A

Seminar report

On

Hadoop

Submitted in partial fulfillment of the requirement for the award of degree
of Bachelor of Technology in Computer Science

SUBMITTED TO:
www.studymafia.org

SUBMITTED BY:
www.studymafia.org

Acknowledgement

I would like to thank respected Mr…….. and Mr. ……..for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

Preface

I have made this report file on the topic **Hadoop**; I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude to ……..who assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

Introduction

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

What is Hadoop?

Hadoop is sub-project of Lucene (a collection of industrial-strength search tools), under the umbrella of the Apache Software Foundation. Hadoop parallelizes data processing across many *nodes* (computers) in a *compute cluster*, speeding up large computations and hiding I/O latency through increased concurrency. Hadoop is especially well-suited to large data processing tasks (like searching and indexing) because it can leverage its distributed file system to cheaply and reliably replicate chunks of data to nodes in the cluster, making data available locally on the machine that is processing it.

Hadoop is written in Java. Hadoop programs can be written using a small API in Java or Python. Hadoop can also run binaries and shell scripts on nodes in the cluster provided that they conform to a particular convention for string input/output.

Hadoop provides to the application programmer the abstraction of map and reduce (which may be familiar to those with functional programming experience). Map and reduce are available in many languages, such as Lisp and Python.

Hadoop Applications

Making Hadoop Applications More Widely Accessible

Apache Hadoop, the open source MapReduce framework, has dramatically lowered the cost barriers to processing and analyzing big data. Technical barriers remain, however, since Hadoop applications and technologies are highly complex and still foreign to most developers and data analysts. Talend, the open source integration company, makes the massive computing power of Hadoop truly accessible by making it easy to work with Hadoop applications and to incorporate Hadoop into enterprise data flows.

A Graphical Abstraction Layer on Top of Hadoop Applications

In keeping with our history as an innovator and leader in open source data integration, Talend is the first provider to offer a pure open source solution to enable big data integration. Talend Open Studio for Big Data, by layering an easy to use graphical development environment on top of powerful Hadoop applications, makes big data management accessible to more companies and more developers than ever before.

With its Eclipse-based graphical workspace, Talend Open Studio for Big Data enables the developer and data scientist to leverage Hadoop loading and processing technologies like HDFS, HBase, Hive, and Pig without having to write Hadoop application code. By simply selecting graphical components from a palette, arranging and configuring them, you can create Hadoop jobs that, for example:

- Load data into HDFS (Hadoop Distributed File System)
- Use Hadoop Pig to transform data in HDFS
- Load data into a Hadoop Hive based data warehouse
- Perform ELT (extract, load, transform) aggregations in Hive
- Leverage Sqoop to integrate relational databases and Hadoop

Hadoop Applications, Seamlessly Integrated

For Hadoop applications to be truly accessible to your organization, they need to be smoothly integrated into your overall data flows. Talend Open Studio for Big Data is the ideal tool for integrating Hadoop applications into your broader data architecture. Talend provides more built-in connector components than any other data integration solution available, with more than 800 connectors that make it easy to read from or write to any major file format, database, or packaged enterprise application.

For example, in Talend Open Studio for Big Data, you can use drag 'n drop configurable components to create data integration flows that move data from delimited log files into Hadoop Hive, perform operations in Hive, and extract data from Hive into a MySQL database (or Oracle, Sybase, SQL Server, and so on).

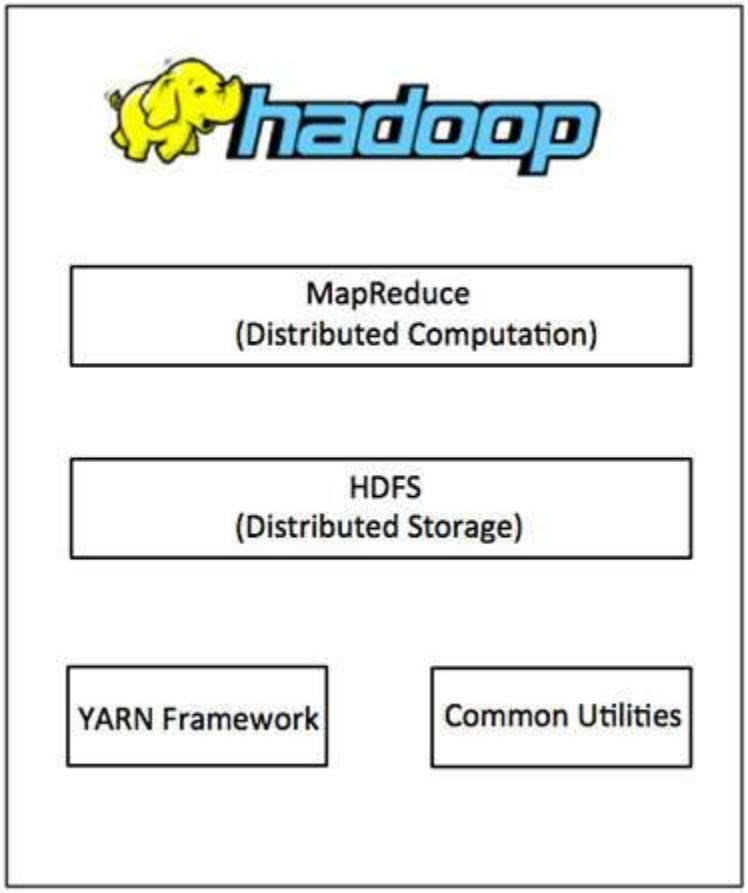
Want to see how easy it can be to work with cutting-edge Hadoop applications? No need to wait -- Talend Open Studio for Big Data is open source software, free to download and use under an Apache license.

Hadoop Architecture

Hadoop framework includes following four modules:

- **Hadoop Common:** These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.
- **Hadoop YARN:** This is a framework for job scheduling and cluster resource management.
- **Hadoop Distributed File System (HDFS™):** A distributed file system that provides high-throughput access to application data.
- **Hadoop MapReduce:** This is YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop framework.



Since 2012, the term "Hadoop" often refers not just to the base modules mentioned above but also to the 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 etc.

MapReduce

Hadoop **MapReduce** is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

- **The Map Task:** This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).
- **The Reduce Task:** This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The MapReduce framework consists of a single master **JobTracker** and one slave **TaskTracker** per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves TaskTracker execute the tasks as directed by the master and provide task-status information to the master periodically.

The JobTracker is a single point of failure for the Hadoop MapReduce service which means if JobTracker goes down, all running jobs are halted.

Why is Hadoop important?

- **Ability to store and process huge amounts of any kind of data, quickly.** With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.
- **Computing power.** Hadoop's distributed computing model processes big data fast. The more computing nodes you use, the more processing power you have.
- **Fault tolerance.** Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically.
- **Flexibility.** Unlike traditional relational databases, you don’t have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. That includes unstructured data like text, images and videos.
- **Low cost.** The open-source framework is free and uses commodity hardware to store large quantities of data.
- **Scalability.** You can easily grow your system to handle more data simply by adding nodes. Little administration is required.

Hadoop Distributed File System

Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture where master consists of a single **NameNode** that manages the file system metadata and one or more slave **DataNodes** that store the actual data.

A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write operation with the file system. They also take care of block creation, deletion and replication based on instruction given by NameNode.

HDFS provides a shell like any other file system and a list of commands are available to interact with the file system. These shell commands will be covered in a separate chapter along with appropriate examples.

How Does Hadoop Work?

Stage 1

A user/application can submit a job to the Hadoop (a hadoop job client) for required process by specifying the following items:

1. The location of the input and output files in the distributed file system.
2. The java classes in the form of jar file containing the implementation of map and reduce functions.
3. The job configuration by setting different parameters specific to the job.

Stage 2

The Hadoop job client then submits the job (jar/executable etc) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Stage 3

The TaskTrackers on different nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system.

Advantages of Hadoop

1. Scalable

Hadoop is a highly scalable storage platform, because it can stores and distribute very large data sets across hundreds of inexpensive servers that operate in parallel. Unlike traditional relational database systems (RDBMS) that can't scale to process large amounts of data, Hadoop enables businesses to run applications on thousands of nodes involving many thousands of terabytes of data.

2. Cost effective

Hadoop also offers a cost effective storage solution for businesses' exploding data sets. The problem with traditional relational database management systems is that it is extremely cost prohibitive to scale to such a degree in order to process such massive volumes of data. In an effort to reduce costs, many companies in the past would have had to down-sample data and classify it based on certain assumptions as to which data was the most valuable. The raw data would be deleted, as it would be too cost-prohibitive to keep. While this approach may have worked in the short term, this meant that when business priorities changed, the complete raw data set was not available, as it was too expensive to store.

3. Flexible

Hadoop enables businesses to easily access new data sources and tap into different types of data (both structured and unstructured) to generate value from that data. This means businesses can use Hadoop to derive valuable business insights from data sources such as social media, email conversations. Hadoop can be used for a wide variety of purposes, such as log processing, recommendation systems, data warehousing, market campaign analysis and fraud detection.

4. Fast

Hadoop's unique storage method is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing. If you're dealing with large volumes of unstructured data, Hadoop is able to efficiently process terabytes of data in just minutes, and petabytes in hours.

5. Resilient to failure

A key advantage of using Hadoop is its fault tolerance. When data is sent to an individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available for use.

Disadvantages of Hadoop:

As the backbone of so many implementations, Hadoop is almost synomous with big data.

1. Security Concerns

Just managing a complex applications such as Hadoop can be challenging. A simple example can be seen in the Hadoop security model, which is disabled by default due to sheer complexity. If whoever managing the platform lacks of know how to enable it, your data could be at huge risk. Hadoop is also missing encryption at the storage and network levels, which is a major selling point for government agencies and others that prefer to keep their data under wraps.

2. Vulnerable By Nature

Speaking of security, the very makeup of Hadoop makes running it a risky proposition. The framework is written almost entirely in Java, one of the most widely used yet controversial programming languages in existence. Java has been heavily exploited by cybercriminals and as a result, implicated in numerous security breaches.

3. Not Fit for Small Data

While big data is not exclusively made for big businesses, not all big data platforms are suited for small data needs. Unfortunately, Hadoop happens to be one of them. Due to its high capacity design, the Hadoop Distributed File System, lacks the ability to efficiently support the random reading of small files. As a result, it is not recommended for organizations with small quantities of data.

4. Potential Stability Issues

Like all open source software, Hadoop has had its fair share of stability issues. To avoid these issues, organizations are strongly recommended to make sure they are running the latest stable version, or run it under a third-party vendor equipped to handle such problems.

5. General Limitations

The article introducesApache Flume, MillWheel, and Google's own Cloud Dataflow as possible solutions. What each of these platforms have in common is the ability to improve the efficiency and reliability of data collection, aggregation, and integration. The main point the article stresses is that companies could be missing out on big benefits by using Hadoop alone.

How Is Hadoop Being Used?

Low-cost storage and data archive

The modest cost of commodity hardware makes Hadoop useful for storing and combining data such as transactional, social media, sensor, machine, scientific, click streams, etc. The low-cost storage lets you keep information that is not deemed currently critical but that you might want to analyze later.

Sandbox for discovery and analysis

Because Hadoop was designed to deal with volumes of data in a variety of shapes and forms, it can run analytical algorithms. Big data analytics on Hadoop can help your organization operate more efficiently, uncover new opportunities and derive next-level competitive advantage. The sandbox approach provides an opportunity to innovate with minimal investment.

Data lake

Data lakes support storing data in its original or exact format. The goal is to offer a raw or unrefined view of data to data scientists and analysts for discovery and analytics. It helps them ask new or difficult questions without constraints. Data lakes are not a replacement for data warehouses. In fact, how to secure and govern data lakes is a huge topic for IT. They may rely on data federation techniques to create a logical data structures.

Complement your data warehouse

We're now seeing Hadoop beginning to sit beside data warehouse environments, as well as certain data sets being offloaded from the data warehouse into Hadoop or new types of data going directly to Hadoop. The end goal for every organization is to have a right platform for storing and processing data of different schema, formats, etc. to support different use cases that can be integrated at different levels.

IoT and Hadoop

Things in the IoT need to know what to communicate and when to act. At the core of the IoT is a streaming, always on torrent of data. Hadoop is often used as the data store for millions or billions of transactions. Massive storage and processing capabilities also allow you to use Hadoop as a sandbox for discovery and definition of patterns to be monitored for prescriptive instruction. You can then continuously improve these instructions, because Hadoop is constantly being updated with new data that doesn’t match previously defined patterns.

Reference

- www.google.com
- www.wikipedia.org
- www.studymafia.org