An Experimental Study of Open-Source Cloud Platforms for Dust Storm Forecasting

Qunying Huang, Jizhe Xia, Chaowei Yang, Kai Liu, Jing Li, Zhipeng Gui
Dept. of Geography and GeoInformation Sciences
George Mason University, Fairfax, VA, 22030, USA
{qhuang1,jxia3,cyang3,kliu4,jlih,zgui}@gmu.edu

Mohammed Hassan, Songqing Chen
Dept. of Computer Science
George Mason University, Fairfax, VA, 22030, USA
{mhassanb, sqchen }@gmu.edu

ABSTRACT
Cloud computing is becoming a viable computing solution for scientific research and several open-source cloud solutions are available to support scientific studies. However, little has been done to systematically investigate the performance of these solutions in supporting scientific pursuits. Taking dust storm forecasting as an example, we test three popular open-source cloud solutions, namely Eucalyptus, OpenNebula, and CloudStack, on the same hardware and compare against a bare cluster. We find that: (1) compared to the bare cluster, a cloud has about 10% virtualization and management overhead when one virtual machine is used. Overhead increases when more virtual machines are used. Leveraging more virtual resources would not necessarily yield better performance. (2) For computing- and communication-intensive dust storm forecasting, the performance overhead is mainly due to virtualized network rather than virtualized computing resources when more than one virtual machine is involved. (3) Compared to Eucalyptus and CloudStack, OpenNebula provides better support for dust storm forecasting with relatively better performance. The results can provide some insights for scientific community in adopting these open-source cloud solutions.

Categories and Subject Descriptors
D.2.8 [**Software**]: Metrics–*Performance measures;* J.2 **[Physical Sciences and Engineering**]: *Earth and atmospheric sciences*

General Terms
Performance, Experimentation.

Keywords
Cluster Computing; Cloud Computing; Open-source Cloud Solutions; Infrasture as a Service; High performance computing

1. INTRODUCTION
Technology advancements, such as multi-core processors and fast networking communication, have driven computing platforms through several paradigms, such as high performance computing, grid computing [1], and most recently, cloud computing [2]. Today, most commercial computing service providers are moving from the traditional computing paradigm and embracing the cloud computing paradigm represented by Amazon EC2, Windows Azure, Rackspace, GoGrid, OpSource, FlexiScale, CloudSigma, Joyent Cloud, Google' Google App Engine, to name a few.

Scientific model simulations often demand accesses to a large amount of computing resources. Traditionally, these needs have been addressed by using high performance computing facilities, such as clusters, supercomputers, and grids. However, these facilities are often difficult to setup, maintain, and operate [3]. On the contrary, cloud computing platforms provide easy accesses to a large pool of computing (and storage) resources through a variety of interfaces similar in spirit to existing grid and HPC resource management and programming systems [4]. Tools and technologies are emerging that can transform an organization's existing computing infrastructure into a private or hybrid cloud [5]. Eucalyptus, CloudStack, and OpenNebula are the most popular open-source cloud solutions used as cloud orchestration to build such community cloud platforms. Eucalyptus enables users who are familiar with existing grid and HPC systems to explore new cloud computing functionalities while maintaining accesses to existing application development software and grid middleware [4]. OpenNebula is an open-source toolkit for building any type of cloud (private, public and hybrid) on both a local pool of resources and external Infrastructure as a Service (IaaS) clouds [5]. CloudStack is also open-source software that is designed to deploy and manage a large number of virtual machines as a highly available and scalable cloud computing platform [6].

Cloud computing facilities are often supported by virtualization techniques, which enable clouds to acquire or release computing resources on-demand and in a manner such that the loss of any component of the system will not cause system failures [7]. A wide range of virtualization solutions have been developed. KVM [8] and Xen [7] are two most popular ones. Eucalyptus and OpenNebula support both KVM and Xen for virtualization while CloudStack currently supports KVM and commercial Xen-based virtualization platform XenServer. Different virtualization techniques may introduce different overhead. Different cloud computing solutions may also introduce different overhead for monitoring and managing the virtualized resources [5]. While several previous studies have investigated the feasibility of cloud-based environments for scientific computing on the commercial cloud platform EC2 [3, 9, 10, 11, 12, 13, 14], there are few studies to assess the performance of these open-source cloud solutions in supporting computing- and communication-intensive model simulations.

In this paper, targeting three popular open-source cloud solutions, we aim to characterize their performance with a typical computing and communication intensive scientific application – dust storm forecasting [15]. With this application, we set to investigate the performance of different open-source cloud solutions and compare to a traditional HPC cluster. Our experiments show that (1) compared to a bare hardware cluster, a cloud with virtualized resources and networking has about 10% performance overhead

ACM SIGSPATIAL GIS '12, November 6-9, 2012. Redondo Beach, CA, USA

using one virtual machine (VM), and the overhead increases significantly when more VMs are used. (2) For computing- and communication-intensive dust storm forecasting, the performance degradation mainly comes from virtualized networks rather than virtualized computing resources. (3) Compared to Eucalyptus and CloudStack, OpenNebula provides better support for dust storm forecasting with higher performance on average.

We briefly introduce our experiment methodology in Section 2. Some preliminary results and analysis are presented in Section 3, with concluding remarks in Section 4.

2. EXPERIMENT METHODOLOGY

In this paper, two sets of experiments are conducted to benchmark against the bare cluster and the cloud computing middleware to transfer physical infrastructure to clouds, and to identify the bottleneck for clouds to support model simulations.

HPC Vs. Clouds: This set of experiments compares the virtual cluster provided by cloud platforms to a traditional cluster. The result of this experiment can show how good clouds are in supporting large scale scientific computing. Within this experiment, a different number of virtualized (1, 2 and 4 VMs) and non-virtualized computing resources (1 to 4 physical hosts) are compared to investigate the impact of virtualized computing power, and networking.

Cloud Middleware Comparison: This experiment tests the capability of different cloud middleware in supporting the computing- and communication-intensive applications with different numbers of VMs based on four physical machines and managed by Eucalyptus, OpenNebula, and CloudStack, respectively. The performance results would indicate the relative performance of these cloud solutions for scientific computing.

3. EXPERIMENTAL RESULTS

3.1 Experiment Setup

The workload in the experiment is a 3-hour dust storm forecasting over 2.3 x 3.5 degree geographic domain. The model execution time includes computing time and communication time.

Most of the tests are performed within a cluster (termed as Facility A), which includes 25 computing nodes running CentOS and all nodes are connected through local area networks (LANs with 1Gbps). Each node has 16GB memory and dual quad-core CPU of 2.33 GHz. The testing platforms include the bare cluster, OpenNebula, CloudStack and Eucalyptus. In the experiments, each platform has 5 nodes, with one serving as the master node, and the other four serving as the computing nodes.

For cloud computing resources, there are two types of VMs created based on the cluster: XLarge VM with 12 GB memory, and 8 virtual CPU cores; and large VM with 6 GB memory and 4 CPU cores.

3.2 HPC Vs. Clouds

Figures 1, 2 and 3 show the execution time (ET), computing time (CmpT) and communication time (CmmT) of the workload using different process numbers and computing resources. We can observe from Figures 1a, 1b and 1c that the bare cluster has the best performance. When only one cluster node and one XLarge VM is used, 1) the performance overhead is about 10%, 2) there is no significant difference between the four platforms when the process number is more than 32, 3) Eucalyptus has the worst performance with 2 to 32 process number, and 4) OpenNebula has the worst performance with one process.

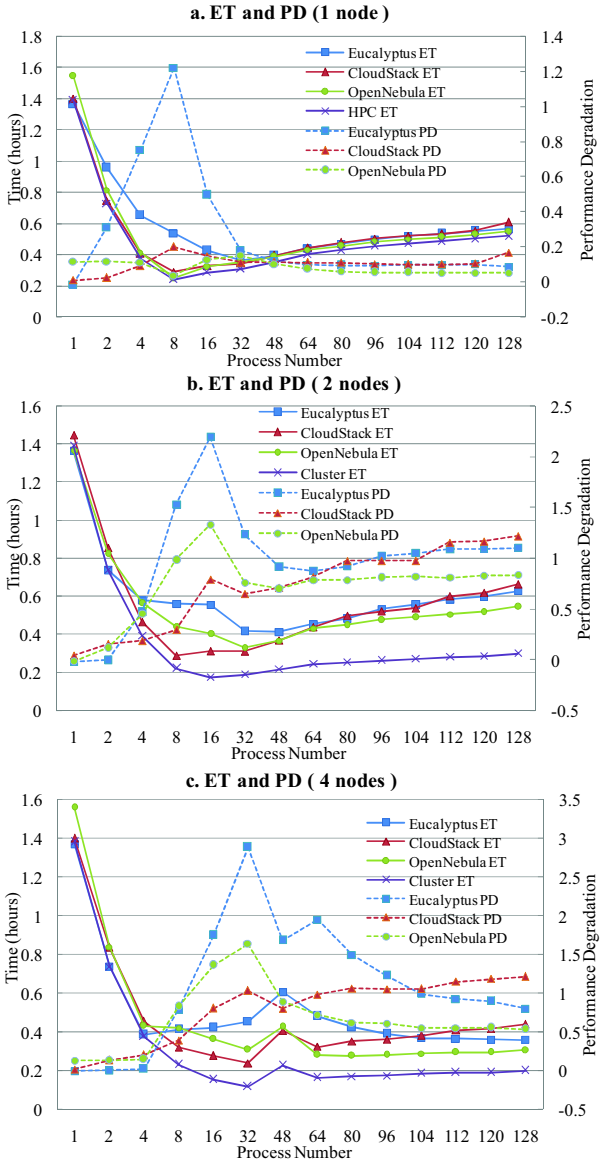

Figure 1. Execution time of the model simulation by different platforms, process numbers, and different computing nodes.

When two cluster nodes and two XLarge VMs or more computing resources are used, the performance degradations of the three cloud platforms are very significant and unpredictable. For example, OpenNebula, Eucalyptus, and CloudStack have a performance degradation (PD) of 120%, 139%, and 116%, respectively, on two Xlarge VMs on average.

Figures 2a, 2b and 2c show the computing time for the same model simulation with different numbers of computing resources and process numbers. It can be observed that the computing times on different platforms with different process numbers are similar to each other, particularly when the number of processes increases to 48 for one VM and when the number of processes increases to 80 for 2, 3, and 4 VMs. These results indicate that the virtualized computing resources by OpenNebula, Eucalyptus, and CloudStack

are very similar to each other. In another word, the overhead for virtualized computing capability is similar.

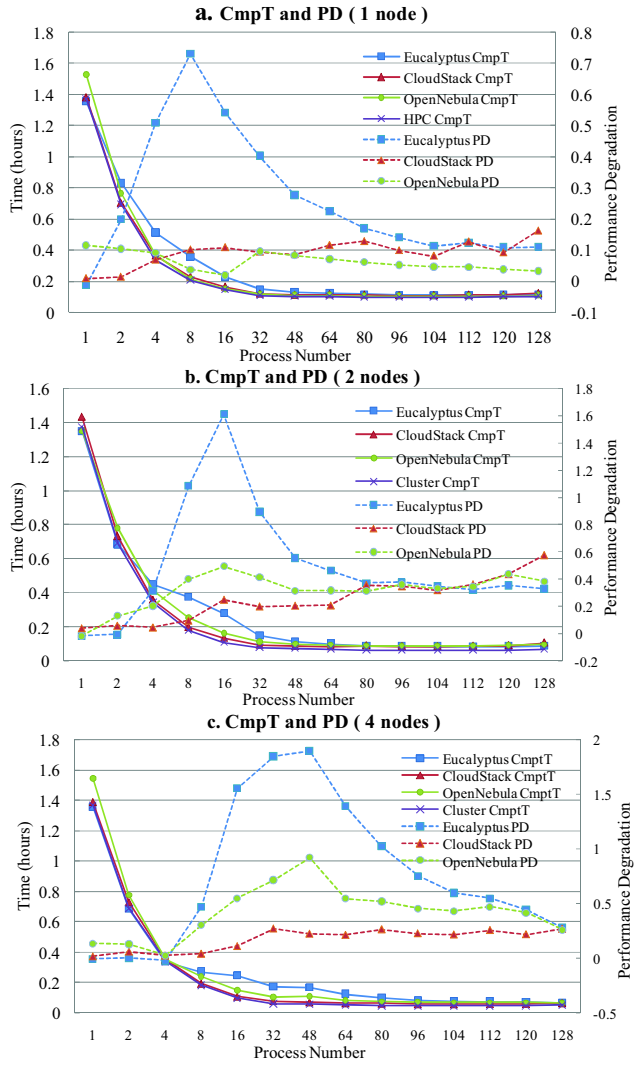

Figure 2. Computing time of the model simulation by different platforms and process numbers.

Figures 3a, 3b, and 3c show the communication overhead under different configurations. With one process, all the communication overhead is the same as expected. The bare cluster has the least communication overhead. The communication overhead increases with the increasing number of VMs involved for all three platforms.

When using one VM, the communication overhead compared to the bare cluster for the three platforms becomes pretty stable with 16 or more processes. Compared to the bare cluster, the average communication increase (CmmI) when 16 to 128 processes are used for OpenNebula, Eucalyptus, and ClockStack, is 9%, 11%, and 14%, respectively. When two VMs are used, the communication overhead increases significantly and such an increase becomes similar across all three platforms with 64 or more processes. The average increase is 95%, 122%, and 121%, respectively, averaging with 64 to 128 processes, and 120%, 139% and 115%, respectively, averaging from 1 to 128 process number, for OpenNebula, Eucalyputs, and CloudStack. With four VMs, the communication overhead for the same number of processes is 65%, 135%, and 140%, respectively, for these three platforms.

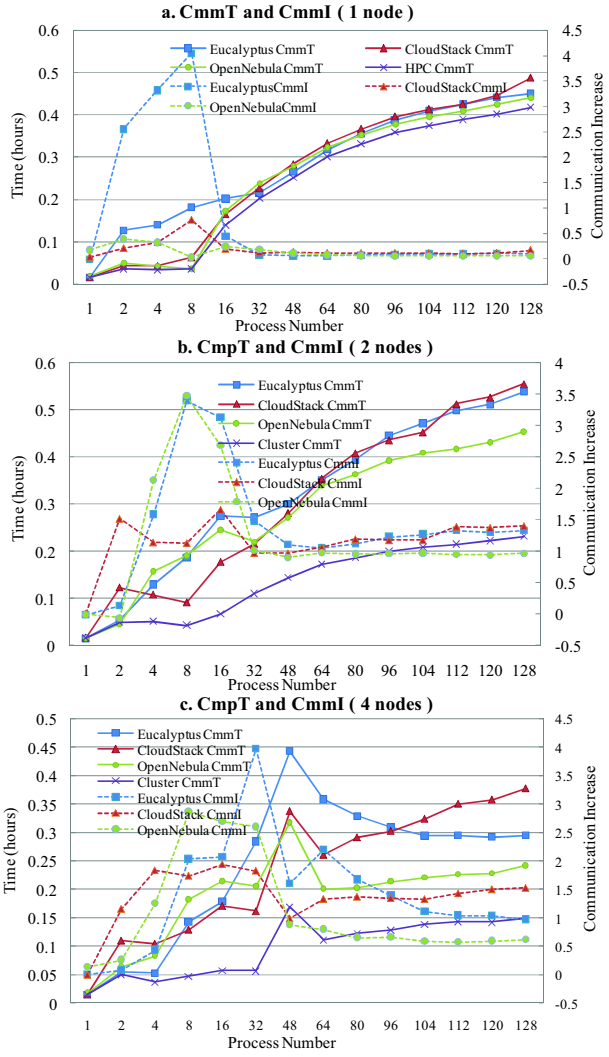

Figure 3. Communication time of the model simulation by different platforms and process numbers.

Based on the analysis above, Eucalyptus often leads to the highest communication overhead, followed by CloudStack. OpenNebula shows the best performance among these three.

This set of experiments shows that it is inevitable that a cloud built on any solution with virtualized resources and networking will degrade the performance of the original cluster for parallel applications with intensive communication involved between threads or processes.

3.3 Cloud Middleware Comparison

Figure 4 shows the performance of two large VMs on different platforms. CloudStack with 8 processes shows better performance than the other two platforms. However, it performs worse than the other two platforms with a different number of processes. With two Large VMs, OpenNebula generally has better performance than Eucalyptus. The communication overhead using two Large VMs is similar for OpenNebula and Eucalyptus because they both use Xen as hypervisor, and the communication for two large instances does not need to go through the network. Instead, it goes

through the host (Dom0) only. However, the computing time for Eucalyptus is higher than OpenNebula.

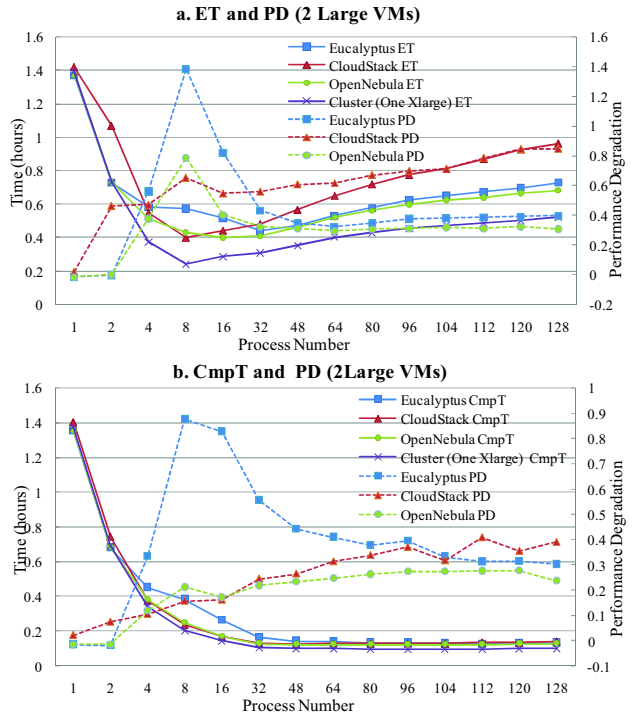

Figure 4. Execution and computing time of the workload by 2 Large VMs on one physical node

Based on the results of the Figures 1, 2, 3 and 4, it can be observed that OpenNebula performs better in supporting large scale scientific computing when multiple VMs are running on different physical resources. It often achieves the best performance. In the worst case, it performs better than one of the other two platforms. In addition, it provides the most stable performance for the model simulation. The performance lines with different process numbers often vary smoothly while Eucalyptus and CloudStack have fluctuations. Therefore, we can conclude that OpenNebula outperforms over Eucalyptus and CloudStack within our current testing environments.

4. CONCLUSION AND FUTURE WORK

In this paper, we have experimentally studied the performance of three open-source cloud solutions and compare against the traditional HPC cluster. Our preliminary results show that these solutions come with non-trivial virtualization overhead and the virtualized network is a particular bottleneck for data-intensive applications like dust storm forecasting. We plan to improve the performance of cloud solutions by considering residential computing with data affinity and optimizing the cloud computing middleware scheduling algorithms while dispatching the VMs on physical hosts.

5. Acknowledgement

This work is supported by NFS (CNS-0746649, CNS-1117300 and IIP-1160979), NASA (NNX12AF89G), and Microsoft Research.

6. REFERENCES

[1] Foster I. and Kesselman c. 1999. Concept and Architecture. in: The Grid: Blueprint for a New Computing Infrastructure, I. Foster and C.Kesselman Eds. CA: *Morgan Kaufmann*, pp. 37–63.

[2] Armbrust, M. et.al. A view of cloud computing, Communications *of the ACM*, vol.53, no.4, 50-58.

[3] Vecchiola C., Pandey, S., Buyya R. 2009. High-Performance Cloud Computing: A View of Scientific Applications. *Proc. Pervasive Systems, Algorithms, and Networks (ISPAN)*, 10th International Symposium on, 4 – 16.

[4] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman S., Youseff, L., Zagorodnov, D. 2009. The Eucalyptus Open-Source Cloud-Computing System. Proc. Cluster Computing and the Grid,. CCGRID '09. *9th IEEE/ACM International Symposium*, 124 – 131.

[5] Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I., 2009. Virtual Infrastructure Management in Private and Hybrid Clouds. Proc. *Internet Computing, IEEE*. 13(5), 14 – 22.

[6] CloudStack, 2012. http://CloudStack.org/

[7] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003. Xen and the art of virtualization. Proc. *The Nineteenth ACM Symposium on Operating Systems Principles*. Bolton Landing, NY, USA, October 19 - 22, ACM, New York, NY, pp. 164-177. DOI= http://doi.acm.org/10.1145/945445.945462.

[8] KVM (Kernel-based Virtual Machine), 2010. http://www.linuxkvm.org.

[9] Hazelhurst, S. 2008. Scientific computing using virtual high-performance computing: a case study using the Amazon elastic computing cloud. Proc. *The 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*. ACM, 94–103.

[10] Walker, E. Benchmarking Amazon EC2 for HP Scientific Computing. 2008. *Login*, 33(5),18-23.

[11] Ostermann. S., Iosup, A., Yigitbasi, M.N., Prodan, R., Fahringer, T., Epema, D.H.J. 2010. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol.34, No.4, 2010115-131, DOI: 10.1007/978-3-642-12636-9_9.

[12] Keahey, K. 2009. Cloud Computing for Science. Proc. *the 21st International Conference on Scientific and Statistical Database Management*. Springer-Verlag, 478.

[13] Schad, J., Dittrich, J., and Quiané-Ruiz, J.A. 2010. Runtime measurements in the cloud: observing, analyzing, and reducing variance. Proc. *VLDB* Endow. Vol.3, No.1-2, Sep.460-471.

[14] Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T., Epema, D.H.J. 2008. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transaction on Parallel and Distributed Systems*, Vol.22, No.6, 931-945.

[15] Xie, J., Yang C., Zhou B., Huang Q., 2010. High performance computing for the simulation of dust storms. *Computers, Environment, and Urban Systems*, 34(4), 278-290.