Comparing the Hadoop Distributed File System (HDFS) with the Cassandra File System (CFS)

White Paper

BY DATASTAX CORPORATION

August 2013

Table of Contents

Abstract

The Hadoop Distributed File System (HDFS) is one of many different components and projects contained within the community Hadoop™ ecosystem. The Apache Hadoop project defines HDFS as "the primary storage system used by Hadoop applications" that enables "reliable, extremely rapid computations." This paper provides a high-level overview of how Hadoop-styled analytics (MapReduce, Hive, Pig and Mahout) can be run on data contained in Apache Cassandra™ without the need for HDFS.

Introduction

The Hadoop Distributed File System (HDFS) is one of many different components and projects contained within the community Hadoop™ ecosystem. The Apache Hadoop project defines HDFS as: "the primary storage system used by Hadoop applications. HDFS creates multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, extremely rapid computations."[1]

Hadoop utilizes a scale-out architecture that makes use of commodity servers configured as a cluster, where each server possesses inexpensive internal disk drives. As the Apache Project's site states, data in Hadoop is broken down into blocks and spread throughout a cluster. Once that happens, MapReduce tasks can be carried out on the smaller subsets of data that may make up a very large dataset overall, thus accomplishing the type of scalability needed for big data processing.

In general, this divide-and-conquer strategy of processing data is nothing really new, but the combination of HDFS being open source software (which overcomes the need for high-priced specialized storage solutions), and its ability to carry out some degree of automatic redundancy and failover make it popular for modern businesses looking for data warehouse batch analytics solutions. This is just one reason why the Hadoop market is expected to grow at an eye-popping compound annual growth rate (CAGR) of 58 percent until 2018.[2]

However, what these businesses are most interested in is not Hadoop's underlying storage structure, but rather what it facilitates in delivering: a cost-effective means for analyzing and processing vast amounts of data for data warehouse use cases. But what about analytics needed for line-of-business application data?

This paper provides a high-level overview of how DataStax uses Apache Cassandra™ to run analytics on Cassandra data that comes from line-of-business applications.

[1] "Hadoop HDFS," Hadoop.Apache.org: http://hadoop.apache.org/hdfs/.

[2] "Hadoop-MapReduce Market Forecast 2013-2018," MarketAnalysis.com, July 31, 2012: http://www.marketanalysis.com/?p=279.

Overview of HDFS

A typical Hadoop deployment with HDFS resembles the following:

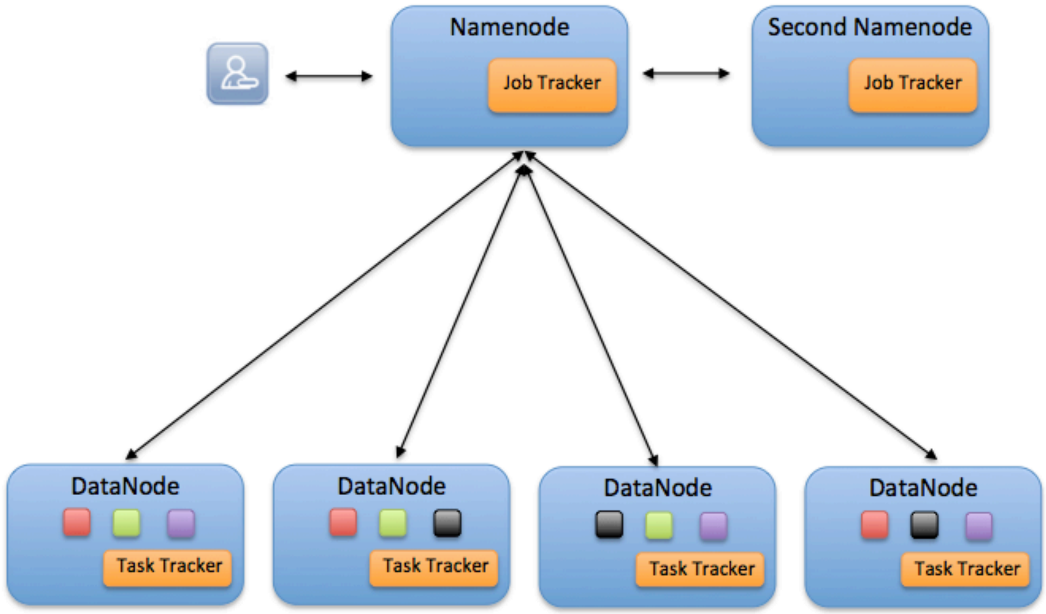


Figure 1: Standard Hadoop Deployment with HDFS

Hadoop and HDFS utilize a master-slave architecture. HDFS is written in Java, with an HDFS cluster consisting of a primary NameNode – a master server that manages the file system namespace and also regulates access to data by clients. An optional secondary NameNode for failover purposes also may be configured.

In addition, there are a number of DataNodes; usually, there is a one-to-one relationship between a DataNode and a physical machine. Each DataNode manages the storage attached to the boxes that it runs on. HDFS uses a file system namespace that enables data to be stored in files. Each file is divided into one or more blocks, which are then divvied up across a set of DataNodes.

The NameNode is responsible for tasks such as opening, renaming, and closing files and data directories. It also tackles the job of mapping blocks to DataNodes, which are then responsible for managing incoming I/O requests from clients. The DataNode handles block replication, creation, and removal of data when instructed to do so by the NameNode.

The Benefits of HDFS

There is little debate that HDFS provides a number of benefits for those who choose to use it. Below are some of the most commonly cited.

Built-In Redundancy and Failover

HDFS supplies out-of-the-box redundancy and failover capabilities that require little to no manual intervention (depending on the use case). Having such features built into the storage layer allows system administrators and developers to concentrate on other responsibilities versus having to create monitoring systems and/or programming routines to compensate for another set of storage software that lacks those capabilities.

Moreover, with downtime being a real threat to many modern businesses' bottom line, features that minimize outages and contribute to keeping a batch analytic data store up, operational, and feeding any online system that requires its input are welcomed by both IT and business professionals.

Big Data Capable

The hallmark of HDFS is its ability to tackle big data use cases and most of the characteristics that comprise them (data velocity, variety, and volume). The rate at which HDFS can supply data to the programming layers of Hadoop equates to faster batch processing times and quicker answers to complex analytic questions.

Portability

Any tenured data professional can relay horror stories of having to transfer, migrate, and convert huge data volumes between disparate storage/software vendors. One benefit of HDFS is its portability between various Hadoop distributions, which helps minimize vendor lock-in.

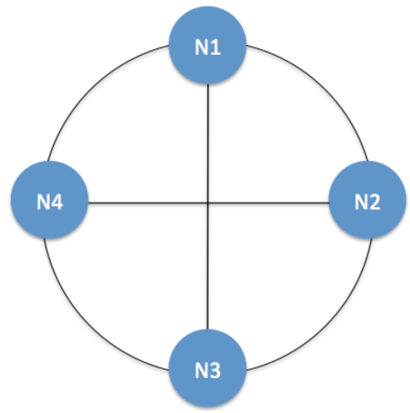
Cost-Effective

As previously stated, HDFS is open source software, which translates into real cost savings for its users. As many companies can attest, high-priced storage solutions can take a significant bite out of IT budgets and are many times completely out of reach for small or startup companies.

Other benefits of HDFS exist, but the four above are the primary reasons why many users deploy HDFS as their analytic storage solution.

What Is Apache Cassandra?

Apache Cassandra is a massively scalable NoSQL database. Used today by numerous modern businesses to manage their critical data infrastructure, Cassandra is known for being the solution technical professionals turn to when they need a real-time NoSQL database that supplies high performance at massive scale, which never goes down.



Rather than using a legacy master-slave or a manual and difficult-to-maintain sharded design, Cassandra has a peer-to-peer distributed "ring" architecture that is much more elegant, easy to setup, and maintain. In Cassandra, all nodes are the same; there is no concept of a master node, with all nodes communicating with each other via a gossip protocol.

Cassandra's built-for-scale architecture means that it is capable of handling petabytes of information and thousands of concurrent users/operations per second across one to many data centers as easily as it can manage much smaller amounts of data and user traffic. It also means that, unlike other master-slave or sharded systems, Cassandra has no single point of failure and therefore is capable of offering true continuous availability.

What Is the Cassandra File System (CFS)?

The Cassandra File System (CFS) was designed by DataStax Corporation to easily run analytics on Cassandra data. Now implemented as part of DataStax Enterprise, which combines Apache Cassandra, and Solr™ together into a unified big data platform, CFS provides the storage foundation that makes running Hadoop-styled analytics on Cassandra data hassle-free.

How Does CFS Work?

In contrast to a master-slave architecture like HDFS, CFS is based on Cassandra, so the implementation is peer-to-peer and "masterless." A user is able to create a cluster that seamlessly stores real-time data in Cassandra, performs analytic operations on that same data, and also handles enterprise search operations.

Cassandra's built-in replication transparently takes care of replicating the data among all real-time, analytic, and search nodes. A user may configure any type of cluster they desire.

Figure 3: A Simple DataStax Enterprise Cluster

CFS stores metadata information regarding analytics data in a Cassandra keyspace, which is analogous to a database in the relational database management system (RDBMS) world. Two Cassandra column families (like tables in an RDBMS) in the keyspace contain the actual data. The data contained in these column families is replicated across the cluster to ensure data protection and fault tolerance.

The column families mirror the two primary HDFS services. The `inode` column family replaces the HDFS NameNode service, which tracks each datafile's metadata and block locations on the participating analytics nodes. Captured information in this column family includes filename, parent path, user, group, permissions, filetype and a list of block ids that make up the file. For block ids, it uses `TimeUUID`, so blocks are ordered sequentially in a natural way. This makes supporting HDFS functions like `append()` easy.

The `sblocks` column family supplants the HDFS DataNode service that stores file blocks. This column family stores the actual contents of any file that is added to an analytics node.

Each row in `sblocks` represents a block of data associated with a row in the `inode` column family. Each row key is a block `TimeUUID` from an `inode` row. The columns are time ordered compressed sub-blocks that, when decompressed and combined, equal one HDFS block.

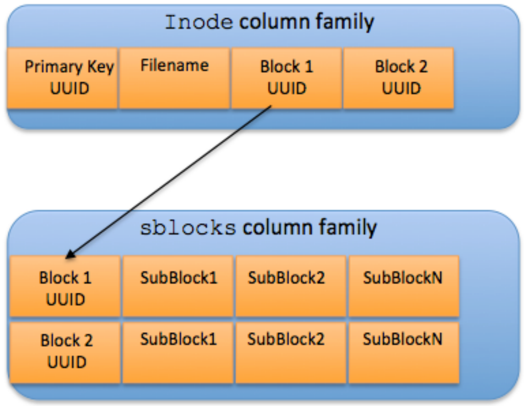


Figure 4: CFS Column Families

When data is added to an analytics node, CFS writes the static metadata attributes to the `inode` column family. It then allocates a new `sblocks` row object, reads a chunk of that data (controlled via the Hadoop parameter `fs.local. block.size`), splits it into sub-blocks (controlled via the parameter `cfs.local.subblock.size`), and compresses them via Google's snappy compression.

Once a specific block is complete, its block id is written to the `inode` column family row and the sub-blocks are written to Cassandra with the block id as the row key and the sub-block ids as the columns.

Reads are handled in a straightforward manner. When a query request comes into an analytics node, CFS reads the `inode` information and finds the block and sub-block(s) needed to satisfy the request.

Hadoop Compatibility and Command Management

CFS implements the Hadoop File System API so it is compatible with the Hadoop stack and third-party tools.

With respect to handling Hadoop commands, CFS provides complementary commands, so little to no learning curve is experienced. For instance, the following Hadoop command:

```
bin/hadoop fs -cat /myfiles/foo.txt
```

would be run inside of CFS/DataStax Enterprise as:

```
bin/dse hadoop fs -cat /myfiles/foo.txt
```

Benefits of CFS

CFS is built on a proven and trusted technology (Apache Cassandra) that powers many applications all over the globe, and possesses a reputation known for scaling and performing extremely well under challenging workloads.

Next, it should be understood that CFS is completely transparent to any developer or end user. There are no changes to any MapReduce, Hive, Pig, Mahout, or other routines that run against CFS vs. HDFS.

There are, however, a number of benefits derived from using CFS over HDFS.

Simpler Deployment

With CFS, there is no need for any master-slave failover configurations, no zookeeper requirements, and no complex storage requirements for storage area networks (SANs). Instead, a cluster can be set up and installed in a few minutes, with all CFS configuration being handled automatically when a inside of DataStax Enterprise marked for analytics is started for the first time.

By using CFS, in essence, three traditional Hadoop services (NameNode, Secondary NameNode, and DataNode) are replaced with one easy-to-understand and use fault tolerant component.

Better Availability

Continuous availability for analytics in a database cluster in CFS is maintained without the need for any shared storage solution (e.g., SANs). Instead, a cluster can consist of vanilla, white-box hardware with local storage and still meet any high-availability SLA requirement. Cassandra's replication and redundancy provides complete customization with respect to how many copies of data should be maintained in a cluster, thus ensuring constant uptime and no chance for data loss.

Multi-Data Center Support

Many modern businesses need to run analytic operations that span more than one data center. CFS's continuous availability benefits include supporting multi-data center, cloud, and hybrid (on-premise and cloud) environments.

CFS supports running a single database cluster across as many data centers as desired, with any node in the cluster being able to service reads and writes. Moreover, an architect can create multiple CFS keyspaces so that each data center has its own local copy of all the data it needs. A Job Tracker for each data center can also be configured so each location has its own for handling MapReduce and other analytic processing jobs.

No Shared Storage Requirement for Failover

The Hadoop documentation is clear on the need for a shared storage solution to support failover:

Shared storage – you will need to have a shared directory which both NameNode machines can have read/write access to. Typically this is a remote filer which supports NFS and is mounted on each of the NameNode machines. Currently only a single shared edits directory is supported. Thus, the availability of the system is limited by the availability of this shared edits directory, and therefore in order to remove all single points of failure there needs to be redundancy for the shared edits directory. Specifically, multiple network paths to the storage, and redundancy in the storage itself (disk, network, and power). Because of this, it is recommended that the shared storage server be a high-quality dedicated NAS appliance rather than a simple Linux server.[3]

No such requirement is needed for CFS failover as everything is automatically and transparently handled by Cassandra.

Full Data Integration

CFS provides the ability to have one big data platform that handles real-time, analytic, and enterprise search workloads in one cluster without any workload affecting the other where data or compute resources are concerned. Instead, full mixed workload support is built and transparently handled by DataStax Enterprise.

This benefit results in the elimination of data "silos" in organizations and the need to create and maintain costly ETL routines to move data between different silos. Any data written to Cassandra is replicated to analytics and search nodes, and vice versa.

Further, even output results from analytics tasks may be replicated. For example, a Hive query on analytic nodes that takes some time to complete and produces a result set is able to have that result set replicated over to Cassandra nodes for real-time query access.

Commodity Hardware Support

CFS runs well on commodity hardware and requires no special server or network equipment be purchased.

[3] HDFS High Availability, Hadoop.Apache.org:
http://hadoop.apache.org/common/docs/current/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailability.html.

Managing and Monitoring Cassandra Deployments

Administering and monitoring the performance of any distributed database system can be challenging, especially when the database spans multiple geographical locations. However, DataStax makes it easy to manage CFS and database clusters with DataStax OpsCenter.

DataStax OpsCenter is a visual management and monitoring solution for Cassandra database clusters. Being web-based, DataStax OpsCenter allows a developer or administrator to easily manage and monitor all aspects of their database from any desktop, laptop, or tablet without installing any client software. This includes databases that span multiple data centers and the cloud.

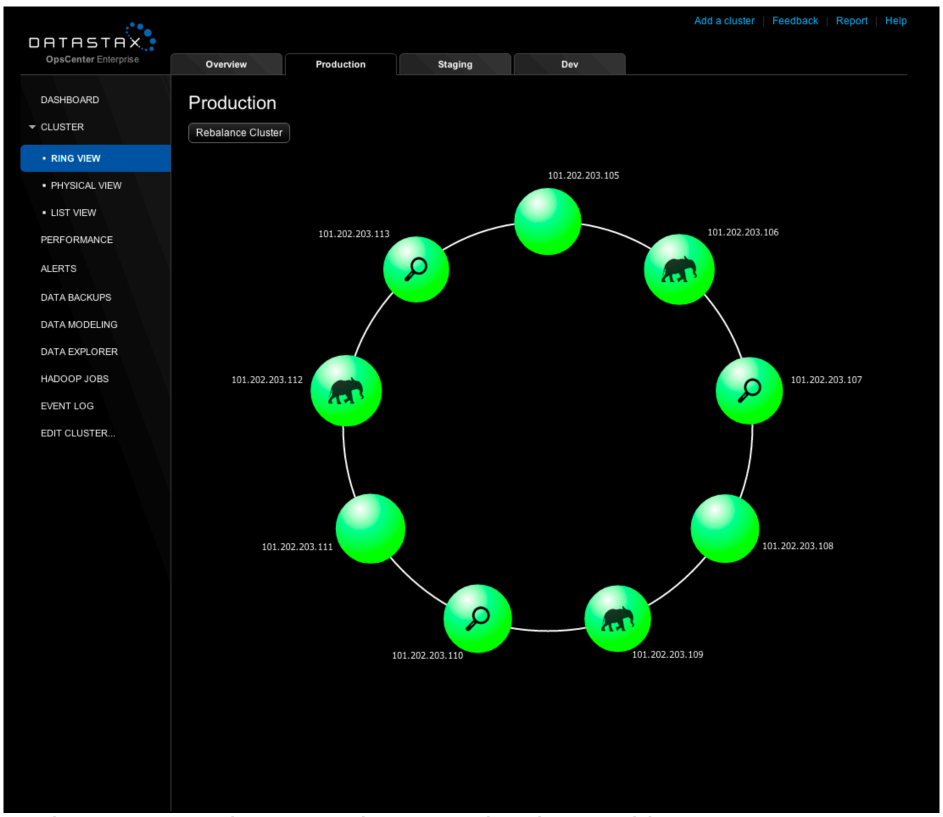


Figure 5: Managing a 9-node Cassandra cluster with DataStax OpsCenter

Other Benefits of DataStax Enterprise

DataStax is the most trusted provider of Cassandra, employing the Apache chair of the Cassandra project as well as most of the committers. For enterprises wanting to use Cassandra in production for line-of-business applications, DataStax supplies DataStax Enterprise Edition, which includes an enterprise-ready version of Cassandra plus the ability to run analytics on Cassandra data with Hadoop and enterprise search operations on Cassandra with Solr. With DataStax Enterprise, modern businesses get a complete enterprise NoSQL database platform that contains:

- A certified version of Cassandra that has passed DataStax's rigorous internal certification process, which includes heavy quality assurance testing, performance benchmarking, and more.
- As previously discussed, integrated analytics capabilities that include MapReduce, Hive, Pig, Mahout, and Sqoop support.
- Bundled enterprise search support with Apache Solr.
- An enterprise version of DataStax OpsCenter, a visual management and monitoring tool.
- Expert, 24x7x365 production support.
- Certified maintenance releases.

Who Is Using Cassandra?

Many modern businesses and organizations are using Cassandra for critical applications today. Here are just some examples:

Figure 6: A sample of companies and organizations using Cassandra in production

Some DataStax customers using DataStax Enterprise with analytics include:

- **eBay** – Uses DataStax Enterprise across multiple data centers, with one data center being devoted to analytics.
- **HealthCare Anytime** – Employs DataStax Enterprise for their online patient portals, with analytics being needed to produce proper billing for Medicare/Medicaid.
- **ReachLocal** – Uses DataStax Enterprise in six different data centers across the world to support their global online advertising business, with analytics being part of their infrastructure.
- **SimpleReach** – Deploys DataStax Enterprise to provide clients with Google analytics ability for their websites, which allows them to know how all their content is being referenced socially.

Conclusion

While HDFS is a good solution for providing cost-effective storage for Hadoop implementations devoted to data warehouse systems, CFS delivers the ability to run analytics on Cassandra data that comes from line-of-business applications.

To find out more about Cassandra and DataStax, and to obtain downloads of Cassandra and DataStax Enterprise software, please visit www.datastax.com or send an email to info@datastax.com. Note that DataStax Enterprise Edition is completely free to use in development environments, while production deployments require a software subscription be purchased.

About DataStax

DataStax powers the big data applications that transform business for more than 300 customers, including startups and 20 of the Fortune 100. DataStax delivers a massively scalable, flexible and continuously available big data platform built on Apache Cassandra™. DataStax integrates enterprise-ready Cassandra, Apache Hadoop™ for analytics and Apache Solr™ for search across multi-data centers and in the cloud.

Companies such as Adobe, Healthcare Anytime, eBay and Netflix rely on DataStax to transform their businesses. Based in San Mateo, Calif., DataStax is backed by industry-leading investors: Lightspeed Venture Partners, Crosslink Capital and Meritech Capital Partners. For more information, visit DataStax or follow us @DataStax.