Cloud Computing: Data Centers

Author: Bronson Edralin

Date: 11/17/14

Abstract: With the availability of open source software such as OpenStack and Apache Hadoop, data centers can be created with ordinary computers or servers. We will take a look at the history, architecture, and functionalities of the revolutionary open software.

1 Introduction

Today, Data Centers can be created using a cluster of ordinary computers and servers by using open source software such as OpenStack and Apache Hadoop. In Section 2, we will discuss OpenStack and its history. We will also look into its architecture that contains components called Nova, Swift, Cinder, Neutron and Sahara. We will also explore other remaining components of OpenStack [4].

In Section 3, we will discuss Apache Hadoop and its history. Then we will explore the architecture and later look into what Map Reduce is and how it was used for Job Tracker and Task Tracker. We will also discuss the fair scheduling algorithm that is used by the Apache Hadoop [4].

2 What is OpenStack

OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds. Backed by some of the biggest companies in software development and hosting, as well as thousands of individual community members. Many feel that OpenStack is the future of cloud computing. Currently, OpenStack is being managed by the OpenStack Foundation, which is a non-profit overseeing both development and community-building around the project [6].

2.1 History

In July 2010, OpenStack was a pilot project launched by Rackspace and NASA. The purpose behind this project was to provide an open source software that enables any organization to create and offer cloud computing services running on standardized hardware. It is believed that many distributions for OpenStack will be founded just like the Linux distribution has multiple flavors supported by foundations such as RedHat and SUSE. Being an open source project, thousands of developers contribute code, written primarily in Python and are freely available under an Apache 2.0 license [11].

2.2 Nova

OpenStack Compute service is called Nova and it is used for hosting and managing cloud computing systems. As a component based architecture, it enables quicker additions of new features. It is fault tolerant, recoverable and provides API-compatibility with systems like Amazon EC2 [5]. Nova is built on messaging architecture and all of its components can be run on several servers. This kind of architecture will allow components to communicate with each other through a message queue. Deferred objects are used to avoid blocking while a component waits in the message queue for a response. Nova and its components share a centralized SQL-based database. This may be suitable for smaller deployments, but for larger deployments, an aggregation system will have to be set in place to manage the data across multiple data stores. [5]

2.3 Swift

OpenStack Object Storage is called Swift. It is ideal for a cost effective, scale-out storage. It provides a fully distributed, API-accessible storage platform that may be integrated directly into applications or used for backup, archiving and data retention. Object storage is not a traditional file system. It is more like a distributed storage system for static data such as virtual machine images, photo storage, email storage, backups and archives. Having no central master point of control provides greater scalability, redundancy and durability. Objects and files are written to multiple disk drives and spread throughout servers in the data center and the OpenStack software will be responsible for ensuring data replication and integrity across the cluster of servers that are capable of storing petabytes of data. Storage clusters scale horizontally by simply adding new servers. Should a server or hard drive fail, OpenStack will replicate its content from other active nodes to new locations in cluster [7].

2.4 Cinder

OpenStack Block Storage is called Cinder, which allows block devices to be exposed and connected to compute instances for expanded storage, better performance and integration with enterprise storage platforms, such as NetApp, Nexenta and SolidFire [7]. With OpenStack compute instances, OpenStack provides persistent block level storage devices for use. This block storage system manages the creation, attaching and detaching of block devices to servers. With the block storage volumes fully integrated into OpenStack Compute and the Dashboard, it will allow users to manage their own storage needs over the cloud. In addition to simple Linux server storage, it has storage support for numerous storage platforms including Ceph, NetApp, Nexenta, SolidFire and Zadara. Block storage is appropriate for performancesensitive scenarios such as database storage, expandable file systems, or providing a server with access to raw block level storage. Snapshots can be used to restore or create new block storage volume [7].

2.5 Neutron

Neutron is an OpenStack networking project focused on delivering networking as a service. Neutron has replaced the original networking application program interface (API) in OpenStack. It was designed to address deficiencies that existed in the network technology found in cloud environments as well as the lack of tenant control over the network topology and addressing [10].

The reality is that cloud environments are starting to put enormous strain on the existing networks. Neutron provides a way for organizations to relieve the stress on the network in cloud environments to make it easier to deliver networking as a service in the cloud. It is designed to provide an option for network operators to enable different technologies via the Quantum API and it also lets tenants create multiple private networks and control the IP addressing on them. Because of this, organizations now have more control over security and compliance policies, quality of service (QoS) and monitoring and troubleshooting, and more [10].

2.6 Sahara

OpenStack Elastic Map Reduce is called Sahara, which allows users to capture, synthesize and quantify data into business value. Sahara's mission is to provide a scalable data processing stack and associated management interfaces and Sahara meets this mission by providing the ability to create and manage Apache Hadoop clusters and easily run workloads across them. With full cluster lifecycle management, provisioning, scaling, and termination, Sahara allows the user to select different Hadoop versions, cluster topology and node hardware details [9].

2.7 Remaining Components of OpenStack

Other components of OpenStack include Horizon, Keystone, Glance, Ceilometer and Heat [6]. Horizon is the dashboard behind OpenStack and it is the only graphical user interface to OpenStack. Developers can access all of the components of

OpenStack individually through an application programming interface (API) but the dashboard provides system administrators a way to view and manage what is going on in the cloud. Keystone provides identity services for OpenStack where it is essentially a central list of all the users of the OpenStack cloud, mapped against all of the services provided by the cloud, which they have permission to use. By providing multiple means of access, developers can easily map their existing user access methods against Keystone. Glance provides image services to OpenStack. These images refer to images or virtual copies of hard disks. Ceilometer provides telemetry services, which allow the cloud to provide billing services to individual users of the cloud. It can also keep track of each user's system usage of each of the various components of an OpenStack cloud. Heat is the orchestration component of OpenStack, which allows developers to store the requirements of a cloud application in a file that defines what resources are necessary for that application [6].

3 What is Apache Hadoop

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is basically a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from very few servers to thousands of machine, each offering local computation and storage. The library is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures. The Apache Hadoop project include these modules:

Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop Yarn, and Hadoop MapReduce [3].

3.1 History

Hadoop was created by Doug Cutting, the creator of Apache Lucene, which was the widely used text search library. Hadoop has its origins in Apache Nuche, an open source web search engine, itself a part of the Lucene project. Building a web search engine from scratch is very complex and expensive. Mike Cafarella and Doug Cutting estimated a system supporting a 1 billion page index would cost around half a million dollars in hardware with a monthly running cost of $30,000 [12]. In 2004, they set about writing an open source implementation, the Nutch Distributed Filesystem (NDFS) [12]. In 2005, all major Nutch algorithms had been ported to run using MapReduce and NDFS [12]. In April 2008, Hadoop broke a world record to become the fastest system to sort a terabyte of data. Running on a 910-node cluster, Hadoop sorted one terabyte in 209 seconds. Later in May 2009, a team at Yahoo reported using Hadoop to sort one terabyte in 62 seconds [12].

3.2 Architecture

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. In comparison to existing distributed file systems, HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware HDFS provides high throughput access to application data and is suitable for applications that have large data. HDFS has been designed to be easily portable from one platform to another [2].

HDFS has a master/slave architecture where an HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. There a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file may be split into one more blocks and these blocks may be stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing and renaming files and directories. It also determines the mapping of blocks to DataNodes. In turn, the DataNodes remain responsible for managing read and write requests from the file system's clients. With instruction from NameNode, DataNodes also perform block creation, deletion, and replication [2].

3.3 Job Tracker and Task Tracker: Map Reduce Engine

Before we look into the Map Reduce Engine, it is important to know what MapReduce is. MapReduce is a programming model, which is used to process large data sets in a batch-processing manner. It is composed of a Map function that performs filtering and sorting such as sorting students by last name into queues, one queue for each name. It is also composed of a Reduce function that performs a summary operation such as counting the number of students in each queue, yielding name frequencies.

Above the file systems comes the MapReduce engine, which consists of one JobTracker, to which client applications submit MapReduce jobs. The JobTracker pushes work out to available TaskTracker nodes in the cluster. With a rack-aware file system, the JobTracker knows which node contains the data and which machines are nearby. If work cannot be hosted on the actual node that contains the data, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, the part of the job will be rescheduled. The TaskTracker on each node spawns off a separate Java Virtual Machine (JVM) process to prevent the TaskTracker itself from failing if the running job crashes the JVM. The TaskTracker checks on the JobTracker every few miutes to see its status. Some checkpointing was added to the process where the JobTracker records what is up to in the file system so data may not be totally lost if a fail occurs [8].

3.4 Schedulers

The FairScheduler is a pluggable scheduler for Apache Hadoop that allows Yet Another Resource Negotiator (YARN) applications in large clusters to share resources. Fair scheduling is a method of assigning resources to jobs such that all jobs get, on average, an equal share of resources over time. The fair scheduler organizes jobs into pools and divide resources fairly between pools. Within each pool, jobs may be scheduled using either fair sharing or First In First Out (FIFO) scheduling. Fair Scheduler can limit the number of concurrent running jobs per user and per pool. This is useful in ensuring a user cannot fill up the disk space on a cluster by overloading it with hundreds of jobs at once. The Fair Scheduler can also limit the number of concurrent running tasks per pool. This can be useful when jobs have a dependency on an external service like a database or web service that could be overloaded if too many map or reduce tasks are run at once [1].

4 Conclusion

Data Centers incorporate much different functionality and are very useful for our daily needs. With open source software such as OpenStack and Apache Hadoop, people can create cloud-computing clusters without hurting the budget. By building a cluster of servers or computers, we can create a data center. The Apache OpenStack distribution is built to suit the needs of what my goal is and contains various modules that will handle the computing, storage, networking and map reducing. They are also built for scalability, which means they are able to be compatible with one another and work together to optimize the computing cluster. With consumer computers, we are finally able to create data centers.

5 References

[1] The Apache Software Foundation. (2013, August). *Fair Scheduler* [Online]. Available: http://hadoop.apache.org/docs/r1.2.1/fair_scheduler.html

[2] The Apache Software Foundation. (2013, August). *HDFS Architecture Guide* [Online]. Available: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[3] The Apache Software Foundation. (2014, September). *Welcome to Apache Hadoop* [Online]. Available: http://hadoop.apache.org/

[4] "EE468 Writing Assignment 3: Cloud Computing, more specifically data centers" attachment for EE468, University of Hawaii, Fall 2014.

[5] Flux7 Labs. (2014, April). (Tutorial) *What is Nova and how to install & use it? | OpenStack* [Online]. Available: http://blog.flux7.com/blogs/openstack/tutorial-what-is-nova-and-how-to-install-use-it-openstack

[6] Opensource. (2014). *What is OpenStack?* [Online]. Available: http://opensource.com/resources/what-is-openstack

[7] OpenStack. (2014). *OpenStack Storage* [Online]. Available: http://www.openstack.org/software/openstack-storage/

[8] S. Bappalige. (2014, August). *An introduction to Apache Hadoop for big data* [Online]. Available: http://opensource.com/life/14/8/intro-apache-hadoop-big-data

[9] S. Cohen. (2014, May). *Sahara: OpenStack Elastic Hadoop on Demand* [Online]. Available: http://redhatstackblog.redhat.com/2014/05/12/sahara-openstack-elastic-hadoop-on-demand/

[10] SDNcentral. (2014). *What is OpenStack Neutron?* [Online]. Available: https://www.sdncentral.com/what-is-openstack-quantum-neutron/

[11] S. Zitzman. (2014, July). *What is OpenStack? A Quick OpenStack Tutorial* [Online]. Available: http://getcloudify.org/2014/07/10/what-is-openstack-tutorial.html

[12] T. White. *Hadoop: The Definitive Guide*. 3rd ed. California: O'Reilly Media, 2012.