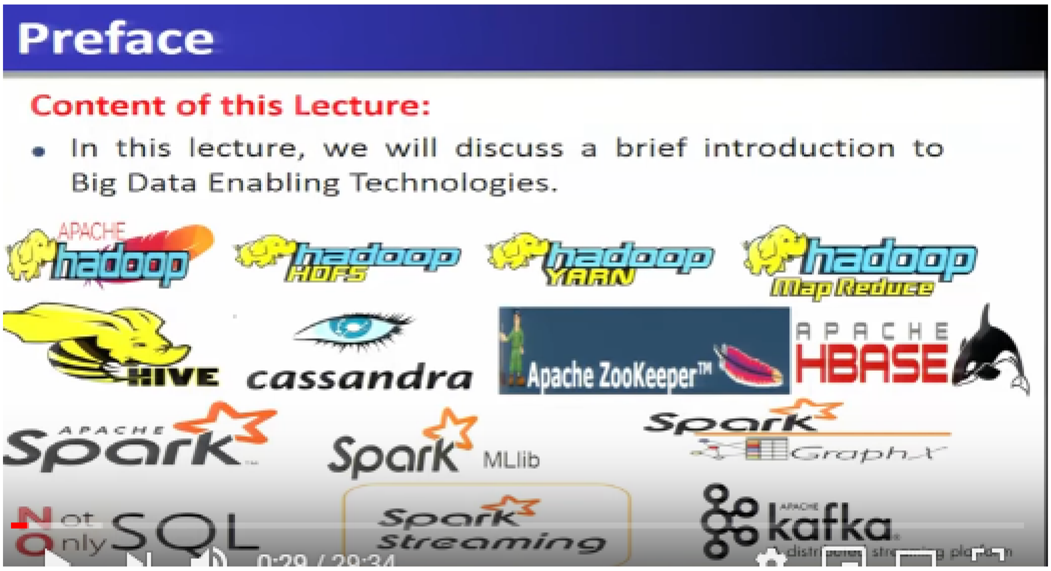
# Lecture 2

# Big Data Enabling Technologies

Big Data Enabling Technologies
Refer Slide Time :( 0:17)



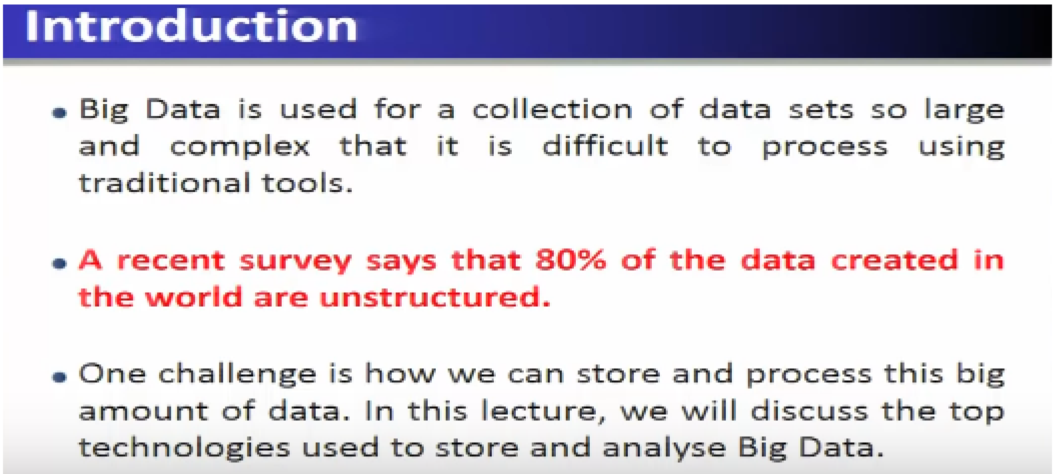
## Preface

**Content of this Lecture:**

- In this lecture, we will discuss a brief introduction to Big Data Enabling Technologies.

Preface Content of this lecture. In this lecture, we will discuss a brief introduction to big data enabling technologies. Here different icons are shown, which are different components, which we are going to discuss,

Refer Slide Time :( 0:33)



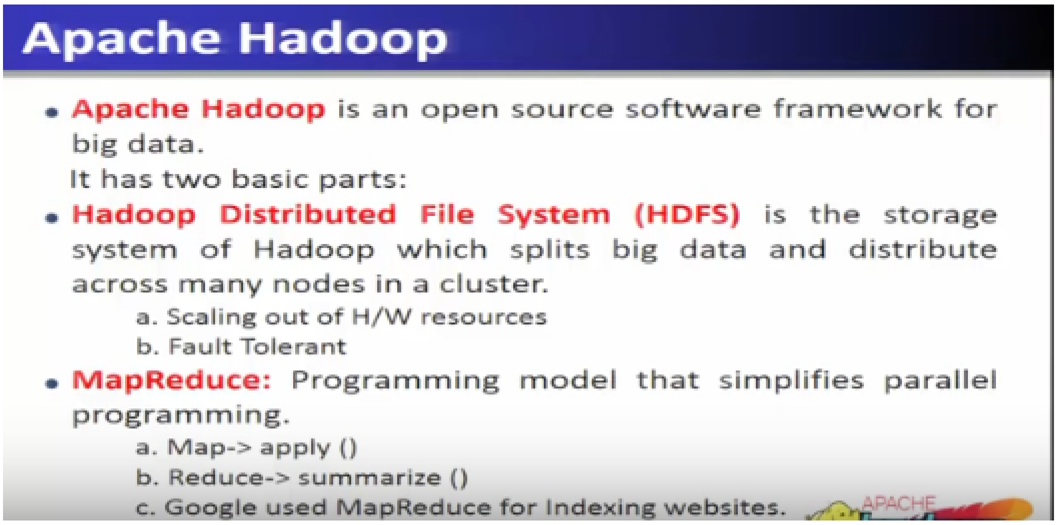
## Introduction

- Big Data is used for a collection of data sets so large and complex that it is difficult to process using traditional tools.
- A recent survey says that 80% of the data created in the world are unstructured.
- One challenge is how we can store and process this big amount of data. In this lecture, we will discuss the top technologies used to store and analyse Big Data.

today. Introduction; Big data is used, for a collection of data sets. So large and complex, that it is difficult to process, using traditional tools. A recent survey, says that 80% of the data created, in the world are, unstructured. Hence traditional tools, will not be able to handle, such a big data motion.

One challenge is, how can we stored and process this big data? In this lecture, we will discuss the technologies and the enabling framework, to process the big data.

Refer Slide Time :( 1:14)

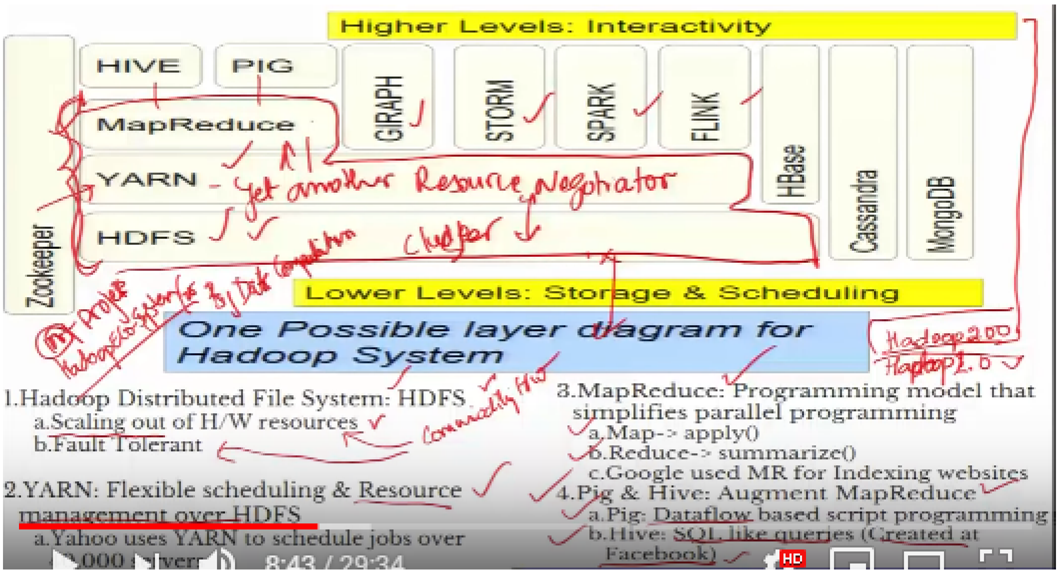


## Apache Hadoop

- Apache Hadoop is an open source software framework for big data.
  It has two basic parts:
- Hadoop Distributed File System (HDFS) is the storage system of Hadoop which splits big data and distribute across many nodes in a cluster.
  a. Scaling out of H/W resources
  b. Fault Tolerant
- MapReduce: Programming model that simplifies parallel programming.
  a. Map-> apply ()
  b. Reduce-> summarize ()
  c. Google used MapReduce for Indexing websites.

Apache Hadoop is the tool which is going to be used for, the big data computation. Apache Hadoop is an open source, software framework, for a big data. And, it has two parts, two basic parts. The first one is called, HDFS, Hadoop Distributed File System, the other is called, 'Programming Model, which is called a, 'Map Reduce'.

Refer Slide Time :( 1:39)



Higher Levels: Interactivity

Zookeeper

HIVE

PIG

MapReduce

YARN - Yet another Resource Negotiator

HDFS

cluster

GIRAPH

STORM

SPARK

FLINK

HBase

Cassandra

MongoDB

Lower Levels: Storage & Scheduling

One Possible layer diagram for Hadoop System

Project

Hadoop Ecosystem for Big Data Computation

Hadoop 2.0

Hadoop 1.0

1.Hadoop Distributed File System: HDFS
   a.Scaling out of H/W resources
   b.Fault Tolerant

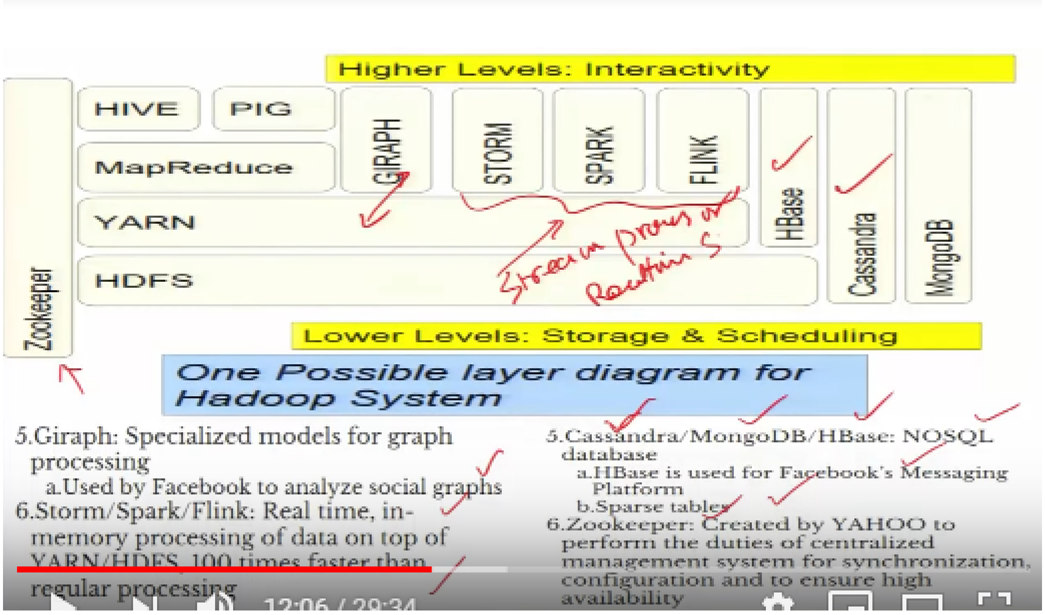
Commodity H/W

2.YARN: Flexible scheduling & Resource management over HDFS
   a.Yahoo uses YARN to schedule jobs over
   ?,000 server

3.MapReduce: Programming model that simplifies parallel programming
   a.Map-> apply()
   b.Reduce-> summarize()
   c.Google used MR for Indexing websites

4.Pig & Hive: Augment MapReduce
   a.Pig: Dataflow based script programming
   b.Hive: SQL like queries (Created at Facebook)

Refer Slide Time :( 8:45)



Higher Levels: Interactivity

HIVE

PIG

MapReduce

GIRAPH

STORM

SPARK

FLINK

HBase

Cassandra

MongoDB

YARN

HDFS

Zookeeper

Lower Levels: Storage & Scheduling

One Possible layer diagram for Hadoop System

5.Giraph: Specialized models for graph processing
  a.Used by Facebook to analyze social graphs
6.Storm/Spark/Flink: Real time, in-memory processing of data on top of YARN/HDFS. 100 times faster than regular processing

5.Cassandra/MongoDB/HBase: NOSQL database
  a.HBase is used for Facebook's Messaging Platform
  b.Sparse tables
6.Zookeeper: Created by YAHOO to perform the duties of centralized management system for synchronization, configuration and to ensure high availability

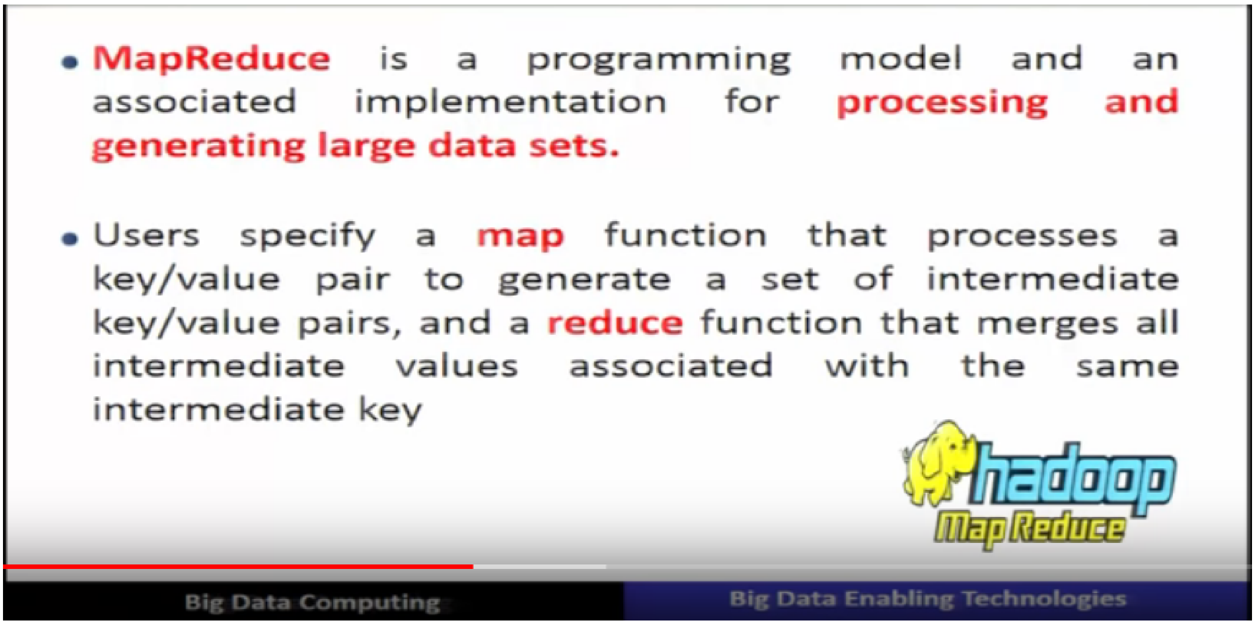
Now, you will be looking up the Hadoop Eco system, for big data computation. So, this is the Hadoop Eco system, for big data computation, which is summarised as follows. So, as I told you that Hadoop has two main components, that was, the old version of Hadoop, which is called a, 'Hadoop, Version 1, 1.0’.Now there was some disadvantage of Hadoop, Version1.0.So, newer version, which is called a, ‘Hadoop 2.0’, came into effect. So this particular picture is, about Hadoop 2.0 version, which includes, besides HDFS, and Map Reduce another version of, Yarn is being, added. So, Yarn is, a Resource Manager, are, for Hadoop. This is yet another Resource negotiator. This is also called, ‘Resource Manager’. For Hadoop which runs over HDFS. Now HDFS, is are distributed file system which is run over the cluster. So, below this HDFS, will have, the cluster infrastructure and over that, Hadoop distributed file system runs, which manages, all the notes, and their corresponding memories, using HDFS. And the Resources which are, required by the application, is being allocated using the Resources Manager, which is called a, Yarn’. And the programming framework for this, big data computation, is a Map Reduce. So, in Hadoop 2.0, there are three different mean components, Map Reduce, Yarn and HDFS which runs over the cluster, now with this introduction of a Yarn.

So, there are some applications, at as, Giraph, Storm, Spark, Flink, they are not going to use, Map Reduce directly, they run over Yarn and HDFS. Similarly, the applications which will simplify the use of, Map Reduce further on, called, Hive and Pig they are on over, Map Reduce. Earlier in Map Reduce, in Hadoop version 1.0, all the applications were running over Map Reduce. Now there is a choice that means Map Reduce, or a non-Map Reduce applications, can run with the help of, Yarn HDFS. So, now as if, HDFS, is it Hadoop 2.0, it is possible now, to have more flexible. So, there are various projects, which we are going to discuss, with the notion, with the introduction, of Yarn in

Hadoop 2.0, lot of new projects, are there. More than hundred projects are available in Hadoop eco system. And they're all open source that is free. These projects we will discuss some of them, which is useful, for big data computation. Let us start one by one, that is from HDFS. So HDFS stands for Hadoop Distributor File System. So, this particular file system runs over the cluster. So it basically, is based on scale out, of hardware Resources, that is the notes can add, can be added. Any number of notes can be added. And this particular style, is called, 'Scaling Out'. So it is not always necessary, that very powerful systems are required. So, the Resources which are required, here is the form of cluster, they are called, 'Commodity Hardware.' So these commodity hardwares are prone to failure, so this HDFS, provides lot of provisioning, for fault tolerance that we will discuss, when we go, in more detail, of HDFS. So, HDFS is a file system, over the cluster, and this is called a, 'Hadoop cluster'. Over HDFS, the Resources are being provided reducing Resource Manager, for the different application and this is called a, 'Yarn', Yarn runs over HDFS. So, the full form Yarn is yet another Resource Manager. It is flexible scheduling and Resource management, over HDFS. Now on, over Yarn, there is programming model for big data computations and this will simplify the, parallel programming notion, using to function, this is called a, 'Map and Reduce'. So, Google earlier uses, Map Reduce for indexing and they are so powerful that, almost all big data applications, can be programmed using this, for align Map Reduce that we will see, in this part of the course, in more details. Now, over Map Reduce, there are two tools, which are available, one is called, 'Pig and Hive'. So, hive, is created at Facebook. There all SQL like queries, which runs over, Map Reduce. So, complex programming of Map Reduce can be avoided, using Hive so, Hive will simplify, the programming, over Map Reduce, for big data computation. So, Hive provides, SQL live queries, so it simplifies the entire programming notion and it was developed or it was created at the Facebook. Now Facebook also uses another tool, which is called the, 'Pig'. Which is, the scripting based or a dataflow based, script programming, runs over the Map Reduce. So, both of this Pig and Hive will augment, the Map Reduce. So, the complex programming of Map Reduce can be simplified by, using these tools, Pig and Hive.

Now, has you know that, within non-Map Reduce application, such as Giraph, Giraph is a graph processing tool, which is, being used by the Facebook, to analyse the social network's graph that was made simplified, when it was made out of Map Reduce. So, it uses Yarn and HDFS and this is non-Map Reduce application, for, computation or computing large graphs, of the social network. So, Giraph is the tool which is now, runs over, Yarn HDFS, and this is used, the big graphs computations that we will see, later on, this part of the course. The next tool which is, there is called, 'Storm, Spark and Flink'. They are going to deal with the real time, in memory processing, of the data or Yarn an HDFS. So the fast data are Streaming data applications, either can we do using; a Storm, Spark and Flink and they basically, are in memory computation, which are faster than regular computation. So, Stream processing, or a real time or the real time, Streaming applications are done using, Star, Spark and Flink or Yarn and HDFS. Now, then, we are going see, the storage part, the big data, how it can be stored. Now most of these big data is stored in the form of a key value pair and they are also, known as, No Sequel Data Store. So, this No Sequel Data Store can be supported by, the data base like, Cassandra, MongoDB and HBase. So, these particular HBase, is over HDFS, Cassandra, MongoDB and they're also, using this kind of for, for, for storing the big data applications. So, HBase was, initially using, used by the Face book for its messaging platform, later on, Face book also used, developed Cassandra, for the key value store purposes. Now, as far as these diffident components, of Hadoop ecosystem and they are also called as, 'Open Source Projects', which is used for, the big data computation. Now, another part is, called the, 'Zookeeper'. Why? Because the names are of, some animals, so the Zookeeper is coordination service, which is useful for all these different projects. So Zookeeper is created, at, by the Yahoo to form, the centralized, management, for synchronization, configuration, to ensure the high availability, that well be see. So this are diffident projects,  of this, open source framework, for big data computation and this is called the Hadoop ecosystem. And, we will be, discussing and using in this part course, in upgrade detail of it.

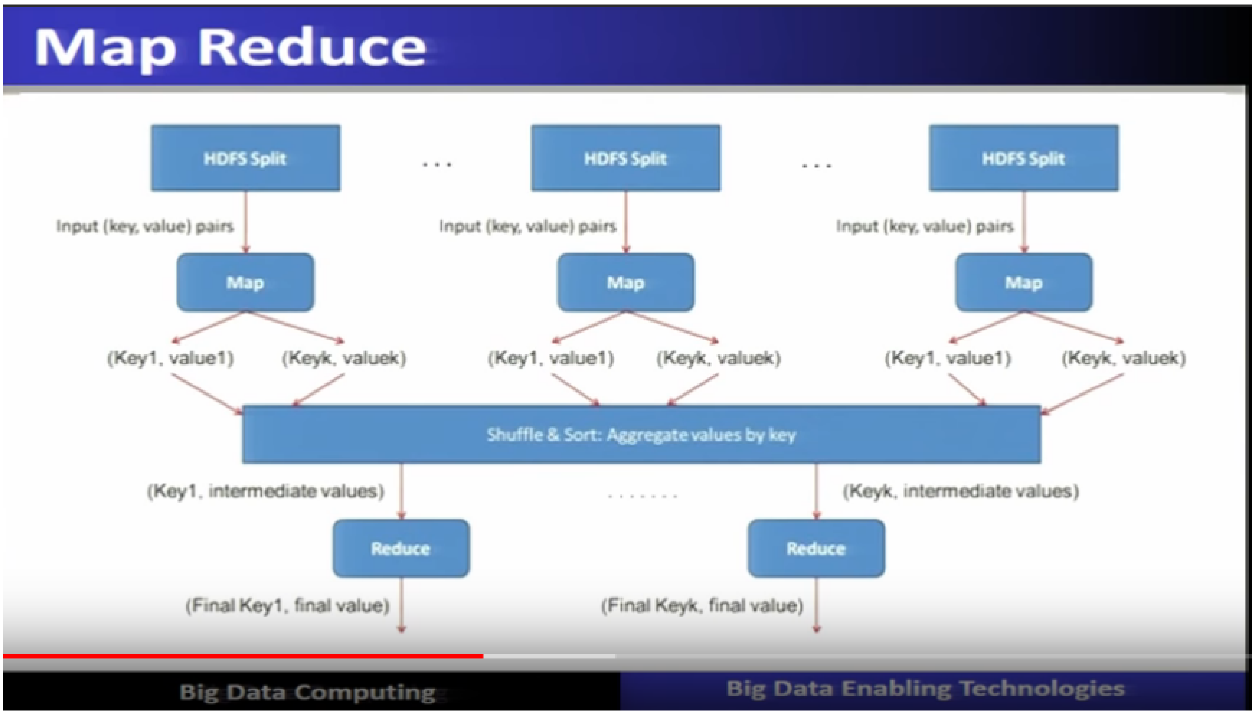
Refer slide time :( 12:07)



- MapReduce is a programming model and an associated implementation for processing and generating large data sets.
- Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key

So, now let as see about the map reduce, which is programming model for, big data component. So it is able to process, the large size data sets, using two functions, which are called, 'Map and reduce'.

Refer slide time :( 12:24)



## Map Reduce

HDFS Split . . . HDFS Split . . . HDFS Split

Input (key, value) pairs

Map

(Key1, value1) (Keyk, valuek)

Shuffle & Sort: Aggregate values by key

(Key1, intermediate values) ....... (Keyk, intermediate values)

Reduce

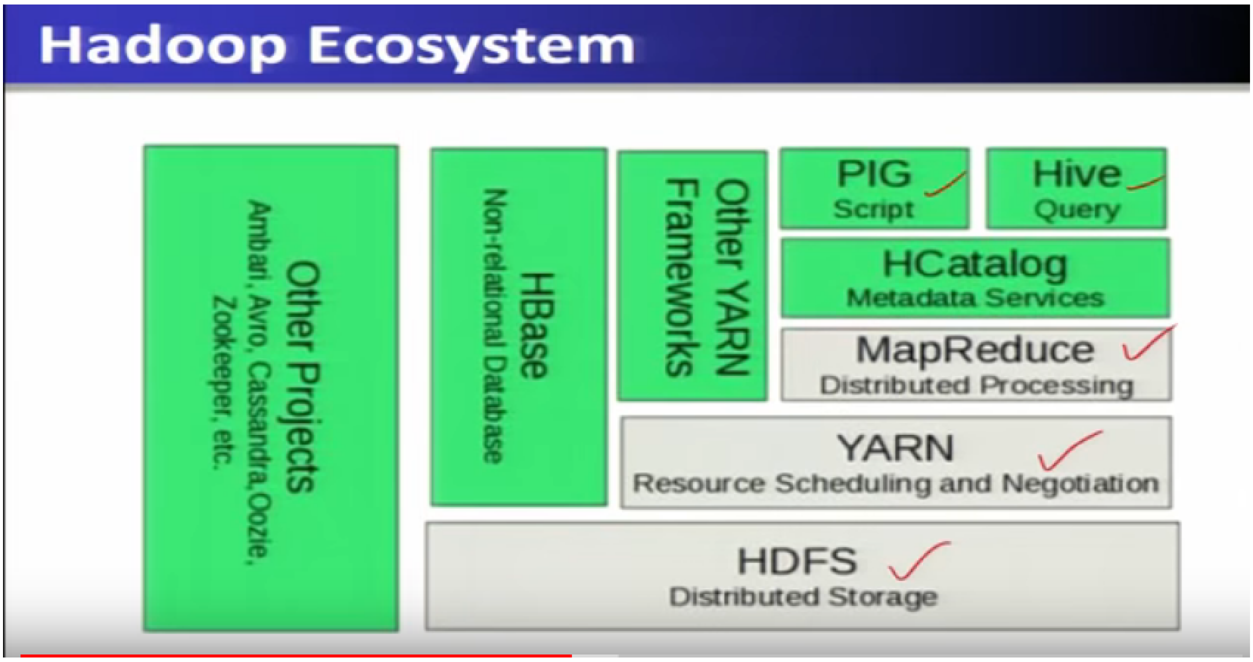
(Final Key1, final value) (Final Keyk, final value)

Big Data Computing

Big Data Enabling Technologies

Which user can specify. And, this runs over, HDFS, in version 1.0 and in version 2.0, using, yarn it can negotiate for the, Resources, from HDFS, and run its application.

Refer slide time :( 12:41)



## Hadoop Ecosystem

Other Projects
Ambari, Avro, Cassandra, Oozie, Zookeeper, etc.

HBase
Non-relational Database

Other YARN Frameworks

PIG
Script

Hive
Query

HCatalog
Metadata Services

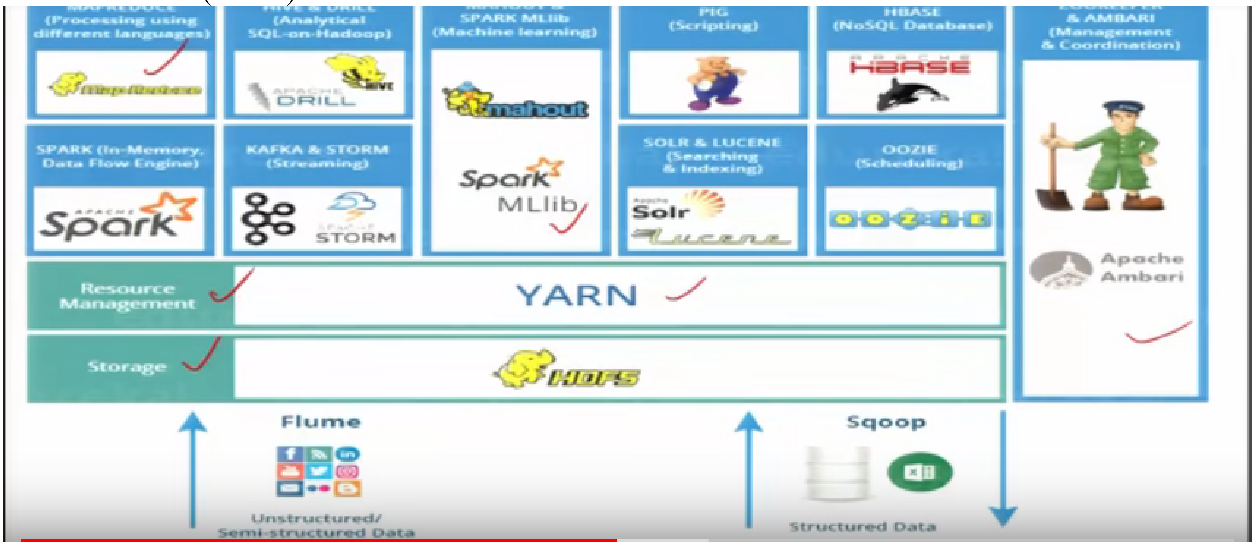
MapReduce
Distributed Processing

YARN
Resource Scheduling and Negotiation

HDFS
Distributed Storage

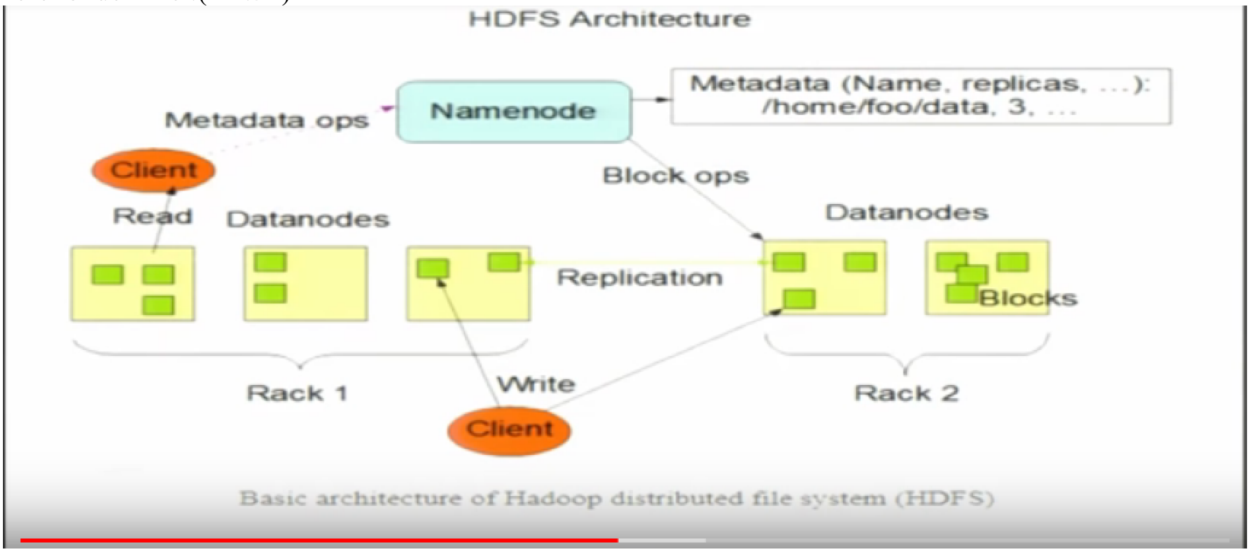
So, this Hadoop ecosystem, we can again see, in this picture, in a great detail. So, HDFS will provide and distributed storage. Yarn will provide Resource Manager for Hadoop and for distributed processing, that is a programming for big data, is done through the, the map reduce. And Pig and Hive, is basically, these tools which will simplify the programming of the map reduce. And the various others, projects which we have seen, which basically use, HDFS, yarn or map reduce.

Refer slide time :( 13:23)


MAPREDUCE (Processing using different languages)
MapReduce
HIVE & DRILL (Analytical SQL-on-Hadoop)
APACHE DRILL
HIVE
MAHOUT & SPARK MLlib (Machine learning)
mahout
Spark MLlib
PIG (Scripting)
HBASE (NoSQL Database)
APACHE HBASE
ZOOKEEPER & AMBARI (Management & Coordination)
SPARK (In-Memory, Data Flow Engine)
APACHE Spark
KAFKA & STORM (Streaming)
STORM
SOLR & LUCENE (Searching & Indexing)
Solr
Lucene
OOZIE (Scheduling)
OOZIE
Apache Ambari
Resource Management
YARN
Storage
HDFS
Flume
Unstructured/ Semi-structured Data
Sqoop
Structured Data

So this particular, entire Hadoop ecosystem is shown in this, simplified picture, that, HDFS provides storage for big data. The Resources like, CPU, Memory and other recourses, are being allocated, using on managed by the yarn, for diffident applications like, map reduce. Similarly, they are diffident, other projects, such as, in memory computation that's called, a 'Purchase Spark. And machine learning over this Spark, called park MLlib and this particular project, which is called, 'Streaming applications', over yarn and HDFS, that is called, 'Kafka and Storm.' And the Zookeeper is a centralized service. So, this particular, entire different project will form the Hadoop Ecosystem.

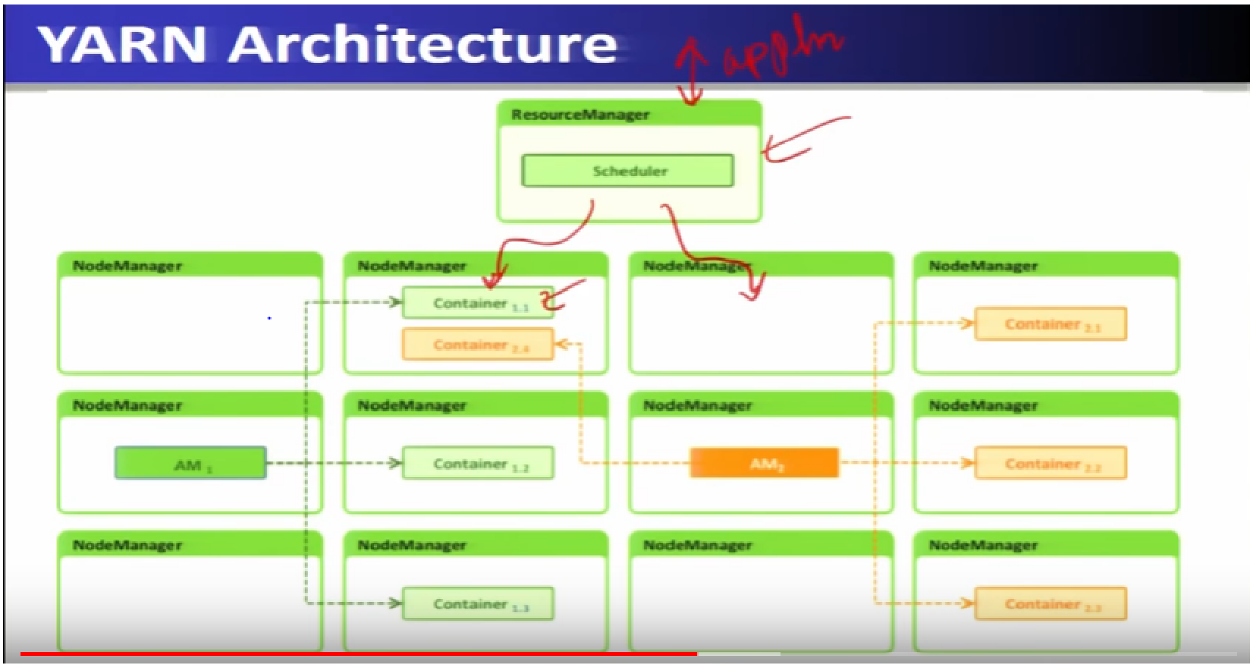
Refer slide time :( 12:32)


HDFS Architecture

Basic architecture of Hadoop distributed file system (HDFS)

And more than hundred such projects, are available free, that's in open source projects, which is use for big data computation.

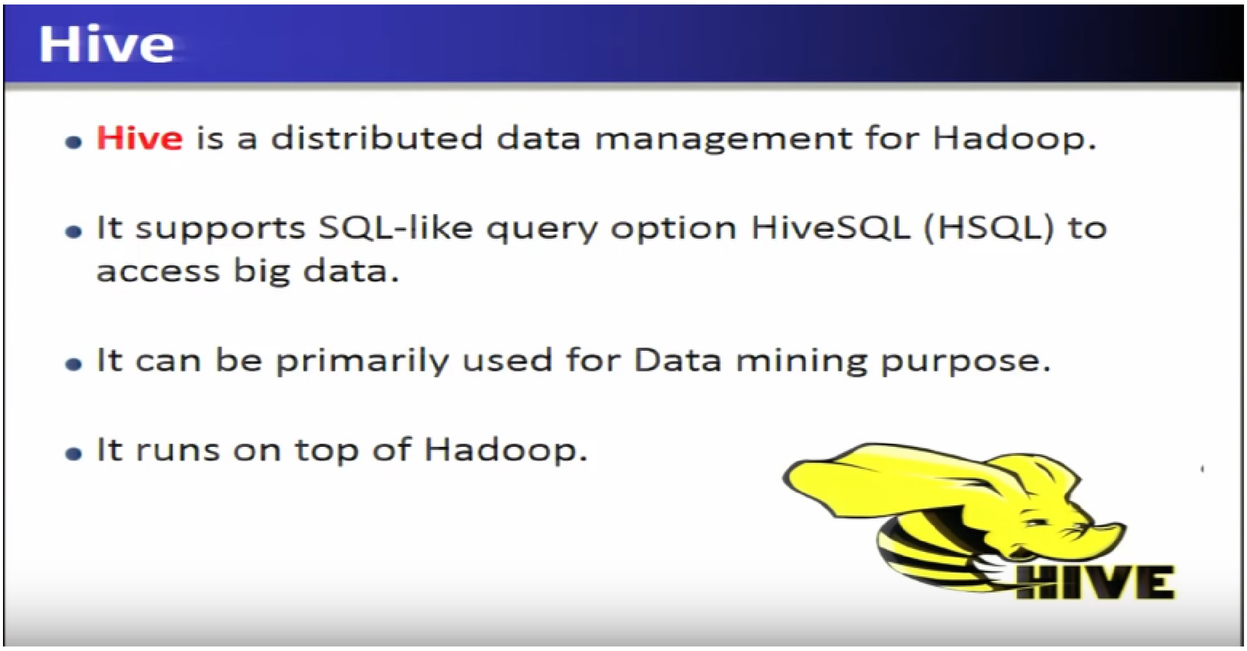
Refer slide time :( 14:40)



# YARN Architecture

That, we will see more details, so, this particular a yarn, Resource Manager, which is yet another Resource negotiator, which is a Resource Manager, for the Hadoop ecosystem, for Hadoop system, is responsible for allocating, the Resources, to the various applications, running in a Hadoop cluster and scheduling task, to be executed, on different cluster Nodes. And this open source Hadoop distributed, framework. It has two parts. One is called, 'Resource manger', and this is Resource Manager, it has scheduler within it. So, this particular, here the applications can interact with this Resource Manager and can demand for different Resources, through yarn. This particular Resource Manager, in turn, knows or allocates the Resources, which are managed, at the Node level, by the Node Manager. And, the Resources are nothing but, the container, in the form of a container. So, whenever Resources is, means required, by the application, it has go through, the Resource Manager, Resource Manager knows and allocates, the Resources to the application, with the help of Node Manager and the Node Manager, Resources, allocates the Resources, in the form the container. So container is the basic resources, which are allocated, to different applications, which are managed internally, by every Node. So every Node is having a node Manager and these particular resources are allocated and reclaim, with help of, a centralized Resource Manager in yarn.

Refer slide time :( 16:26)

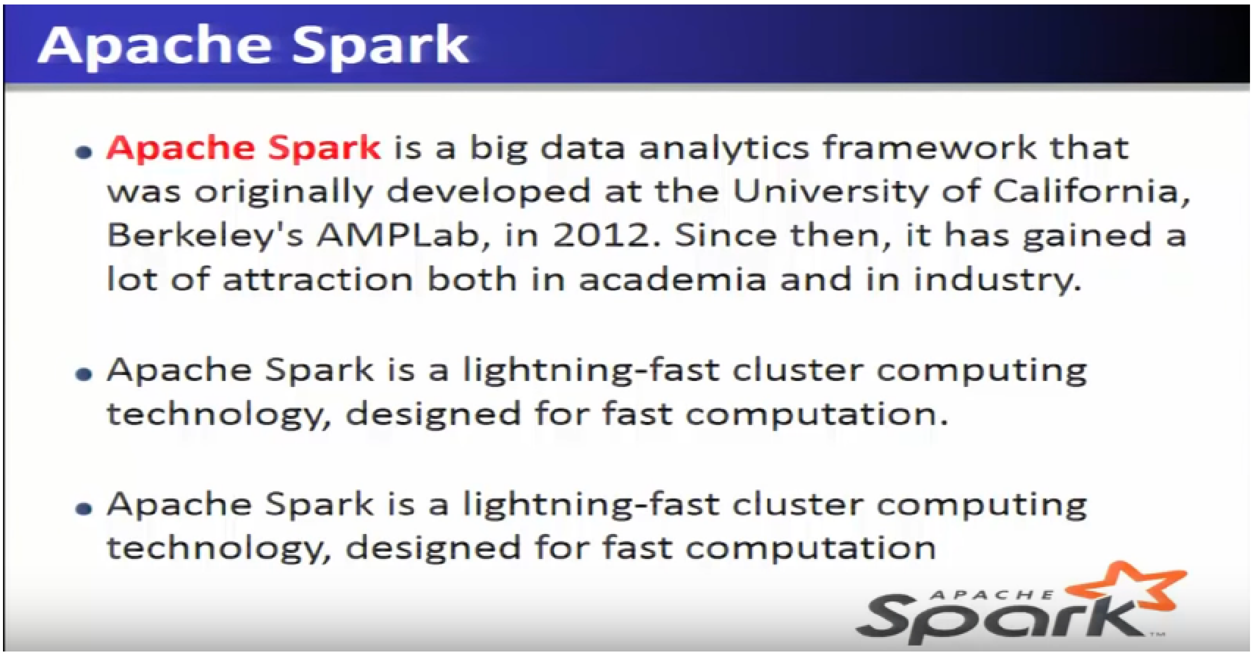


## Hive

- Hive is a distributed data management for Hadoop.
- It supports SQL-like query option HiveSQL (HSQL) to access big data.
- It can be primarily used for Data mining purpose.
- It runs on top of Hadoop.

And Hive, is a distributed data management, for Hadoop, and it supports, SQL like query, option that is called, 'Hive SQL', to access big data and it is primarily used, for data mining purpose.

Refer slide time :( 16:26)

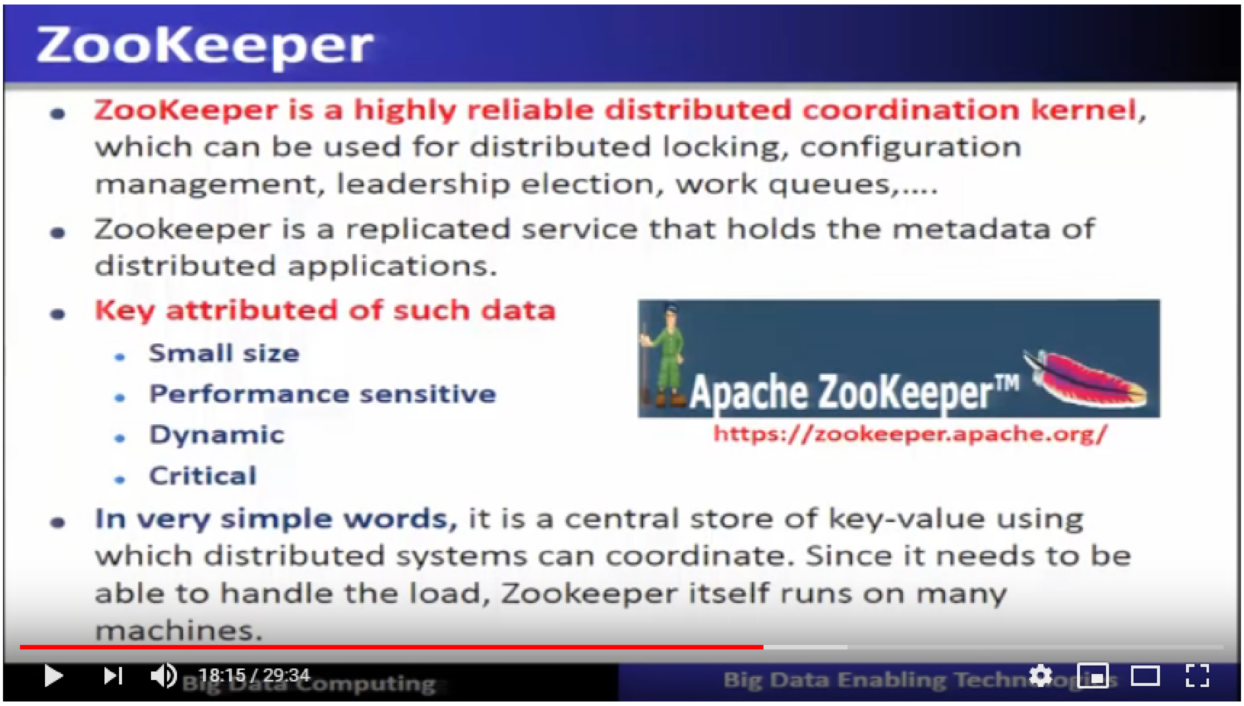


## Apache Spark

- Apache Spark is a big data analytics framework that was originally developed at the University of California, Berkeley's AMPLab, in 2012. Since then, it has gained a lot of attraction both in academia and in industry.
- Apache Spark is a lightning-fast cluster computing technology, designed for fast computation.
- Apache Spark is a lightning-fast cluster computing technology, designed for fast computation

Now we will see the Apache Spark. Apache Spark project, runs over, HDFS and this is a big data analytics frame work, which in memory computation.  So that the, lightning fast cluster computation, is being performed. So several applications like, Stream processing, Machine learning and Large Scale Graph Processing, are already implemented, over this Spark

Refer slide time :( 17:20)

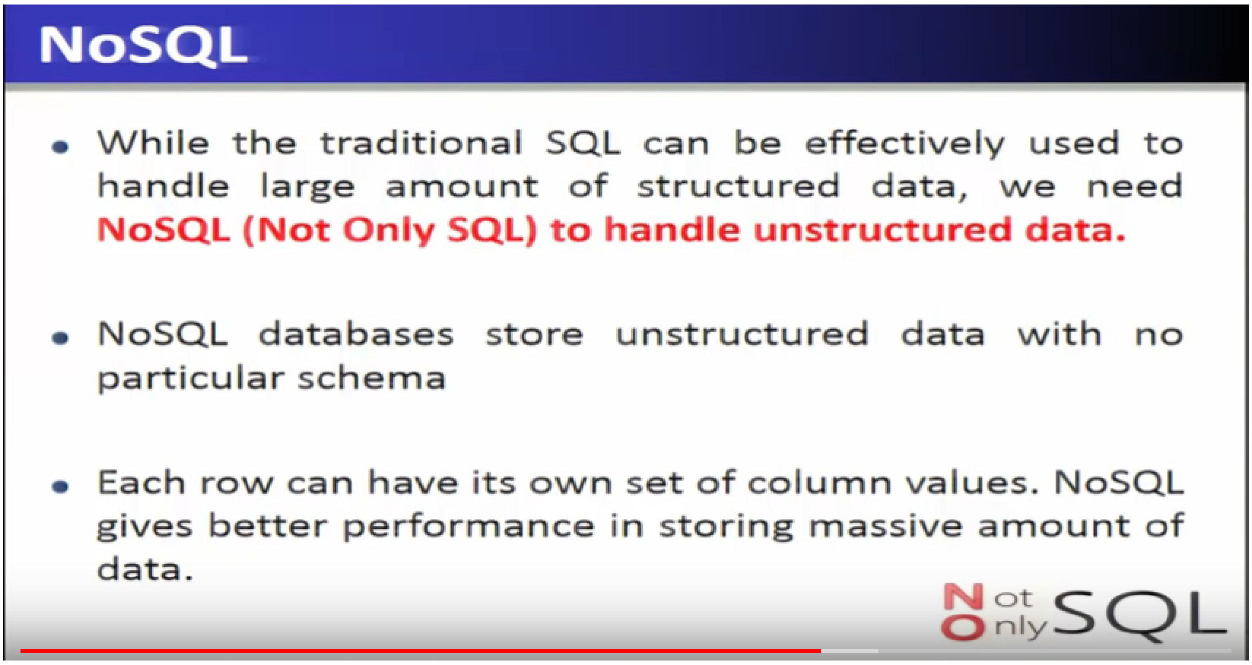


## ZooKeeper

- ZooKeeper is a highly reliable distributed coordination kernel, which can be used for distributed locking, configuration management, leadership election, work queues,....
- Zookeeper is a replicated service that holds the metadata of distributed applications.
- Key attributed of such data
  - Small size
  - Performance sensitive
  - Dynamic
  - Critical
- In very simple words, it is a central store of key-value using which distributed systems can coordinate. Since it needs to be able to handle the load, Zookeeper itself runs on many machines.

Core. And we will see that these, different projects which are more than hundred plus, to manage them, there is a centralized service, which called, 'The Zookeeper'. So Zookeeper is a highly reliable, distributed, coordination service, and which can be used for, locking, configuration management, either election, and so on. So Zookeeper is a replicated service and it has the attributes, key attributes, such as, Small size, Performance Sensitive, Critical. And it is a part of the Apache Zookeeper, Open source, project. And in simple words, it a central store for, the, the key value pair, using distributed system, which can coordinate, since it needs to be, able to handle the load. So the Zookeeper runs itself, on many machines.

Refer slide time :( 18:17)

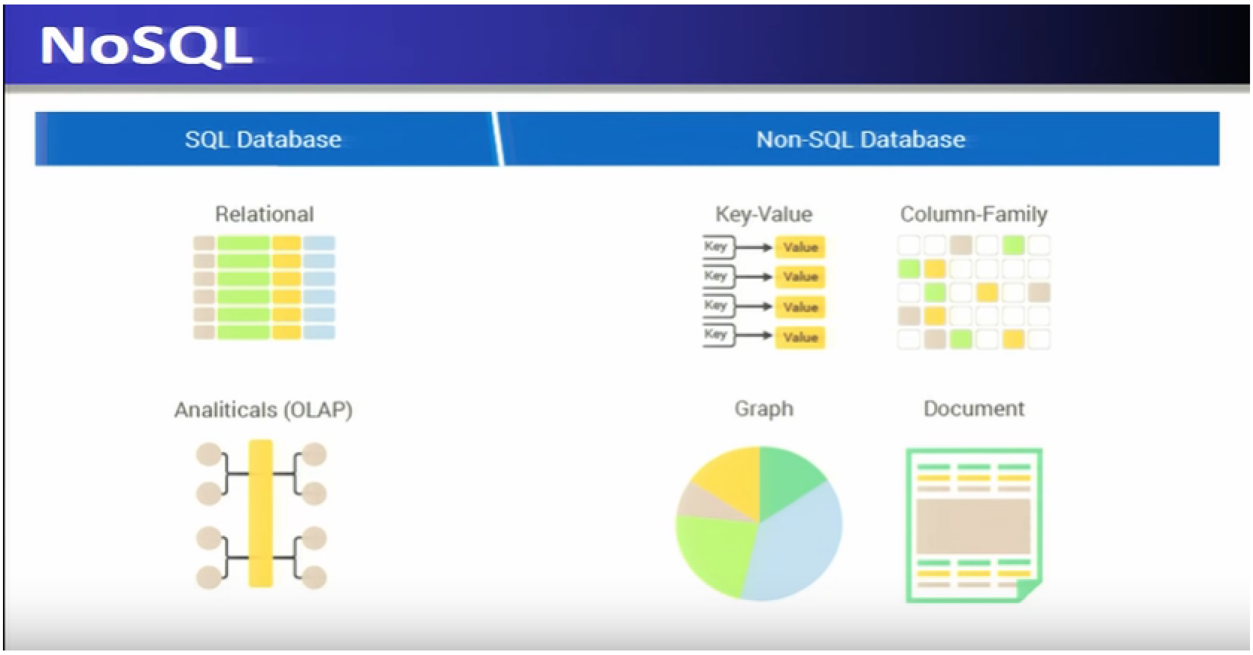


## NoSQL

- While the traditional SQL can be effectively used to handle large amount of structured data, we need **NoSQL (Not Only SQL) to handle unstructured data.**
- NoSQL databases store unstructured data with no particular schema
- Each row can have its own set of column values. NoSQL gives better performance in storing massive amount of data.

So, that is the internal working of the Zookeeper, but the primarily Zookeeper, is used to, make, make the coordination and configuration service of all, the projects, which are running, which are executing the big data computation, using different projects. So, that is why, this, the name Zookeeper is being made. Now NoSQL, we see that, that traditional SQL, can be effectively used to handle the large amount of, structured data. But here in the big data, most of the information is, unstructured form of the data, so basically, noSQL that is, is required to handle that information, because, traditional SQL required, by the, the data to be, in the structured, data format. So NoSQL data base is, stored unstructured data also, however, it is not, enforced to follow a particular, fixed schema structure and schema keeps on, changing, dynamically. So, each row can have its own set of column values. NoSQL gives a better performance, in storing the massive amount of data compared to the SQL, structure.

Refer slide time :( 19:39)



## NoSQL

SQL Database | Non-SQL Database

Relational

Analiticals (OLAP)

Key-Value

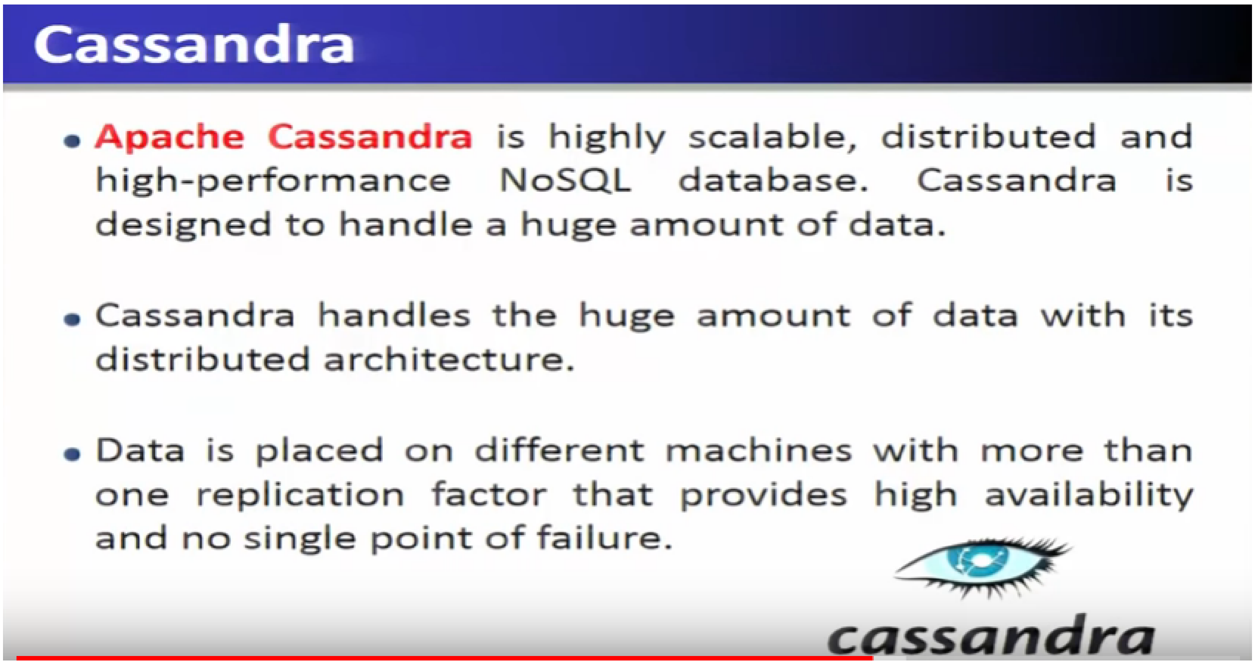
Column-Family

Graph

Document

So, here you can see that, NoSQL database is primarily a key value store. It is also called a, 'Column Family', and it is different and the relational database management system, which is in the form of, tables and this, a Column family. So, column vise, the data is stored, in the form of a key value, pairs. So, we will see, more detail, about NoSQL databases. How they are, able to store the big data, in further, slides?

Refer slide time :( 20:15)

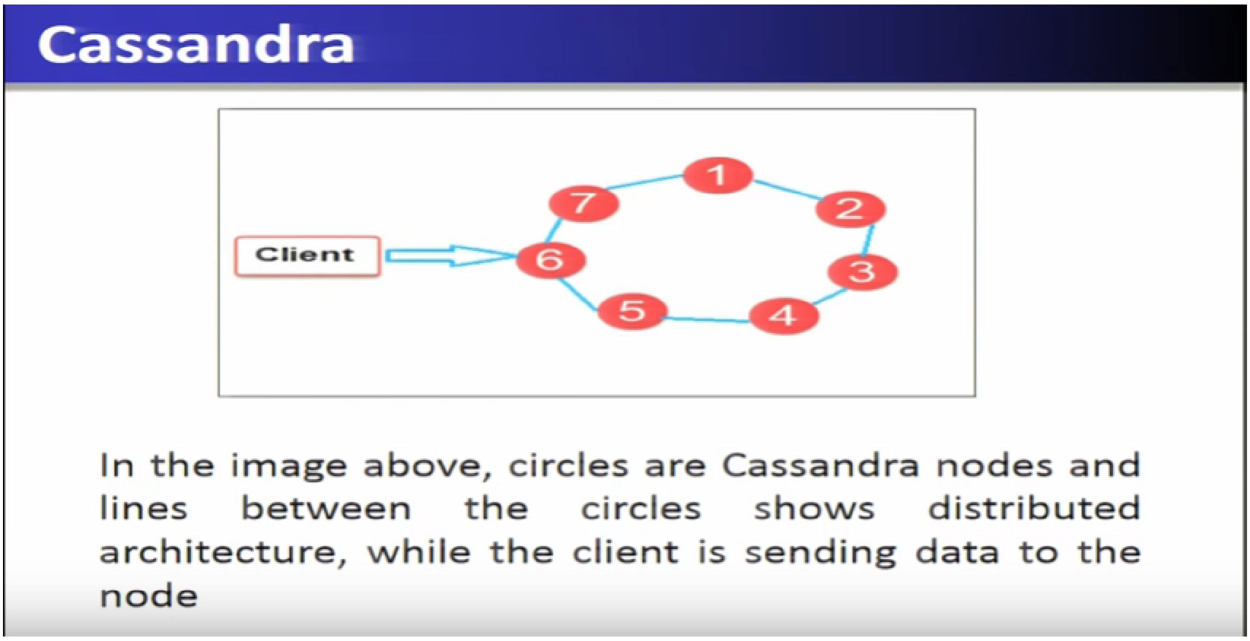


## Cassandra

- Apache Cassandra is highly scalable, distributed and high-performance NoSQL database. Cassandra is designed to handle a huge amount of data.
- Cassandra handles the huge amount of data with its distributed architecture.
- Data is placed on different machines with more than one replication factor that provides high availability and no single point of failure.

Now, another data base, which supports, this particular data model, that is, NoSQL data model, is called, 'Cassandra'. So Apache Cassandra is highly scalable, distributed and high-performance, NoSQL database. So Cassandra is designed, to handle the huge amount of information and the Cassandra handles, this huge data, with its distributed architecture, that we will discuss, in this part of the course.

Refer slide time :( 20:47)

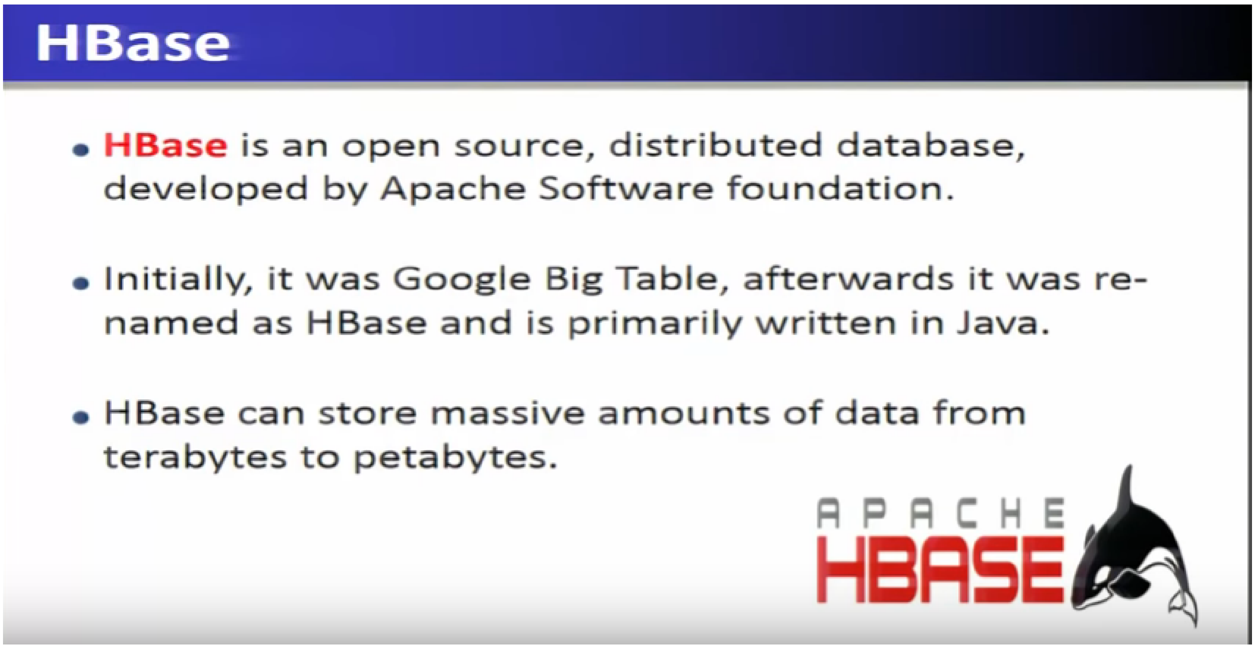


## Cassandra

In the image above, circles are Cassandra nodes and lines between the circles shows distributed architecture, while the client is sending data to the node

These particular notes, shown over here, that is, the notes are, I mean, they presented, in the form of a, in the ring. So, all the notes are you see that, they are, placed around circle, that is called a, 'Ring'. And, this particular ring, is the architecture, how, this particular notes are, basically the data centre notes are, organized, virtually in the Cassandra architecture.

Refer slide time :( 21:18)



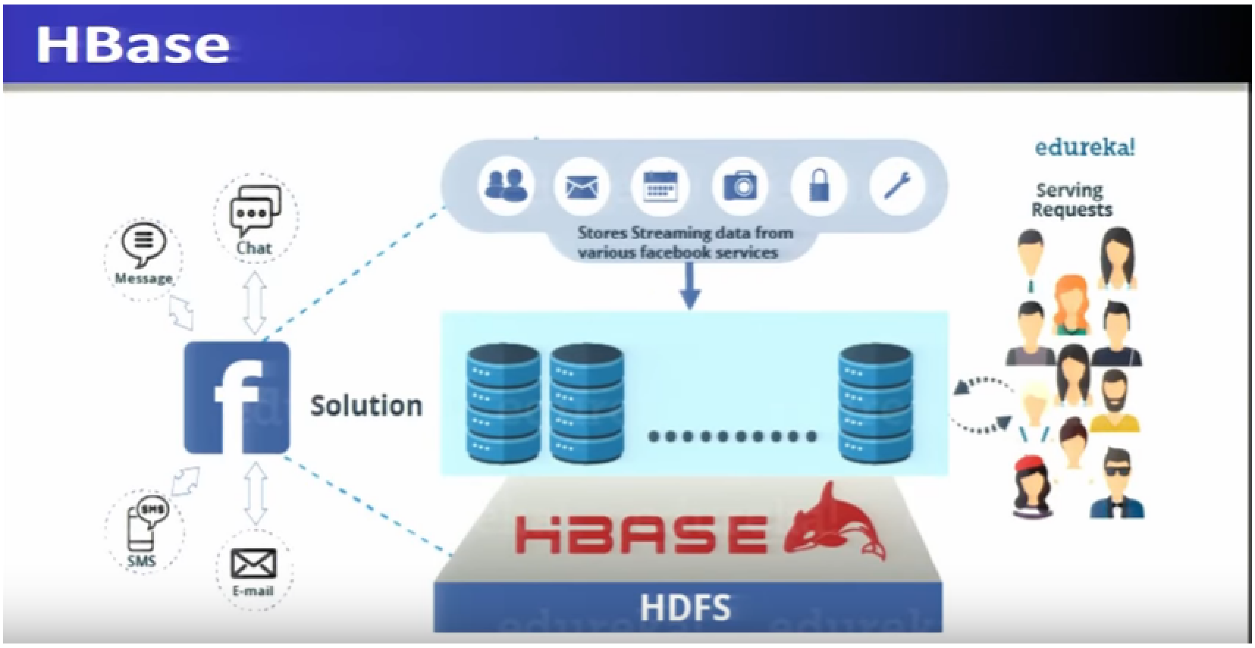
## HBase

- HBase is an open source, distributed database, developed by Apache Software foundation.
- Initially, it was Google Big Table, afterwards it was re-named as HBase and is primarily written in Java.
- HBase can store massive amounts of data from terabytes to petabytes.

APACHE
HBASE

We will see these things, in more detail. So, HBase is an open source, distributed databases, which is developed by, Apache Software foundation. Initially, it was the Google Big Table and afterwards, it was re-named as, HBase and is primarily, written with high level program language. HBase can store massive amount of data from, terabytes to petabytes.

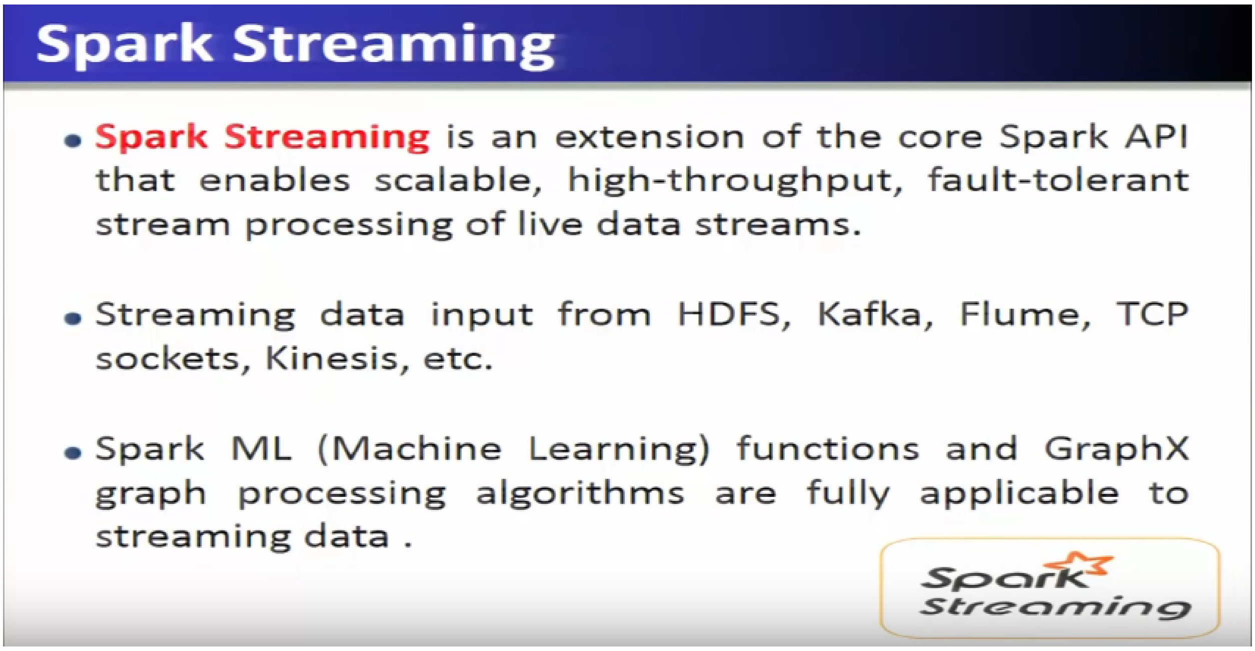
Refer slide time :( 21:44)



HBase

So this is, an HBASE wish runs over HDFS. And

Refer slide time :( 21:52)



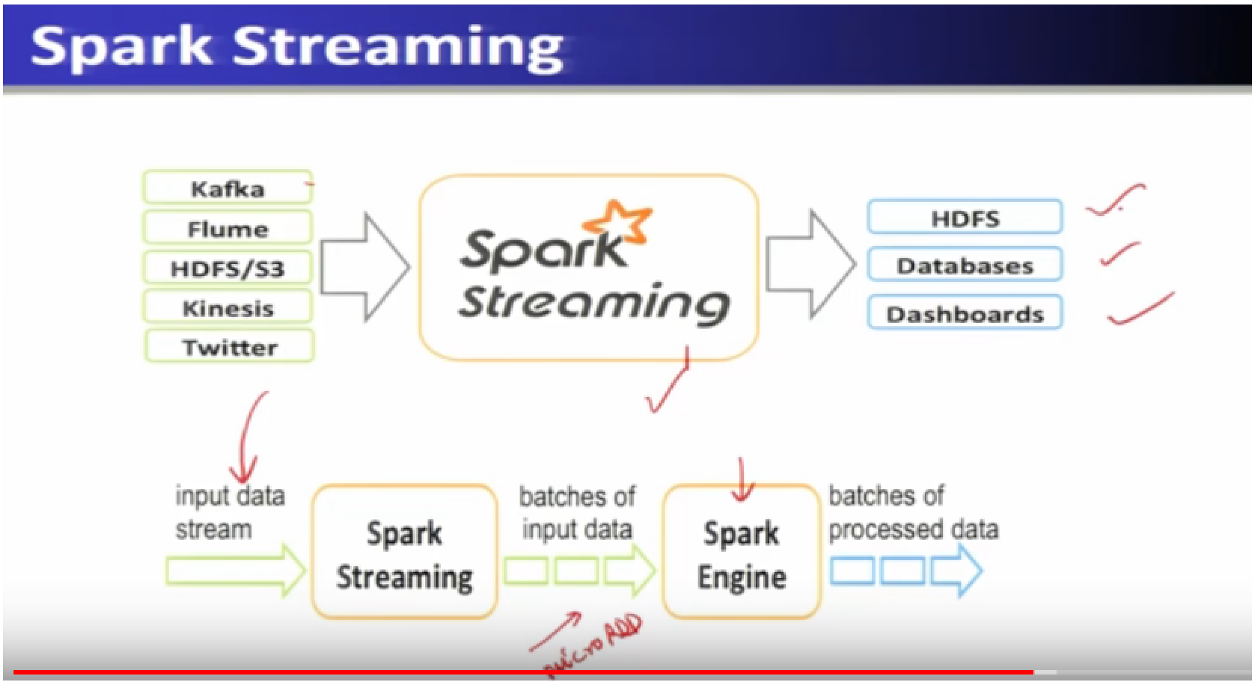
## Spark Streaming

- **Spark Streaming** is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.
- Streaming data input from HDFS, Kafka, Flume, TCP sockets, Kinesis, etc.
- Spark ML (Machine Learning) functions and GraphX graph processing algorithms are fully applicable to streaming data .

Spark Streaming

now, another project is called, ‘Spark Streaming’. So, Spark Streaming is used for real time stream processing applications and this also is called a, Fast Data Computations. So, stream processing, Spark Streaming, is an extension of the Spark core, that enables scalable, flow port, high fault tolerance, stream processing, of live, data streams. Streaming data, input from HDFS, Kafka, Flume, TCP, etc., is taken up, in this particular system, for computation and is Spark MLlib functions and graph, graphx, are fully compatible to the Streaming data, for computation.

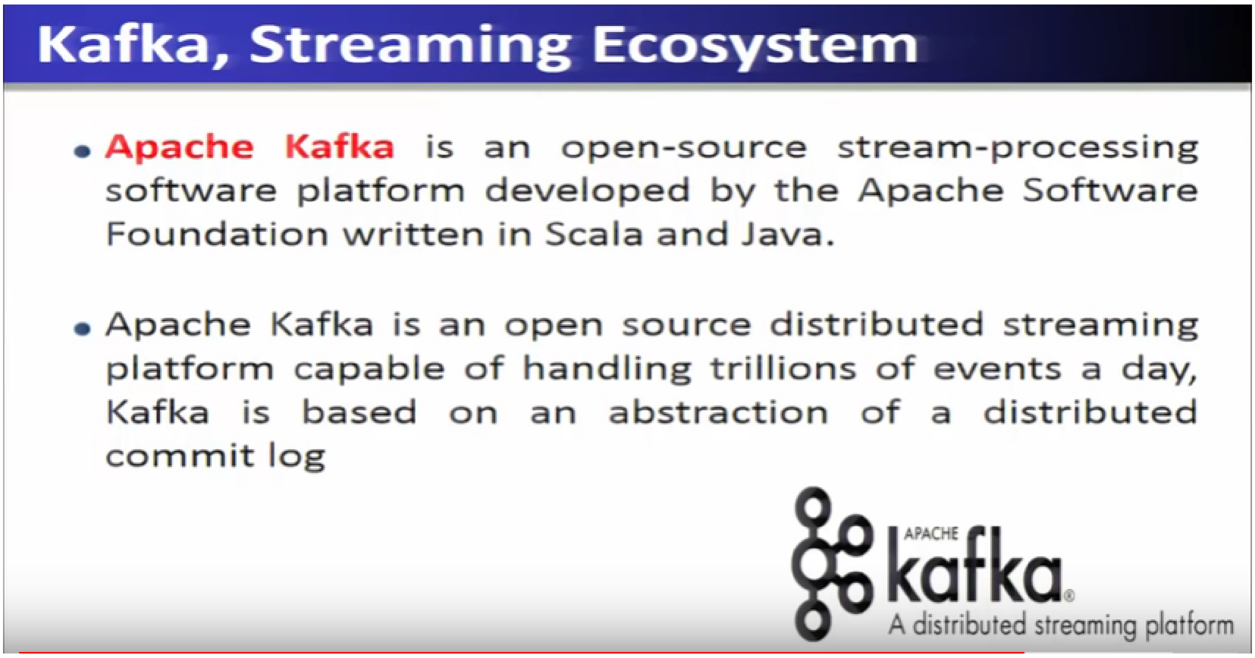
Refer slide time :( 22:44)



Spark Streaming

This particular diagram explains the entire computation of, the stream processing. So at the core you can see there are is a Spark Streaming and the data. Which is, in the form of the stream, enters, into the Spark Streaming system, using, the data will captured, either using Kafka, Flume, HDFS, Kinesis and Twitter. And Spark Streaming, will, do the computation, on the streams of, which enters into the Spark core. And after the computation, the data will be, output will be stored, in HDFS or in databases or basically the visual analysis, can be shown in the dashboard. So, for example, if whenever, stream, for example,  here, the Twitter Stream, or Kinesis or HDFS or Flume, Kafka, is right in, input in the form of a Stream, through the Spark Streaming, it will be divided into the form of micro batches, this is micro RDD, that we will explain later on. And these micro batches, will be processed, into the Spark engine. And, and then, this processed values are being, generated output, either in the form of, either will be stored in HDFS, databases or dashboard.  That we have, that we will see, in more detail later on.

Refer slide time :( 24:13)



## Kafka, Streaming Ecosystem

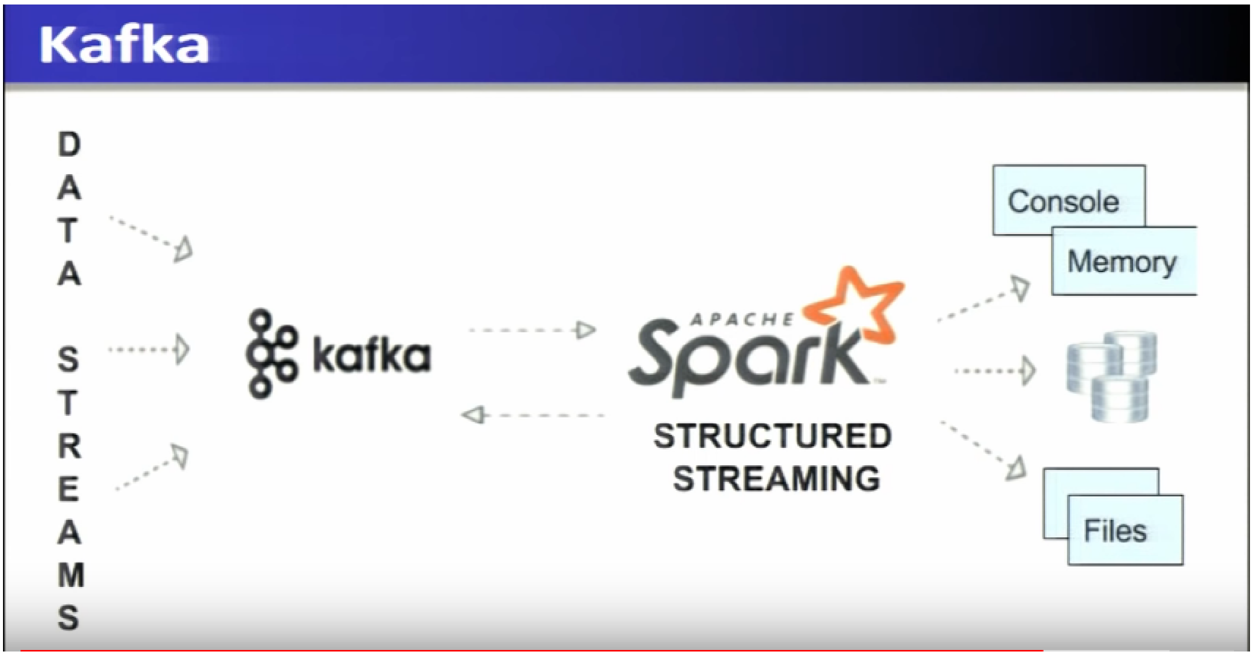
- Apache Kafka is an open-source stream-processing software platform developed by the Apache Software Foundation written in Scala and Java.
- Apache Kafka is an open source distributed streaming platform capable of handling trillions of events a day, Kafka is based on an abstraction of a distributed commit log

APACHE kafka

A distributed streaming platform

So, the data capturing method for, Streaming is, one of such method is called, ‘Apache Kafka’, is an open source, distributed stream processing, software farm work and this we will discuss later on.

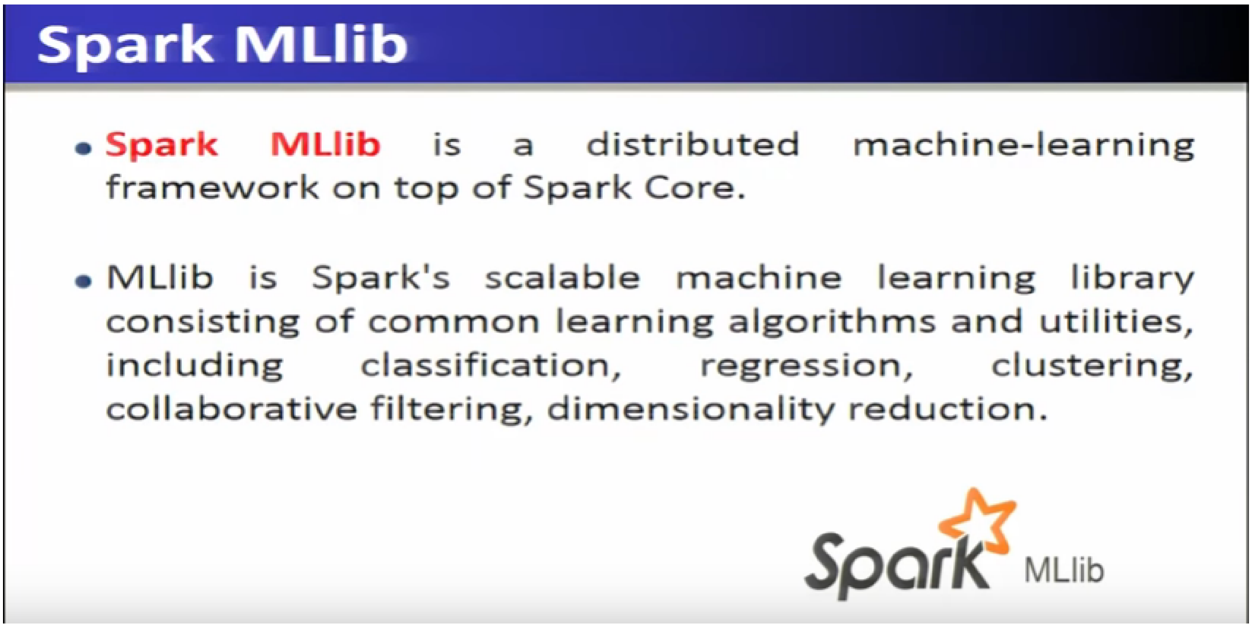
Refer slide time :( 24:31)



## Kafka

So, through Kafka data streams can be, submitted to the Apache Spark, for doing the computations. So this will form a pipeline. So, we will see, how this stream processing, data pipeline, we can form, using these different projects, which are open source frame work available, Apache Open Source frame work.

Refer slide time :( 25:03)



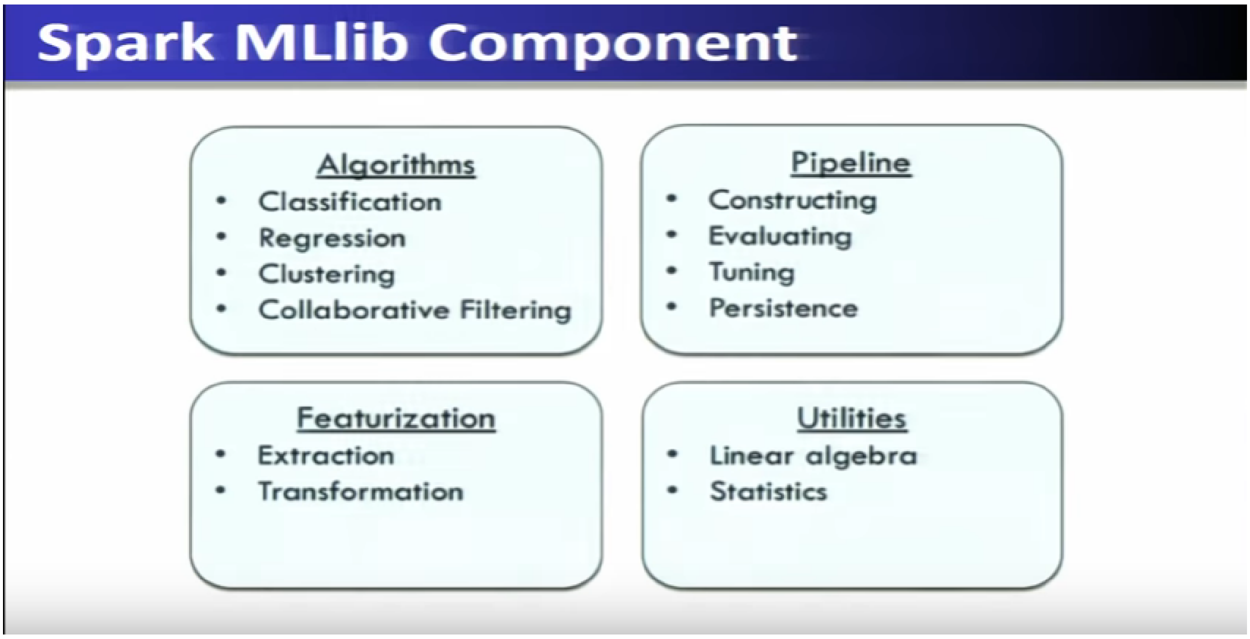
## Spark MLlib

- Spark MLlib is a distributed machine-learning framework on top of Spark Core.
- MLlib is Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction.

Spark MLlib

The next project is called, ‘Spark MLlib’. Spark MLlib is a distributed machine learning frame work, on top of Spark core. So, MLlib is the Spark’s scalable, machine learning library, which consists of common, machine learning algorithm and utilities. Such as, the Classification algorithm, Regression algorithm, Clustering, Collaborative, filtering, and Dimensionality reduction and all the algorithms, which are there in machine learning, they are a implemented in this particular frame work. And that is why they are called, ‘Scalable machine learning and it is there available in the form of libraries that we will see.

Refer slide time :( 25:33)



## Spark MLlib Component

**Algorithms**
- Classification
- Regression
- Clustering
- Collaborative Filtering

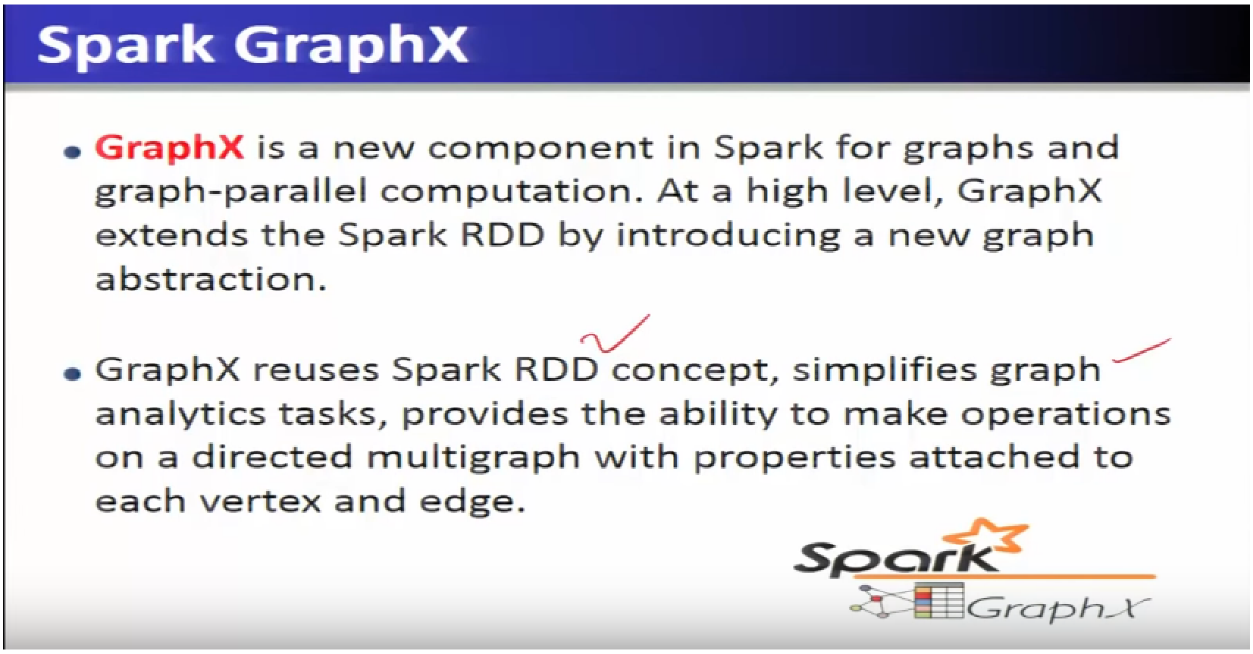
**Pipeline**
- Constructing
- Evaluating
- Tuning
- Persistence

**Featurization**
- Extraction
- Transformation

**Utilities**
- Linear algebra
- Statistics

So, the algorithms which are available are shown over here. And we will see in more detail, how we will we are going to use them, in this big data, I mean, Ecosystem.

Refer slide time :( 25:49)



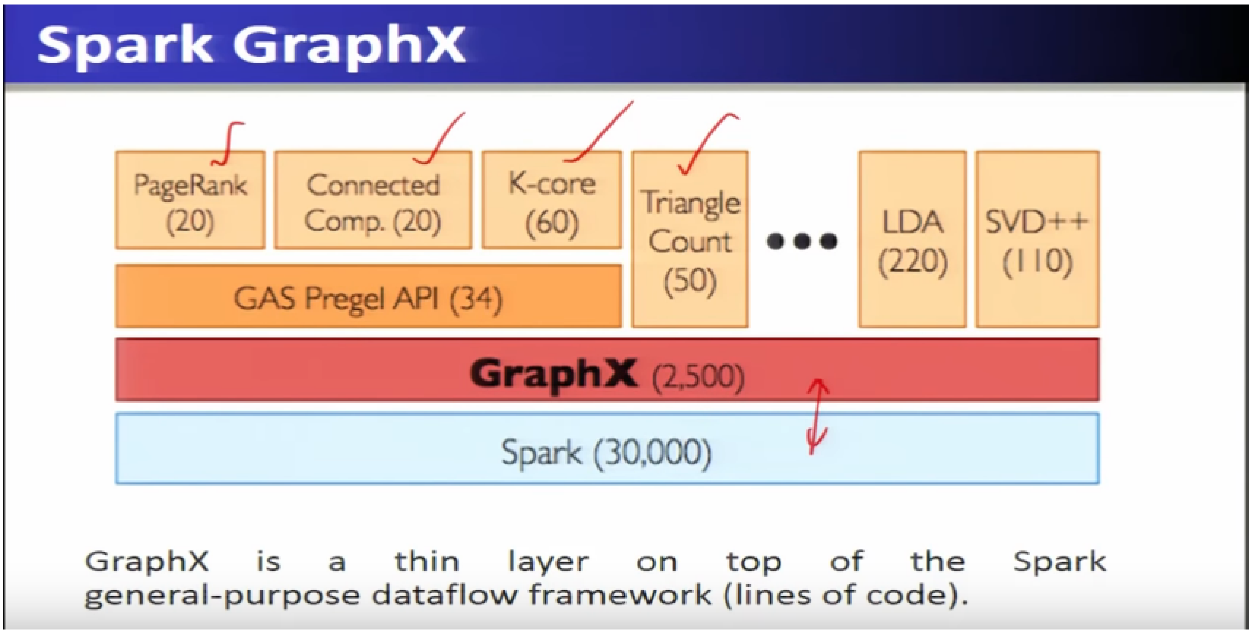
## Spark GraphX

- **GraphX** is a new component in Spark for graphs and graph-parallel computation. At a high level, GraphX extends the Spark RDD by introducing a new graph abstraction.
- GraphX reuses Spark RDD concept, simplifies graph analytics tasks, provides the ability to make operations on a directed multigraph with properties attached to each vertex and edge.

Spark
GraphX

So, Spark GraphX is another Hadoop open source, Apache project and this is the component, which is build over the, core Spark, for computation of a large scale graphs. That is parallel computation of a graph is done using graphx. So, graphx extends the Spark RDD's, by introducing the new graph abstraction. And graphx reuses by Spark RDD concepts that we will see litter on and simplifies than on top of it, using the different graph analytics, which basically are graph algorithms, which will be applied on this frame work.

Refer slide time :( 26:42)

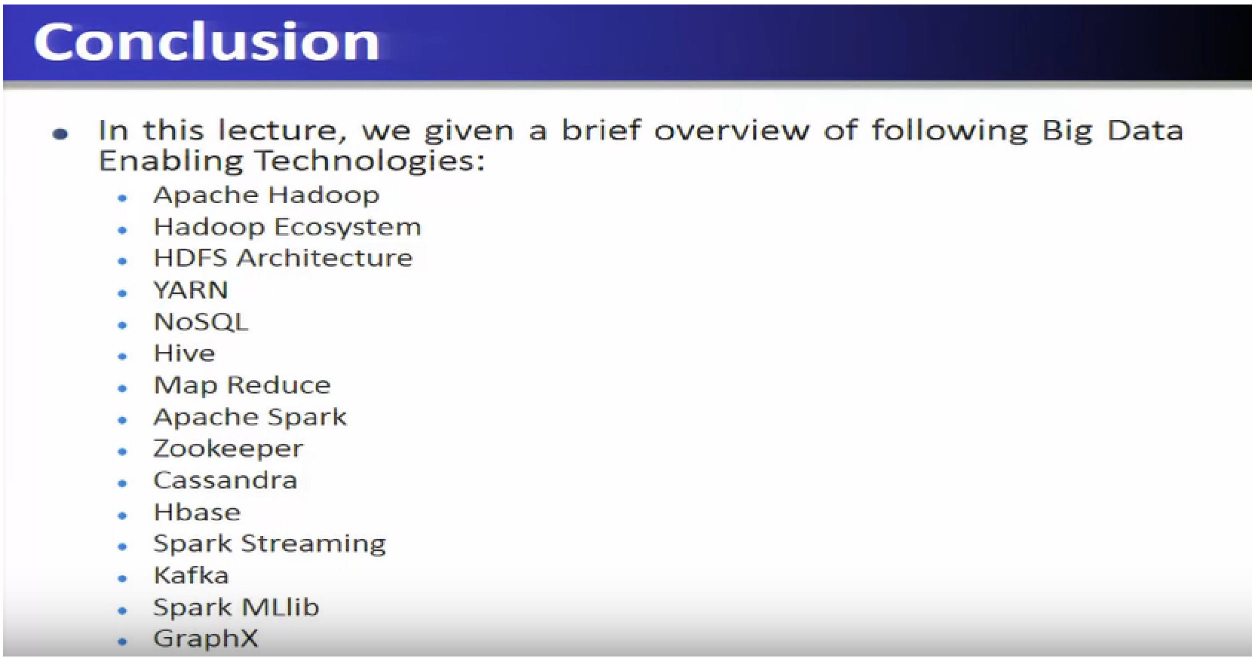


## Spark GraphX

GraphX is a thin layer on top of the Spark general-purpose dataflow framework (lines of code).

So, graphx if you see in more detail and this over Spark core and the algorithms which are available, for, for doing the graph analytics, are Page Rank algorithm, Connected Component algorithm, and the K – core and the Triangle Count, all these algorithms are already implemented and made available as a form in, in the graphx. So, graphx is thin layer on top of the Spark general purpose data flow and that we will see in more detail and this is being heavily used for, various analytics, that is, ‘The Graph Analytics’.

Refer slide time :( 27:25)



## Conclusion

- In this lecture, we given a brief overview of following Big Data Enabling Technologies:
  - Apache Hadoop
  - Hadoop Ecosystem
  - HDFS Architecture
  - YARN
  - NoSQL
  - Hive
  - Map Reduce
  - Apache Spark
  - Zookeeper
  - Cassandra
  - Hbase
  - Spark Streaming
  - Kafka
  - Spark MLlib
  - GraphX

In conclusion, we have briefly, overviewed the big data enabling technologies, in short, we have discussed about the, Hadoop Ecosystem ,which comprise of, the, the HDFS, File system, Yarn, Map Reduce, on then, all map reduce we have seen, Hive and Pig. And then, various NoSQL data databases, we have seen, the, the Cassandra, HBase and then we have seen, using this HDFS, then we have seen in this, frame work, Apache Spark and or Spark we have also seen, the, the frame work, for the Streaming computation, .that is Spark Streaming and graph computation, that is called graphx and for machine learning, we have seen MLlib. Also we have seen our Kafka, for, getting the, means, streams, into the Spark Streaming system, for computation. So, that is all and in the next videos, we will be going in, more details, of these different, Apache Spark project, for big data computation and we will see how, we will be using them for different applications, these projects.
Thank you