ISBN 978-93-5156-328-0
International Conference of Advance Research and Innovation (ICARI-2014)

Hadoop: A Big Data Philosophy

Prashant Kumar [a,*], Khushboo Pandey [b]
[a] Department of Computer Science Engineering, Shri Venkateshwara University, Gajraula, U.P, India
[b] Department of Computer Science Engineering, DSITM, Gaziyabad, Utter Pradesh, India

Article Info

Article history:
Received 29 December 2013
Received in revised form
10 January 2014
Accepted 20 January 2014
Available online 1 February 2014

Keywords

Abstract

Apache Hadoop is an open-source software framework for storage and large scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users.[2] It is licensed under the Apache License 2.0.

1. Introduction

The Apache Hadoop framework is composed of the various modules, these are hereunder:

- Hadoop Common – It contains libraries and utilities needed by other Hadoop modules
- Hadoop Distributed File System (HDFS) –This is a distributed file-system that stores data on the commodity machines, providing very high aggregate bandwidth across the cluster.
- Hadoop YARN – It is a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
- Hadoop Map Reduce – This is a programming model for large scale data processing

All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines or racks of machines) are common and thus should be automatically handled in software by the framework. Apache Hadoop's MapReduce and HDFS components originally derived respectively from Google's MapReduce and Google File System (GFS) papers.

2. The Old Way: Data Analysis

Traditionally, data processing for analytic purposes followed a fairly static blueprint. Namely, through the regular course of business enterprises create modest amounts of structured data with stable data models via enterprise applications like CRM, ERP and financial systems. Data integration tools are used to extract, transform and load the data from enterprise applications and transactional databases to a staging area where data quality and data normalization (hopefully) occur and the data is modeled into neat rows and tables. The modeled, cleansed data is then loaded into an enterprise data warehouse. This routine usually occurs on a scheduled basis – usually daily or weekly, sometimes more frequently.

From there, data warehouse administrators create and schedule regular reports to run against normalized data stored in the warehouse, which are distributed to the business. They also create dashboards and other limited visualization tools for executives and management.

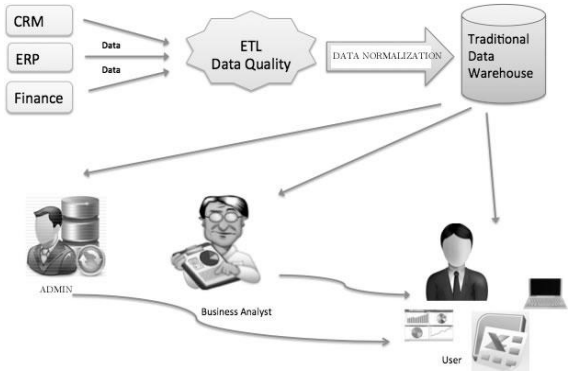

Fig: 1. Traditional data Processing

Business analysts, meanwhile, use data analytics tools/engines to run advanced analytics against the warehouse, or more often against sample data migrated to a local data mart due to size limitations. Non-expert business users perform basic data visualization and limited analytics against the data warehouse via front-end business intelligence

*** Corresponding Author,**
E-mail address: Prashant.iftn@gmail.com

tools from vendors like SAP Business Objects and IBM Cognos. Data volumes in traditional data warehouses rarely exceeded multiple terabytes (and even that much is rare) as large volumes of data strain warehouse resources and degrade performance.

3. Nature of Big Data

Big Data has important, distinct qualities that differentiate it from "outmoded" corporate data. No longer centralized, highly structured and easily manageable, now more than ever data is highly distributed, loosely structured (if structured at all), and increasingly large in volume

3.1. Traditional Approaches

The advent of the Web, mobile devices and other technologies has caused a fundamental change to the nature of data. Big Data has important, distinct qualities that differentiate it from "traditional" corporate data. No longer centralized, highly structured and easily manageable, now more than ever data is highly distributed, loosely structured (if structured at all), and increasingly large in volume
These are:

3.1.1. Volume – The amount of data created both inside corporations and outside the firewall via the web, mobile devices, IT infrastructure, and other sources is increasing exponentially each year.

3.1.2. Type – The variety of data types is increasing, namely unstructured text-based data and semi-structured data like social media data, location-based data, and log-file data.

3.1.3. Speed – The speed at which new data is being created – and the need for real-time analytics to derive business value from it -- is increasing thanks to digitization of transactions, mobile computing and the sheer number of internet and mobile device users.

The sources generated by BIG data are hereunder:

3.1.4. Social Networking Media: There are currently over 900 million Facebook users, 350 million Twitter users and 256 million public blogs. Each Facebook update, Tweet, blog post and comment creates multiple new data points - structured, semi-structured and unstructured - sometimes called Data Exhaust.

Mobile Devices: There are over 12 billion mobile phones in use worldwide. Each call, text and instant message is logged as data. Mobile devices, particularly smart phones and tablets, also make it easier to use social media and other data-generating applications. Mobile devices also collect and transmit location data.

3.1.5. Internet Communications: Millions of online purchases, stock trades and other transactions happen every day, including countless automated transactions. Each creates a number of data points collected by retailers, banks, credit cards, credit agencies and others.

3.1.6. Networked Campaigns and Instruments: Electronic devices of all sorts – including servers and other IT hardware, smart energy meters and temperature sensors -- all create semi-structured log data that record every action.

New Approaches to Big Data Processing and Analytics

4. New Approaches

There are number of approaches to processing and analysing Big Data, but most have some common features. Namely, they take advantage of commodity hardware to enable scale-out, parallel processing techniques; employ non-relational data storage capabilities in order to process unstructured and semi-structured data; and apply advanced analytics and data visualization technology to Big Data to convey insights to end-users

4.1. Hadoop: An Approach Towards Big Data Processing

Hadoop is an open source framework for processing, storing and analysing massive amounts of distributed, unstructured data. Originally created by Doug Cutting at Yahoo!, Hadoop was inspired by MapReduce, a user-defined function developed by Google in early 2000s for indexing the Web. It was designed to handle petabytes and Exabyte's of data distributed over multiple nodes in parallel.

Hadoop clusters run on inexpensive commodity hardware so projects can scale-out without breaking the bank. Hadoop is now a project of the Apache Software Foundation, where hundreds of contributors continuously improve the core technology.

Fundamental concept: Rather than banging away at one, huge block of data with a single machine, Hadoop breaks up Big Data into multiple parts so each part can be processed and analysed at the same time.

5. How Its Works

A client accesses unstructured and semi-structured data from sources including log files, social media feeds and internal data stores. It breaks the data up into "parts," which are then loaded into a file system made up of multiple nodes running on commodity hardware. The default file store in Hadoop is the Hadoop Distributed File System, or HDFS. File systems such as HDFS are adept at

storing large volumes of unstructured and semi-structured data as they do not require data to be organized into relational rows and columns.

Each "portion" is replicated multiple times and loaded into the file system so that if a node fails, another node has a copy of the data contained on the failed node. A Name Node acts as facilitator, communicating back to the client information such as which nodes are available, where in the cluster certain data resides, and which nodes have failed.

Once the data is loaded into the cluster, it is ready to be analysed via the MapReduce framework. The client submits a "Map" job -- usually a query written in Java – to one of the nodes in the cluster known as the Job Tracker. The Job Tracker refers to the Name Node to determine which data it needs to access to complete the job and where in the cluster that data is located. Once determined, the Job Tracker submits the query to the relevant nodes. Rather than bringing all the data back into a central location for processing, processing then occurs at each node simultaneously, or in parallel. This is an essential characteristic of Hadoop.

When the each node has finished processing its given job, it stores the results. The client initiates a "Reduce" job through the Job Tracker in which results of the map phase stored locally on individual nodes are aggregated to determine the "answer" to the original query, then loaded on to another node in the cluster. The client accesses these results, which can then be loaded into one of number of analytic environments for analysis. The MapReduce job has now been completed.

Once the MapReduce phase is complete, the processed data is ready for further analysis by Data Scientists and others with advanced data analytics skills. Data Scientists can manipulate and analyze the data using any of a number of tools for any number of uses, including to search for hidden insights and patterns or to use as the foundation to build user-facing analytic applications. The data can also be modeled and transferred from Hadoop clusters into existing relational databases, data warehouses and other traditional IT systems for further analysis and/or to support transactional processing.

6. Technical Components Of Hadoop

A Hadoop "hoard" is made up of a number of components. They include:

6.1. Hadoop Distributed File System (HDFS)

The default storage layer in any given Hadoop cluster;

6.2. Name Node

The node in a Hadoop cluster that provides the client information on where in the cluster particular data is stored and if any nodes fail;

Secondary Node: A backup to the Name Node, it periodically replicates and stores data from the Name Node should it fail;

6.3. Job Tracker

The node in a Hadoop cluster that initiates and coordinates MapReduce jobs, or the processing of the data

6.4. Slave Nodes

The grunts of any Hadoop cluster, slave nodes store data and take direction to process it from the Job Tracker.

6.5. NoSQL

A related new style of database called NoSQL (Not Only SQL) has emerged to, like Hadoop, process large volumes of multi-structured data. However, where as Hadoop is adept at supporting large-scale, batch-style historical analysis, NoSQL databases are aimed, for the most part (though there are some important exceptions) at serving up discrete data stored among large volumes of multi-structured data to end-user and automated Big Data applications. This capability is sorely lacking from relational database technology, which simply can't maintain needed application performance levels at Big Data scale.

In some cases, NoSQL and Hadoop work in conjunction. The aforementioned HBase, for example, is a popular NoSQL database modeled after Google BigTable that is often deployed on top of HDFS, the Hadoop Distributed File System, to provide low-latency, quick lookups in Hadoop.

7. Future for Big Data

Action Item: Enterprises across all industries should evaluate current and potential Big Data use cases and engage the Big Data community to understand the latest technological developments. Work with the community, like-minded organizations and vendors to identify areas Big Data can provide business value. Next, consider the level of Big Data skills within your enterprise to determine if you are in a position to begin experimenting with Big Data approaches like Hadoop. If so, engage both IT and the business to develop a plan to integrate Big Data tools, technology and approaches into your existing IT infrastructure. Most importantly, begin to cultivate a data-driven culture among workers at all levels and encourage data experimentation. When this foundation has been laid, apply Big Data

technologies and techniques today where they deliver the most business value and continuously re-evaluate new areas ripe for Big Data approaches.

IT vendors should help enterprises identify the most profitable and practical Big Data use cases, and develop products and services make Big Data technologies easier to deploy, manage and use. Embrace an open, rather than proprietary, approach, to give customers the flexibility needed to experiment with new Big Data technologies and tools.

Reference

[1] Hadoop Releases, Hadoop.apache.org. Retrieved 2013-04-08
[2] Applications and organizations using Hadoop, Wiki.apache.org. 2013-06-19 Retrieved 2013-10-17
[3] Hadoop-related projects at, 4, Hadoop.apache.org. Retrieved 2013-10-17
[4] [nlpatumd] Adventures with Hadoop and Perl, Mail-archive.com. 2010-05-02 Retrieved 2013-04-05
[5] Michael J. Cafarella, Web.eecs.umich.edu. Retrieved 2013-04-05
[6] Hadoop creator goes to Cloudera[dead link]
[7] Ashlee Vance, Hadoop, a Free 8 Software Program, Finds Uses Beyond Search, New York, Times Archived from the original on, 2010
[8] Hadoop contains the distributed computing platform that was formerly a part of Nutch, This includes the Hadoop Distributed Filesystem (HDFS) and an implementation of MapReduce, About Hadoop[dead link]
[9] Harsh Chouraria, MR2 and YARN Briefly Explained, cloudera.com. Cloudera, 2013
[10] HDFS User Guide, Hadoop.apache.org Retrieved 2012 [dead link]
[11] Running Hadoop on Ubuntu Linux (Multi-Node Cluster
[12] Running Hadoop on Ubuntu Linux, Single-Node Cluster, 2013
[13] HDFS Architecture, 2013
[14] Jump up to: a b Yaniv Pessach, Distributed Storage, Distributed Storage: Concepts, Algorithms, and Implementations ed., Amazon, 2013
[15] Provides for manual failover and they are working on automatic failover, 2.0 Hadoop.apache.org. 2013
[16] Improving MapReduce performance through data placement in heterogeneous Hadoop Clusters (PDF). Eng.auburn.ed. 2010
[17] HDFS Users Guide - Rack Awareness Hadoop.apache.org. Retrieved 2013
[18] http://www.slideshare.net/mcsrivas/design-scale-and-performance-of-maprs-distribution-for-hadoop
[19] http://www.mapr.com/products/mapr-editions
[20] Jump up http://aws.amazon.com/elasticmapreduce/mapr
[21] http://wikibon.org/wiki/v/Big_Data:_Hadoop,_Business_Analytics_and_Beyond