ISSN 2319 – 7757 EISSN 2319 – 7765

Indian Journal of Engineering

A Promising future- Hadoop

Abhas Vashist[1], Chankit Dureja[2], Draun Pareek[3], Jay Madan[4]

1. Dronacharya College of engineering, Gurgaon, India, E-mail: Vashist.1991@gmail.com
2. Dronacharya College of engineering, Gurgaon, India, E-mail: chankitdureja@yahoo.in
3. Dronacharya College of engineering, Gurgaon, India, E-mail: draunpareek@gmail.com
4. Dronacharya College of engineering, Gurgaon, India, E-mail: jaimadan82@gmail.com

Received 25 September; accepted 18 October; published online 01 November; printed 16 November 2012

ABSTRACT

The basic scope of this paper is to help people with better understanding of Hadoop and its various uses that can prove to be really helpful in today's world. The reason for writing this paper is that many students and basically to be the Computer Science Engineers do not have any knowledge of what Hadoop is and how is it changing the world of Computer Science and Information Technology.

Key Words: *Hadoop, HDFS,Map Reduce, Hadoops distributed file system, Big Data, Apache hadoop*

1. INTRODUCTION

Hadoop is a framework for running applications on large clusters built of commodity hardware. The hadoop framework transparently provides applications both reliability and data motion. Hadoopimpliments a computational paradigm named Map/Reduce, where the application is divided into small fragements of work, each of which may be executed or reexecuted on any node in the cluster. In addition it provides a distributed file system (HDFS) that stores data on compute nodes, providing very high aggregate bandwidth acroos the cluster. Both maps reduce and distributed file system are designed so that node faliures are automatically handled by the framework.

2. HADOOPS DISTRIBUTED FILE SYSTEM

Hadoops distributed file system is designed to reliably store very large files across machines in a large cluster. It is inspired by the google file system. Hadoops DFS stores each file as a sequence of blocks, each file except the last block is the same size. Blocks belonging to a file are replicated for fault tolerance. The block size and replication factor are configurable per file. Files in HDFS are "write once" and have strictly one writer at any time.

2.1. HDFS architecture

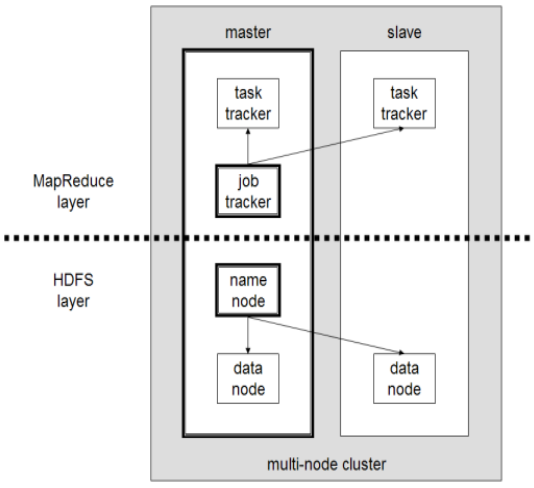

Figure 1
HDFS - master/slave architecture

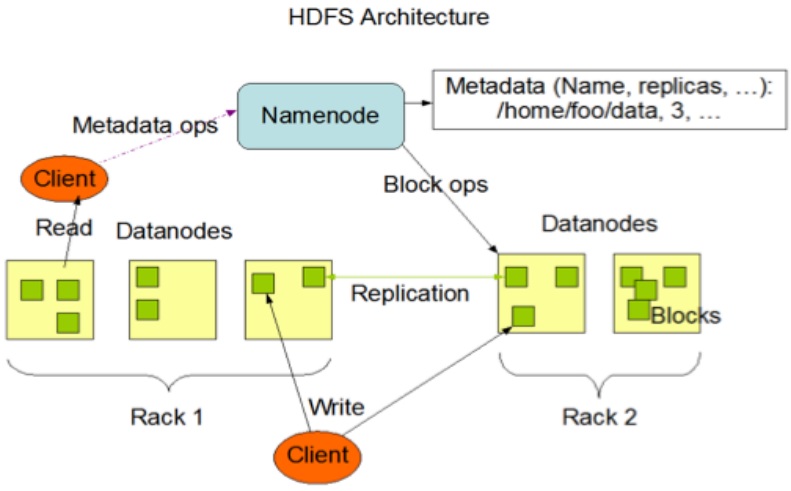

Figure 2
Hadoops distributed file system architecture

HDFS follows a master/slave architecture, which can be defined as shown in Figure 1. An HDFS installation consists of a single Namenode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of Datanodes, one per node in the cluster, which manage storage attached to the nodes that they run on.The Namenode makes filesystem namespace operations like opening, closing, renaming etc. of files and directories available via an RPC interface. It also determines the mapping of blocks to Datanodes. The Datanodes are

Abhas Vashist et al.
A Promising future- Hadoop,
Indian Journal of Engineering, 2012, 1(1), 98-99,

responsible for serving read and write requests from filesystem clients; they also perform block creation, deletion, and replication upon instruction from the Namenode. Figure 2 defines the Hadoops distributed file system architecture.

2.2. Goals of Hadoops distributed file system

- Store large data sets
- Cope with hardware failure
- Emphasise streaming data access

3. MAP REDUCE

The Hadoop Map/Reduce framework harnesses a cluster of machines and executes user defined Map/Reduce jobs across the nodes in the cluster. A Map/Reduce computation has two phases, a map phase and a reduce phase. The input to the computation is a data set of key/value pairs. Tasks in each phase are executed in a fault tolerance manner; if node(s) fail in the middle of the computation the tasks assigned to them are redistributed among the remaining nodes. Having many map and reduce tasks enables good load balancing and allows failed task to be re-run with small runtime overhead.

3.1. Map Reduce architecture

The Hadoop Map/Reduce framework harnesses a cluster of machines and executes user defined Map/Reduce jobs across the nodes in the cluster. A Map/Reduce computation has two phases, a map phase and a reduce phase. The input to the computation is a data set of key/value pairs.Tasks in each phase are executed in a fault-tolerant manner; if node(s) fail in the middle of a computation the tasks assigned to them are re-distributed among the remaining nodes. Having many map and reduce tasks enables good load balancing and allows failed tasks to be re-run with small runtime overhead. The Hadoop Map/Reduce framework has master/slave architecture. It has a single master server or jobtracker and several slave servers or tasktrackers, one per node in the cluster. The jobtracker is the point of interaction between users and the framework. Users submit map/reduce jobs to the jobtracker, which puts them in a queue of pending jobs and executes them on a first-come/first-served basis. The jobtracker manages the assignment of map and reduce tasks to the tasktrackers. The tasktrackers execute tasks upon instruction from the jobtracker and also handle data motion between the map and reduce phases. This is hence explaines with the help of Figure 3.

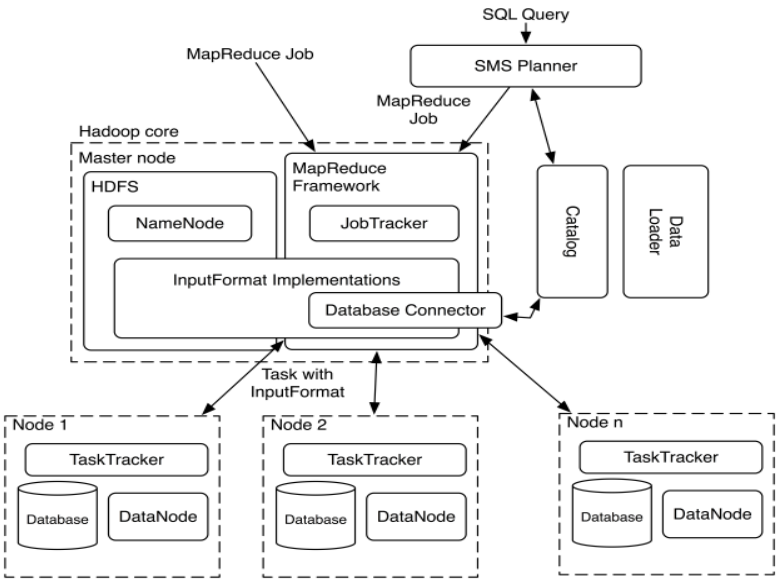

Figure 3

Map Reduce architecture

3.2. Goals of Hadoops Map/Reduce

- Process large data sets
- Cope with hardware failure
- High throughput

4. DOWNLOADING AND INSTALLING HADOOP

Hadoop can be downloaded from one of the Apache download mirrors (http://www.apache.org/dyn/closer.cgi/hadoop/). Select a directory to install Hadoop under (lets say /foo/bar/hadoop-install) and under the tarball in the directory. A directory corresponding to the version of hadoop downloaded will be created under the /foo/bar/hadoop-install directory. For instance, if version 0.6.0 of Hadoop was downloaded untarring as defined above will create the directory /foo/bar/hadoop-0.6.0. the example in this paper assume the existence of an environment variable $HADOOP_INSTALL that represents the path to all versions of the Hadoop installed. In the above instance HADOOP_INSTALL=/foo/bar/hadoop-install. They futhur assume the existence of a symlink named hadoop in $HADOOP_INSTALL that points to the version of hadoop being used. For instance,if version 0.6.0 is being used then $HADOOP_INSTALL/hadoop-> hadoop-0.6.0. all tools used to run Hadoop will be present in the directory $HADOOP_INSTALL/hadoop/bin. All configuration files for Hadoop will be present in the directory $HADOOP_INSTALL/hadoop/conf.

5. CONCLUSION

Hence we can conclude that, Hadoop is a lot more flexible. Unlike parallel RDBMS you don't have to pre-process the data before you can use it. You don't have to design a star schema or update some data dictionary or manipulate it with a separate ETL process. Hadoop is more scalable. The largest publicly discussed Hadoop cluster (Facebook's) was at 30 petabytes mid last year and it's grown since then.Hadoop is more economical. Factoring in all the costs you can get down to ~$500 / TB pretty easily with Hadoop and even lower if cost is what really matters to you. There's no parallel RDBMS that comes close to those numbers to my knowledge. Since log, text and image data is often much bulkier than transactional data it's often been kept out of parallel RDBMS since the economics just wouldn't make any sense. Where as the basic limitations of Hadoop can be, that it is a framework, not a solution, also the deployment is easy, fast and free, but very costly to maintain and develop.

REFERENCES

1. http://www.michael-noll.com/wiki/Running_Hadoop_On_Ubuntu_Linux_(Single-Node_Cluster)
http://wiki.apache.org/hadoop/

Abhas Vashist et al.
A Promising future- Hadoop,
Indian Journal of Engineering, 2012, 1(1), 98-99,