
# of Nodes		3592	200	xxx
CPU	Model	Xeon E5-2670	Xeon E5-2670	Opteron 63xx
	Freq.	2.60GHz	2.60GHz	2.30GHz
	#Cores	8 (2 sockets)	8 (2 sockets)	8 (2 sockets)
	L2 Cache	20MB (shared)	20MB (shared)	8x2MB
Memory per node		32GB	120GB	64GB
Operating System		CentOS 6.7	CentOS 6.7	CentOS 6.7
File System		Lustre	NFS	NFS
Interconnect		FDR IB	56GbE	10GbE
MPI Library		OMPI 1.8	OMPI 1.8 (Yalla)	OMPI 1.8 (tcp btl)
VMM		Native	OpenStack (KVM)	OpenStack (KVM)

HCAs are configured by OpenStack to assign a VLAN ID to each VF. When a guest instructs a VF to send data on the network, the HCA firmware automatically adds a VLAN tag to the Ethernet frames. Conversely, when Ethernet frames arrive at the HCA with a VLAN tag, the tag is stripped before the frame data is supplied to the destination guest. The guest is not permitted to control this process, nor even to discover that it is occurring. The VF/VLAN ID table is controlled by Neutron, the virtual networking component of OpenStack. In this way, each tenant network exists in a separate Layer 2 broadcast domain from each other network. Similarly, the underlying physical network is another separate Layer 2 broadcast domain. This prevents tenants from accessing each others' networks or the underlying physical network without authorisation.

We are currently investigating the possibility of an analogous system wherein the Mellanox HCAs would be flashed to Infiniband instead of Ethernet, and tenant networks would be isolated from each other by use of Infiniband partitions rather than Ethernet VLAN tags.

5. Experimental Setup

We have compared three different compute platforms. The details of the experiments are provided in Table 1. The first platform is a 1.2 Petaflop supercomputer at the NCI, hosted by the Australian National University. There are 3592 nodes consisting of Fujitsu Primergy blade servers and a Mellanox FDR based non-blocking full fat-tree topology which is used both for compute and for access to over 40 Petabytes of Lustre object-based storage.

The second platform is a virtualised cluster running on top of an OpenStack [26] cloud deployment housed at the NCI. This cloud deployment is based on OpenStack Kilo and is backed by a full fat-tree FDR Infiniband fabric flashed with Ethernet. The cloud uses Single Root I/O Virtualisation (SR-IOV) to provide 56Gb Ethernet directly to the virtual machines.

The third platform is a virtual cluster deployed on an OpenStack Kilo build with a commodity 10GbE solution and AMD Opteron processors. [1].

All applications used in benchmarking were compiled by `gcc -O3` option natively on each platform unless mentioned otherwise. Specialised development tools such as Intel compilers were not used due to licensing issues on the remote site and in order to ensure a fair comparison.

OpenMPI [27] version 1.8 was used for all the experiments. For FDR-SC, a native OpenIB byte transport layer (btl) was used. For 56GbE-Cloud, an experimental transport layer named Yalla developed by Mellanox was used, which essentially provides RDMA access. For the 10GbE cloud, native `tcp btl` was used. All the virtual machines instantiated for the experiments consisted of 16 cores each. It may be noted that the OpenStack Icehouse release does not support Non-Uniform Memory Access (NUMA).

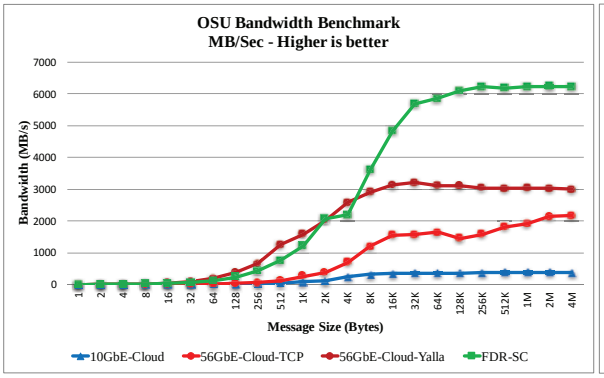
6. Results

In this Section, we present the results from micro and application benchmarks. We used OSU MPI bandwidth and latency benchmarks [7] to measure sustained message passing bandwidth and latency. The NAS Parallel Bench-

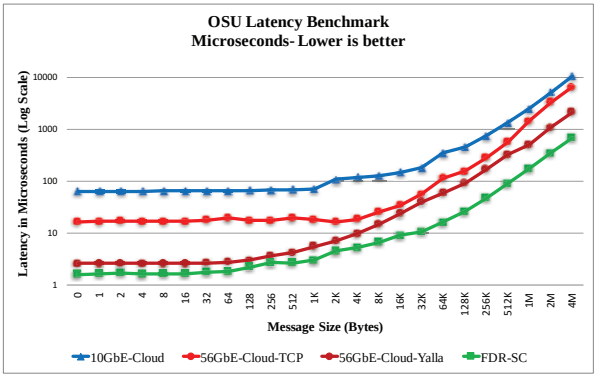
[1] Due to sensitivities, we are not providing the site name

Fig. 1: OSU Benchmark comparison of an FDR supercomputer, 56G ethernet Cloud and 10G ethernet cloud

(a) OSU MPI bandwidth results



(b) OSU MPI latency results



mark (NPB)[8] MPI suite version 3.4 was used to compare the performance of the network for different patterns of communication.

We also benchmarked the system with real-world scientific applications and data. We used a popular computational chemistry code NAMD[28] and a custom hybrid monte-carlo code from the computational physics domain.

Each experiment was conducted 10 times and averages are discussed in subsequent subsections. The standard deviation of all the experiments were within 3% for 10GbE-Cloud whereas 56GbE-Cloud and FDR-SC had standard deviations within 1%.

6.1. OSU MPI Benchmarks

The OSU point-to-point MPI bandwidth and latency benchmarks[7] measure the sustained message passing bandwidth and latency between two processes. For these experiments, we launch a two process MPI application with each process on a distinct compute node. For optimal results, we ensured that each process was running on the socket which had the PCIe network interface. This ensured that the Intel QuickPath Interconnect (QPI) between the sockets was not used, since it introduces additional latency.

In Figures 1a and 1b, the x-axis represents message size in bytes and y-axis for bandwidth (MB/s) and latency (s) using a log scale respectively. As expected, the 10GbE-Cloud was an order of magnitude slower than both 56GbE and FDR (56Gb) infiniband. 56GbE-Cloud using `tcp` byte transport layer (`btl`) was around 5.6 times faster than the 10GbE-Cloud at a message size of 2MB, showing that the difference in link speeds (a factor of 5.6) is being reflected in this benchmark. As the virtual machines on 56GbE-Cloud use SR-IOV, we ran the same benchmark with Mellanox's `Yalla` point to point management library. `Yalla` makes use of Mellanox Messaging Library (`MXM`) to provide RDMA capabilities over the Ethernet fabric. It can be seen from the Figure 1a that Yalla consistently performed better than `tcp` on 56GbE-Cloud, showing the benefits of using RDMA relative to TCP in an HPC environment.

OSU Latency benchmarks also presented similar trends with 56GbE outperforming 10GbE. For smaller message sizes (<1024 bytes) the performance of 10GbE TCP and 56GbE Yalla differed by a factor greater than 16.5. However the performance of native InfiniBand and that of the 56GbE solution with SR-IOV on virtual machines differed by a factor of 1.5.

6.2. NAS Parallel Benchmarks

We used NAS Parallel Benchmarks (NPB) class 'C' to determine the impact of different CPU and communication loads on three platforms. The NPB are a small set of programs designed to help evaluate the performance of parallel supercomputers. These application kernel benchmarks are derived from computational fluid dynamics (CFD) applications. Figure 2 presents the normalised elapsed times of the benchmarks on FDR-SC and 56GbE-Cloud with respect

Fig. 2: NPB (class C) execution time normalised w.r.t 32 processes on 10GbE-Cloud.

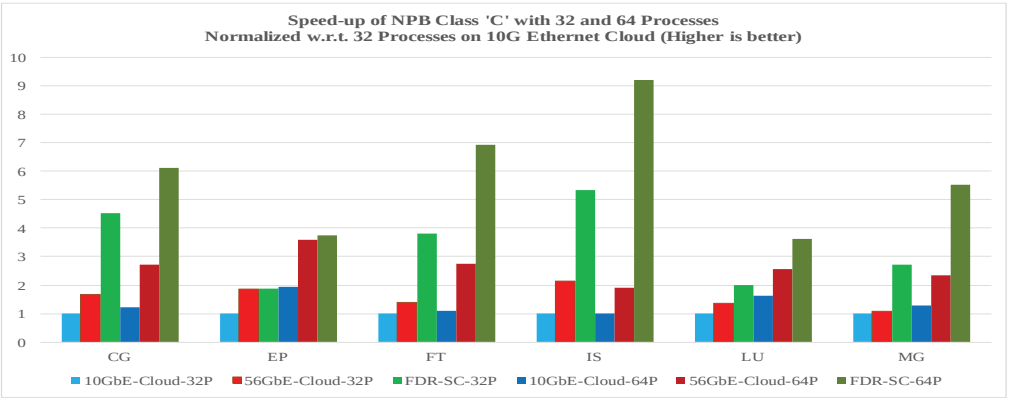
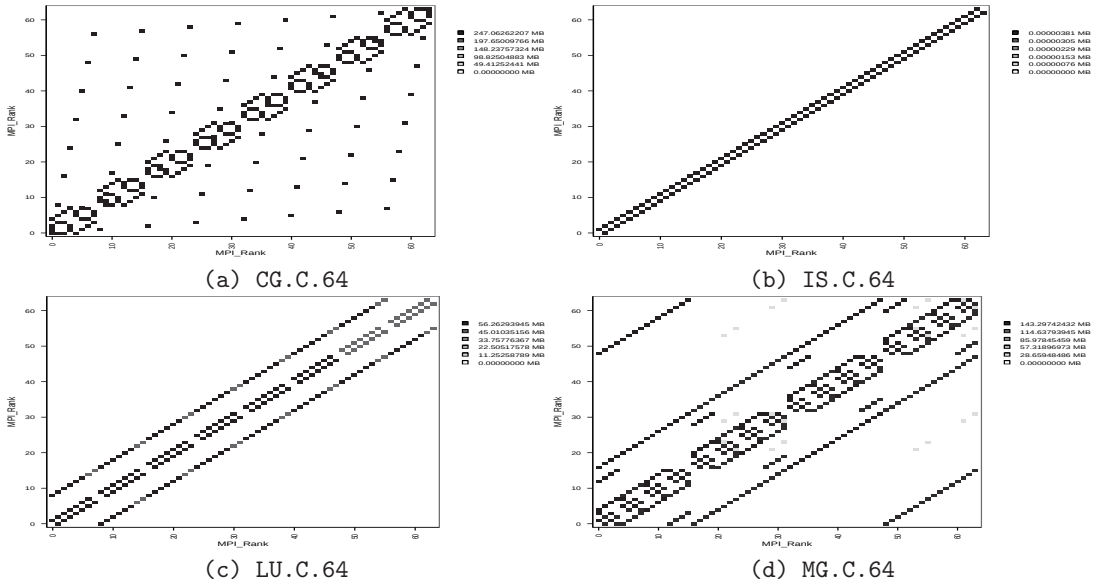


Fig. 3: Communication profile of NPB C.64 benchmarks



to 10GbE-Cloud. We used 32 and 64 processes for each of the benchmarks to see the scalability trends. The 56GbE `TCP BTL` results of 56GbE-Cloud were in line with the findings from Section 6.1 and hence not shown in Figure 2.

The Conjugate Gradient (CG) Benchmark uses `MPI_Send` and processes mostly communicate with their neighbours. As shown in Figure 3 (a), the processes communicate in blocks of 8 processes while occasionally communicating with distant processes (`rank`. As the compute nodes have 16 cores each, the inter-node communication with distant processes does not tax the bandwidth of a single network interface (HCA) and all three experimental platforms scale well. However, 10GbE-Cloud performs poorly compared to 56GbE-Cloud and FDR-SC. Interestingly, 56GbE-Cloud shows the best scalability (not performance) for the CG benchmark. We believe this to be due to the workload distribution of Class 'C', which is less intense for 64 processes compared to the 32 process benchmark. This results in less stress on the memory subsystem during a 64 process run which suffers from lack of non-uniform memory access (NUMA) in OpenStack.

In the case of the Embarrassingly Parallel (EP) benchmark, where there is no communication, FDR-SC, 56GbE-Cloud and 10GbE Cloud show almost linear speed-ups. The results for FDR-SC and 56GbE-Cloud are similar as they share the same CPU architecture. The 10GbE cloud uses AMD Opteron and is 80% slower than the Intel processors.

The FT Benchmark uses `MPI_AlltoAll` communication, where each task in the communication group performs a scatter operation, sending a distinct message to all the tasks in the group in order by index. For the FT benchmark we see FDR-SC and 56GbE-Cloud scaling quite well. However 10GbE-Cloud does not scale at all due to the intense all to all communication pattern. We believe this due to the overheads of TCP.

The Integer Sort (IS) Benchmark scales almost linearly on FDR-SC but we see 10GbE-Cloud and 56GbE-Cloud actually slow down. IS is a memory intensive benchmark and the processes communicate with neighbours as shown in Figure 3 (b). As the OpenStack Icehouse release is not NUMA aware, 10GbE-Cloud and 56GbE-Cloud do not scale. We expect IS to scale well with the introduction of NUMA-aware scheduling of virtual machines in OpenStack.

The Lower-Upper symmetric Gauss-Seidel (LU) Benchmark is a latency and floating point bound application kernel. We see the benchmark scales well on all three platforms. However, the speed-up on 10GbE-Cloud is significantly lower than that of 56GbE-Cloud and FDR-SC. It can be seen in Figure 3 (c) that there is occasional inter-node communication like that seen in the CG benchmark. However, as the benchmark is not bandwidth bound (notice the number of bytes transferred in the Figure), 10GbE-Cloud scales better in the case of the LU benchmark than in the case of the CG benchmark.

The Multi-Grid (MG) Benchmark has communication patterns similar to CG and it also uses neighbour communication in clusters. However, processes also communicate with distant processes in the communication world, making the communication pattern more complex than CG benchmark. Due to more inter-node communication overheads of TCP, the benchmark does not scale on 10GbE-Cloud but scales well on 56GbE-Cloud and FDR-SC. FDR-SC performs better than 10GbE and 56GbE-Cloud due to native IB.

From NPB benchmarks, we conclude that using 56GbE and RDMA for HPC workloads on the cloud have advantages over traditional 10GbE. We have seen 56GbE to scale 2 to 3 times better than 10GbE. However bare-metal clusters still hold the advantage due to native IB and presence of NUMA awareness.

6.3. Scalability of Scientific Applications

We chose three real scientific applications from distinct computational science areas namely Computational Chemistry and Computational Physics. The results are discussed in detail in the following sections.

6.3.1. Computational Chemistry Workload - NAMD

NAMD is a high-performance molecular dynamics code developed by the Theoretical and Computational Biophysics Group in the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign[28]. It is highly parallelised, scaling well to thousands of processors on a variety of parallel computer architectures, thus becoming one of the stock chemistry programs for benchmarking on HPC platforms (see, for example,[29,30]). The NAMD website provides a standard benchmark Apolipoprotein-A1 (ApoA1) system consisting of 92,224 atoms, running the particle-mesh Ewald (PME)[31] method. PME is a computation intensive algorithm and has been optimised for scalability using the CHARM++ dynamic load-balancing message-driven framework[32], such that communication is the dominant factor affecting performance. NAMD version 2.9 built against the GNU compilers and OpenMPI 1.8.2. The speed-up results are shown in Figure 4a.

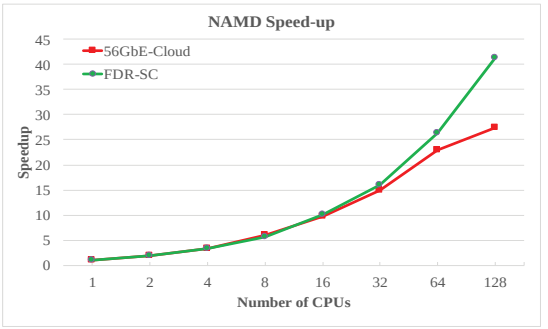
As the 10GbE-Cloud was significantly slower due to its processor architecture, we decided to remove its results and present results from alike architectures. The performance of the ApoA1 benchmark, as measured by step time, for a representative subset of the various platforms and communication models investigated, is presented in Table and plotted in Figure. The performance and scaling to 128 processors of the cloud matches that of the HPC system showing it to be a competitive alternative. Of note is that the `yalla` communication model does not perform as well as TCP, which was also found for other communication models (not presented here). This indicates the importance of choosing the right model for the problem, which warrants further investigation. This sensitivity to communication also reflects the findings of Poghosyan et al.[30], who investigated NAMD parallel performance against a variety of network interconnects.

6.3.2. Lattice QCD

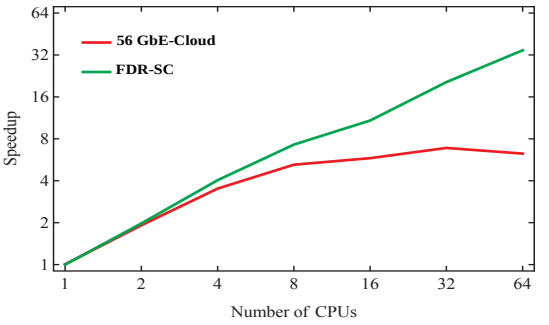
Lattice QCD is a well-established technique for investigating quantum chromodynamics (QCD) – the theory describing the strong interaction, one of the four fundamental forces of Nature, between quarks and gluons. These calculations are extremely computationally intensive, and are run on many of the largest supercomputers in the world.

Fig. 4: Comparison of NAMD and Cola Lattice

(a) Speed-up comparison of NAMD



(b) Speed-up of COLA lattice QCD code



We use the COLA package to investigate the performance of such simulations on the systems of interest. We use a space-time lattice size of $(x, y, z, t) = (8, 8, 8, 128)$ so that we decompose the lattice in the t-direction only for each n_{CPU} investigated, and an even-odd preconditioned, FLIC fermion action. Matrix inversions are calculated using the bi-conjugate gradient stabilised (BiCGSTAB) method.

In Fig. 4b we show the speedup obtained as we increase the number of cores involved in the calculation on FDR-SC and 56GbE-Cloud using four transport mechanisms, normalised to 1 core on FDR-SC using the TCP BTL. Scaling on FDR-SC is nearly perfect for all transport mechanisms up to 8 CPUs, although above this the small lattice size manifests as extra communication overhead and limits scalability. It is expected that a larger lattice (in the spatial directions) will result in improved scalability to a larger number of cores.

On the other hand, the 56GbE-Cloud results show significant scalability issues beyond 4 CPUs. We suspect this is due to the lack of NUMA visibility in the guest operating system. The physical hardware has two NUMA nodes, each with 8 CPUs and 64GB of memory; however this is presented to the guest as a single NUMA node with all 16 cores. As a result, some processes end up with their memory allocations being placed on the wrong (physical) NUMA node, and since this is a memory-bound calculation the impact of NUMA-remote accesses is significant.

7. Conclusion and Future Work

We used the OSU MPI micro-benchmarks, the NPB MPI macro-benchmarks to characterise performance of three different platforms including a peak supercomputing HPC cluster, a private cloud based virtual cluster with 56G Ethernet and a HPC Cluster hosted on a 10G Ethernet cloud. Importantly, we also used applications from computational chemistry and computational physics to evaluate and analyse these platforms. The key finding from our experimental results is while HPC in the cloud still lags behind the bare-metal supercomputers, we have seen applications scale better when RDMA over Ethernet is used over traditional 10GbE TCP based solutions. We have seen applications sale up to 2.5 times when we used 56GbE and Intel processors with AVX. NUMA-awareness still seems to cause significant slow down for memory intensive applications. We expect this situation to improve when OpenStack becomes NUMA-aware.

In future, we will test large scale jobs on the cloud spanning thousands of cores and use technologies like Docker.

Acknowledgements

This research was made possible by the Australian government's National Collaborative Research Infrastructure Strategy (NCRIS) funding.

References

1. "Amazon Elastic Compute Cloud (EC2)," http://www.amazon.com/ec2, Jul 2015. [Online]. Available: http://www.amazon.com/ec2

2. "Google Cloud Engine," Oct 2015. [Online]. Available: https://cloud.google.com/compute/
3. "Microsoft Azure," Dec 2014. [Online]. Available: http://azure.microsoft.com/
4. P. Strazdins, J. Cai, M. Atif, and J. Antony, "Scientific application performance on hpc, private and public cloud resources: A case study using climate, cardiac model codes and the npb benchmark suite," in *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International*, May 2012, pp. 1416–1424.
5. Q. He, S. Zhou, B. Kobler, D. Duffy, and T. McGlynn, "Case study for running HPC applications in public clouds," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 395–401. [Online]. Available: http://doi.acm.org/10.1145/1851476.1851535
6. C. Evangelinos and C. N. Hill, "Cloud Computing for parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2." Cloud Computing and Its Applications, October 2008. [Online]. Available: http://www.cca08.org/speakers/evangelinos.php
7. "OSU Benchmarks," Oct 2010. [Online]. Available: http://nowlab.cse.ohio-state.edu/projects/mpi-iba/
8. "NAS Parallel Benchmarks," http://www.nas.nasa.gov/Software/NPB, Sep 2010. [Online]. Available: http://www.nas.nasa.gov/Software/NPB/
9. D. Skinner, "Performance monitoring of parallel scientific applications," Lawrence Berkeley National Laboratory, Tech. Rep. LBNL Paper LBNL-PUB-5503, 2005.
10. K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. Wasserman, and N. Wright, "Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 30 2010-dec. 3 2010, pp. 159 –168.
11. K. Ren, Q. Zheng, S. Patil, and G. Gibson, "Indexfs: Scaling file system metadata performance with stateless caching and bulk insertion," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 237–248. [Online]. Available: http://dx.doi.org/10.1109/SC.2014.25
12. M. Schatz, B. Langmead, and S. Salzberg, "Cloud computing and the dna data race." *Nat Biotechnol*, vol. 28, no. 7, pp. 691–3, 2010.
13. M. Humphrey, Z. Hill, K. Jackson, C. van Ingen, and Y. Ryu, "Assessing the value of cloudbursting: A case study of satellite image processing on windows azure," in *E-Science (e-Science), 2011 IEEE 7th International Conference on*, dec. 2011, pp. 126 –133.
14. J. Wang, D. Crawl, and I. Altintas, "Kepler + hadoop: a general architecture facilitating data-intensive applications in scientific workflow systems," in *Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science*, ser. WORKS '09. New York, NY, USA: ACM, 2009, pp. 12:1–12:8. [Online]. Available: http://doi.acm.org/10.1145/1645164.1645176
15. H. Kim, M. Parashar, D. Foran, and L. Yang, "Investigating the use of autonomic cloudbursts for high-throughput medical image registration," in *Grid Computing, 2009 10th IEEE/ACM International Conference on*, oct. 2009, pp. 34 –41.
16. L. Ramakrishnan, K. R. Jackson, S. Canon, S. Cholia, and J. Shalf, "Defining future platform requirements for e-Science clouds," in *Proceedings of the 1st ACM symposium on Cloud computing*, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 101–106. [Online]. Available: http://doi.acm.org/10.1145/1807128.1807145
17. L. Ramakrishnan, P. T. Zbiegel, S. Campbell, R. Bradshaw, R. S. Canon, S. Coghlan, I. Sakrejda, N. Desai, T. Declerck, and A. Liu, "Magellan: experiences from a science cloud," in *Proceedings of the 2nd international workshop on Scientific cloud computing*, ser. ScienceCloud '11. New York, NY, USA: ACM, 2011, pp. 49–58. [Online]. Available: http://doi.acm.org/10.1145/1996109.1996119
18. T. Davies, M. J. P. Cullen, A. J. Malcolm, M. H. Mawson, A. Staniforth, A. A. White, and N. Wood, "A new dynamical core for the Met Office's global and regional modelling of the atmosphere," *Q. J. R. Meteorol. Soc*, vol. 131, pp. 1759–1782, 2005.
19. J. Pitt-Francis and al, "Chaste: A test-driven approach to software development for biological modelling," *Computer Physics Communications*, vol. 180, no. 12, pp. 2452–2471, 2009.
20. S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 Cloud computing services for scientific computing," in *Cloud Computing*, ser. Lecture Notes of the Institute for Computer Sciences, D. R. Avresky, M. Diaz, A. Bode, B. Ciciani, and E. Dekel, Eds. Springer, 2010, vol. 34, no. 4, ch. 4, pp. 115–131. [Online]. Available: http://www.springerlink.com/content/t640753r2597524u/fulltext.pdf
21. L. McVoy and C. Staelin, "lmbench: portable tools for performance analysis," in *Proceedings of the 1996 annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 1996, pp. 23–23. [Online]. Available: http://dl.acm.org/citation.cfm?id=1268299.1268322
22. Phillip J. Mucci and Kevin London and Philip J. Mucci, "The CacheBench Report," Tech. Rep., 1998.
23. P. Luszczek, J. J. Dongarra, D. Koester, R. Rabenseifner, B. Lucas, J. Kepner, J. Mccalpin, D. Bailey, and D. Takahashi, "Introduction to the hpc challenge benchmark suite," Tech. Rep., 2005.
24. K. R. Jackson, L. Ramakrishnan, K. J. Runge, and R. C. Thomas, "Seeking supernovae in the clouds: a performance study," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 421–429. [Online]. Available: http://doi.acm.org/10.1145/1851476.1851538
25. "RedHat OpenStack Distribution," 2015. [Online]. Available: https://www.rdoproject.org
26. "OpenStack," 2015. [Online]. Available: https://www.openstack.org
27. "OpenMPI," http://www.open-mpi.org/, June 2015.
28. J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, "Scalable molecular dynamics with namd," *Journal of computational chemistry*, vol. 26, no. 16, pp. 1781–1802, 2005.
29. S. Kumar, C. Huang, G. Zheng, E. Bohm, A. Bhatele, J. C. Phillips, H. Yu, and L. V. Kalé, "Scalable molecular dynamics with namd on the ibm blue gene/l system," *IBM Journal of Research and Development*, vol. 52, no. 1.2, pp. 177–188, 2008.
30. A. Poghosyan, L. Arsenyan, H. Astsatryan, M. Gyurjyan, H. Keropyan, and A. Shahinyan, "Namd package benchmarking on the base of armenian grid infrastructure," 2012.
31. T. Darden, D. York, and L. Pedersen, "Particle mesh ewald: An n log (n) method for ewald sums in large systems," *The Journal of chemical physics*, vol. 98, no. 12, pp. 10 089–10 092, 1993.
32. L. V. Kale and S. Krishnan, *CHARM++: a portable concurrent object oriented system based on C++*. ACM, 1993, vol. 28, no. 10.