Hadoop Technology

Shifna Peter
Department Of Computer Science, Carmel College Mala,
Thrissur, India

Abstract: **Hadoop is an open-source, java-based implementation of Google's MapReduce framework. Hadoop is designed for any application which can take advantage of massively parallel distributed-processing, particularly with clusters composed of unreliable hardware. For example, suppose you have ten terabytes of data, and you want to process it somehow, (suppose you need to sort it). Using a single computer, this could take a very long time. Traditionally, a high end super computer with exotic hardware would be required to do this in a reasonable amount of time. Hadoop provides a framework to process data of this size using a computing cluster made from normal, commodity hardware. There are two major components to Hadoop: the file system, which is a distributed file system that splits up large files onto multiple computers, and the MapReduce framework, which is an application framework used to process large data stored on the file system.**

Keywords: Definition, Components of Hadoop, modules of Hadoop framework, users of Hadoop, history, working of Hadoop, Hadoop clusters, Hadoop distributed file system, other applications, conclusion, bibliography.

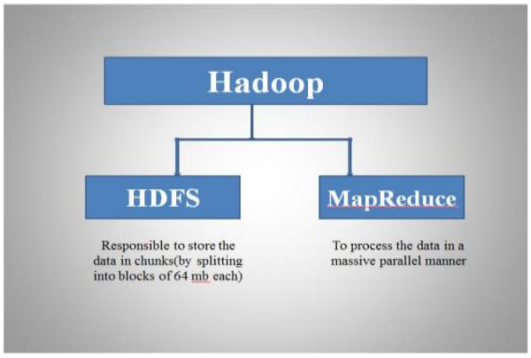
I. INTRODUCTION

The amount of data in the world has exploded during the last decade. This has lead to setting where traditional methods and tools are not anymore suitable for capturing, curating, managing and processing data whose volume exceeds terabytes. Therefore new approaches have been developed to cope with big data. One of the most prominent frameworks for processing big data is apache Hadoop, which is developed for distributed processing of large data sets across clusters of computers. The framework is written in java, but a language binding exists for most of the commonly used languages. This seminar is about apache Hadoop and related projects. The apache Hadoop software library is 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 single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself 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.

II. DEFINITION

Apache Hadoop is an open-source software framework written in java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. all the modules in Hadoop are designed with fundamental assumption the hardware failures(of individual machines or racks of machines) are common place and does should be automatically handled in software by the frame work.

III. COMPONENTS OF HADOOP

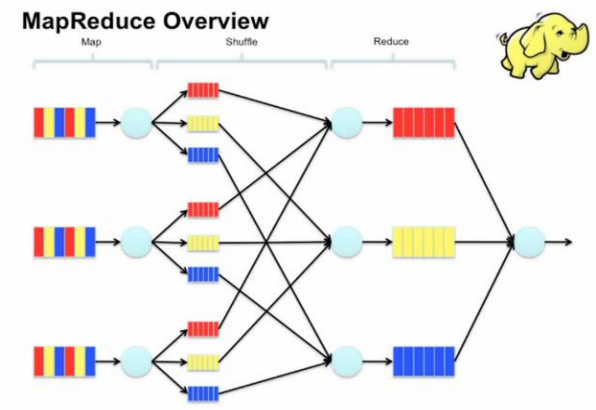


The core of apache Hadoop consists of a storage part (Hadoop distributed file system (hdfs)) and processing part (mapreduce). Hadoop splits files in to large blocks and distributes them amongst the notes in the cluster. To process the data, Hadoop mapreduce transfers packaged code for nodes to processing parallel, based on the data each node needs to process. This approach takes advantaged of data locality nodes manipulating the data that they have one hand-to allow the data to be processed faster and more efficiently than it would be in amore conventional architecture that relies on a parallel file system where commutation and data are connected via high speed networking.

IV. MODULES OF HADOOP FRAMEWORK

The base apache Hadoop frame work is composed of the following modules

A. Hadoop common-contains libraries and utilities needed by other Hadoop modules
B. Hadoop distributed file system(HDFS)-a distributed file- system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
C. Hadoop yarn- a resource management flatform responsible for managing computing resources in clusters and using them for scheduling of users applications; and
D. Hadoop mapreduce- a programming models for large scale data processing.



The term "Hadoop" has come to refer not just to the base modules above, but also to the "ecosystem" , or collection of additional software packages that can be installed on top of or alongside Hadoop, such as apache pig , apache hive , apache HBase , apache spark , and others.

V. USERS OF HADOOP

Apache Hadoop mapreduce and HDFS components were inspired by Google papers on their mapreduce and Google file system. The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell script. For end-users, though mapreduce java code is common, any programming language can be used with "Hadoop streaming" to implement the "map" and "reduce" parts of the user's program. Other related projects expose other higher- level user interfaces. Prominent corporate users of Hadoop include facebook and yahoo. It can be deployed in traditional on-site data centers but has also been implemented in public cloud spaces such as Microsoft Azure, Amazon web services, Google compute engine, and IBM blue mix. Apache Hadoop's a registered trademark of the apache software foundation.

VI.HISTORY

Hadoop was created by Doug Cutting and Mike Cafarella in 2005. Cutting, who was working at yahoo at the time, named it after his son's toy elephant .it was originally develop to support distribution for the Nutch search engine project.

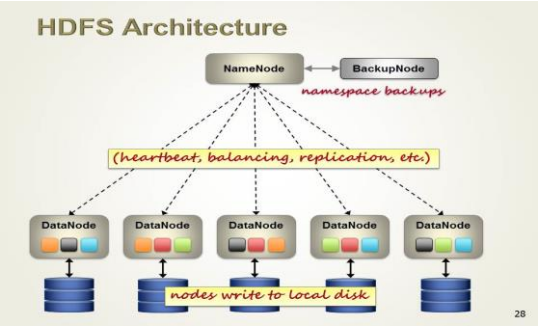
VII. WORKING OF HADOOP

Hadoop consist of the Hadoop common package, which provides file system and OS level abstractions, a mapreduce engine (mapreduce /MRI or YARN / MR2) and the Hadoop distributed file system (HDFS). The Hadoop common package contains the necessary java archive (JAR) files and scripts needed to start Hadoop. The package also provides source code, documentation, and a contribution section that includes projects from the Hadoop community. For effective scheduling of work, every Hadoop- compatible file system should provide location awareness: the name of the rack (more precisely, of the network switch) where a worker node is. Hadoop applications can use this information to run work on the node where the data is, and, failing that, on the same rack /switch, reducing backbone traffic. HDFS uses this method when replicating data to try to keep different copies of the data on different racks the goal is to reduce the impact of a rack power outage or switch failer, so that even if these events occur, the data may still be readable.

VIII. HADOOP CLUSTER

The small Hadoop cluster includes a single master and multiple worker nodes the master node consist of a job tracker, TaskTracker, NameNode, and data node. a slave or worker node act as both a data node and task tracker, tough it is possible data only worker nodes and compute only worker nodes. These are normally used only in nonstandard applications. Hadoop requires java runtime environment (JRE) or 1.6 or higher. The standard startup and shutdown scripts require that secure shell (Ssh) be set up between nodes in the cluster. In a large cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary name node that can generate snapshots of the name node's memory structures thus preventing file-system corruption and reducing loss of data. Similarly, a standalone job tracker server can manage job scheduling. In clusters where the Hadoop mapreduce engine is deployed against an alternative file system, the name node, secondary name node, and DataNode architecture of HDFS are replaced by the file system specific equivalents.

IX. HADOOP DISTRIBUTED FILE SYSTEM (HDFS)



The Hadoop distributed file system is distributed, scalable, and portable file-system written in java for the Hadoop framework. A Hadoop cluster has nominally a single NameNode plus a cluster of DataNode, although redundancy its criticality. Each DataNode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses TCP/ IP sockets for communication. Clients use remote procedure call (RPC) to communicate between each other. HDFS stores large files (typically in the range of gigabytes to terabytes across multiple machines. it achieves reliability by replicating the data across multiple hosts, and hence theoretically does not require RAIT storage on hosts (but to increase I/O Performance some RAID configurations are still useful).

With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack. Data nodes can talk to each other to rebalance data, to move copies around, and to keep the replication of data high. HDFS is not fully POSIX- compliant file system is increased performance for data throughput and support for non- POSIX operations such as append. HDFS added the high-availability capabilities, as announced for release 2.0 in may 2012, letting the main metadata server (the name node) fail over manually to a backup. The project has also started developing automatic fail-over. The HDFS file system includes a so-called secondary NameNode; a misleading name that some might incorrectly interprets a backup NameNode for when the primary NameNode goes offline. In fact, the secondary NameNode regularly connects with the primary name node and builds snapshots of the primary name node's directory information, which he system then saves to local or remote directories. These check pointed images can be used to restart a failed primary NameNode without having to repay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure. Because the NameNode is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files. HDFS federation, a new addition, aims to tackle this, problem to a certain extent by allowing multiple name spaces served by separate NameNode. An advantage of using HDFS is data awareness between the job tracker and task tracker. The job tracker schedules map or reduce jobs to task trackers with an awareness of data location. For example: if node A contains data (x,y,z) and node B contains data (a,b,c), the job tracker schedules node B to perform map or reduce tasks on (a,b,c) and node A would be scheduled to perform map or reduce tasks on (x,y,z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer. When Hadoop is used with other file systems, this advantage is not always available. This can have a significant impact on job-completion times, which has been demonstrated when running data-intensive jobs HDFS was designed for mostly immutable files and may not be suitable for systems requiring concurrent write-operations. HDFS can be mounted directly with a file system in users pace(FUSE) virtual file system on Linux and some other UNIX systems. File access can be achieved through the native java API, the thrift API to generate a client in the language of the users' choosing (C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk, and OCaml), the command- line interface, browsed through the HDFS-UI webapp over HTTP, or via 3rd- party network client libraries.

X. OTHER APPLICATIONS

The HDFS file system is not restricted to MapReduce jobs. It can be used for other applications, many of which are under development at Apache. The list includes the HBase database, the Apache Mahout Machine learning system, and the Apache Hive data warehouse system. Hadoop can in theory be used for any sort of work that is batch-oriented rather than real-time, is very data-intensive, and benefits from parallel processing of data. It can also be used to complement a real-time system, such as lambda architecture. As of October 2009, commercial applications of Hadoop included:

A. Log and /or click stream analysis of various kinds
B. Marketing analytics
C. Machine learning and /or sophisticated data mining
D. Image processing
E. Processing of XML messages
F. Web crawling and/or text processing
G. General archiving, including of relational/ tabular data, e.g. for compliance

XI. CONCLUSION

There are many distributed computing frameworks, but what is particularly notable about Hadoop (and Google's MapReduce) is the built – in fault tolerance .it is designed to run on product hardware, and therefore it expects computers to be braking frequently. The underlying file system is highly- redundant (blocks of data are replicated across multiple computers) and the MapReduce processing framework automatically handles computer failures which occur during a processing job by reassigning the processing to another computer in the cluster.

REFERENCE

[1]. Lam, Chuck (July 28, 2010). Hadoop in Action (1st Ed.). Manning public actions. P.325. ISBN 1-935-18219-6.
[2]. Venner, Jason (June 22, 2009). Pro Hadoop (1st Ed.). Apress .p.440.ISBN 1-430-21942-4.
[3]. White, Tom (June 16, 2009). Hadoop: The Definitive Guide (1stEd.). O'Reilly media p.524.ISBN 0-596-52197-9.