Haddop – open source software platform for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware.

Distributed Storage – Big Data, Terabytes of info. Adding more and more storage spaces as one single file system. It is redundant.

Distributed Processing – can distribute the data. Do it all in the Parallel Manner. Get through all the data very quickly. Parallel Computing.

Large Dataset

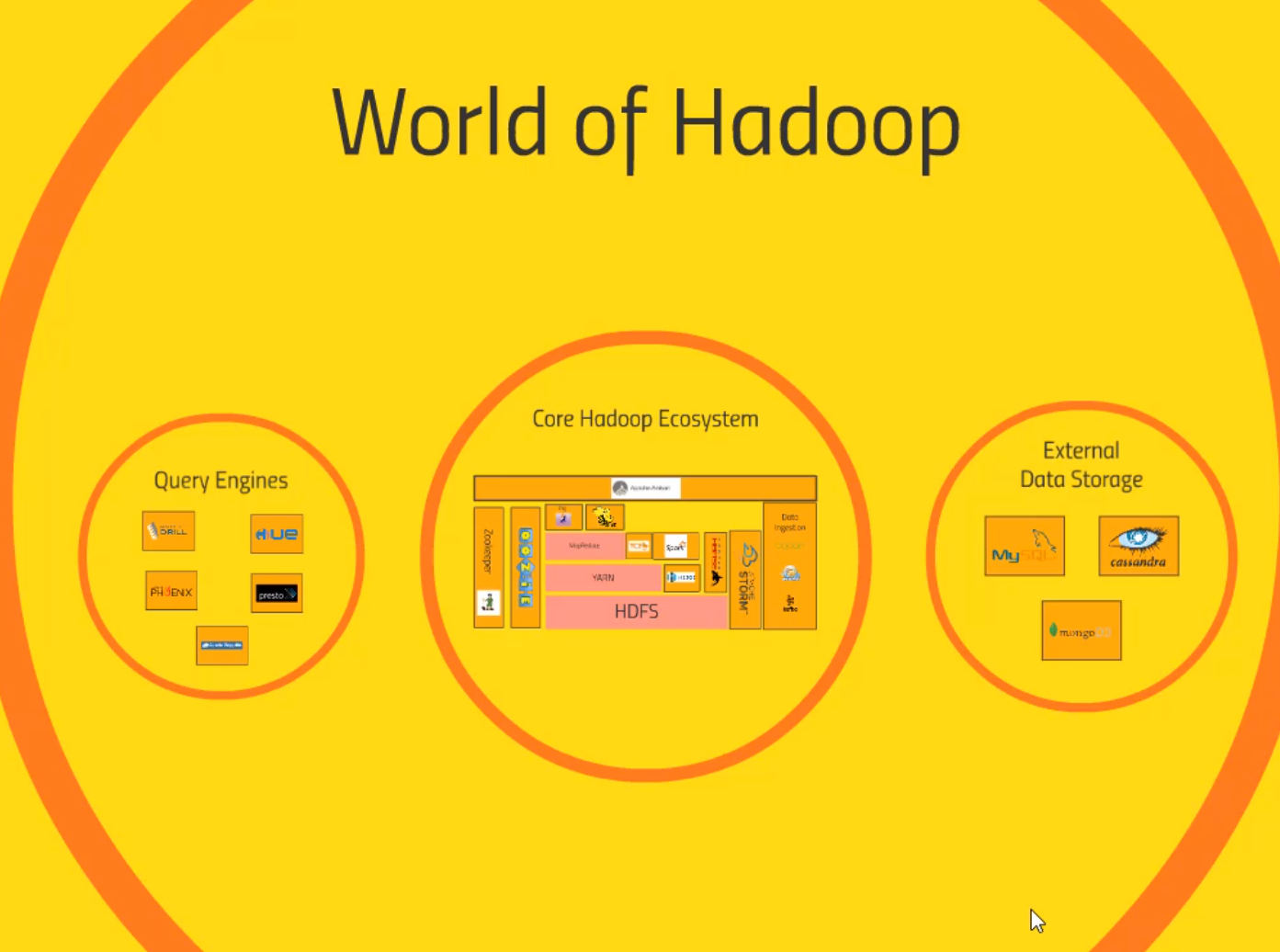
Commodity hardware – AWS, throw into Hadoop cluster and go forward with it.

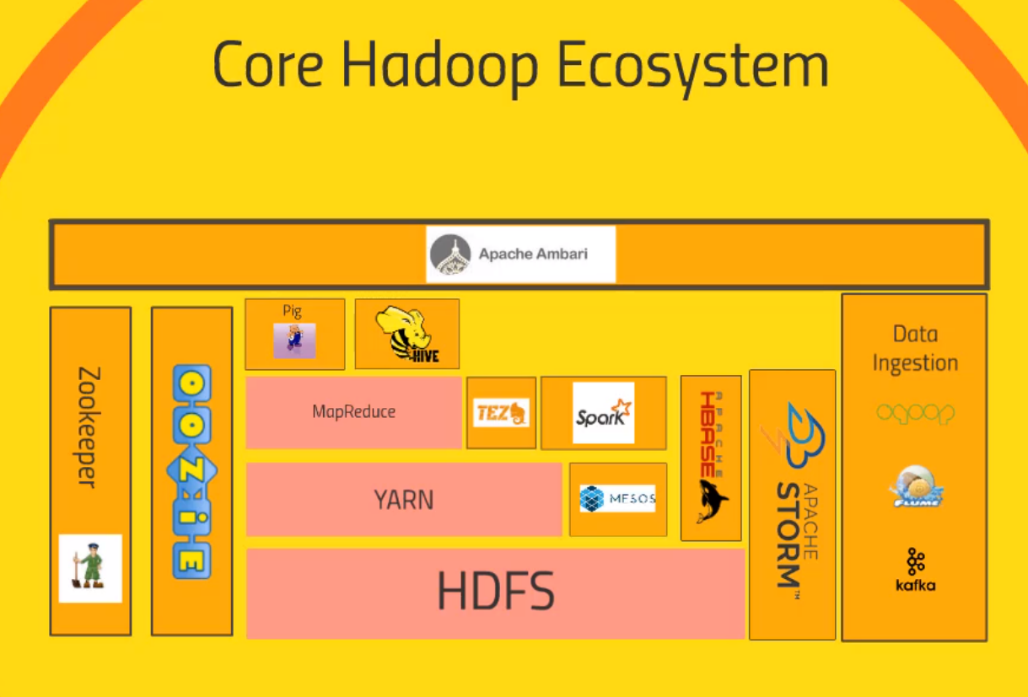
GFS and Map Reduce from Google

Hadoop has evolved.

Why Hadoop?

* Data is too darn big – terabytes per day
* Vertical Scaling doesn’t cut it
  + Disk seek times – Seeking in parallel
  + Hardware Failures – Hadoop has redundancy. Have back up copies
  + Processing times – Clusters means parallel processing
* Horizontal scaling is linear – more PC means more processing
* Hadoop: It’s not just for batch processing anymore.
* Interact with Queries.

**Overview of Hadoop Ecosystem**



Hadoop Distributed File System – HDFS

* Distribute the storage of Big Data across the cluster of computers. All of the small hard drives in a giant File system
* Maintain redundant copy of the data. It has back up

Yet another resource negotiator – YARN

* Data Processing part of Hadoop
* System that manages the resources on your Computing Cluster
* Decides what get to run what tasks when, what nodes are available and etc.
* Heart beat that keeps your Clusters going

MapReduce

* Programming model that allow you to process your data across your entire clusters
* Mappers and Reducers, Scripts that you will write
* Mappers -Transform your data in parallel across your entire computing cluster in an efficient manner.
* Reducers – Aggregate your data together

Pig (Sit on top of MapReduce)

* Scripting language something like SQL Syntax
* High Level Programming API that allow you to write simple scripts that does complex queries without needing to write Python or Java
* Pig will transform the script that will write understandable script on MapReduce to get the answer what you want

Hive – Similar to Pig

* More directly like a database
* Takes in SQL language scripts
* SQL Database
* Connect through a Shell client, executes SQL queries on the data stored in your Cluster.
* Familiar with SQL, Hive might be your data platforms

Apache Ambari – Sits on top of everything. – HortonWorks, Cloudera

* View of your cluster, how much resources, what are running
* Execute Hive queries
* Import data into Hive
* Have a view into the state of your cluster.

Mesos – Isn’t part of Hadoop

* Alternative to YARN
* Resource Negotiator
* Basically solve the problems in Different way.

Spark – Most exciting technology

* Same level of MapReduce
* Run Queries on your data. Like MapReduce need Programming. Python, Java or Scala. Scala preferred.
* Extremely fast, active development. Exciting Technology.
* Process data on Hadoop cluster
* Handle SQL, Machine Learning across entire Cluster, Handle Streaming data in Real time.

Tez – Similar to Spark

* Same technique as Spark, DAG
* Produce more optimal plans to executing your queries
* Conjunction with Hive to accelerate it.
* Can also sit on top of Hive.

Apache HBASE

* NoSQL database
* Really fast database meant for very large transaction rates
* Hitting from Website, app, OLTP type of transaction
* Expose the data that is stored in your cluster. Transformed in some way by spark or mapreduce.
* Expose the results to other systems

Apache Storm

* A way of processing streaming data. Streaming data from sensors.
* Process in real time
* Spark Streaming solve the same problem
* Made for processing streaming data quickly in real time. No need batch. Update your ML model, Transform data into your database in real time

OOZIE

* Way of scheduling jobs on your clusters
* Tasks that involve many steps
* Into Jobs that can be run on some sort of schedules.
* OOZIE can manage that all for you and to run reliably

Zookeeper

* Technology for coordinating everything, keeping track of which nodes are up which nodes are down.
* Keep track of shared states across your clusters that different application can use.
* Many application rely on zookeeper to maintain reliable and consistent performance of the cluster even if a node randomly goes down
* Keep track of Who is the master node, whose down or whose up

Data Ingestion – How the data from other sources comes into Hadoop

SCOOP

* Tying your Hadoop into relational database.
* Connector between Hadoop and legacy databases

Flume

* Transforming Web logs at a very large scale
* Web servers, listens to the web logs, coming in from those web servers in real time and into your clusters by processing by storm or spark streaming

Kafka

* General purpose
* Collect data of any sorts, cluster of PCs and broadcast it into the Hadoop Cluster

External Data Storage

MySQL

* Any SQL DataBase, Integrate with Hadoop, Export into MYSQL. JDBC, ODBC database.

Cassandra and MongoDB

* Like HBase, good choices for exposing your data for real time usage for web application.
* Sitting between real time application and your cluster.
* Vending simple key value data stores at very large transaction database.

Query Engines – SQL, Extracting meaning from your cluster.

Apache Drill – Allows you to SQL queries that will work across NOSQL Database, Talk to hbase, or Cassandra or mongodb.

Hue – Create queries, works well with HIVE, Cloudera, sits on top of Ambari.

Apache Phoenix – Similar to Drill, SQL queries, across the range of datastorage technologies. Make your not SQL Hadoop datastore look a lot like relational datastores and relation guarantee

Presto , Apache Zeppelin – Notebook type of approach to your UI. And interact with your clusters.