marlabs
driving digital agility

WHITEPAPER

Big Data and Hadoop

Abstract

This paper explains the significance of Hadoop, an emerging yet rapidly growing technology. The prime goal of this paper is to unveil the potential of Hadoop framework to business analysts who frequently face problems in managing big data.

Introduction

The world is moving towards cloud computing, a new technological era, which has just begun. Have you ever pondered why the word 'Cloud' was introduced as a terminology in Information Technology?

In cloud computing, the word 'Cloud' is used as a metaphor for 'Internet'. Having said that, cloud computing is a kind of Internet based computing where a wide variety of services like storage, servers, and applications are offered to the enterprises and individuals through the Internet. Typically, cloud computing encompasses multiple computing resources rather than local servers or dedicated devices to handle complex applications. This mechanism is extremely as it harnesses unused or idle computers in the network to solve problems that are too intensive for any standalone computer.

Several designs, prototypes, and methodologies have been developed over the past few years to tackle parallel computing problems. Moreover, specially designed servers were tailored to meet parallel computing requirements. The major problem was that these servers were too expensive and yet less productive. With the advent of multi-core processors and virtualization technology, the problems have been diminishing. Hence effective and powerful tools are being built to achieve parallelization using commodity machines—Hadoop is one such tool.

Cloud computing is necessary when the analysis of data cannot be performed by a single computer. Typically, when a thorough analysis is done on massive data, multiple computers are required to balance the data load and analysis. This massive data is sometimes referred to as big data.

What is "Big Data"?

One of the key components in business analytics is data. Data is ubiquitous in every field, and helps forecast, vet, transact, and consolidate any given analytical problems. It also plays a major role as a fail-safe option by maintaining the history of each event carried out during the course of development. In today's competitive business world, the demand for data has been increasing exponentially. Currently, the magnitude and type of data available to enterprises and the need for analyzing the data in real time for maximizing business benefits is growing rapidly. With the advent of social media and networking services like Facebook and Twitter; search engines like Google, MSN, and Yahoo; and e-commerce services and online banking services, data is multiplying in lightning speed. These data may be unstructured and semi structured. We call this big data.

> *Traditional data management and analytical tools and technologies are struggling to process the large volume of produced by the rise of social and mobile. But there are also new approaches emerging to deal with this problem which will help the enterprises gain maximum value from Big Data.*

Big data is measured in terabytes, petabytes, exabytes and more. Processing, managing, and analyzing data with this magnitude has been a highly strenuous and daunting task for business analysts.

Traditional data management and analytical tools and technologies are struggling to process the large volume of produced by the rise of social and mobile. But there are also new approaches emerging to deal with this problem, which will help the enterprises gain maximum value from big data.

Introduction to Hadoop

What is HADOOP?

Hadoop is an open source software framework licensed under Apache Software Foundation, built for supporting data intensive applications running on large clusters and grids, to offer scalable, reliable, and distributed computing. Apache Hadoop framework is predominantly designed for the distributed processing of large sets of data residing in clusters of computers using simple programming paradigms. It can be operated from single server or tens of thousands of computers, where each computer is responsible for local computation and storage. Apart from this, Hadoop framework identifies and tackles node failures at the application layer there by offering high availability of the service.

Since using Hadoop framework involves more than a single machine, it is imperative to understand the meaning of clusters and grids, although both work in a similar fashion with a subtle difference in their setup.

What is a cluster?

Typically, a cluster is a group of computers (nodes) with identical hardware configurations connected to each other through a fast local area network, where each node performs a desired task. The results from all the nodes are aggregated to solve problems that usually require high availability of the system with low latency.

What is a Grid?

A grid is similar to cluster but with a subtle difference. Multiple nodes in a grid are distributed in different geographical locations and are connected to each other through the Internet. Apart from this, each node in the grid can have different operating systems and hardware configurations.

What makes Hadoop a paramount tool

In a distributed environment, data should be meticulously arranged across several computers to avoid inconsistency and redundancy in the results of any given problem. Moreover, data should be processed carefully to achieve low latency. Hence, several factors influence the speed of the operation in a distributed computing system—the way data is stored, the storage algorithm to manage the distributed data, the parallel computing algorithm to process distributed data, and the fault tolerance check on each node.

In a nut shell, Apache Hadoop uses a programming model called MapReduce for processing and generating large data sets using parallel computing. The MapReduce programming model was initially introduced by Google to support distributed computing on large data sets in clusters of computers. Inspired by Google's work, Apache came up with Hadoop, an open source framework for distributed computing. Written in Java, Hadoop is platform independent and easy to install and use in any commodity machine with Java. This eliminates the use of heavier hardware to process big data.

The Apache Hadoop framework consists of three major modules:

1. Hadoop Kernel

2. MapReduce

3.Hadoop Distributed File System

Hadoop Kernel

Hadoop Kernel, also known as Hadoop Common, provides an efficient way to access the file systems supported by Hadoop. This common package includes necessary Java Archive (JAR) files and scripts required to start Hadoop.

MapReduce

MapReduce is a programming model primarily implemented for processing large data sets. This model was originally developed by Sanjay Ghemawat and Jeffrey Dean at Google. In a nut shell, MapReduce programming model takes a big task and divides it into discrete tasks that can be done in parallel. At its crux, MapReduce is a composite of two functions, map and reduce.

The map function processes a key/value pair to generate a set of intermediate key/value pairs and the reduce function merges all intermediate values corresponding to the same intermediate key. Finally, one group is created for each unique key.

The reduce function is applied in parallel to each group to produce a collection of values in the same domain. Each reduce call can produce either one value or an empty result.

Consider the following pseudo code for counting the number of occurrences of each word in a large collection of documents:

```
map(String key, String value):
// key: Document name
// value: Document content
For each word w in value:
emitIntermediate(w, “1”);
reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int result = 0;
for each v in values:
result = result + v;
emit(result);
```

i.e. map (k1, v1) -> list (k2, v2)

reduce (k2,list(v2)) -> list(v3)

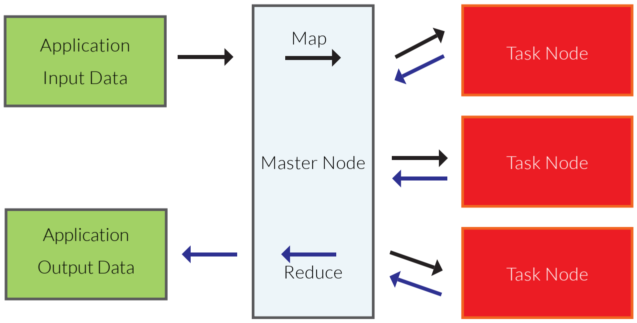
Typically, the map and reduce functions execute in parallel for maximum throughput. Each map and reduce function is executed as a separate thread. The final aggregated result from the reduce function is stored in the local disk.

High Level Working of MapReduce

Hadoop Distributed File System (HDFS)

HDFS is a subproject of the Apache Hadoop project. Hadoop uses HDFS to achieve high data throughput access. HDFS is built using Java and runs on top of the local file system. This was designed to process, read, and write large data files with size ranging from terabytes to petabytes. An ideal file size is a multiple of 64 MB. HDFS stores large files across multiple commodity machines. Using HDFS you can easily access and store large data files split across multiple computers, as if you were accessing or storing local files. High reliability is gained by replicating the data across multiple nodes and hence does not require expensive hardware infrastructure like RAID storage on the nodes. The default replication value is 3 and hence data is replicated on three nodes.

One of the advantages of using HDFS is data awareness between JobTracker and TaskTracker. The JobTracker schedules the map and reduce jobs to TaskTrackers with an awareness of data location. For example, assume that node A contains data (a, b, c, d) and node B contains data (x, y, z). The JobTracker will schedule node A to perform map/reduce tasks on (a, b, c, d) and node B will be scheduled to perform map/reduce tasks on (x, y, z). This will greatly reduce the amount of traffic over the network and prevent unnecessary data transfer. Bringing the data to the place where map function resides is more expensive and time-con-



suming than letting the map function execute at the place where the data resides. This advantage is not available in any other file systems.

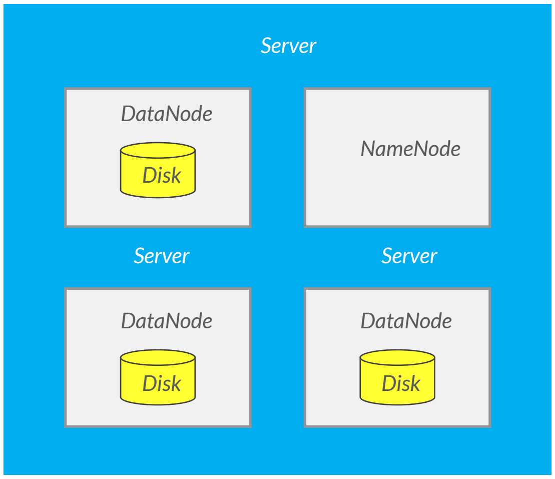
Hadoop uses several types of nodes to form a proper reliable cluster. The NameNode is the major part of the HDFS file system. Its main goal is to maintain the directory tree of all the files in the file system and track the location of the file data across the cluster. It does not store the data of these files by itself. Applications interact with NameNode to create, copy, move, and delete files in the HDFS file system.

Apart from this, a DataNode stores data in the HDFS file system. On Hadoop system startup, a DataNode connects to the NameNode and waits until the service is up and running. The DataNode will respond to the request from the NameNode for file system operations. Applications can directly talk to a DataNode once the NameNode has provided sufficient information about the location of the data.

In this process, the map/reduce tasks are performed by the TaskTracker node near a DataNode. One of the important performance tunings is to have the TaskTracker instance deployed on the same server where the DataNode instance exists. This will allow MapReduce operations to be performed close to the data. Typically, the HDFS file system uses TCP/IP layer for communication.

HDFS Architecture

HDFS Cluster



Pros & Cons

Pros

Distributed data and computation: The data is split into multiple chunks and are distributed across several computers. Computations are done on each data chunk by the system where it resides. This prevents network overhead and unwanted traffic.

Easy to use and cost effective: The distributed environment is easy to setup and can run on any cheap commodity hardware, be it a single core, dual core or multi core. Hadoop can even run on modern tablet PCs or smart phones supporting Linux and Java.

Reliability: By incorporating HDFS, multiple copies of the data are maintained automatically in several nodes in the cluster. This ensures the reliability of data in case of any node failure.

Fault Tolerance: Hadoop framework has a powerful fault tolerant engine, which helps detect faults in a node in the event of failure and automatically applies the fix and recovers quickly.

Processing of huge data: HDFS is designed to manage huge files, unlike the traditional relational database management system (RDBMS), which has the technical limitations of processing data with capacity more 10 terabytes. HDFS also offers low latency while reading or writing the files into the HDFS file system.

Cons

Hadoop is not a substitute for a database: Databases are easy to use when it comes to accessing, updating, and deleting information, just by specifying SQL statements, which is a worldwide standard. Databases also offer indexing of the data so that the frequently used data can be retrieved quickly with an amazing response time. To retrieve information from HDFS, the Map Reduce program needs to be written and run through all the data. In contrary, Hadoop does not solve database problems.

MapReduce is not always the best solution: MapReduce is a programming model for parallel computing where each concurrently running task is independent of each other. Moreover, it cannot be used in small scale data mining. It is mainly designed for managing large scale data mining, when traditional databases reach a physical limit in managing mammoth data. Hadoop adds additional overheads while processing data, which is negligible for processing data with unimaginable size. However, for data with manageable size, it will lead to poor response time. These overheads are inevitable due to its internal design.

Designed for Linux: Works efficiently on Linux platforms. For Windows platforms, separate tools like virtual machines or Linux emulators such as Cygwin are required to simulate the Linux environment. Again, several overheads are added by the virtual machine tools or emulators, which will prevent a hundred percent performance boost.

Cumbersome data migration: For maximum data throughput, the problem data should be stored in HDFS file system. Hence, files that are intended to be processed by Hadoop need to be migrated from local file system to HDFS file system. This should be done manually. Moreover, the migration should be done periodically as and then new data is added to the system making it cumbersome to handle.

The Apache Hadoop Framework is flourishing in the world of cloud computing and has been encouraged by several enterprises looking at simplicity, scalability, and reliability for confronting Big data problems. It shows that that even a commodity desktop PC can be used efficiently for the computation of complex and massive data by forming a cluster of PCs, which indeed minimizes the CPU idle time and judiciously delegates the tasks to the processors, making it cost effective.

No matter how big your data is and how fast it grows, users always crave for higher data retrieval speed forgetting the complexity of arranging and processing data. The bottom line is that irrespective of the data size, the speed and accuracy of analysis should not be compromised. Hence, cloud computing is a better solution for nonfunctional requirements like scalability and performance.

References

http://en.wikipedia.org/wiki/Apache_Hadoop

http://hadoop.apache.org/

http://wiki.apache.org/hadoop/

http://www.javaworld.com/javaworld/jw-09-2008/jw-09-hadoop.html

http://ayende.com/blog/4435/map-reduce-a-visual-explanation