The Open Cloud Testbed: Supporting Open Source Cloud Computing Systems Based on Large Scale High Performance, Dynamic Network Services

Robert Grossman[1, 2], Yunhong Gu[1], Michal Sabala[1], Colin Bennet[2],
Jonathan Seidman[2], and Joe Mambratti[3]

[1] National Center for Data Mining, University of Illinois at Chicago
[2] Open Data Group
[3] International Center for Advanced Internet Research, Northwestern University

Abstract. Recently, a number of cloud platforms and services have been developed for data intensive computing, including Hadoop, Sector, CloudStore (formerly KFS), HBase, and Thrift. In order to benchmark the performance of these systems, to investigate their interoperability, and to experiment with new services based on flexible compute node and network provisioning capabilities, we have designed and implemented a large scale testbed called the Open Cloud Testbed (OCT). Currently OCT has 120 nodes in 4 data centers: Baltimore, Chicago (two locations), and San Diego. In contrast to other cloud testbeds, which are in small geographic areas and which are based on commodity Internet services, the OCT is a wide area testbed and the 4 data centers are connected with a high performance 10Gb/s network, based on a foundation of dedicated lightpaths. This testbed can address the requirements of extremely large data streams that challenge other types of distributed infrastructure. We have also developed several utilities to support the development of cloud computing systems and services, including novel node and network provisioning services, a monitoring system, and an RPC system. In this paper, we describe the OCT concepts, architecture, infrastructure, a few benchmarks that were developed for this platform, interoperability studies, and results.

1 Introduction

Cloud computing has become quite popular during the last few years, in part because it provides a practical solution to multiple types of application requirements. First, the popular cloud computing services are very easy to use. Users can request computing resources on demand from cloud service providers. Also, most users find the Hadoop Distributed File System (HDFS) and the Hadoop implementation of MapReduce [8] very easy to use compared to traditional high performance computing programming frameworks, such as MPI. Second, basic cloud facilities can be readily deployed. The basic unit consists of racks of standard compute servers. Third, the payment model provides advantages to communities with variable resource requirements. For example, it allows for quick implementation of processing resources without an upfront capital investment.

T. Doulamis et al.: (Eds.): GridNets 2009, LNICST 25, pp. 89–97, 2010.

Basically, there are three types of cloud software systems (Figure 1): 1) the low level resource manager and provider (Amazon EC2 [5], Eucalyptus [4]), which has been called Infrastructure as a Service or IaaS; 2) distributed storage and data processing services such as those provided by Hadoop [2], CloudStore [3], and Sector/Sphere [1] that can be used to build data intensive applications (Platform as a Service or PaaS); and 3) domain specific software, or Software as a Service (SaaS) such as Google Docs.

As the number of different cloud services grows, it has become clear that potential users can benefit from an environment that could be used for benchmarking different systems and testing their interoperability. The Open Cloud Testbed (OCT) was designed to benchmark cloud systems, to investigate their interoperability, and to experiment with implementations on novel infrastructure, such as large scale high performance optical networks. Also, networks are integrated as "first class" controllable,

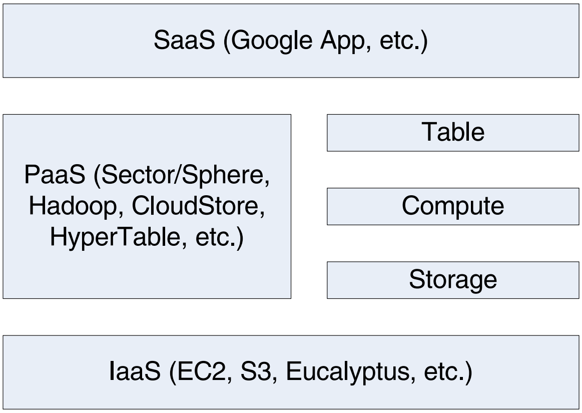

SaaS (Google App, etc.)
PaaS (Sector/Sphere, Hadoop, CloudStore, HyperTable, etc.)
Table
Compute
Storage
IaaS (EC2, S3, Eucalyptus, etc.)

Fig. 1. Cloud Computing Stack

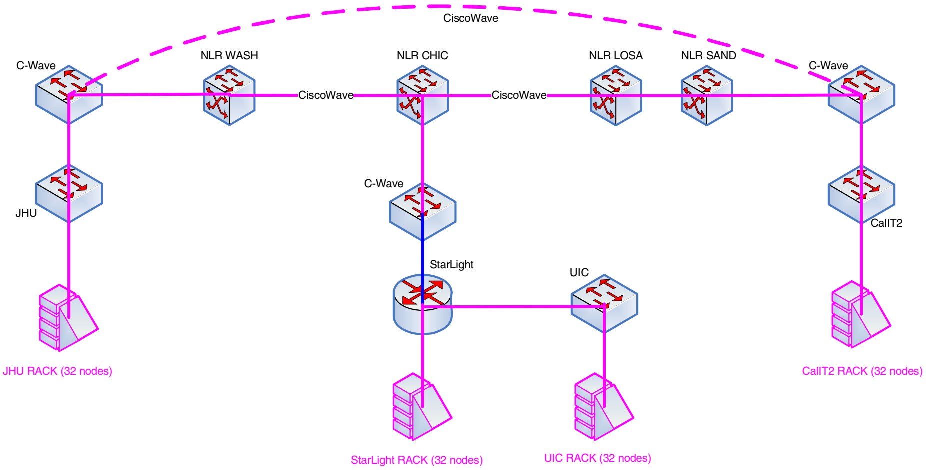


Fig. 2. The Open Cloud Testbed System Diagram

adjustable resources not merely as external resources. We have installed and tested Eucalyptus, Hadoop, CloudStore (KosmosFS), Sector/Sphere, and Thrift with various applications.

In addition, we have also developed network libraries, monitoring systems, and benchmark suites to support the development and experimental studies of cloud computing stacks.

In this paper, we introduce the OCT concept, architecture, infrastructure, the software we built to utilize the testbed, the experimental studies we conducted with various cloud software, and preliminary results of those studies.

2 The Open Cloud Testbed

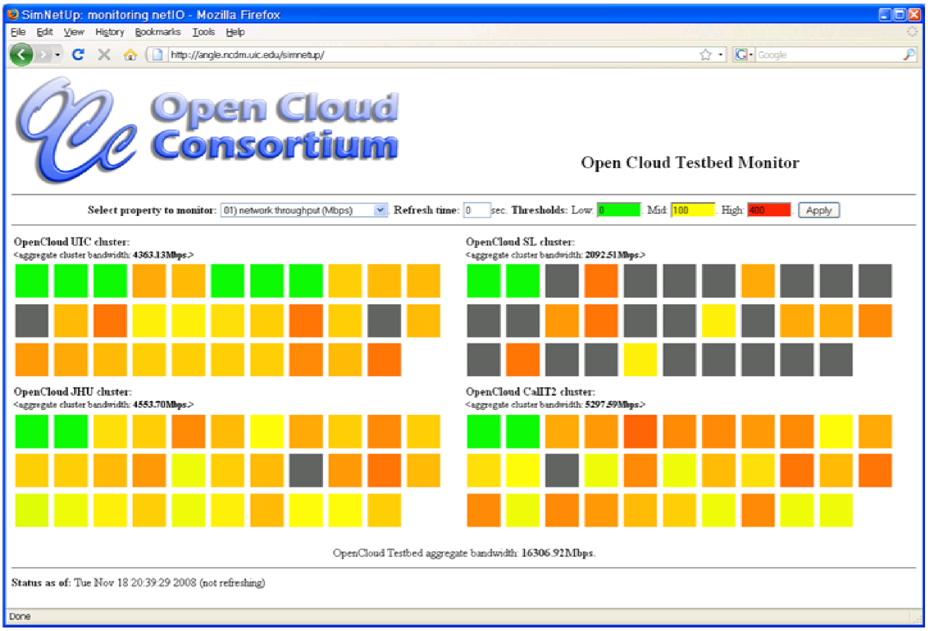
2.1 Concepts and Objectives

The OCT architecture envisions the emergence of powerful large scale applications supported by services and processes based on highly distributed, integrated facilities and infrastructure. This infrastructure is an extremely flexible programmable platform that enables new functions and capabilities to easily be created and implemented. OCT represents a departure from existing clouds is several ways. For example, as its name implies, it is based on a concept of interoperability and openness. Also, traditional clouds use fairly generic common components and protocols across their services and infrastructure. The OCT architecture incorporates high performance services, protocols, and infrastructure at all levels. Instead of using the commodity Internet, it uses a national high performance 10 Gb/s network based on extremely fast transport protocols supported by dedicated light paths. Although such capabilities are fairly rare today, this approach is being used to model future distributed infrastructure, which will provide much more capacity and capabilities than current systems. For example, as commonly implemented, clouds do not support large data streams well. In contrast, OCT is being designed not only to manage millions of small streams and small amounts of information but also extremely large data sets.

The objectives of the OCT initiative extend beyond creative novel high performance capabilities. Another important research objective is to develop standards and frameworks for interoperating between different cloud software. Currently, we have tested Eucalyptus, CloudStore (formerly KosmosFS), Hadoop, Sector/Sphere, and Thrift. In particular, we developed an interface so that Hadoop can use Sector as its storage system.

2.2 The OCT Infrastructure

As illustrated in Figure 2, currently there are 4 racks of servers in OCT, located in 4 data centers at Johns Hopkins University (Baltimore), StarLight (Chicago), the University of Illinois (Chicago), and the University of California (San Diego). Each rack has 32 nodes. Each node has dual dual-core AMD 2.4GHz CPU, 12GB memory, 1TB single SATA disk, and dual 1GE NICs. Two Cisco 3750E switches connect the 32 nodes, which then connects to the outside by a 10Gb/s uplink.


SimNetUp: monitoring netIO - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://angle.ncdm.uic.edu/simnetup/
Google
Open Cloud Consortium
Apply
Done

Open Cloud Testbed Monitor

Select property to monitor: 01) network throughput (Mbps) Refresh time: 0 sec. Thresholds: Low: 0 Mid: 100 High: 400

OpenCloud UIC cluster:
<aggregate cluster bandwidth: **4363.13Mbps**.>

OpenCloud SL cluster:
<aggregate cluster bandwidth: **2092.51Mbps**.>

OpenCloud JHU cluster:
<aggregate cluster bandwidth: **4553.70Mbps**.>

OpenCloud CalIT2 cluster:
<aggregate cluster bandwidth: **5297.59Mbps**.>

OpenCloud Testbed aggregate bandwidth: 16306.92Mbps.

Status as of: Tue Nov 18 20:39:29 2008 (not refreshing)

Fig. 3. The Open Cloud Testbed monitoring and visualization system

In contrast to other testbeds, the OCT utilizes wide area high performance networks, not the familiar commodity Internet. There is 10Gb/s connection (provided by the CiscoWave national testbed infrastructure, which spans the US east, west, north and south) between any two data centers. The majority of experimental studies we perform extend over all four geographically distributed racks. Almost all other cloud testbeds operate each data center locally and independently.

We are currently installing two more racks (60 nodes) and expect to install two additional racks by the end of 2009. By then the OCT will have about 250 nodes and 1000 cores. We will also extend the 10GE network to MIT Lincoln Lab (Cambridge) and Pittsburgh Supercompter Center/Carnegie Mellon University (Pittsburgh).

3 Monitoring and Visualization

When using distributed resources, managing and monitoring the OCT requires substantial time as the testbed grows larger. Management costs can increase sharply as the number of sites increases. To mitigate this effect, we have developed a simple but effective monitoring and real time visualization system to track the behavior of every node in the OCT [11]. The OCT monitoring system reads the resource utilization (including CPU, memory, disk, NIC, etc.) on each node and a web-based visualization allows us to easily and directly monitor the entire system.

In addition, the monitoring system also helps with benchmarking and debugging cloud software. The visualization effectively indicates the real time status of the testbed (for example, if it is fully utilized and if the load is balanced across the testbed). This capability has proved invaluable when debugging cloud services.

A snapshot of the web-based visualization can be found in Figure 3. Each block represents a server node, while each group of blocks represent a cluster. The color of each block represents the usage of a particular resource, in this case, network IO throughput. Color on the green/light side means the machine is idle; color on the red/dark side means the machine is busy.

In the Sector/Sphere software, we also have built-in a monitoring system that is used to improve load balancing and to remove nodes and/or network segments that exhibit poor performance. While the OCT monitoring system reads per node information, the built-in monitoring system of Sector reads the resource usage of the Sector process on each node and the network utilization based on the topology.

Sector assumes that the underlying network has a hierarchical topology, such as the one used by OCT. Based on this topology, Sector computes the aggregate network throughput on each link, in addition to each node. This process helps to identity a malfunctioning link or node and Sector can therefore remove such underperforming resources from the system.

The OCT architecture is based on an assumption that services are based on flexible, not static, foundation resources. The architecture provides for continuous monitoring of conditions, analysis, and dynamic change, including at the level of network resources. OCT leverages and builds on recent research and development trends in architecture and technologies that provide for dynamically provisioned network resources [13]. Using such capabilities for OCT implementations demonstrates the importance of designing dynamic vs. static network resources.

4 GMP: A Messaging Protocol

We have also designed and developed a unique high performance messaging protocol called GMP (Group Messaging Protocol) for Sector and other distributed systems. GMP allows an application to send a message quickly and reliably to another node. High performance messaging is essential for rapid reconfigurations of core resources under changing conditions. The protocol is suitable for delivering small control messages in distributed systems.

GMP is a connection-less protocol, which uses a single UDP port and which can send messages to any GMP instances or receive messages from other GMP instances. Because there is no connection setup required, GMP is much faster than TCP, which requires a connection set up between the involved nodes. On the other hand, UDP is unreliable. GMP, which is built on top of UDP, does not have any virtual connection, but maintains a list of states for each peer addresses to which it sends messages or from which it receives messages.

Every GMP message contains a session ID and a sequence number. Upon receiving a message, GMP sends back an acknowledgment; if no acknowledgment is received, the message will be sent again. The sequence number is used to make sure that no duplicated message will be delivered. The session ID is used to differentiate messages from the same address but different processes (e.g., if one process is restarted it will use a different session ID). If the message size is greater than a single UDP packet can hold, GMP will set up a UDT [12] connection to deliver the large message. However, we expect such a situation to be rare for GMP.

In Sector, we also developed a light-weight high performance RPC mechanism on top of GMP. The RPC library simply sends out a request in a GMP message and then it waits for the response to come back.

5 Benchmarks

We developed various standard benchmark applications to evaluate cloud software. Certain benchmarks designed previously for other systems, such as Terasort, have also been adopted.

In this section we described a benchmark called MalStone [14], which is specially designed to evaluate the ability of cloud systems in the support of distributed data intensive applications.

MalStone is a stylized analytic computation that requires the analysis of log files containing events about computers (entities) visiting web sites of the following form:

| Event ID | Timestamp | Site ID | Compromise Flag | Entity ID

The assumption is that some of the entities that visit certain web sites become compromised. In order to find out the bad sites that compromise computers, the benchmark application requires that a ratio be computed *for each site* that for a specified time window measures the percent of entities that become compromised at any time in the window.

Two sub-tasks are formed. MalStone-A computes the overall ratio per site. Malstone-B computes a series of windows-based ratio per site.

This type of computation requires only a few lines of code if the data is on a single machine (and can be done easily in a database). On the other hand, if the data is distributed over the nodes in a cloud, then we have found this type of computation turns out to be a useful benchmark for comparing different storage and compute services.

An example of a situation that might result in these types of log files is what are sometimes termed drive-by exploits [10]. These incidents result when users visit web sites containing code that can infect and compromise vulnerable systems. Not all visitors become infected but some do.

We have also developed a data generator for the MalStone benchmark called MalGen.

6 Experimental Studies

In this section, we describe several experimental studies we currently conduct on OCT.

In the first series of experiments, we used MalGen to generate 500 million 100-byte records on 20 nodes (for a total of 10 billion records or 1 TB of data) in the Open Cloud Testbed and compared the MalStone performance using: 1) the Hadoop Distributed File System (HDFS) with Hadoop's implementation of MapReduce; 2) the Hadoop HDFS with Streams and MalStone coded in Python; and, 3) the Sector Distributed File System and MalStone coded in Sector's User Defined Functions (UDF). The results are below (Table 1):

Table 1. Hadoop version 0.18.3 and Sector version 1.20 were used for these tests. Times are expressed in minutes (m) and seconds (s).

	Malstone-A	MalStone-B
Hadoop MapReduce	454m 13s	840m 50s
Hadoop Streams with Python	87m 29s	142m 32s
Sector/Sphere	33m 40s	43m 44s

Sector/Sphere benefit significantly from its data movement optimization (bandwidth load balancing and UDT data transfer) and it performs much faster than Hadoop.

We have conducted extensive experimental studies with different versions of Sector and Hadoop. The benchmark applications include distributed sorting and MalStone.

Table 2. This table compares the performance of Hadoop and Sector for a computation performed in one location using 28 nodes and 4 locations using 7 nodes each

	28 Local Nodes	7 * 4 Distributed Nodes	Wide Area Penalty
Hadoop (3 replicas)	8650	11600	34%
Hadoop (1 replica)	7300	9600	31%
Sector	4200	4400	4.7%

In the second series of experiments, we used MalGen to 15 billion on 28 nodes in one location and compared these results to the same computation performed when the nodes where distributed over four locations in the testbed. See Table 2. The experiment shows the impact of wide area networks on the performance of such applications. The performance penalty on Hadoop is 31~34%, while Sector suffers a 4.7% performance drop.

There are two major reasons for the better performance of Sector over wide area networks. First, Sector employs a load balancing mechanism to smoothly distribute the network traffic within the system. Second, Sector uses UDT [12] for data transfer. UDT is a high performance protocol that performs significantly better than TCP over wide area networks. The limitations of TCP are well documented [13].

7 Related Testbeds

There are several other testbeds that serve similar or related goals as that of OCT. The most closely related one in the Open Cirrus Testbed [9]. Open Cirrus consists of 6 sites with various number of nodes between 128 and 480 per site. While Open Cirrus contains more nodes than OCT, there is no data exchange between any two Open Cirrus sites. That is, Open Cirrus is designed for systems that run in a single data center.

Another cloud computing testbed is the Google-IBM testbed, which is similar to Open Cirrus, but it is smaller in scale.

Amazon's EC2 provides an economical alternative for certain cloud computing research. However, in EC2 users cannot control data locality and network configuration, thus it limits the value of related system research. In addition, while EC2 is inexpensive for temporary use, it actually poses a higher expense for long term system research purpose.

Other computing testbeds such as the TeraGrid were designed for mostly application research. Their usage on cloud computing system research were very limited.

8 Conclusion

Cloud computing has proven to be an important resource for many types of applications, from those oriented to consumers to those focused on the enterprise to those that support large scale science. This trend is expected to continue to the foreseeable future. However, more progress is required to fully utilize the cloud model. The OCT initiative was established in order to advance the state of cloud architecture and implementations. By benchmarking the performance of existing systems, investigating their interoperability, and experimenting with new services based on flexible compute node and network provisioning capabilities, more of the potential promised by the cloud model can be realized. Therefore, we have created and conducted experiments on the Open Cloud Testbed (OCT) a wide area testbed, based on a foundation of dedicated lightpaths. This testbed was used to investigate multiple services, protocols, processes and components, including novel node and network provisioning services, a monitoring system, and an RPC system. The results are extremely promising, and they indicate that the concepts described here are an important direction for additional research projects. Also, we believe that OCT is an important step to standardize cloud computing software and it is also a unique platform to benchmark the performance and interoperability of different clouds. By building on the concepts presented here, it may be possible to achieve significant advances in the development of next generation clouds over the next few years.

References

[1] Gu, Y., Grossman, R.: Sector and Sphere: The Design and Implementation of a High Performance Data Cloud. Theme Issue of the Philosophical Transactions of the Royal Society A: Crossing Boundaries: Computational Science, E-Science and Global E-Infrastructure 367(1897), 2429–2445 (2009)

[2] Hadoop, http://hadoop.apache.org/core/

[3] CloudStore, http://kosmosfs.sourceforge.net/

[4] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-source Cloud-computing System. In: Proceedings of Cloud Computing and Its Applications, Chicago, Illinois (October 2008)

[5] Amazon EC2 and S3, `http://aws.amazon.com/`

[6] Thrift, `http://developers.facebook.com/thrift/`

[7] Ghemawat, S., Gobioff, H., Leung, S.-T.: The Google File System, pub. In: 19th ACM Symposium on Operating Systems Principles, Lake George, NY (October 2003)

[8] Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI 2004: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA (December 2004)

[9] HP Technical Report, HP-2009-122: Open Cirrus Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research

[10] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., Modadugu, N.: The Ghost In The Browser - Analysis of Web-based Malware. In: Proceedings of the First Workshop on Hot Topics in Understanding Botnets, HotBots (2007)

[11] OCT Monitor, `http://angle.ncdm.uic.edu/simnetup/`

[12] Gu, Y., Grossman, R.L.: UDT: UDP-based Data Transfer for High-Speed Wide Area Networks. Computer Networks 51(7) (May 2007)

[13] Travestino, F., Mambretti, J., Karmous-Edwards, G.: GridNetworks: Enabling Grids With Advanced Communication Services. Wiley, Chichester (2006)

[14] Bennett, C., Grossman, R., Seidman, J.: Open Cloud Consortium Technical Report TR-09-01, MalStone: A Benchmark for Data Intensive Computing (April 2009)