

Lecture 3

Hadoop Stack For big data

A dupe a stack for Big Data preface

Refer slide time :(0:17)

What is Hadoop ?

- Apache Hadoop is an open source software framework for storage and large scale processing of the data-sets on clusters of commodity hardware.

Handwritten notes:

- Today's leading technology is Hadoop which is an open source framework for reliable, scalable, distributed computing for Big Data Analytics.
- 100s/1000s of machines Hadoop - Storage - Large Scale Processing
- Diagram: A cluster of nodes (racks) with Hadoop running on them.

Content of this lecture, in this lecture we will provide insight into Hadoop technologies, its opportunities and challenges for Big Data computing. We will also look into the Hadoop stack and the various applications and technology associated with it for Big Data solutions; let us trace back the beginning of the Hadoop. So, today the leading Big Data technology is Hadoop, which is an open source, software framework, for the reliable, scalable distributed computing, for big data analytics. So, this particular Apache Hadoop, open source software framework is free and is used for software for storage and, large scale processing of a big datasets, on cluster of commodity hardware. So, we assume here in this case there exists a, a cluster of commodity machines which is nothing but in the form of the racks, and several such racks, will form a cluster of nodes. So, this particular Hadoop will run on this cluster machine, and provides, the solution for storage. so, storage will be available on these nodes ,which are hundreds and thousands of the nodes, which Hadoop will use, as a storage.


Similarly the computation on these particular data sets, which are stored on this particular cluster, will require the large-scale computation. Wherever the data is stored. So, the data so this Hadoop, will provide a framework for storage and the large-scale processing, large-scale processing, by this we mean that the data set is too large and it cannot fit in the existing, conventional systems, maybe whatever is the size of the conventional system, one system cannot basically accommodate the entire data set. So, for a large size data set it requires hundreds and thousands, of such machines to store the data. Now when a computation is to be performed on these particular data sets, then it is not possible to bring all the data at one place for the computation rather, computation has to be moved wherever data is stored hence, it is called a large-scale processing or a computation of that particular data set, and that is possible with the help of clusters with of commodity hardware, which comprises of hundreds and thousands of machines. So, in this particular discussion we will first see the, the functionality of Hadoop, we will see its beginning also and all these things we are going to discuss. So, again in a nutshell we have to before we proceed ,we have to understand that the today's leading technology, leading big data technology is Hadoop , which is, which is an open source framework for reliable, scalable, distributed computing, for the big data analytics, we will

explain in this particular lecture. What is we mean by the reliable? What we mean by the scalable? And this particular entire set up, of computation on a cluster is called distributed computing, and the large-scale data sets are, being computed and being performed the analysis which is called a big data analytics, which is useful for driving various applications all these things is uncovered, under the hadoo that is why today's leading Big Data technology is Hadoop. So, with this simple explanation, now we will see into the Hadoop. What this Hadoop? Means, what are the different application it provides and which we are going to use for computing the big data.

Refer slide time :(06:26)

Hadoop Beginnings

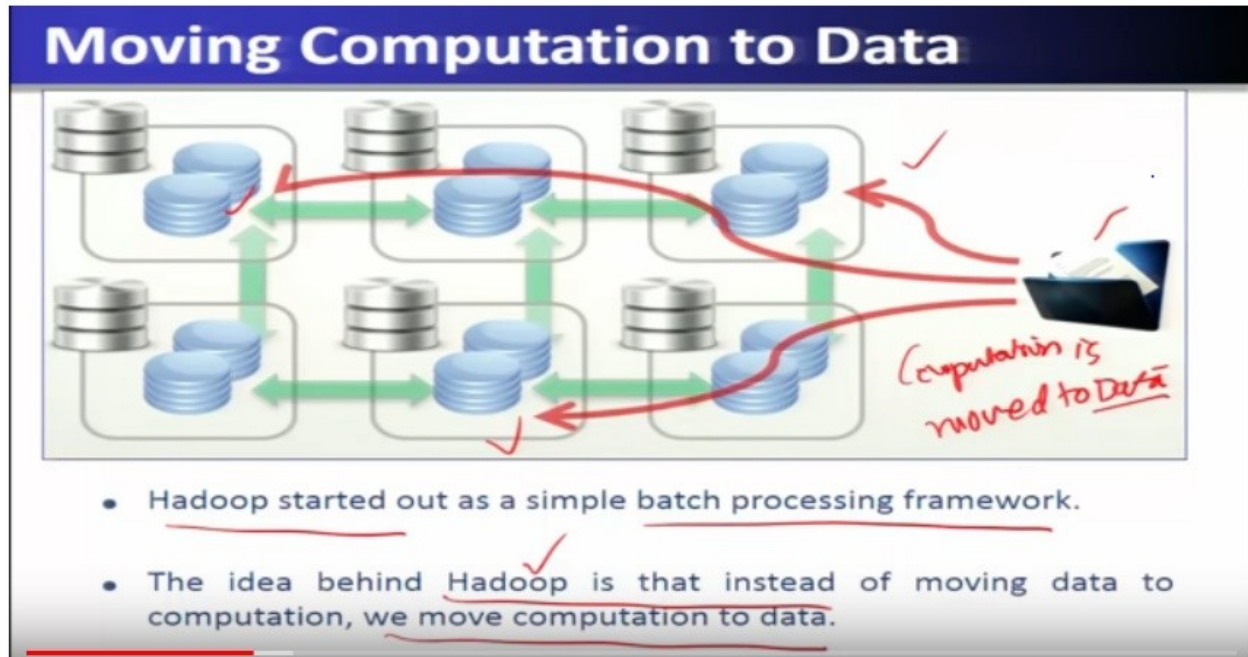
- Hadoop was created by Doug Cutting and Mike Cafarella in 2005
- It was originally developed to support distribution of the Nutch Search Engine Project.
- Doug, who was working at Yahoo at the time, who is now actually a chief architect at Cloudera, has named this project after his son's toy elephant, Hadoop.



The slide titled 'Hadoop Beginnings' contains three bullet points. The first bullet point, 'Hadoop was created by Doug Cutting and Mike Cafarella in 2005', is underlined with a red line. The third bullet point, 'Doug, who was working at Yahoo at the time, who is now actually a chief architect at Cloudera, has named this project after his son's toy elephant, Hadoop.', has a red arrow pointing from the word 'Hadoop' to the Apache Hadoop logo at the bottom right. The logo consists of a yellow elephant icon and the text 'APACHE hadoop' in a stylized font.

So, Hadoop was created by, the cuttings and Mike Kerala in 2005, that is long back. It was originally developed to support the distribution of the nudge search engine project. Doug cutting, who was working at yahoo at that time? Who is now chief? Chief architect at Cloudera, named his project after his sons to I which is an elephant and named as Hadoop. So, Hadoop is the name given to this particular project out of this context. So, this particular icon, which is their representing the Hadoop, is that Y, which duct cutting has named this particular project Hadoop.

Refer slide time: (07:33)



Now, let us go in more details of this particular leading, Big Data technology which is Hadoop. Now, here we see that this big data is too large; to be fit in to a particular system hence it is being stored in the cluster systems. Now, if it is stored in hundreds and thousands, of the nodes then the computation also has to be moved, wherever the data is stored see these particular arrows. So, this shows that the computation, if you want to perform the configuration on this large-scale data sets. So computation has to move wherever the data is stored. And after doing the computation the results will be collected back to the same environment .so, hence the Hadoop is started out as a simple batch processing framework. Why because data is to be store, and that computation can be performed at any point of time, and this is called a batch processing framework. So, Hadoop initially was designed as the simple batch processing framework, and the idea behind the Hadoop is that instead of moving data, to the computation, here we have shown you in this particular picture, the computation is moving, move to the data where our data is stored, and the computation is performing hence the title says that moving. The computation to the data is one of the main idea behind the success of the Hadoop which is the leading technology for big data.

Refer slide time :(09:24)

Scalability

- Scalability's at it's core of a Hadoop system.
- We have cheap computing storage.
- We can distribute and scale across very easily in a very cost effective manner.

Scalability - \hookrightarrow 100s & 1000s of commodity hardware
Scale out -

The second aspect is called scalability, by scalability we mean that, that whatever is the size and this particular size, of particular data, can be processed stored, and being computed that is called scalability. This is achieved using hundreds and thousands of, of machines, which is called a commodity, hardware machines, why commodity hardware they are not very specialized or super specialized computers they are normal computing machines hence it is called commodity hardware, and this ensures our scalability. So, as this particular this is called scale out, scale out technology says that, if you keep on adding more resources, more hardware it will provide, more resources to accommodate this kind of and this is called scalability. So, again to repeat that the scalability is at the core of the design of Hadoop system, and it is basically achieved out of hundreds and thousands of commodity hardware, which is a cheap computing device for storage and for computations. So, hence a large data set, is can be distributed across these hundreds and thousands of the machine, and we can add these machines more in number to scale without any modifications, and we can get more improved performance if more number of such machines, we keep on adding to accommodate the, the, the scale or the size of the data ion also to form the computation, at higher-speed in a very cost-effective manner hence it is called a scalability.

Refer slide time :(11:27)

Reliability

- Hardware Failures Handles Automatically!

failure is norm!



- If we think about an individual machine or rack of machines, or a large cluster or super computer, they all fail at some point of time or some of their components will fail. These failures are so common that we have to account for them ahead of the time.
- And all of these are actually handled within the Hadoop framework system. So the Apache's Hadoop MapReduce and HDFS components were originally derived from the Google's MapReduce and Google's file system. Another very interesting thing that Hadoop brings is a new approach to data.

Now, another important thing is so, this is called reliability. So, when we see that we are using the hundreds and thousands of commodity machines, commodity hardware which are not very sophisticated, hence they are prone to the failures. So, failure is a norm, and this is the basically prime, design, aspect which is called reliability that is the fault tolerance is one of the basic design paradigms for in the design of the hadoop system that we will see. So, if we see that an individual machine, are basically the rack of machines, or a large cluster or the, the big supercomputer they can fail at any point of time or some of their components can also fail. So, failure is very common, and that, that have to be accounted for this kind of failure well ahead in time, without any disruption in the application computation. So, and all of these are actually handled within the Hadoop framework .so, Apaches Hadoop Map Reduce and SPFs component were originally derived from Google's, Map Reduce, and Google's, Google file system. So, there the design of the Google file system, and Google Map Reduce already has attended or handled this reliability or a fault tolerant as one of the basic design issue. So, we will see that how this is achieved here in the hadoop system about, reliability of these failures to tolerate with that.

Refer slide time :(13:19)

New Approach to Data: Keep all data



- A new approach is, we can keep all the data that we have, and we can take that data and analyze it in new interesting ways. We can do something that's called schema and read style.
- And we can actually allow new analysis. We can bring more data into simple algorithms, which has shown that with more granularity, you can actually achieve often better results in taking a small amount of data and then some really complex analytics on it.

Now, here doing all this new approach to the data keeping all the data always. So, we see that in the new approach in Hadoop, we are going to keep all the data we have, and we can take the data and do the analysis or in many interesting ways. So, we are not constrained about fitting the data into the schema and then storing it into that form rather all the data will be kept as it is and whenever it is read it will be fitted into the schema and being available for the analysis. So, that means that keeping all the data that means the data will be while reading, while reading the data, it will be fit into the reading into the schema. So, this will simplify the analysis, and at the time of storing, we have to just keep all the data without any constraint of fitting it into the schema, and this will bring into the more data into the simple algorithms, and the analysis becomes, easier in here in this case. So, while we will discuss the analytics, part complex analytics, when we perform on the big data then, we will see that this aspect of this particular design that is reading into the schema becomes, quite easier to design the complex analytics angel.

Refer slide time :(15:00)

Apache Hadoop Framework & its Basic Modules

Summary Design issues in Hadoop
↳ Reliability
↳ Scalable
↳ Keeping all data ready on schedule
↳ moving computation to the data

So, with this different design issue, which we have discussed, let us summarize them one the issue which was called reliability. The second design issues, that we have just covered let us summarize, it first issue was the reliability, scalability, and keeping, all the data. So, that while reading, on the schema, and this was the different and moving data to the computation, moving computation, to the data. So, these are the four different main design issues, which basically have made the leading technology for big data computation, which has made this Hadoo as the leading technology for big data computation, Apache Hadoop framework and its basic modules.

Refer slide time :(16:34)

Apache Framework Basic Modules

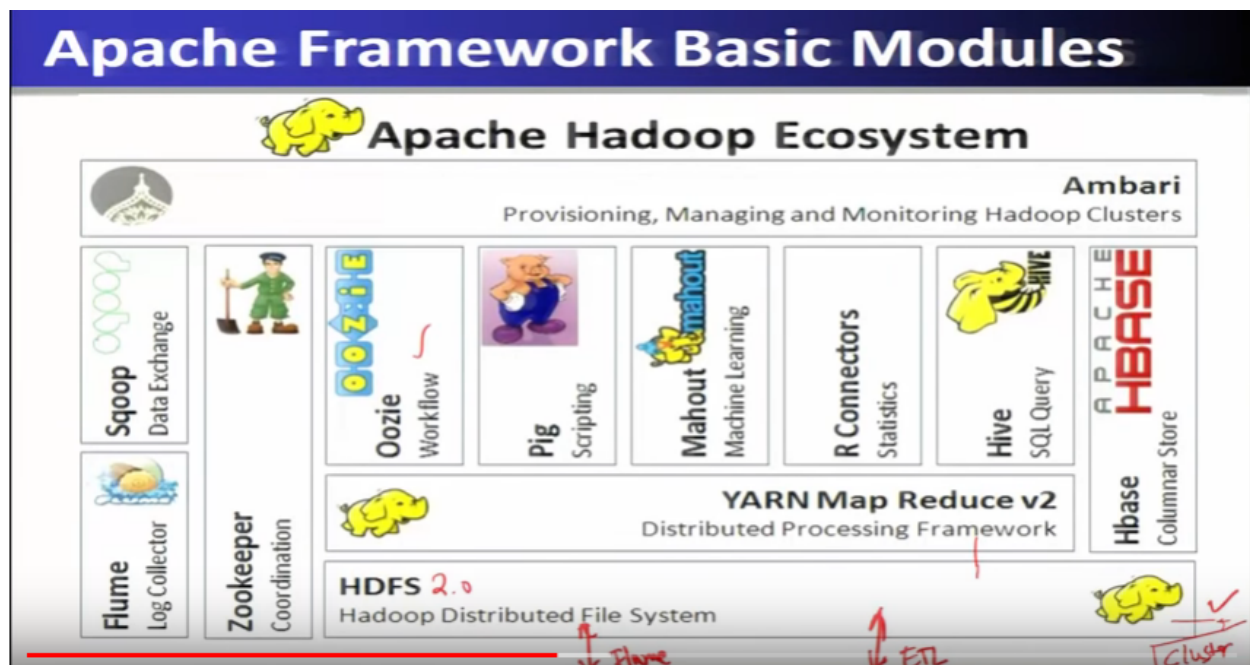
- **Hadoop Common:** It contains libraries and utilities needed by other Hadoop modules.
- **Hadoop Distributed File System (HDFS):** It is a distributed file system that stores data on a commodity machine. Providing very high aggregate bandwidth across the entire cluster.
- **Hadoop YARN:** It is a resource management platform responsible for managing compute resources in the cluster and using them in order to schedule users and applications. *Resource Manager Scheduler ✓*
- **Hadoop MapReduce:** It is a programming model that scales data across a lot of different processes.

So, Apache Hadoop has four different basic modules, the first one is called Hadoop common, it contains the libraries and utility is needed by other Hadoop model, modules. The second module is called Hadoop distributed file system that is HDFS, it is a distributed file system that stores, the data on the commodity

machines. So, this file system will insure how this particular data or a big data is to be distributed, and stored or across different hundreds and thousands of the node, and keep track of all these data whether they are available, or on the node whether the nodes are alive containing those data or they're not alive in all the cases this particular, Hadoop distributed file system will do the management, and user will be given this kind of service. So, this Hadoop will provide a very high, aggregate bandwidth across the entire cluster so, this storage and the retrieval of the huge amount of data over the commodity, over the cluster, will become is possible here using Hadoop to provide this axis at a very high educated bandwidth so, that it is, it's performance also we are going to cover in this part of the course. Now, the next module which is another basic module of Apache Hadoop is called yarn. So, yarn is the resource manager or it does the resource management for managing the computer resources in the cluster and using them in order to schedule the user ion application.

So, it is the resource manager and the she jeweler, by saying that resource manager and the she doula means the resources which are available on the cluster it will keep track of it and whenever is required by the different application, it will do the she ruling that means it will allocate and schedule for different resources, for different users and application that is called yarn in the map in the hadoop. And the fourth basic framework ,of Hadoop is called Map Reduce this is the programming paradigm and which will ensure the computations will reach wherever the data is using this particular program that is called Map Reduce. So, Map Reduce is the basic programming model, using which all the programs, application programs can be, can be designed to run in this entire cluster system. So, this particular programming model ensures that scalability is ensured while it is achieving the performance, while computing the big data that we will see in more detail about all these four different modules of Apache Hadoop.

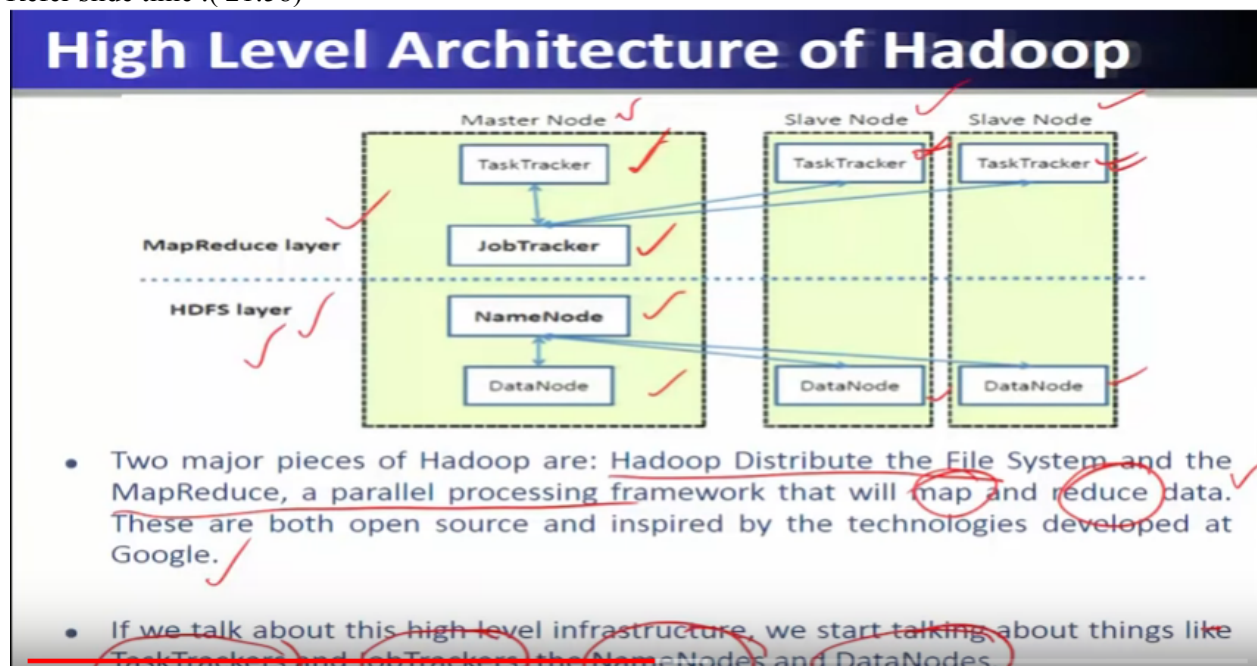
Refer slide time :(20:01)



So, Apache Hadoop basic modules, we can see here and in this particular slide, which basically is about the Apache Hadoop ecosystem. Let us see at the end or at the bottom most, part of this particular ecosystem, here is an HDFS, and below this the data will be pulled either through the flume or through

the through the databases, that is the ETL, which will provide the data, into this particular system which will be stored using Hadoop file system on the cluster. So, this HDFS, Hadoop file system, Hadoop distributed file system will run on the cluster. So, the hardware is the cluster on which the Hadoop distributed file system would run yarn, and will be running over top of HDFS. So, this HDFS is 2.0 versions. So, yarn and Map Reduce version 2 they will run on top of this HDFS, and using John and Map Reduce there are different other applications, of Hadoop system which will run that is called Suzie Pig mahute and hi H Base , skew ,flume, zookeeper. Now, let us see in more detail about all these different applications, which are available in the Hadoop ecosystem and we will summarize, them they are used and they are going to be useful for the Big Data computation.

Refer slide time :(21:58)



Before going, there let us see the high-level architecture of the hadoop. So, as you know that the Hadoop runs on the cluster system, and cluster system comprises of hundreds and thousands of the nodes, and each node, an they will be running different applications, or the modules of Hadoop system and we call them one such models called the master, the other nodes are called the slave nodes. So, as far as Map Reduce HDFS, layer is concerned so this particular master will contain, master node contains the module of HDFS layer, which is called a name node and all other nodes will contain another module of HDFS layer which is called a data node. So, all the data nodes which are running on different nodes I will communicate with a single name node of HDFS layer, similarly these nodes are also being used for the, the Map Reduce layer. So, the Map Reduce layer will contain the task tracker this particular job, of task tracker will run on those nodes similarly there will be for every node there will be a task tracker and whenever the jobs are running they will basically form the task job tracker. So, task trackers will be running on all the nodes and there will be a single job tracker. So, for every application this job tracker will be created and the, the task of that job or the application will be actually running using the tasks using the task tracker. So, this particular combination of these two different services run by the Map Reduce and HDFS, will run on will be launched on the same set of the nodes. So, hence two pieces of Hadoop are, the two major components of the Hadoop are HDFS, file system and the Map Reduce

parallel processing framework and that will map and reduce the data. Means that the data which is stored by HDFS will be performed for computation, that is called parallel processing framework given by the map and reduce. So, the programs which are called map will reach the computation wherever the data is stored called a map, and after performing those computation the, the aggregate values are being collected that is called reduce of that particular on that data item and these technologies were initially conceived at the Google, and which was open source by the Yahoo, that is an Apache Hadoop, which is basically is available on free for big data computations. Here, in this particular high-level architecture we have referred the names like task tracker, job tracker i named nodes i on data nodes all these different components which meow used here in this high-level infrastructure, for describing the architecture of Hadoop is going to be described in more detail in further slides.

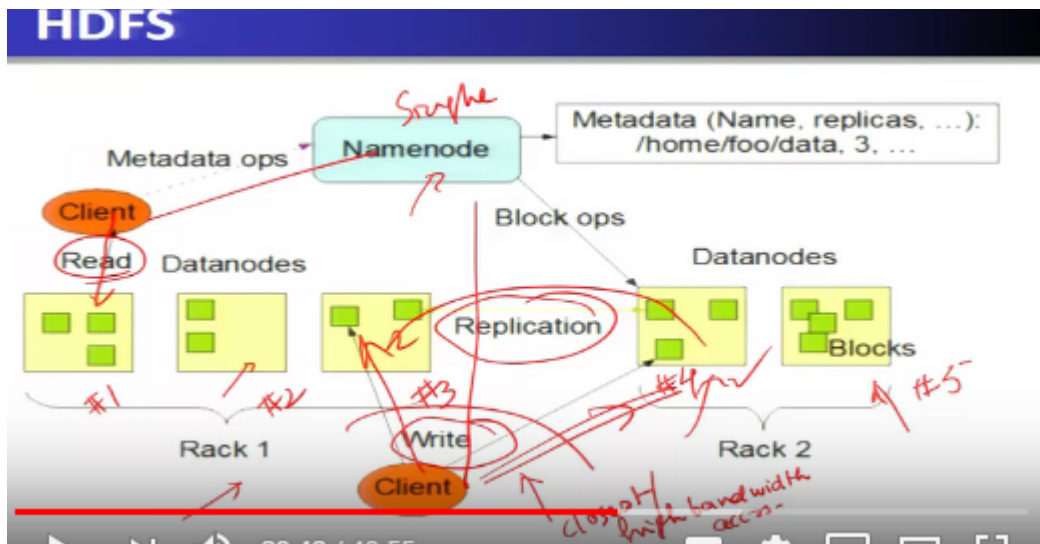
Refer Slide Time :(25:56)

HDFS: Hadoop distributed file system

- Distributed, scalable, and portable file-system written in Java for the Hadoop framework.
- Each node in Hadoop instance typically has a single name node, and a cluster of data nodes that formed this HDFS cluster.
- Each HDFS stores large files, typically in ranges of gigabytes to terabytes, and now petabytes, across multiple machines. And it can achieve reliability by replicating the cross multiple hosts, and therefore does not require any range storage on hosts.

so, as GF s stands, for Hadoop distributed file system which is a scalable, reliable, distributed computing or storage platform and this particular platform is, written in the Java, now the node in Hadoop instance has a single name node and also the different data nodes, together will form an H and HDFS cluster. So, that means, that in HDFS cluster the there is, one node which is called as our name node there is a single name node, and there are multiple data nodes which basically, will be running this HDFS cluster. now each HDFS will store the large files typically, in the range of kilobytes to, terabytes and, now the petabytes, across multiple machines, and it can achieve the reliability by, replicating across multiple hosts, and therefore does not, require any range storage on the coast.

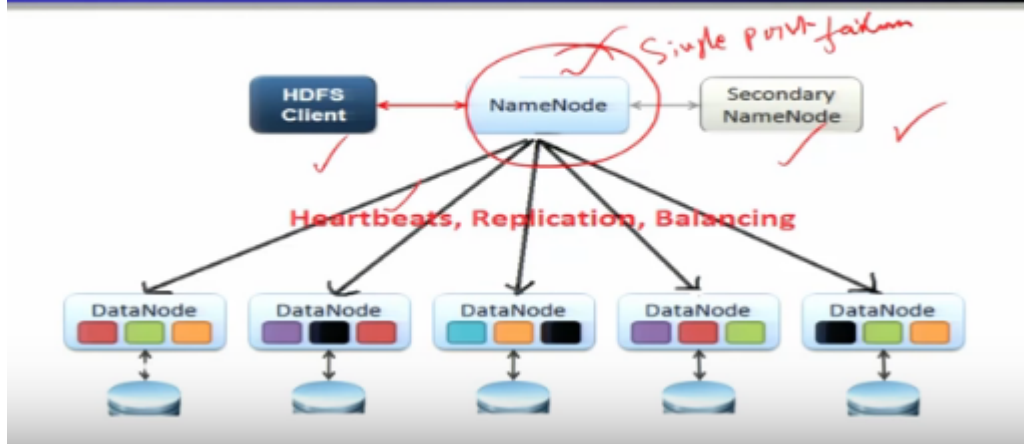
Refer Slide Time :(27:23)



So, let us see the more details of these name nodes, and data nodes in this particular diagram you see that there is, a single name node there are multiple data nodes which are running on different machines. let us assume that this rack has machine one, machine two, iron machine three, different nodes similarly, on another rack it has another machine that is, machine number four, and machine number five, these different one, two, three, four, these machines are running the data nodes and there is one of them will be running a single name node on the same machine let us assume that, this particular HDFS this is, the architecture of HDFS when it runs on the on, on the cluster. Which comprises of several racks? So, whenever a client wants to do, the operation which is called the write operation, and the read operation, they have to go through this, particular name node, and name node, will then guide them where the data is stored actually, and the closest of them for example data is stored at the node 3 and as, for as node 4 so, it will prefer the closest one this is called closest. To provide the high bandwidth access, to the data similarly and it will write on this particular the closest one and then all the replicas, for example replication of this, particular data at more than one nodes, will be done in the terms of blocks, that is called replication these details we are going to explain in the further slides, in more detail similarly there is another operation which is called a read so when a client wants to read a particular data which is stored in HDFS it will know, through the name node where those data is stored and then the closest one or the nearest one replicas will serve as, the read for that client.

Refer Slide Time :(29:46)

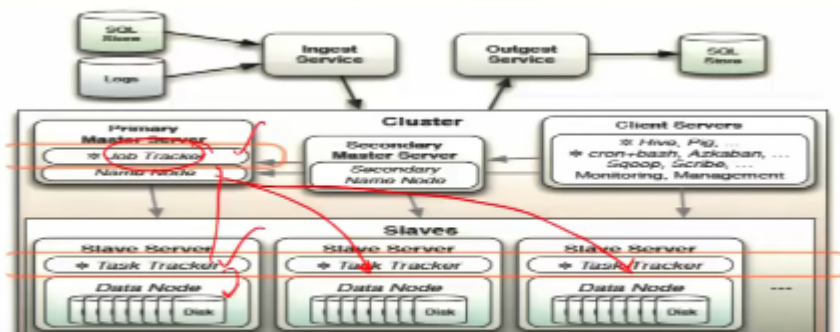
HDFS



Now as far, as the name node is concerned there is a single name node it will keep track of the data which are now stored on the data nodes, whether they are, alive or they are dead so hence these heartbeats are continuously, exchanged between the name node and, the data node to know, the position and the situation of the data which is being managed by or restored by, the data nodes and whenever they are down that means a fault tolerance has to takes place then it will be removed out of this, name node name node will not give the reference to all other clients, requests for that particular date are not referencing and when ever again it will come up then again it will be modified or it will be made consistent, synchronized with all the previous updates, for the replicas and then only it will be made available to the further client requests. Now there is a single name node which is a prone to the failure, single point, failure to avoid this, kind of situation a secondary name node passively, try to keep a most recent backups so, if the name node downs, then secondary name node can become the, the, the primary name node and can continue to operate in this, environment that is Hadoop distributed file system.

Refer Slide Time :(31:19)

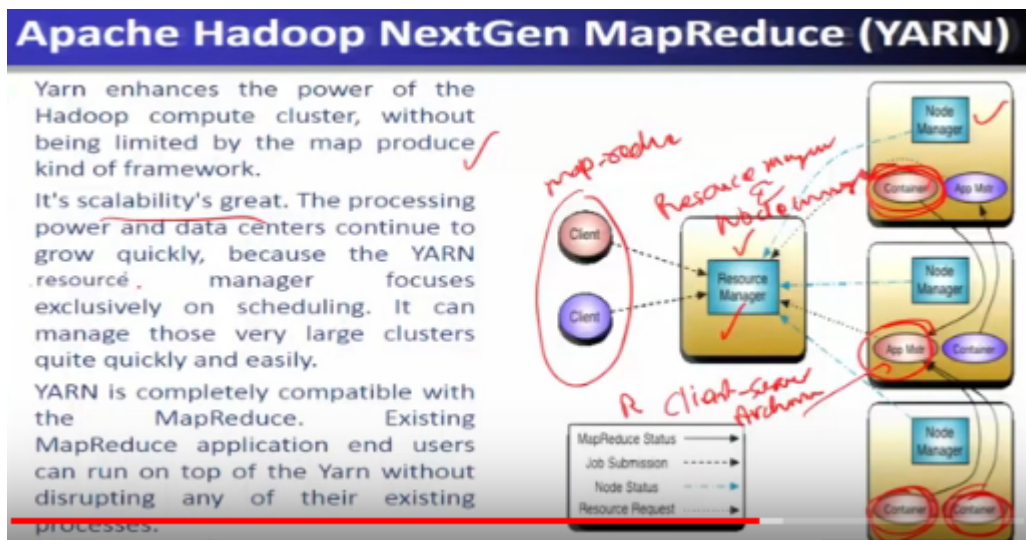
MapReduce Engine



- The typical MapReduce engine will consist of a job tracker, to which client applications can submit MapReduce jobs, and this job tracker typically pushes work out to all the available task trackers, now it's in the cluster. Struggling to keep the word as close to the data as possible, as balanced as possible.

Operations, now the compute engine which is called the Map Reduce engine will perform the parallel computation or this, architecture which is provided by ,HDFS so, if you see here that HDFS was having name node and the data nodes, the similar kind of notion ,Map Reduce also, will provide so, Map Reduce will have the job tracker, and task tracker, so job tracker, will now keep on, tracking or keep on communicating with the help of task trackers, and the task trackers, will actually use the, the data nodes, and perform the contradictions, that is the map operation wherever the data is available that is done by, the task tracker, and after the computation the reduced function will be performed and hence there is, a combination of a job tracker and, a task tracker so, again, let us summarize the typical, Map Reduce engine will consist of a job tracker, to which the client applications can submit the Map Reduce jobs, and this job tracker typically pushes, the workouts to all other available task trackers so, as I told you that this job tracker will push out to all other task trackers, now it's the cluster, now which is there in the cluster? struggling to keep the words as close, as close to the data as possible and it will be done, through the balancing so, all these optimizations are basically, online managed by, these set of tasks job tracker and, the task tracker combination in the Map Reduce engine.

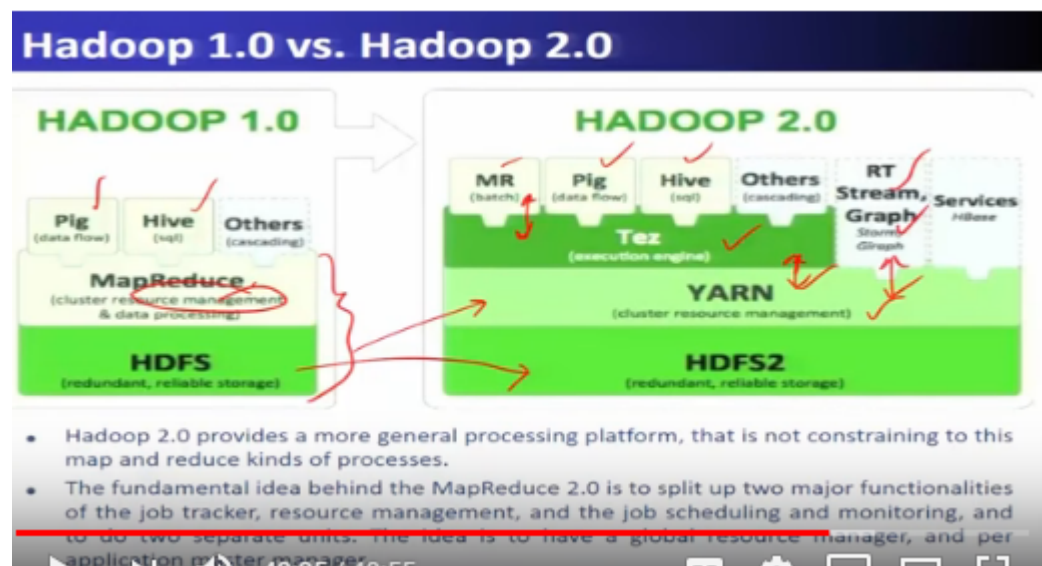
Refer Slide Time :(33:22)



Now the next important component of the Hadoop is called yawn, now yawn highest yawn, is there to provide two important services, one is called resource manager which is, is which is, at the core at the heart, the other one is called scheduling how? this, particular different resources are scheduled to the Map Reduce jobs, and this task, this is the task, of the yawn so, yawn enhances, the power of the Hadoop compute cluster without being limited by, the map produced Map Reduce kind of framework now its scalability, is great in the sense that the processing power and the data center continue to grow quickly because, Yawns Resource Manager focuses exclusively, on scheduling it can manage those very large clusters quickly, and easily, so yon is completely compatible with Map Reduce and existing Map Reduce application and user can run on top of yon without any disrupting of their existing processes, let us see through this particular figure about the components ,of the yawn so, yawn has two components here shown as the resource manager and, the node manager so, there is a single resource manager so, whenever the client submits their job let us say it's a Map Reduce job which requires, the resources are to

be allocated hence this is acting as the resource manager. So, the resource manager after receiving the request, from the client it will now, contact to the node manager and, the node manager in turn will ,will assign the resources as, far as, the application is concerned every application has, application master and whatever resources is being allocated by, the node manager through the resource manager these, particular are assigned to the application master and which is known, as the containers here they, are shown as in the pink color so, this particular application master will know where? what are the different containers, are allocated by, the node by the node manager and this, particular application master will independently run so ,these application master container they are, are the resource blocks, which are allocated by ,the resource manager with the help of node manager so, note when a resource manager and, the node manager, is the client-server architecture of the yarn resource manager and, the node manager is, client-server architecture so, yarn is work in the client-server architecture where the node manager will take the request from the client, and with the help of several node managers it is, trying to allocate the resources and do the Scheduling of these activities, we are going to discuss this in more details.

Refer Slide Time :(36:58)



Now, we will see that, the transition from Hadoop 1.0 to, Hadoop 2.0 so, Hadoop 2.0 has, become very much flexible so, newer applications are also able to execute or under Hadoop 2.0 which was not possible using these drawbacks, of Hadoop 1.0 let us, see these details why nowadays, we are using Hadoop 2.0 and, discarded to use the original Hadoop 1.0 so, Hadoop 1.0 has, two components only, that is HDFS and Map Reduce so, that means the resource management and scheduling also was the part of the, the programming paradigm that is called Map Reduce with this, release of this responsibility a yarn is being included so, that is source management and scheduling which is done by, the yarn so, yarn is, a cluster resource manager, and HDFS therefore has some more capabilities, added hence its name is HDFS version 2, and which supports to run the yarn on top of it and about that yarn there is execution, engine which is called at age? And in some of the distributions, the stage is used and about age this Map Reduce programs, will run as the batch processing of the big data computations, there are other applications such as Pig high, which used to run earlier in the version Map Reduce version 1, about the Map Reduce

nowadays, this big and high are now running using the Map Reduce and this version 2, which is shown over here big and high that is running continuously, similarly, other applications such as, the stream processing ,or graph processing, and machine learning, all they run and storm and 0 up they run above the yarn system without the Map Reduce. So, Map Reduce 2.0 provides, a more germinal, processing framework that is not constraining to this map and reduced kind of processes, so the fundamental idea behind Map Reduce 2.0 is, to split two major functionality, of a job tracker resource management and the job scheduling and monitoring to two separate units, hence some of the responsibilities, that is the resource management responsibilities, of a Hadoop of a Map Reduce point one is now, broken up as Map Reduce version 2, I yarn. Hence this becomes, more flexible and for paving a way to a newer application that we will see which are possible under Hadoop 2.0, version.

Refer Slide Time :(40:08)

What is Yarn ?

- Yarn enhances the power of the Hadoop compute cluster, without being limited by the map produce kind of framework.
- It's scalability's great. The processing power and data centers continue to grow quickly, because the YARN research manager focuses exclusively on scheduling. It can manage those very large clusters quite quickly and easily.
- YARN is completely compatible with the MapReduce. Existing MapReduce application end users can run on top of the Yarn without disrupting any of their existing processes.
- It does have a Improved cluster utilization as well. The resource manager is a pure schedule or they just optimize this cluster utilization according to the criteria such as capacity, guarantees, fairness, how to be fair, maybe different SLA's or service level agreements.

Scalability MapReduce Compatibility Improved cluster utilization

Refer Slide Time :(40:33)

What is Yarn ?

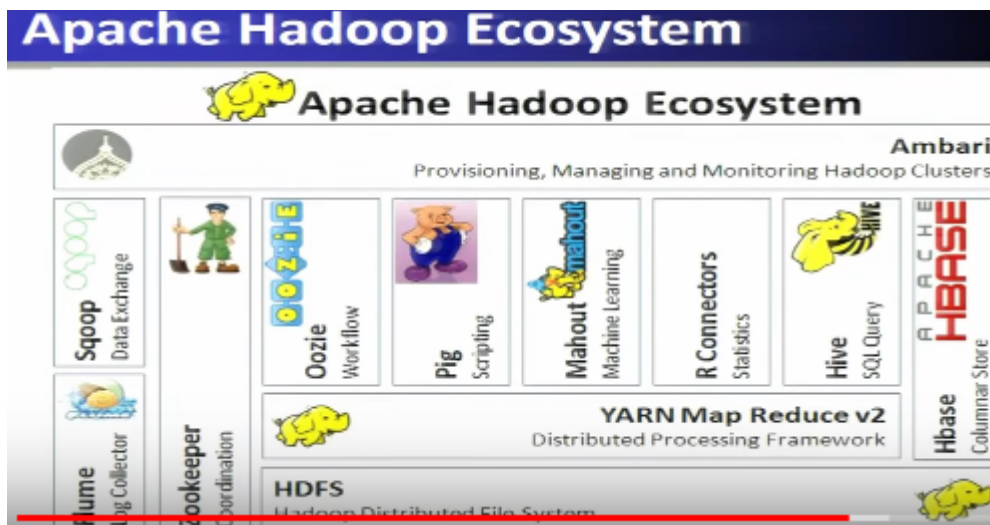
- It supports other work flows other than just map reduce.
- Now we can bring in additional programming models, such as graph process or iterative modeling, and now it's possible to process the data in your base. This is especially useful when we talk about machine learning applications.
- Yarn allows multiple access engines, either open source or proprietary, to use Hadoop as a common standard for either batch or interactive processing, and even real time engines that can simultaneous acts as a lot of different data, so you can put streaming kind of applications on top of YARN inside a Hadoop architecture, and seamlessly work and communicate between these environments.

Fairness Iterative Modeling Machine Learning Multiple Access Engines

Supports Other Workloads

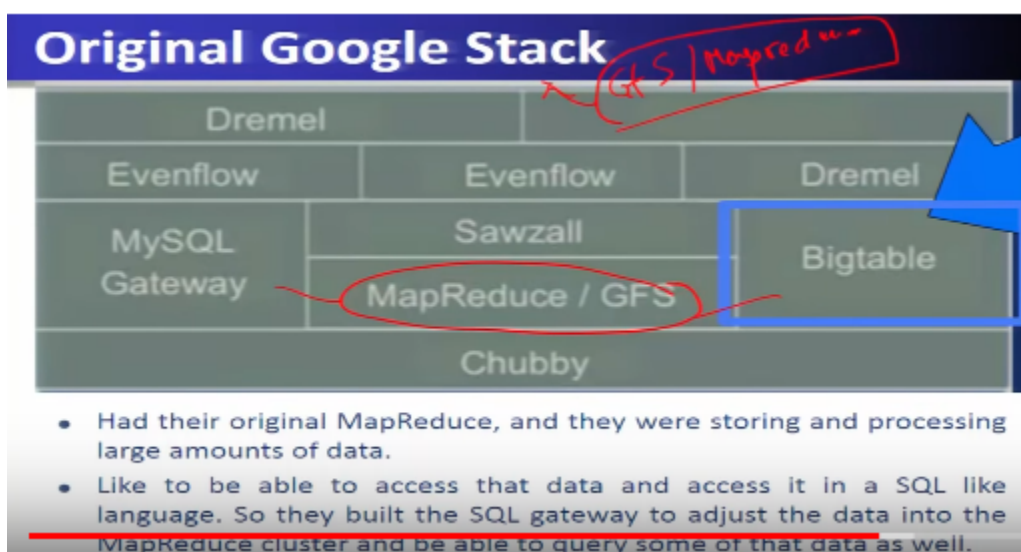
So, what is the yawn? So yawn is a resource manager? that is the full form of yawn is yet another resource negotiator so, yawn enhances, the power of Hadoop compute cluster without being limited by, the map and reduce so, map all these things, we have already covered.

Refer Slide Time :(40:34)



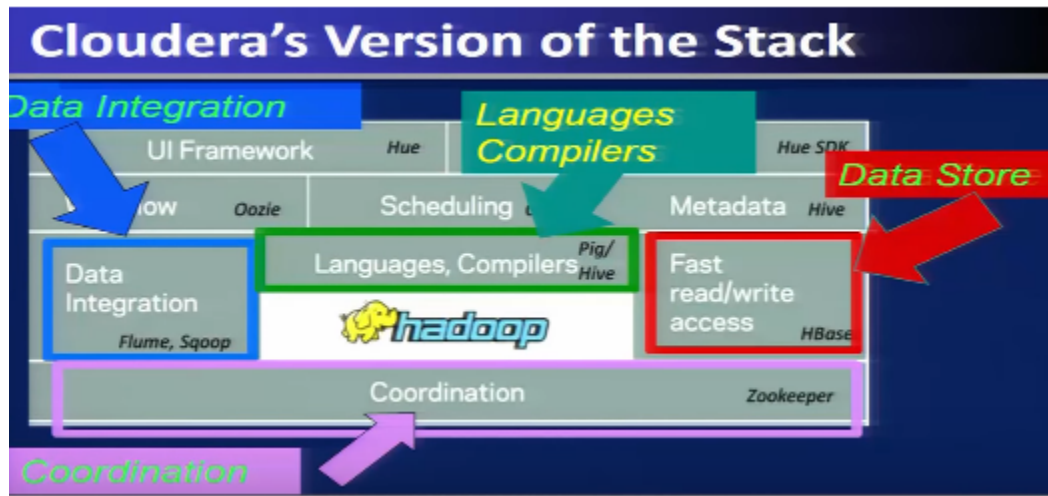
Now, let us go and see what do, we mean by the Hadoop Zoo. Now in this Hadoop ecosystem we see there are lot of icons, and most of these icons, that is Hadoop is representing try a different similarly, there are some toy Pig and all these different animals, which are becoming an icon to these applications are require a coordination service hence the word zookeeper, is being used as the coordination service. So we will see all these more applications, in more detail one by one.

Refer Slide Time:(41:12)



Now before that we will see the distributions so, the distributions, of Hadoop are this big data is called the stack so, initially it was the, the Google GFS, Google file system and the Map Reduce hence it was called a Google stack initially, so Google stack was using initially, the Google file system and the Map Reduce and it was also using the database which is called a big table,, and MySQL gateway and, and so, on so this, was called a Google stack.

Refer Slide Time :(42:00)

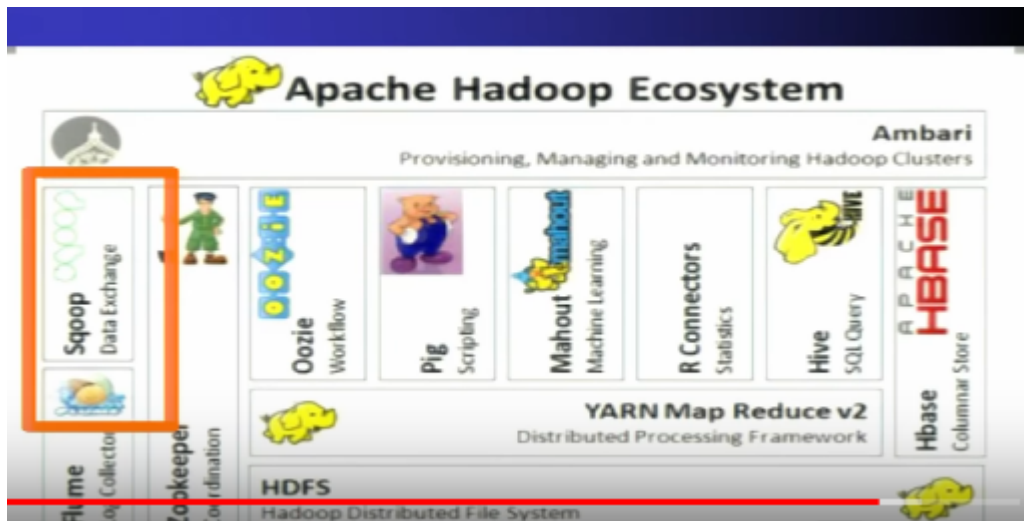


So, the now as, far as the next distribution, of this Hadoop is another distribution is from the Face book and under this particular distribution we see that there will be there is a Hadoop and then zookeeper, H Base hi all these different applications, together will give another stack Yahoo's, version of this distribution is called Yahoo stack, which is nothing but a Hadoop stack so Hadoop stack also contains Hadoop and this H Base and all these components, which are shown over here. Link Dell, link dell, distribution is also called its version of stack. The most important version most important distribution is called the cloud Eros distribution.

Refer Slide Time :(42:55)

Hadoop Ecosystem Major Components

Refer Slide Time :(42:58)



Now let us see the Hadoop ecosystem and its major component let us start with a with a scoop.

Refer Slide Time :(43:02)

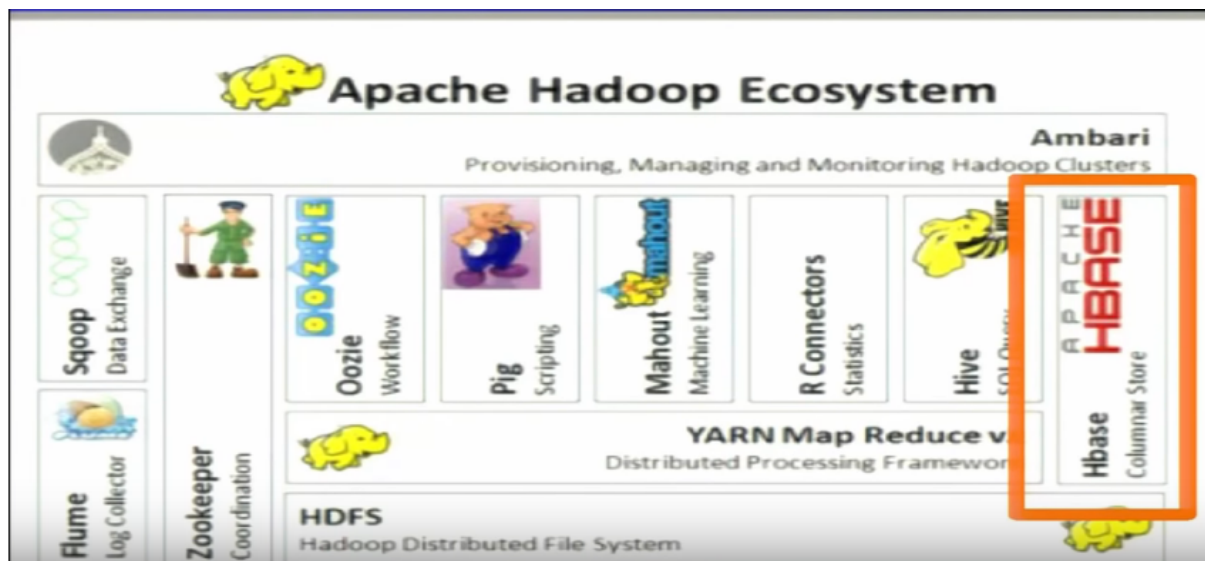
Apache Sqoop

- Tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases

The diagram shows a cylinder representing a database. The top part is labeled "hadoop" with a yellow elephant logo. The middle part is labeled "SQOOP" in yellow. The bottom part is labeled "SQL" and is circled in red. Two blue arrows point upwards from the SQL section to the Hadoop section, and one blue arrow points downwards from the Hadoop section to the SQL section. A red arrow points from the SQL section to the SQOOP section.

so, you see that a scoop is application of the Hadoop ecosystem so patch a sqoop is full form is basically the SQL on Hadoop so, you see that the SQL is basically the database and this entire database is now pulled into the Hadoop system hence it is called sqoop that is SQL on the Hadoop it is so it is the tool and it is the application for efficiently transporting bulk data between the Apache Hadoop and the SQL data store.

Refer Slide Time :(43:50)



Now, the next application is we are going to touch upon is Apache H Base.

Refer Slide Time :(44:00)

HBASE

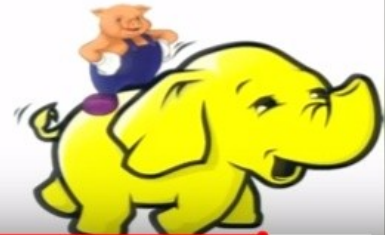
- Hbase is a key component of the Hadoop stack, as its design caters to applications that require really fast random access to significant data set.
- Column-oriented database management system *Detected*
- Key-value store ✓
- Based on Google Big Table ✓
- Can hold extremely large data ✓
- Dynamic data model
- Not a Relational DBMS

So, HBASE is a key component of Hadoop stack and it and its design data to the application that require really fast random access to the significant data set. So, it is H Base is nothing but a column oriented, distributed, database management system, which is based on key value store, the design of H Base is based on the original Google's Big Table and it can hold extremely large data, data set for storage and writable purposes so, it is a, it is now based on the dynamic data model and it is not a relational DBMS, hence its it is are no sequel, data model the next application is called the pig.

Refer Slide Time :(44:55)

PIG

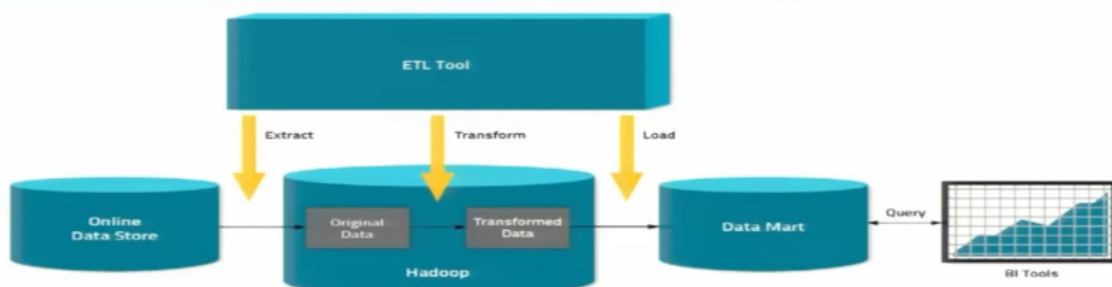
- High level programming on top of Hadoop MapReduce
- The language: Pig Latin ✓
- Data analysis problems as data flows
- Originally developed at Yahoo 2006



which is a scripting language on top of Hadoop Map Reduce .so instead of going to the complication of a complex Map Reduce application program, rather simple view of this scripting language is being provided and that language is called a pig Latin, and this is useful for the data analysis and as the data flow. So, it is based on data, data flow model and it was originally developed at Yahoo in 2006.

Refer Slide Time :(45:34)

PIG for ETL



- A good example of PIG applications is ETL transaction model that describes how a process will extract data from a source, transporting according to the rules set that we specify, and then load it into a data store.
- PIG can ingest data from files, streams, or any other sources using the UDF: a user-defined functions that we can write ourselves.
- When it has all the data it can perform, select, iterate and do kinds of transformations.

And pig is used for ETL and here, you can see that the, the traditional ETL technologies, ETL stands for extract transform and load. So, out of the different databases it will store and this pig is used for doing the analysis.

Refer Slide Time : (46:05)

Apache Hive

- Data warehouse software facilitates querying and managing large datasets residing in distributed storage
- SQL-like language!
- Facilitates querying and managing large datasets in HDFS
- Mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL

The next application is hive, which is an SQL query. So, using SQL query or the Map Reduce, this hive will basically perform the, the storage system and the, the analysis in a much easier manner.

Refer Slide Time : (46:26)

Oozie



- Workflow scheduler system to manage Apache Hadoop jobs
- Oozie Coordinator jobs!
- Supports MapReduce, Pig, Apache Hive, and Sqoop, etc.

Another application is called Oozie. And here, the workflow scheduler system is to manage the Hadoop jobs using Oozie. So, you see is another coordinator of jobs and it supports Map Reduce Pig hive and is sqoop.

Refer Slide Time :(46:39)

Zookeeper

- Provides operational services for a Hadoop cluster group services
- Centralized service for: maintaining configuration information naming services
- Providing distributed synchronization and providing group services

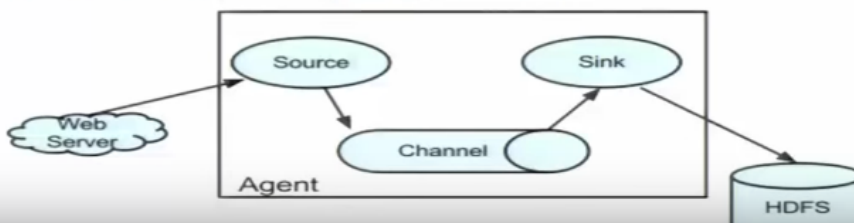


Another coordination service is called a zookeeper, which provides the coordination service and it will give you a centralized service, for maintaining the configuration and the naming service, it provides the distributed synchronization and the group services.

Refer Slide Time :(47:04)

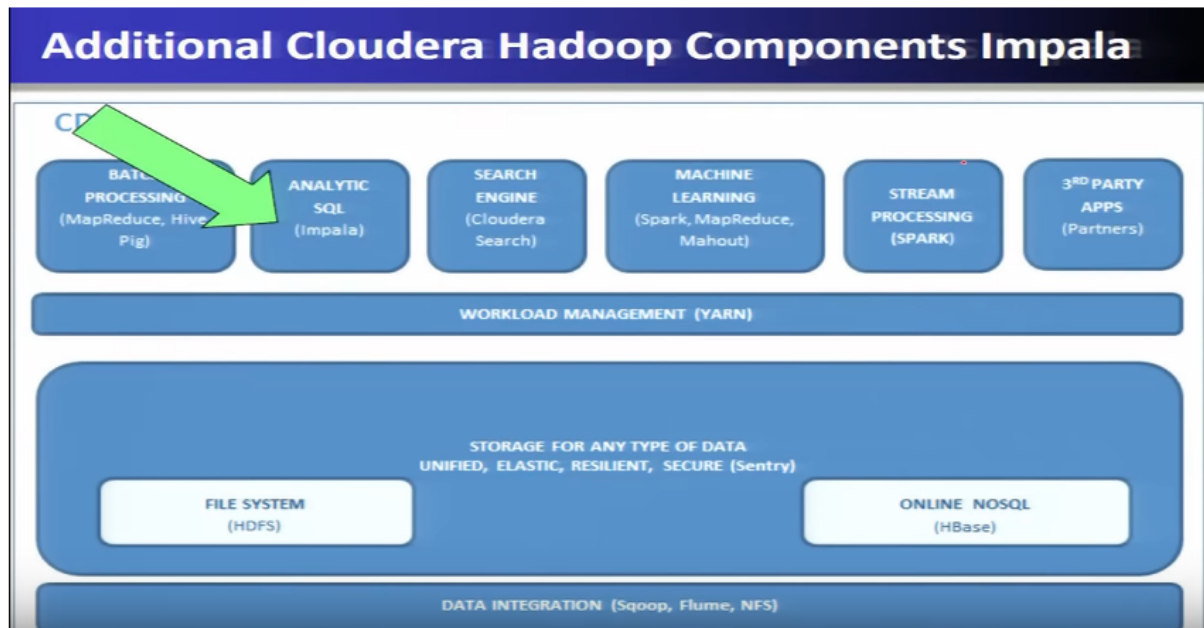
Flume

- Distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data — *data ingestion*
- It has a simple and very flexible architecture based on streaming data flows. It's quite robust and fault tolerant, and it's really tunable to enhance the reliability mechanisms, fail over, recovery, and all the other mechanisms that keep the cluster safe and reliable.
- It uses simple extensible data model that allows us to apply all kinds of online analytic applications.



Finally another application is called a flume, which is a distributed reliable available service, for efficiently collecting aggregating moving, a large amount of data into the, of the locks into the HDFS system hence, it is used for data injection, please use for data ingestion that is the flume system.

Refer Slide Time :(47:34)




Now, another component is called the Impala and that is nothing but an analytics engine, which is SQL beast engine.

Refer Slide Time :(47:48)

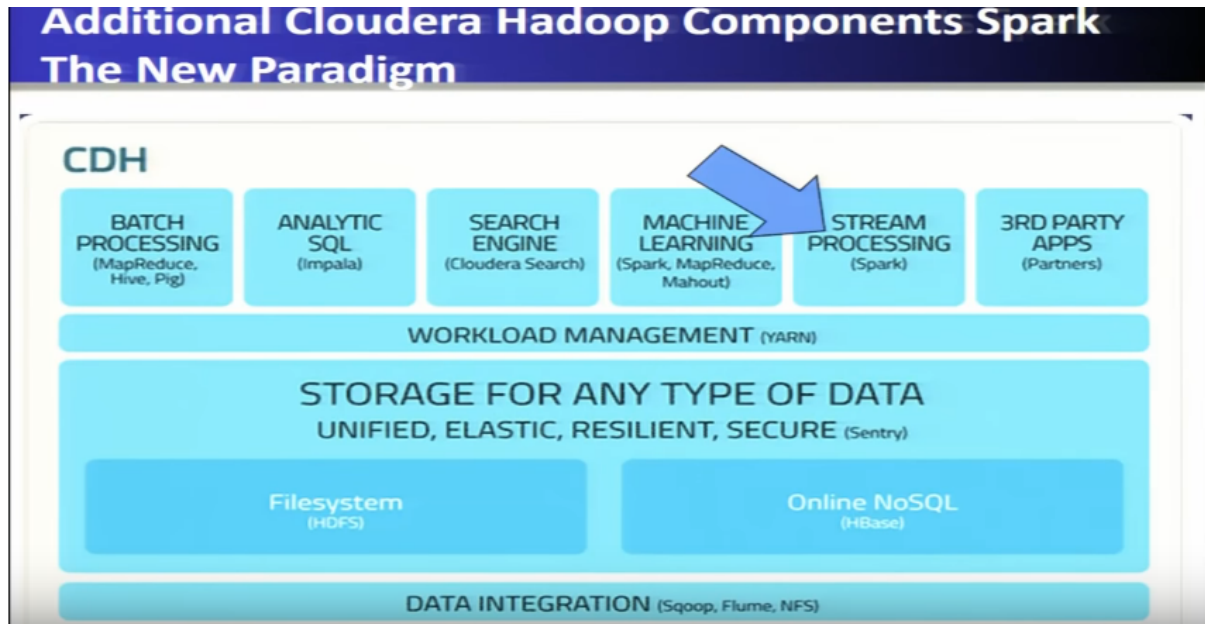
Impala

- Cloudera, Impala was designed specifically at Cloudera, and it's a query engine that runs on top of the Apache Hadoop. The project was officially announced at the end of 2012, and became a publicly available, open source distribution.
- Impala brings scalable parallel database technology to Hadoop and allows users to submit low latencies queries to the data that's stored within the HDFS or the Hbase without acquiring a ton of data movement and manipulation.
- Impala is integrated with Hadoop, and it works within the same power system, within the same format metadata, all the security and reliability resources and management workflows.
- It brings that scalable parallel database technology on top of the Hadoop. It actually allows us to submit SQL like queries at much faster speeds with a lot less latency.



So, its query engine runs on top of Apache Hadoop .so, Impala brings a scalable parallel database technology to the Hadoop and allows user to submit low latency queries within a particular system.

Refer Slide Time :(48:06)



Now, another component which Hadoop supports is the spark.

Refer Slide Time :(48:14)

Spark

- Apache Spark™ is a fast and general engine for large-scale data processing
- Spark is a scalable data analytics platform that incorporates primitives for in-memory computing and therefore, is allowing to exercise some different performance advantages over traditional Hadoop's cluster storage system approach. And it's implemented and supports something called Scala language, and provides unique environment for data processing.
- Spark is really great for more complex kinds of analytics, and it's great at supporting machine learning libraries.
- It is yet again another open source computing frame work and it was originally developed at MP labs at the University of California Berkeley and it was later donated to the Apache software foundation where it remains today as well.

which is a spark, is a fast general-purpose engine for a large-scale data processing. So, spark is a scalable data analytics platform and it supports the in-memory computation, I enhance its performance is much

better why because it supports in-memory computation. So, if Sparkle supports complex kind of analytics which is called a big data analytics and hence it is of great interest in today's the big data computation spark engine.

Refer Slide Time :(48:44)

Spark Benefits

- In contrast to Hadoop's two stage disk based MapReduce paradigm Multi-stage in-memory primitives provides performance up to 100 times faster for certain applications.
- Allows user programs to load data into a cluster's memory and query it repeatedly
- Spark is really well suited for these machined learning kinds of applications that often times have iterative sorting in memory kinds of computation.
- Spark requires a cluster management and a distributed storage system. So for the cluster management, Spark supports standalone native Spark clusters, or you can actually run Spark on top of a Hadoop yarn, or via patching mesas.
- For distributor storage, Spark can interface with any of the variety of storage systems, including the HDFS, Amazon S3.

So, the spark engine, Spark has the performance 100 times faster, than for applications if spark is used.

Refer Slide Time :(49:07)

Conclusion

- In this lecture, we have discussed the specific components and basic processes of the Hadoop architecture, software stack, and execution environment.

So, conclusion in this lecture, we have discussed the components and the applications of Hadoop ecosystem and also we have covered about its execution environment.

Thank you.