Peak2Cloud: Scientific Computing on the Cloud

JOSEPH ANTHONY C. HERMOCILLA, University of the Philippines Los Banos

Peak2Cloud (P2C) is an Openstack-based private cloud for scientific and high performance computing. First, we present how P2C was configured and tested. Then we describe vcluster, a tool for rapidly deploying message-passing clusters on P2C. Lastly, we analyze some benchmark results on the performance of P2C deployed virtual clusters.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Network operating systems

Additional Key Words and Phrases: cloud computing, high-performance computing

ACM Reference Format:
Joseph Anthony C. Hermocilla, 2010.Peak2Cloud: Scientific Computing on the Cloud. *ACM Trans. Comput. Syst.* 2014, 1, Article 2 (June 2014), 8 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Cloud computing has become a buzzword in today's modern computing although there is no agreed upon meaning of the term. [Mell and Grance 2011] of NIST published a definition that is widely quoted and used. The popularity of cloud computing mainly comes from its ability to provision additional resources on demand with minimal intervention from the provider. It leverages advances in virtualization and web service technologies. For example, an owner who observes a sudden increase in workload on his website can start another server machine (possibly virtual) almost instantaneously to accommodate the additional load. Extensive discussions of what cloud computing is can be found in the literature[Buyya et al. 2009][Qian et al. 2009] [Armbrust et al. 2010][Zhang et al. 2010].

A cloud can be deployed in several ways, depending on who can access the services it provides. *Private* clouds are operated for an organization. *Community* clouds are shared by several organizations to support a community with shared concerns. *Public* clouds are available to the public. Lastly, *hybrid* clouds are composition of two or more clouds [Mell and Grance 2011].

Cloud computing offers service models which include *Software-as-a-Service(SaaS), Platform-as-a-Service(PaaS), and Infrastructure-as-a-Service(IaaS)*. Most are familiar with SaaS as it is provides user functionality, Google Docs and Dropbox being notable examples. Developers on the other hand will be more acquainted with PaaS because they use APIs to develop applications in which Google App Engine is an example.IaaS allows the consumer to provision computing resources(hardware,network, storage) to

This work is supported by the Philippine Department of Science and Technology (DOST) Accelerated Science and Technology Human Resource Development Program (ASTHRDP).
Author's address: J. A. C. Hermocilla, Institute of Computer Science, University of the Philippines Los Banos

DOI:http://dx.doi.org/10.1145/0000000.0000000

run arbitrary software including operating systems[Mell and Grance 2011]. A popular IaaS public cloud is Amazon's Elastic Compute Cloud(EC2) and Simple Storage Service(S3). Most IaaS providers use proprietary technologies in their implementation. Lately, a number of open source frameworks have been released for deploying IaaS private clouds. This paper will focus on IaaS private clouds using Openstack described in the next section. Succeeding references to cloud will pertain to IaaS.

Clouds today are used mainly for hosting web sites and deploying online services (web applications). They provide instances with operating systems that can run web server software, scripting engine, and database management systems. Linux-Apache-MySQL-PHP (LAMP) stack is a typical configuration for an instance.

The success of this technology is enormous and people are still looking for uses beyond hosting. The science community is one group that is interested in leveraging the use of cloud. They are looking at the possibility of running entire scientific applications on the cloud. However, these applications are compute and communication intensive compared to web applications. This work explores this possibility through P2C.

2. OPENSTACK

Openstack is an open source software framework for deploying IaaS clouds [Sefraoui 2012]. This framework is widely supported by the community and has a large user base, including NASA. It provides a control interface that is compatible with Amazon's EC2, allowing easy transition for new users. Components of Openstack are developed separately providing modularity to the system. Below are the main components or modules of Openstack (the name in parenthesis represents the project name as referred to by the developers).

— *Object Store("Swift")* - provides storage
— *Image("Glance"* - provides a catalog and repository for virtual disk images
— *Compute("Nova")* - provides virtual servers upon demand
— *Dashboard("Horizon")* - provides a modular web-based user interface for all the Openstack services
— *Identity("Keystone")* - provides authentication and authorization for all Openstack services
— *Networking("Quantum")* - provides "network connectivity as a service"
— *Block Storage("Cinder")* - provides persistent block storage for guest VMs

Figure 1 shows the interaction of the major Openstack components. The dashboard represents the front-end to access compute, storage, and networking resources.

3. RELATED WORK

Studies have been published to evaluate the applicability of the cloud for scientific computing. The works described below focus on the performance evaluation of clouds for scientific applications.

[Walker 2008] showed that a performance gap between running HPC applications on a baremetal cluster and on an Amazon's EC2 provisioned cluster. They suggested that in order for cloud computing to be a viable alternative for HPC, providers must improve in the area of network interconnection.

[Evangelinos and Hill 2008] found that Amazon's EC2 may be a credible solution for on-demand and small-sized HPC applications. They supported this conclusion by running a low-order coupled atmosphere-ocean simulation on EC2.

[Ekanayake and Fox 2010] presented performance analysis of HPC applications on virtualized resources. They concluded that cloud techonologies work well for pleasingly- parallel problems. The main limitation of cloud technologies is the high

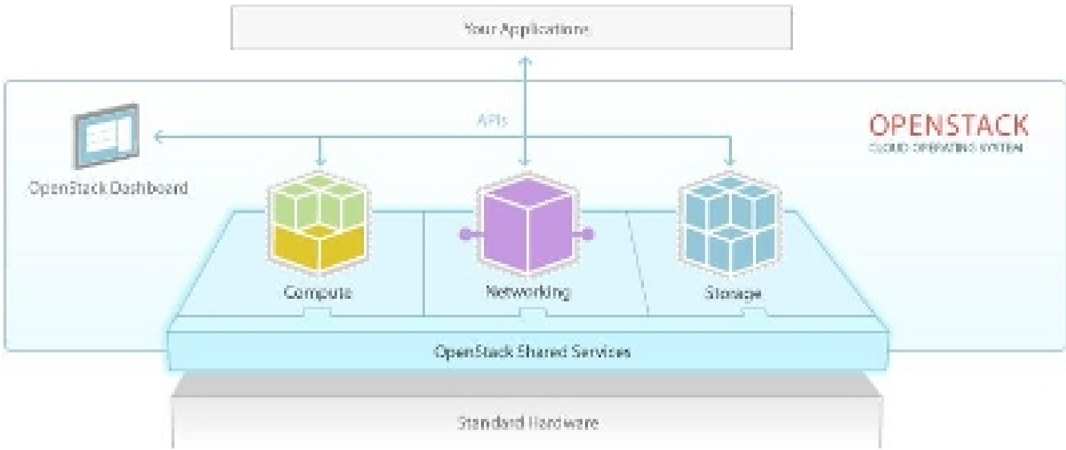

Fig. 1. Openstack at a glance.

overhead for applications with complex communication patterns, even with large data sets.

[Jackson et al. 2010] compared the performance of conventional HPC platforms to Amazon EC2. Their results showed that EC2 is six times slower than a typical mid-range Linux cluster, and twenty times slower than a modern HPC system. This is mainly because of the communication overhead. They also noted that variability in performance can be significant due to the shared nature of the cloud environment.

[Zhai et al. 2011] conducted a comprehensive comparison of the performance of a baremetal cluster(connected using Infiniband) and a cluster deployed using Amazon's Cluster Compute Instances (CCI). The study also revealed that running MPI applications in the cloud yielded more positive results compared to published results. They also highlight the flexibility and elasticity advantage of using cloud.

[Mauch et al. 2013] presented the High Performance Cloud Computing (HPC2) model. This model enables the provisioning of elastic virtual clusters which avoids the initial cost for physically owned hardware. They also presented a novel architecture for HPC IaaS clouds which support InfiniBand with QoS mechanisms since existing platforms still use Ethernet.

[Expsito et al. 2013] concluded that HPC application scalability depends mainly on the communication performance. Their study involved the use of Amazon's EC2 Cluster Compute Instances (CCI) platform targeted to HPC applications. This platform provides access to a high-speed network(10 Gigabit Ethernet).

[Ludescher et al. 2013] presented a novel code execution framework (CEF) to execute problem solving environment (PSE) source code in parallel on a cloud. The paper emphasized that the use of a public cloud can result to a magnitude of cost savings.

A recurring observation based on the above works is that primarily, the public cloud suffers from variability in performance probably because of multitenancy and limited network infrastructure.

4. METHODOLOGY

P2C is a combination of hardware, software, and network configuration. In this section we describe how these are combined to achieve the desired results.

Fig. 2. Hardware used in P2C.

Fig. 3. Switch used in P2C.

4.1. Hardware

P2C uses commercial-off-the-shelf(COTS) hardware. The cloud controller(1 unit) and compute nodes(2 units) are four-core Intel(R) Core(TM) i3-2000 3.10GHz CPU with 4GB RAM and 100GB disk. A 1GBps, 16-port Dell PowerConnect 2716 switch connects the controller and the nodes. Figure 2 and Figure 3 show the nodes and the switch respectively.

4.2. Software

P2C uses the Havana version of Openstack. Figure 4 shows what Openstack component is installed on each node. The host openstack-clc is the cloud controller and contains Keystone, Glance, Nova, Horizon, and Nova-Network. The compute nodes, hosts openstack-compute-01 and openstack-compute-02, has Nova and Nova-Network installed. Ubuntu Server 12.01 was used as the host operating system for the nodes. MPICH-2 was selected as the clustering software.

4.3. Network Topology

Figure 4 shows the network topology of P2C. Each node has two NICs connected to different networks. The first NIC is connected through a 100MBps Ethernet accesible

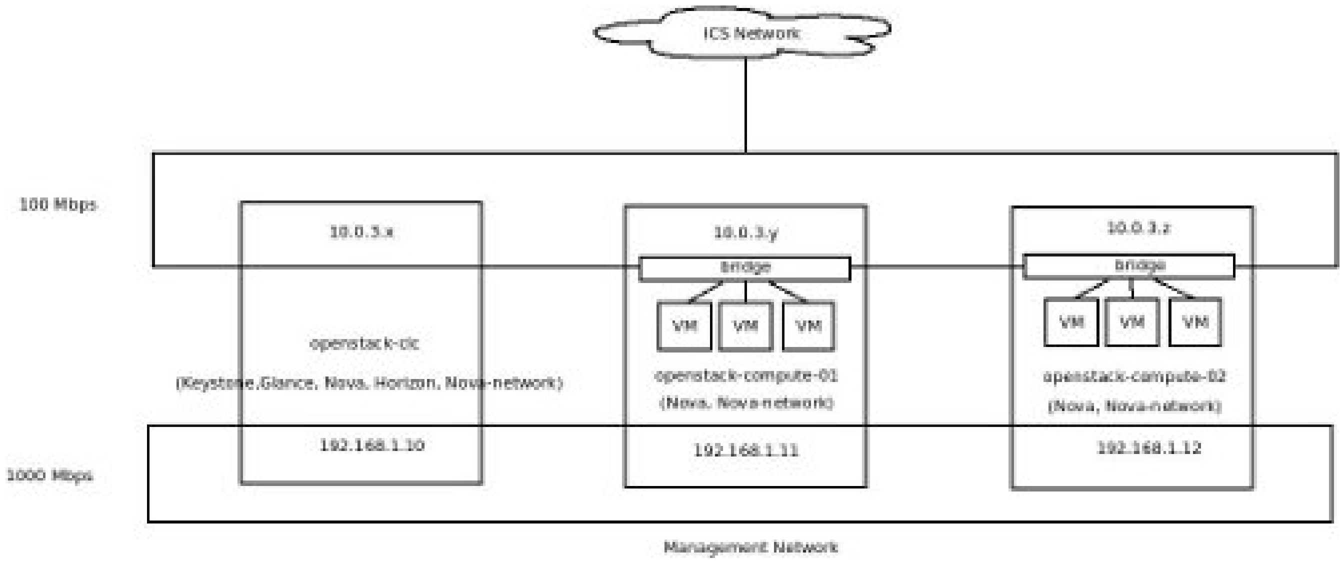

Fig. 4. Network topology.

through the institute's network (10.0.3.0/24 subnet). This setup allows staff to access the instances from their own computers through the bridged interface in the nodes. The second NIC is connected to a 1GBps Ethernet(192.168.1.0/24 subnet) used to manage the setup.

4.4. VM Images

Openstack supports different virtual machine monitors(VMMs) such as Xen and KVM. In P2C, we use KVM since it it the default that comes with Ubuntu. In order to start instances, virtual machine(VM) images must be selected. Ubuntu Server 12.04, Debian 7, and Windows XP are the supported images for the moment.

4.5. Running a Web application on P2C

In order to test P2C, a web appplication, called INSTANT, was deployed on an instance in P2C. It is an application to administer exam to students. The machine instance used has 4 VCPU, 1G RAM, and 10G disk. It uses the standard LAMP stack mentioned above.

4.6. vcluster

Deploying an MPI cluster manually in P2C is very similar to doing it using actual hardware and is quite cumbersome. After the host OS is installed, clustering packages such as MPICH2, OpenMPI, or LAM-MPI must be installed. Also, most cluster deployment use NFS to share the master filesystem to slave nodes. vcluster automates this process and allows users to deploy a working cluster on demand and terminate it after use.

4.7. Benchmarks

5. RESULTS AND DISCUSSION

Figure 8 shows the P2C dashboard.

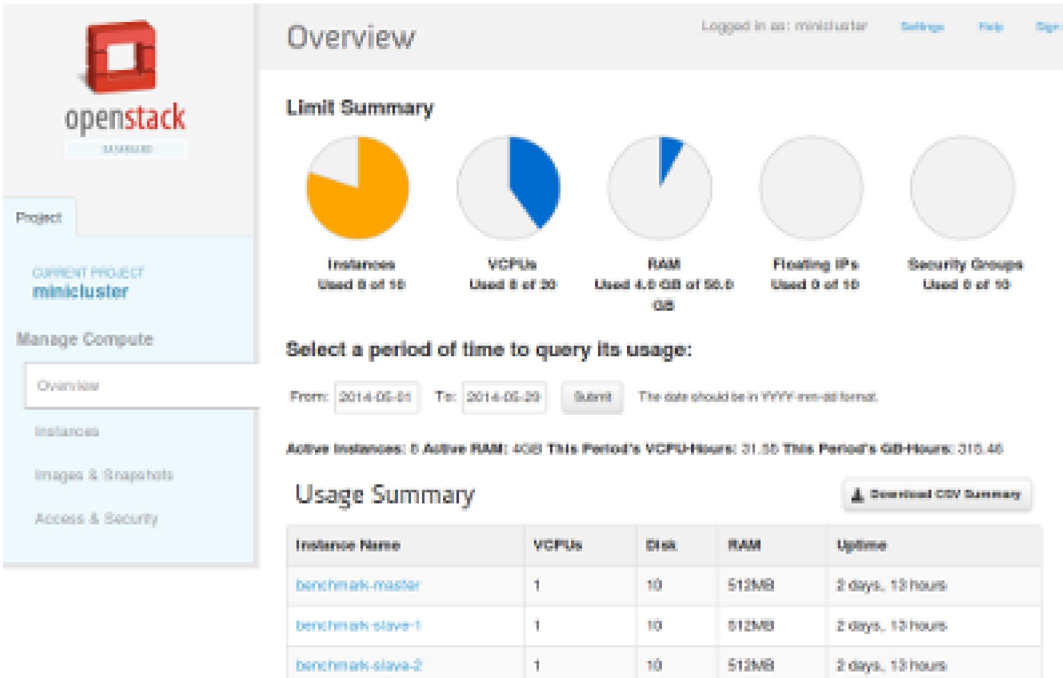

Fig. 5. P2C Dashboard.

Table I. AWSTATS Statistics

Unique visitors	225
Number of visits	395 (1.75 visits/visitor)
Pages	146, 032 (369.7 pages/visit)
Hits	158,237 (400.6 hits/visit)
Bandwidth	700.77 MB

5.1. INSTANT Results

The performance of the application was measured using awstats and is shown in Table I

Figure 5 shows how vcluster is started. The nodes in the cluster are shown in Figure 6. Testing the cluster deployed using vcluster is shown in Figure 7.

6. CONCLUSIONS

ACKNOWLEDGMENTS

The author would like to thank the Lord.

REFERENCES

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010), 5058. http://dl.acm.org/citation.cfm?id=1721672

Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 25, 6 (June 2009), 599–616. DOI:http://dx.doi.org/10.1016/j.future.2008.12.001

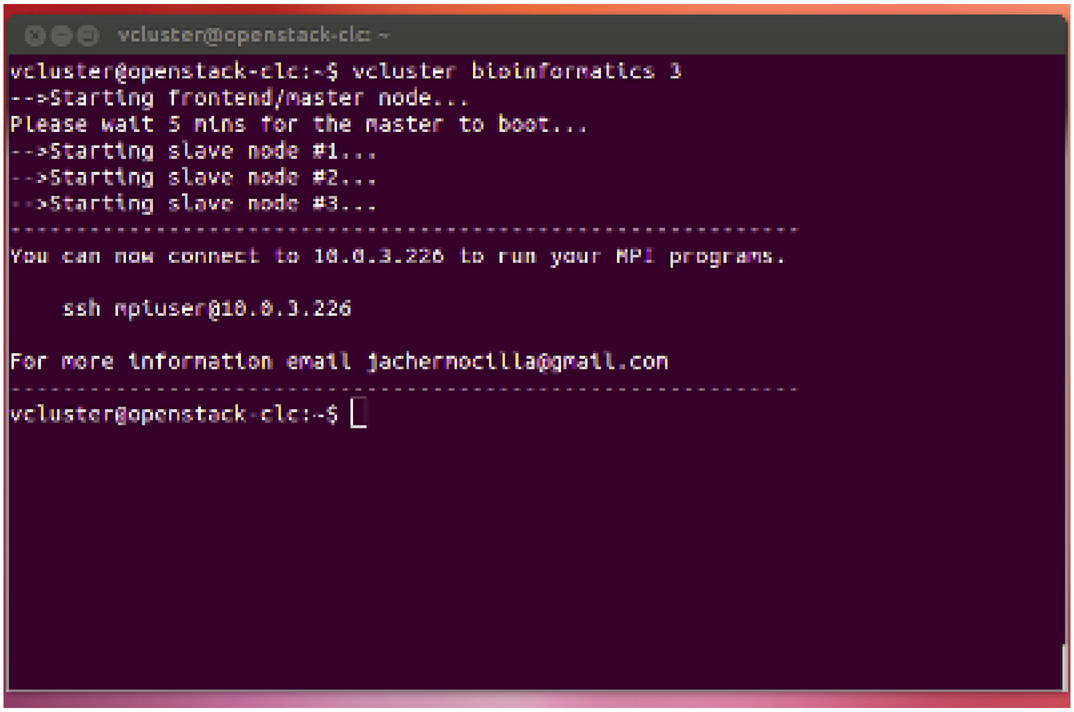

Fig. 6. Starting vcluster.

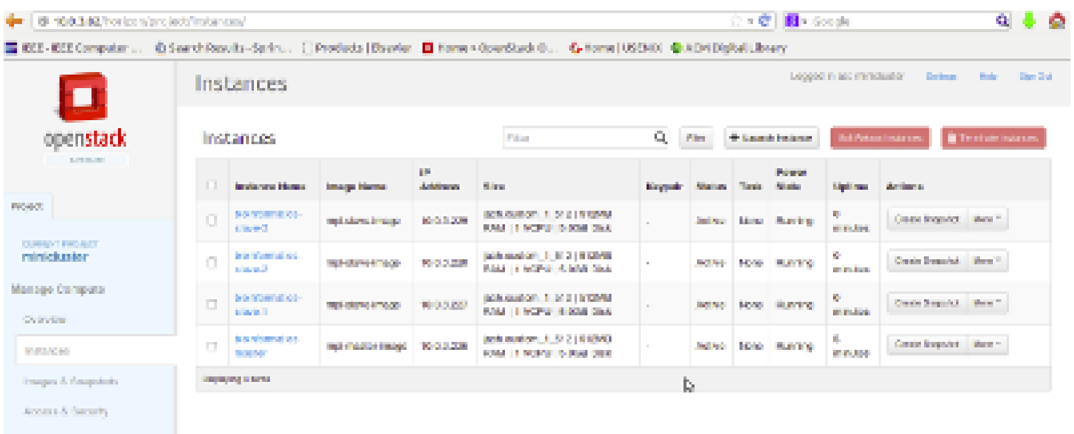

Fig. 7. vcluster nodes as shown in the dashboard.

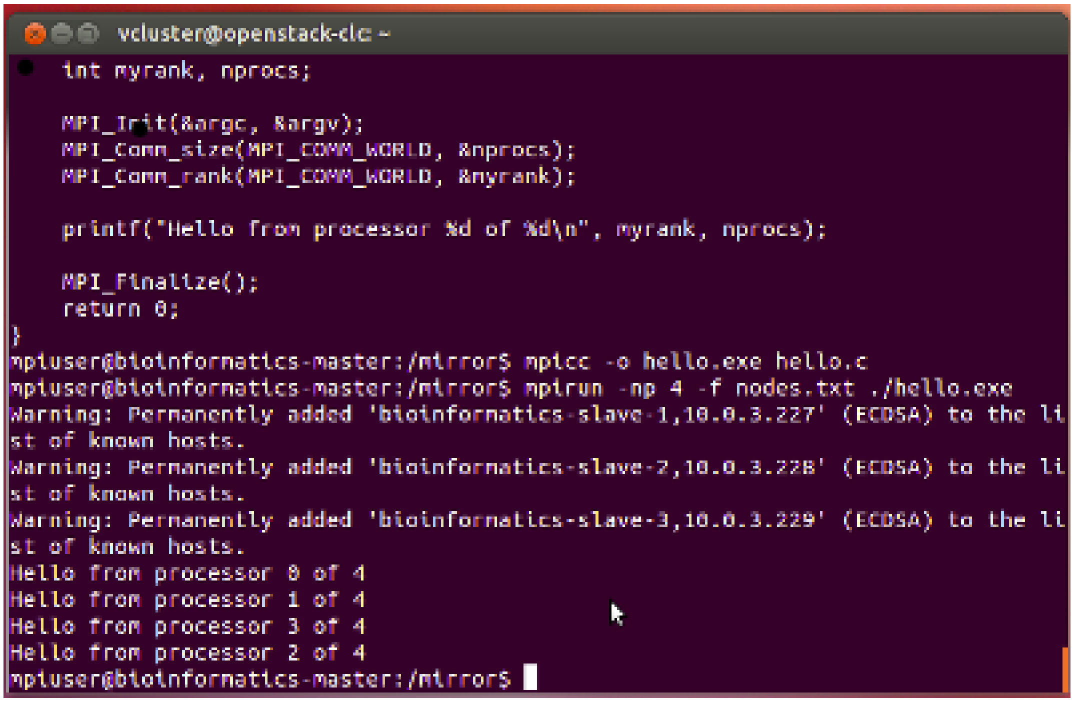

Fig. 8. Using the cluster deployed using vcluster.

Jaliya Ekanayake and Geoffrey Fox. 2010. High performance parallel computing with clouds and cloud technologies. In *Cloud Computing*. Springer, 2038. http://link.springer.com/chapter/10.1007/978-3-642-12636-9_2

Constantinos Evangelinos and C. Hill. 2008. Cloud computing for parallel scientific hpc applications: Feasibility of running coupled atmosphere-ocean climate models on amazons ec2. *ratio* 2, 2.40 (2008), 234. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.296.3779&rep=rep1&type=pdf

Roberto R. Expsito, Guillermo L. Taboada, Sabela Ramos, Juan Tourio, and Ramn Doallo. 2013. Performance analysis of HPC applications in the cloud. *Future Generation Computer Systems* 29, 1 (Jan. 2013), 218–229. DOI:http://dx.doi.org/10.1016/j.future.2012.06.009

Keith R. Jackson, Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf, Harvey J. Wasserman, and Nicholas J. Wright. 2010. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. IEEE, 159–168. DOI:http://dx.doi.org/10.1109/CloudCom.2010.69

Thomas Ludescher, Thomas Feilhauer, and Peter Brezany. 2013. Cloud-Based Code Execution Framework for scientific problem solving environments. *Journal of Cloud Computing: Advances, Systems and Applications* 2, 1 (2013), 11. http://www.journalofcloudcomputing.com/content/2/1/11

Viktor Mauch, Marcel Kunze, and Marius Hillenbrand. 2013. High performance cloud computing. *Including Special sections: High Performance Computing in the Cloud & Resource Discovery Mechanisms for P2P Systems* 29, 6 (Aug. 2013), 1408–1416. DOI:http://dx.doi.org/10.1016/j.future.2012.03.011

Peter Mell and Timothy Grance. 2011. The NIST definition of cloud computing (draft). *NIST special publication* 800, 145 (2011), 7. http://pre-developer.att.com/home/learn/enablingtechnologies/The_NIST_Definition_of_Cloud_Computing.pdf

Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. 2009. Cloud computing: an overview. In *Cloud Computing*. Springer, 626631. http://link.springer.com/chapter/10.1007/978-3-642-10665-1_63

Omar Sefraoui. 2012. OpenStack: Toward an Open-source Solution for Cloud Computing. *International Journal of Computer Applications* 55, 3 (Oct. 2012), 38–42.

Edward Walker. 2008. Benchmarking Amazon EC2 for high-performance scientific computing. *;Login* 33, 5 (Oct. 2008). https://www.usenix.org/legacy/publications/login/2008-10/openpdfs/walker.pdf

Yan Zhai, Mingliang Liu, Jidong Zhai, Xiaosong Ma, and Wenguang Chen. 2011. Cloud versus in-house cluster: evaluating Amazon cluster compute instances for running MPI applications. In *State of the Practice Reports*. ACM, 11. http://dl.acm.org/citation.cfm?id=2063363

Qi Zhang, Lu Cheng, and Raouf Boutaba. 2010. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications* 1, 1 (April 2010), 7–18. DOI:http://dx.doi.org/10.1007/s13174-010-0007-6

Received May 2014; revised June 2014; accepted June 2014