HADOOP ON EMC ELASTIC CLOUD STORAGE

Big Data Analytics on Hyperscale Software Defined Storage

ABSTRACT

This White Paper details the way EMC® Elastic Cloud Storage (ECS™) can be used to support the Hadoop data analytics workflow for an enterprise. It describes the core architectural components, and highlights the features an enterprise can use to gain reliable business insights.

May, 2015

REDEFINE

EMC WHITE PAPER

To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, contact your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the EMC Store

Part Number H13999.1

TABLE OF CONTENTS

INTRODUCTION TO HADOOP

Apache Hadoop is an open-source software framework for storage and large-scale processing of data sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users.

The Apache Hadoop framework is composed of the following modules:

- Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
- Hadoop YARN – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of user applications.
- Hadoop MapReduce – a programming model for large-scale data processing.
- Hadoop Ecosystem - an ecosystem of Apache projects, such as Pig, Hive, Sqoop, Flume, Oozie, Spark, HBase, Zookeeper, etc. that extend the value of Hadoop and improve its usability.

HADOOP DISTRIBUTED FILESYSTEM (HDFS)

The Hadoop Distributed File System (HDFS) is a block-based, distributed, scalable, and portable file system written for the Hadoop framework. HDFS splits files into large blocks and distributes the blocks amongst the nodes in the cluster.

The main components of HDFS are:

- **NameNode**: NameNode is the centerpiece of HDFS, which serves as the metadata server for the file system. HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the NameNode's memory structures, thus preventing file-system corruption and reducing loss of data.
- **DataNode**: In HDFS, individual files are broken into blocks of a fixed size. These blocks are stored across a cluster of one or more machines with data storage capacity referred to as DataNodes. DataNodes serve read and write requests to and from clients as directed by the NameNode.

YARN

Apache Hadoop YARN (Yet Another Resource Negotiator) is a cluster management technology which is a key feature of the second-generation Hadoop. Originally described by Apache as a redesigned resource manager, YARN is now characterized as a large-scale, distributed operating system for big data applications.

YARN combines a central Resource Manager that reconciles the way applications use Hadoop system resources with Node Manager agents that monitor the processing operations of individual cluster nodes. Separating HDFS from MapReduce with YARN makes the Hadoop environment more suitable for operational applications that can't wait for batch jobs to finish.

MAPREDUCE

Hadoop MapReduce is a software framework for easily writing applications which process large amounts of data, in parallel, on large clusters of commodity compute nodes.

A MapReduce job comprises multiple Map tasks, which are distributed and processed in parallel. The framework sorts the output of Map tasks, which are then used as inputs for the next phase: Reduce. Both input and output of MapReduce jobs are stored across the cluster on HDFS. The framework also takes care of scheduling tasks, monitoring them and re-executing failed tasks.

By aligning compute tasks as close to the resident data as possible, the framework avoids network bottlenecks caused by data movement. This is one of the core paradigms of the Hadoop architecture.

HADOOP ECOSYSTEM AND VENDORS

Apache Hadoop is an open source project. A number of vendors have developed their own distributions, adding new functionality or improving the code base.

Apart from the basic set of Hadoop components, there are other components - such as Apache HBase, Apache Hive, Apache Pig, Apache Zookeeper, etc.- that are widely used to solve specific tasks, perform co-ordination, speed up computations, provide an SQL interface, optimize routine tasks, etc.

Vendor distributions are designed to overcome issues with the open source edition and provide additional value to customers, with a focus on the following aspects:

- Reliability: The vendors react faster when bugs are detected. They deliver fixes and patches promptly, which makes their solutions more stable.
- Support: A variety of companies provide technical assistance, which makes it possible to adopt the platforms for mission-critical and enterprise-grade tasks.
- Completeness: Very often Hadoop distributions are supplemented with other tools to address specific tasks.

In addition, vendors participate in improving the standard Hadoop distribution by giving back updated code to the open source repository, fostering the growth of the overall community. Apache Hadoop's website lists many commercially supported Hadoop distributions: http://wiki.apache.org/hadoop/Distribution and Commercial Support

The most popular among them are:

- Cloudera (CDH)
- Hortonworks (HDP)
- Pivotal (PHD)
- MapR
- Amazon Elastic MapReduce
- Microsoft Azure (HDInsights)
- IBM BigInsights

HADOOP ARCHITECTURE

The main components of Apache Hadoop are depicted in this block diagram (courtesy Pivotal Inc.). Apart from the core-Hadoop components, commercial vendors (for enterprises and cloud) provide many value-added services in their distribution.

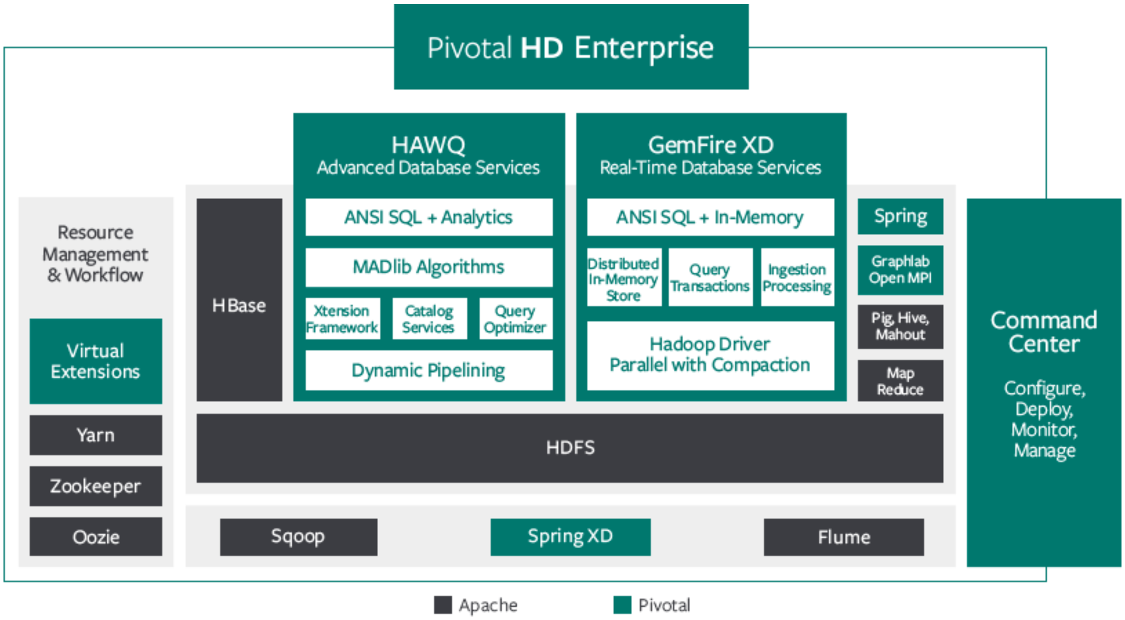

Figure 1. Apache Hadoop Components (Source: Pivotal, Inc.)

LIMITATIONS OF STOCK HDFS

Stock HDFS has the following limitations:

- **Centralized NameNode**. The HDFS file system namespace is managed by a single server and is maintained in memory. The size of namespace that can be managed by the central NameNode is limited by the amount of memory available on the NameNode and performance of the file system is limited by the performance of the NameNode because all NameNode lookups need to be done from a single server.
- **File system reliability**. Prior to Hadoop 2.x, NameNode was a single-point of failure. Failure of the NameNode resulted in the cluster being unavailable. Recently, High Availability features have been added to HDFS but they have limitations: the hot Standby NameNode cannot actively process requests and new hardware is needed for the Stand-by NameNode.
- **Single-protocol**. The stock HDFS implementation provides for a single protocol for data access and does not expose other storage protocols like Object or File interfaces.
- **Storage overhead**. By default, HDFS performs a 3X replication of all data blocks. This results in 2x overhead which is overkill for certain use cases, like archiving.
- **Small-file problem**. HDFS is inefficient at handling a large volume of small files because metadata for each file in the file system needs to be stored in the memory of a single server, the NameNode. For a million files in HDFS, a small number in big data terms, the NameNode consumes about 3GB RAM!
- **Architecture**. HDFS was architected almost a decade ago and was designed for unreliable commodity disks and legacy network infrastructure (1GbE). A distributed system was assumed to be a bottle-neck due to network speeds and not the disk. These assumptions don't hold true in today's infrastructure.
- **Lack of enterprise-grade features**. HDFS lacks enterprise-grade features like Geo-Distribution, Disaster Recovery, Consistent Snapshots, Deduplication, Metering, etc.
- **Lacks multi-tenancy**. HDFS lacks robust multi-tenancy features, which can provide data and performance isolation guarantees to tenant users. This results in multiple siloed clusters in a large enterprise, resulting in low cluster utilization.

ELASTIC CLOUD STORAGE

OVERVIEW

EMC Elastic Cloud Storage (ECS) ECS is a software-defined, scale-out object storage platform that deploys entirely in software. ECS features a unique unstructured storage engine that supports Amazon S3, OpenStack Swift, EMC Atmos and Centera CAS object storage APIs. ECS also enables HDFS access on the same unstructured storage engine. Customers have the choice of deploying ECS in two ways:

- ECS Software on Commodity: Customer can deploy the ECS software on their choice of commodity server and operating systems and writing data to certified 3rd party commodity JBOD (today ECS supports HP SL4540). The customer is primarily responsible for management and operations, though ECS can monitor the health of the nodes.
- ECS Appliance: Customers can deploy ECS as an integrated appliance that includes all software and hardware. EMC provides management and operational support.

ECS Appliance provides a complete hyper-scale storage infrastructure designed to meet the requirements of cloud, mobile, Big Data and social applications. EMC ECS appliance brings the cost-profile, simplicity, and scale of a public cloud to anyone - with the trust, reliability, and support you expect from EMC.

ECS ARCHITECTURE

ECS supports the storage, manipulation, and analysis of unstructured data on a massive scale on commodity infrastructure. ECS features an unstructured storage engine that supports object and HDFS data services:

Object Data Service

The Object data service enables the storage, access, and manipulation of unstructured data as objects on commodity-based systems, such as the HP SL4540, and the ECS Appliance. The Object data service is compatible with Amazon S3, OpenStack Swift, EMC Atmos and Centera CAS object storage APIs.

HDFS Data Service

The HDFS data service provides support for the Hadoop Distributed File System (HDFS). The HDFS data service enables organizations to build a big data repository at scale. With the HDFS data service, organizations can use the ECS storage environment as a big data repository against which they can run Hadoop analytic applications.

The ECS unstructured storage engine is the primary component that ensures data availability and protection against data corruption, hardware failures, and data center disasters. The engine exposes multi-head access across object and HDFS so that an application can write object and read through HDFS or vice versa, and allows access to the same data concurrently through multiple protocols. Today, ECS supports object and HDFS interfaces. In the future it will also support NFS and CIFS.

HADOOP ON ECS

OVERVIEW

The HDFS data service is a Hadoop Compatible File System (HCFS) that enables you to run Hadoop 2.3 applications on top of your ECS. Customers can configure any Hadoop distribution to run against the built-in Hadoop file system (stock HDFS), against HDFS data service, or any combination of HDFS, ECS HDFS data service, or other Hadoop Compatible File Systems available in the environment.

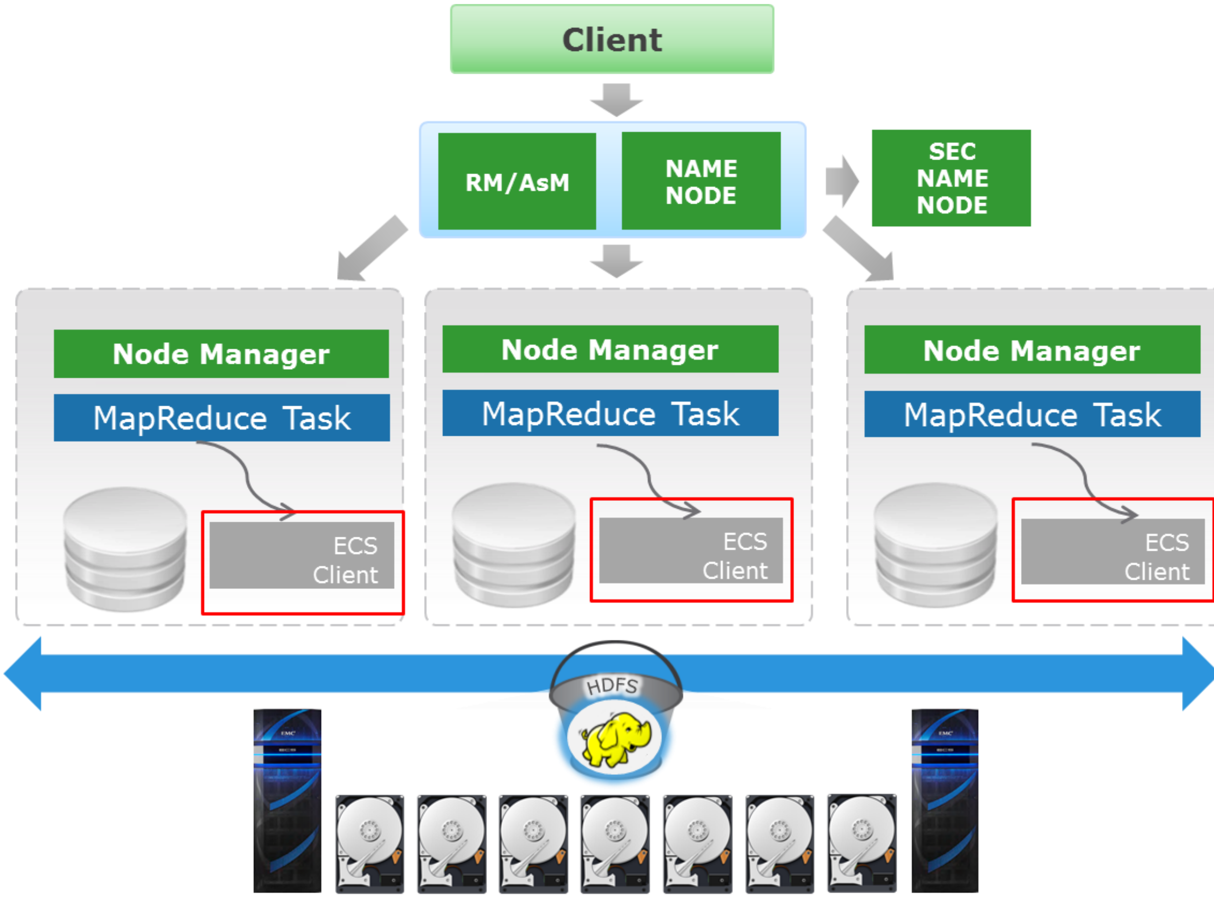

Figure 2. ECS HDFS Data Service Architecture

When customers set up the Hadoop client to use ECS HDFS data service instead of traditional HDFS, its configuration points to the ECS HDFS data service to do all the HDFS activity. On each Hadoop HDFS client node, any traditional Hadoop component would use the ECS HDFS data service client (JAR) to perform the HDFS activity.

ENTERPRISE-GRADE STORAGE FOR HADOOP

The ECS HDFS data service brings critical enterprise-grade features to Hadoop storage. The ECS HDFS data service increases the efficiency, speed and reliability of Hadoop and provides:

- Massive scalability. The ECS Appliance has been proven to effortlessly scale to Petabyte and Exabyte storage demands. The ECS architecture allows compute and storage to scale independently.
- Multi-Protocol Access. ECS offers unparalleled accessibility in a single platform with support for multiple Object APIs as well as HDFS access. Developers are free to focus on their application and not operations.
- Geo-protection. ECS geo-protection provides full protection against a site failure should a disaster or other calamity force an entire site offline. A geo-protection layer in ECS protects data across geo- distributed sites and ensures that applications seamlessly function in the event of a site failure.
- Multi-Site Access. ECS provides strong, consistent views of data regardless of where the data resides. With geo-protection enabled, applications can access data immediately through any ECS site, regardless of where the data was last written.

- Efficiency. Erasure coding provides storage efficiency without compromising data protection or access. The ECS storage engine implements the Reed Solomon 12/4 erasure-coding scheme in which a chunk is broken into 12 data fragments and 4 coding fragments. The resulting 16 fragments are dispersed across nodes at the local site. The storage engine can reconstruct a chunk from a minimum of 12 fragments.
- Flexibility. The ECS HDFS data service allows IT departments to choose multiple Hadoop vendors and connect them all to use the ECS HDFS data service as the storage substrate to allow in-place analytics. Snapshots, Versioning and Journaling are also natively built into the platform.
- Storage Efficiency for Large and Small Files. The ECS storage engine is adept at handling both a high volume of small files as well as very large. Using a technique called *box-carting*, ECS can execute a large number of user transactions concurrently with very little latency. This enables ECS to support workloads with high transaction rates. ECS is also very efficient when handling very large files. All ECS nodes can process write requests for the same object simultaneously and each node can write to a set of three disks.
- Federal Compliance. ECS is IPv6 and dual-stack compliant, FIPS 140-2 compliant, and Section 508 compliant.

CONCLUSION

Hadoop is a low-cost, highly scalable, distributed analytics engine that can significantly reduce the time and resources needed by an enterprise to derive valuable insight from their Big Data assets. Hadoop and the ECS HDFS data service integration allows organizations to utilize a software-defined storage infrastructure as a native part of their Hadoop architecture while also providing simplicity, flexibility and rapid deployment.