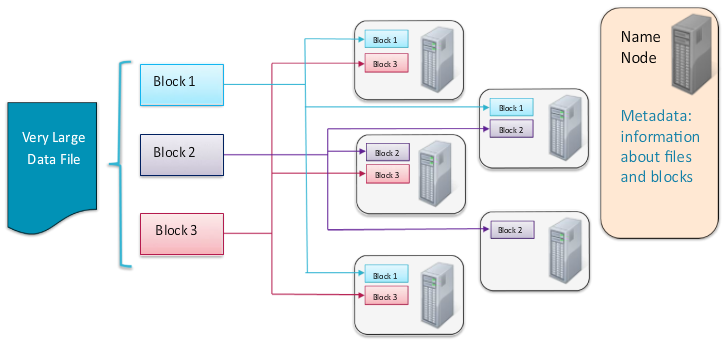
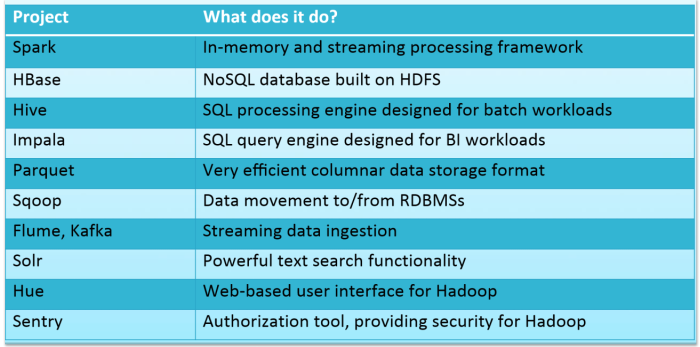
**Hadoop**

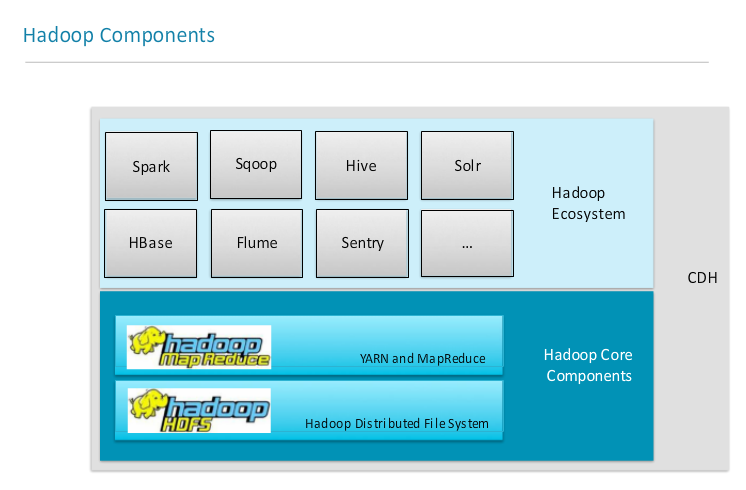
* Hadoop is a distributed data storage and processing platform.
* Core Hadoop is a File system and processing framework
* HDFS – Hadoop Distributed File system – stores data on the cluster
  + Any type of file can be stored
  + Data is split into chunks and replicated as it is written
  + Files in HDFS are “write-only”
    - Appends are permitted
    - No random writes are allowed
* Basic concept of HDFS
  + HDFS is a filesystem written in Java
  + Sits on top of a native filesystem
  + Supports efficient processing with MapReduce, Spark and other frameworks
  + Data files are split into blocks and distributed to data nodes
  + Each block is replicated on multiple nodes
  + NameNode stores the metadata info about the files and blocks

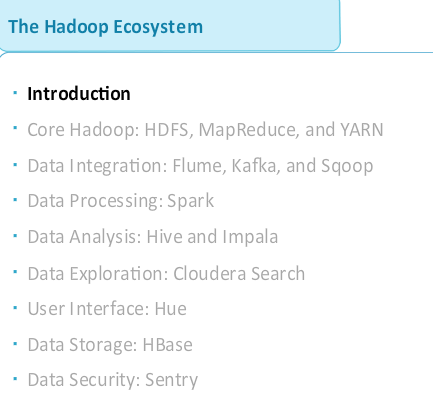


* MapReduce – process data on the cluster
  + A distributed processing framework / programming model
  + Neither platform – nor language – specific
  + Record oriented data processing (key and value)
* YARN – Yet Another Resource Negotiator – Schedules work on cluster
  + Manages the processing resources of the Hadoop cluster
  + Schedules jobs
  + Runs processing frameworks
* Motivation for YARN
  + Multiple frameworks may exist on a single cluster
  + Each framework competes for compute and memory resources on the nodes
  + YARN allocates resources to different frameworks based on demand and on system admin settings
* Adding node adds capacity. Node failure is inevitable
  + System continues to function
  + Master re-assigns work to different node
  + Data replication happens
  + Nodes which recovers rejoins the cluster automatically
* Hadoop ecosystem – simplifies distributed computing so programmers can focus on the application

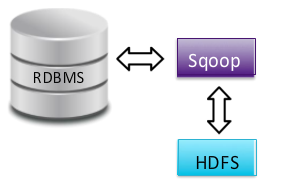
Example



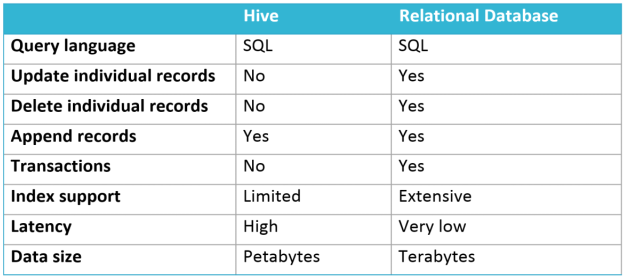




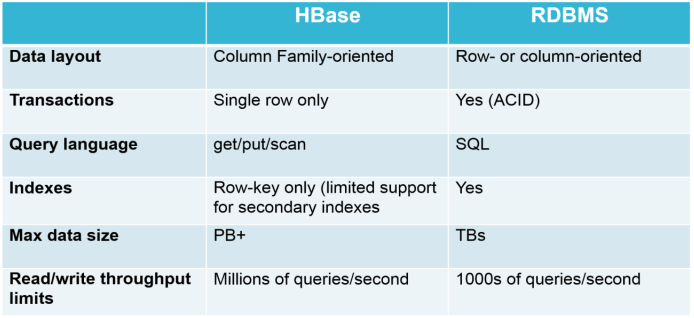
* Flume and Kafka – tools for ingesting event data into Hadoop as the data is being generated (eg. Files, logs, sysout)
  + Flume is typically easier to configure but Kafka provides more functionality
  + Flume generally provides a path from a data source to HDFS or to a streaming framework like Spark
  + Kafka uses a “Publish/Subscribe” model - allows data to be consumed by diff systems, including writing to HDFS
* Sqoop – rapidly moves large amount data from RDBMS to HDFS

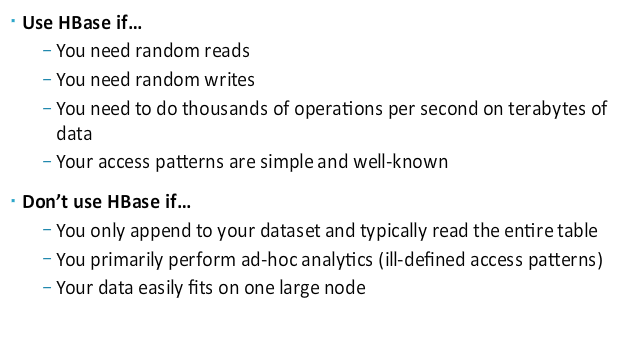


* + Uses JDBC to connect to the database
* Spark – large scale data processing engine
  + Provides the ability to process data as they are being generated
  + Faster than MapReduce
  + Code can be written using Java, Python or Scala
  + Well suited to iterative processing algorithms
* Hive – Abstraction layer on top of Hadoop
  + Uses SQL language called HiveQL
  + Uses MapReduce or Spark to process data
  + JDBC and ODBC connection are available
  + Data can be loaded before the table is defined (Do not need to know the data’s structure prior to loading it)
  + Suited for structured data



* Impala – high performance SQL engine
  + Supports a dialect of SQL similar to Hive
* Impala or Hive
  + Impala is best suited for ad-hoc analytics and situations where multiple people querying the cluster simultaneously
  + Hive is suited for batch processing
* Couldera Search – provides interactive full-text search for data in Hadoop cluster
  + Supports real time and batch intdexing
  + Dynamic web-based dashboard Search interface with Hue
  + Allows non-technical users to access data
* Hue – provides web front-end to a Hadoop
  + Upload, Browse data, Query tables in Hive and Impala, Search, etc.
  + Provides access control to users before accessing the data
* HBase – NoSQL distributed database
  + Stores data into HDFS





* Sentry – provides fine grained access control(authorization) to Hadoop ecosystem components (Hive, Impala)
  + Combined with Kerberos authentication, Sentry authorization provides a complete cluster security solution

Hadoop Architecture

* Hadoop Cluster – is a group of computers working together
* Node – is an individual computer in the cluster
* Daemon – is a program running on a node
* Master Node – manage the work
  + Daemons running on master node ensures that the entire cluster works
  + A failed daemon could cause the entire cluster to become unstable
* Worker Node – do the work
  + Daemons running on worker node handle actual data processing