

# What is Hadoop?

"an open source **software platform** for **distributed storage** and **distributed processing** of **very large data sets** on **computer clusters** built from commodity hardware" - *Hortonworks*

1.

**Big Data:** Imagine your favorite online store constantly collecting data on every click, purchase, and review. This massive amount of information, encompassing structured (transaction details) and unstructured data (reviews, social media mentions), is **big data**. It's too complex and voluminous for traditional software to handle.

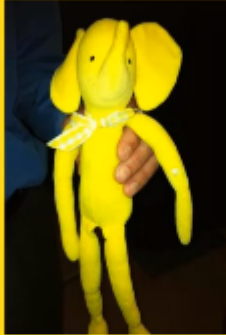
**Hadoop:** Here's where **Hadoop** comes in. It's a framework that helps store, process, and analyze big data. Think of it as a team of data janitors working together to sort, clean, and make sense of this massive warehouse of information. Hadoop does this by splitting the data into smaller chunks, distributing them across a network of computers, and then analyzing them simultaneously. This parallel processing allows Hadoop to crunch through big data much faster than a single computer ever could.

**Practical Example:** Now, let's see how this benefits our online store. By analyzing their big data with Hadoop, they can:

- **Predict Customer Behavior:** Identify patterns in past purchases and browsing habits to recommend personalized products and offers, increasing sales and customer satisfaction.

3. Hadoop was introduced as GFS, MapReduce ...but yahoo stole trade secrets and built

## Hadoop History

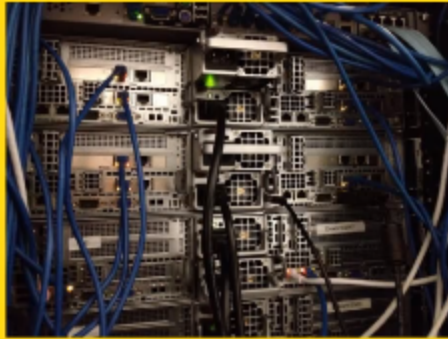


- Google published GFS and MapReduce papers in 2003-2004
- Yahoo! was building "Nutch," an open source web search engine at the same time
- Hadoop was primarily driven by Doug Cutting and Tom White in 2006
- It's been evolving ever since...

hadoop  
elephant

it was named after a toy

## Why Hadoop?

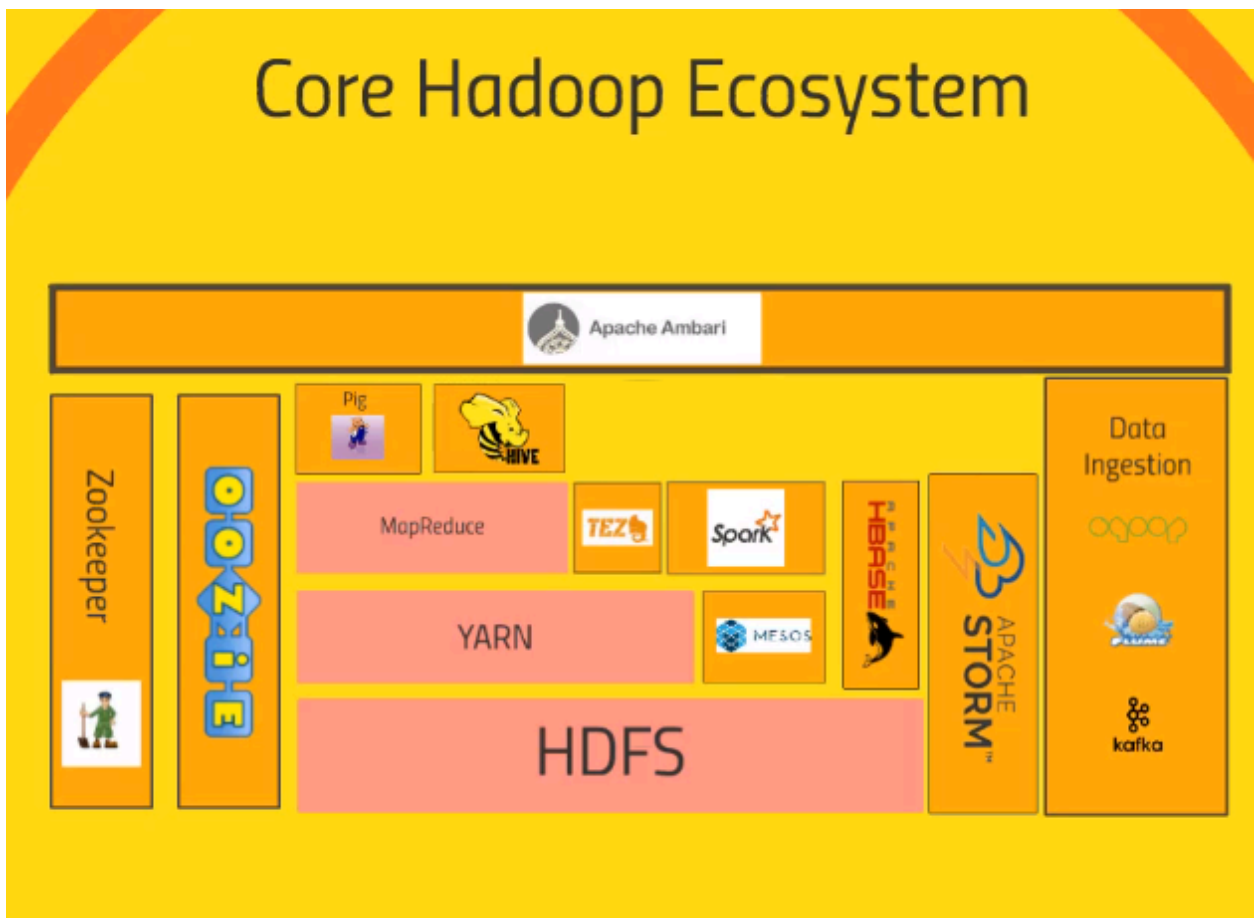


- Data's too darn big - terabytes per day
- Vertical scaling doesn't cut it
  - Disk seek times
  - Hardware failures
  - Processing times
- Horizontal scaling is linear
- Hadoop: It's not just for batch processing anymore

- 4.
5. Cuz data is too large we may get data from sensors,stock market data,search engine data etc
6. Vertical scaling means one pc with more power & storage..
7. Horizontal scale means more computers etc

Overview of Hadoop Ecosystem

# Core Hadoop Ecosystem



- 1.
2. Now lets learn HDFS-hadoop distributed file sys

HDFS, which stands for Hadoop Distributed File System, is a crucial component of the Hadoop ecosystem specifically designed to handle big data storage and management. Think of it as the storage room in your giant office building filled with mountains of files, but instead of messy stacks, everything is neatly organized and easily accessible thanks to a team of efficient data librarians.

Here's HDFS explained with a practical example:

Imagine you're a media company with petabytes of raw video footage from various TV shows and documentaries. Traditionally, storing this data on a single server would be impractical and expensive. This is where HDFS comes in:

1. **Dividing the Data:** HDFS chops your massive video files into smaller, manageable "blocks" (think like individual file folders).
2. **Distributed Storage:** These blocks are then spread across multiple computers in the Hadoop cluster, like placing the folders in different rooms of your office building.
3. **Replication for Reliability:** For extra protection, HDFS stores multiple copies of each block on different computers. This ensures access even if

- 3.
4. Lets learn YARN-yet another resource negotiator
5. So yarn is basically the system that manages the resources on your computing cluster.
6. It's what decides what gets to run tasks when what nodes are available for extra work which nodes are not, which ones are available, which ones are not available so it's kind of the heartbeat that keeps your cluster going.

Imagine you're launching a massive marketing campaign for a new product launch. You have piles of data to analyze from different sources: web traffic, social media mentions, purchase records, and customer surveys. You need to process this data **fast, effectively, and simultaneously** to understand what's working and what's not. That's where **YARN** comes in.

**YARN** stands for **Yet Another Resource Negotiator**, and it's the resource manager and job scheduler in the Hadoop ecosystem. Think of it as the conductor of a big orchestra, coordinating different groups of musicians (processing tasks) to play together (analyze data) and produce a beautiful melody (insights).

- 7.
8. MapReduce
9. It consists of mappers and reducers.
10. These are both different scripts that you might write or different functions if you will when you're writing a map reduce program. Mappers have the ability to transform your data in parallel across your entire computing cluster in a very efficient manner. And reducers are what aggregate that data together and it may sound like a very simple model but its not

MapReduce is a groundbreaking programming model and processing technique designed for handling massive datasets across distributed computing environments. It consists of two main steps: the Map phase and the Reduce phase.

In the Map phase, the input data is divided into smaller chunks, and a map function is applied to each chunk independently. This function transforms the input data into a set of key-value pairs. This stage is all about breaking down the problem into manageable pieces that can be processed in parallel.

Now, let's dive into an example to illustrate the power of MapReduce. Imagine you have a colossal dataset containing information about website visits, and you want to count the number of times each unique URL appears. The Map function would take each webpage's URL and emit a key-value pair, where the URL is the key, and the value is set to 1.

In the Reduce phase, the framework groups all key-value pairs with the same key together and applies a reduce function to aggregate the values. In our example, the reduce function would sum up the 1s associated with each URL, providing the total count of visits for each webpage.

#### 12. PiG

Apache Pig is a high-level programming API designed for processing and analyzing large-scale data without the need for writing complex Java or Python MapReduce code. It provides a scripting language with a SQL-style syntax, allowing users to write simple scripts that resemble SQL queries. Pig enables the chaining together of these scripts to perform complex data transformations and analyses. The key advantage is that users can work with big data using a more familiar scripting language without delving into the intricacies of Java or Python. Pig scripts are then translated into MapReduce jobs, which leverage Hadoop's ecosystem, including YARN and HDFS, to process and retrieve the desired results efficiently.

#### 14. HiVE

15. it really more directly looks like a SQL database so hive is a way of actually taking SQL queries and making this distributed data that's just really sitting on your file



system somewhere look like a SQL database. So for all intents and purposes it's just like a database. You can even connect to it through a shell client or ODBC or what have you.

16. Apache Ambari - So Ambari is what sits on top of all this and lets you have a view into the actual state of your cluster and the applications that are running on it.

17. MESOS

18. It is alternate to YERN

19. SPARK

20. But SPARK is kind of where it's at right now it is extremely fast it's under a lot of active development

21. So if you need to very quickly and efficiently and reliably process data on your cluster SPARK is a

22. really good choice for that.

23. And it's also very versatile it can do things like handle SQL queries that can do machine learning across an entire cluster of information.

24. TEZ(refer bard)

Tez is a **powerful and flexible data-flow processing framework** built on top of YARN in the Hadoop ecosystem. It provides an alternative to MapReduce with several key advantages, making it a popular choice for many big data processing tasks.

Think of Tez as a **flexible pipeline system** in a factory. Instead of a rigid assembly line, Tez allows you to define and customize the flow of data through different processing stages (Map, Shuffle, Reduce, etc.) as independent units, offering greater control and efficiency.

Here's how Tez differs from MapReduce and its benefits, explained with an example:

25.