Towards Understanding HPC-Big Data Convergence Using Cloud Platforms (Unrefereed Workshop Manuscript)

Shweta Salaria[1,a)] Kevin Brown[1,b)] Hideyuki Jitsumoto[1,c)] Satoshi Matsuoka[1,d)]

Abstract: The path to HPC-big data convergence has resulted in numerous researches that demonstrate the performance-cost tradeoff between running applications on supercomputers and cloud platforms. Previous studies typically focus on either scientific HPC benchmarks or a specific cloud configuration, failing to consider all the opportunities offered by cloud platforms. We present a comparative study of the performance of representative big data benchmarks, or "Big Data Ogres", and HPC benchmarks running on supercomputer and cloud. Our work distinguishes itself from previous studies in a way that we explore multiple cloud configurations: Shared, Dedicated and Spot Instances. Our results provide a more comprehensive performance-cost trade-off, thereby highlighting the gap that needs to be bridged to attain HPC-big data convergence.

1. Introduction

High performance computing (HPC) systems are fundamental to solving complex scientific problems by utilizing the high computation power of thousands of processors and high-throughput networks. However, the use of HPC systems for Big Data problems is becoming more common across HPC centers. Big Data refers to the diverse, complex, and massive data sets that can contain structured, semi-structured and unstructured data. These data sets are difficult to be stored, processed and analyzed by traditional database technologies. Hence, it seeks a set of new technologies and advanced data analytics methods for data distribution, management and processing. The trend in the recent decade has been to extend the application of HPC systems beyond the computationally-intensive scientific domain to data-intensive domains, or the domain of "Big Data". While HPC has its roots in solving compute-intensive scientific and large-scale distributed problems, Big Data problems have been proven to benefit from typical HPC environments: high processing power, low-latency networks and non-blocking communications [1]. Also, such environments are used for running some typical data analytics problems such as graph processing and its application in cybersecurity, social networks, medical informatics etc. There are various approaches proposed for employing HPC to support data-intensive applications e.g. Map-Reduce-MPI [2], Pilot-MapReduce [3] etc.

Cloud environments have been ideal for tackling big data problem, addressing the problem's need for handling data at scales far beyond the limits of most monolithic systems. These environments such as Amazon EC2 support the big data processing by providing a managed Hadoop framework and making it quick and cost-effective by the ease of scalibilty offered by these platforms. They also allow people to run other popular distributed frameworks such as Apache Spark and Presto.

HPC has given the most significant contribution to scientific discovery and is exploring its applications in business analytics. However, the use of HPC systems is limited to scientists and researchers who have access to supercomputing labs and centers. Cloud environments can help bringing HPC access to users who can't afford complex setup and huge infrastructure costs. To ensure that the cloud is a suitable environment, it is important to understand the capabilities of these platforms so that users have a clear understanding of what to expect in terms of performance. Amazon EC2, the leading infrastucture as a service (IaaS) provider announced the new generation of compute-optimized c4 instances in 2015 with the aim of targeting HPC audience. These instance are supposed to offer the power of an HPC system while leveraging the flexibility of the cloud, ideal for the true HPC-Big Data convergence.

This work assesses the performance tradeoff when using HPC-like clouds versus dedicated HPC systems. For this evaluation, we compare the performance of a traditional HPC scientific application, NICAM, and a traditional Big Data application, Graph500, on the TSUABME2.5 supercomputer and Amazon AWS EC2 cloud. We show the effectiveness of various AWS EC2 new generation of compute-optimized instances against a world leading supercomputer.

Our work distinguishes itself from the past cloud vs. HPC evaluation works in the following aspects:

1. We compare the computation and communication performance of on-demand shared and dedicated instances, spot instances on AWS EC2,

[1] Tokyo Institute of Technology
a) salaria.s.aa@m.titech.ac.jp
b) brown.k.aa@m.titech.ac.jp
c) jitumoto@gsic.titech.ac.jp
d) matsu@is.titech.ac.jp

2. We evaluate the suitability of the new AWS c4 instances as replacements for traditional supercomputers by using a microbenchmark, HPC miniapp, and a traditional big data application, and
3. We comparatively evaluate the usability of supercomputers and clouds for solving data-intensive and compute-intensive problems.

2. HPC in Cloud

There have been several studies to evaluate the promise of cloud platforms for HPC users. In this section, we try to answer the various questions concerned with the use of HPC in cloud.

2.1 Why do we need to use cloud for HPC applications?

High Performance Computing refers to the practice of aggregating computing power to deliver much higher performance than typical desktop computers or laptops in order to solve computationally large problems in science, engineering or business. However, its access is limited to scientists and researchers who use supercomputers to solve complex problems in scientific fields such as molecular dynamics, genome analysis, weather forecasting etc. However, its application in business has been most sought in recent years such as transaction processing, data warehousing etc. There needs to be an effective channel by which it is accessible to the masses so that they can enhance their small to medium scale business. Also, there should be a cost effective alternative of using HPC by researchers in academic institutions.

Cloud platforms exhibit pay-as-you-go, elasticity, flexibility, usability and scalability, which have been attracting users to use these environments as a cost effective measure to run their applications or businesses. HPC can be made available to a larger audience by running it on cloud. It can alleviate the infrastructure cost that is otherwise needed to build an in-house HPC cluster. Clouds can be a promising alternative to supercomputers for users who cant afford their own supercomputers.

2.2 What are the instances suitable for HPC applications?

Amazon Elastic Compute Cloud (EC2) is the most evaluated cloud platform to analyze the feasibility of running HPC applications on clouds. It announced the latest generation of compute-optimized instances i.e. c4 instances in 2015 for applications that are compute-bound and seek parallelism offered by multi-core processors. These instances were typically designed to target HPC applications with the highest performing processors and enhanced networking.

2.3 What are the different cloud configurations?

Amazon EC2 allows users to choose the environment based on the needs of their applications by providing different configurations: 1) On-Demand instances 2) Spot instances 3) Dedicated hosts.

1) On-Demand instances: These instances can be used on an hourly basis without any long-term commitment or upfront costs. They offer two types of tenancy: Shared and Dedicated. Dedicated instances imply that they are dedicated for your use and the underlying host machine is not shared by any other instances. While if the tenancy is shared, the host hardware can be shared by multiple instances.

2) Spot instances: Its a cost-effective way to utilize the unused EC2 instances by Amazon EC2. It sets the bid price and then the acquisition of the instance is dependent on the demand and supply for these instances. However, the application is susceptible to interruptions if the instance current market price exceeds users bid price.

3) Dedicated hosts: Dedicated hosts are similar to dedicated instances. The entire server is dedicated for a single users use. However, dedicated hosts give an additional visibility and control to the owner over the placement of instances on different hosts.

2.4 Why do we want to evaluate different cloud configurations in AWS?

AWS has been trying to constantly evolve to meet the needs of HPC users and towards the direction of HPC-oriented clouds. It is important to fully explore the capabilities offered by cloud in the form of various configurations.

3. Evaluation

We conducted performance comparison between Amazon EC2 instances and the supercomputer, TSUBAME2.5. NICAM-DC-MINI, a miniapp from Fiber and the Graph500 benchmark were evaluated on both the platforms.

3.1 Experimental Setup

3.1.1 Amazon EC2 instances

We used the latest generation of compute-optimized instances, c4.8xlarge instances in our cloud experiments. Each c4.8xlarge instance has Intel Xeon E5-2666 v3 (Haswell) 2.93 processors with 60GB memory and 36 vCPUs. These instances are interconnected with 10Gigabit Ethernet and offer optimized Elastic Block Store (EBS). We tested c4.8xlarge with three cloud configurations: Shared, Dedicated and Spot instances. We used upto 16 nodes for our experiments. We built Lustre Parallel file system [4] on these instances as a shared file system. For the file system, we used 1 dedicated node for Management Data Server and another dedicated node for Object Storage Server.

3.1.2 TSUBAME2.5

TSUBAME2.5 is a production supercomputer operated by Global Scientific Information and Computing Center (GSIC), Tokyo Institute of Technology. It has 1,408 thin computing nodes interconnected by QDR Infiniband of dual rail with non-blocking and full bisectional performance.

Each node has two Intel Xeon X5670 2.93GHz processors, three Nvidia tesla K20X GPUs and 54 GB of memory. Each of the CPUs has six physical cores and supports up to 12 threads with Intels hyperthreading technology. The 7PB storage is constructed by using Lustre file system.

3.2 Benchmark and Applications

3.2.1 Intel Microbenchmark

The Intel MPI Benchmarks [5] perform a set of MPI performance measurements for point-to-point and global communication operations for a range of message sizes. We used this microbenchmark to evaluate the communication performance of Amazon EC2 c4.8xlarge instances and TSUBAME2.5. Ping-Pong, MPI_Allreduce and MPI_Alltoall were tested on both the platforms.

3.2.2 NICAM-DC-MINI

NICAM-DC-MINI is a Fiber miniapp that is developed and maintained at RIKEN Advanced Institute for Computational Science. [6, 7] It is a subset of NICAM-DC application and contains the minimum computational procedures to run baroclinic wave test case Jablonowski [8], which is a well-known benchmark of atmospheric general circulation model reproducing the unsteasy baroclinic wave oscillation. As such, it retains the same computational workload characteristics as NICAM-DC, while enabling the performance evaluation in compact manner.

3.2.3 Graph500

Graph500 [9] is a data-intensive benchmark. It performs breadth-first searches in an weighted, undirected large graphs generated by scalable data generator based on a Kronecker graph. It consists of two kernels: The first kernel constructs an undirected graph used by the second kernel. The second kernel performs a breadth-first search of the graph from a randomly chosen source vertex in the graph. Both the kernels are timed and this benchmark uses the performance metric Traversed Edges Per second (TEPS) to compare the benchmark performance across multiple architectures, programming models and frameworks. TEPS is measured by benchmarking the second kernel. There is a validation phase at the end of each breadth-first search by second kernel to verify the correctness of the result. For large datasets, it is difficult to show that the resulting breadth first tree matches the reference result so the validation phase performs the soft checking of the result.

3.3 Results

3.3.1 NICAM-DC-MINI Results

We performed the weak scaling test of jablonowski for 1, 4 and 16 nodes both on TSUBAME and AWS EC2. Also, we used Score-P [10] which is Scalable Performance Measurement Architecture for Parallel Codes to trace our application performance. It is highly scalable and easy to use tool for profiling and tracing of HPC applications. When using 1 node for running the miniapp, all the c4.8xlarge instances i.e. ondemand dedicated (Fig. 3.2.2), shared (Fig. 1b) and spot instances (Fig. 1c) perform three times better than TSUBAME2.5 (Fig. 1d). We could not carry out the experiments for 16 nodes using dedicated and spot instances. Both the computation and communication times were faster in AWS as compared to TSUBAME2.5. Stall time corresponds to the time spent by the function MPI_Waitall in the graphs. For 16 nodes, c4.8xlarge ondemand shared instances performance was still 6 times better than TSUBAME2.5.

3.3.2 Intel MPI Benchmark (IMB) Results

To investigate the cause of the performance differences in executing NICAM-DC-MINI on TSUBAME2.5 and AWS EC2 c4.8xlarge instances, we thought to analyze the communication performance by using Intel MPI benchmark on both the platforms. These are shown in figure 2 with log scale on the y axis. TSUBAME2.5's network performance is consistently better than that of the cloud, therefore the fact that NICAM performs better on the cloud cannot be due the the the cloud's network capability.

4. Discussion

In NICAM-DC-MINI, the c4.8xlarge shared instances performed better for 1, 4 and 16 nodes than TSUBAME2.5, the dedicated and spot instances for 1 and 4 nodes. The performance exhibited by shared instances could be because of the reason that the instances were created on the same host. But, a detailed study that consists of up to 128 on both shared and dedicated instances is required to assess the processing capabilities offered by these cloud platforms.

5. Related Work

There have been several studies in the past that examines the feasibility of using public clouds for high performance computing. Amazon EC2, the leading public IaaS platform is oftenly used to carry out these studies. The cloud platforms have been evaluated with different focuses and goals. There was much focus on evaluating cloud platforms for tightly-coupled, MPI-based applications [11–13]. Also, some studies were conducted to assess the cloud environments for running scientific workloads and exhibit the performance-cost tradeoff [14]. Amazon EC2 introduced new instances called Cluster Compute instances (CCIs) in 2010 and claimed that these instances are suitable for compute-intensive HPC applications. This follows the studies on these new instances and DOE Magellan project [15, 16] which discussed that there is a mismatch between the requirements of HPC and the characteristics of cloud environment. With the aim of enabling HPC in cloud, Amazon came up with the new generation of compute-optimized c4 instances in 2015 [17]. The capabilities of new c4 instances have not been studied. Since the price of dedicated instances was dropped in 2013, it invites users with higher performance needs to use the dedicated instances.

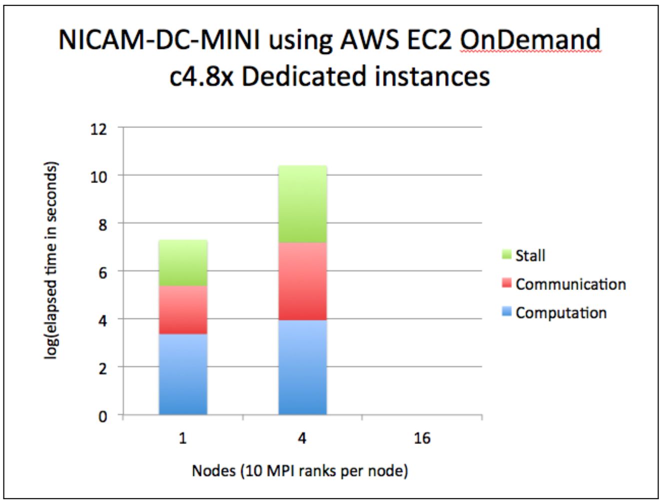
6. Summary and Future work

In this work, we conducted a comprehensive evaluation of the recently released Amazon EC2 c4 instances, which are intended for high performance computing. We assessed the feasibility of the new AWS c4 instances as replacements for traditional supercomputers by using a microbenchmark, HPC miniapp, and a traditional big data application. Our study reveals a picture considerably more positive for running HPC application on public clouds, as compared to the previous studies. As a future work, we would like to conduct these experiments at a bigger scale. We want to go upto 128 nodes for assessing the performance of cloud environments at a larger scale.

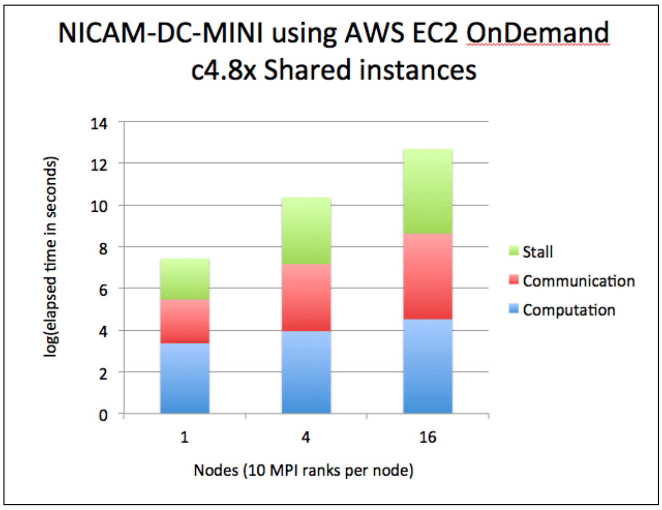
Acknowledgments This research was supported by JST, CREST (Research Area: Advanced Core Technologies for Big Data Integration).

References

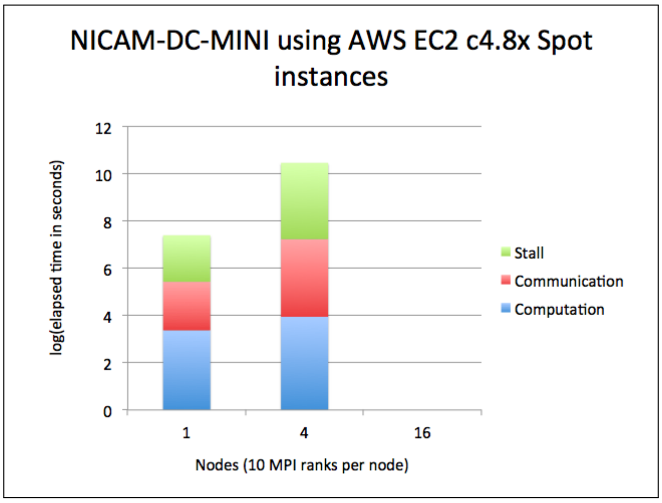
[1] Hoefler, T., Lumsdaine, A. and Dongarra, J.: Towards Efficient MapReduce Using MPI, *Proceedings of the 16th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Berlin, Heidelberg, Springer-Verlag, pp. 240–249 (online), DOI: 10.1007/978-3-642-03770-2_30



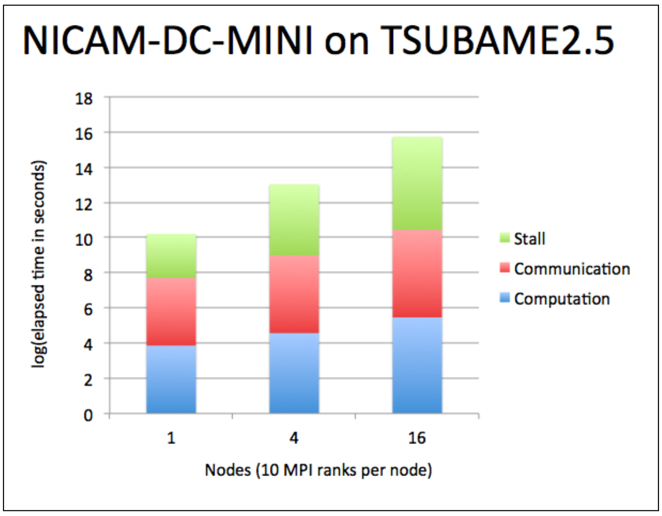
(a) Chart showing the performance breakdown of NICAM ondemand dedicated instances.



(b) Chart showing the performance breakdown of NICAM on Shared instances.



(c) Chart showing the performance breakdown of NICAM on spot instances.



(d) Chart showing the performance breakdown of NICAM on TSUBAME2.5 supercomputer.

Fig. 1 NICAM performance results on various cloud configurations and TSUBAME2.5

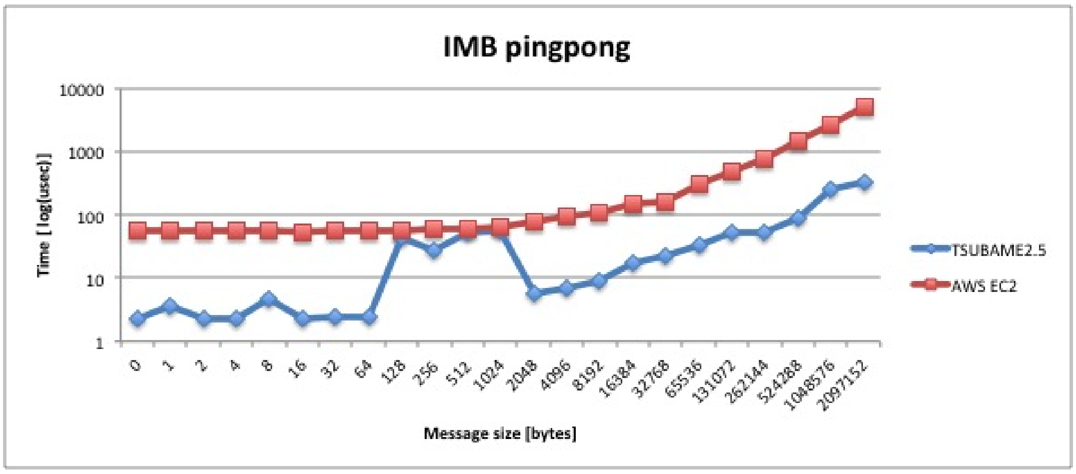


Fig. 2 Graph comparing the communication time of Intel MPI Benchmark (IMB) pingpong on TSUBAME2.5 vs. Amazon EC2 c4.8xlarge instance.

(2009).
[2] Luckow, A., Santcroos, M., Merzky, A., Weidner, O., Mantha, P. and Jha, S.: P*: A model of pilot-abstractions, *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pp. 1–10 (online), DOI: 10.1109/eScience.2012.6404423 (2012).
[3] Mantha, P. K., Luckow, A. and Jha, S.: Pilot-MapReduce: An Extensible and Flexible MapReduce Implementation for Distributed Data, *Proceedings of Third International Workshop on MapReduce and Its Applications Date*, MapReduce '12, New York, NY, USA, ACM, pp. 17–24 (online), DOI: 10.1145/2287016.2287020 (2012).
[4] Lustre: Lustre Parallel File System, `http://lustre.org/`.
[5] Intel Corp.: Intel MPI Benchmarks, `https://software.intel.com/en-us/software/products/20810`.
[6] Maruyama, N.: Miniapps for Enabling Architecture-Application Codesign for Exascale Supercomputing, *19th Workshop on Sustained Simulation Performance* (2014).
[7] Maruyama, N. et al.: Fiber Miniapp Suite, `http://fiber-miniapp.github.io/`.
[8] Jablonowski, C. and Williamson, D. L.: A baroclinic instability test case for atmospheric model dynamical cores, *Quarterly Journal of the Royal Meteorological Society*, Vol. 132, No. 621C, pp. 2943–2975 (online), DOI: 10.1256/qj.06.12 (2006).
[9] Graph500.org: The Graph500 List, `http://www.graph500.org/`.
[10] VI-HPS: Score-P, `http://www.vi-hps.org/projects/score-p/`.
[11] He, Q., Zhou, S., Kobler, B., Duffy, D. and McGlynn, T.: Case Study for Running HPC Applications in Public Clouds, *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, New York, NY, USA, ACM, pp. 395–401 (online), DOI: 10.1145/1851476.1851535 (2010).
[12] Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H. J. and Wright, N. J.: Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud, *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pp. 159–168 (online), DOI: 10.1109/CloudCom.2010.69 (2010).
[13] Napper, J. and Bientinesi, P.: Can Cloud Computing Reach the Top500?, *Proceedings of the Combined Workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop*, UCHPC-MAW '09, New York, NY, USA, ACM, pp. 17–20 (online), DOI: 10.1145/1531666.1531671 (2009).
[14] Iosup, A., Ostermann, S., Yigitbasi, M. N., Prodan, R., Fahringer, T. and Epema, D. H.: Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 6, pp. 931–945 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.66 (2011).
[15] Zhai, Y., Liu, M., Zhai, J., Ma, X. and Chen, W.: Cloud Versus In-house Cluster: Evaluating Amazon Cluster Compute Instances for Running MPI Applications, *State of the Practice Reports*, SC '11, New York, NY, USA, ACM, pp. 11:1–11:10 (online), DOI: 10.1145/2063348.2063363 (2011).
[16] Yelick, K., Coghlan, S., Draney, B. and Canon, R. S.: The Magellan Report on Cloud Computing for Science, Technical report, U.S. Department of Energy Office of Science Office of Advanced Scientific Computing Research (ASCR) (2011).
[17] Amazon web services: Now Available – New C4 Instances – Amazon Blog, `https://aws.amazon.com/jp/blogs/aws/now-available-new-c4-instances/`.