PRIVATE CLOUD INITIATIVES USING BIOINFORMATICS RESOURCES AND APPLICATIONS FACILITY (BRAF)

Bhagyashri Pathak, Rajesh SR, Sankalp Jain, Amit Saxena, Rashmi Mahajan and Rajendra Joshi*

Bioinformatics Group, Centre for Development of Advanced Computing
Pune, INDIA - 411007
rajendra@cdac.in, http://bioinfo-portal.cdac.in

ABSTRACT

The bioinformatics research community has a demand of enormous compute resources to run bioinformatics tools. Next generation sequencing technologies have further increased the overall demand for computational analysis. The traditional Cluster and Grid computing are having their own complexities to program and use while there is a silver-line in cloud for on-demand high-performance infrastructures with the advent of cloud computing era and its advantages. We have adopted the technology so that it can prove its mandate with more benefits to the community. We are able to bring out a private cloud using BRAF which is a high end cluster facility dedicated for bioinformatics. Open source equivalents to prominent commercially available solutions are used as the cloud middle-ware stack. In this paper we will summarize our implementation of a virtualized private cloud environment using Eucalyptus.

KEYWORDS

Cloud computing, Private cloud, High performance computing, Scientific computing.

1. INTRODUCTION

Cloud computing is not a new technology, it was always there but after being defined as a new technology it has been noticed considerably. The big cloud commonly used in diagrammatic representation showing enormous power and resources on Internet is today's cloud. The virtualization started with the need of having many Operating System (OS) on a same physical hardware without rebooting the machine. VMware [1] and Xen [2] made it possible to cross the barrier of single OS on a physical machine without rebooting the physical machine. Service oriented architecture (SOA) provided a convenient way by exposing software as a service using web technologies. Cloud computing was already used by us as a web based email service like Hotmail [3], Gmail [4]; social networking like Facebook [5]; Office solutions like Google Docs [6] and many more. Cloud services are broadly classified under three categories viz. IaaS, PaaS, SaaS which provides Infrastructure as a service, Platform as a service and Software as a service over Internet. Open source cloud building tools are available to provide a better management of computation power in an efficient way. It saves energy by utilizing the servers in its full capacity for multiple purposes. The whole stack of network switches, firewall and servers can be virtualized using virtualization technologies. SOA and Web 2.0 tools help a service provider to provide the required resources to the client. BRAF [7] acts as a service provider to the research community consisting of scientists and research students. Cloud computing technologies can

DOI : 10.5121/ijccsa.2012.2602

prove as an enabler for efficient and user friendly way of accessing resources like compute and storage for advanced research projects. The private cloud can be a start which can meet public cloud in future and become source for infinite resources appearance. During the design of our cloud we have explored open source solutions like Eucalyptus [8] and OpenNebula [9]. We took advantage of our existing grid computing expertise of Garuda Grid [10]. We expect that the cloud will become capable of hosting highly demanding, mission critical and advanced bioinformatics applications available to the large community of bioinformaticians with a highly efficient user-friendly interface.

2. LEGACY SETUPS

HPC solutions consist of big clusters running Unix/Linux. A cluster is supported by a package of cluster management software like Rocks, Ganglia etc. A cluster has a high speed communication channel like InfiniBand or PARAMNet [11] which spawns mpi based parallel programs generally written in C or Fortran. Clusters use multiuser setup and job managers like SGE or PBS are used for managing multiple jobs by multiple users. The users of such HPC facilities are often limited to one organization which has such a big and expensive facility. The facility requires a datacenter setup and consumes electric power mostly utilized for cooling and powering the HPC nodes. The system administration team has a tough time to maintain the HPC facility especially when some technical or power problem occurs and the facility is unavailable to the users. The software and libraries are also locked with the OS versions. Some software may require new libraries but it is practically impossible to install the libraries if it is not supported with the current version of the base OS. The whole cluster requires an upgrade due to some upgrade in existing software or installation of new software. The HPC cluster has to maintain homogeneous nodes and should have similar environment and libraries on every node. Also HPC clusters have a prefixed performance figure generally denoted by FLOPS (floating point operations per second). The typical setup has a lot of issues with up-gradation and adoption of changes. If a cluster is not shared then the most common problem is under utilization resulting in wastage of power for running no-op CPU cycles. Grid computing came as a boon for resource sharing and scientific collaboration. Underutilized resources were shared and used to solve big problems of scientific importance. Making a Grid operational also became a big task by itself and it requires a suite of advanced middle-ware like Globus to make the communication among the shared resources possible. A Grid works on donor and user ratio mechanism. If a user of resources exceeds the donor itself then the whole mechanism of grid fails. Grid comprises mainly of clusters along with other resources like scientific instruments. Thus it inherits the intrinsic problems of clusters also. The legacy systems, in-spite of having some inherent problems are widely used due to its unmatched performance and ability to solve real life science problems. The existing BRAF facility is having a similar setup but it is providing access to external users through ssh and is not limited by the organization boundary. It has users across the globe and practically anyone satisfying its usage policy can connect using Internet to utilize the facility.

3. PRIVATE CLOUD

The private cloud is a shared multi-tenant environment built on a highly efficient automated and virtualized infrastructure using the in-house resources. Many public cloud services are also available in the market with the advantage of pay as per usage policy. Some of the major players are Amazon (EC2 and S3 services) [12], [13], Google (AppEngine PaaS) [14] and Microsoft (Azure cloud services) [15]. The public cloud providers have various charging policies depending on the usage of raw compute infrastructure or services using compute and storage. Business solutions can be hosted on a cloud just by using a credit card and Internet connection. Research and academic organizations can also take the advantage of cloud based infrastructure and services by using the available in-house resources in the form of virtualized private cloud. Open Source implementation of cloud infrastructure like Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) [16] and OpenNebula [17] allows creating cloud infrastructure compatible with public clouds. These software are the major building blocks in building a private cloud. The initial installation of the applications can be done on private cloud and later can be shifted to the public cloud. Concepts like cloud bursting [18] can be very useful to provide unlimited resources at the disposal by using public cloud resources if the local resources are exhausted. The private and public cloud have an analogy with the term intranet and internet. The intranet uses the same underlying technology as used by internet but the difference is that intranet works within the boundary of an organization behind a firewall but the internet has no organization boundaries. Similarly a private cloud can belong to an organization solely but it can meet the public cloud the same way intranet can access internet. A Private cloud is significant to those organizations where a large scale infrastructure is already in place and the benefits of a cloud like automated flexibility are required.

We have tried to explore the available open source technologies like Eucalyptus and Open Nebula for deploying our private cloud. Our experience has a learning curve with both the prominent open source technologies. We are able to set up a fully functional private cloud using Eucalyptus and BRAF hardware resources. We have a future road-map to provide a web based cloud access portal.

Other key element of the private cloud include standardized software environment provided as a service and Amazon EC2 compatible open source clients like HybridFox [19], euca2ools [20] and S3Curl [21] that enables user to request and manage capacity for their applications.

4. EUCALYPTUS AS A CLOUD ENABLER

Eucalyptus was designed keeping in mind the de-facto standard of cloud computing i.e. Amazon EC2. Eucalyptus interfaces are compatible with the commercial EC2 and the tools to access EC2 are compatible with both the platforms. HybridFox is a Firefox plugin to manage both Eucalyptus and EC2 with a single interface. Eucalyptus also provides Walrus [16] which is similar to Amazon S3 storage solution. The Eucalyptus cloud computing platform has five high-level components: Cloud Controller (CLC), Cluster Controller (CC), Node Controller (NC), Storage Controller (SC) and Walrus [16]. CLC receives all the incoming requests from client to cloud; CC has a responsibility of managing clusters and virtual networking between VMs and external users; NC actually controls the VM instances; SC manages all the persistent data in cloud; Walrus provides a bucket-based storage similar to S3 (Simple Storage Service) from AWS. Each Component is implemented as a web service. All together these components provide a fully functional private cloud setup having features compatible with the standard commercial public cloud counterpart like Amazon. The cloud setup at BRAF is hosted on a server farm with a set of clustered hardware resources. The overview of the actual implementation is explained in the coming text.

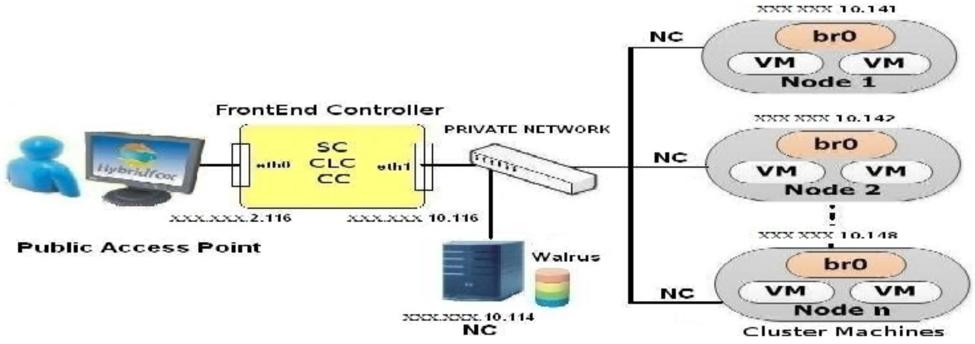

Figure 1. Private Cloud Architecture

4.1. Components that constitutes the Eucalyptus based setup

Eucalyptus offers a suite of software used at various component level of the overall cloud solution. The private cloud setup has used available resources for an internal private cloud. It can be used in the private network of BRAF. The various components shown in Figure. 1 are explained as follows.

4.1.1. FrontEnd Controller

This is the entry gateway for the cloud setup. The eucalyptus components viz. Storage Controller, Cluster Controller and Cloud Controller are installed on the same machine. An intranet IP XXX.XXX.2.116 is allocated to this server for accessing over local area network (LAN). It has another network card connected to a private network switch and is as signed IP XXX.XXX.10.116. These IP will be used as the reference to the machines further.

4.1.2. Walrus Controller

The server with Walrus controller is a high memory server with higher capacity disk mounted on it. Walrus behaves as a get/put repository interface for the bucket and objects. The storage space for buckets & objects is mapped to a single, non-shared file system directory mounted on the machine. The IP XXX.XXX.10.114 is given to this machine and it can communicate with Front-end Controller using the private network switch.

4.1.3. Cluster Machines

Node Controllers with IP in the range XXX.XXX.10.141 to XXX.XXX.10.148 with KVM as underlying hyper-visor are installed. These are the components of our cloud setup that host VM instances and is responsible for instance start up, inspection, shutdown, and cleanup. One Node Controller is used to run multiple VM instances. These are 1U rack server with dual socket 4 core Intel(R) Xeon(R) CPU E5410. One NC is also with quad socket 12 cores AMD Opteron(tm) Processor 6174 with IP XXX.XXX.10.114.

The Frontend is installed directly through binaries available from Ubuntu Enterprise Cloud (UEC) [22] and all node controllers are installed using Eucalyptus Source repositories (EEC) [23]. So interoperability between UEC & EEC is again an issue which we solved manually by doing changes in some of the configuration files of Eucalyptus. The interoperability issues are solved by changing installation directories, owner and group, creating different symbolic links and changing libvirtd.conf.

4.2. The steps of communication between the various components are:

- Request a Virtual Machine (VM) via euca2ools or HybridFox.
- VM template disk image pushed to Node Controller from Walrus.
- Disk imaged padded to the correct size and packaged for Hypervisor.
- The Node Controller (NC) provides a virtual Network Interface NIC with virtual MAC address.
- DHCP on Cluster Controller provides IP.
- VM starts by Hypervisor KVM/Xen.
- User can login via SSH.

4.3. Bundling Linux Images

The EMI (Eucalyptus Machine Images) can be downloaded from eucalyptus archived certified images repository [24]. To enable a VM image as an executable entity, a user/admin must add a root disk image, a kernel/ramdisk pair (ramdisk may be optional) to Walrus and register the uploaded data with Eucalyptus. Each is added to Walrus and registered with Eucalyptus separately, using three EC2 commands. The following code shows the steps:

4.3.1. Unpack it to any directory, add the kernel to Walrus and register it with Eucalyptus

- euca-bundle-image -i <kernelfile> --kernel true
- euca-upload-bundle -b <kernelbucket> -m /tmp/<kernel file>.manifest.xml
- euca-register <kernel-bucket>/<kernelfile>.manifest.xml

4.3.2. Add the ram disk to Walrus, and register it with Eucalyptus

- euca-bundle-image -i <initrd file> --ramdisk true
- euca-upload-bundle -b <initrd bucket> -m /tmp/<initrd file>.manifest.xml
- euca-register <initrd bucket>/<initrdfile>.manifest.xml

4.3.3. Next, add the root file system image to Walrus

- euca-bundle-image -i <vm image file>
- euca-upload-bundle -b <image bucket> -m /tmp/<vm image file>.manifest.xml
- euca-register <image bucket>/<vm imagefile>.manifest.xml

4.4. Installation of Bioinformatics tools in customized images

The virtual images are customized to provide biologists ready to use bioinformatics applications. It avoids the complex states of installation and compilation using various OS dependent functionality and compilers for different platforms. The user can choose a pre-built image according to the application domain and create a running instance just by click of a mouse using interfaces like HybridFox. The bioinformatics software with all the required libraries and most commonly used input components can be used in seconds. Instances of general purpose bioinformatics software images like Cloud BioLinux [25] can also be used. The following code shows the steps that we have used to make customized images:

4.4.1. Download image from walrus

- euca-download-bundle -b name_of_bucket -d destination_directory

4.4.2. Unbundle image downloaded previously

- euca-unbundle-image -m path_manifest-file -d destination_directory

4.4.3. Resize image to carry tools

- sudo fsck.ext3 -f IMAGE
- sudo resize2fs IMAGE SIZE

4.4.4. Mount the root file system to host OS

- mkdir temp-mnt
- sudo losetup /dev/free_loop IMAGE
- sudo mount /dev/free_loop temp-mnt
- mount -o bind /proc temp-mnt/proc
- mount -o bind /dev temp-mnt/sys
- mount -o bind /dev temp-mnt/dev

4.4.5. Change root

- sudo chroot temp-mnt

4.4.6. Now the bioinformatics software for customized images are installed

4.4.7. After installation is complete unmount device

- sudo umount temp-mnt/dev
- sudo umount temp-mnt/sys
- sudo umount temp-mnt/proc
- sudo umount /dev/free_loop

4.4.8. Upload back the image to walrus

4.5. Client Interfaces

HybridFox is a Firefox add-on used to manage the private cloud setup. HybridFox can be used as a client for Amazon EC2 as well as Eucalyptus based cloud setup by selecting the appropriate region. Preferred Region can be set to Eucalyptus end-point URL http://hostname:8773/services/Eucalyptus.

Hybridfox can do everything that could be possibly done using Eucalyptus based cloud setup. All the management activities required for the Eucalyptus computing environment like managing images and instances, raising and stopping instances, managing elastic IP, managing security groups, managing key-pairs, managing Elastic Block Storage(EBS) [26] can be managed by using HybridFox. HybridFox is provided as a Firefox browser integrated graphical interface plugin.

The above mentioned eucalyptus based setup provides a fully functional private cloud services which can be used by any authorized user in an organization to access the virtualized infrastructure. The users mainly bioinformaticians have an unmatched benefit of using prebuilt bioinformatics images having the most commonly used and customized bioinformatics software.

5. BENCHMARKING

Benchmarking was done for comparison of performance on Physical nodes v/s Virtual nodes. Smith-Waterman (ssearch) comparison included in the Fasta package version 35 [27] for database similarity searching against the UniProt knowledge base Release 2011 04 [28] was used for benchmarking. Mycobacterium tuberculosis H37Rv [29] database, the causative agent of tuberculosis was used. The benchmarking is an attempt to see whether virtualized infrastructure can be used to get the results based on the real life biological problem. If yes then what is the penalty of using resources over cloud than physical servers and to find out whether the ranges are permissible by the scientific community or not.

5.1. Experimental Setup

Private cloud installed using BRAF resources viz. 1U rack server with AMD Opteron(tm) Processor 252 ; eight 1U rack servers with dual socket 4 core Intel(R) Xeon(R) CPU E5410 and a server with quad socket 12 cores AMD Opteron(tm) Processor 6174 was used. Two setups are used for comparison of performance as follows:

- Virtual Setup: The cluster nodes with host OS Ubuntu 10.04 and CentOS release 5.7 with KVM hypervisor were integrated as a cloud using Eucalyptus 1.6. The virtual machine disk image was created with a size of 4.5 GB. One Customized image with required Fasta executable was prepared following the steps mentioned in section D. Four instances of the custom image are created using HybridFox client interface. Every instance running on cloud is assigned an IP from the pool of reserved IP allocated by the Eucalyptus services.

- Physical Setup: The same cluster nodes were used as a physical cluster mounted with NFS with access to the executable and input data. Passwordless ssh [30] configuration was done among the nodes for mpi communication. MPICH2 with Hydra Process Manager has been used for process communication. As it is an experimental testbed no job manager or other cluster management tools are installed. Machinefile is used for node information. Every physical node has an static intranet IP in the range XXX.XXX.2.141 to XXX.XXX.2.148.

5.2. Private Cloud Performance Comparison

We have run the Fasta ssearch code using two different environment one comprising 4 nodes (each having 8 cores) of 1U rack based physical machines and other of 4 instance of 8 core virtual machines created using Eucalyptus based private cloud setup managed by HybridFox interface. Time of execution is measured by running same code with same input size and other input parameters on the two test environments. The objective behind running these tests is to compare the performance between physical machines and virtual machines created in a private cloud environment.

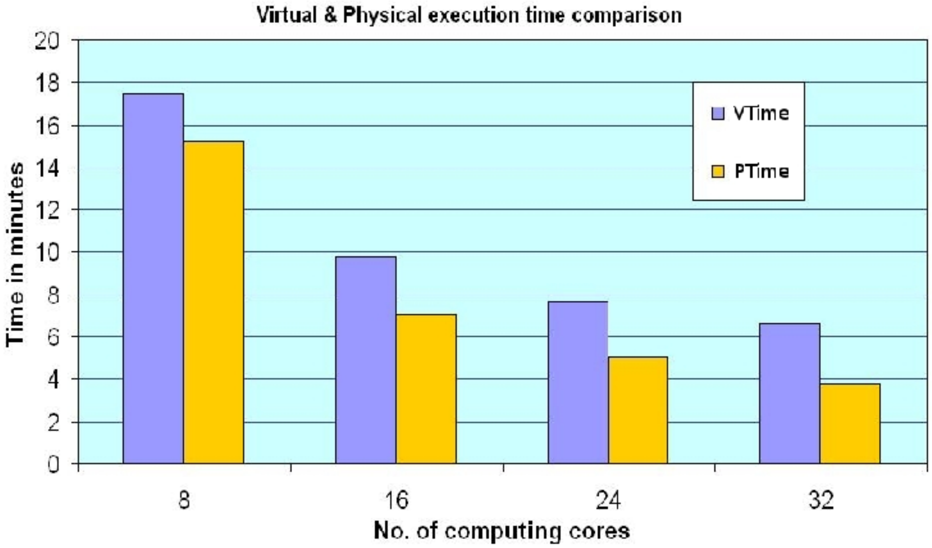

Figure 2. Benchmark Results(time scale)

The Figure. 2 shows the four set of run with the deployment of 4 virtual machine. If we take the case of 32 cores alone each virtual machine instance takes approximately 6.61 minutes while on the other side the physical machine takes 3.75 minutes for running the same application with same input values and other parameters. The results of 8, 16 and 24 and 32 cores are the sets represented in the form (VTime , PTime) where VTime is the time taken to run the ssearch code on virtual machine instance created dynamically on demand using the Eucalyptus based private cloud setup and PTime is the time taken to run the same code on physical machine with same number of cores.

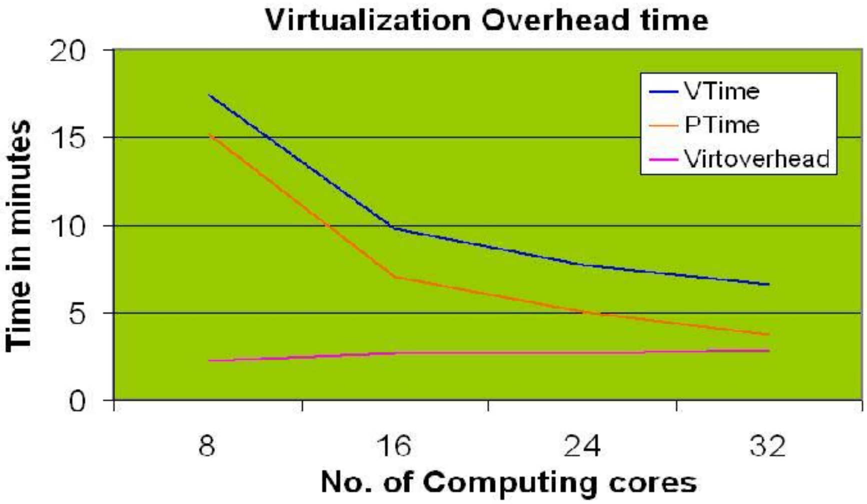

Figure 3. Virtualization Overhead time analysis

The results are shown in Figure. 3. In this figure we present an analysis to show that the virtualization does not have any incremental effect with the application running with increasing number of cores. One run is for the physical machine using 8, 16, 24 and 32 Intel Xeon cores across the 4 rack servers and corresponding run on virtual environment with virtual cores. The graph shows that the four runs show a consistent decline in execution time using virtual machine images and is slightly more than the execution time on the Physical machine run. We denote the time difference between PTime and VTime as Virtoverhead . Thus

$$Virtoverhead = VTime - PTime \qquad (1)$$

The Figure. 3. shows that the Virtoverhead time for every run is not showing any relation with the change VTime and PTime and can be considered as a constant entity. Thus if the execution time of the experiment is very big then Virtoverhead can be considered as negligible.

6. DISCUSSIONS

Eucalyptus is one of the most commonly used open source cloud enabler. It has a great advantage of Amazon EC2 compatibility in IaaS implementation. Researchers gain the benefit of secured environment behind the intranet firewall in a private cloud setup. Some components of existing BRAF infrastructure have been transformed to a flexible virtual computing environment which inherits the advantage of cloud computing technology. Custom built virtual images provide biologists ready to use bioinformatics applications just on a click. The bioinformatics software with all the required libraries and most commonly used input components can be used avoiding the complex states of installation and compilation of bioinformatics software. Virtualization always have a time penalty over actual physical resources but it may not be significant in the cases where the availability of resources is more important than some execution delay.

7. CONCLUSIONS

We are able to bring up a fully functional private cloud setup using BRAF resources. The open source software used are based on the existing industry de-facto standards. The previous experiences with the grid computing and HPC facility of BRAF has helped to understand the complexity involved in the implementation of the virtualized cloud environment. The benchmarks demonstrates that using virtual resources have a performance penalty but it does not over-weigh the benefits of availability on demand and better managed resources with dynamic scalability. This is an early experience of the IaaS facility having a great potential of HPC using virtualization over private cloud setup. In future browser based user-friendly cloud management utility may be developed specifically for the private cloud setup. These initiatives will enable the researchers to use the resources that were otherwise underutilized and solve the real life biological problems.

ACKNOWLEDGMENTS

The authors acknowledge the Department of Electronics & Information Technology, Ministry of Communications & Information Technology (Govt. of India) and Bioinformatics Resources & Applications Facility (BRAF) of CDAC.

REFERENCES

[1] “VMware Virtualization Software homepage,” Internet: http://www.vmware.com.
[2] “Homepage of Xen hypervisor,” Internet: http://xen.org/.
[3] “Homepage of Hotmail web based email,” Internet: www.hotmail.com.
[4] “Homepage of Gmail web based search engine,” Internet: www.gmail.com.
[5] “Facebook web based social networking,” Internet: www.facebook.com.
[6] “Google Docs Online access to documents from anywhere,” Internet: http://docs.google.com/.

[7] “Bioinformatics Resources and Application Facility portal,” Internet: http://bioinfo-portal.cdac.in/braf.html.

[8] “Eucalyptus open-source cloud software,” Internet: http://open.eucalyptus.com/.

[9] “OpenNebula open-source cloud management platform,” Internet: http://opennebula.org/.

[10] “GARUDA India’s first national grid initiative,” Internet: http://www.garudaindia.in/.

[11] P. K. Sinha, S. Dixit, N. Mohanram, S. Purohit, R. Arora and S. Ramakrishnan, “Current state and future trends in high performance computing and communications (HPCC) research in India,” in Distributed Computing Systems, 2004. FTDCS 2004. Proceedings. 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, may 2004, pp. 217–220.

[12] “Amazon EC2 service homepage,” Internet: http://aws.amazon.com/.

[13] “Amazon Simple Storage Service,” Internet: http://aws.amazon.com/s3/.

[14] “Google App Engine to build and host web applications on Google’s infrastructure,” Internet: https://appengine.google.com/.

[15] “Windows Azure an open and flexible cloud platform,” Internet: http://www.windowsazure.com/en-us/.

[16] D.Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, “The Eucalyptus Open-Source Cloud-Computing System,” in 2009 9th IEEEACM International Symposium on Cluster Computing and the Grid, vol. 0, no. 9, may 2009, pp. 124–131.

[17] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, “Capacity Leasing in Cloud Systems using the OpenNebula Engine,” in Cloud Computing and Applications 2008 (CCA08), vol. 2008, 2008, pp. 1–5.

[18] T. Bicer, D. Chiu, and G. Agrawal, “A Framework for DataIntensive Computing with Cloud Bursting,” in SC 11 2011 IEEE International Conference on Cluster Computing, 2011, pp. 169–177.

[19] “HybridFox open source Mozilla Firefox extension to manage cloud,” Internet: http://code.google.com/p/hybridfox/.

[20] “Euca2ools command-line tools for interacting with cloud services,” Internet: http://www.eucalyptus.com/download/euca2ools.

[21] “S3 Curl tool that allows users to interact with Walrus,” Internet: http://www.eucalyptus.com/eucalyptus-cloud/tools/s3curl.

[22] “The Ubuntu Enterprise Cloud,” Internet: http://www.ubuntu.com/cloud/.

[23] “Download and Install Eucalyptus,” Internet: http://www.eucalyptus.com/download/eucalyptus.

[24] “Eucalyptus Archived certified pre-packaged virtual machines,” Internet: http://open.eucalyptus.com/wiki/EucalyptusArchivedCertifiedImages.

[25] K. Krampis, T. Booth, B. Chapman, B. Tiwari, M. Bicak, D. Field, and K. Nelson, “Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community,” BMC Bioinformatics, vol. 13, no. 1, p. 42, 2012.

[26] “Amazon Elastic Block Store provides block level storage volumes,” Internet: http://aws.amazon.com/ebs/.

[27] W. R. Pearson and D. J. Lipman, “Improved tools for biological sequence comparison,” in Proceedings of the National Academy of Sciences of the United States of America, vol. 85, no. 8, 1988, pp. 2444–2448.

[28] T. U. Consortium, “Ongoing and future developments at the Universal Protein Resource,” in Nucleic Acids Research, vol. 39, no. Database issue, 2011, pp. D214–D219.

[29] J. C. Camus, M. J. Pryor, C. Mdigue, and S. T. Cole, “Reannotation of the genome sequence of Mycobacterium tuberculosis H37Rv.” Microbiology, vol. 148, no. Pt 10, pp. 2967–2973,2002.

[30] “Password less ssh,” Internet: http://linuxconfig.org/Passwordless ssh.