Why Big Data

- Extremely large dtasets that are hard to deal with using Relational Databases
  - Storgae/cost
  - Search/Performance
  - Analytics and Visualization
- Need for Parallel Processing on hundreds of Machines
  - ETL cannot complete within a reasonable time

Big Data Facts

- - Facebook now uploads over [300 TB] of data daily.
- - The U.S. Library of Congress has accumulated over [1.5 PB] of data as of the latest update.
- - Walmart's data handling has increased, now managing more than [3 PB] of data generated from customer transactions every hour.
- - Google's data processing has surged to [30 PB] per day.
- - The digital universe has expanded, with an estimated [5 ZB] of data existing currently.

Distributed Data processing

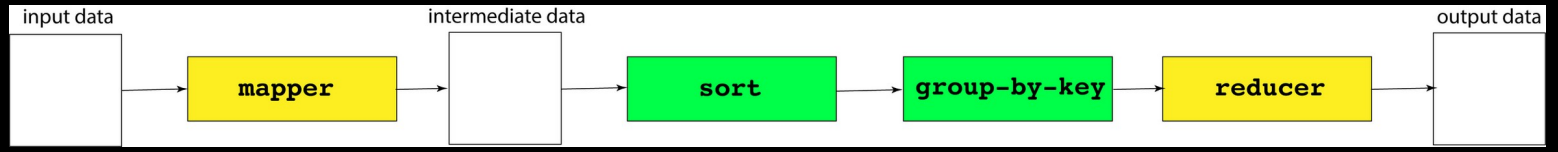
- Scalability to large data volumes:
  - Scan 1000 TB on 1 node @ 100 MB/s = 24 days
  - Scan on 1000-node cluster = 35 minutes

- Cost-efficiency:
  - Commodity nodes /network
  - » Cheap, but not high bandwidth, sometime unreliable
  - Automatic fault-tolerance (fewer admins)
  - Easy to use (fewer programmers)

Overview of MapReduce

MapReduce is a parallel, distributed programming model and implementation used to process and generate large data sets.

- The map component of a MapReduce job typically parses input data and distills it down to some intermediate result.
- The reduce component of a MapReduce job collates these intermediate results and distills them down even further to the desired output.

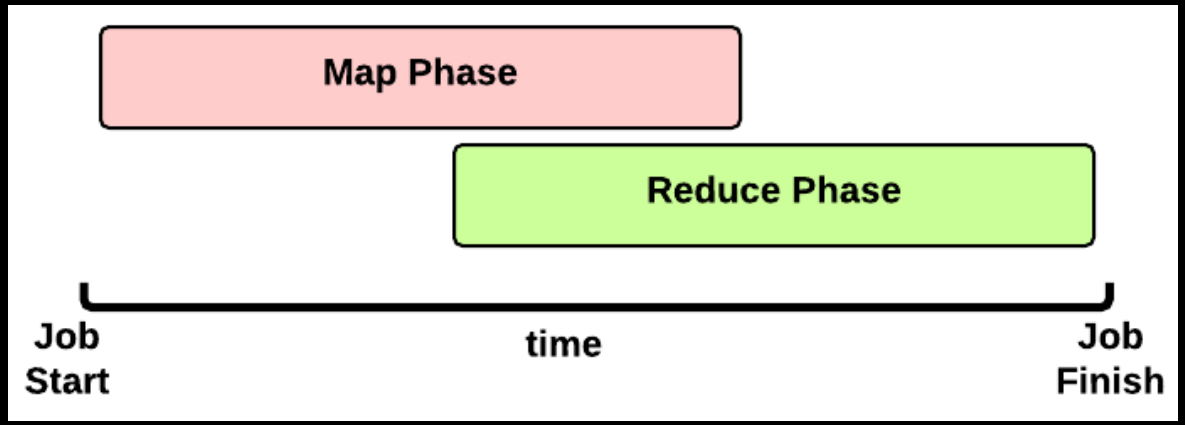


Functions

- Automatic parallelization & distribution
- Fault-tolerance
- Status and monitoring tools
- A clean abstraction for programmers
  - » Functional programming meets distributed computing
  - » A batch data processing system

Anatomy of a MapReduce Job

- In MapReduce, a **YARN** application is called a Job.



Notice that the Reduce Phase may start before the end of Map Phase. Hence, an interleaving between them is possible.

MapReduce: Programming model

- Have multiple map tasks and reduce tasks
- Users implement interface of two primary methods:
  - Map: (key1, val1) → (key2, val2)
  - Reduce: (key2, [val2]) → [val3]
    - Key should not have duplicates
    - Keys are unique
    - Values can have duplicates

Job details

- Job sets the overall MapReduce job configuration
- Primary interface for a user to describe MapReduce job to the Hadoop framework for execution

  Components of Mapreduce jobs:

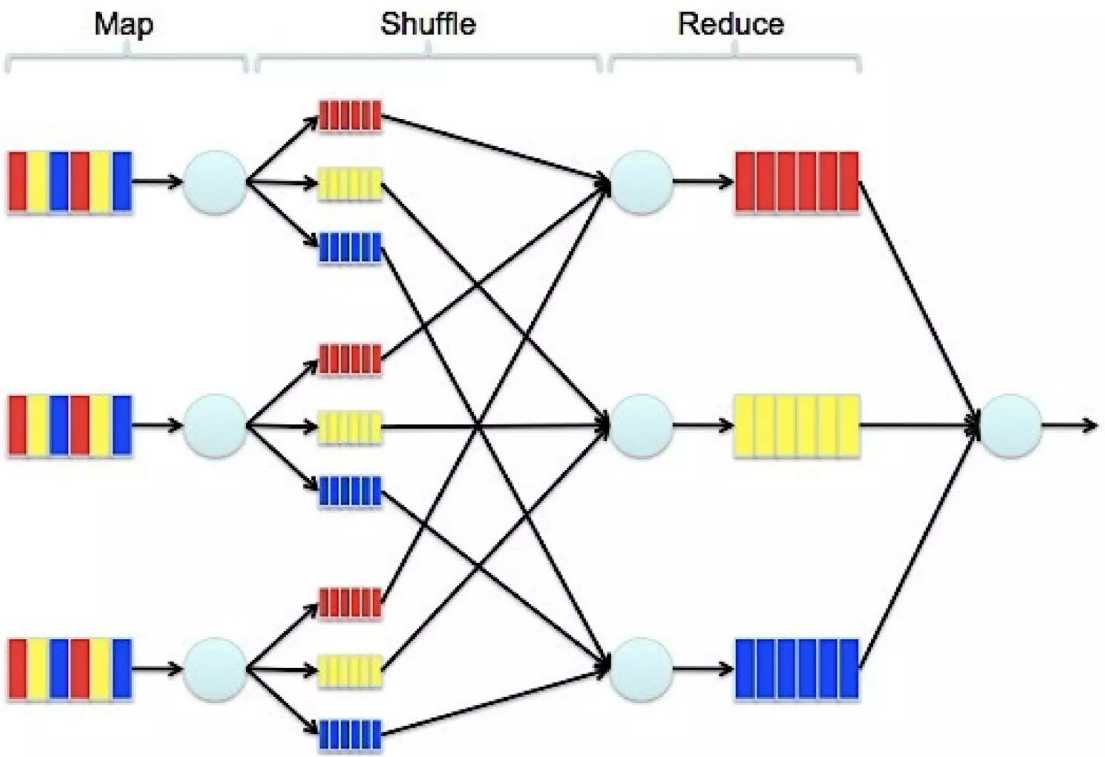
  - Input data
  - Map function
  - Shuffle and sort
  - Reducer function
  - Output

Running in Java is bit complicated

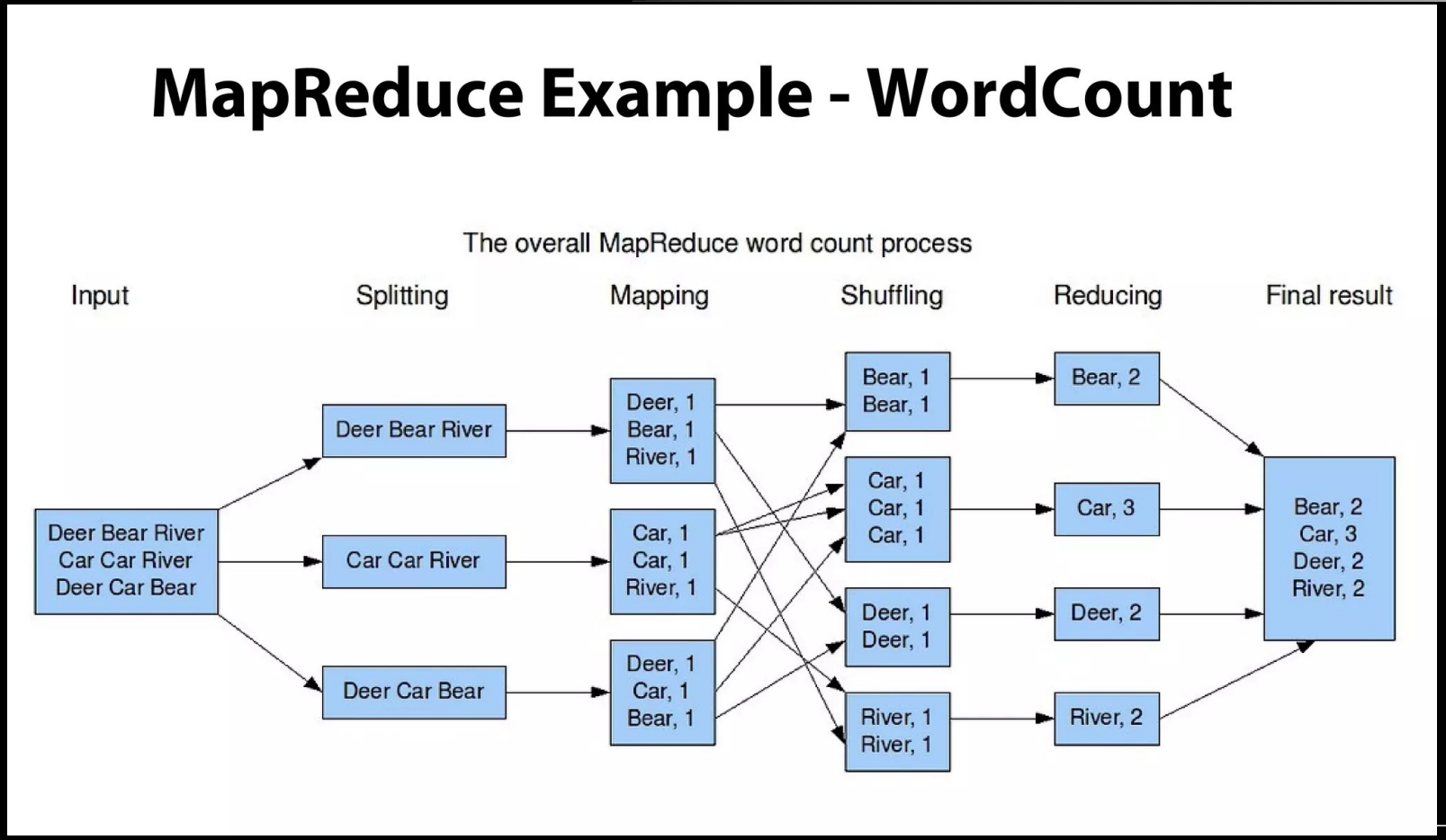
for me…

So, lets see implementation in Python

MapReduce Job – Logical View



How actually Wordcount works !!



HDFS

- Distributed File system
- Traditional Hierarchical file organization
- Single namespace for entire cluster
- Write-once-read-many access model

Hadoop Cluster

- A Small Hadoop Cluster include a
  - Single master and
  - Multiple worker nodes or slave nodes

Master node:
- Name node
- Secondary name node
- Job Tracker

Slave node:
- Data node
- Task Tracker

Hadoop architecture

- Hadoop has master slave architecture
- Typically one machine in the cluster is desiginated as the name node and another machine as job tracker, exclusively
  - *These are the masters*
- The rest of the machines in the cluster act as both data node and task tracker
  - *These are the masters*

Name node - Features

- Maintaines all the information (metadata)
- Records each data node for each block
- Free space
- Save address of every node
- File attributes, creation time
- Monitor data nodes by receiving heartbeats for every 30 seconds
- Executes file system namespace operation – opening, closing, renaming the directories

Hadoop architecture

- *JobTracker (master)*
  - *Receive jobs from client.*
  - *Talks to the NameNode to determine the location of the data*
  - *Manage and schedule the entire job*
  - *Split and assign tasks to slaves (TaskTrackers).*
  - *Monitor the slave nodes by receiving heartbeats.*
- *TaskTrackers (slaves)*
  - *Manage individual tasks assigned by the JobTracker, including Map operations and Reduce operations.*
  - *Every TaskTracker is configured with a set of slots, these indicate the number of tasks that it can accept.*
  - *Send out heartbeat messages to the JobTracker to tell that it is still alive.*
  - *Notify the JobTracker when succeeds or fails.*

Hadoop architecture

- *Redundant (3 copies)*
- *For large files – large blocks*
- *64 or 128 per block*
- *Can scale to 1000s of nodes*

Hadoop Cluster HDFS (Physical) Storage

One Name Node

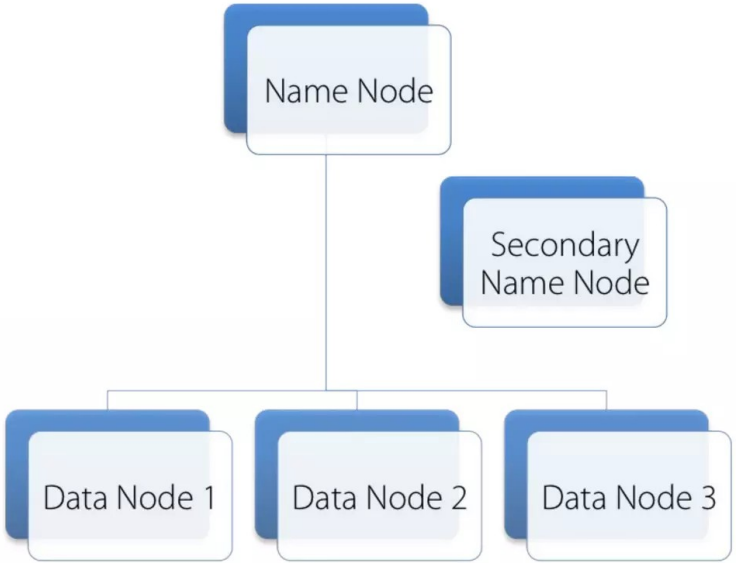
- Contains web site to view cluster information
- V2 Hadoop uses multiple Name Nodes for HA

Many Data Nodes

- 3 copies of each node by default

Work with data in HDFS

- Using common Linux shell commands
- Block size is 64 or 128 MB



Hadoop is set of Apache Framework and more...

Data Storage (HDFS)

- Runs on commodity hardware (usually linux)
- Horizontally scalable

Processing (Mapreduce)

- Parallelized (scalable) processing
- Fault tolerant

Examples of Hadoop mapreduce

distributed processing across multiple nodes in a Hadoop cluster

- Log Analysis
- Text Processing
- E-commerce Recommendations
- Social Network Analysis
- Genomic Data Analysis
- Image Processing
- Machine Learning
- Fraud Detection
- Large-scale Graph Processing
- Lot more...idk

Hadoop Daemons

Hadoop daemons are the processes or services that run on the various nodes of a Hadoop cluster to facilitate distributed storage and processing of large datasets

- Name node
- Secondary NameNode
- DataNode
- Resource Manager
- Node Manager
- Job Tracker
- Task Tracker

Hadoop Architecture