SciCloud: Cloud for high-performance scientific computing

Joseph Anthony C. Hermocilla

1 Thesis

High-performance scientific computing can be done over cloud infrastructure and can be guaranteed a performance that is comparable to bare metal machines currently used in supercomputers or dedicated commodity clusters.

2 Rationale

Since the digital computer was invented, it has been an important tool in doing science. In fact, without it, progress in science will be quite slow. Several scientific breakthroughs have been made possible because of digital computers. From the sequencing of the human genome to the discovery of the Higg's Boson particle, the digital computer has its very important role. From simple arithmetic to modeling biological and physical systems, the digital computer is able to deliver the result of experiments in an accurate and timely manner.

Scientific applications, such as equation solvers and simulations, are very CPU-intensive and Memory-intensive. These applications, because of their heavy computational and storage requirements, are usually run in high-performance computing (HPC) infrastructures using supercomputers or commodity clusters for distributed computations. Setting up such infrastructures requires a lot of initial capital and technical investments. Once the infrastructure has been set up, maintenance costs will also be required to update the hardware and software components. Although these infrastructures are indeed useful, studies have shown that they are underutilized. Their usage depends on the computing demand pattern of scientists and researchers which often make these infrastructures idle. Most HPC clusters in research centers cater to a mix of researchers from different study areas. For example, physicists and chemists may share the same HPC cluster running on Linux operating system. Differences in usage requirements of these researchers affect the overall operation of the cluster. Provisioning the hardware and software requirements needed for scientific computing is thus very challenging because of these factors.

Despite the recent advances in hardware (increased processing speed and memory) and their decreasing costs, typical users do not harness the full capacity of their machines, either because the software they are using does not support them or their computing needs do not require access to the full capacity of their machines. Web browsing and word processing, for example, usually do not require high-performance machines whereas gaming needs them for an enhanced user experience. Another example is the case of laboratory computers used for instruction in academic institutions. Despite having the latest processors and a good amount of memory, they are typically used just for simple programming exercises which do not require the full access to hardware performance. These underutilized machines can be used for scientific computing needs.

Machines used in clusters are usually dedicated, which means that they cannot be used for other general-purpose computing needs. When running tasks, the number of machines used from the clusters used are usually statically set often with the assistance of system administrators managing the cluster. A scientist who needs to perform a simulation need to contact the cluster administrator to schedule the job that must be run.

2.1 What is Scientific Computing?

According to Wikipedia, Scientific Computing, also known as Computational Science, involves the use of computers for solving problems in science. It is concerned with model construction, which is mostly mathematical, as well as quantitative analysis. It is also involved in computer simulation and other forms of computation that are used to solve problems in various scientific disciplines. Scientists develop software that models the system being studied and run them with various parameters. The nature of the models requires a lot computations(floating point operations) and data.

2.2 Desktop Computers, Supercomputers, Clusters, and Grid Computing

The nature of software created for science, involving complex calculations, will require a different approach to their implementation and execution. Desktop computers are useful for small scale simulations but are slow for complex experiments. Supercomputers are computers, way beyond the capabilities of desktop computers, that have a specialized architecture optimized for floating point operations. However, these types of computers are usually very costly. Clusters are a collection of typical desktop computers connected through a local area network and is less costly. They execute scientific applications in a distributed manner usually using distributed shared memory or message passing. Grid computing extends the capabilities of clusters by going beyond the local area network extending through a wide area network, in a totally distributed manner. Grid computing is usually federated.

2.3 What is Cloud Computing?

Cloud computing is a recent buzzword in computing. The diversity of applications running on top of the Internet, from e-commerce and banking to social networks, forces the vendors to address the issue of scale. In the case of social networking sites, for example, the continuously increasing number of users will demand additional physical computing resources to be provisioned. Application developers who develop software can focus on implementing functionality instead on optimizing code to address increasing demand.

The widely accepted definition of cloud computing is from the National Institute of Standards and Technology[4]. It defines cloud computing as

> "a model for enabling ubiquitous, 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 . This cloud model promotes availability and is composed of five essential characteristics , three service models , and four deployment models ."

There are five main characteristics of cloud computing[4]. First, on-demand access allows users to provision computing resources without requiring human interaction from the service provider. Second, the capabilities are available over the network accessible through standard mechanisms. Third, the resources are pooled and serve multiple users in a multi-tenant model. The physical and virtual resources are dynamically assigned and reassigned depending on the demand. Fourth, provisioning of capabilities is elastic to quickly scale out and rapidly released to quickly scale in. Lastly, cloud systems provide a metering capability at some level of abstraction appropriate to the type of service.

Service models for cloud computing include Software-as-a-Service(SaaS), Platform-as-a-Service(PaaS), and Infrastructure-as-a-Service(IaaS)[4]. SaaS allows users to access and use software running on a cloud infrastructure, for example Google Docs. PaaS allows users to deploy their own applications in a cloud infrastructure using programming languages and frameworks supported by the provider. IaaS provides the user with the most basic computing resources such as processor, storage, and network.

There are four ways to deploy a cloud[4]. A private cloud is operated for a single organization which can be managed by the organization itself or a third party and can be housed within or outside the organization's premises. A community cloud is shared by several organizations to support a specific community. Public clouds are available for the general public. Lastly, hybrid clouds is a composition of two or more clouds.

Another notable characterization of cloud computing was developed at the University of California Berkeley[1].

3 Related Work

A few studies have been made regarding the use of cloud for scientific computing. In 2010, Ostermann et. al. conducted a performance analysis of EC2 cloud computing services for scientific computing using microbenchmarks and kernels. In 2011, Iosup et. al. conducted a performance analysis of cloud computing services for Many-Tasks Scientific Computing[2]. In 2012, Zhao et. al. presented an early effort in designing and building CloudDragon, a scientific cloud platform which is based on OpenNebula. Most recently, Ludescher et. al. developed a cloud-based execution frameworks for scientific problem solving environments which combines a private and public cloud setup[3].In 2010, Schad et. al. performed runtime measurements in the cloud and noted the variability due to the multitenant nature of the cloud[6].

4 Objectives

The main objective of this research is to develop cloud computing solutions to improve high-performance scientific computing. Specifically, this work aims to

1. evaluate existing public cloud offerings (Amazon EC2, Microsoft Azure) for applicability in scientific computing tasks;
2. develop and deploy a private cloud over commodity desktop computers which satisfy scientific applications requirements;
3. develop a novel framework and platform to make it easy and efficient for scientists to develop scientific applications; and
4. deploy a private cloud with software for scientific applications.

5 Methodology

To achieve Objective 1

1. Obtain accounts from Amazon EC2 and Microsoft Azure.
2. Review existing benchmarks and execute them on the platforms mentioned above.

To achieve Objective 2

1. Identify key characteristics of scientific applications in terms of computation, storage, and network.
2. Deploy a 5-node private cloud setup for testing.
3. Study internal architecture of existing cloud computing frameworks (Eucalyptus[5] and OpenStack[7]) to identify modules that can be modified to support the requirements of scientific applications.

4. Design and implement a cloud system with an architecture and implementation specific to support scientific applications.

To achieve Objective 3

1. Identify key characteristics of scientific applications in terms of architectural and data representation.
2. Identify patterns in creating scientific applications and develop software abstractions.
3. Create an application programming interface (API) for scientific applications.
4. Create sample applications using the newly created API.

To achieve Objective 4

1. Survey a group of scientists to identify their most commonly used scientific applications.
2. Partially implement the features of the identified applications using the API developed in Objective 3
3. Deploy the application as a Software-as-a-Service.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[2] A Iosup, S Ostermann, M N Yigitbasi, R Prodan, T Fahringer, and D H J Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):931–945, June 2011.

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

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

[5] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The eucalyptus open-source cloud-computing system. pages 124–131. IEEE, 2009.

[6] JÃ¶rg Schad, Jens Dittrich, and Jorge-Arnulfo QuianÃ©-Ruiz. Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proceedings of the VLDB Endowment*, 3(1-2):460–471, 2010.

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