Mahout and R are used for the same purpose. They are the analytical tools to find the hidden patterns in the data.

**Hive:**

Hive is often referred to as the data warehouse for a Hadoop cluster. Hive provides the data integrity and schema for the data. It can structure the data and store it in HDFS file system. Data is not indexed in Hadoop but in can be indexed in database world. In Hadoop you cannot do row level operation. You can update or delete the data. Hive support most of the data types supported by databases. In Hive you can create 2 types of tables, Managed or Internal tables and External tables. When you drop internal table data and metadata will be lost. When you drop the external table only the metadata is lost. Hive file formats are Avro, ORC, RegEx, Parquet and CSV. You can store the data in any of these formats in Hive. Hive is most commonly used with HBase

**Impala:**

Impala is pretty similar to Hive with some key differences. It’s a MPP SQL query engine for processing huge volumes of data that is stored in Hadoop clusters. Its written in C++. With Impala users can communicate with HDFS or HBase using SQL queries in a much faster way compared to other SQL engines like Hive. Impala can read almost all the file formats as Hive. The Key difference between Impala and Hive is that, the former doesn’t use MapReduce algorithms. It implements a distributed architecture based on daemon processes that are responsible for all the aspects of query execution that run on the same machines. Impala works best with Parquet file format (columnar storage).

**Sqoop** is used for importing and exporting data from RDBMS

**Oozie:**

Apache Oozie can be used to design complex job workflows and coordinate them to occur at regular intervals. They are similar to ETL Jobs in database world. Some of the examples of the workflows are, partitioning the Hive table every one hour, Execute a map reduce job, run a java main class or run a Hive or Pig script on a regular basis. Oozie workflows are written as an XML file representing a directed acyclic graph (DAG). Hue makes it easy to design and execute complicated data flows by using its UI application.

**Flume:**

Apache Flume is a tool/service/date ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log data, events etc from various web services to a centralized data store like HDFS and HBase

Here is an example:

We want to import all the latest news from the RSS feed. The steps are

Make a request to pull the RSS feed data (Python Program)

Use the Flume to execute the Python script. There are 3 things need to be configured here Source, Channel and Sink. Source is your python script, channel can be a memory or the disk based on how you want to store the data. Sink can be a machine where the data should be loaded to. In our case we will load it a machine and use spark steaming to process these records. Spark operate in batch and process them every 30 seconds. The results could be dumped to HDFS.

Difference between SORT BY and ORDER BY:

Sort by will sort the data only in selected reducers. ORDER BY will sort the data in all the reducers

Parallel data warehouse is one other choice if we don’t want to use Hadoop. It has its own advantages and disadvantages. PDW mainly accepts the data in a particular format or schema. Hadoop on the other end is not schema bound. It can accept the data in any format.

Technologies to learn:

Mahout, Sqoop, Zookeeper, Ambari, Oozie, Solr

HBase, Cassandra, MongoDB

Cloudera, Hue

Maven, Git

Languages to learn Python, GO, Node JS

Scheduling tool AutoSys

Javascript API, Rest API

Mesos, Marathon, Docker, RabbitMQ, Puppet

Digital Media: Operative, DFP, Moat, Krux, Google Analytics, CRM

How does the Linux permissions work?

<https://www.digitalocean.com/community/tutorials/an-introduction-to-linux-permissions>

What is spark RDD? What are its advantages compared to regular data sets?

<https://www.youtube.com/watch?v=V1sPV2dReZA>

Difference between Spark and Hadoop

<https://www.youtube.com/watch?v=mUQ1Yp8R6xI>

Tutorial on SparkSQL:

<https://www.youtube.com/watch?v=KiAnxVo8aQY>

Spark replaces map-reduce in Hadoop ecosystem. All the computations can be done in memory (and also on disk if needed). As there is less IO to the disk Spark runs much faster. It is up to 100 times faster than map-reduce queries. Even when you write the data to the disk it’s 2-5 times faster. Data once written to the disk it becomes Immutable. It cannot be changed. If you are running some transformations the new modified data should be loaded into a new RDD. Which can be stored in memory or written back to the disk. Spark is used to eliminate redundant phases like Map, Shuffle and Reduce.

How spark jobs are run in production?

Running Spark on Cluster

Package the code into a JAR file

submit the JAR file to the spark-submit command. It distributes the jar and the job to the clusters

spark-submit is used for deployment and running jobs in production. we can wrap it into a cron job and run on schedule basis

SBT is a more advanced way of packaging your scala projects. It packages all the dependancies into a one neat JAR file, which can be executed on any machine.

Copy data from relational database to HDFS using Sqoop

<https://www.youtube.com/watch?v=sAMUyNkX_cM&list=PLlP5IEabs4dvXinG6-8gKHPZ2eQC0EW6J&index=4>

Hive tutorials:

<https://www.youtube.com/watch?v=q2vK3VdSbIw&list=PLsxbzAdvhorg3kKYqVXEvOGreyGvW-k-i&index=2>

What are the cases in which Hive cannot be used?

What are the differences between Pig and Hive?

<http://stackoverflow.com/questions/3356259/difference-between-pig-and-hive-why-have-both>

<http://www.aptibook.com/Articles/Pig-and-hive-advantages-disadvantages-features>

What is Hue?

<https://www.youtube.com/watch?v=GrNZ9ZcJZcE>

What is MapR? What is its architecture?

<https://www.mapr.com/blog/in-depth-look-hbase-architecture>

Difference between the Linux file system and Hadoop file system

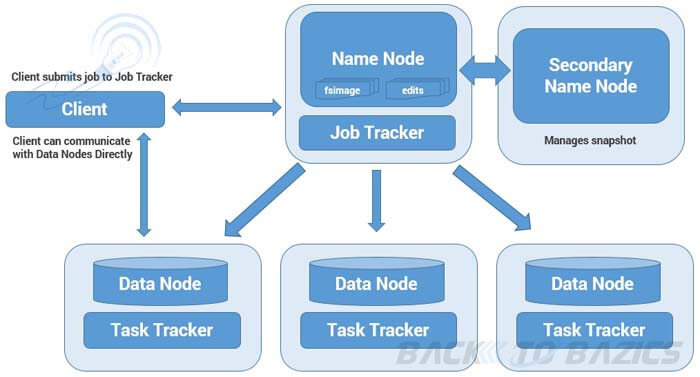
<https://www.youtube.com/watch?v=ZYQhHrq5kj8>

Can you explain about the HDFS architecture?

<https://www.youtube.com/watch?v=DLutRT6K2rM>

The block size of HDFS is 64MB. If the block is not fully loaded it can be reused for a different file. Components of HDFS are

1. Name Node
2. Secondary Name Node
3. Job Tracker
4. Data Node
5. Task Tracker



Zookeeper: It is used for managing clusters.

Kafka: It is a distributed publish subscribe messaging solution. It’s fast, scalable and durable compared to the traditional messaging systems. Ex: for the trading system where there are million transactions per second, all the activities like placing the order, paying, checking with the bank etc doesn’t need to be done at the same time. First an order can be placed and the formalities of fulfilling can be handled during off peak hours. This is where Kafka comes into picture. It will queue different tasks and perform them asynchronously at different times.

Both Zookeeper and Kafka are for transactional systems and they are not of use for Data Warehouse or analytical applications

HBase, Cassandra and MongoDB are all No SQL databases.

How to choose the right NO SQL database

<https://www.youtube.com/watch?v=gJFG04Sy6NY>

Job Search keywords: Big Data, Hive, Pig, Scala, Spark

Information about Cassandra database

<https://www.youtube.com/watch?v=4WX2MZvmzho>

**Hadoop Distributions:**

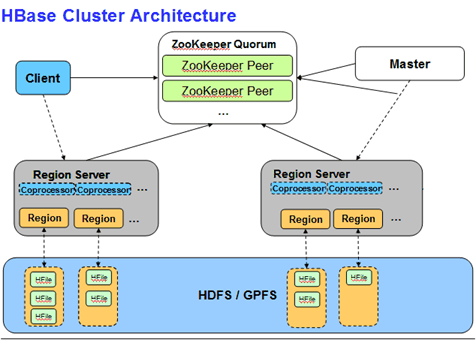
Hadoop is an open source platform. A number of vendors have taken advantage of hadoop’s open ended framework and tweaked its codes to change and enhance its functionalities. In the process they have been able to fix the inherent drawbacks of Apache Hadoop. There are mainly 3 companies that distribute Hadoop with their own customizations. They are Cloudera, MapR and Hortonworks

Cloudera: Although the core of the distribution is based on Apache Hadoop, it also provides a proprietary Cloudera Management Suite to automate the installation process and provide other services to enhance convenience of users which include reducing deployment time, displaying real time nodes count, etc.

**Analyzing Twitter data with Hadoop:**

<http://blog.cloudera.com/blog/2012/09/analyzing-twitter-data-with-hadoop/>

**HBase:** HBase is a Key-Value pair NOSQL database. It stores the data in the memory and periodically writes to the disk. Data is split as shards and stored in individual regions inside a regional server. Each regional server can hold multiple regions belonging to the same or different tables. When the data is loaded to the disk it is written in the form of H file. Zookeeper works in the background and takes cares of the tasks like Load Balancing, Resource management, Fault tolerance and replication. If it is down clusters won’t work. All the rows are indexed and ordered by the row key. HBase can be connected using GUI like Hue, HBase shell, Hive, Pig or Java related APIs



HBase data storage

<http://www.larsgeorge.com/2009/10/hbase-architecture-101-storage.html>

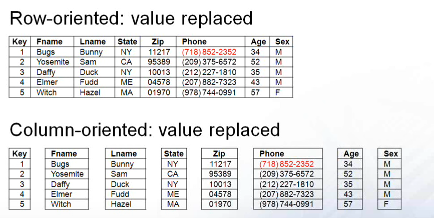
**MPP (Massive Parallel Processing):**

In massive parallel processing, data is divided into data partitions. Those partitions are spread across independent computing nodes. Nodes have their own RAM, CPUs and their own storage. Control node marshals computing nodes which work on their individual piece of overall query. Control node consolidates partition results into one result set which can be returned to the client. Examples of SQL MPP are Microsoft PDW, Teradata, Netezza, Redshift etc. Examples of Non-SQL MPP are BigTable, Hadoop, Spark etc

**Columnar Database:**

Columnar databases produce automatic vertical partitioning of data. Each column is stored in a separate file. By doing so we can read only the columns or the files that we need and improve the performance. As each file contain data belonging to a particular data type high level of compression can also be achieved. We can apply filtered indexing on the columns where you want to search for a range of values. As the data is partitioned queries will access only the partitions that has the data. In this way it improves the performance of the query. Adding a column is easier and very efficient in this structure. It is just like adding a new file. In a regular row based storage adding a column will result in all the data in the pages shifting to accommodate the new column. This is heavy resource intensive.

Columnar databases cannot be used for OLTP transactions where we will be accessing all the contents of a single row at a time. It is mainly used for OLAP applications. It is also not suitable for single Insert, Update or Delete transaction. In a row based storage it deletes the record from a single file. But in columnar those operations are performed on multiple files as each column is on a separate file.



**Sharding**: It is the process of splitting the data across multiple machine. It is horizontal partitioning of the data. It will help store large amounts of data across multiple nodes and to add capacity as needed. This improves performance. As we will be using the data only from a few set of partitions in most of the reads it improves the performance of the query.

**Document database:**

In the document databases, the basic unit of data is document. It follows the key-value pair structure but it is not a key-value database. MongoDB is one of the examples of document database. JSON is the standard format of the data that we deal in these databases. Documents are grouped into collections. Document databases are schema less. Document databases are used for event logging, blogs, websites, web analytics/real time analytics or E-Commerce applications. Other examples of document databases are CouchDB, Redis and Apache Cassandra

What is the difference between RDBMS and Document databases?

RDBMS are relational databases. They have tables and the schema is predefined. On the other hand document databases are schema less. If you are constantly adding or removing new columns then document databases are the right choice. In document databases, collections are equivalent to tables in RDBMS. Joining the tables is easy in RDBMS but in document databases it’s not that easy. Document databases provide the addition advantage of database sharding where the data is split into blocks and spread across multiple machine. This increases the performance significantly.

**More on Spark:**

A form of transformation that can be applied to Map Reduce or RDDs is to Import the data, convert it into String, parse out the XML, extract the elements out of it, and remove the characters that we don't need. Do some aggregations and load the results to some file or database. Transformation create new RDDs in spark. Actions are operations that return a result to the driver program or write it to storage and kick off a computation such as count() and first(). Another point to note is, spark won’t start execution until it sees an action.

Free Mongodb on cloud

<https://mlab.com/databases/mongdb#users>

shell> mongo ds145158.mlab.com:45158/mongdb -u <dbuser> -p <dbpassword>

user: k\_rasuri

pwd: submit27

**Cloudera**, user: [k\_rasuri@yahoo.com](mailto:k_rasuri@yahoo.com), pwd: Submit27

Cloudera Manager

<http://ec2-54-163-135-151.compute-1.amazonaws.com:7180>

Hue:

<http://ec2-54-163-135-151.compute-1.amazonaws.com:8888>

User: admin

User: admin

[www.pluralsight.com](http://www.pluralsight.com)

krasuri

submit27

**Questions:**

What is spark Architecture? What are its components?

What is Hadoop job tracker?

What is Flume and what are its uses?

What is the difference between MPP and Hadoop systems?

Difference between Document, Columnar NoSQL, Key Value and Graph NoSQL databases

What is Yarn? What is the difference between MapReduce and Yarn?

How does the indexing work in Hadoop?

How does the columnar databases work?

What is redshift architecture?

What do you know about Amazon EMR (Elastic Map Reduce)?

How does the JVM work and how it helps in improving performance?

List the differences between Hadoop 1.x and 2.x?

How does JOINS work in Hive?

What are the advantages and disadvantages of internal and external tables in HIVE?

What is the difference between static and dynamic partitioning in HIVE?

Explain the differences between different file formats (RC, Parquet, Avro etc)in Hadoop? When are they used?

What is the difference between Flume and Spark Streaming?

Write about the Spark Internals?

What is Lambda Architecture in Hadoop?

How do you do the incremental updates in Hadoop?

Explain how DStreams work in Spark Streaming?

What are the differences between spark 1.x and 2.0?

How do you load XML file into Hadoop?

How do you create sub workflows using Oozie? How do you create oozie job manually?

How do you loop through files using java?

How to create an ETL job using Spark?

What is HCatalog and what is its purpose?

How do you do the Data Quality Checks in HIVE, IMPALA and Scala?

What is Cloudera Navigator?

What is Cloudera Centry and how it can be used for permissions?

**Projects:**

Proj1:

Install MySQL on Ubuntu

Connect to MySQL from client machine, Create database and tables

Import data using Apache Scoop

Change Hadoop to Pseudo-Mode

Create the workflow using Oozie?

Proj2:

Install MongoDB

Proj3:

Install CDH5

Run basic map reduce program

Proj4:

Install Spark

Write Spark programs

<https://www.youtube.com/watch?v=YZnNb0BTrS4>

<https://www.cloudera.com/documentation/enterprise/5-6-x/topics/spark_first.html>

SparkQL

Install Machine Learning packages and run programs

Proj5:

HBase

**Courses:**

Hive and Sqoop course on Udemy

Spark course on Udemy

HBase course on PluralSight

**TroubleShooting:**

To fix the memory issues with spark

<https://community.cloudera.com/t5/Advanced-Analytics-Apache-Spark/Hive-on-Spark-CDH-5-7-Failed-to-create-spark-client/td-p/39831>

Spark word count sample code

val myfile = sc.textFile("/user/cloudera/wordcount/input")

val counts = myfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_ + \_)

counts.saveAsTextFile("/user/cloudera/wordcount/output")

**Configure and execute HIVE on Spark (Shark)**

<https://www.youtube.com/watch?v=morvk4pI5OM>

<https://www.tutorialspoint.com/spark_sql/spark_sql_hive_tables.htm>

Using HIVE on Apache Spark

<https://hortonworks.com/hadoop-tutorial/using-hive-with-orc-from-apache-spark/>

https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.4.0/bk\_spark-guide/content/hive-on-spark.html

How spark gives better performance compared to Hadoop?

<https://www.youtube.com/watch?v=U8xQnGslkFI>

Difference between the Avro and Parquet data formats

<https://www.youtube.com/watch?v=AY1dEfyFeHc>

Loading data in Avro and Parquet format

<https://acadgild.com/blog/parquet-file-format-hadoop/>

https://acadgild.com/blog/working-with-avro-and-parquet-files-in-spark/

Importing data from twitter to HDFS using Flume

<https://www.youtube.com/watch?v=PdY31i25SL0>

Streaming data from YouTube project

<https://github.com/DataDanSandler/YouTubeREADME>

**Login into MySQL** (account: root, pwd: admin01)

mysql –u root –p

Enter

Give the password

Difference between Kafka and Flume

<http://blog.cloudera.com/blog/2014/09/apache-kafka-for-beginners/>

<https://www.eduonix.com/blog/bigdata-and-hadoop/flume-installation-and-streaming-twitter-data-using-flume/>

Difference between Hadoop 1.0 and 2.0?

<https://acadgild.com/blog/10-big-differences-between-hadoop1-and-hadoop2/>

Spark Tutorials

<https://www.youtube.com/watch?v=3OsVa28ZSjc&index=5&list=PLbu9W4c-C0iB--RNIzTXVksxQEI4vLK5->

Create spark application on Maven

<https://www.youtube.com/watch?v=aB4-RD_MMf0>

https://www.youtube.com/watch?v=\_15v5V\_nlPQ

Package and deploy spark program using sbt

<https://www.youtube.com/watch?v=1BeTWT8ADfE>

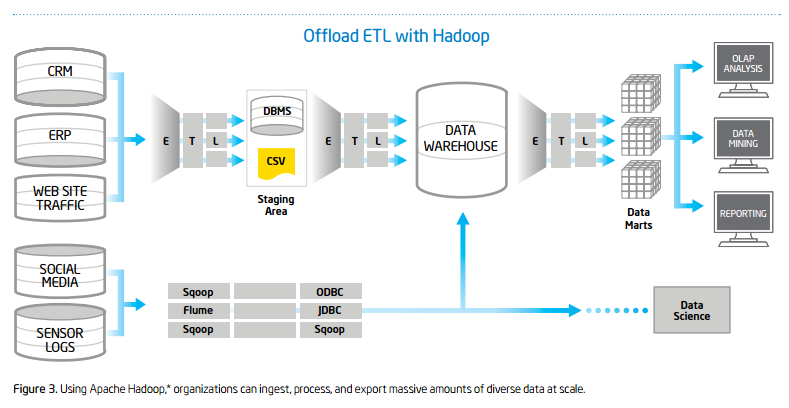
Data Warehousing and ETL process in Hadoop

<https://www.youtube.com/playlist?list=PLf0swTFhTI8q5X0ZBeUaK1yl_TP-tJInC>

Slowly changing dimensions in Hadoop

<https://community.hortonworks.com/articles/48843/slowly-changing-dimensions-on-hadoop-part-1.html>

<https://united.softserveinc.com/blogs/process-slowly-changing-dimensions-hive/>



Schedule the spark job

<http://stackoverflow.com/questions/30375571/running-scheduled-spark-job>

Scheduling the spark jobs with Oozie

<https://aws.amazon.com/blogs/big-data/use-apache-oozie-workflows-to-automate-apache-spark-jobs-and-more-on-amazon-emr/>

Apache Kafka Tutorials (Theory)

<https://www.youtube.com/watch?v=gg-VwXSRnmg&list=PLkz1SCf5iB4enAR00Z46JwY9GGkaS2NON>

Comparing Zeppelin, Spark Notebook and Jupyter

<http://queirozf.com/entries/comparing-interactive-solutions-for-running-apache-spark-zeppelin-spark-notebook-and-jupyter-scala>

Spark Streaming example. Loading from Socket

<https://www.youtube.com/watch?v=P7KBCgAGY9Q>

Data parallelism in Scala

<http://nikgrozev.com/2016/10/23/summary-of-the-parallel-programming-with-scala-mooc/>

<https://www.slideshare.net/knoldus/scala-parallelcollections-13883706?next_slideshow=1>

<https://www.slideshare.net/AleksandarProkopec/scala-parallel-collections>

Why do we need Yarn?

<https://www.youtube.com/watch?v=nmaA5_d4E8c>

What is JVM?

<http://www.artima.com/insidejvm/ed2/jvm.html>

What is fair scheduler and capacity scheduler?

Difference between the narrow and wide dependencies

<https://www.youtube.com/watch?v=LDdA1RW_6xo>

How do you reduce the shuffle and improve the performance in Spark

<https://www.youtube.com/watch?v=kbQmZiT1gnA>

Parse XML using Spark Data Frames

<https://www.youtube.com/watch?v=NemEp53yGbI>

Analyzing JSON data in HIVE

<https://www.youtube.com/watch?v=i4pHvftawtw>

Installing the Kafka Server

<https://www.digitalocean.com/community/tutorials/how-to-install-apache-kafka-on-ubuntu-14-04>

How to write text file to Kafka producer

<https://stackoverflow.com/questions/33273587/how-to-write-a-file-to-kafka-producer>

Very good videos on Spark and Hadoop videos

<https://www.youtube.com/channel/UCdXbm4fY-fqZcH2Ht49ctyA/videos>

aggregateByKey example:

scala> val babyNamesCSV = sc.parallelize(List(("David", 6), ("Abby", 4), ("David", 5), ("Abby", 5)))

scala> babyNamesCSV.reduceByKey((n,c) => n + c).collect

res0: Array[(String, Int)] = Array((Abby,9), (David,11))

scala> babyNamesCSV.aggregateByKey((0, 0))(

(acc, value) => (acc.\_1 + value, acc.\_2 + 1),

(acc1, acc2) => (acc1.\_1 + acc2.\_1, acc1.\_2 + acc2.\_2))

.mapValues(sumCount => 1.0 \* sumCount.\_1 / sumCount.\_2)

.collect

res1: Array[(String, Double)] = Array((Abby,4.5), (David,5.5))

Another example on aggregateByKey:

<https://stackoverflow.com/questions/24804619/how-does-spark-aggregate-function-aggregatebykey-work>

Partitioning in Spark

<https://www.youtube.com/watch?v=AK1khvHMUvE>

Why do we need spark accumulators?

<https://www.edureka.co/blog/spark-accumulators-explained>

Difference Narrow and Wide dependencies

<https://www.coursera.org/learn/scala-spark-big-data/lecture/shGAX/wide-vs-narrow-dependencies>

Tuning and debugging Spark

<https://www.youtube.com/watch?v=kkOG_aJ9KjQ>

http://blog.cloudera.com/blog/2015/03/how-to-tune-your-apache-spark-jobs-part-1/

<http://blog.cloudera.com/blog/2015/03/how-to-tune-your-apache-spark-jobs-part-2/>

Spark under the hood

<https://virtuslab.com/blog/spark-sql-hood-part-i/>

How to partition external tables

<http://blog.zhengdong.me/2012/02/22/hive-external-table-with-partitions/>