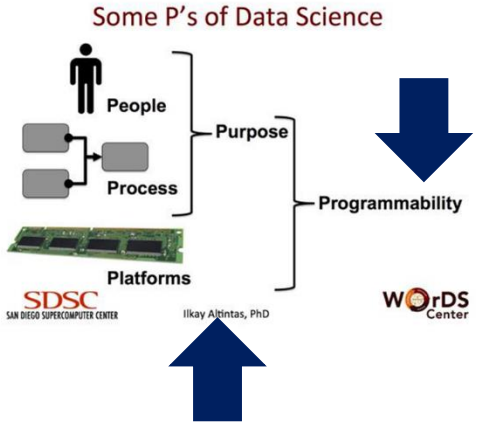
Computation Concepts and Hadoop environment (Platform and Programability)

Introduction to Big Data

Universidad
Francisco de Vitoria
UFV Madrid

Index

- Big data stack
- Virtualization
- What is a DFS (Distributed File System) ?
- Computation and scalability
- Hadoop ecosystem:
    - Introduction
    - Ecosystem
    - HDFS
- Map Reduce
    - YARN
    - When Hadoop is not applicable?



Big data stack

- Like any **important data architecture**, you should design a model that takes a holistic look at how all the elements need to come together
- The environment must include considerations for hardware, infrastructure software, operational software, management software, well-defined application programming interfaces (APIs), and even software developer tools
- Your architecture will have to be able to address all the foundational requirements :
  - **Capture**
  - **Integrate**
  - **Organize**
  - **Analyze**
  - **Act**

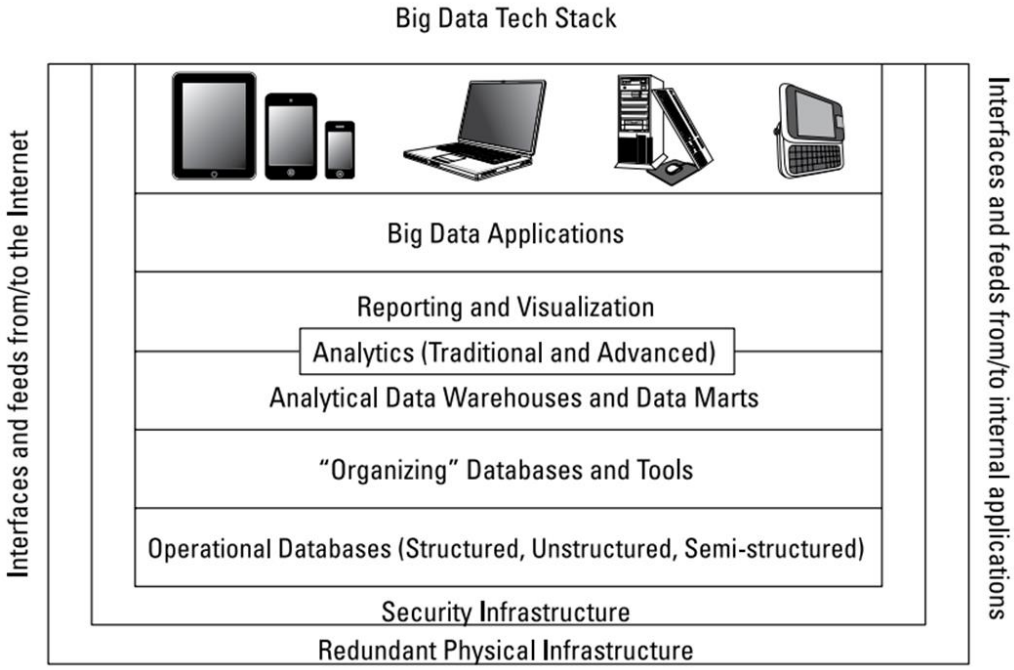


Figure 4-1: The big data technology stack.

Big data stack

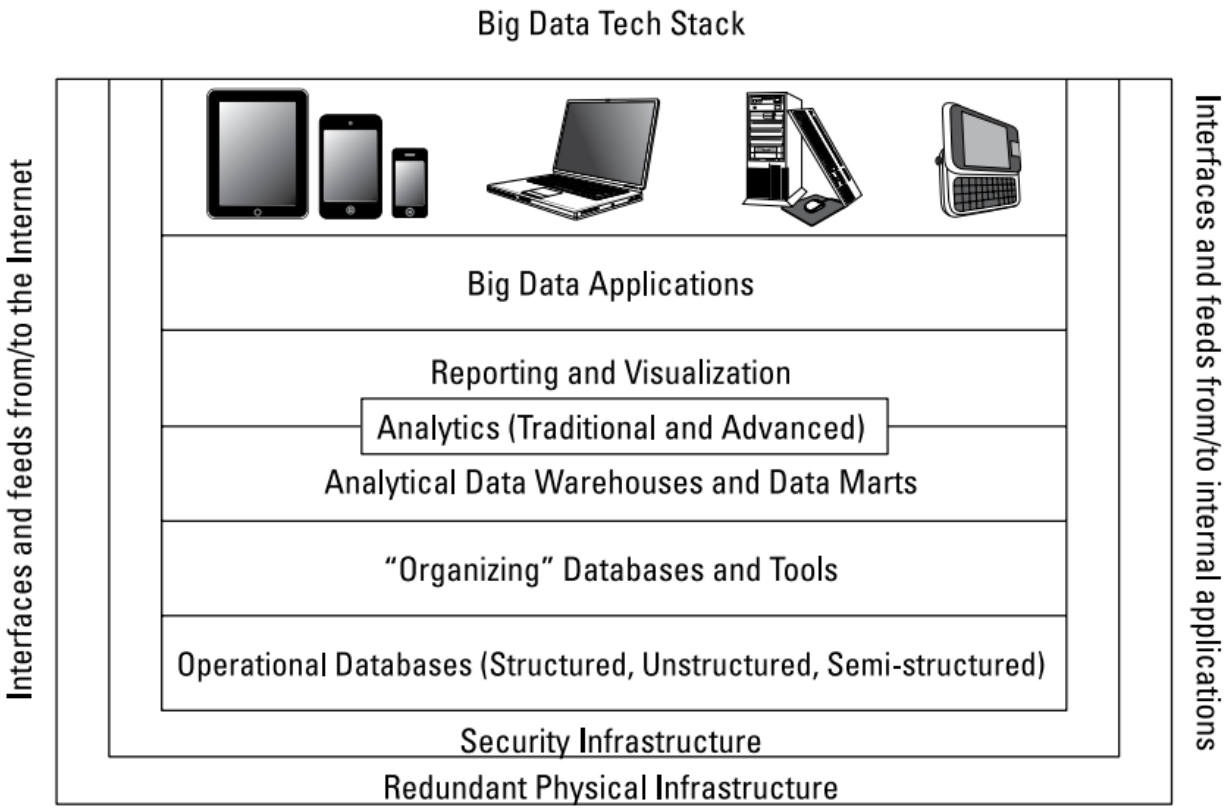


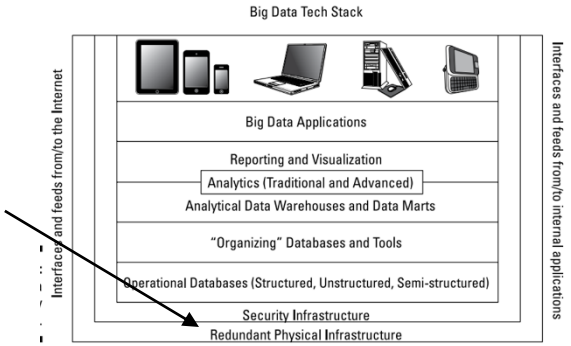
Figure 4-1: The big data technology stack.

Big data stack

- **Layer 0:** At the lowest level of the stack is the **physical infrastructure** — the hardware, network, and so on. As you start to think about your big data implementation, it is important to have some overarching principles that you can apply to the approach. A prioritized list of these principles should include statements about the following:
  - **Performance;** How responsive. Performance, also called latency, is often measured end to end, based on a single transaction or query request. Very fast (high-performance, low latency) infrastructures tend to be very expensive
  - **Availability**: Do you need a 100 percent uptime guarantee of service? Highly available tend to be very expensive
  - **Scalability**: How big does your infrastructure need to be? Typically, you need to decide what you need and then add a little more scale for unexpected challenges
  - **Flexibility:** How quickly can you add more resources to the infrastructure?
  - **Cost**: What can you afford?

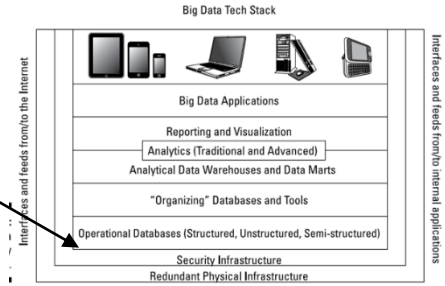
Most big data implementations need to be highly available, so the networks, servers, and physical storage must be both resilient and redundant.

Resiliency and redundancy are interrelated



Big data stack

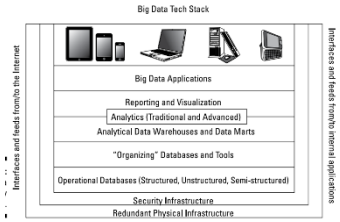
- **Layer 1: Security and privacy** requirements for big data are similar to the requirements for conventional data environments. The security requirements have to be closely aligned to specific business needs
    - **Data access**: The data should be available only to those who have a legitimate business need for examining or interacting with it
    - **Application access**; Most application programming interfaces (APIs) offer protection from unauthorized usage or access
    - **Data encryption**; Data encryption is the most challenging aspect of security in a big data environment. **With the volume, velocity, and varieties associated with big data, this problem is exacerbated.**
    - **Threat detection**: It is therefore important that organizations take a multiperimeter approach to security

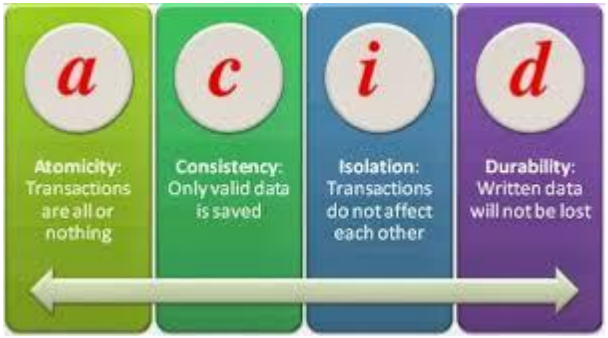


Interfaces feeds from/to applications and the internet: The next level in the stack is the interfaces that provide bidirectional access to all the components of the stack

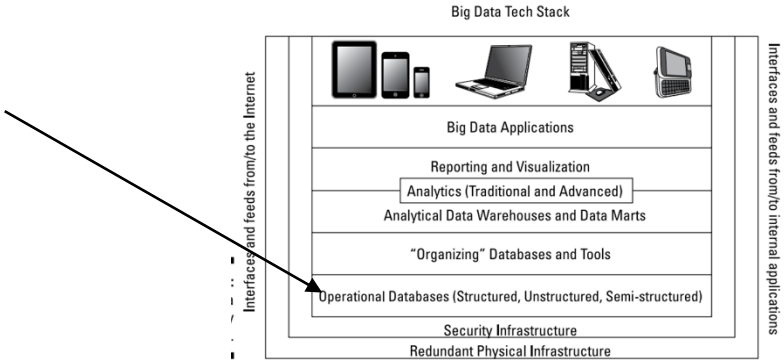
Big data stack

- **Layer 2: Operational databases** At the core of any big data environment are the **database engines** containing the collections of data elements relevant to your business. It is possible to use relational database management systems (RDBMSs) for all your big data implementations, it is not practical to do so because of performance, scale, or even cost
- It is very important to understand what types of data can be manipulated by the database and whether it supports true transactional behavior. Database designers describe this behavior with the acronym **ACID**. It stands for
  - **Atomicity**: A transaction is **"all or nothing"** when it is atomic. If any part of the transaction or the underlying system fails, the entire transaction fails.
  - **Consistency**: **Only transactions with valid data** will be performed on the database. If the data is corrupt or improper, the transaction will not complete and the data will not be written to the database.
  - **Isolation**: Multiple, simultaneous transactions will not interfere with each other. All valid transactions will execute until completed and in the order they were submitted for processing.
  - **Durability**: After the data from the transaction is written to the database, it stays there "forever."



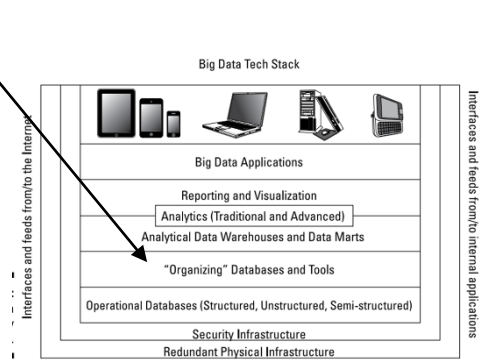


Big data stack



Big data stack

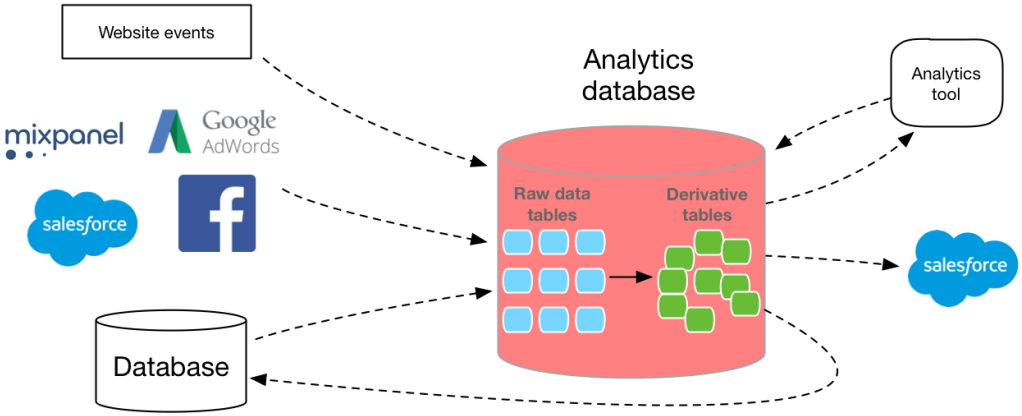
- **Layer 3: *Organizing data services and tools*** capture, validate, and assemble various big data elements into contextually relevant collections. Are an ecosystem of tools and technologies that can be used to gather and assemble data in preparation for further processing.
- Technologies in this layer include the following:
  - **A distributed file system**: Necessary to accommodate the decomposition of data streams and to provide scale and storage capacity
  - **Serialization services:** Serialization is the process of translating data structures or objects state into binary or textual form to transport the data over network or to store on some persistent storage. Necessary for persistent data storage and multilanguage remote procedure calls (RPCs)
  - **Coordination services**: Necessary for building distributed applications (locking and so on)
  - **Extract, transform, and load (ETL) tools**: Necessary for the loading and conversion of structured and unstructured data into Hadoop
  - **Workflow services**: Necessary for scheduling jobs and providing a structure for synchronizing process elements across layers

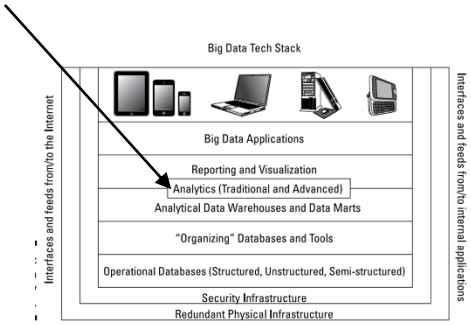


Big data stack

- **Layer 4: Analytical data warehouses** contain normalized data gathered from a variety of sources and assembled to facilitate analysis of the business
- **Data warehouses and data marts** (A data mart is a subset of a data warehouse oriented to a specific business line. Data marts contain repositories of summarized data collected for analysis on a specific section or unit within an organization, for example, the sales department) contain normalized data gathered from a variety of sources and assembled to facilitate analysis of the business.
- As the organization of the data and its readiness for analysis are key, most data warehouse implementations are kept current via batch processing, but real time data warehouses will likely be required

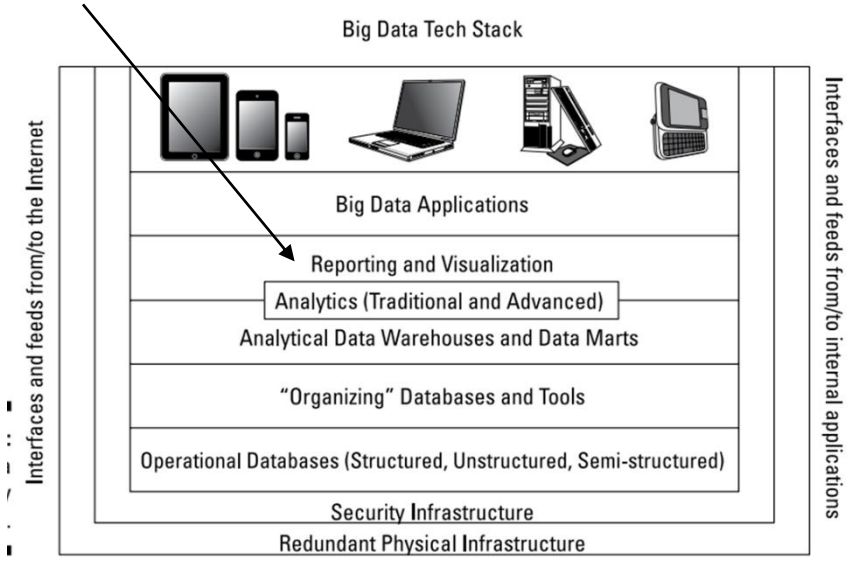
Data warehouses and marts simplify the creation of reports and the visualization of disparate data items





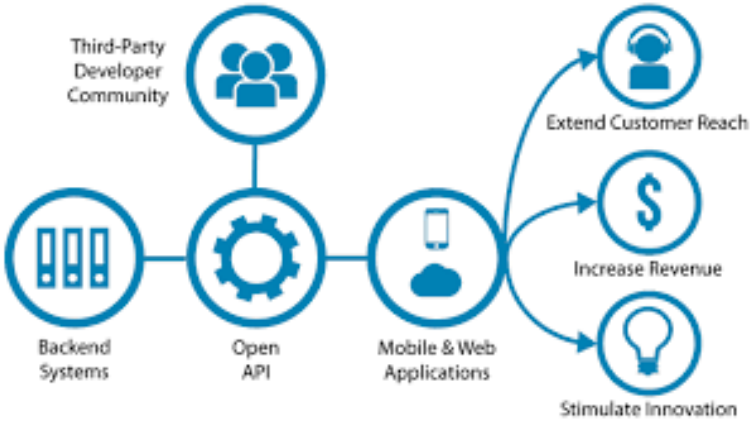
Big data stack

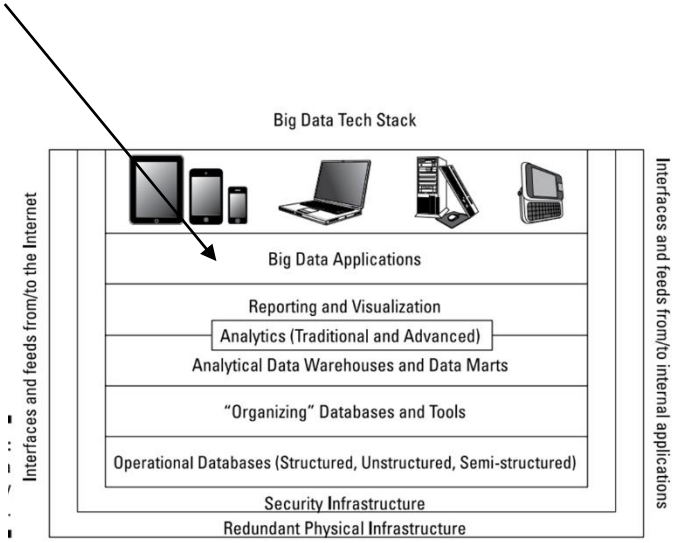
- **Big data analytics**: Existing analytics tools and techniques will be very helpful in making sense of big data. However, there is a catch. The algorithms that are part of these tools have to be able to work with large amounts of potentially real-time and disparate data. Three classes of tools:
  - **Reporting and Dashboards**
  - **Visualization**: a dynamic evolution of the dashboards
  - **Analytics and advance analytics**: giving predictions, recommendations, inferences, trends, ….



Big data stack

- **Big data applications**: Custom and third-party applications offer an alternative method of sharing and examining big data sources
  - These applications **are either horizontal**, in that they address problems that are common across industries**, or vertical**, in that they are intended to help solve an industry-specific problem
  - Most business applications wanting to leverage big data will need to subscribe to **APIs** across the entire stack. It may be necessary to process raw data from the low-level data stores and combine the raw data with synthesized output from the warehouses





Virtualization

- **Virtualization is** the process of **creating a software-based, or virtual,** representation of something, such as virtual applications, servers, storage and networks
- Virtualization **separates resources and services from the underlying physical delivery environment**, enabling you to create many virtual systems within a single physical system
- **Rather than assigning a dedicated set of physical resources to each set of tasks,** a pooled set of virtual resources can be quickly allocated as needed across all workloads

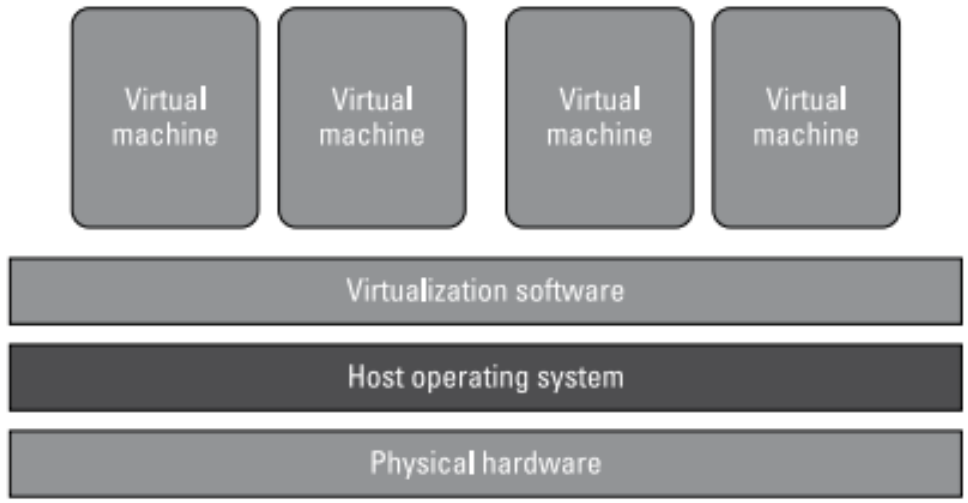


Figure 5-1: Using virtualization software to create several virtual systems within a single physical system.

Virtualization

- Using a distributed set of physical resources, such as servers, in a more flexible and efficient way delivers significant benefits in terms of cost savings and improvements in productivity.
- The practice has several benefits, including the following:
  - **Virtualization of physical resources** (such as servers, storage, and networks) enables substantial improvement in the utilization of these resources.
  - Virtualization enables **improved control over the usage and performance** of your IT resources.
  - Virtualization can provide a level of automation and standardization to optimize your computing environment.
  - Virtualization provides a foundation for cloud computing.

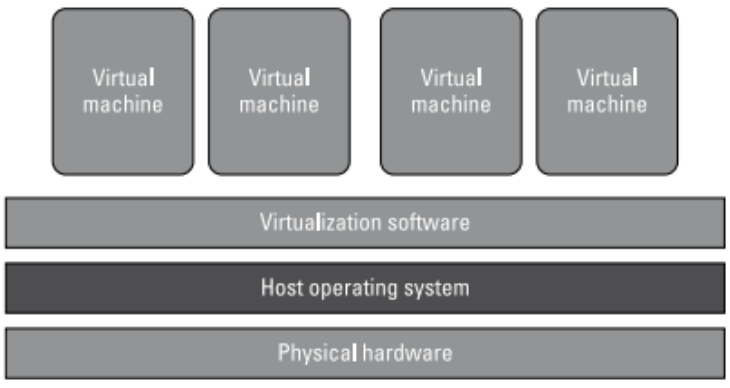


Figure 5-1: Using virtualization software to create several virtual systems within a single physical system.

Virtualization

Virtualization has three characteristics that support the scalability and operating efficiency required for big data environments

- **Partitioning:** In virtualization, many applications and operating systems are supported in a single physical system by partitioning (separating) the available resources

- **Isolation:** Each virtual machine is isolated from its host physical system and other virtualized machines. Because of this isolation, if one virtual instance crashes, the other virtual machines and the host system aren't affected. In addition, data isn't shared between one virtual instance and another

- **Encapsulation:** A virtual machine can be represented (and even stored) as a single file, so you can identify it easily based on the services it provides

Virtualization

- **Server virtualization:** In server virtualization, one physical server is partitioned into multiple virtual servers. The hardware and resources of a machine — including the random-access memory (RAM), CPU, hard drive, and network controller — can be virtualized (logically split) into a series of virtual machines that each runs its own applications and operating system. A virtual machine (VM) is a software representation of a physical machine that can execute or perform the same functions as the physical machine
- **Processor virtualization** helps to optimize the processor and maximize performance. Memory virtualization decouples memory from the servers.

- **Application infrastructure virtualization** provides an efficient way to manage applications in context with customer demand. The application is encapsulated in a way that removes its dependencies from the underlying physical computer system. This helps to improve the overall manageability and portability of the application

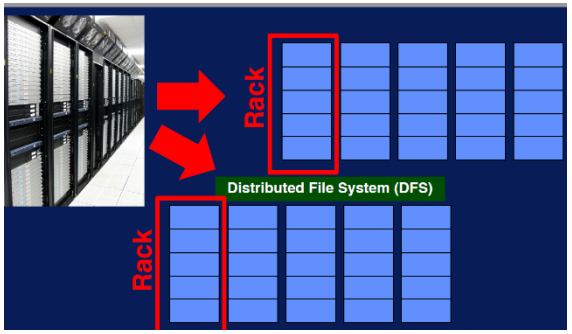
- **Network virtualization** — software-defined networking — provides an efficient way to use networking as a pool of connection resources. Networks are virtualized in a similar fashion to other physical technologies.

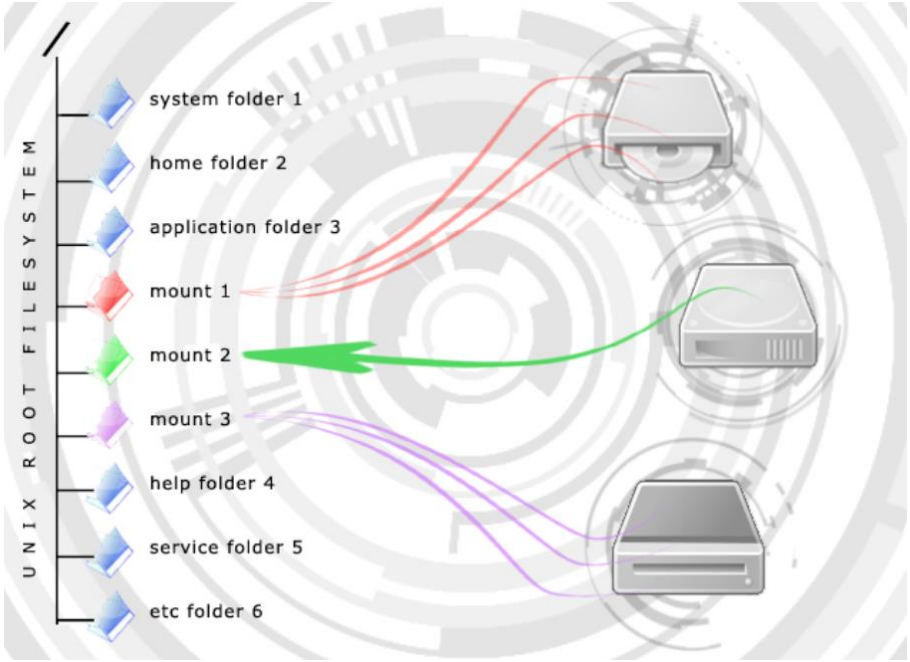
- **Data virtualization** can be used to create a platform for dynamic linked data services. This allows data to be easily searched and linked through a unified reference source

DFS and Virtualization

DFS Distributed File System

- **Why?**
  - Access later to the results of a process
  - Save big amounts of data
  - Allow Access to multiple processes

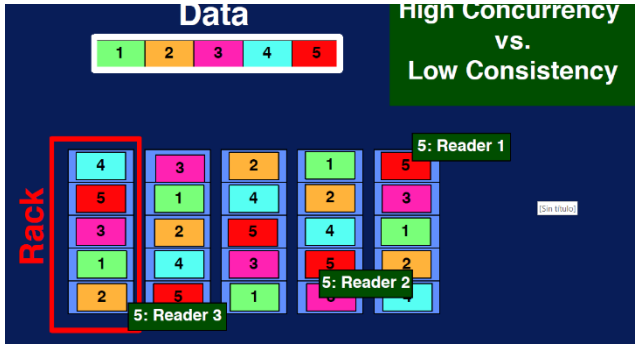




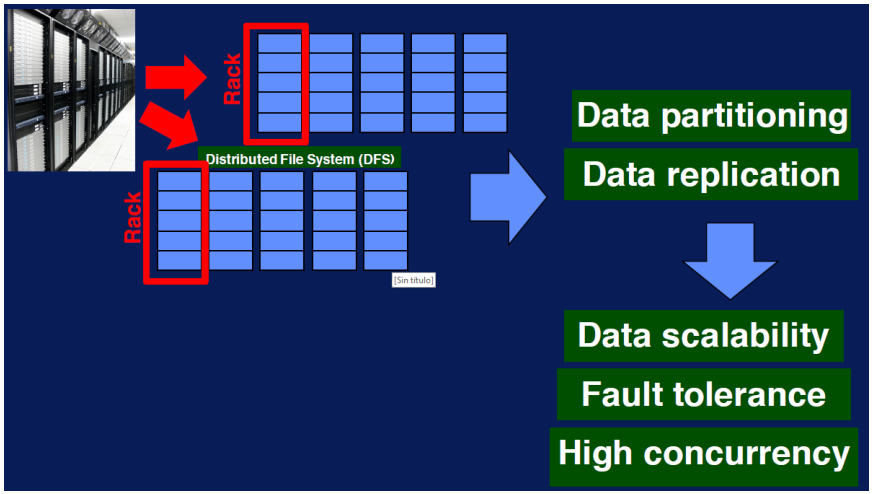
What is a DFS?

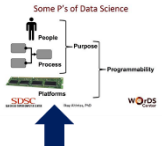
DFS Distributed File System

- **Consequences**
  - **High concurrency**
    - Data partition
    - Data replication
  - **Data scalability**
  - **Fault tolerance**



implies





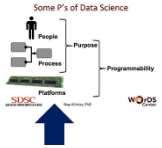
Computation and scalability

- **Commodity cluster (underlaying hardware)**
  - **Data parallelism**
  - **Commodity cluster architecture and fault tolerance**

Affordable costs. Reduce computation costs

Non specialise

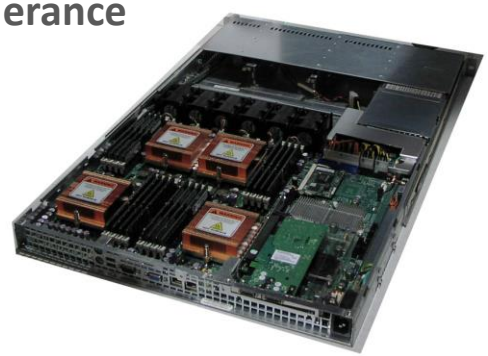
Allows distributed computation over internet



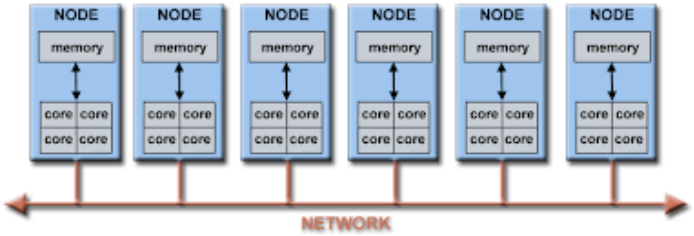
Computation and scalability

- **Commodity cluster (underlaying hardware)**
  - **Data parallelism**
  - **Commodity cluster architecture and fault tolerance**

One computer



Parellel computers



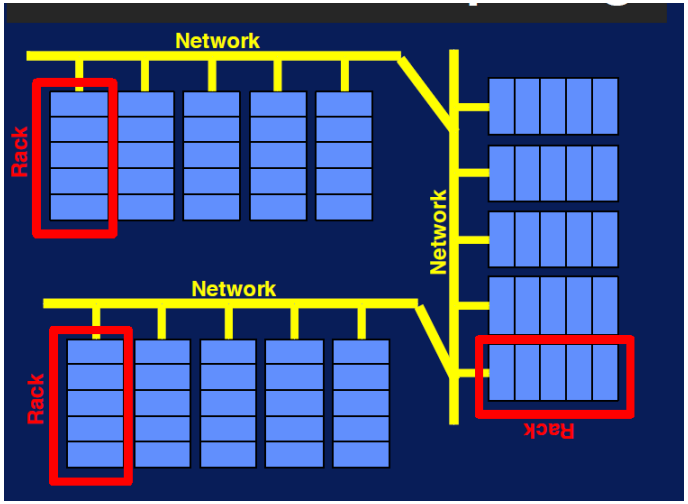
Computation and scalability

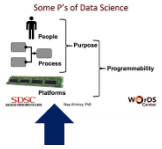
- **Commodity cluster (underlaying hardware)**
  - **Data parallelism**
  - **Commodity cluster architecture and fault tolerance**

Allows parallel computation

Redundant data warehousing against failure

Restart parallel processing





Hadoop: Why?

Allows scalability

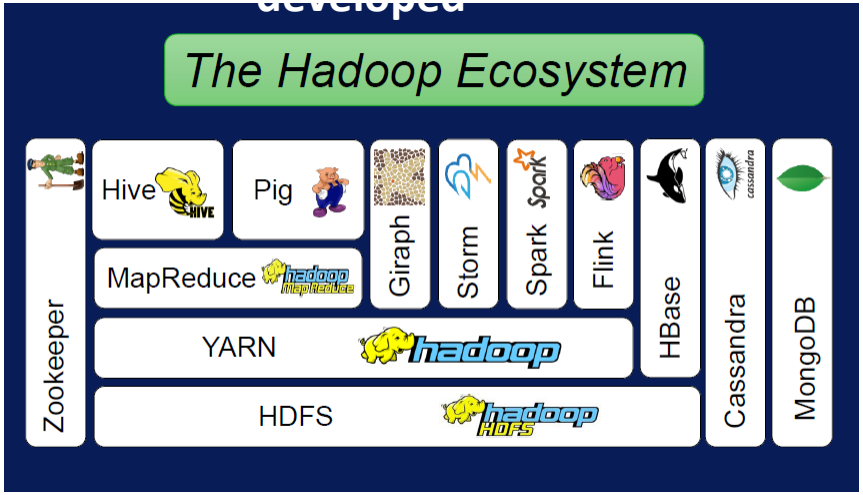
Allows fault tolerance

Allows data variety

Enables compatible environment

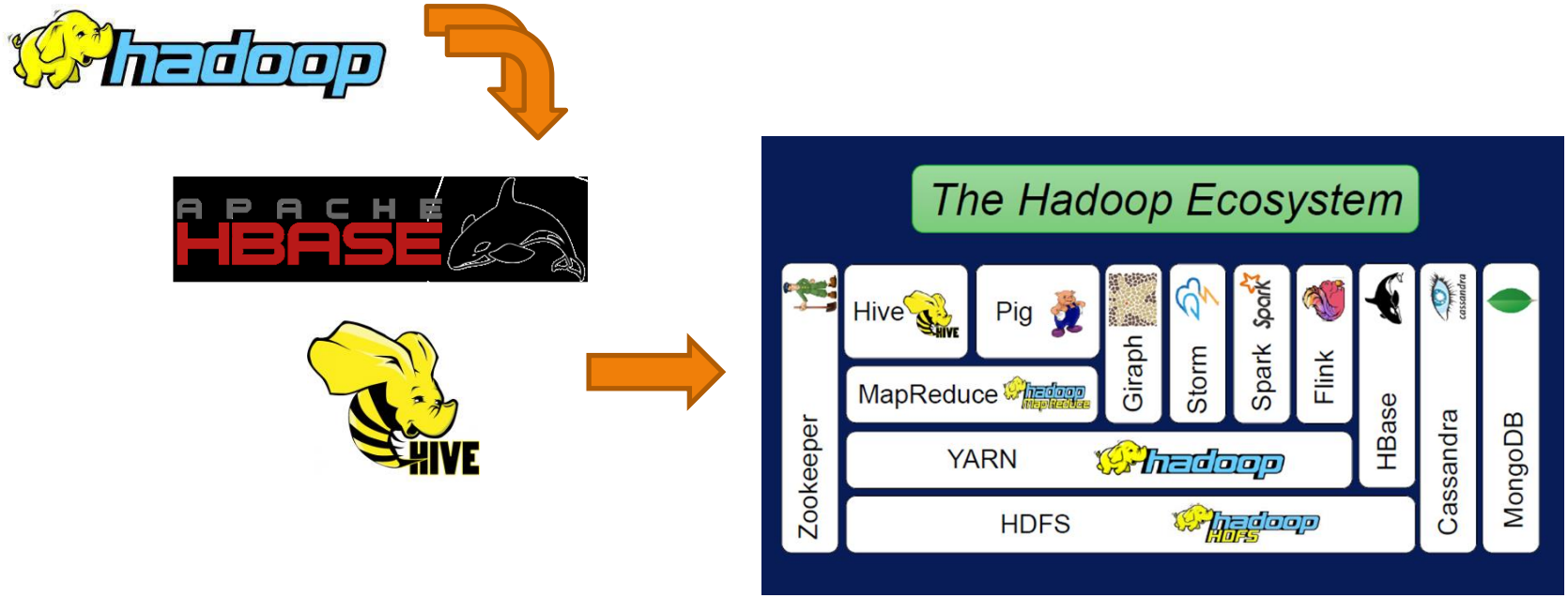
Creates value with a big support community and with multiple applications developed

AI the applications are for free and opensource



Hadoop: Ecosystem

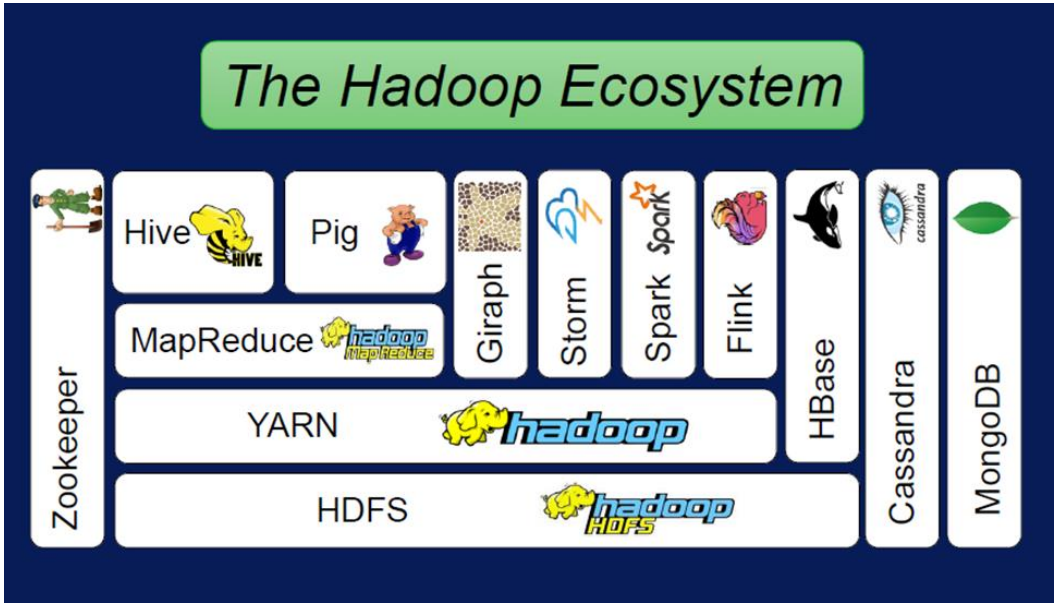
- **Created by Yahoo in 2005**
- **New work environments have been added**
- **Now there are more tan 100**



Hadoop: Ecosystem

Higher leel of interactivity

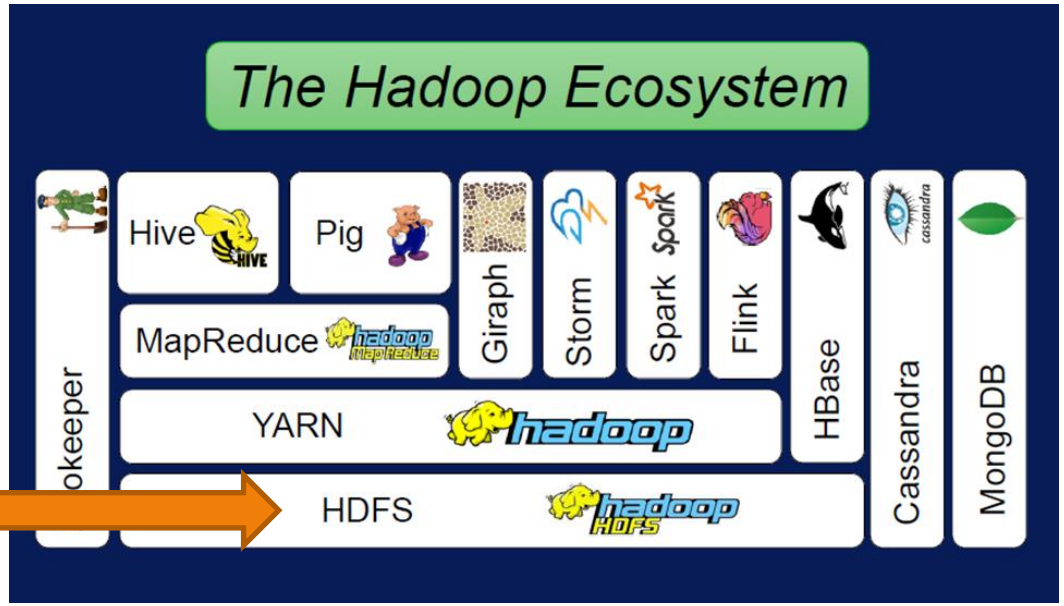
Data storage and activity scheduling



Hadoop: Ecosystem

Scalable data warehouse based on DFS

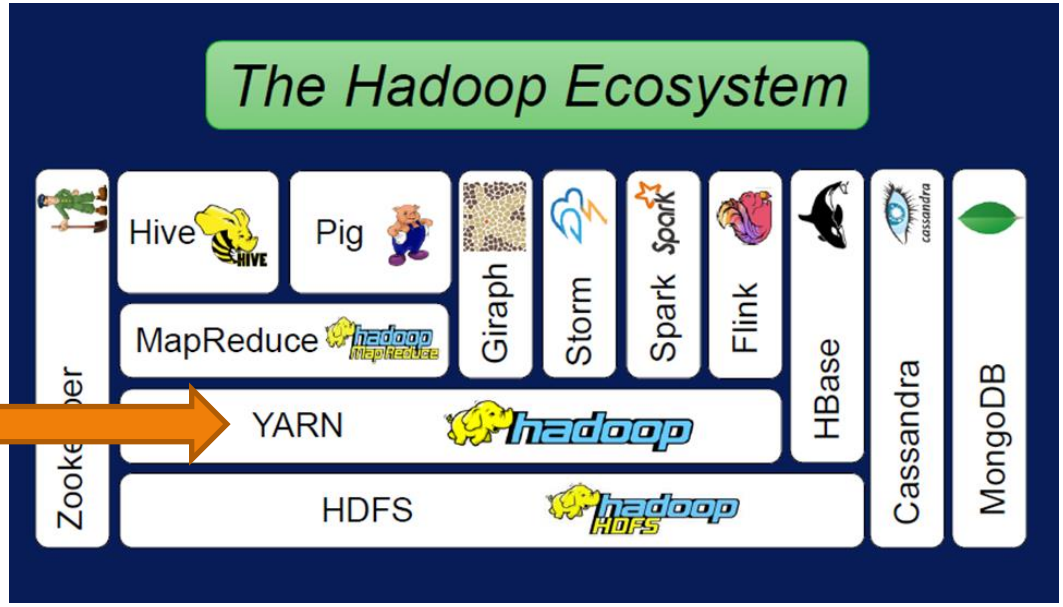
Fault tolerance



Hadoop: Ecosystem

Flexible Job scheduling

Resource management

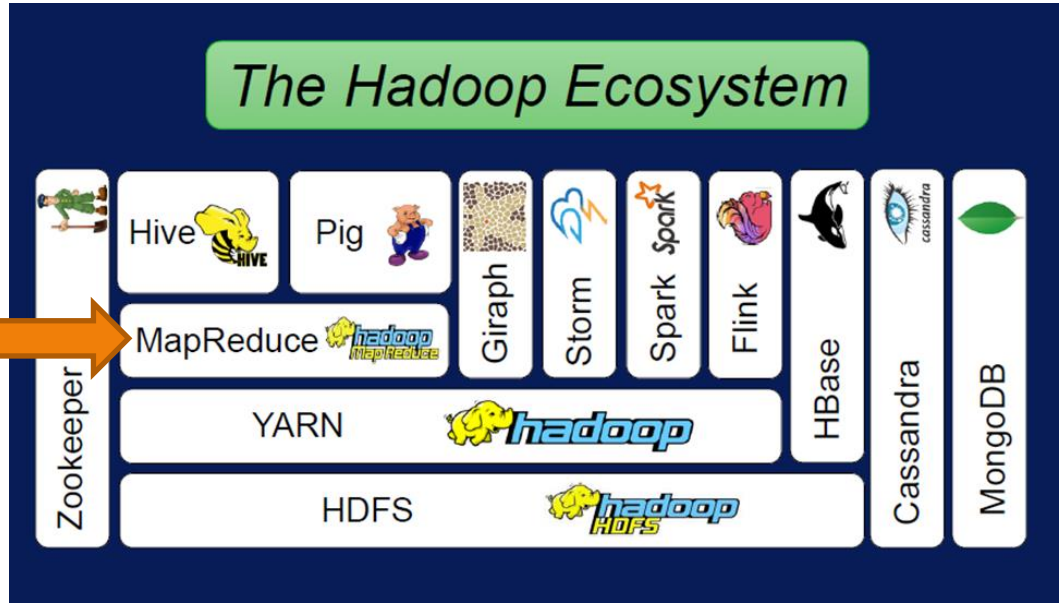


Yahoo uses it scheduling Jobs using 40.000 servers

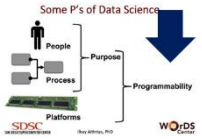
Hadoop: Ecosystem

Simplified programming scheme

Allows parallel computing over Hadoop



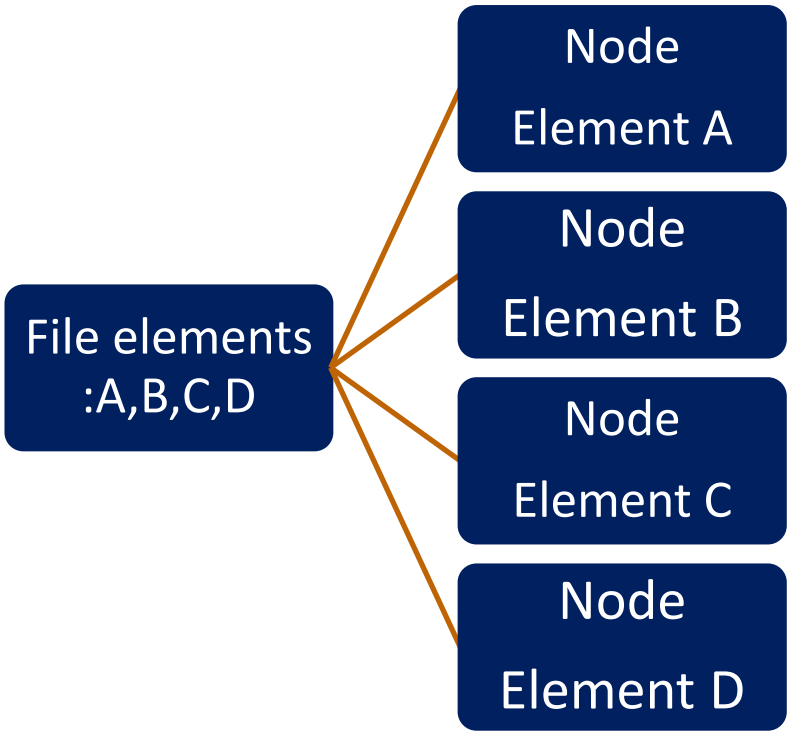
Google uses Mapreduce for web indexing

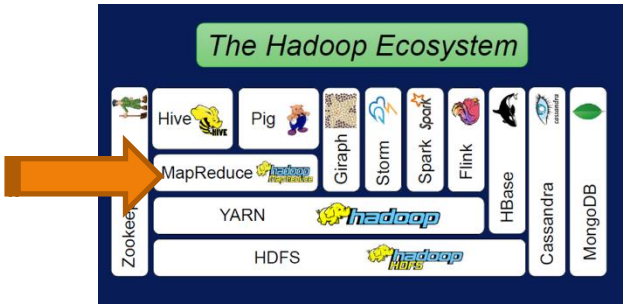


Hadoop: MapReduce

MapReduce: example Wordcount

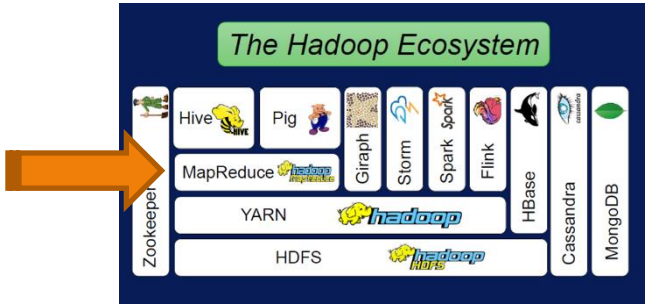
- **Step 0: the file is store in HDFS in different nodes**





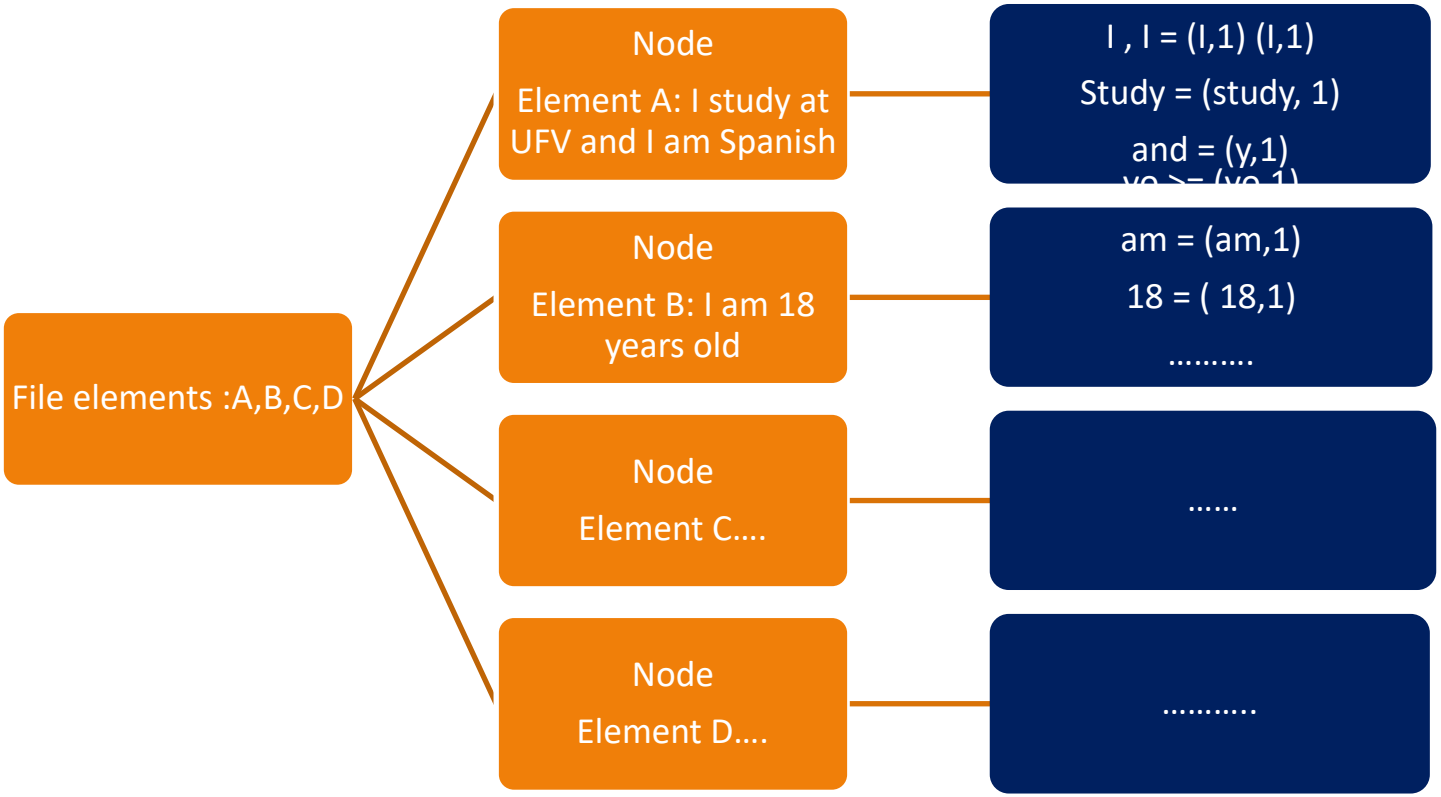
https://www.ibm.com/developerworks/library/bd-yarn-intro/

Hadoop: MapReduce



MapReduce: example Wordcount

- **Step 1: Map each node**
  - **Mapping consists of generate key value pairs**

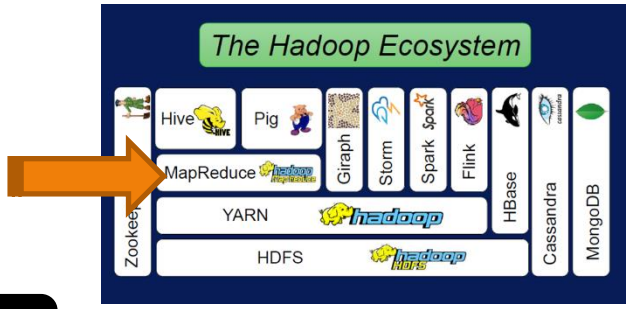


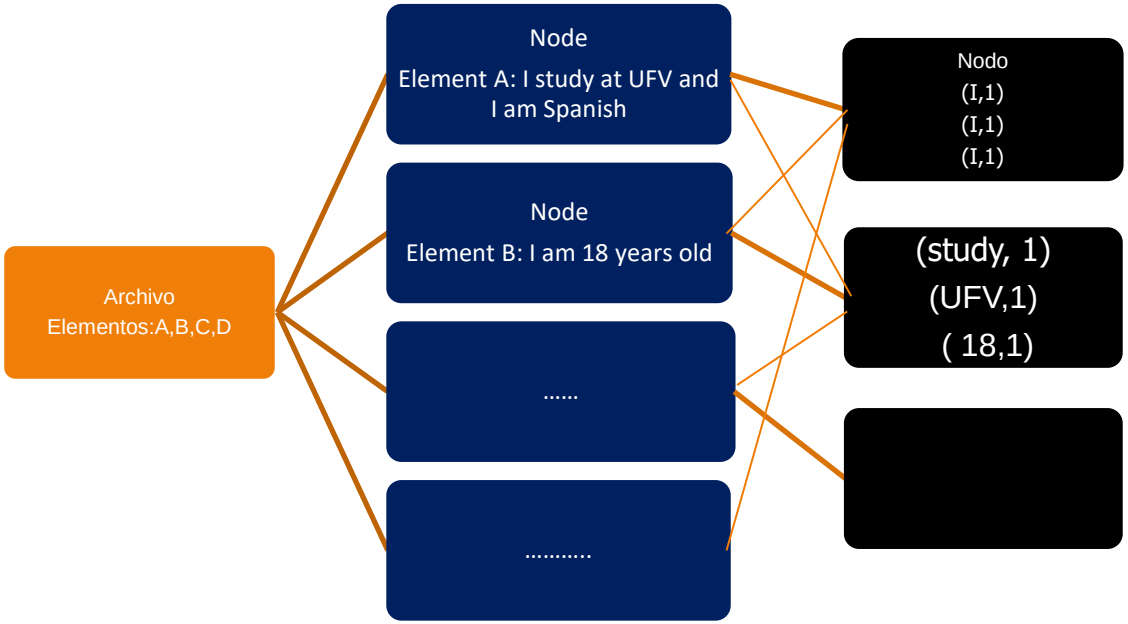
https://www.ibm.com/developerworks/library/bd-yarn-intro/

Hadoop: MapReduce

MapReduce: Word count

- **Paso 2: Sort&shuffle in each node**
  - **Move Equal"key value pairs" to the same node**

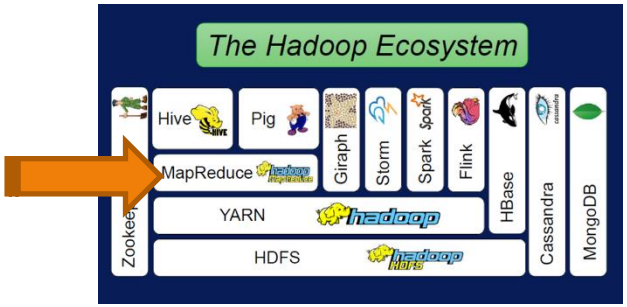


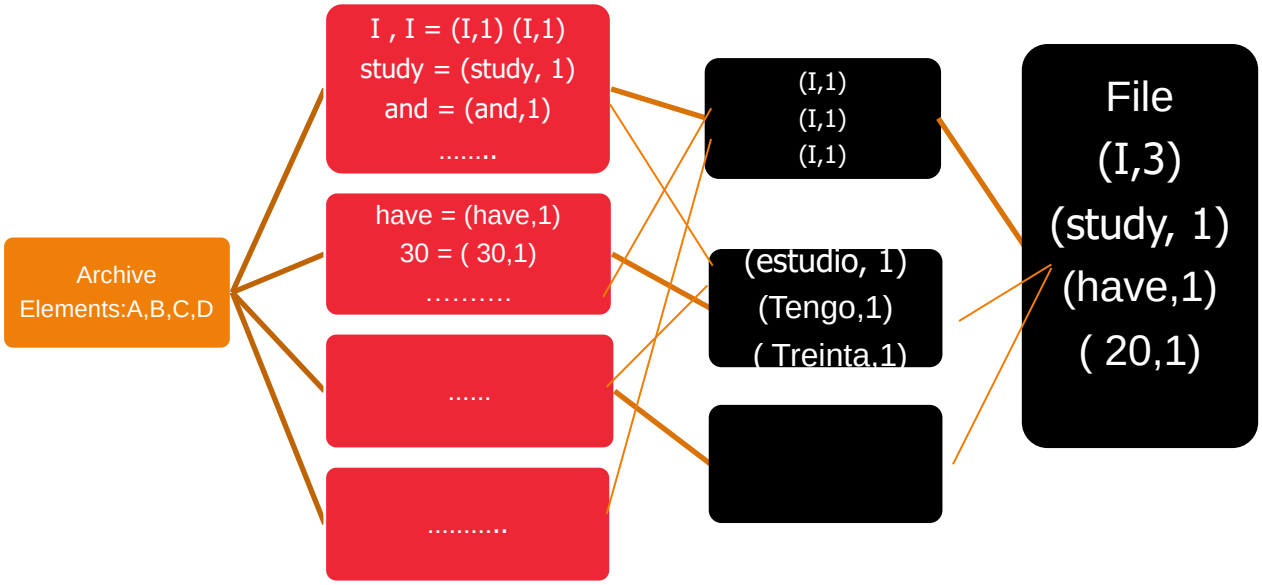


https://www.ibm.com/developerworks/library/bd-yarn-intro/

MapReduce: word count

- **Step 3: Reduce**
  - **Add the values of the same "key-value pairs"**

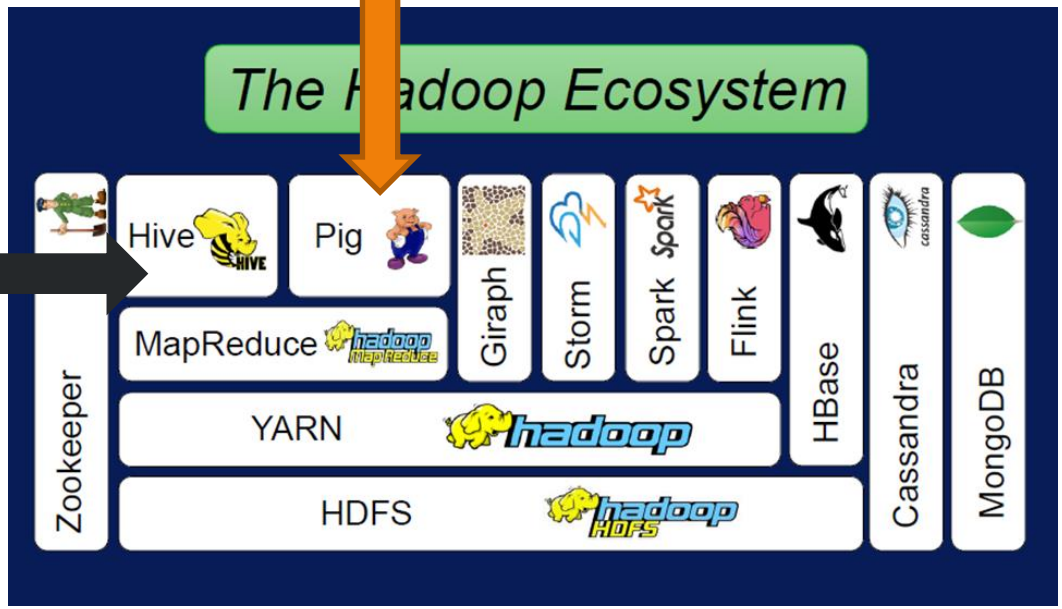




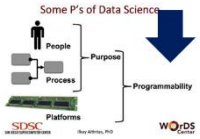
Hadoop: Ecosystem

Programming environment for Data Flow management

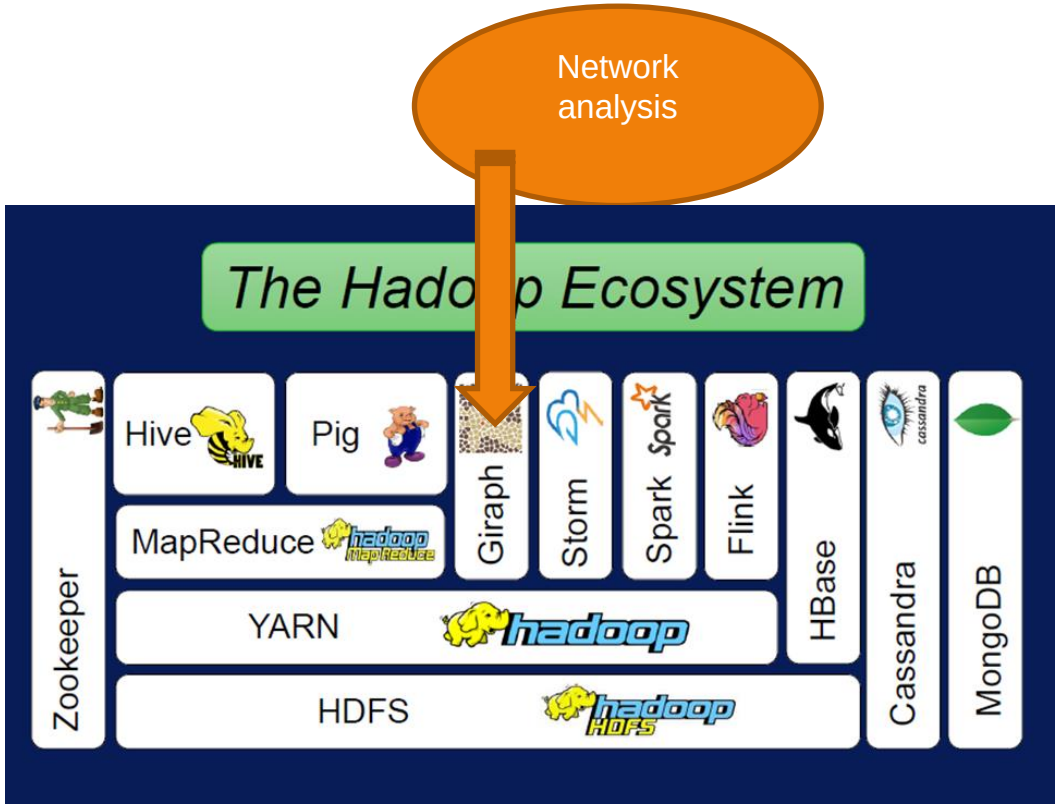
For data query management to SQL data bases



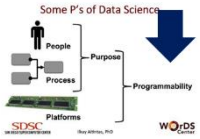
Pig was created byYahoo and Hive by Facebook



Hadoop: Ecosystem

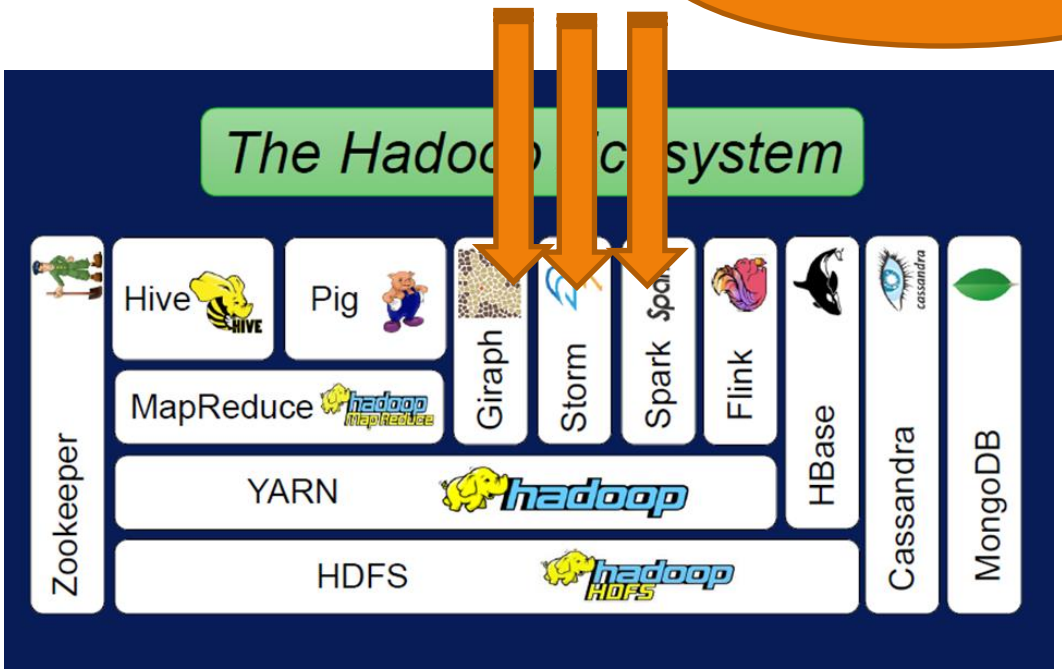


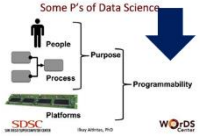
Facebook uses it to analyze social networks



Hadoop: Ecosystem

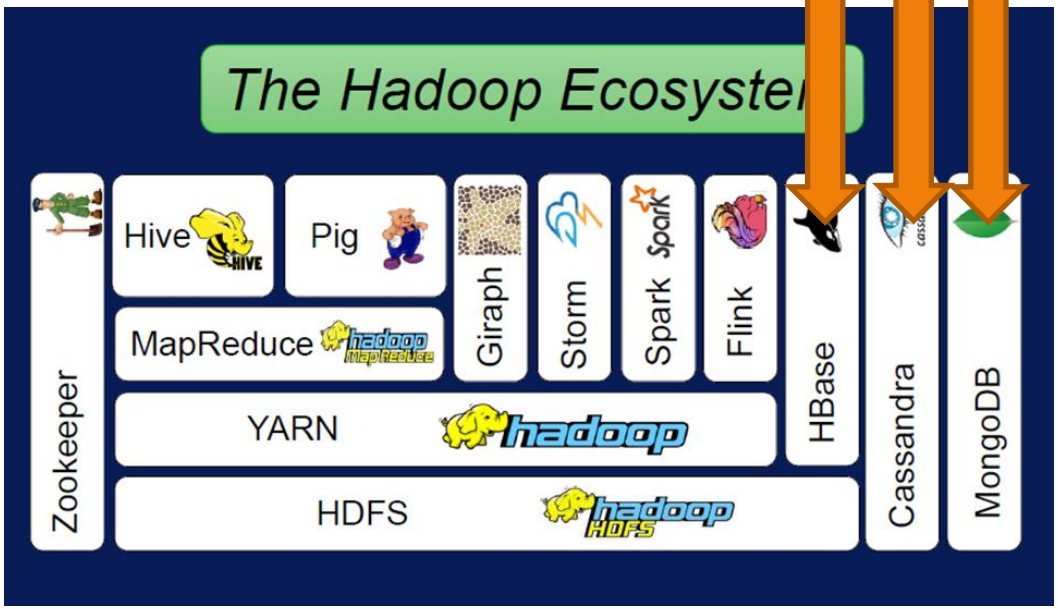
Environments for real-time data processing



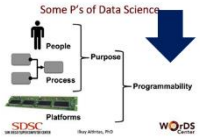


Hadoop: Ecosystem

Non-SQL database management environment. (Hypodense tables)

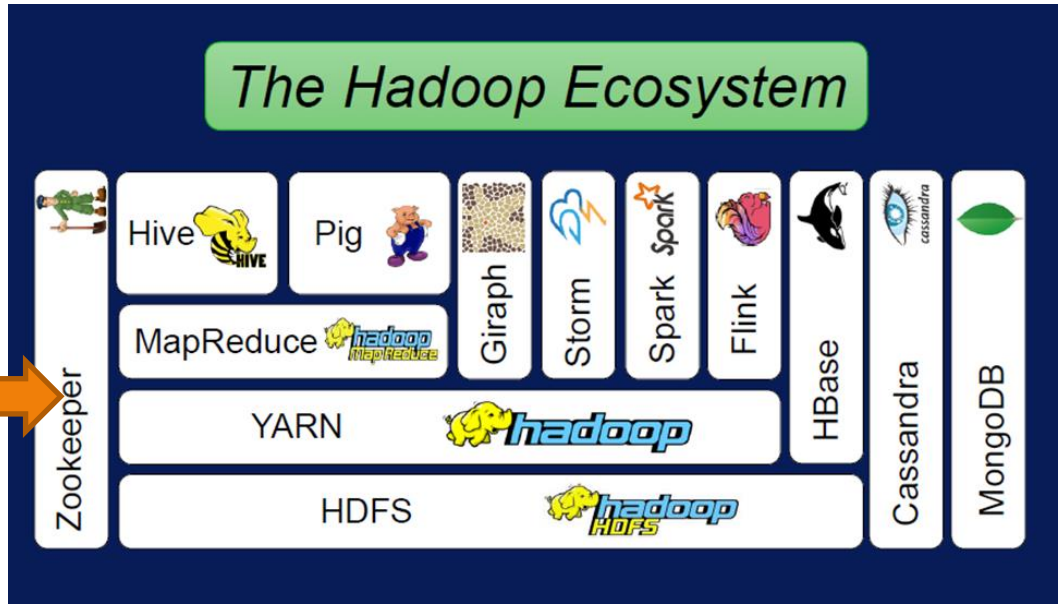


HBase uses Facebook for the Instant messaging platform

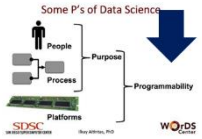


Hadoop: Ecosystem

Synchronization management and high availability



Created by Yahoo to coordinate the environment with "Animal name"

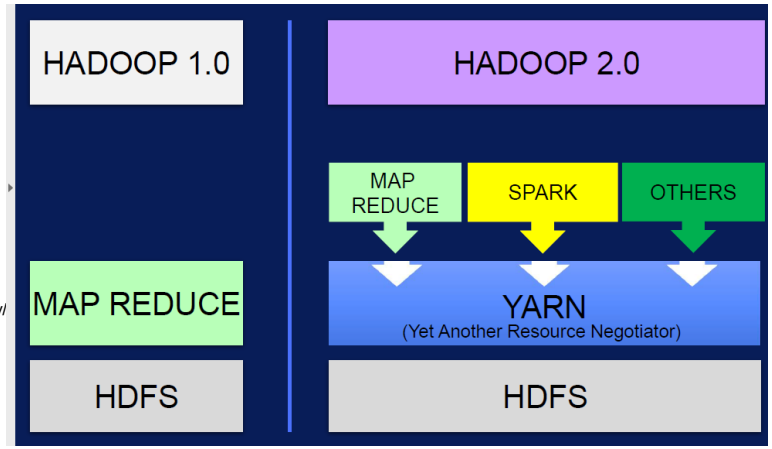


Hadoop: YARN

YARN: Resource Manager for Hadoop

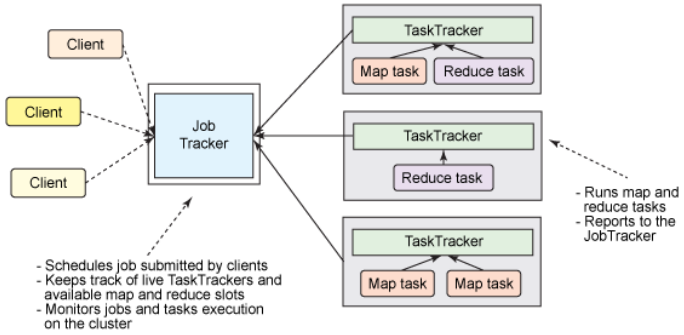
- **For different applications to share the HDFs of Hadoop**
  - Hadoop has evolved over time
  - Hadoop 1.0
  - Only MapReduce-based jobs
  - Can use the resources
- **Hadoop 2.0**
  - Many applications
  - Includes YARN to be able to handle and coordinate resources

https://www.ibm.com/developerworks/library/



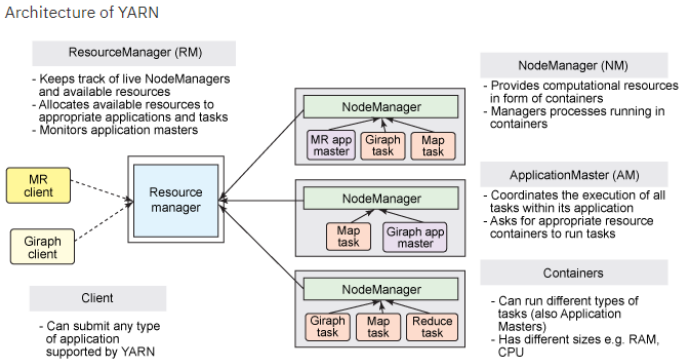
Hadoop: YARN

Ejecución de trabajos en el marco Map reduce



- **MAP REDUCE FRAME**
- **Coordinates the work and maps the map and the reduction**
- **TaskTracker**
- **Executes the work and periodically reports progress to the JobTracker**
- **The problem is:**
- **The JobTrJobTrackeracker is one of bottle and limits the scalability (40,000 tasks maximally)**
- **Clusters do not use resources optimally**
- **Does not allow the use tasks without MapReduce (Giraph for example)**
-

Ejecución de trabajos en el marco YARN



- **YARN FRAME**
- **Has a resource manager**
- **This in a machine to part and arbitrate LSO resources between concurrent applications**
- **It has a NodeManager instead of a Jobtracker**
- **Instead of a fixed amount of data containers is flexible**

https://www.ibm.com/developerworks/library/bd-yarn-intro/

Hadoop: When not to use it

- **HADOOP:**
-
- **Is suitable when**
  - Great data growth anticipated: volume and variety
  - Data is expected to be accessible for a long time
  - Many platforms will be used

- **Is not suitable when**
  - Are small databases
  - Very advanced Algorithms (https://tez.apache.org/)
  - Random access to data
  - Latency-sensitive tasks (http://storm.apache.org/)
  - Security-sensitive tasks and data (http://ranger.apache.org/)

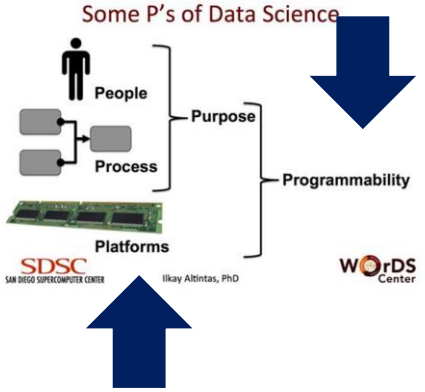
https://www.ibm.com/developerworks/library/bd-yarn-intro/

BOOKs READINGS APPLIED TO THIS PRESENTATION

Book1: Big Data for Dummies

- Chapter 4
- Chapter 5
- Chapter 8
- Chapter 10

Index

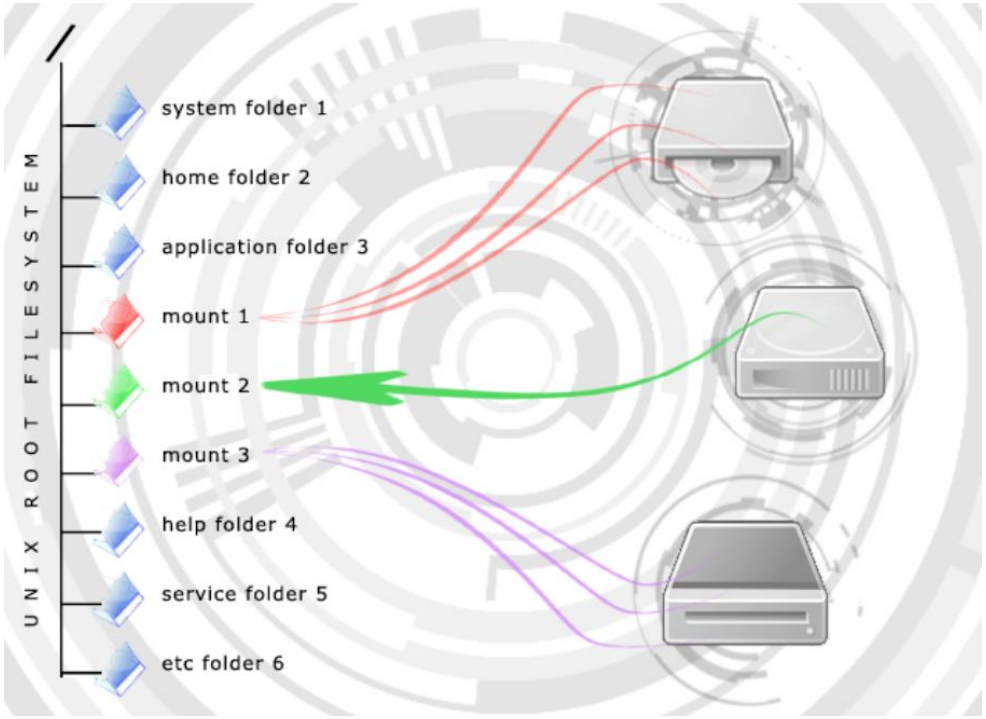


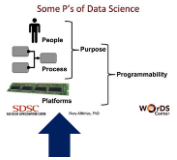
01.01- What is a DFS?

DFS Distributed File System

- **Why?**
  - Access later to the results of a process
  - Save big amounts of data
  - Allow Access to multiple processes

If it doesnt fit?

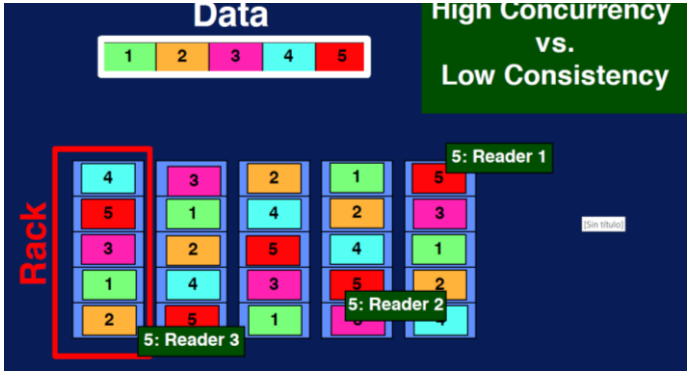




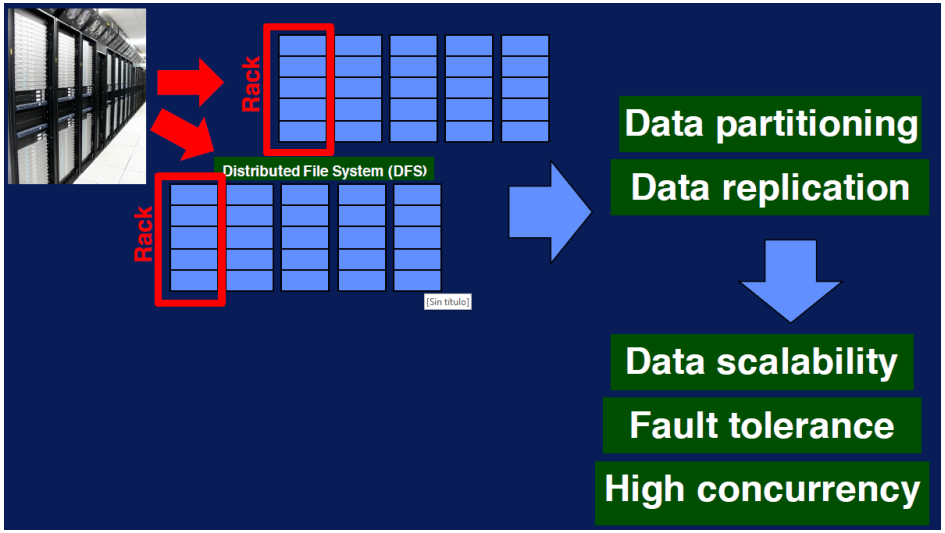
01.01- What is a DFS?

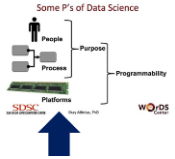
DFS Distributed File System

- **Consequences**
  - High concurrency vs low consistency
    - Data partition
    - Data replication
  - Data scalability
  - Fault tolerance
  - High concurrency



implies





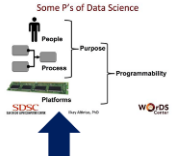
01.02 – Computation and scalability

- **Commodity cluster (underlaying hardware)**
  - **Data parallelism**
  - **Commodity cluster architecture and fault tolerance**

Affordable costs. Reduce computation costs

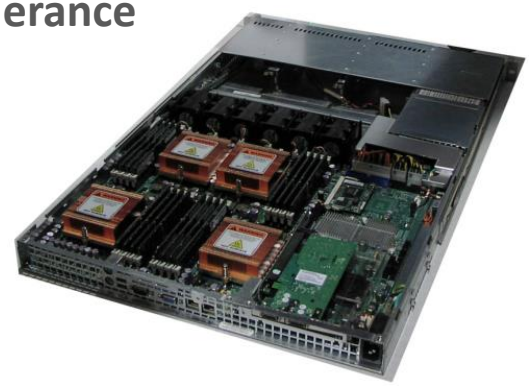
Non specialise

Allows distributed computation over internet

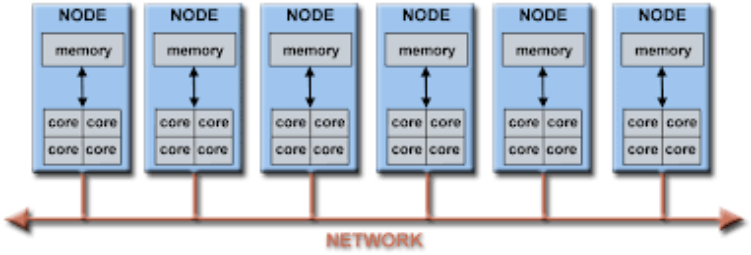


- **Commodity cluster (underlaying hardware)**
  - **Data parallelism**
  - **Commodity cluster architecture and fault tolerance**

One computer



Parellel computers



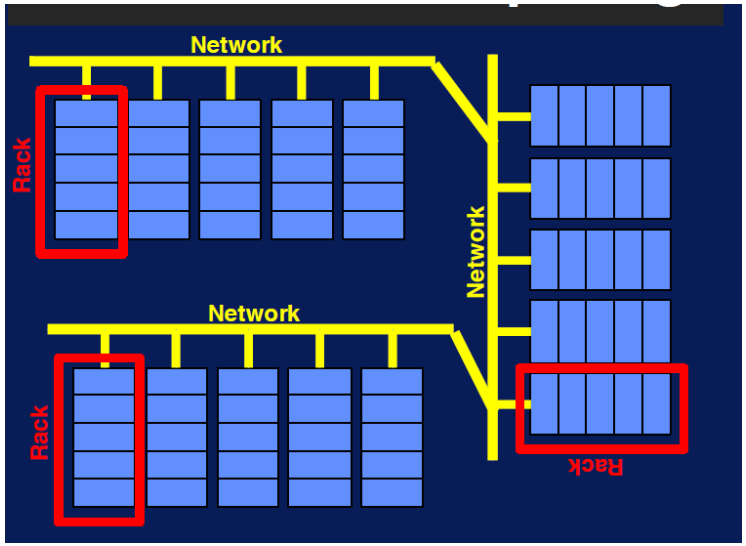
01.02 – Computation and scalability

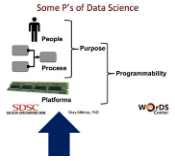
- **Commodity cluster (underlaying hardware)**
  - **Data parallelism**
  - **Commodity cluster architecture and fault tolerance**

Allows parallel computation

Redundant data warehousing against failure

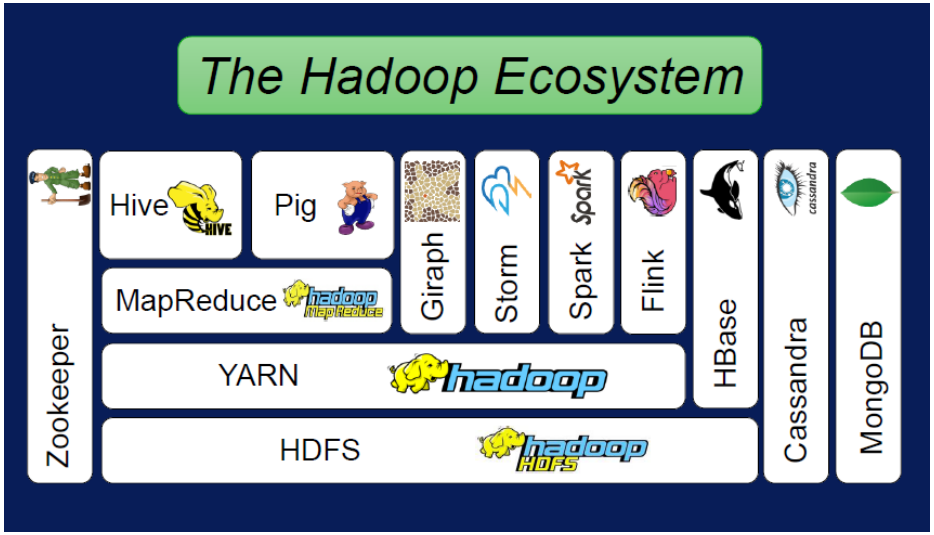
Restart parallel processing

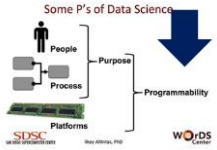




01.03 – Hadoop: Why?

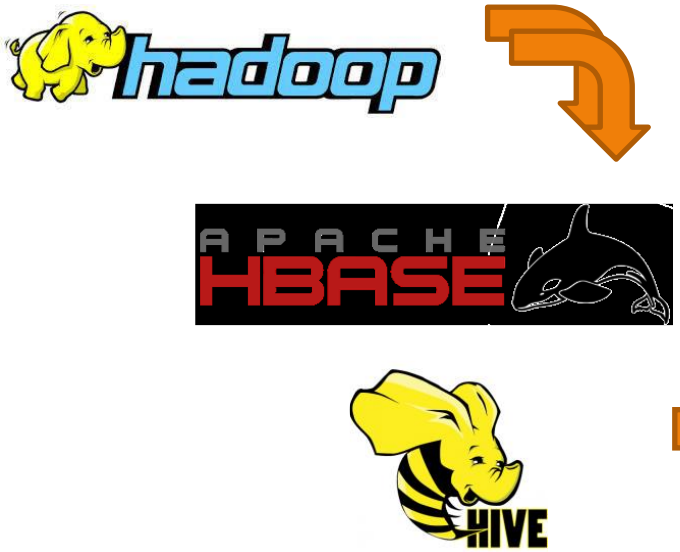
Allows scalability	Allows fault tolerance	Allows data variety
Enables compatible environment	Creates value with a big support community and with multiple applications developed	Al the applications are for free and opensource

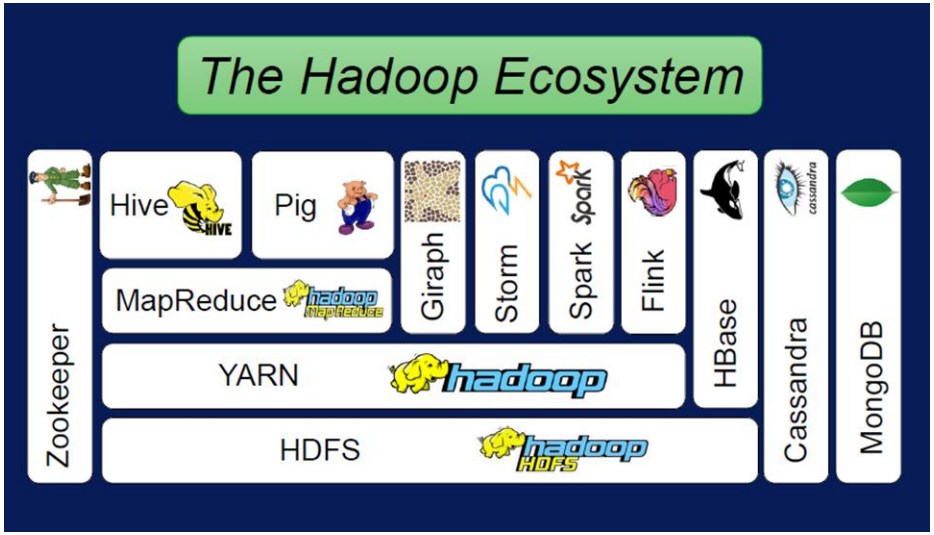


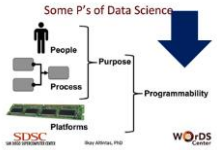


02.03 – Hadoop: Ecosystem

- **Created by Yahoo in 2005**
- **New work environments have been added**
- **Now there are more tan 100**



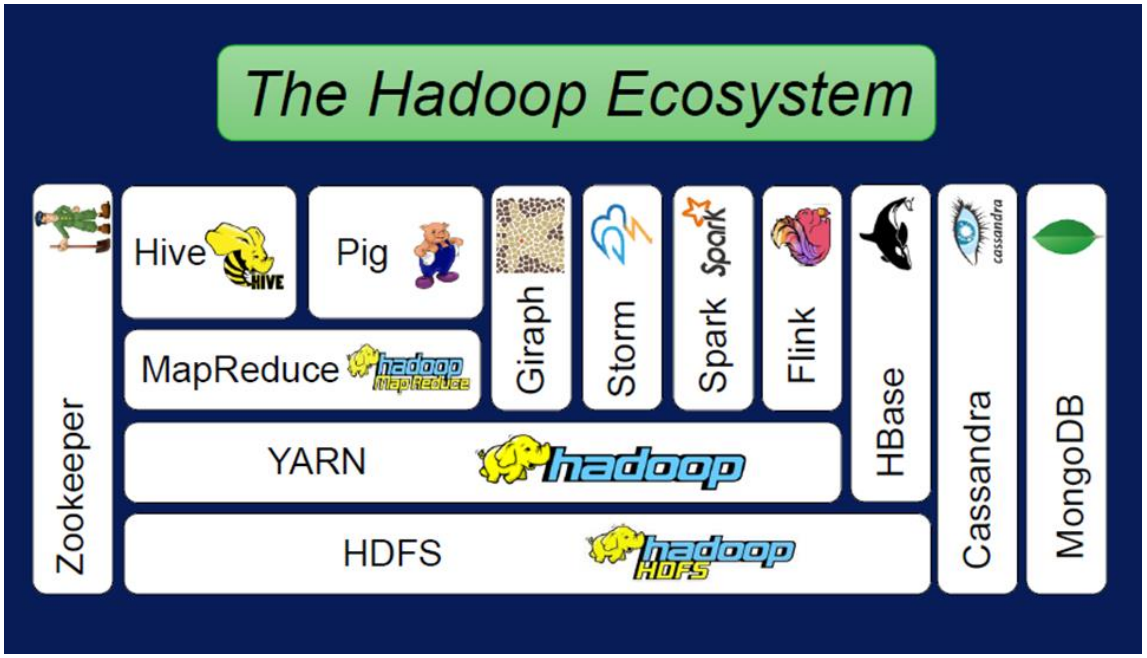


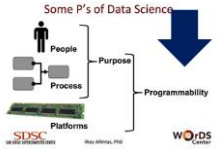


02.03 – Hadoop: Ecosystem

Higher leel of interactivity

Data storage and activity scheduling

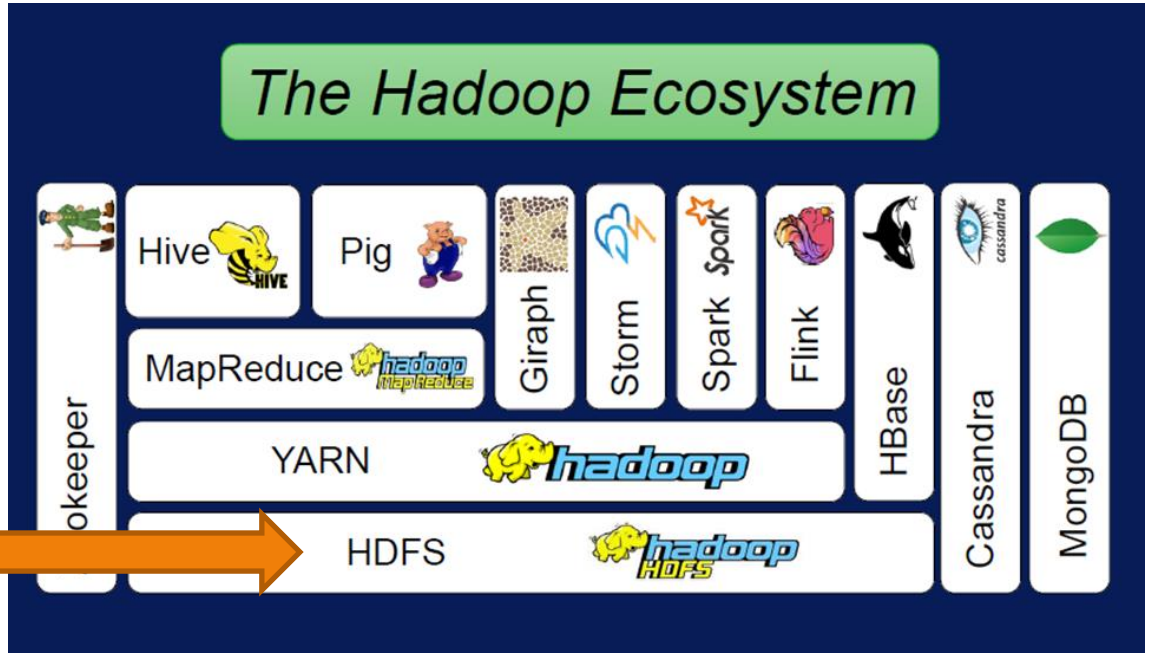


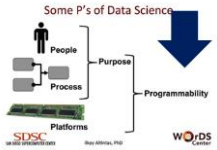


02.03 – Hadoop: Ecosystem

Scalable data warehouse based on DFS

Fault tolerance

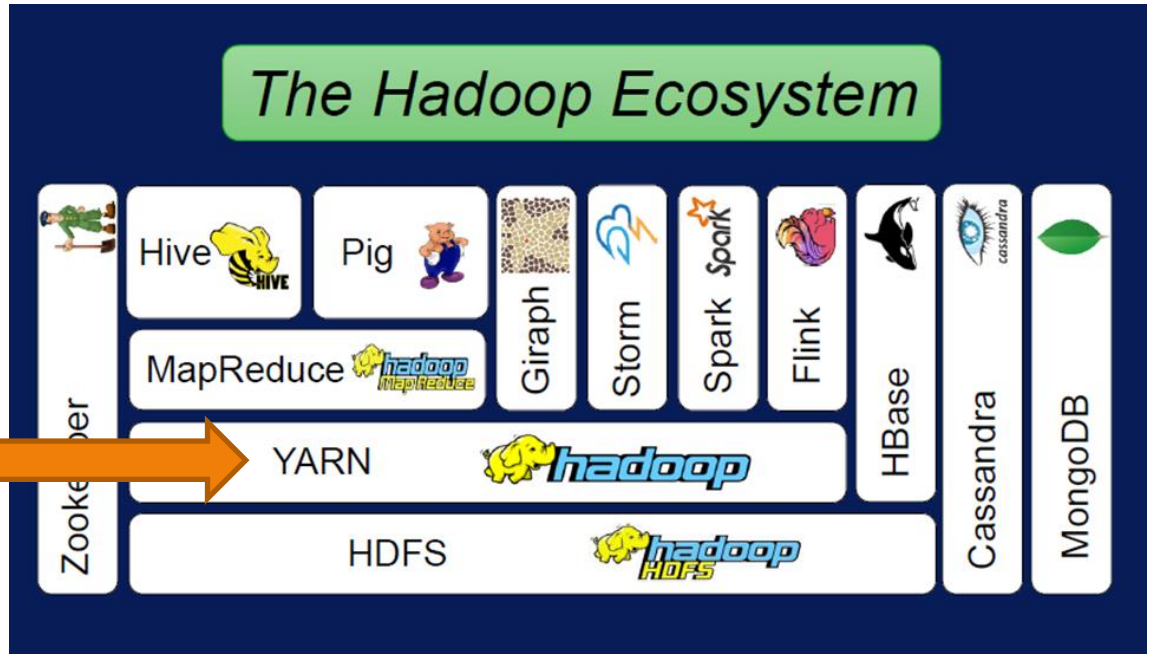




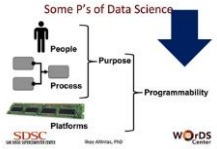
02.03 – Hadoop: Ecosystem

Flexible Job scheduling

Resource management



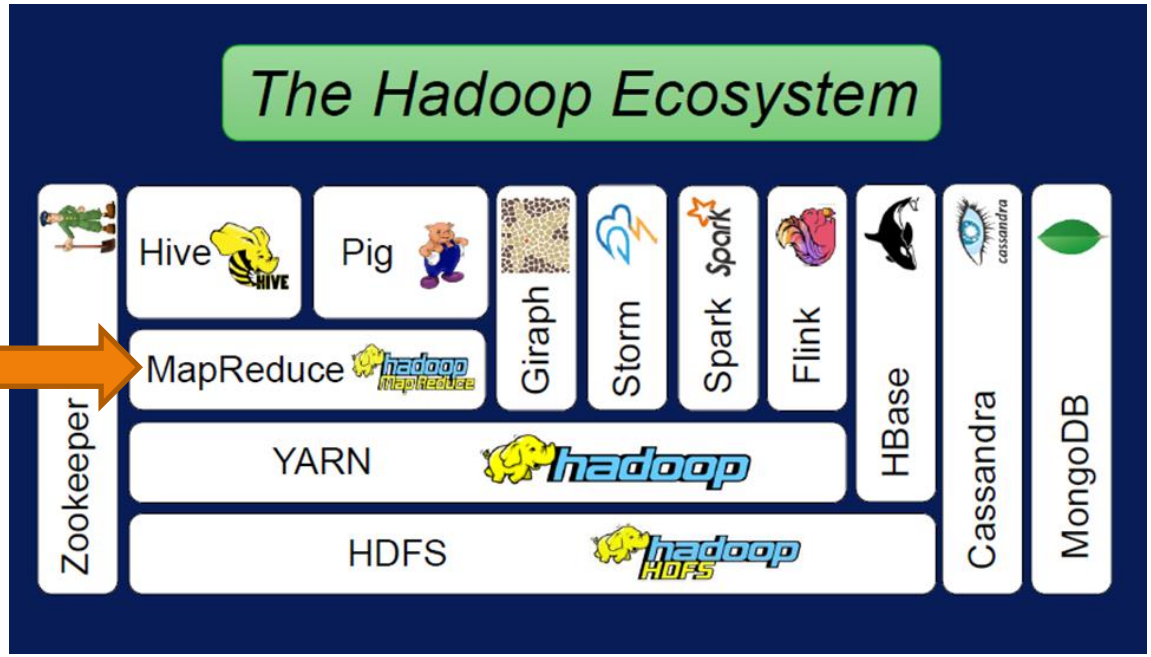
Yahoo uses it scheduling Jobs using 40.000 servers



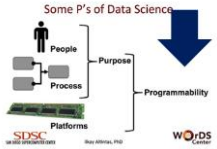
02.03 – Hadoop: Ecosystem

Simplified programming scheme

Allows parallel computing over Hadoop

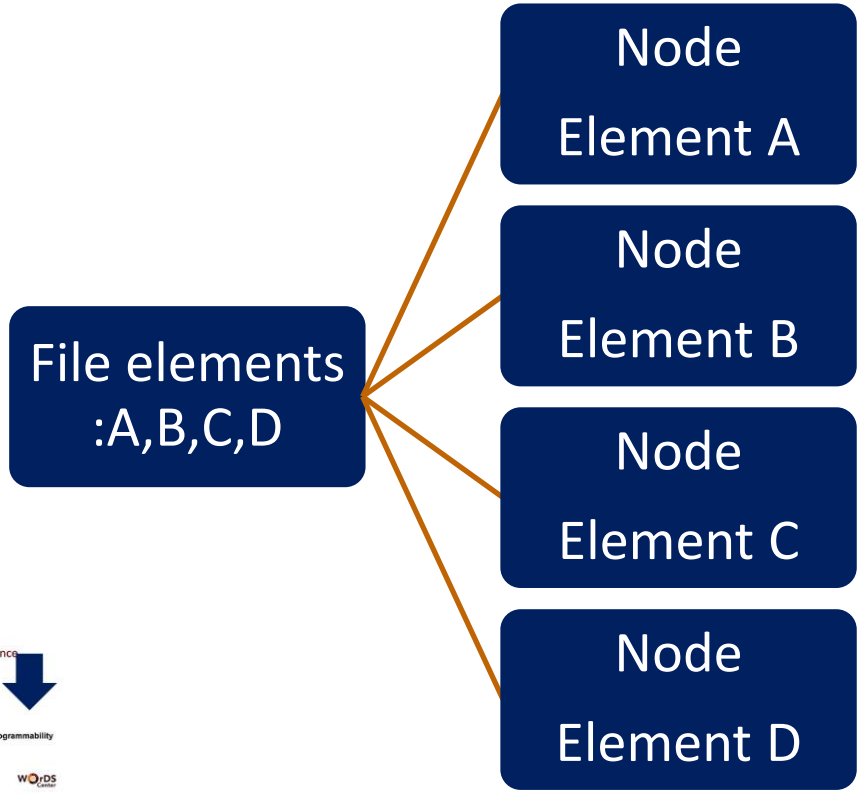


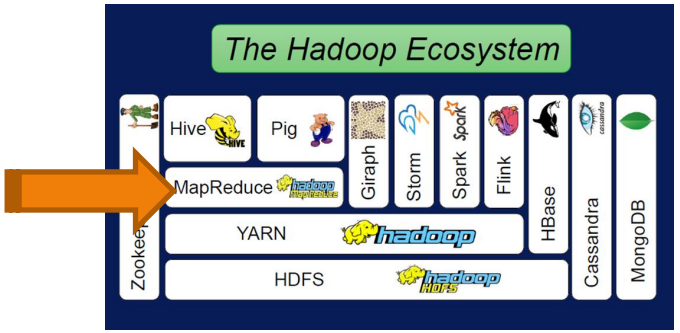
Google uses Mapreduce for web indexing

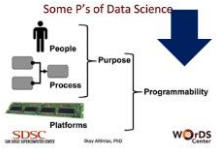


MapReduce: example Wordcount

- **Step 0: the file is store in HDFS in different nodes**

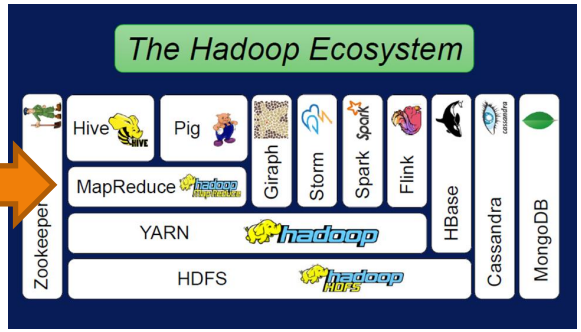






https://www.ibm.com/developerworks/library/bd-yarn-intro/

02.03 – Hadoop: MapReduce



MapReduce: example Wordcount

- **Step 1: Map each node**
  - **Mapping consists of generate key value pairs**

File elements :A,B,C,D

Node
Element A: I study at UFV and I am Spanish

I , I = (I,1) (I,1)
Study = (study, 1)
and = (y,1)

Node
Element B: I am 18 years old

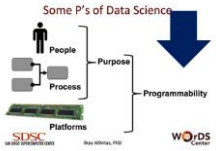
am = (am,1)
18 = (18,1)
…….

Node
Element C….

……

Node
Element D….

…….

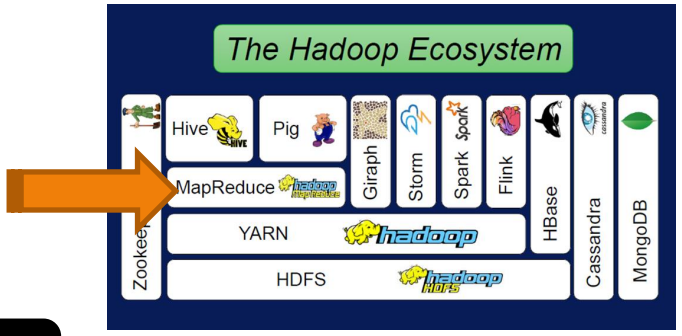


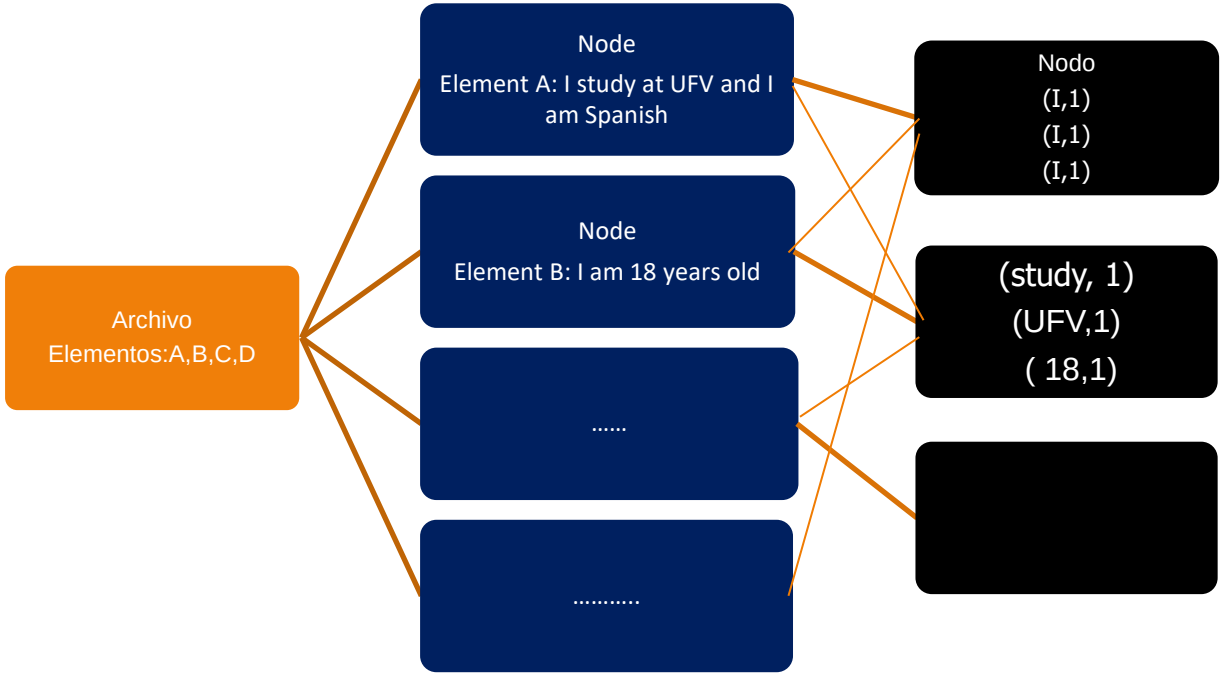
https://www.ibm.com/developerworks/library/bd-yarn-intro/

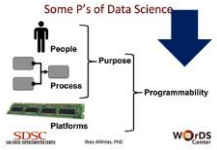
02.03 – Hadoop: MapReduce

MapReduce: contar palabras

- **Paso 2: Sort&shuffle in each node**
  - **Move Equal"key value pairs" to the same node**



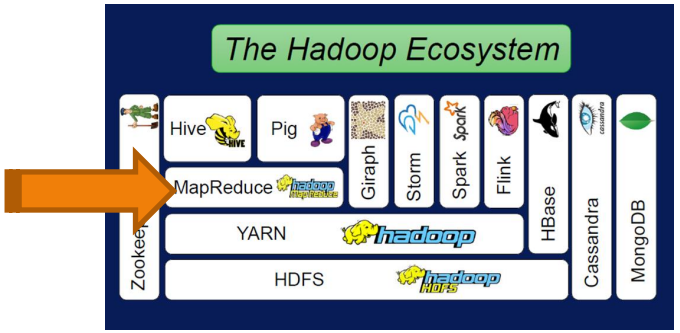


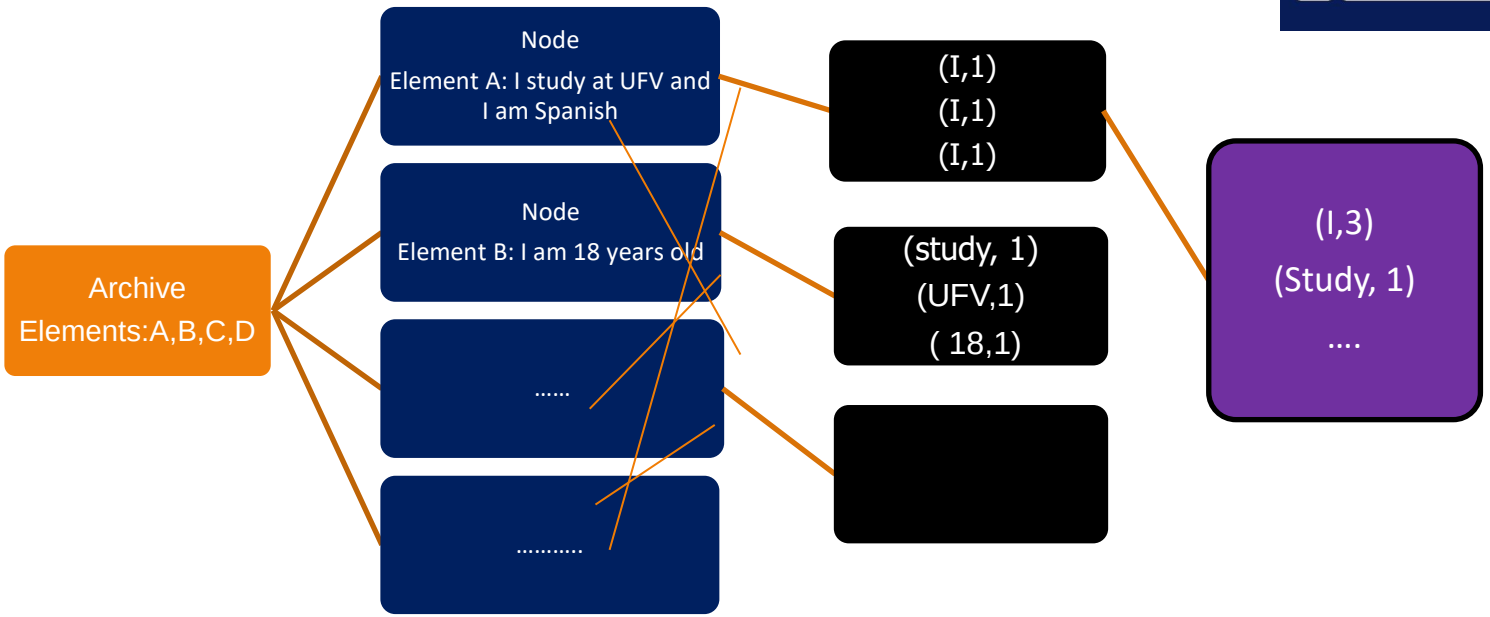


https://www.ibm.com/developerworks/library/bd-yarn-intro/

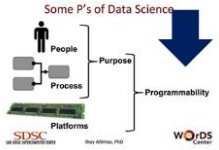
MapReduce: Wordcount

- **Step 3: Reduce**
  - **Add the same "key-value pairs"**





https://www.ibm.com/developerworks/library/bd-yarn-intro/



BOOKs READINGS APPLIED TO THIS PRESENTATION AND INCLUDED FOR THE EXAMS

Book1: Big Data for Dummies

- Chapter 9: Exploring the world of Hadoop
  - Pg. 112_119
- Chapter 10: the Hadoop Foundation and Ecosystem
  - Pg. 121_128 (only read not study)