P2C: Towards Scientific Computing on Private Clouds

Joseph Anthony C. Hermocilla
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
jchermocilla@up.edu.ph

ABSTRACT

Peak-Two Cloud (P2C) is an Openstack-based private cloud on top of commodity hardware targeted for scientific and high-performance computing. We describe how it was configured and deployed then present use cases for education and actual production runs. We also introduce our own tools, vcluster and vhadoop, for rapid and on-demand deployment of MPI and Hadoop clusters, respectively, on P2C. Finally, we report some benchmarks results on clusters deployed using these tools.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Network Operating Systems*

General Terms

Management, Measurement, Performance

Keywords

Cloud Computing, High-performance Computing, Clusters

1. INTRODUCTION

Cloud computing has become a buzzword in today's modern computing although there is no agreed upon meaning of the term. NIST published a definition that is widely quoted and used[19]. 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 virtual server machine (called an instance) almost instantaneously to accommodate the additional load. Extensive discussions of what cloud computing is can be found in the literature[12][20] [11][24].

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 [19].

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 provides user functionality directly, 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 run arbitrary software including operating systems[19]. 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 work focuses on IaaS private clouds using Openstack described in the next section.

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 a cloud instance.

The success of this technology is enormous and people are still looking for uses beyond web application 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 and are often run on distributed systems and clusters. The setup of these requires highly specialized skills which a typical scientist will not have.

Thus, the main goal of P2C is to provide a cloud environment usable for scientific applications. It aims to harness commodity hardware used in teaching laboratories to maximize their utilization since most of the time they are used for simple programming purposes only. This approach of using teaching laboratory machines eliminates the upfront cost of purchasing dedicated hardware for clusters, which also saves on power consumption. In addition, dedicated clusters are often idle since they are not used as frequent as computers used in teaching laboratories and run out of date in terms of hardware specifications. P2C will allow scientists and students who would like to use a cluster to deploy one on demand then stop the cluster after running jobs. We also look into the performance of MPI clusters running on P2C, though no comparison to bare-metal clusters was attempted for the time being.

2. OPENSTACK

Openstack is an open source software framework for deploying IaaS clouds [21][8]. 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 network/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. Applications run on top of these resources.

3. RELATED WORK

Studies have been published to evaluate the applicability of public clouds for scientific computing. Walker 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[22]. Evangelinos and Hill 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[14]. Ekanayake and Fox 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 overhead for applications with complex communication patterns, even with large data sets[13]. Jackson et. al. 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[16]. Zhai et. al. 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[23]. Mauch et. al. 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[18]. Exposito et. al. 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)[15]. Ludescher et. al. 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[17].

A recurring observation based on the previous works mentioned is that public clouds suffer from variability in performance due to multitenancy and limited network infrastructure.

4. METHODOLOGY

P2C is a combination of hardware, software, and network configuration. The succeeding subsections describe each component.

4.1 Hardware

P2C uses commercial-off-the-shelf (COTS) hardware. The cloud controller (1 unit) and compute nodes (22 units) are four-core Intel(R) Core(TM) i3-2000 3.10GHz CPU with 4GB RAM and 100GB disk. All the nodes are connected through cascaded 1000Mbps switches.

4.2 Software

P2C uses the Havana version of Openstack[7]. Figure 2 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, have Nova and Nova-network installed. Ubuntu Server 12.01 LTS was used as the host operating system for the nodes. MPICH-2[5] was selected as the clustering software.

4.3 Network

Figure 2 shows the network topology of P2C. The cloud controller (openstack-clc) and two compute nodes (openstack-

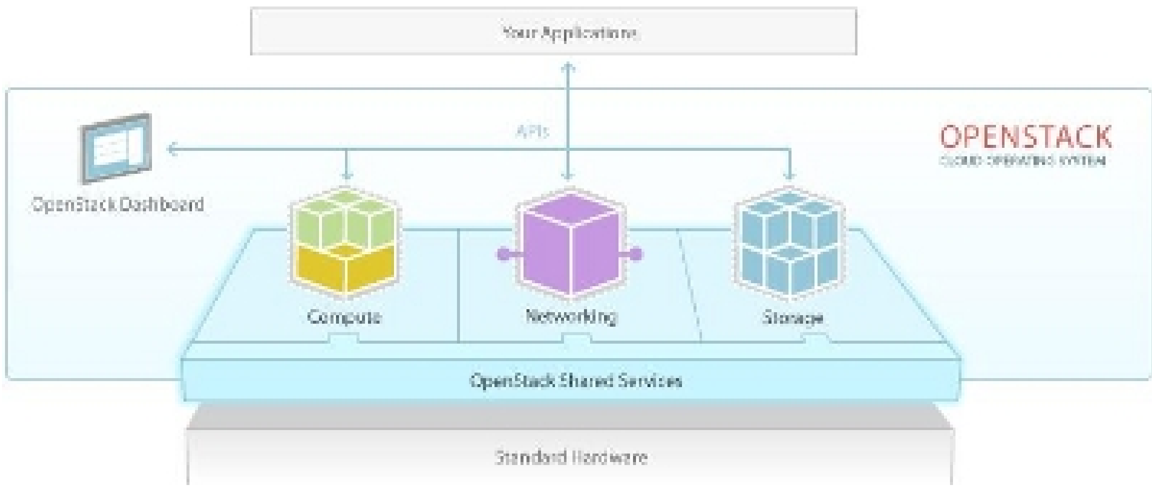

Figure 1: Openstack at a glance[21].

compute-01 and openstack-compute-02) have two network interface cards (NIC). One card is connected to the teaching laboratory network (10.0.4.0/24) and the other to the public network (10.0.3.0/24). The remaining 20 compute nodes have a single NIC connected to the teaching laboratory network. Instances are automatically assigned with IP addresses within the range of 192.168.10.0/24 by the DHCP server built into Openstack. Floating IPs (from 10.0.3.220 to 10.0.3.240 netmask 255.255.255.0) will allow public access to the instances.

4.4 VM Images and Flavors

VM images serve as templates for instances. Openstack supports different virtual machine monitors (VMMs), including Xen and KVM[4], capable of starting instances from images. In P2C, we use KVM since it it the default that comes with Ubuntu. In order to start instances, virtual machine images must be selected and assigned certain properties or flavors. A flavor includes the number of CPU cores, amount of RAM, and disk space to be allocated for an instance. Ubuntu Server 12.04, Debian 7, and Windows XP are the supported images at the moment by P2C.

4.5 Running a Web application on a P2C instance

In order to test P2C, a web appplication, called INSTANT[10], was deployed on an instance in P2C. It is an application to administer exam to students. It uses the standard LAMP stack mentioned above. The instance used for the test has a flavor with 4 VCPU, 1G RAM, and 10G disk. One hundred fifty students simultaneously accessed the application from the teaching laboratory network (10.0.4.y) for 1.5 hours (7:00-8:30pm). Settings for Apache, MySQL, and PHP were adjusted to provide optimal performance. This include increasing the maximum number of concurrent connections that can be accepted by Apache and MySQL.

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 the Network Filesystem(NFS) to share the master filesystem to slaves[9]. Our tool, vcluster, automates this process and allows users to deploy a working MPI cluster on demand and terminate it after use. Each node in the cluster has a flavor of 1 VCPU, 512MB RAM, and 10GB disk. This flavor can be adjusted as the need arises. In order for vcluster to work, two images were created, mpi-master-image and and mpi-slave-image which serve as base for master and slaves respectively. The preparation of these images is important since slave instances must communicate with the master instance. The trick is to use the capability of Openstack to inject startup scripts as an instance boots. Secure Shell(SSH)[6] is also used extensively for data transfer and remote command execution. To start, for example, a cluster named bioinformatics with 32 slaves, the user inputs *vcluster bioinformatics 32* on the command line. To stop the cluster, the user issues *stop-vcluster bioinformatics 32*.

4.7 vhadoop

Apache Hadoop is a software framework originally developed to process large data sets using MapReduce[1]. Our tool, vhadoop, automates the deployment of Hadoop clusters on P2C. Similar to the interface of vcluster, a user who would like deploy a Hadoop cluster named recommender with 16 slaves will input *vhadoop recommender 16* on the command line. Two VM images were also created for vhadoop, hadoop-master-image and hadoop-slave-image. To stop the Hadoop cluster a user issues *stop-vhadoop recommender 16* on the command line.

4.8 Benchmarking

In order to test the performance of P2C for scientific applications, the Intel MPI Benchmarks (IMB) were executed on a vcluster deployed cluster[3]. IMB consists of several tests, however we selected one test for single transfer (PingPong), parallel transfer (Sendrecv), and collective operations (bcast). The test cluster has one master and 9 slaves started using vcluster.

5. RESULTS AND DISCUSSION

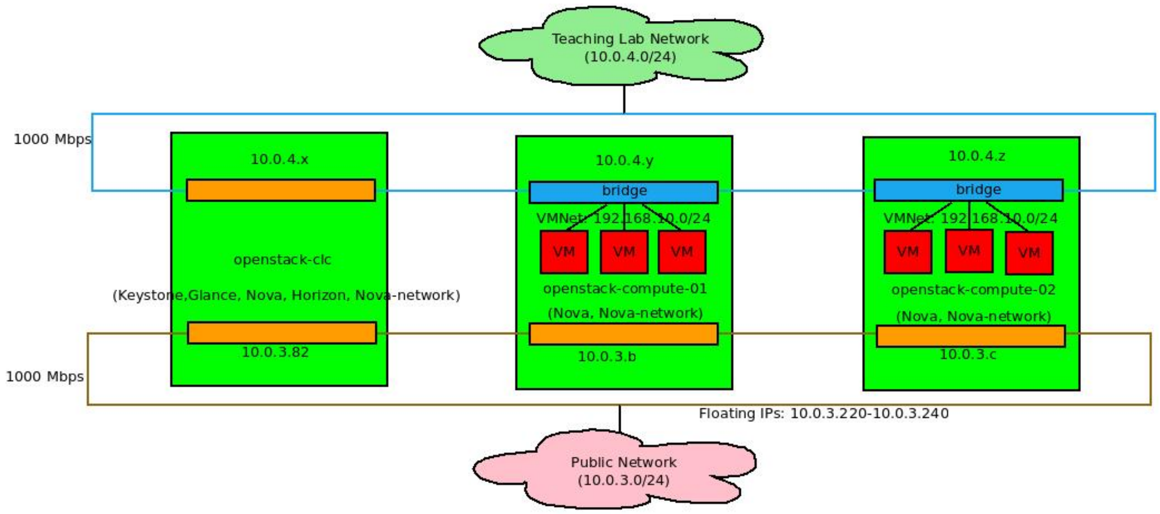

Figure 2: Network topology.

Table 1: AWStats Statistics for INSTANT running on an instance with 4VCPU, 1GB RAM, 10GB disk

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

Figure 3 shows the P2C dashboard. This interface allows users to manage the instances belonging to a project. The control panel on the left has options for starting and terminating instances and viewing statistics. As shown in the dashboard, P2C can currently support up to 88(22units*4cores) VCPUs, 86GB RAM, and 9.8TB of disk space. We plan to increase the capacity in the future by incorporating the remaining teaching laboratories to P2C.

The performance of INSTANT was measured using AWStats[2] and is shown in Table 1. During the deployment, the performance of the OS, Apache and MySQL was closely monitored. Apache was able to handle the number of concurrent connections but MySQL failed from time to time. This may be attributed to the limited memory allocated to the instance which is only 1GB.

Figure 4 shows how vcluster is started as described in the methodology. The figure shows the user started the bioinformatics cluster with 8 slaves. The dashboard also reflects the instances created by vcluster as shown in Figure 5. Figure 6 shows running an MPI program on the bioinformatics cluster. It can be observed that the hostname of the frontend machine is the same as the name specified when vcluster was started. This is a useful feature of vcluster because it makes it possible to have multiple clusters in P2C that are isolated from each other.

Figure 7 shows how to start a Hadoop cluster with 4 slaves. The user can connect to the front-end node and begin executing MapReduce jobs.

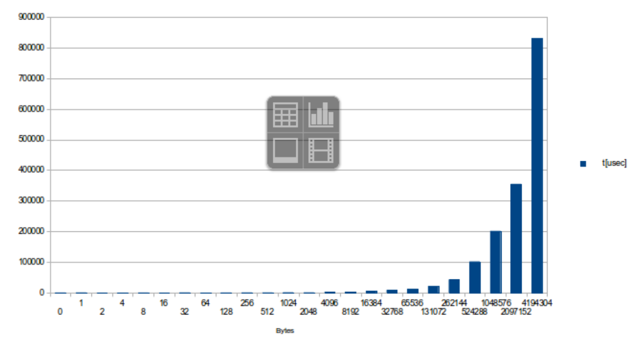

Figure 8: PingPong Latency results.

Figures 8 to 12 shows the results of running the IMB benchmarks. The general trend as shown in the figures indicate the increase in latency and throughput as the message size increases. However, there is a slight deviation in this trend on the PingPong Throughput test result in which certain fluctuations are observed.

6. CONCLUSION

We have shown how cloud computing can be used for scientific computing through the Openstack-based private cloud P2C. Using vcluster and vhadoop, end-users can rapidly deploy clusters on demand with zero upfront costs for dedicated hardware. The use of computers in teaching laboratories reduces power consumption which may otherwise be consumed by dedicated clusters. Initial benchmark results show some promise in terms of performance but a more comprehensive comparison with bare-metal clusters is still needed.

7. ACKOWLEDGEMENTS

This work is supported by the Philippine Department of Science and Technology (DOST) Accelerated Science and Technology Human Resource Development Program (ASTHRDP). The author would like to thank Dr. Eliezer A. Albacea, Dr.

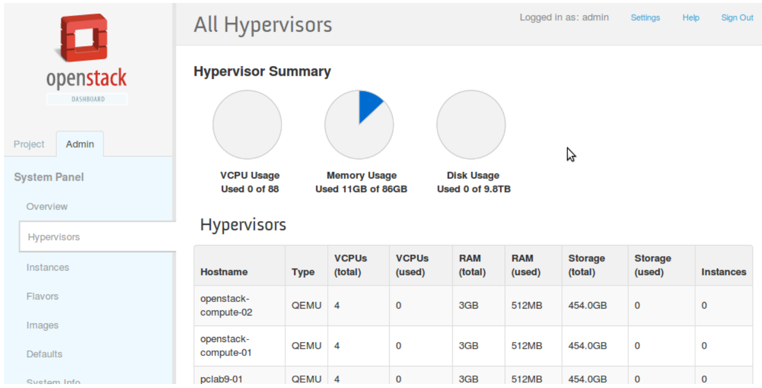

Figure 3: P2C Dashboard.

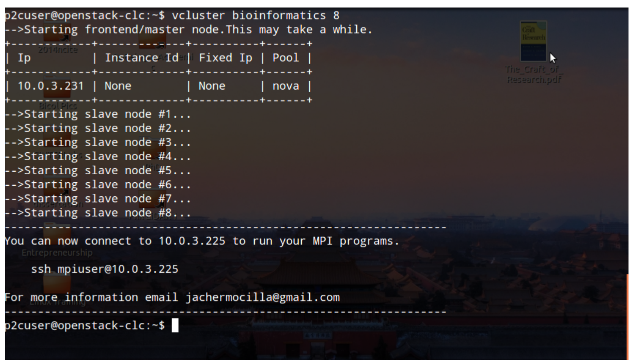

Figure 4: Starting vcluster.

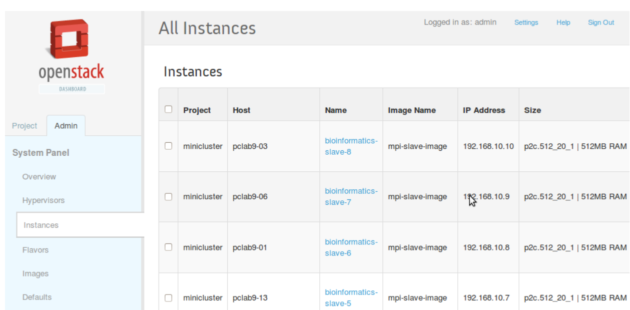

Figure 5: Instances automatically created for the cluster are shown in the dashboard.

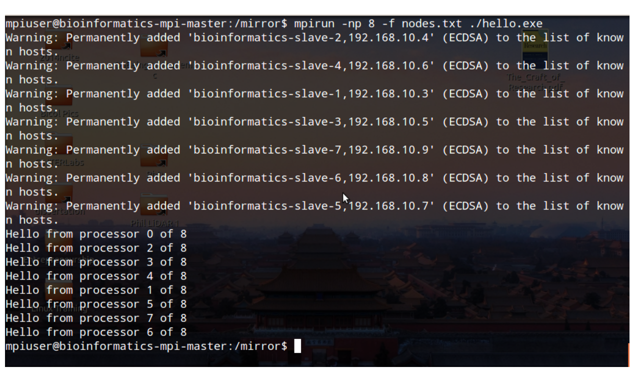

Figure 6: Running MPI programs on a cluster deployed using vcluster.

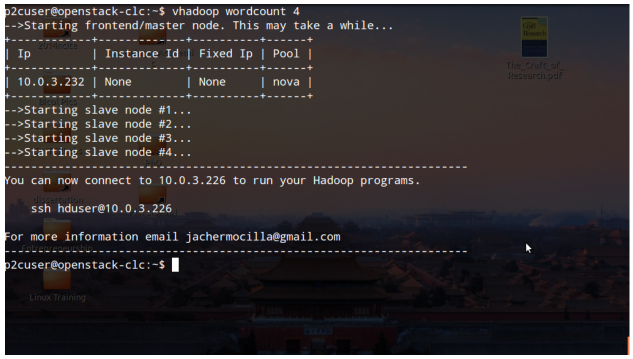

Figure 7: Starting vhadoop.

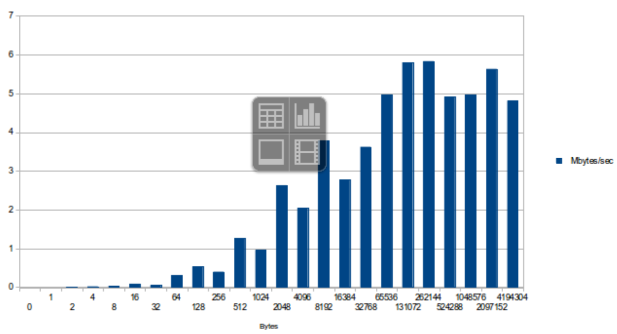

Figure 9: PingPong Throughput results.

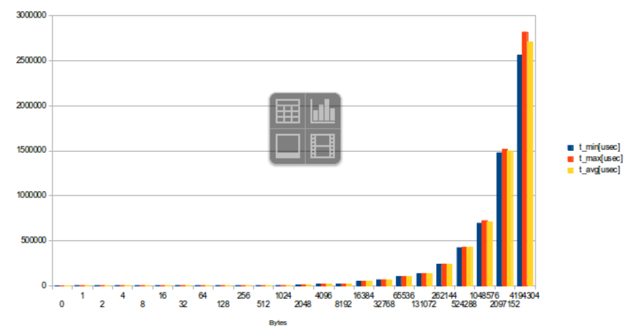

Figure 12: Bcast Latency results.

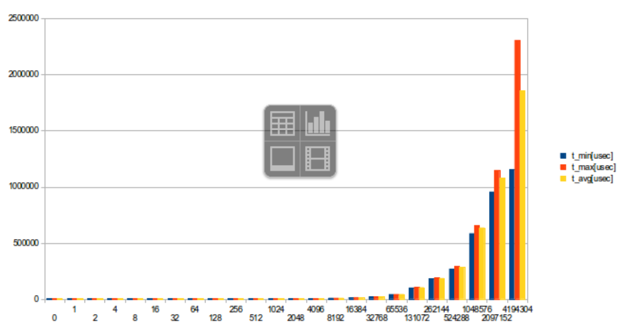

Figure 10: SendRecv Latency results.

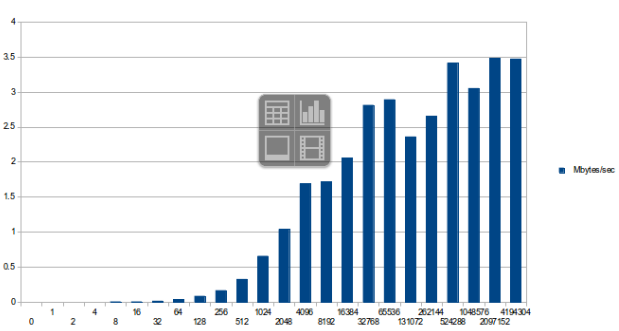

Figure 11: SendRecv Throughput results.

Vladimir Y. Mariano, Dr. Zita VJ. Albacea, Dr. Marlon N. Manalo, Danilo J. Mercado, Arian J. Jacildo, and the reviewers for their comments.

8. REFERENCES

[1] Apache Hadoop Website. `https://hadoop.apache.org/`.

[2] AWStats. `http://www.awstats.org/`.

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

[4] Kernel-based Virtual Machine. `http://www.linux-kvm.org/page/Main_Page`.

[5] MPICH2. `http://www.mcs.anl.gov/research/projects/mpich2/`.

[6] OpenSSH. `http://www.openssh.com/`.

[7] Openstack Havana Release. `http://www.openstack.org/software/havana/`.

[8] Openstack Website. `http://www.openstack.org`.

[9] Setting Up an MPICH2 Cluster in Ubuntu. `https://help.ubuntu.com/community/MpichCluster`.

[10] J. M. C. Abriol-Santos and K. J. Magno. Instant: Automated examination system. In *11th National Conference on Information Technology Education*, 2013.

[11] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica. A view of cloud computing. *Communications of the ACM*, 53(4):50âĂŞ58, 2010.

[12] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and

I. Brandic. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599–616, June 2009.

[13] J. Ekanayake and G. Fox. High performance parallel computing with clouds and cloud technologies. In *Cloud Computing*, page 20âĂŞ38. Springer, 2010.

[14] C. Evangelinos and C. Hill. Cloud computing for parallel scientific hpc applications: Feasibility of running coupled atmosphere-ocean climate models on amazonâĂŹs ec2. *ratio*, 2(2.40):2âĂŞ34, 2008.

[15] R. R. ExpÃşsito, G. L. Taboada, S. Ramos, J. TouriÃśo, and R. Doallo. Performance analysis of HPC applications in the cloud. *Future Generation Computer Systems*, 29(1):218–229, Jan. 2013.

[16] 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. pages 159–168. IEEE, Nov. 2010.

[17] T. Ludescher, T. Feilhauer, and P. Brezany. Cloud-based code execution framework for scientific problem solving environments. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1):11, 2013.

[18] V. Mauch, M. Kunze, and M. Hillenbrand. High performance cloud computing. *Including Special sections: High Performance Computing in the Cloud & Resource Discovery Mechanisms for P2P Systems*, 29(6):1408–1416, Aug. 2013.

[19] P. Mell and T. Grance. The NIST definition of cloud computing (draft). *NIST special publication*, 800(145):7, 2011.

[20] L. Qian, Z. Luo, Y. Du, and L. Guo. Cloud computing: an overview. In *Cloud Computing*, page 626âĂŞ631. Springer, 2009.

[21] O. Sefraoui. OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications*, 55(3):38–42, Oct. 2012.

[22] E. Walker. Benchmarking amazon EC2 for high-performance scientific computing. *;Login*, 33(5), Oct. 2008.

[23] Y. Zhai, M. Liu, J. Zhai, X. Ma, and W. Chen. Cloud versus in-house cluster: evaluating amazon cluster compute instances for running MPI applications. In *State of the Practice Reports*, page 11. ACM, 2011.

[24] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, Apr. 2010.