Towards Understanding HPC-Big Data Convergence Using Cloud Platforms

[Extended Abstract]

Shweta Salaria*, Kevin Brown, Hideyuki Jitsumoto, Satoshi Matsuoka
Tokyo Institute of Technology, Tokyo, Japan
*Email: salaria.s.aa@m.titech.ac.jp

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. Big Data refers to the diverse, complex, and massive data sets that can contain structured, semi-structured and unstructured data. 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. However, the access to these HPC centers is restricted and sometimes the demand exceeds the availability of resources.

Cloud platforms exhibit 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. These 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. However, the feasibility of running high performance computing applications on clouds has always been a concern mainly due to virtualization overheads and high-latency interconnection network. There have been many studies in the past that evaluated the cloud platforms for running HPC applications.

Amazon EC2, the leading infrastructure as a service (IaaS) provider announced the new generation of compute-optimized C4 instances [1] last year with the aim of targeting HPC audience. These instances 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.

Our work assesses the performance tradeoff when using HPC-like clouds versus dedicated HPC systems. For this evaluation, we compare the performance of Intel MPI benchmark [2], a traditional HPC scientific mini application, NICAM-DC-MINI [3], and a traditional Big Data benchmark Graph500 [4], on the TSUBAME-KFC supercomputer and Amazon AWS EC2 Cloud c4.8xlarge instances.

2 EVALUATION

We conducted performance comparison between Amazon EC2 c4.8xlarge instances and the supercomputer, TSUBAME-KFC. The Intel MPI Benchmark, NICAM-DC-MINI, a miniapp from Fiber and the Graph500 benchmark were evaluated on both the platforms.

2.1 Experimental Setup

2.1.1 Amazon EC2 C4

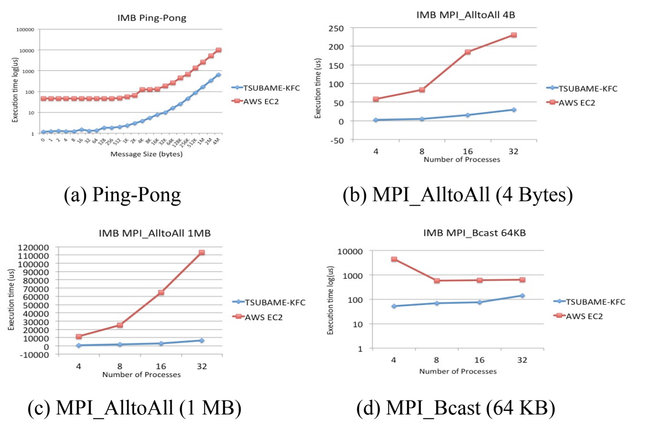
We used the latest generation of compute-optimized instances, c4.8xlarge instances in our cloud experiments. Each c4.8xlarge instance has Intel Xeon E5-2666 v3 (Haswell) 2.93 processors with 60GB memory and 36 vCPUs. These instances are interconnected with 10 Gigabit Ethernet and offer optimized Elastic Block Store (EBS). We built Lustre Parallel file system 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.

2.1.2 TSUBAME-KFC

The TSUBAME-KFC supercomputer was developed and maintained by Global Scientific Information and Computing Center (GSIC), Tokyo Institute of Technology, Tokyo, Japan. The system is composed of 42 compute nodes interconnected by 54 Gbps 4xFDR InfiniBand network. Each compute node is equipped with two Intel Xeon E5-2620 v2 processors (Ivy Bridge EP) and four NVIDIA Tesla K80 GPU boards (two GPUs on a board).

2.2 The Intel MPI Benchmark Results

We used Intel MPI benchmark to evaluate the communication performance of Amazon EC2 C4 instances and TSUBAME-KFC.



(a) Ping-Pong (b) MPI_AlltoAll (4 Bytes)

(c) MPI_AlltoAll (1 MB) (d) MPI_Bcast (64 KB)

Figure 1: Intel MPI Benchmark

The Ping-Pong results in Figure 1(a) shows significant performance gap on message latency between TSUBAME-KFC and AWS EC2. Figures 1(b) and 1(c) depict the results of collective operation, MPI_AlltoAll for small (4 Bytes) and large (1 MB) message sizes. For both sizes, TSUBAME-KFC outperforms AWS EC2 and the performance gap widens as the number of processes increases. We also show MPI_Bcast results for a moderate size (64KB) in Figure 1 (d) which also depicts the huge performance difference between TSUBAME-KFC and AWS EC2.

2.3 NICAM-DC-MINI Results

NICAM-DC-MINI is a Fiber miniapp that is developed and maintained by RIKEN Advanced Institute for Computational Science. It is a subset of NICAM-DC application and contains the minimum computational procedures to run baroclinic wave test case Jablonowski, which is a well-known benchmark of atmospheric general circulation model reproducing the unsteady baroclinic wave oscillation. It is a stencil code on the icosahedral grid where each node transfers the boundary data to the nearest-neighbour nodes.

Since NICAM-DC-MINI is a compute-intensive miniapp and AWS EC2 c4.8xlarge instances have more powerful processors than TSUBAME-KFC, it performs 2x faster on AWS EC2 than TSUBAME-KFC as shown in Figure 2. However, if we look at the communication performance, TSUBAME-KFC outperforms AWS EC2. The total execution time is dominated by the time spent on computation. Stall time corresponds to time spent in MPI_Waitall.

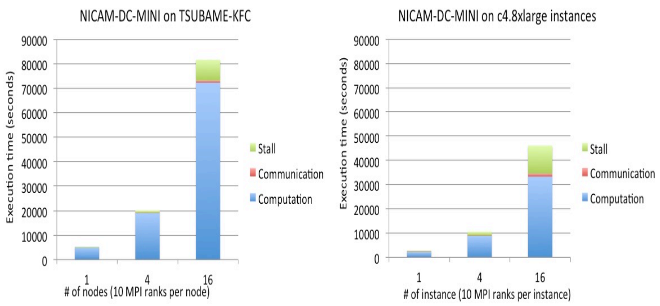


Figure 2: NICAM-DC-MINI on TSUBAME-KFC and AWS EC2

2.4 Graph 500 benchmark Results

Graph500 is a data-intensive benchmark. It performs breadth-first searches in a weighted, undirected large graphs generated by scalable data generator based on a Kronecker graph. For our experiments, we chose the following problem class:

Problem class	Scale	Edge factor	Approx. storage size in TB
Toy (level 10)	26	16	0.0172

Number of vertices = $2^{scale} = 2^{26}$ = 67,108,864
Number of edges = Edge factor * Number of vertices = $16*2^{26}$

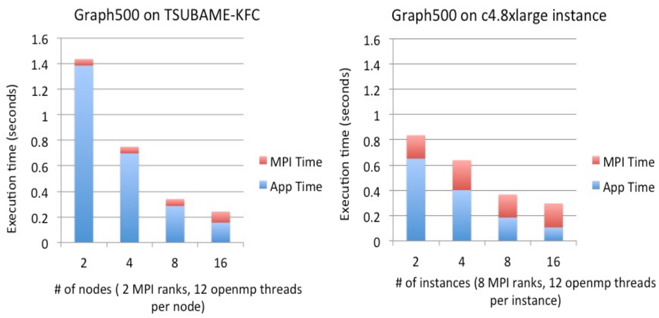


Figure 3: Graph500 benchmark on TSUBAME-KFC and AWS EC2

Figure 3 corresponds to results of replicated-csr implementation of Graph 500 benchmark. MPI time and App Time correspond to time spent on MPI calls and Non-MPI calls respectively during BFS phase of the benchmark. As expected, Non-MPI performance of AWS is better than that of TSUBAME-KFC. As the number of nodes increase, MPI time becomes more significant which drops the performance in AWS instances.

3 CONCLUSION & FUTURE WORK

We conducted experiments to evaluate the new generation of compute optimized AWS EC2 C4 instances by using a microbenchmark, HPC miniapp and a traditional big data application. These new instances with high-speed processors are favorable for compute-intensive HPC applications. However, high-latency introduced by 10-Gigabit Ethernet interconnection will make it difficult for communication-intensive HPC to scale and perform well.

In future, we plan to conduct extensive experimentation of the evaluated and new HPC Dwarfs and Big Ogres at larger scales with varying workloads. Also, we will establish the performance-cost tradeoff between C4 instances and TSUBAME-KFC.

4 ACKNOWLEDGEMENT

This research was partially supported by JST, CREST (Research Area: Advanced Core Technologies for Big Data Integration) and AWS Cloud Credits for Research Program.

5 REFERENCES

[1] Amazon EC2 c4 instances, https://aws.amazon.com/blogs/aws/new-c4-instances/

[2] Intel MPI Benchmarks, https://software.intel.com/en-us/articles/intel-mpi-benchmarks

[3] NICAM-DC-MINI, Fiber Miniapp Suite, https://github.com/fiber-miniapp/nicam-dc-mini

[4] Graph 500 Benchmark, http://www.graph500.org/specifications