BIG DATA WITH R and Hadoop

COMPILED BY,

DR. PRAKASH KALINGRAO AITHAL

4 V Model of Big Data

- Volume: How much data is collected
- Veracity: How reliable the data is
- Velocity: How quickly data can be generated, gathered, and analyzed
- Variety: How many points of reference are used to collect data not = volume

Velocity

Velocity refers to the low latency, real-time speed at which the analytics need to be applied. A typical example of this would be to perform analytics on a continuous stream of data originating from a social networking site or aggregation of disparate sources of data.

Volume

Volume refers to the size of the dataset. It may be in KB, MB, GB, TB, or PB based on the type of the application that generates or receives the data.

Variety

Variety refers to the various types of the data that can exist, for example, text, audio, video, and photos.

Introducing Hadoop

Hadoop enables scalable, cost-effective, flexible, fault-tolerant solutions.

Features of Hadoop

Hadoop Distributed File System (HDFS)

MapReduce

Hadoop Ecosystem

- **Mahout**: This is an extensive library of machine learning algorithms.

- **Pig**: Pig is a high-level language (such as PERL) to analyze large datasets with its own language syntax for expressing data analysis programs, coupled with infrastructure for evaluating these programs.

- **Hive**: Hive is a data warehouse system for Hadoop that facilitates easy data summarization, ad hoc queries, and the analysis of large datasets stored in HDFS. It has its own SQL-like query language called **Hive Query Language** (**HQL**), which is used to issue query commands to Hadoop.

Hadoop Ecosystem

Learn from iPad PPT notes

HBase: **HBase** (**Hadoop Database**) is a distributed, column-oriented database. HBase uses HDFS for the underlying storage. It supports both batch style computations using MapReduce and atomic queries (random reads).

- **Sqoop**: Apache Sqoop is a tool designed for eficiently transferring bulk data between Hadoop and Structured Relational Databases. **Sqoop** is an abbreviation for (**SQ**)L to Had(**oop**).

- **ZooKeper**: ZooKeeper is a centralized service to maintain configuration information, naming, providing distributed synchronization, and group services, which are very useful for a variety of distributed systems.

Hadoop Ecosystem

Ambari: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters, which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig, and Sqoop.

Big Data Analytics Techniques

Regression: In statistics, regression is a classic technique to identify the scalar relationship between two or more variables by fitting the state line on the variable values. That relationship will help to predict the variable value for future events. For example, any variable y can be modeled as linear function of another variable x with the formula $y = mx+c$. Here, x is the predictor variable, y is the response variable, m is slope of the line, and c is the intercept. Sales forecasting of products or services and predicting the price of stocks can be achieved through this regression.

Big Data Analytics Techniques

Clustering: This technique is all about organizing similar items into groups from the given collection of items. User segmentation and image compression are the most common applications of clustering. Market segmentation, social network analysis, organizing the computer clustering, and astronomical data analysis are applications of clustering. Google News uses these techniques to group similar news items into the same category.

Big Data Analytics Techniques

Recommendation: The recommendation algorithms are used in recommender systems where these systems are the most immediately recognizable machine learning techniques in use today. Web content recommendations may include similar websites, blogs, videos, or related content. Also, recommendation of online items can be helpful for cross-selling and up-selling. We have all seen online shopping portals that attempt to recommend books, mobiles, or any items that can be sold on the Web based on the user's past behavior.

Big Data Analytics Techniques

Classification: This is a machine-learning technique used for labeling the set of observations provided for training examples. With this, we can classify the observations into one or more labels. The likelihood of sales, online fraud detection, and cancer classification (for medical science) are common applications of classification problems. Google Mail uses this technique to classify e-mails as spam or not.

Hadoop Installation Modes

The standalone mode: In this mode, you do not need to start any Hadoop daemons. Instead, just call `~/Hadoop-directory/bin/hadoop` that will execute a Hadoop operation as a single Java process. This is recommended for testing purposes. This is the default mode and you don't need to configure anything else. All daemons, such as NameNode, DataNode, JobTracker, and TaskTracker run in a single Java process.

Hadoop Installation Modes

The pseudo mode: In this mode, you configure Hadoop for all the nodes. A separate **Java Virtual Machine** (**JVM**) is spawned for each of the Hadoop components or daemons like mini cluster on a single host.

Hadoop Installation Modes

The fully distributed mode: In this mode, Hadoop is distributed across multiple machines. Dedicated hosts are configured for Hadoop components. Therefore, separate JVM processes are present for all daemons.

Characteristics of HDFS

- Fault tolerant
- Runs with commodity hardware
- Able to handle large datasets
- Master slave paradigm
- Write once read many times

community hardware =>
- fault tolerant -- master slave paradigm
- able to handle large datasets -- write once read many time

Map-Reduce

Hadoop MapReduce is a software framework for writing applications easily, which process large amounts of data (multiterabyte datasets) in parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. This MapReduce paradigm is divided into two phases, Map and Reduce that mainly deal with key and value pairs of data. The Map and Reduce task run sequentially in a cluster; the output of the Map phase becomes the input for the Reduce phase.

Map-Reduce

- **Map phase**: Once divided, datasets are assigned to the task tracker to perform the Map phase. The data functional operation will be performed over the data, emitting the mapped key and value pairs as the output of the Map phase.
- **Reduce phase**: The master node then collects the answers to all the subproblems and combines them in some way to form the output; the answer to the problem it was originally trying to solve.

Steps of Parallel Computing

1. Preparing the `Map()` input: This will take the input data row wise and emit key value pairs per rows, or we can explicitly change as per the requirement.

° Map input: list (k1, v1)

2. Run the user-provided `Map()` code

° Map output: list (k2, v2)

3. Shuflfe the Map output to the Reduce processors. Also, shuffle the similar keys (grouping them) and input them to the same reducer.

4. Run the user-provided `Reduce()` code: This phase will run the custom reducer code designed by developer to run on shuffled data and emit key and value.

° Reduce input: (k2, list(v2))

° Reduce output: (k3, v3)

5. Produce the final output: Finally, the master node collects all reducer output and combines and writes them in a text file.

HDFS Architecture

Name node – Master

Data nodes – Slaves

NameNode is a sever that manages the filesystem namespace and adjusts the access (open, close, rename, and more) to iles by the client. It divides the input data into blocks and announces which data block will be store in which DataNode. DataNode is a slave machine that stores the replicas of the partitioned dataset and serves the data as the request comes. It also performs block creation and deletion.

HDFS Architecture

The internal mechanism of HDFS divides the file into one or more blocks; these blocks are stored in a set of data nodes. Under normal circumstances of the replication factor three, the HDFS strategy is to place the first copy on the local node, second copy on the local rack with a different node, and a third copy into different racks with different nodes. As HDFS is designed to support large files, the HDFS block size is defined as 64 MB. If required, this can be increased.

HDFS Components

- **NameNode**: This is the master of the HDFS system. It maintains the directories, files, and manages the blocks that are present on the DataNodes.
- **DataNode**: These are slaves that are deployed on each machine and provide actual storage. They are responsible for serving read-and-write data requests for the clients.
- **Secondary NameNode**: This is responsible for performing periodic checkpoints. So, if the NameNode fails at any time, it can be replaced with a snapshot image stored by the secondary NameNode checkpoints.

MapReduce Architecture

MapReduce is also implemented over master-slave architectures. Classic MapReduce contains job submission, job initialization, task assignment, task execution, progress and status update, and job completion-related activities, which are mainly managed by the JobTracker node and executed by TaskTracker. Client application submits a job to the JobTracker. Then input is divided across the cluster. The JobTracker then calculates the number of map and reducer to be processed. It commands the TaskTracker to start executing the job. Now, the TaskTracker copies the resources to a local machine and launches JVM to map and reduce program over the data. Along with this, the TaskTracker periodically sends update to the JobTracker, which can be considered as the heartbeat that helps to update JobID, job status, and usage of resources.

MapReduce Components

• **JobTracker**: This is the master node of the MapReduce system, which manages the jobs and resources in the cluster (TaskTrackers). The JobTracker tries to schedule each map as close to the actual data being processed on the TaskTracker, which is running on the same DataNode as the underlying

block.

• **TaskTracker**: These are the slaves that are deployed on each machine. They are responsible for running the map and reducing tasks as instructed by the JobTracker.

map, reduce both done by TaskTracker

References

Vignesh Prajapathi, Big Data Analytics with R and Hadoop, Packt Publishing, 2013.