Evaluation of HPC Applications on Cloud

Abhishek Gupta
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
gupta59@illinois.edu

Dejan Milojicic
HP Labs
Palo Alto, CA, USA
dejan.milojicic@hp.com

Abstract—**HPC applications are increasingly being used in academia and laboratories for scientific research and in industries for business and analytics. Cloud computing offers the benefits of virtualization, elasticity of resources and elimination of cluster setup cost and time to HPC applications users. However, poor network performance, performance variation and OS noise are some of the challenges for execution of HPC applications on Cloud. In this paper, we propose that Cloud can be viable platform for some HPC applications depending upon application characteristics such as communication volume and pattern and sensitivity to OS noise and scale. We present an evaluation of the performance and cost tradeoffs of HPC applications on a range of platforms varying from Cloud (with and without virtualization) to HPC-optimized cluster. Our results show that Cloud is viable platform for some applications, specifically, non communication-intensive applications such as embarrassingly parallel and tree-structured computations up to high processor count and for communication-intensive applications up to low processor count.**

Index Terms—**High Performance Computing, Clouds, Parallel Processing, Performance Evaluation**

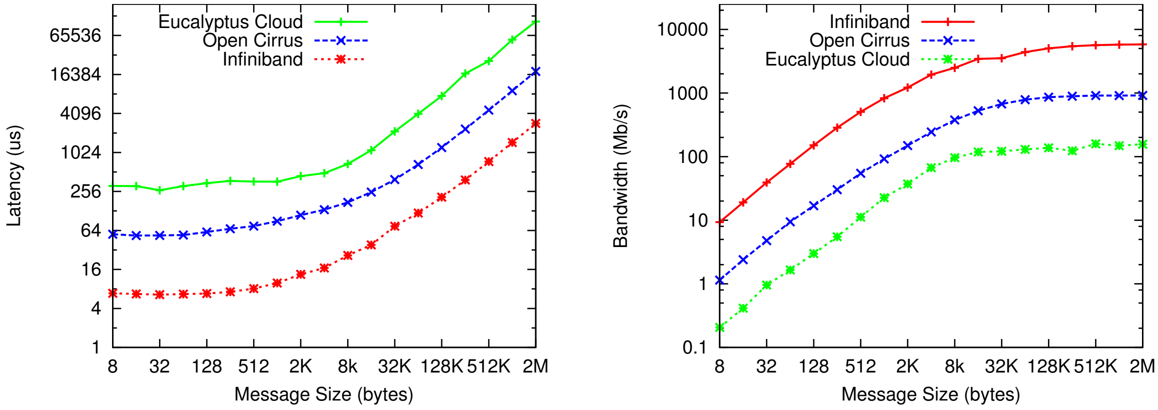
I. Introduction

High Performance Computing (HPC) applications are increasingly being used in academia and laboratories for scientific research and in industries for business and analytics. Clouds can act as a cost-effective and timely solution (e.g. substitute/addition when Supercomputers are heavily loaded such as in case of academic deadlines) to the needs of some academic and commercial HPC users since they do not involve cluster startup and maintenance costs and cluster creation time. In addition, Cloud provides elastic resources which results in elimination of risks caused by under-provisioning and avoidance of wastage of resources (including energy) resulting from underutilization of computing power in case of over-provisioning. Further, Cloud provides the benefits of virtualization to HPC community.

However, traditionally, Clouds have been designed for running business and web applications, whose resource requirements are different from HPC applications. Unlike web applications, HPC applications typically require low latency and high bandwidth inter-processor communication to achieve best performance. In case of Cloud, presence of commodity interconnect and effect of virtualization result in interconnect becoming a bottleneck for HPC applications. Figure 1 shows a comparison of three platforms using Converse [1] ping-pong benchmark and shows that network performance of Cloud is one to two orders of magnitude worse as compared to Infiniband, which is commonly used interconnect in Supercomputers. Furthermore, Supercomputers have operating systems and I/O subsystems specifically tailored to match HPC application demands. However, recent efforts towards HPC-optimized Clouds, such as Magellan [2] and Nebula [3] are promising signs for research community.

Past research [4]–[6] on HPC applications on Cloud has focused on performance as the metric and studies have been mostly on small scale with focus on Amazon EC2 [7]. The results from most studies have been pessimistic. Despite the benefits offered by Cloud computing, it still remains unclear whether Cloud can offer a viable alternate to Supercomputers for HPC applications.

In this paper, we take an alternate approach: we propose that Cloud can be suitable for *some* HPC applications and *not all* HPC applications. Also, for the same application, it can be better to run on Cloud for some range of processors whereas dedicated HPC system for other. We propose that HPC application characteristics specifically sensitivity to network parameters and performance requirements would advocate the platform where an HPC application can be most economically run. The research questions we address in this paper are the following: What are the performance-cost tradeoffs



(a) Latency

(b) Bandwidth

Fig. 1. Latency and Bandwidth vs. Message Size on Eucalyptus based Cloud, Open Cirrus Cloud and a cluster containing Infiniband network. Network performance on Eucalyptus Cloud is off by almost two orders of magnitude as compared to Infiniband, which is commonly used in Supercomputers

in using Cloud vs. high-end machines for HPC applications? Is the above applicable for all HPC applications? In particular, what is the impact of the communication characteristics of applications?

To this end, we evaluate the performance-cost tradeoffs of running HPC application using benchmarks (NPB [8] suite) and two real world applications up-to 256 cores. We consider three different platforms -

1) Taub which is a physical cluster using Infiniband as interconnect and scientific Linux as OS
2) Open Cirrus test-bed [9] which is a physical cluster with commodity interconnect and vanilla Linux
3) A Eucalyptus based Cloud [10] which uses KVM [11] for virtualization

The understanding of these tradeoffs can benefit system designer, application developers and application users alike. System designers can make platform design decisions according to the characteristics of applications of interest. Application users can benefit by choosing the optimal platform for their application. Last, application developers can concentrate on implementing application to leverage the platform which best suites it.

The remainder of the paper is organized as follows: related work is discussed in section II. Section III discusses our evaluation methodology. Section IV presents results using benchmarks and real applications. Finally, conclusions and future work are left for the final section.

II. Related Work

There have been several studies [4]–[6] to evaluate Amazon EC2 [7] for HPC applications using benchmarks. He et al. [12] extended previous research to three public cloud computing platforms and use a real application in addition to running classical benchmarks and compare the results with that from dedicated HPC systems. Ekanayake et al. [13] compare applications with different communication and computation complexities and observe that latency-sensitive applications experience higher performance degradation rather than bandwidth-sensitive applications. Jackson et al. [14] perform a comprehensive evaluation comparing conventional HPC platforms to Amazon EC2, using real applications representative of the workload at a typical supercomputing center and conclude that the interconnect on the EC2 cloud platform severely limits performance and causes significant variability.

Perhaps, the work by Napper et al. [15] is most similar to ours. They conclude that Clouds cannot compete Supercomputers based on the metric GFLOPS/$, since memory and network performance is insufficient to compete existing scalable HPC systems.

Our work is unique since we propose that Cloud would be suitable for *some* HPC applications and *not all*. We emphasize that application characteristics are crucial for the purpose of determining the optimal platform for its execution. Moreover, for the same application, it might be better to run on Cloud for some range of processors and on dedicated HPC system for other. Such tradeoffs have not been investigated in past studies [4]–[6], [15].

III. Evaluation Methodology

A. Experimental Test-bed

For our experiments, we chose a set of different platforms as representatives of different classes available to HPC community. Table I shows the configuration of the nodes in these platforms. For Eucalyptus Cloud, a node refers to a virtual machine and a core refers to a virtual core. To achieve best performance, virtual

Resource	Platform		
	Taub	Open Cirrus	Eucalyptus Cloud
Processors in a Node/VM	12×Intel Xeon X5650 @2.67 GHz	4×Intel Xeon E5450 @3.00 GHz	2×QEMU Virtual CPU @2.67 GHz
Memory	48 GB	48 GB	6 GB
Network	Voltaire QDR Infiniband	10 Gbps Ethernet internal, 1 Gbps Ethernet x-rack	Emulated Intel e1000 card, KVM hypervisor (1Gbps Physical Ethernet)
Operating System	Scientific Linux	Ubuntu 10.04	Ubuntu 10.04

TABLE I
EXPERIMENTAL TEST-BED

cores were configured not to share physical cores. For Open Cirrus and Eucalyptus Cloud, we setup a shared file system using NFS since most parallel programming systems such as MPI and CHARM++ [16] assume the presence of a shared file system for running applications.

B. Benchmarks and Applications

First, similar to previous work [4], [5], we evaluate these platforms using NAS Parallel Benchmarks (NPB) [8], which comprise a widely used set of programs designed to evaluate the performance of HPC systems. We ran the MPI [17] implementation of NPB 3.3 (NPB3.3-MPI). Unlike previous work, we perform experiments on a larger scale (up to 256 cores) to gain insights into their scalability. In addition, we present results from two real-world HPC applications:

- NAMD [18] - A highly scalable molecular dynamics application used ubiquitously on Supercomputers and
- NQueens - A backtracking search problem to place N queens on a N by N chessboard so that they do not attack each other. The NQueens implementation used here is representative of all-solution state space search computations which require heavy computational resources since they are commonly NP-hard.

We chose these applications since they differ in the nature and amount of communication performed, which is an essential requirement to validate our hypothesis. Moreover, NAMD is an iterative application whereas NQueens is a tree structured computation where communication happens only for load-balancing. For NAMD, we used ApoA1 benchmark (92K atoms) and for NQueens we ran a 19-queen instance.

IV. RESULTS

A. Performance

In this section, we discuss the performance results obtained on our test-bed for the applications described in section III. Figure 2 shows how these applications scale with the increase in the number of cores on different platforms. We can infer that the performance gap between Taub, Open Cirrus and Eucalyptus Cloud is larger in case of NAMD and LU as compared to EP and NQueens. In addition, the gap increases as we increase the number of cores. EP has very little communication and hence scales very well on all the platforms. Rest of the NPB benchmarks failed to scale beyond 64 cores. Both, LU and NAMD are communication intensive and stop scaling after 64 cores on Eucalyptus Cloud since communication becomes bottleneck (see figure 1). NQueens performs communication only for load-balancing dynamically generated work. The implementation used by us performs work-stealing for balancing load between parallel processes. NQueens scaled well despite being communication intensive since most of the communication is hidden by computation because we perform proactive work-stealing (load request is issued before the local queue of work items is empty).

B. Cost

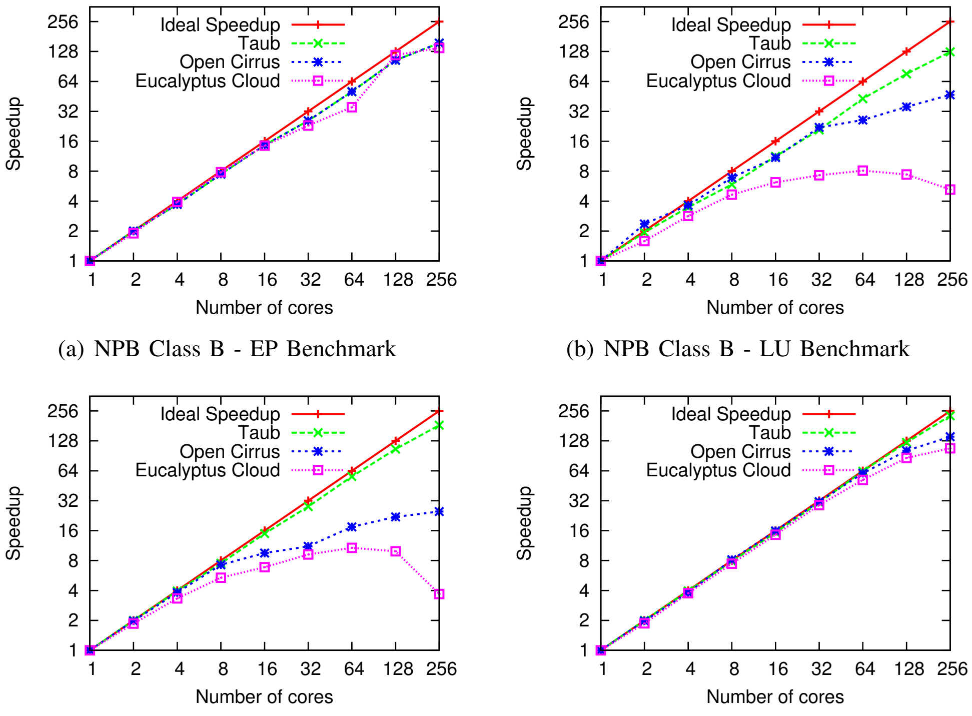
In this section, we evaluate the cost-performance tradeoff of running an HPC application on Cloud vs HPC-optimized platform, in our case, Taub. We use a simple charging based cost model to evaluate the cost of running an HPC application. For Cloud, we use a charging rate of $0.15 per core hour (Amazon EC2 pricing model for similar hardware). For Taub, we make a reasonable and conservative assumption for a charging rate of $1.00 per core hour [19], [20]. Note that our primary interest is to observe the shape of cost-curve and not the actual values. With these charging rates, the cost of executing an application on P processors becomes

$$\$0.15 \times P \times T_{\text{Eucalyptus Cloud}}$$

for Eucalyptus Cloud and

$$\$1.00 \times P \times T_{Taub}$$

for Taub. However, a direct comparison of the cost for running an application on Taub vs Eucalyptus Cloud for identical number of processors would be unfair since they can achieve different sequential performance and different speedups for same value of P. A fair comparison would be to study cost as a function of execution time.



(a) NPB Class B - EP Benchmark

(b) NPB Class B - LU Benchmark

(c) NAMD (Molecular Dynamics)

(d) NQueens (State Space Search)

Fig. 2. Speedup vs. Number of cores for different applications on different platforms. All the applications scale very well on Taub and moderately well on Open Cirrus. On Eucalyptus Cloud, NAMD and LU fail to scale after 64 cores whereas EP and NQueens scale well

Figure 3 shows this tradeoff for NAMD and NQueens. Due to space constraints, we do not show the plots for EP and LU since they follow similar patterns. Each point on the curves represents a measurement from execution on the labeled number of processors. For all cases, cost increases as execution time decreases because of non-linear speedup. An ideal speedup would result in a flat cost curve. We note that for NAMD, its better to run on Taub (except for larger execution time) whereas for NQueens, Eucalyptus Cloud is the better platform. This difference can be attributed to poor scaling of NAMD on Cloud. Hence, we note that depending upon application characteristics (such as communication sensitivity) and user's preferences (cost, performance) or constraints (e.g limited budget or upper bound on execution time) it might be better to run on one platform in some scenarios and on the other in some different scenarios. Moreover, for the same application, the optimal platform can vary depending upon the desired performance.

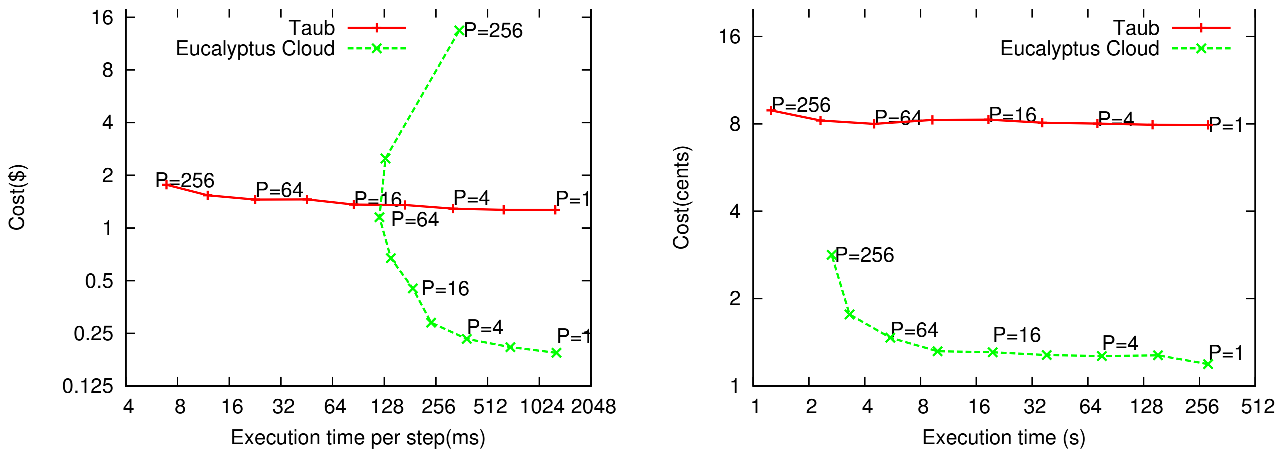
C. Discussion

We have seen that Cloud offers a viable platform for some HPC applications, such as EP and NQueens. However, for most applications, it is more cost-effective to use high-end machines except for very small core configurations. With the advent of HPC-optimized Clouds such as Magellan [2] and Nebula [3], the bottlenecks due to poor network performance should be minimized. Moreover, to achieve the best performance on Cloud, some application-specific performance re-tuning may be required since most HPC applications have been fine tuned to achieve best performance on Supercomputers. To validate this hypothesis, we experimented with the decomposition and work-stealing thresholds in NQueens and were able to improve performance by 20-50% for different core counts. Thus, with possible improvements to Cloud platform and application tuning, we believe that Cloud can compete with high-end machines for a wide range of HPC applications in future.

V. Conclusions and Future Work

In this paper, we studied the performance and cost tradeoffs of scaling HPC applications on a range of platforms from Cloud with virtualization, cluster with commodity interconnect to HPC-optimized cluster. We found that currently Clouds are more cost-effective for low communication-intensive applications such as embarrassingly parallel and tree-structured computations and HPC-optimized clusters are better for the rest.

Future work consists of HPC application characterization with the goal of automated determination of optimal



(a) NAMD

(b) NQueens

Fig. 3. Cost vs. Execution Time for two applications on Taub and Eucalyptus Cloud. We see two different patterns here - for NAMD, it is better to run on Taub (except for small scale) whereas for NQueens, Cloud is the optimal platform

platform for its execution. Also, we plan to extend the study to compare performance of the platforms used in this work with a Supercomputer and a cloud with better underlying physical interconnect.

Acknowledgments

The authors gratefully acknowledge the use of the parallel computing resource provided by the Computational Science and Engineering Program at the University of Illinois. The CSE computing resource, provided as part of the Taub cluster, is devoted to high performance computing in engineering and science.

References

[1] *The CONVERSE programming language manual*, Department of Computer Science,University of Illinois at Urbana-Champaign, Urbana, IL, 2006.

[2] "Magellan - Argonne's DoE Cloud Computing," http://magellan.alcf.anl.gov.

[3] "Nebula Cloud Computing Platform," http://nebula.nasa.gov.

[4] E. Walker, "Benchmarking Amazon EC2 for high-performance scientific computing," *LOGIN*, pp. 18–23, 2008.

[5] 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, Oct. 2008.

[6] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, pp. 931–945, June 2011.

[7] "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon.com/ec2.

[8] "NAS Parallel Benchmarks," http://www.nas.nasa.gov/Resources/Software/npb.html.

[9] A. I. Avetisyan, R. Campbell, I. Gupta, M. T. Heath, S. Y. Ko, G. R. Ganger, M. A. Kozuch, D. O'Hallaron, M. Kunze, T. T. Kwan, K. Lai, M. Lyons, D. S. Milojicic, H. Y. Lee, Y. C. Soh, N. K. Ming, J.-Y. Luke, and H. Namgoong, "Open Cirrus: A Global Cloud Computing Testbed," *Computer*, vol. 43, pp. 35–43, April 2010.

[10] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-source Cloud-computing System," in *Proceedings of Cloud Computing and Its Applications*, Oct. 2008.

[11] "kvm – Kernel-based Virtual Machine," Redhat, Inc., Tech. Rep., 2009.

[12] 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.

[13] J. Ekanayake, X. Qiu, T. Gunarathne, S. Beason, and G. C. Fox, *High Performance Parallel Computing with Clouds and Cloud Technologies*. CRC Press (Taylor and Francis), 07/2010 2010.

[14] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, "Performance analysis of high performance computing applications on the amazon web services cloud," in *CloudCom'10*, 2010.

[15] J. Napper and P. Bientinesi, "Can cloud computing reach the top500?" in *Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop*, ser. UCHPC-MAW '09. New York, NY, USA: ACM, 2009, pp. 17–20.

[16] L. Kalé, "The Chare Kernel parallel programming language and system," in *Proceedings of the International Conference on Parallel Processing*, vol. II, Aug. 1990, pp. 17–25.

[17] "Mpi: A message passing interface standard," in *M. P. I. Forum*, 1994.

[18] A. Bhatele, S. Kumar, C. Mei, J. C. Phillips, G. Zheng, and L. V. Kale, "Overcoming scaling challenges in biomolecular simulations across multiple platforms," in *Proceedings of IEEE International Parallel and Distributed Processing Symposium 2008*, April 2008, pp. 1–12.

[19] "ISC launches hosted HPC service," http://insidehpc.com/2008/04/01/isc-launches-hosted-hpc-service.

[20] "SGI Announces Cyclone Cloud Computing for Technical Applications," http://www.sgi.com/company_info/newsroom/press_releases/2010/february/cyclone.html.