A NEW ERA OF HADOOP – HADOOP 2.X

P. Praveen Yadav[1], Dr. A. Suresh Babu[2], Smt.S.J.Saritha[3]

Department of CSE, JNTUA Engineering College (Autonomous), Pulivendula, India[1]

Assistant Professor, Department of CSE, JNTUA Engineering College (Autonomous), Anantapur, India[2]

Assistant Professor, Department of CSE, JNTUA Engineering College (Autonomous), Pulivendula, India[3]

ABSTRACT— Hadoop 2.X is an extended version of Hadoop [1](also called Hadoop 1.X).This contains fundamental changes in the architecture, taking the compute platform beyond MapReduce and introducing new application paradigms. Hadoop 2.X storage subsystem also supports other frameworks besides HDFS. This new version improves scalability and performance in both the compute and storage layers, with disk performance up to five times faster. The compute layer scaling to clusters are increased with more than100k concurrent tasks. The failover of main node servers will be reduced and high availability is provided. All other key Hadoop 2.X features are discussed here.

IndexTerms—Cluster Architecture, Hadoop 2.X, HDFS, HDFS Snapshots, YARN, ZKFC

1 INTRODUCTION

Apache Hadoop is a software framework that supports data-intensive distributed applications. It has been used by many big technology companies, such as Amazon, Facebook, Yahoo, and IBM. Hadoop [2] is best known for a computational framework (MapReduce) and its distributed file system (HDFS). The main objective of Hadoop is it focuses on tasks that require all the available data for examination [3]. Apache Hadoop is a scalable framework for storing and processing data on a cluster of commodity hardware nodes. To scale up from a single node to thousands of nodes hadoop is designed. HDFS uses the commodity server nodes and storage drives to store the data. These same set of server nodes are used for computation. This enables scalable and efficient ways of storing and processing data. By adding more servers, capacity of storage, computation along with I/O bandwidth can also be scaled.

The high-level components of a Hadoop cluster [4] are illustrated in fig 1.

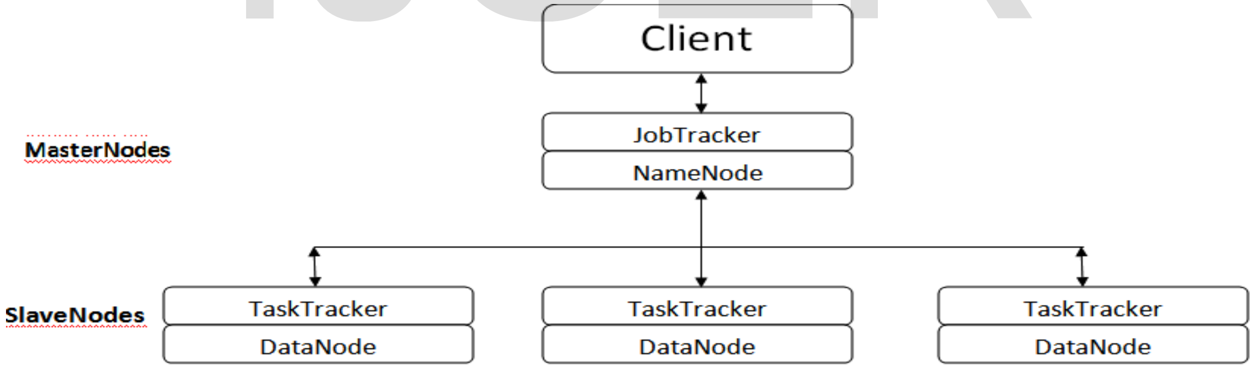

Fig. 1. Simple illustration of the Hadoop cluster architecture

NameNode is a single master server for storing the file system metadata in HDFS. The files stored on HDFS are split into blocks. These blocks are stored on slave nodes called DataNodes. For data reliability, multiple replicas of blocks are stored on Data Nodes. At the NameNode client does an operation such as producing, altering, and erasing files at the file system. The NameNode actively monitors the DataNodes, so if a replica of a block is lost due to any failure, new replicas are created. Computation framework supports parallel, distributed programming paradigms over which a vast amount of data can be processed in a reliable and fault-tolerant manner. Typically, the data is processed in parallel using multiple tasks where each task processes a subset of the data.

But Hadoop 1.X is facing drawbacks like Low Latency, No Updates, Single point of failure (NameNode, JobTracker), Lots of small files not solving, OS dependent (Linux), Job Tracker Resource allocation and Scheduling the jobs. Of all these some are solved in latest version of hadoop named Hadoop 2.X.

Two of the most important advances in Hadoop 2.X are the introduction of HDFS federation and the resource manager YARN (yet another resource negotiator).The HDFS federation adds important measures of scalability and reliability to Hadoop1.X. YARN brings significant performance improvements for some applications along with supporting extra processing models, and by implementing a more flexible execution engine. The architecture of YARN is illustrated in fig 2.

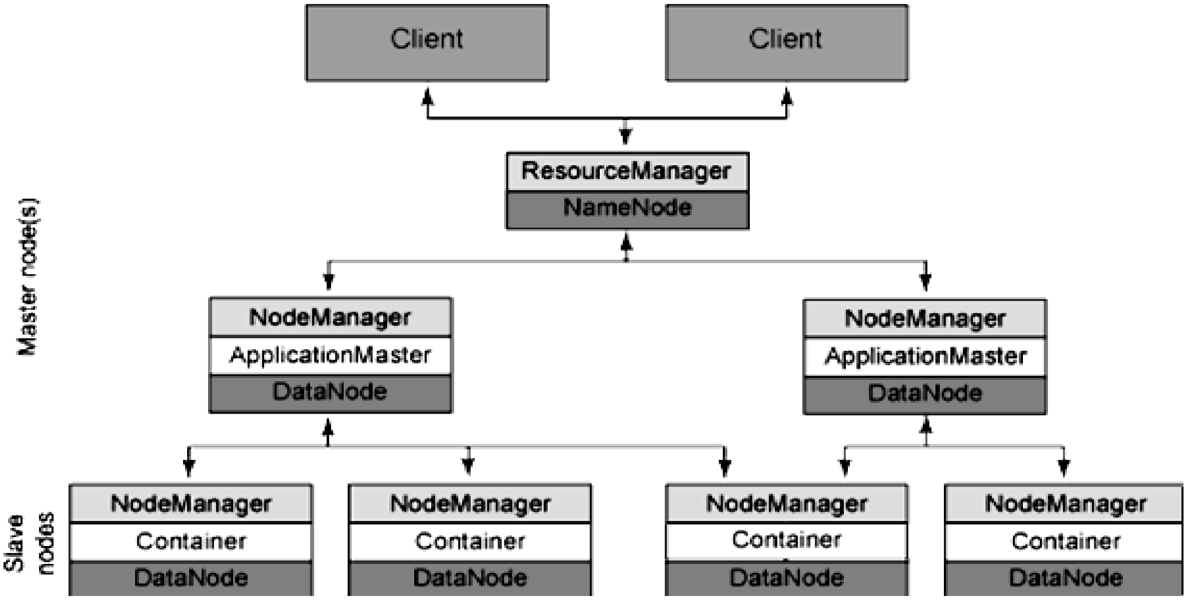

Fig. 2. The new architecture for YARN

2 Improved Features of Hadoop 2.X

2.1 Architectural Evolution:

Fundamental architectural changes are made for both the compute and storage sides of the platform in Hadoop 2.X [5].

2.1.1 Compute Layer Architectural Evolution

Earlier every non-Map Reduce applications were forced to be modeled as MapReduce as compute resources in Hadoop1.X were only available to MapReduce programs. In Hadoop 2.X, YARN(also called MRv2)[6] component generalizes the compute layer to execute not just MapReduce style but other new breed of applications, such as stream processing, to be supported in a first-class manner. The new architecture is more decentralized and allows Hadoop clusters to be scaled significantly to more cores and servers. The comparison is shown in fig 3.

In hadoop 1.X we have only JobTracker to manage both the compute resources and the jobs that use the resources. In hadoop 2.X, YARN is a resource manager that splits function into two. First is a Resource Manager (RM) which focuses on managing the cluster resources and second is an Application Master (AM), which manages each running application (such as a MapReduce job) one-per-running-application. The AM requests resources from the RM, based on the needs and characteristics of the application being run. YARN is designed to allow multiple, diverse user applications to run on a multi-tenant platform. In addition to MapReduce YARN supports multiple processing models. YARN is also called as next-generation execution layer of Hadoop. [7] YARN eliminates bottleneck problem of job execution in MapReduce.

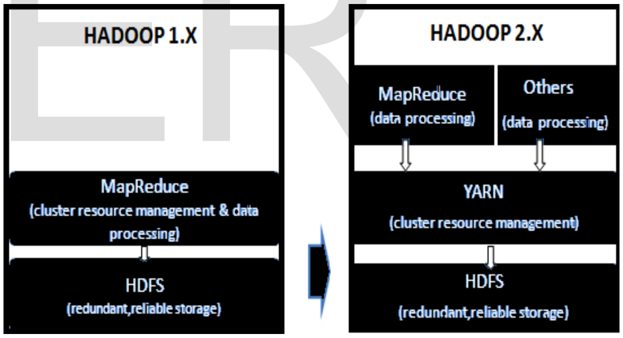
HADOOP 1.X
MapReduce
(cluster resource management & data processing)
HDFS
(redundant,reliable storage)
HADOOP 2.X
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant,reliable storage)

Fig.3.Architecture Comparison of Hadoop 1 and Hadoop 2.X

YARN enhances the power of a Hadoop compute cluster in the following: [8]

◆ **Compatibility With Map Reduce**: Existing MapReduce applications and users can run on top of YARN without interruption to their surviving processes.

◆ **Scalability**: As YARN ResourceManager concentrates entirely on scheduling; it can handle larger clusters more easily.

◆ **Better Cluster Uilization**: The ResourceManager optimizes cluster usage according to measures such as capacity guarantees, equity, and SLAs (service level agreements). Also, there are no named map and reduce slots, which helps to better utilize cluster resources.

◆ **Support For Workloads Other Than MapReduce**: Programming models like graph processing and iterative modeling are possible for data processing, allowing enterprises to realize near real-time processing and increased ROI (return on investment) on their Hadoop investments.

◆ **Agility**: With MapReduce becoming a user-land library, it can evolve independently of the underlying resource manager layer and in a much more agile manner.

Hadoop 2.X we have specialized AM for MapReduce and also a more generalized application framework called Tez allowing generic directed-acyclic-graphs (DAGs) of execution. Tez allows Hive and Pig programs to be executed more naturally as a single job instead of multiple MapReduce phases, resulting in performance improvements. By this we can have a consolidation of clusters and compute resources to run heterogeneous applications, resulting in less resource fragmentation and more efficient utilization.

2.1.2 Storage Layer Architectural Evolution

In Hadoop, the cluster's storage resources are available only to HDFS. But now in Hadoop 2.X the new storage architecture generalizes the block storage layer so that it can be used not only by HDFS but also other storage services which is as similar to that of YARN. Hadoop 2.X also support for heterogeneous storage. Hadoop 1.X treated all storage devices such as spinning disks on a DataNode as a single uniform pool.

Hadoop 2.X will differentiate between storage types along with making the storage type information available to frameworks and applications by which they can take advantage of storage properties.

2.2 NameNode High Availability:

Hadoop 1.X has a single master server called NameNode where all the metadata is stored. When the NameNode is brought down by any software or hardware failure, the cluster would be unavailable until it is restarted. Hadoop 2.X handles this situation by triggering automatic failover by which the standby NameNode becomes active. Here ZKFC (Zookeeper-based Failover Controller) manages failover of NameNodes. On each of the NameNodes this daemon runs and a session is maintained with the Zookeeper. An active local NameNode is elected by one of the ZKFC with the coordination of Zookeeper. Periodically NameNode health check is done by ZKFC. The local ZKFC resigns as the leader when the active NameNode fails health check. Similarly, when failure occur in the active NameNode machine, Zookeeper detects the loss and removes the ZKFC from the failed node as the leader and the ZKFC running on standby becomes the leader by makes the local standby NameNode active. This results in automatic failover.

2.3 HDFS Snapshots

Hadoop 2.X adds support for file system snapshots. A snapshot is a point-in-time image of the entire file system or a subtree of a file system. A snapshot has many uses:

◆ **Protection Against User Errors**: If we take snapshots periodically we can restore files deleted accidentally by the users.

◆**Backup**: We can take a back up of the entire file system or a subtree in the file system. Incremental backups are then taken by copying the difference between two snapshots.

◆ **Disaster Recovery**: Snapshots can be used for copying consistent point-in-time images over to a remote site for disaster recovery.

The snapshots feature supports read-only snapshots which are implemented only in the NameNode and when the snapshot is taken no copy of data is made. Snapshot creation is instantaneous. All the changes made to the snapshotted directory are tracked using modified persistent data structures to ensure efficient storage on the NameNode.

2.4 RPC Improvements And Wire Compatibility

Hadoop 2.X is having several betterments to the RPC layer shared by HDFS, YARN, and MapReduce v2. The on-the-wire protocol instead of using java serialization uses protocol buffers which helps in extending the protocol in the future without breaking the wire protocol compatibility. RPC also adds support for client-side retries of the operation, a key functionality for supporting highly available server implementation. These betterments help in running different versions of daemons within the cluster, paving the way for rolling upgrades.

3 OTHER HDFS IMPROVEMENTS

3.1 I/O Improvements

On ongoing basis improvements to HDFS speed and efficiency are imparted. There is much betterment to HDFS interfaces. A improved short-circuit interface based on UNIX Domain sockets rather than allowing client read inefficiently over a socket from the DataNode permits s to read from the local file system directly. Zero copy reads is supported by this interface. The CRC checksum calculation is optimized using the Intel SSE4.2 CRC32 instruction. These improvements have made I/O 2 to 5 times faster than the previous releases.

3.2 Windows Support

Hadoop 1.X was developed to support only the UNIX family of operating systems. But With Hadoop 2.X, the Windows operating system is indigenously supported because of the fact that Hadoop was written in Java. The compute and storage resource which were dependent on UNIX have been generalized to support Windows. This broadens Hadoop to reach Windows Server market.

3.3 OpenStack Cloud Support

Hadoop 2.X supports the OpenStack Swift file system. It also has topology improvements for virtualized environments. With OpenStack support for spinning Hadoop clusters up and down, Hadoop can now be run on virtualized hardware, both in public and private datacenter clouds.

4 FUTURE ENHANCEMENTS

Hadoop, which was originally designed around batch processing using commodity disks and servers, is changing in the face of a number of trends. The Big Data application space and Hadoop usage pattern, along with the underlying hardware technology and platform, are rapidly evolving. Further, the increasing prevalence of cloud infrastructure,

both public and private, is
influencing Hadoop development. Hadoop is evolving to deal with changes in how clusters are being built. HDFS and YARN architecture are growing to adapt to such changes. Hadoop has become the de facto kernel for the Big Data platform. These exciting developments are being driven by a dedicated Apache community, all in the open.

4 5 CONCLUSION

We compared the features of Hadoop 1.X and Hadoop 2.X.We came to know that Hadoop 2.X introduced YARN, permitting to develop new distributed applications outside MapReduce letting them to exist at the same time with each other in the same cluster. Hadoop 2.X overcomes most of limitations of Hadoop 1.X but not all. Although the performance of computation and storage are better, still we need new versions to come over to make it more flexible and efficient.

REFERENCES

[1] HADOOP 2 what's New by Sanjay Radia and Suresh Srinivas
[2] Tom White, Hadoop: The Definitive Guide. First Edition. PP 12 June 2009
[3] http://www.vmware.com/appliances/director/uploaded_files/What%20is%20Hadoop.pdf
[4] http://www.ibm.com/developerworks/library/bd-hadoopyarn/
[5] http://www.tomsitpro.com/articles/hadoop-2-vs-1,2-718.html
[6] http://www.bigdatapartnership.com/introducing-yarn hadoop-no-more-a-baby-elephant/
[7] http://www.infoworld.com/d/big-data/yarn-unwinds-mapreduces-grip-hadoop
[8] http://www.hadoopuniversity.in/hadoop-yarn training/