Magellan: Experiences from a Science Cloud

Lavanya Ramakrishnan,
Scott Campbell, Richard S. Canon,
Tina Declerck, Iwona Sakrejda
NERSC, Lawrence Berkeley National Lab
Berkeley, CA
(lramakrishnan,scampbell,
scanon, tmdeclerck,isakrejda)@lbl.gov

Susan Coghlan, Piotr T. Zbiegiel,
Rick Bradshaw, Narayan Desai,
Anping Liu
Argonne National Lab
Argonne, IL
(smc,pzbiegiel,aliu)@alcf.anl.gov,
(desai, bradshaw)@mcs.anl.gov

ABSTRACT

Cloud resources promise to be an avenue to address new categories of scientific applications including data-intensive science applications, on-demand/surge computing, and applications that require customized software environments. However, there is a limited understanding on how to operate and use clouds for scientific applications. Magellan, a project funded through the Department of Energy's (DOE) Advanced Scientific Computing Research (ASCR) program, is investigating the use of cloud computing for science at the Argonne Leadership Computing Facility (ALCF) and the National Energy Research Scientific Computing Facility (NERSC). In this paper, we detail the experiences to date at both sites and identify the gaps and open challenges from both a resource provider as well as application perspective.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer Communication Networks—*Distributed Systems*

General Terms

Design, Performance

Keywords

cloud computing, data parallel computing, mapreduce, scientific computing

1. INTRODUCTION

Cloud computing has served the needs of web applications for the last few years. Among other benefits, cloud computing leverages the features of the MapReduce [7] programming model and virtualization technology. Cloud resources promise to be an avenue to address new categories of scientific applications, including data-intensive science applications, on-demand/surge computing, and applications that require customized software environments. A number of groups in the scientific community are investigating and tracking how the cloud software and business model might impact the services offered to the scientific community and the evolution of software infrastructure to manage these resources [18, 31, 8]. However, its use for serving the needs of scientific applications is still relatively unexplored.

The goal of Magellan, a project funded through DOE ASCR, is to investigate how the cloud computing business model can be used to serve the needs of mid-range computing and future data-intensive computing workloads for the DOE Office of Science that are not served through DOE data center facilities today. The distributed testbed infrastructure has been deployed at the Argonne Leadership Computing Facility (ALCF) and the National Energy Research Scientific Computing Facility (NERSC). The testbed consists of IBM iDataPlex servers and a mix of special servers, including Active Storage, Big Memory, and GPU servers. The testbed also has a mix of storage options, including both distributed and global disk storage, archival storage, and two classes of flash storage.

Cloud computing has similarities with other distributed computing models such as Grid and Utility computing. However, the use of virtualization technology, MapReduce programming model and tools such as Eucalyptus and Hadoop, require us to study the impact of cloud computing on scientific environments. The Magellan project is focused on understanding the unique requirements of DOE science applications and the role cloud computing can play in scientific communities. Towards meeting this goal, the project has deployed a testbed across both sites and is testing a diverse set of cloud software stacks to address different scientific needs. The project is using the testbed to advance understanding of how private cloud software operates and identify its gaps and limitations. In concert with scientific groups, we also explore the design and challenges of scientific environments using cloud resources. Specifically, in this paper:

- We detail the specific requirements from cloud resources for scientific use.
- We describe our experiences in operating a testbed with virtualization software such as Eucalyptus, OpenStack, and Hadoop, an open source implementation of MapReduce. We identify gaps in current cloud software for scientific use.
- We describe early adopters that are leveraging these two environments and the challenges faced in application design and development.

The rest of this paper is organized as follows. Section 2 provides an overview of cloud computing features and de-

ScienceCloud'11, June 8, 2011, San Jose, California, USA.

tails the requirements of scientific applications that could potentially benefit from cloud computing. We provide an overview of the Magellan testbed in Section 3 and discuss challenges in operating cloud software stacks. In Section 4, we describe the early science adopters of cloud technologies and discuss the impact of these technologies on application design decisions. Section 5 provides answers to a number of key questions related to applicability of cloud computing for scientific workflows. Section 6 details related work, and we conclude in Section 7.

2. OVERVIEW

The National Institute of Standards and Technology (NIST) definition of cloud computing describes it as *a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction* [25]. Scientific environments at high performance computing (HPC) centers today provide a number of these key features, including resource pooling, broad network access, and measured services based on user allocations. Cloud computing introduces a new usage or business model and additional new technologies and features. Users with applications that have more interactive needs could benefit from *on-demand, self-service environments* and *rapid elasticity* through the use of virtualization technology, and the MapReduce programming model to manage loosely coupled application runs.

The Magellan team has been working with user groups interested in using cloud computing resources. In this section, we detail some of the unique characteristics that impact science clouds. The needs of science user groups have influenced system software decisions at both sites and also helped us identify gaps in the middleware and software.

2.1 Computational Models

Scientific workloads can be classified into three broad categories based on their resource requirements: large-scale tightly coupled computations, mid-range computing, and high throughput computing. In this section, we provide a high-level classification of workloads in the scientific space based on their resource requirements and delve into the details of why cloud computing is attractive to these application spaces.

Large-Scale Tightly Coupled. These are complex scientific codes generally running at large-scale supercomputing centers across the nation. Typically, these are MPI codes using a large number of processors (often in the order of thousands) and may have long-running jobs. These jobs are serviced at supercomputing centers through batch queue systems. Users wait in a managed queue to access the resources requested, and their jobs are run when the required resources are available and no other jobs are ahead of them in the priority list. Most supercomputing centers provide archival storage and parallel file system access for the storage and I/O needs of these applications. Our earlier work shows that this class of applications takes a performance hit when working in virtualized cloud environments [15].

Mid-range Tightly Coupled. These applications run at a smaller scale than the large-scale jobs. There are a number of codes that need tens to hundreds of processors. Some of these applications run at supercomputing centers and backfill the queues. More commonly, users rely on small compute clusters that are managed by the scientific groups themselves to satisfy these needs. These mid-range applications are good candidates for cloud computing even though they might incur some performance hit.

High Throughput. Some scientific explorations are performed on the desktop or local clusters and have asynchronous, massively independent computations. Even in the case of large-scale science problems, a number of the data pre- and post-processing steps, such as visualization, are often performed on the scientist's desktop. The increased scale of digital data due to low-cost sensors and other technologies has resulted in the need for these applications to scale [22]. These applications are often penalized due to scheduling policies used at supercomputing centers. The requirements of such applications are similar to those of the Internet applications that currently dominate the cloud computing space, but with far greater data storage and throughput requirements. These workloads may also benefit from the MapReduce programming model by simplifying the programming and execution of these class of applications.

2.2 On-demand Customized Virtual Environments

The "Infrastructure as a Service" (IaaS) facility commonly provided by commercial cloud computing addresses a key shortcoming of large-scale grid and HPC systems; that is, the relative lack of application portability. This issue is widely cited as a key failure of grid systems, where a single software stack is deployed across systems distributed across geographic locales as well as organizational boundaries. A key design goal of these unified software stacks is providing the best software for the widest range of applications. Unfortunately, scientific applications frequently require specific versions of infrastructure libraries; when these libraries aren't available, applications may run poorly or not at all. For example, the Supernova Factory project that is building tools to measure the expansion of the universe and energy has a large number of custom modules [3]. The complexity of the pipeline makes it necessary to have specific library and OS versions and ends up being a barrier to making use of large resources that might become available. User-customized operating system images provided by application groups tuned for a particular application would address this issue.

2.3 Resource Availability and Quality of Service

Some scientific users prefer to run their own private clusters for a number of reasons. They often don't need the concurrency levels achievable at supercomputing centers, but do require guaranteed access to resources for specific periods of time. They also often need a shared environment between collaborators since setting up the software environment under each user space can be tedious and time consuming, and clouds might be a viable platform to satisfy this need.

2.4 ScienceClouds: Best of Both Worlds

Science users have used HPC centers for a number of years and often have large volumes of legacy data stored in current systems. Users are also interested in a private cloud that would enable them to get the benefits of cloud environments in conjunction with other facilities provided at supercomputing centers. For example, HPC systems typically provide high-performance parallel file systems that enable parallel coordinated writes to a shared filesystem with high bandwidth and capacity. HPC centers also typically provide an archival storage system to archive critical output and results.

3. MAGELLAN TESTBED

The user requirements for cloud computing are diverse, ranging from access to custom environments to the MapReduce programming model. These diverse requirements guided our *flexible* software stack at both sites - Argonne and NERSC. Users have access to customized virtual machines through Eucalyptus, enabling users to port between commercial providers and the private cloud, along with a Hadoop installation that allows users to evaluate the MapReduce programming model and the Hadoop Distributed File System. NERSC also provides access to a traditional batch cluster environment. This environment is used to establish baseline performance and collect data on workload characteristics for typical mid-range science applications that are considered suitable for cloud computing.

In this section, we detail our hardware and system software setup, as well as discuss our experiences with virtualization software and Hadoop.

3.1 Testbed Setup

The Magellan testbed hardware has been architected to facilitate exploring a variety of usage models and understanding the impact of various design choices. As a result, the testbed incorporates a diverse collection of hardware resources, including compute nodes, large-memory nodes, GPU servers, and various storage technologies. Eventually, Magellan is expected to be connected to the 100 Gb network planned for deployment by the DOE-SC-funded Advanced Networking Initiative.

Both Argonne and NERSC deployed compute clusters based on IBM's iDataplex solution. This solution is targeted towards large-scale deployments and emphasizes energy efficiency, density, and serviceability. The configuration for the iDataplex systems are similar at both sites. Each compute node has dual 2.66 GHz Intel Quad-core Nehalem processors, with 24 GB of memory, a local SATA drive, 40Gb Infiniband (4X QDR), and 1 Gb Ethernet with IPMI. The system provides a high-performance InfiniBand network which is often used in HPC-oriented clusters, but is not yet common in mainstream commercial cloud systems. Since the network has such a large influence on the performance of many HPC and mid-range applications, the ability to explore the range of networking options from native InfiniBand to virtualized Ethernet was an important design goal. The testbed is architected for flexibility and to support research. The hardware deployed is similar to high-end hardware in HPC clusters, thus catering to scientific applications

Argonne. The Magellan testbed at Argonne includes computational, storage, and networking infrastructure. There is a total of the 504 iDataplex nodes described above. In addition to the core compute cloud, Argonne's Magellan has three types of hardware that one might expect to see within a typical HPC cluster: Active Storage servers, Big Memory servers, and GPU servers. There are 200 Active Storage servers, each with dual Intel Nehalem quad-core processors, 24 GB of memory, 8x500 GB SATA drives, 4x50 GB SSD, and a QDR InfiniBand adapter. There are 15 Big Memory servers; each has 1 TB of memory, along with 4 Intel Nehalem quad-core processors, 2x500 GB local disks, and a QDR InfiniBand adapter. There are 133 GPU servers, each with dual 6 GB NVidia Fermi GPU, dual 8-core AMD Opteron processors, 24 GB memory, 2x500 GB local disks, and a QDR InfiniBand adapter. Finally, there is 160 terabytes (TB) of global storage. In total, the system has over 150 TF of peak floating point performance with 8,240 cores, 42 TB of memory, 1.4 PB of storage space, and a single 10-gigabit (Gb) external network connection.

The Argonne Magellan testbed uses the Argonne-developed tools bcfg2 [4] and Heckle [13] to provide advanced, bare-metal provisioning (”Hardware as a Service" or HaaS) and configuration management. The core compute servers have numerous cloud software stacks installed in various stages of availability to the users. There is always some portion of the cloud configured as a public cloud running a software stack that provides an Amazon EC2-compatible API. The public cloud is open to all users for science, development, and testing. In addition, a portion of the compute hardware has been set up as a Nimbus cloud, also open to all users. The remainder of the compute resources are dedicated to development, and are usually running various cloud software stacks at different times, including Open Stack, Eucalyptus 2.0, Ubuntu Enterprise Cloud [UEC] v1, and OpenNebula. A portion of the Active Storage servers are configured as a persistent HADOOP cluster. The remainder are configured as HaaS and are part of the raw provisioning pool, along with the GPU and Big Memory servers.

NERSC. The NERSC Magellan test bed also provides a combination of computing, storage, and networking resources. There are 720 iDataplex nodes as described earlier. The total system has over 60 TF of peak floating point performance. The InfiniBand fabric was built using InfiniBand switches from Voltaire. Since the system is too large to fit within a single switch chassis, multiple switches are used, and they are connected together via 12x QDR links (120 Gb/s) configured as a fully connected mesh topology. This topology is less expensive than a traditional, full fat tree network yet still provides a relatively high bisection bandwidth. A variety of storage hardware has also been deployed, including nearly 1 PB of disk storage, archival storage, and two classes of flash storage. The system has approximately 8 TB of high-performance flash storage capable of delivering 20 GB/s of aggregate bandwidth. There is also 10 TB of consumer-grade SATA storage installed in 40 of the compute nodes running Hadoop. While flash storage is not typically found in current cloud offerings, it is an emerging technology that can potentially play an important role for data-intensive computing problems and will likely appear in future cloud offerings. Finally, the Magellan Testbed at NERSC also has 18 service nodes that are connected to the WAN via 10Gb.

NERSC is using IBM's xCAT[33] and Adaptive Computing's Moab Adaptive Computing Suite to provision and manage the Magellan testbed. xCAT can provision nodes using a variety of methods, including diskless, disk-full, and hybrid. The framework can also utilize the IPMI management interface to automatically power on and off nodes. In addition to providing command-line tools, xCAT also provides an API that can be used by external resource managers such as Moab. Coupling xCAT with Moab will enable

NERSC to explore other models of delivering capabilities associated with cloud computing and dynamically repurpose resources. This includes on-demand bare metal provisioning of various OS images, which may be a better model for providing access to custom images for scientific workloads.

3.2 Virtualization Software

At the start of the project, Eucalyptus 1.6.2 had the most advanced feature set and hence was our logical choice for evaluation of user instantiated virtualized environments. Since then, other offerings such as OpenStack and upgrades to Eucalyptus have become available. In this section, we detail our experiences with these software packages.

3.2.1 Eucalyptus 1.6.2

The Eucalyptus project provides an open source platform for conducting cloud computing research. It is API-compatible with Amazon's EC2. Eucalyptus supports nearly all the core features of EC2, including creating virtual instances, elastic block storage (EBS), S3, and elastic IP addresses. Since the software is open-source, it is possible to make modifications and add hooks and callouts to gather data. This makes it a useful tool for exploring cloud computing for DOE. Eucalyptus provides a convenient platform for creating and managing a private cloud platform for multiple users. However, Eucalyptus, like much of the cloud computing software, is still in its infancy and has design principles that may not be compatible with supercomputing center policies. In addition, bugs of various flavors have been encountered.

Scalability. Eucalyptus permits a cloud system to include a number of cluster controllers. Our current deployment is limited to a single cluster controller that exposes a number of scaling problems. The Eucalyptus network model forwards all traffic from the virtual machines running in each cluster through the cluster controller. This setup allows the system to implement security groups more easily; however, it also creates a potential network bottleneck for the running virtual machines. Even a moderate amount of network traffic spread across many virtual machines can saturate the cluster controller's network bandwidth. Testing of the virtual machine capacity revealed that Eucalyptus had a limit to the number of simultaneously running virtual machines between 750 and 800. The limit is related to the message size limit in a communication protocol and enforces a hard limit to the number of simultaneously running virtual machines. Also, many of the cluster controller operations are iterative. While this setup does not define hard limits for the system, it does mean that as the cluster is extended and node controllers are added, certain operations take longer to complete. When the cluster was expanded to more than 200 node controllers, there was a noticeable deterioration of performance during certain operations. For instance, terminating large numbers of running virtual machines can cause delays for other operations such as new virtual machine requests.

Image Management. In today's environments, sites manage the operating system and some base software packages, and scientists manage the specific application software and data. This division clearly demarcates the responsibility of system and application software maintenance between the site system administrators and the users. In the virtual machine model, users need to have an understanding of the OS administration. Support lists at both sites have received a large quantity of e-mails that pertain to basic system administration questions. In the long term, this will impact the user support model available at the sites. Users need to know how to create, upload, and register images, and have system administration skills on the OS in the virtual machine.

Although users manage their virtual machine images, the kernel and ramdisk are registered and managed by the system administrators, allowing sites to control kernel versions. System administrators now need to manage a large number of kernels and OS versions; this can be tedious and error prone.

Infrastructure. Eucalyptus uses a number of system services such as DHCP. It is difficult to get Eucalyptus to coexist with other services running on the system, since Eucalyptus and other similar cloud software assume they have complete control over system software in the cluster. The virtual machines that are provided also have limited support to offer features such as parallel file systems and Infiniband that are commonly used by scientific applications. Furthermore, the software stack needs a fair amount of tuning to get the optimal feature set and performance from the system. For example, an earlier version of KVM (*kvm-83-maint-snapshot-20090205*) used on the NERSC testbed did not expose the advanced Nehalem CPU instructions in the virtual machine.

Security. Deploying a new resource allocation model always requires a risk analysis of the new model in order to determine if modifications to existing resource protections will be required. Our analysis is based on understanding the typical use scenario of these technologies. A user will select a system image to boot on one or more virtual machines using Eucalyptus. Within the context of that virtual machine, the user is root and has complete administrative control. This is a significant deviation from current HPC environments. HPC centers tasked with operating a scientific cloud will not be able to control the operating system images running within the virtual machines. Users of this system will come with diverse backgrounds; however, they are unlikely to be well versed in managing operation system configurations on their virtual machines. This introduces a risk that, while not completely unique to clouds, is not a prevalent risk in many HPC cluster operations.

In addition, users might produce and make available hostile system images for use within the cloud. These images can be purposefully hostile or merely benignly so. Purposefully hostile images would be images customized and outfitted with malicious software that would allow the malicious user to have control of any virtual machine instances started with the given image. Benignly hostile machine images could be created by a non-malicious user inadvertently. These images could have a vulnerable service installed, or they could be configured poorly, allowing their compromise by malicious individuals.

Current open source cloud software stacks (e.g., Eucalyptus, OpenStack) allow end users to manage the firewall conduits that control access to their virtual machine instances. A user, without proper understanding of the impact of their choice, could allow any system on the Internet to interact with their virtual machines. This opens the machines up to attack and can ultimately lead to compromise.

All of these scenarios are a shift from most existing HPC cluster environments that have set access controls, set menus of image options, and do not typically allow users to have full root privileges on their compute nodes. Network activity monitoring is the key to mitigating these risks, since without access to running virtual machine instances, it is a challenge for a cloud operations team to identify problems directly on the virtual systems themselves. Network monitoring can be used to identify system misconfigurations, suspicious traffic, and any potentially compromised systems within the VM population. Given these constraints, we have identified a number of strategies and implemented one or more at each site:

- Run Intrusion Detection Systems at strategic points in the cloud system network to identify and study the vast majority of network traffic.
- Add additional host-based firewall rules to protect infrastructure from VMs.
- Continuously run ssh scans to quickly pick up accounts with bad passwords and instances with poorly configured remote access services.
- Provide documentation and educate users on systems administration best practices to assist with configuring their images properly.
- For site-supported images, run Nessus on the system. In addition, run scans often and look for hostile or vulnerable services as well as unexpected changes.
- For user-provided images, ask for Syslog to copy messages to the internal server and perhaps run instrumented SSHD.

Allocation and Accounting. The open source version of Eucalyptus does not provide an allocation and accounting model. Thus, it is impossible to ensure fairness or enforce any allocation policies among a group of users who might be trying to launch virtual machines.

Logging and Monitoring. Eucalyptus has verbose logging that could be beneficial to tracking events occurring on the systems. However, the monitoring to understand the behavior of the virtual machines, detect failure, and rectify failure events at the service level is limited. Eucalyptus 1.6.2 also has limited ability to recover gracefully from system-level failures. For example, restarting the Cloud Controller would typically result in the loss of IP address assignments for running instances and require all running instances to be terminated in order to cleanly recover.

Portability. Eucalyptus provides an Amazon-compatible user API to manage virtual machines. This enables easy portability of tools and applications between the public and private cloud. However, moving images between the two systems still requires a fair amount of IT expertise and can be time consuming and tedious.

3.2.2 Alternate Stacks: OpenStack and Eucalyptus 2.0

Eucalyptus is the leading open source software stack that helps set up private clouds. However, there are a number of gaps in the software features, such as accounting, allocation, and security policies that will need to be investigated and implemented for science clouds. We are investigating Eucalyptus 2.0 and other cloud software stacks such as OpenStack. In early testing, Eucalyptus 2.0 has shown improved stability and networking capabilities, leading to improved startup and shutdown times of the virtual machines.

OpenStack is a joint project between the National Aeronautics and Space Administration (NASA) and Rackspace that implements cloud compute and virtualized scalable storage components. OpenStack provides better configuration support, since the user details are stored in a database. OpenStack offers greater flexibility. For example, it allows roles to be assigned to users and projects can be assigned subnets. Unlike Eucalyptus, users are also allowed to register kernels and ramdisk images.

As cloud computing software matures in the next few years, problems related to system tooling and the lack of stability are likely to improve. However, the need for users to understand basic system administration and security policies will remain, and sites moving to cloud computing will need to take this additional burden for the users into consideration.

3.3 Hadoop

Hadoop is open source software that provides capabilities to harness commodity clusters for distributed processing of large data sets through the MapReduce [7] model. The Hadoop streaming model allows one to create map-and-reduce jobs with any executable or script as the mapper and/or the reducer. This is the most suitable model for scientific applications that have years of code in place capturing complex scientific processes.

The Hadoop File System (HDFS) is the primary storage model used in Hadoop. HDFS is modeled after the Google File system and has several features that are specifically suited to Hadoop/MapReduce. Those features include exposing data locality and data replication. Data locality is a key aspect of how Hadoop achieves good scaling and performance: Hadoop attempts to locate computation close to the data. This is especially true in the map phase, which is often the most I/O-intensive phase.

Both sites have Hadoop setups that are being used by different communities. In this section, we identify some of the core design features that are likely to impact its use in scientific environments.

Security Model. Hadoop 0.20 runs all jobs as user *hadoop*. This results in a situation where users may not be able to access non-HDFS files generated by the job. Thus, the permissions need to be fixed after the application completes, and the world-readable files make it hard to ensure data privacy for the end users. A recently released version fixes this model using Kerberos.

File System Access. The ability of Hadoop's MapReduce framework to use HDFS's data locality features can be useful to applications that need to process large volumes of data. However, Hadoop considers only the data locality for a single file and does not handle applications that might have multiple input sets. HDFS also does not expose a POSIX interface, which makes it difficult for legacy applications to leverage the file system directly, potentially compromising the data locality speedups that might be possible.

Configuration. The Hadoop configuration has a number of site-specific and job-specific parameters that are hard to tune to achieve optimal performance.

4. APPLICATION MANAGEMENT

In this section, we detail the case studies of some early applications running on Magellan and discuss the challenges and general design decisions that are impacted by the cloud characteristics.

4.1 Application Case Studies

A diverse set of applications are running on the Magellan resources at both sites. Our early case studies have been more in the space of scientific applications that are largely data parallel since they are good candidates for cloud computing. These applications are primarily throughput-oriented (i.e., there is no tight coupling between tasks). Second, the data requirements are large but well constrained. Finally, some of these applications have complex software pipelines and thus can benefit from customized environments in clouds. Cloud computing systems typically provide greater flexibility to customize the environment when compared with traditional supercomputers and shared clusters.

4.1.1 Genome Sequencing of Soil Samples

Magellan resources at both Argonne and NERSC were used to perform genome sequencing of soil samples pulled from two plots at the Rothamsted Research Center in the UK. The specific aim of this project was to understand the impact of long-term plant influence (rhizosphere) on microbial community composition and function. Two distinct fields were selected to understand the differences in microbial populations associated with different land management practices.

The demonstration used Argonne's MG-RAST metagenomics analysis software to gain a preliminary overview of the microbial populations of these two soil types (Grassland and Bare-Fallow). The MG-RAST software draws on ideas for distributing non-coupled computations from volunteer computing projects such as Boinc. It distributes work over HTTP with a DHCP-style lease, making the approach fault tolerant and capable of targeting multiple platforms and architectures simultaneously.

The goal of the demostration was to perform real science in the cloud, utilizing the testbeds at both sites, and to explore potential fail-over techniques within the cloud. The demostration used 150 nodes from Argonne's Magellan to perform the primary computations over the course of a week, with NERSC's Magellan acting as a failover cloud. Half a dozen machines on Argonne's Magellan were intentionally failed. Upon detecting the failures, the software automatically started replacement machines on the NERSC Magellan, allowing the computation to continue with only a slight interruption.

The same virtual machine image was used on both the Argonne and NERSC Magellan clouds. However, this was not a simple port and required some changes to the image. The instance uses all eight cores on each cloud node and about 40% of the available memory. The demostration was a single run within the Deep Soil project and represents only $\tilde{1}$/30th of the work to be performed. The project is continuing to run on the Magellan cloud.

This demonstration showed the feasibility of running a workflow across both the cloud sites, using one site as a fail-over resource.

4.1.2 Integrated Metagenomics Pipeline

The Integrated Microbial Genomes (IMG) system hosted at the DOE Joint Genome Institute (JGI) supports analysis of microbial community metagenomes in the integrated context of all public reference isolate microbial genomes. The content maintenance cycle for data involves running BLAST for identifying pair-wise gene similarities between new metagenome and reference genomes, where the reference genome baseline is updated with new (approximately 500) genomes every four months. This processing takes about three weeks on a Linux cluster with 256 cores. Since the size of the databases is growing, it is important that the processing can still be accomplished in a timely manner. The primary computation in the IMG pipeline is BLAST, a data parallel application that does not require communication between tasks and thus has similarities with traditional cloud applications. The need for on-demand access to resources makes clouds an attractive platform for this workload.

The pipeline is primarily written in Perl, but it includes components written in Java, as well as compiled components written in C and C++. The pipeline also uses several reference collections (typically called databases), including one for RNA alignment and a periodically updated reference database for BLAST. The pipeline and databases are currently around 16 GB in size. This does not include Bio-perl, BLAST, and other utilities. The pipeline was run across both Magellan sites through Eucalyptus. A simple task farmer framework was used to distribute workload across both sites. As virtual machines came up, a client would query the main server for work and run the computation.

The IMG pipeline has also been tested in the Hadoop framework to manage a set of parallel BLAST computations. The performance across a HPC machine, virtual machines, and a Hadoop cluster was found to be comparable (within 10%), making this a feasible application for clouds as well.

4.1.3 Climate 100

Climate scientists are better able to understand global climate change and evaluate the effectiveness of possible mitigations by generating and sharing increasingly large amounts of data. The Climate 100 data consists of on the order of a million files that average a few hundred megabytes each. Climate simulations running on large-scale supercomputers are used to generate these data sets. However, the analysis of these simulations can be performed on a variety of resources and are well-suited for cloud resources. Climate 100 simulation analysis has been run on virtual machines on Magellan [26], as well as using Hadoop. The results from this endeavor demonstrated that virtualized environments had portability benefits, and the performance made it a viable option for such large-scale data analysis. The loosely coupled analysis runs are well-suited for the diverse cloud environments.

4.1.4 STAR

Another early application on Magellan is the STAR nuclear physics experiment. STAR studies fundamental properties of nuclear matter from the data collected at Brookhaven National Laboratory's Relativistic Heavy Ion Collider. STAR needs on-demand resources for processing data in real-time and that makes cloud resources an attractive platform for this group. Previously, STAR has demonstrated the use of cloud resources both on Amazon and at other local sites [11, 21]. STAR is using NERSC Magellan resources to process

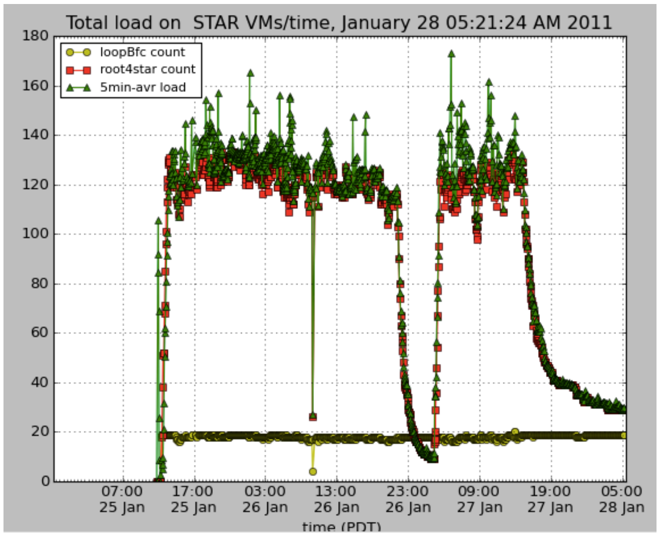

Figure 1: A plot of the status of processing near real-time data from the STAR experiment using Magellan at NERSC over a three-day period. The yellow circles show the number of instances running, the green triangles reflect the total load average across the instances, and the red squares plot the number of running processing tasks. (Image courtesy of Jan Balewski, STAR Collaboration).

near real-time data from Brookhaven for the 2011 run data. The need for on-demand access to resources to process real-time data and a complex software stack makes it useful to consider clouds as a platform for this application. Figure 1 shows the virtual machines used and corresponding aggregated load for a single run over three days.

4.2 Application Design and Management

Cloud computing promises to be useful to scientific applications due to advantages such as on-demand access to resources and control over the user environment. However, numerous challenges exist in terms of composing the application programming model, designing and constructing images, distributing the work across compute resources, and managing data.

4.2.1 Programming Model

Cloud computing has an impact on the programming model and other programming aspects of scientific applications. Scientific codes running at supercomputing centers are predominantly based on the MPI programming model. Ad hoc scripts and workflow tools are also commonly used to compose and manage such computations. Tools like Hadoop provide a way to compose task farming or parametric studies. Legacy applications are limited to using the streaming model that may not harness the full benefits of the MapReduce framework. The MapReduce programming model implementation in Hadoop is closely tied to HDFS, and its non-POSIX compliant interface is a major barrier to adoption of these technologies. In addition, frameworks such as Hadoop focus on each map task operating on a single independent data piece, and thus only data locality of the single file is considered.

4.2.2 Customizable Environments

One of the advantages that virtual environments provide is the ability to customize the environment and port it to different sites (e.g., CERN VM [6]). However, with that flexibility comes the responsibility and effort of creating and maintaining the environment. Tools exist to bundle a running operating system and upload it to the cloud system. However, some customization is typically required. Users need to have an understanding of standard Linux system administration, including managing *ssh daemons, ntp*, etc. Furthermore, debugging and testing can be tedious, since it often requires repacking and booting instances to verify the correct behavior. This also requires experimentation to determine what applications and data are best to include in the image or handle through some other mechanism. The simplicity of bundling everything in the image needs to be balanced with the need to make dynamic changes to applications and data. This process is complex and often requires users to carefully analyze what software pieces will be required for their application, including libraries, utilities, and supporting datasets. If the application or supporting datasets are extremely large or change quickly, then the user stores the data outside of the image due to limits on image size and its impact on virtual machine boot-up times.

4.2.3 Managing Computation

Another challenge in using cloud systems is developing a mechanism to distribute work. This is complicated by the fact that cloud systems like AWS are inherently ephemeral and subject to failure. Applications must be designed to dynamically adjust to compute nodes entering and leaving the resource pool. It must be capable of dealing with failures and rescheduling work. In traditional clusters, batch systems are routinely used to manage workflows. Batch systems such as Torque, Sun GridEngine, and Condor can and have been deployed in virtualized cloud environments [29, 16]. However, these deployments typically require system administration expertise with batch systems and an understanding of how to best configure them for the cloud environment. Grid tools can also play a role, but they require the user to understand and manage certificates and deploy Globus. To lower the entry barrier for scientific users, Magellan personnel have developed and deployed Torque and Globus-based system images.

Each of our applications have used different mechanisms to distribute work. MG-RAST and the IMG task farmer are examples of locally built tools that handle this problem. Hadoop might be run on top of virtual machines for this purpose; however, it will suffer from lack of knowledge of data locality. STAR jobs are embarrassingly parallel applications (i.e., non-MPI codes), where each job fits in one core and uses custom scripts to handle workflow and data management.

4.2.4 Data Management

The last significant challenge is managing the data for the workload. This includes both input and output data. For the bioinformatic workloads, the input data includes both the new sequence data as well as the reference data. Some consideration has to be given to where this data will be stored and read from, how it will be transported, and how this will scale with many worker nodes. The users would typically have access to a cluster-wide file system on a traditional batch cluster. However, with EC2-style cloud systems, there is a different set of building blocks: volatile local storage, persistent block storage associated with a single instance (EBS), and a scalable put/get storage system (S3). The image also can be used to store static data. Each of these options has differing performance characteristics that are dependent on applications. Thus, the choice of storage components depends on the volume of data and the access patterns (and cost, in the case of EC2).

4.2.5 *Cross-Site*

One key benefit of cloud computing is its ability to easily port the software environment as virtual machine images. This enables users to utilize resources from different computing sites. This allows scientists to expand their computing resources in a steady-state fashion, to handle burst workloads, or to address fault tolerance.

Both MG-RAST and the IMG pipeline were run successfully across both sites. However, application scripts were needed to handle registering the images at both sites, managing discrete image IDs across the sites, and handling distribution of work and associated policies across the sites.

5. DISCUSSION

The primary goal of Magellan is to understand if cloud computing can benefit science. In this section, we detail our experiences to date on a number of key questions that pertain to the applicability of cloud computing to scientific applications.

What applications can efficiently run on a cloud?

Virtualized cloud environments are useful for a number of applications that require customizable software stacks. In previous efforts [15], we have shown that scientific applications with minimal communication and I/O can run efficiently in virtualized cloud environments. However, high-end, tightly coupled applications are impacted significantly by the performance and reliability characteristics of today's clouds. Furthermore, porting and managing application stacks on cloud environments still have a number of open challenges that will need to be addressed by both computer science research, as well as specific scientific group solutions. Thus, while there might be some applications that can benefit from clouds, exact efficiency will vary, depending on the application characteristics.

Are cloud computing programming models such as MapReduce (Hadoop) effective for scientific applications?

The explosion of sensor data in the last few years has resulted in a class of scientific applications that are loosely coupled and data intensive. These applications are scaling up from desktops and departmental clusters and now require access to larger resources. These are typically high-throughput, serial applications that do not fit into the scheduling policies of many HPC centers. They also could benefit greatly from the features of the MapReduce programming model. In our early studies, Hadoop, the open source implementation of MapReduce and HDFS, the associated distributed file system, have proven to be promising for some scientific applications. The built-in replication and fault tolerance in Hadoop is advantageous for managing this class of workloads. In Magellan, we have also experimented with running Hadoop through a batch queue system. This approach can enable users to reap the benefits of Hadoop while running within the scheduling policies geared towards large parallel jobs.

Is it practical to deploy a single logical cloud across multiple sites?

It is possible to have a single logical cloud across multiple sites, but extensions to existing software components will be necessary. The single logical view can be built either in the application view or the site level.

Our experiences with the MG-RAST and IMG pipelines show how applications can effectively leverage multiple cloud sites to address resource needs. Co-allocation of cloud sites to effectively harness resources from a single logical cloud can be handled by application middleware. However, users need to manage images, metadata associated with virtual machines, and running instances of the virtual machines dynamically, since there are no simple tools available for seamless access of resources across sites.

Cloud software such as Eucalyptus has the ability to run multiple clusters under a single logical cloud. This could be used to operate a single logical cloud across multiple physical sites. However, there are gaps in current cloud software that cannot adequately represent some of the complicated policies that are necessary for such cross-site logical clouds. Cluster software such as Moab and Platform are also exploring cloud alternatives that might enable scheduling policies that would facilitate such cross-site integration

Can scientific applications use a data-as-a-service or software-as-a-service model?

Scientific groups have complicated software dependencies, and cloud technologies such as virtualization provide an easy way to package and distribute entire software stacks that can then be easily deployed at multiple sites without worrying about operating system and other package dependencies. For example, the ATLAS community uses the CERNVM [6] to package and distribute the ATLAS software. Scientists would often like to share observational and processed data, as well as software environments, with collaborators. Clouds can also facilitate deploying science gateways - web frontends that provide access to specific scientific computations and data. Software-as-a-service and data-as-a-service provide convenient models to share software and data with a large number of collaborators.

What are the security implications of user-controlled cloud images?

User-controlled images introduce additional security risk when compared to traditional login/batch-oriented environments, and they require a new set of controls and monitoring to secure. Sites typically rely on OS-level controls to implement many security policies. Most of these controls must be shifted into the hypervisor or alternative approaches must be employed. Implementing some simple, yet key security practices and policies on private clouds, such as running an intrusion detection system (IDS), capturing syslog data from the virtual machines, and constant monitoring, can avert a large number of the risks.

Moving forward, we stress two related, but ultimately different directions. The first is research focused: What can we do to explore changes in the environment that are substantially different from the current environment? The second is operational: How can we do better security overall? An example that strides the boundary between the two is an IDS that is kept aware of changes made to the configuration of Eucalyptus (such as a new cluster group starting) and applies local security policy to both the changes and inhabitant behaviors.

What are the unique needs and features of a science cloud?

The digital data and computational needs of scientific applications are seeing a tremendous growth. The diverse needs of these applications have resulted in a need to have a multitude of high-end hardware and software solutions. Some scientific applications can benefit from existing cloud technologies and infrastructure. But there are certain needs that are unique to science (below).

- *Science clouds need access to parallel filesystems and low-latency high bandwidth interconnect.* Virtualized cloud environments are limited largely by networking and I/O available in the virtual machines. Access to parallel file systems such as GPFS, Lustre, etc. and low-latency, high bandwidth interconnects such as InfiniBand within a virtual machine would enable more scientific applications to benefit from virtual environments with minimal overheads.

- *Science clouds need access to legacy data sets.* Efficient, easy, and cost-effective access to legacy data sets that reside in HPC centers today is critical for applications that run in cloud environments.

- *Science clouds need MapReduce implementations that account for characteristics of scientific data and analysis methods.* MapReduce can be useful for data-intensive scientific applications. However, there is a need for MapReduce frameworks that are not closely tied to HDFS and are available to use with other POSIX file systems. In addition, MapReduce implementations that account for scientific data access patterns (such as considering data locality of multiple input files) are desired.

- *Science clouds need bare metal provisioning for applications that require custom environments but cannot tolerate the performance hit from virtual machines.* Virtual machines are useful for end users who need specific customizable environments. However, the overheads of virtualization are significant for certain tightly coupled applications. These applications could benefit from bare metal provisioning or other approaches to providing custom environments.

- *Science clouds need preinstalled, pre-tuned application software stacks.* User-created virtual images are powerful. However, there is also a need for a standard set of base images and simple tools to reduce the entry barrier for scientists.

- *Science clouds need customizations for site-specific policies.* Cloud software solutions will need customization to handle site-specific resource allocation, security policies, accounting, and monitoring.

On-demand access to unlimited resources has been touted as an attractive feature of clouds for science. However, this is far from true in practice. Virtual machine startup overheads at large scale can impose significant delays [16]. Furthermore, the promise of unlimited resources is unrealistic for a scientific environment where scientists have an almost endless set of experiments they would like to run. Guaranteed access to resources can be provided by either over-provisioning (i.e., buying hardware for peak load and allowing the resources to go idle at other times), over-committing (where users are impacted by having limited access to resources on a node), or scheduling policies that allow lower-priority workloads to be terminated. The cloud software stacks require more sophisticated scheduling methods that can ensure fairness across different users and arbitration between user requests. Furthermore, on-demand access to resources could be achieved at existing HPC centers by implementing different resource allocation policies but would likely impact the effective utilization of the resources. This is similar to Amazon EC2 today, where resource allocation, performance, and availability have been reported to be highly variable, and higher-priced services are available for guaranteed access.

6. RELATED WORK

The Magellan project explores a range of topics that includes evaluating current private cloud software and understanding gaps and limitations, application software setup, etc. To the best of our knowledge, there is no prior work that does such an exhaustive study of various aspects of cloud computing for scientific applications. A number of different groups have conducted feasibility and benchmarking studies of running their scientific applications in the Amazon cloud [28, 12, 9, 18, 20, 17]. Standard benchmarks have also been evaluated on Amazon EC2 [24, 10, 27, 30, 32]. Our experiments show that high-end tightly-coupled applications are impacted by the performance characteristics of current cloud environments. However the focus of this paper is to outline our experiences and identify the gaps in current private cloud software.

The FutureGrid project [1] provides a testbed, including a geographically distributed set of heterogeneous computing systems that includes cloud resources. The aim of the project is to provide a capability that makes it possible for researchers to tackle complex research challenges in computer science, whereas Magellan is more focused on serving the needs of the science.

7. CONCLUSIONS AND FUTURE WORK

Magellan is investigating the use of cloud computing for science at the Argonne Leadership Computing Facility and the National Energy Research Scientific Computing Facility. The sites have deployed testbeds running a diverse cloud software stack and identified that, while current cloud software can be used for science clouds, there exists a need for enhancements to provide better performance, scalability, and stability. The paper details the case studies of running data parallel science applications on the cloud and the use of multi-site clouds for expandability and fault-tolerance.

Virtualized environments facilitate the custom software environments that are required by many applications for simplicity and portability. Large-scale data parallel scientific applications benefit from the MapReduce programming model for managing the large number of computations. The fault tolerance and data locality ideas built into implementations such as Hadoop enhance run-time efficiency for this class of applications.

Magellan continues to explore the suitability of cloud software stack and technologies for science, including the cost efficiency of cloud computing and detailed workload analysis for identifying applications that can benefit from clouds. Current cloud computing has a number of gaps and open challenges for scientific applications that will need to be addressed in future work. Cloud software such as Eucalyptus and Hadoop need continued work to ensure stability at large scale and better integration with site-specific services such as allocation, accounting, etc. Data parallel scientific applications will benefit from the MapReduce programming model, but current implementations do not take into account the characteristics of scientific data. There is also a limited set of tools available that facilitate and simplify cross-site management and access of clouds. Current scientific workloads at HPC centers are unlikely to be able to leverage current cloud offerings due to the performance impact. HPC centers should examine mechanisms to provide *on-demand, customized environments* to fill this gap.

8. ACKNOWLEDGMENTS

This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357, funded through the American Recovery and Reinvestment Act of 2009. Research at NERSC was funded by the Department of Energy from The American Recovery and Reinvestment Act of 2009 under contract number DE-AC02-05CH11231. The authors would like to thank John Shalf, Keith Jackson, Harvey Wasserman, Krishna Muriki, Shreyas Cholia, Jeff Broughton, Kathy Yelick, Nick Wright, Brent Draney, Jared Wilkening, Gabe West, Ed Holohan, Doug Olson, Jan Balewski, STAR collaboration, K. John Wu, and Alex Sim.

9. REFERENCES

[1] The futuregrid project. `https://portal.futuregrid.org/`.

[2] A Performance Comparision of Massively Parallel Sequence Matching Computations on Cloud Computing Platforms and HPC Clusters Using Hadoop. In *Using clouds for parallel computations in systems biology, Held in conjunction ACM/IEEE SC2009 Conference on High Performance Computing, Networking, Storage and Analysis, Portland Oregon*, November 2009.

[3] G. Aldering, G. Adam, P. Antilogus, P. Astier, R. Bacon, S. Bongard, C. Bonnaud, Y. Copin, D. Hardin, F. Henault, D. A. Howell, J. Lemonnier, J. Levy, S. C. Loken, P. E. Nugent, R. Pain, A. Pecontal, E. Pecontal, S. Perlmutter, R. M. Quimby, K. Schahmaneche, G. Smadja, and W. M. Wood-Vasey. Overview of the Nearby Supernova Factory. In J. A. Tyson & S. Wolff, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4836 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 61–72, Dec. 2002.

[4] bcfg2. `http://trac.mcs.anl.gov/projects/bcfg2`.

[5] J. Brevik, D. Nurmi, and R. Wolski. Predicting Bounds on Queueing Delay for Batch-scheduled Parallel Machines. In *Principles and Practice of Parallel Programming (PPoPP 2006)*. ACM, 2006.

[6] Cern Virtual Machines. `http://rbuilder.cern.ch/project/cernvm/releases`.

[7] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI '04*, pages 137–150, 2004.

[8] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The Cost of Doing Science on the Cloud: The Montage Example. In *Proceedings of SC'08*, Austin, TX, 2008. IEEE.

[9] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The cost of doing science on the cloud: the montage example. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12. IEEE Press, 2008.

[10] 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):2–34, 2008.

[11] M. Fenn, S. Goasguen, and J. Lauret. Contextualization in practice: The clemson experience. In *ACAT Proceedings*, 2010.

[12] S. Hazelhurst. Scientific computing using virtual high-performance computing: a case study using the Amazon elastic computing cloud. In *Proceedings of 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*, pages 94–103. ACM, 2008.

[13] Heckle. `http://trac.mcs.anl.gov/projects/Heckle`.

[14] HPSS Website. `http://www.hpss-collaboration.org/`.

[15] K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. Wasserman, and N. Wright. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. In *2nd IEEE International Conference on Cloud Computing Technology and Science*, 2010.

[16] G. Juve and E. Deelman. Scientific workflows and clouds. *Crossroads*, 16:14–18, March 2010.

[17] K. Keahey. Cloud Computing for Science. In *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, page 478. Springer-Verlag, 2009.

[18] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, and M. Tsugawa. Science clouds: Early experiences in cloud computing for scientific applications. *Cloud Computing and Applications*, 2008, 2008.

[19] K. Keahey and T. Freeman. Science Clouds: Early Experiences in Cloud Computing for Scientific Applications. In *Cloud Computing and its Applications (CCA)*, 2008.

[20] K. Keahey, T. Freeman, J. Lauret, and D. Olson. Virtual workspaces for scientific applications. In *Journal of Physics: Conference Series*, volume 78, page 012038. Institute of Physics Publishing, 2007.

[21] J. Lauret, M. Walker, S. Goasguen, and L. Hajdu. From Grid to cloud, the STAR experience. *SciDAC 2010 Proceedings*, 2010.

[22] J. Li, D. Agarwal, M. Humphrey, C. van Ingen, K. Jackson, and Y. Ryu. eScience in the Cloud: A

MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform. In *IPDPS*, 2010.

[23] R. Masud. High Performance Computing with Clouds.

[24] 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*, pages 17–20. ACM, 2009.

[25] Nist cloud. `http://csrc.nist.gov/groups/SNS/cloud-computing/`.

[26] F. T. C. on a Cloud Computing Cluster: Using Parallel Virtualization for Large-Scale Climate Simulation Analysis. Daren Hasenkamp and Alex Sim and Michael Wehner and Kesheng Wu. In *2nd IEEE International Conference on Cloud Computing Technology and Science*, 2010.

[27] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. An early performance analysis of cloud computing services for scientific computing. *Delft University of Technology, Tech. Rep*, 2008.

[28] M. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel. Amazon S3 for science grids: a viable solution? In *Proceedings of the 2008 international workshop on Data-aware distributed computing*, pages 55–64. ACM, 2008.

[29] L. Ramakrishnan, D. Nurmi, A. Mandal, C. Koelbel, D. Gannon, T. M. Huang, Y.-S. Kee, G. Oberteli, K. Thyagaraja, R. Wolski, A. Yarkhan, and D. Zagorodnov. VGrADS: Enabling e-Science Workflows on Grids and Clouds with Fault Tolerance. In *Proceedings of the ACM/IEEE SC2009 Conference on High Performance Computing, Networking, Storage and Analysis, Portland, Oregon*, Portland, Oregon, November 2009.

[30] J. Rehr, F. Vila, J. Gardner, L. Svec, and M. Prange. Scientific computing in the cloud. *Computing in Science and Engineering*, 99(PrePrints), 2010.

[31] M. C. Schatz. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, pages 1363–1369, June 2009.

[32] G. Wang and T. E. Ng. The impact of virtualization on network performance of amazon ec2 data center. In *Proceedings of IEEE INFOCOM*, 2010.

[33] From Clusters To Clouds: xCAT 2 Is Out Of The Bag. `http://www.linux-mag.com/id/7230/`.