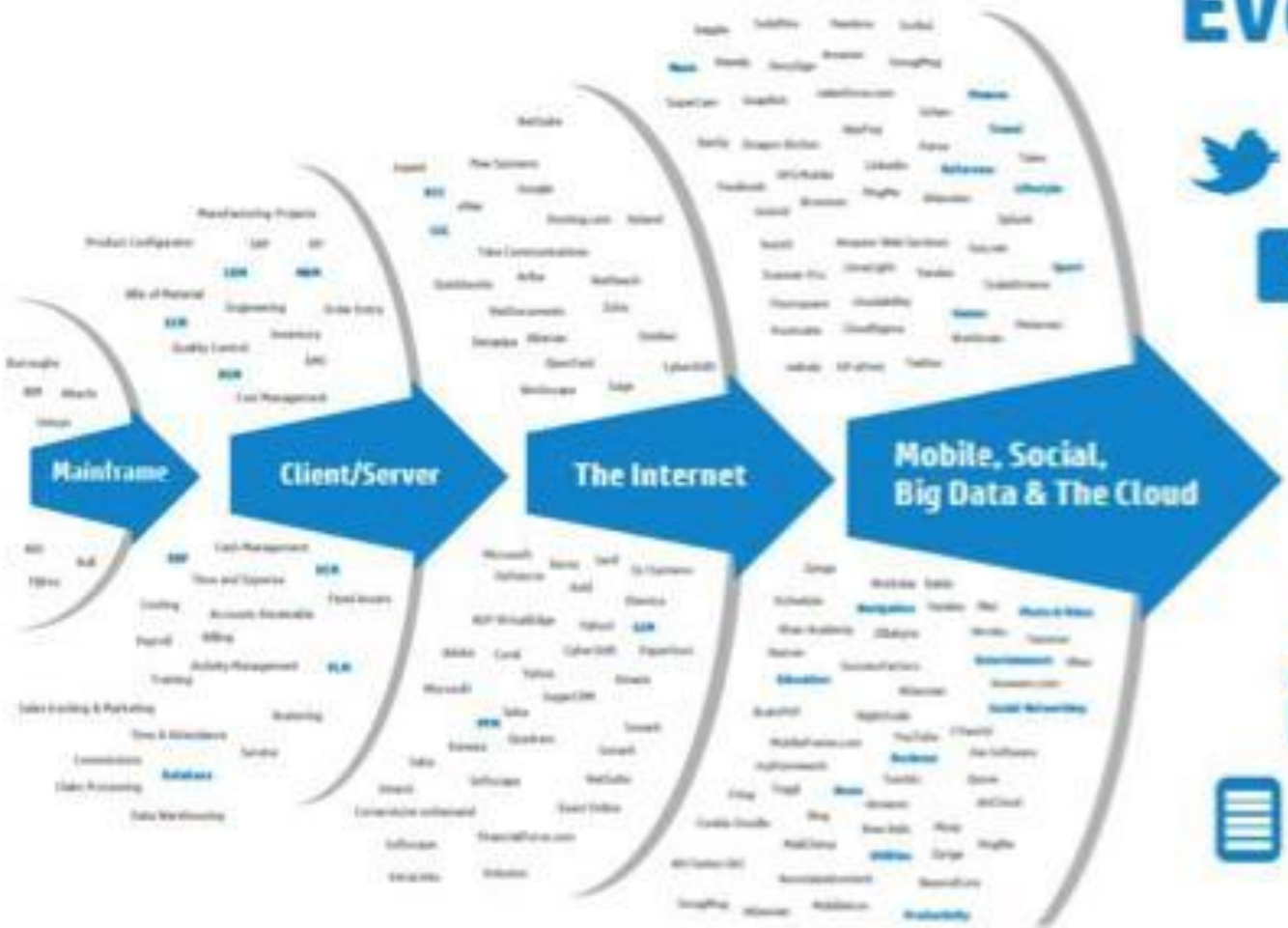
BIG DATA TESTING

Big Data

- **Big Data** is the data which is beyond the storage and processing capacity of a conventional database management systems is called “Big Data”.
- A Huge amount of data is generated daily in Peta Bytes, and data generation rate is rapidly increasing.

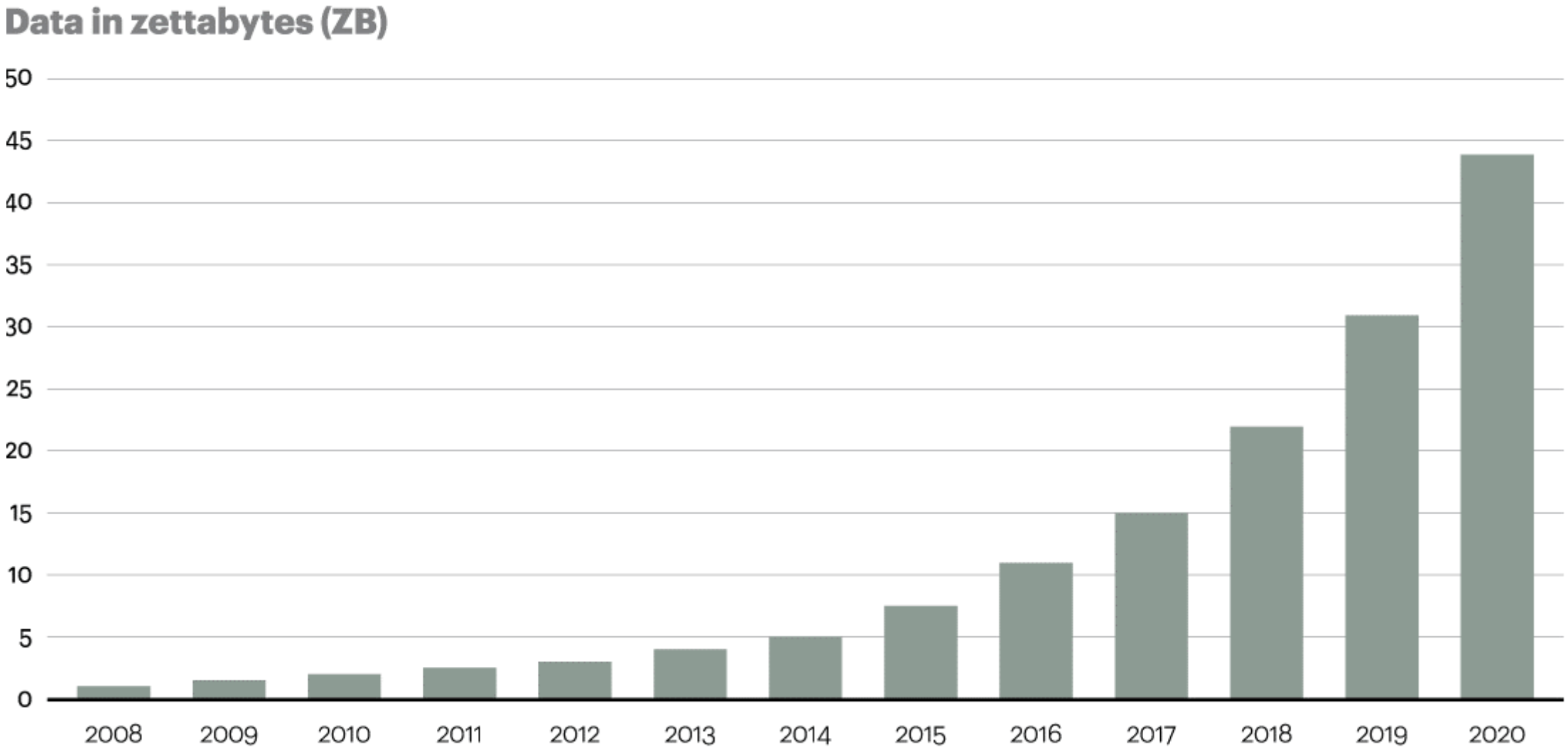
A new style of IT emerging



Every 60 seconds

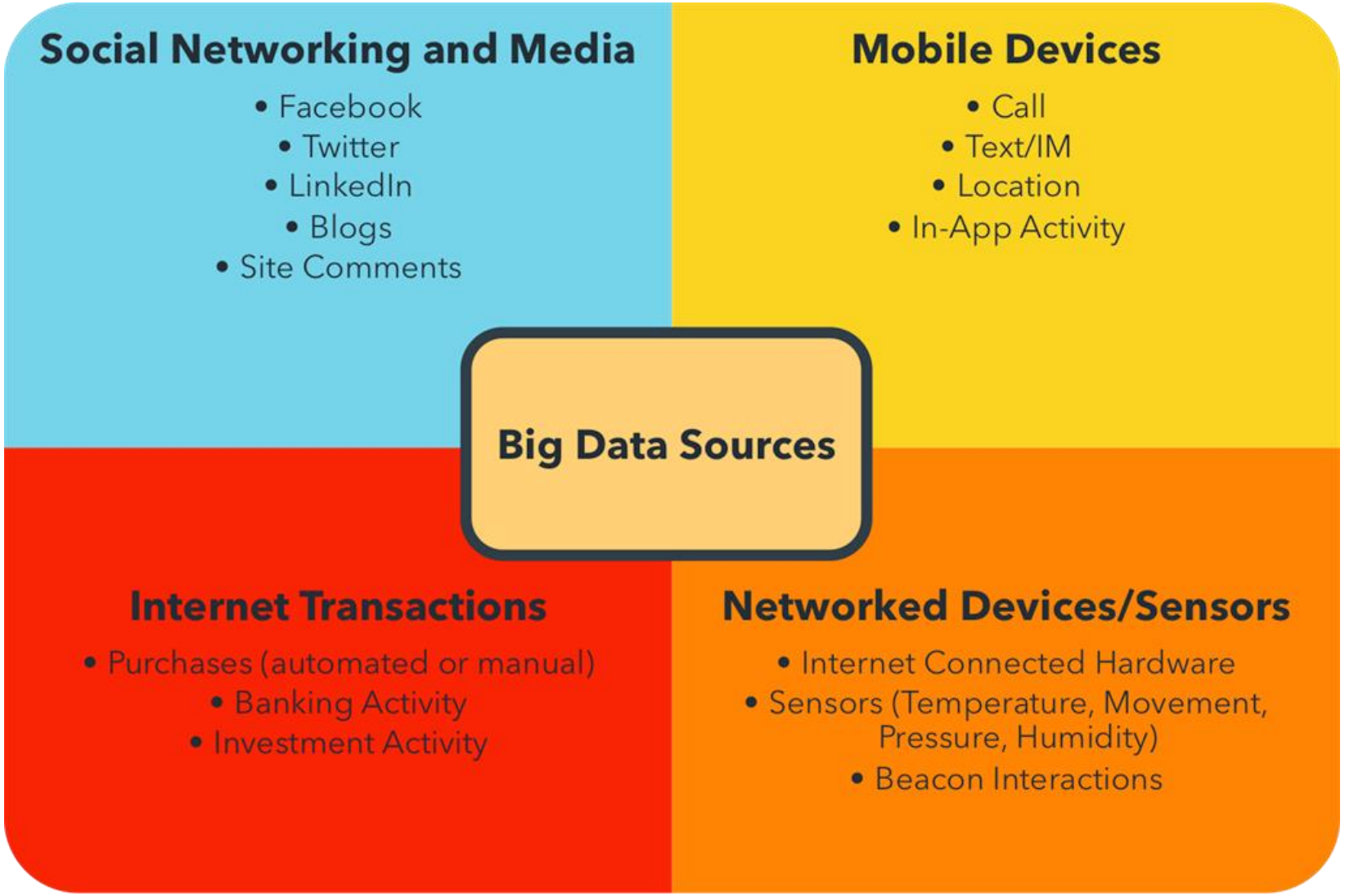
- **98,000+** tweets
- **695,000** status updates
- **11million** instant messages
- **698,445** Google searches
- **168 million+** emails sent
- **1,820TB** of data created
- **217** new mobile web users

Data is growing at a 40 percent compound annual rate, reaching nearly 45 ZB by 2020



Source: Oracle, 2012

Sources of Big Data



Big Data Characteristics

- Volume (size of data)
- Velocity (processing speed of data)
- Variety (types of data)

- If any data having above 3 Characteristics called as BIGDATA

Big Data Classification

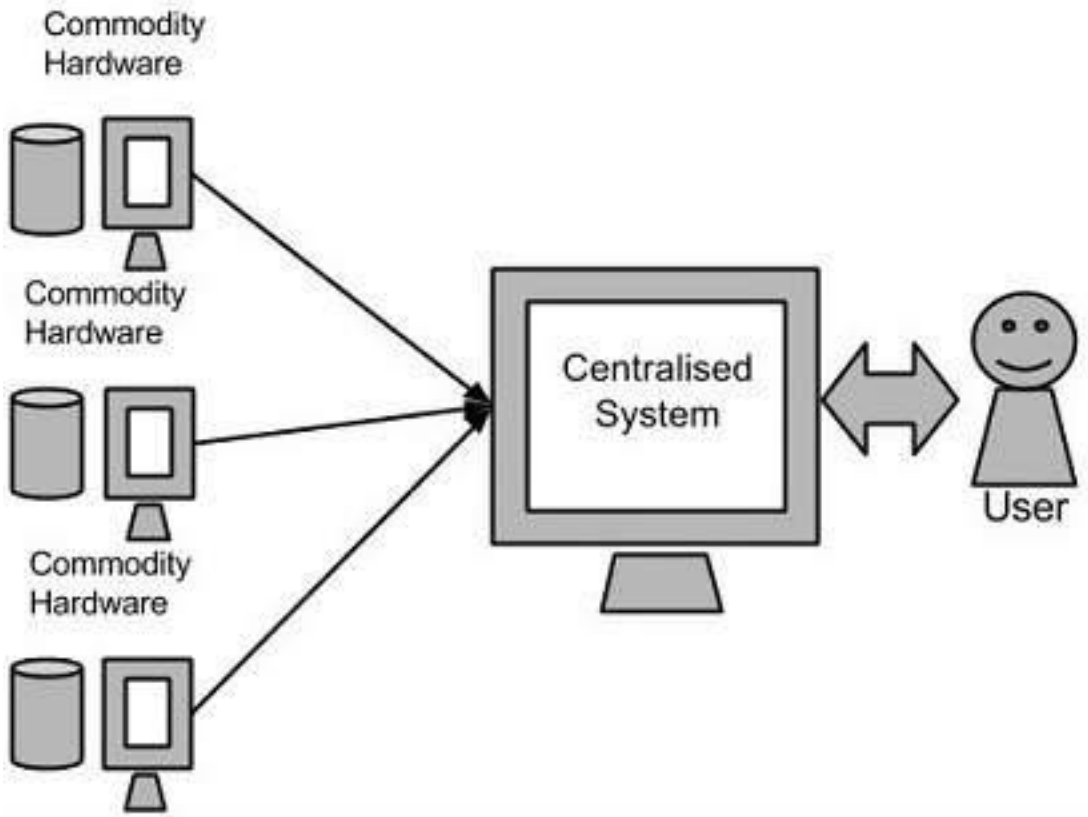
- **Structured data:**
  - Data which has proper structure and which can be easily stored in tabular form in any relational databases like MySQL, Oracle etc is known as structured data. Ex: Employee data
- **Semi Structured data:**
  - Data which has some structure but cannot be saved in a tabular form in relational databases is known as semi structured data. Ex: XML Data, email messages etc.
- **Unstructured data:**
  - Data which is not having any structure and cannot be saved in tabular form of relational databases is known as unstructured data. Ex: Video files, Audio files, Text file etc.

Challenges of Big Data

- Capturing data
- Storage
- Searching
- Sharing
- Transfer
- Analysis
- Presentation

Google’s Solution

- Google solved this problem using an algorithm called **MapReduce**.
- This algorithm divides the task into small parts and assigns those parts to many computers connected over the network, and collects the results to form the final result dataset.



History of Hadoop

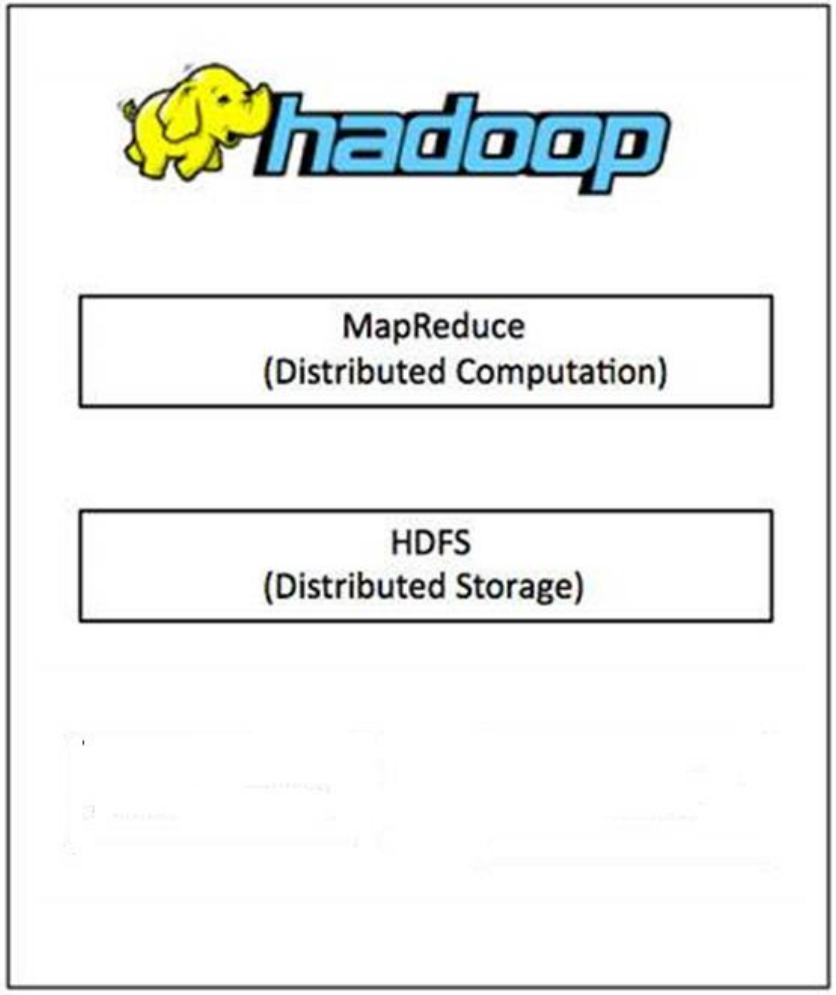
- Google got huge data in 1990's. They thought how to manage data.
- Came up with idea after 13 years
  - 2003- Google File System (GFS) – For storage
  - 2004- Map Reduce (For processing)
- Yahoo
  - 2007 – HDFS (Hadoop Distribute File System)
  - 2007-08 (Map Reduce)
- Doug Cutting, who was working at Yahoo! at the time, named it after his son's toy elephant.

What is Hadoop?

- Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment.
- It is part of the Apache project sponsored by the Apache Software Foundation.
- Hadoop used commodity(cheaper) hardware and clusters concepts.
- Recommended for huge datasets but not smaller datasets.

Hadoop Architecture

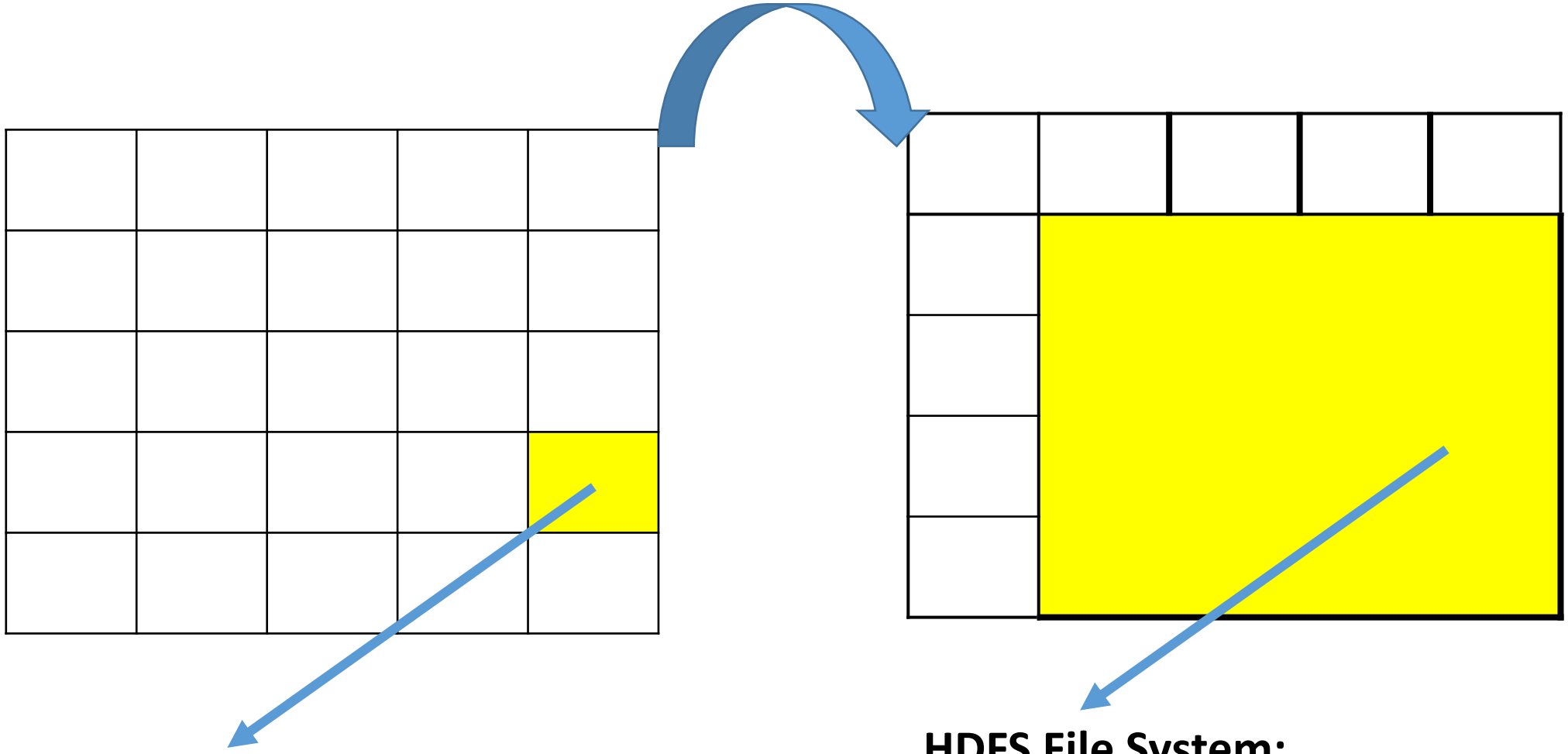
- Hadoop Components:
- **MapReduce**
- Processing data which is stored on HDFS.
- **HDFS**
- Storing huge data sets.



HDFS (Hadoop File System)

- HDFS is a specially designed file system for storing huge data set with cluster of commodity hardware with streaming access pattern.
- Cluster : Group of Systems connected to LAN
- Commodity H/W : Cheap Hardware
- Streaming access pattern : Streaming access pattern means you can write once, read any number of times but can't change the content of that file once it is kept in HDFS.
- WORM (Write Once Read Many)

Why Hadoop is specially designed file system?



Normal File System:
Default Block size is 4 KB.

HDFS File System:
Default Block size is 128 MB.

Difference between Unix file system and HDFS?

- **In Unix file system** default size of the block is of 4KB. Suppose your file is of 6KB, then you require 2 blocks in Unix each of 4KB. So total 8KB is used, but actually you require 6KB that extra 2KB is wasted.

- **In HDFS default** size of the block is of 128MB. Suppose your file is of 200MB, then HDFS requires 2 blocks (1 of 128MB + 1 of 73MB). Extra space of 50MB in last block is not engaged. Instead extra space is relieved.

HDFS Services(Daemons/Nodes)

- HDFS Have 5 services
- **Master services**

1. Name node
2. Secondary name node
3. Job Tracker (Resource Manger)

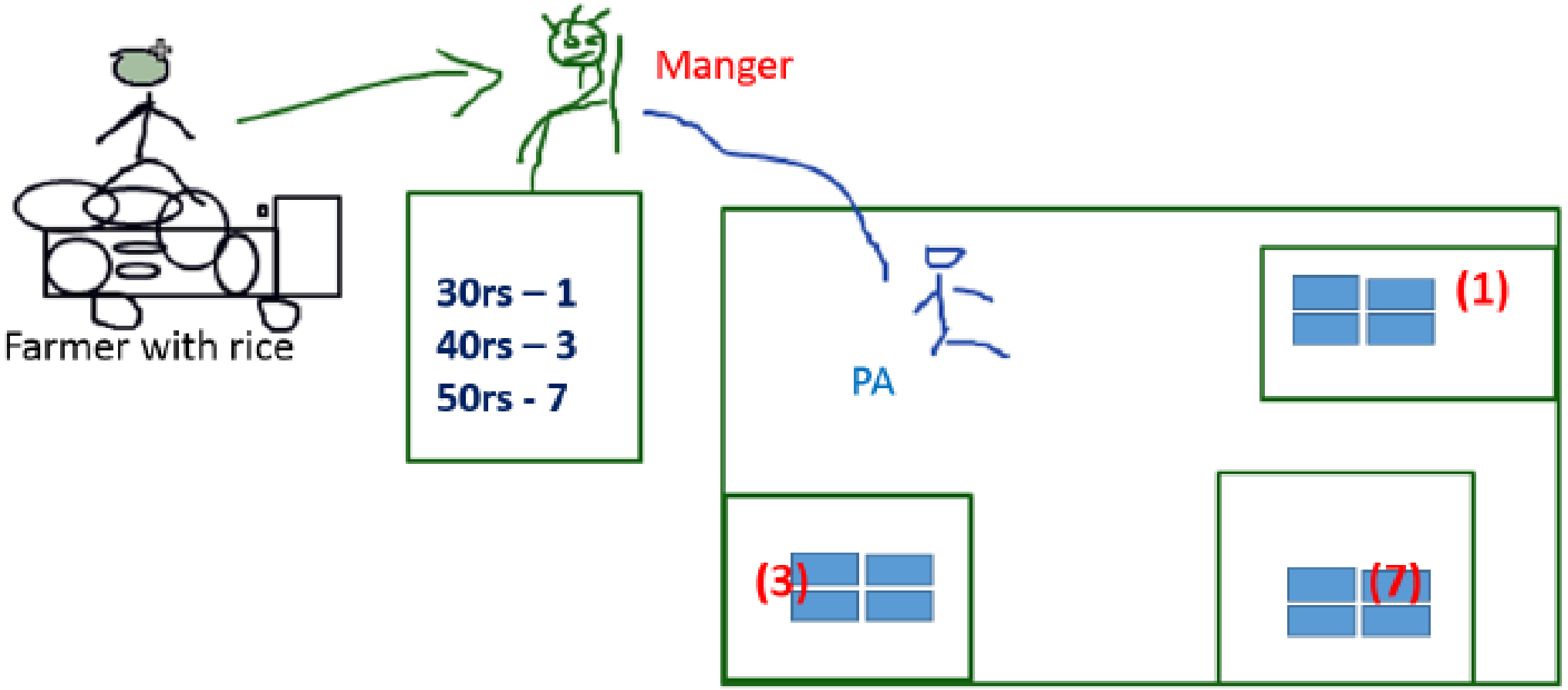
- **Slave Services**

4. Data node
5. Task tracker (Node Manger)

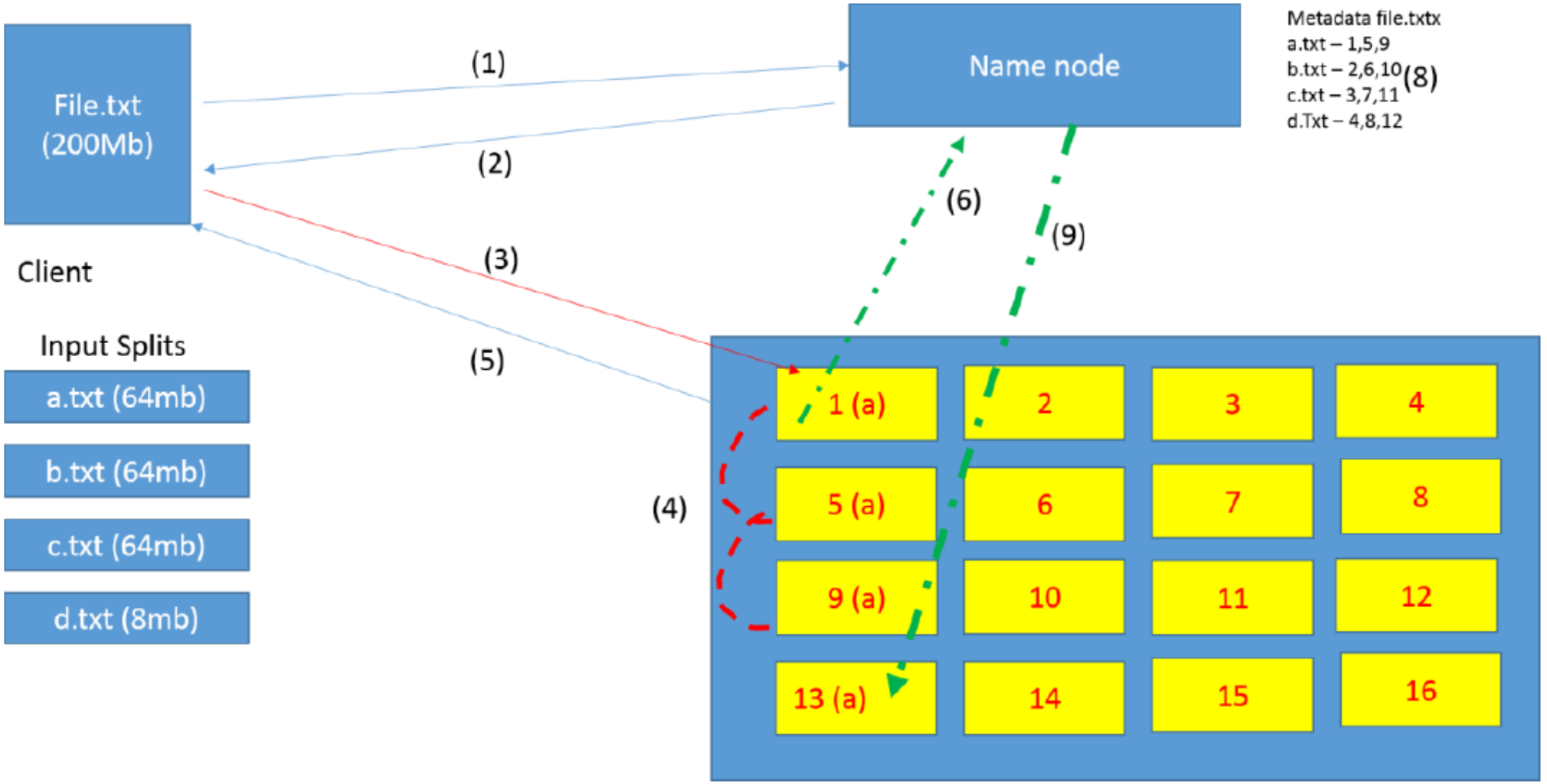
- First **3 services are master services** and the **last 2 services are slave services**.
- Master service can communicate with each other. Similarly true for slave services.
- In master-slave communication, a **NameNode can only communicate with a DataNode** and a **JobTracker can only communicate with a TaskTracker**.
- If a NameNode fail, SecondaryNameNode is available. However the failure of JobTracker is the single point of failure in hadoop architecture.

General Use case

One day a farmer want to store his rice packets into a godown. We went to godown and contacted manager. Manager asked what type of rice packets you have. He said 50 packs of 30rs, 50 of 40rs, 100 of 50rs. He noted down on a bill paper he called his PA to store these 30rs paks in 1st room, 40rs pacs in 3rd room, 50rs pacs in 7th room.


Farmer with rice
Manger
30rs – 1
40rs – 3
50rs - 7
PA
(1)
(3)
(7)

Storing/Writing data in HDFS



Storing data into HDFS

- Client want to store file.txt 200Mb into HDFS
- File.txt is divided into 64mb of sub files. These are called as **Input Splits**.
- Client send these to NameNode. Namenode reads the metadata , and sends available datanodes to client.
- Client send each input split to datanodes.
- A.txt file stores into datanode-1
- And HDFS also maintains the replica of data. by default replica is 3.
- So a.txt is stored in datanodes 5,9 as well
- Acknowledgment is sent to 9 to 5, 5 to 1, 1 to client.
- And the response is sent to client.
- For every storage operation datanode send Block Report and Heartbeat to namenode.
- Datanode sent heartbeat to namenode for every short period of time.
- If any datanode fails, namenode copies that data to another datanode

MapReduce (Retrieving Data)

- If you want process data we write one program(10KB)
- Processing(it is called as Map)
- client submits commands/program to JobTracker for applying job on a.txt
- JobTracker asks Namenode
- namenode replies metadata to JobTracker
- JobTracker knows on which blocks data is stored in Datanodes
- A.txt stored in 1,3,9
- Using metadata Job tracker contacts Tasktracker near to JobTracker(datanode-1)

- JobTracker assigns this task to Tasktracker
- Tasktracker apply this job on a.txt. this is called “**MAP**”
- Similarly on b.txt, c.txt, d.txt Tasktracker runs program, called **MAP**
- **a,b,c,d txt files called Input Splits**
- **No. of Input Splits = No. of Mappers**
- If any Tasktracker is not able to do the job, it informs JobTracker.
- JobTracker assigns to nearest another Tasktracker.
- for every 3secs Tasktracker send heartbeat to JobTracker.
- if 10sec also if JobTracker doesn’t get heartbeat, it thinks Tasktracker is dead or slow .

- Tasktracker may slow if it running more jobs ex. 100 jobs.
- JobTracker knows the how many jobs running by Tasktracker .
- if any Tasktracker is busy , JobTracker chooses another Tasktracker .
- After applying program, a.txt returns 10kb of data, b.txt 10kb, c 10kb, d 4kb.
- After getting outputs **Reducer** combines all these output data to file.
- **No of Reducers = No. of Output files .**
- **The Output file may save in local data node, or some other node.**
- The Datanode sends heartbeat to Namenode where output file is saved.
- NameNode save this metadata .
- Client contacts NameNode for Output file location .

MapReduce Examples

- **Steps in MapReduce:**

1. Main job is splitting into sub-jobs.

2. Map this sub-jobs to different CPUs/Processor

3.Collect the output from the different processors or Mappers.

4.Reducing the output to produce the final result.

MapReduce Example

- **Mapping of MapReduce**

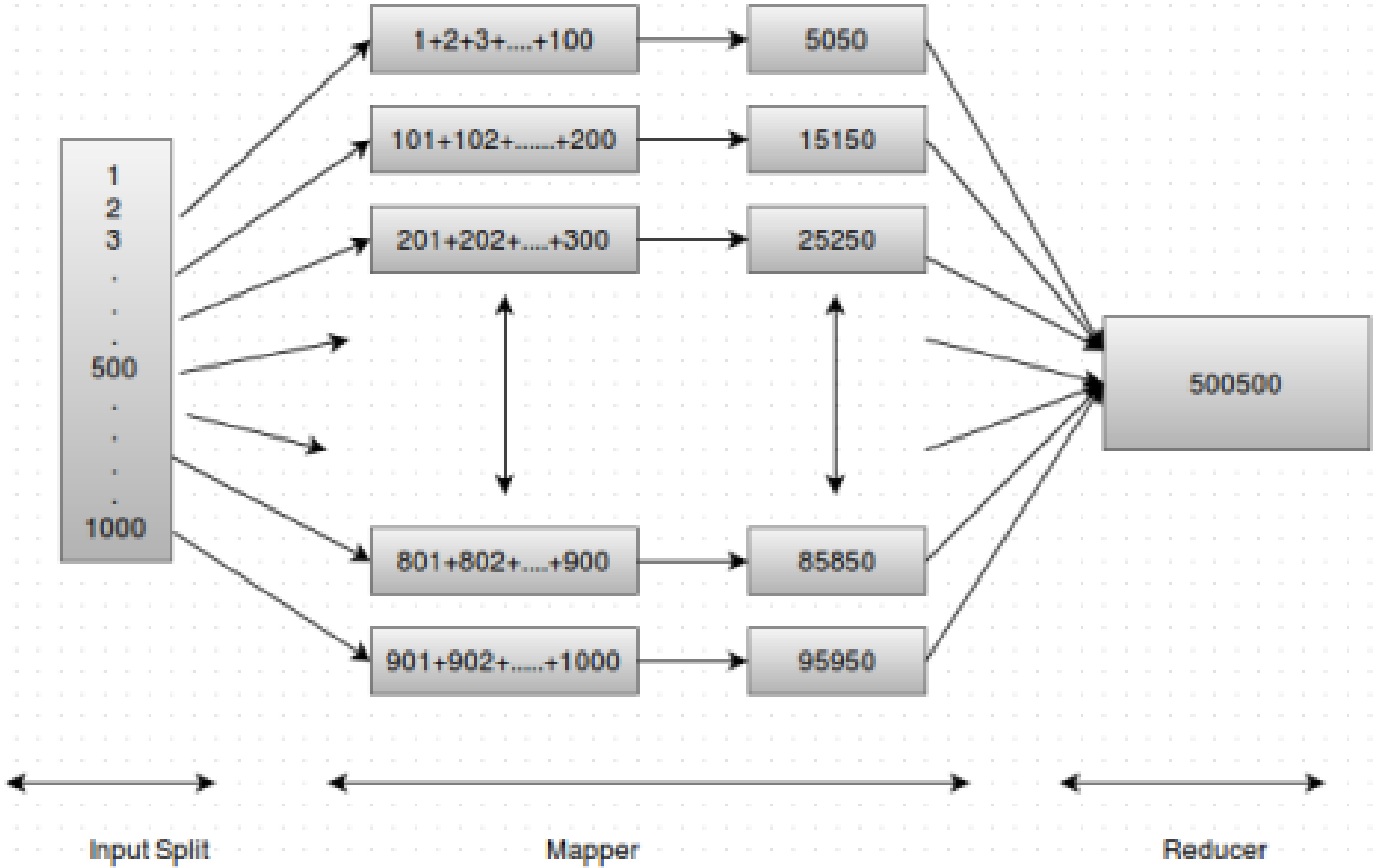
  Sum of 1000 numbers.

  We have 10 members with us.

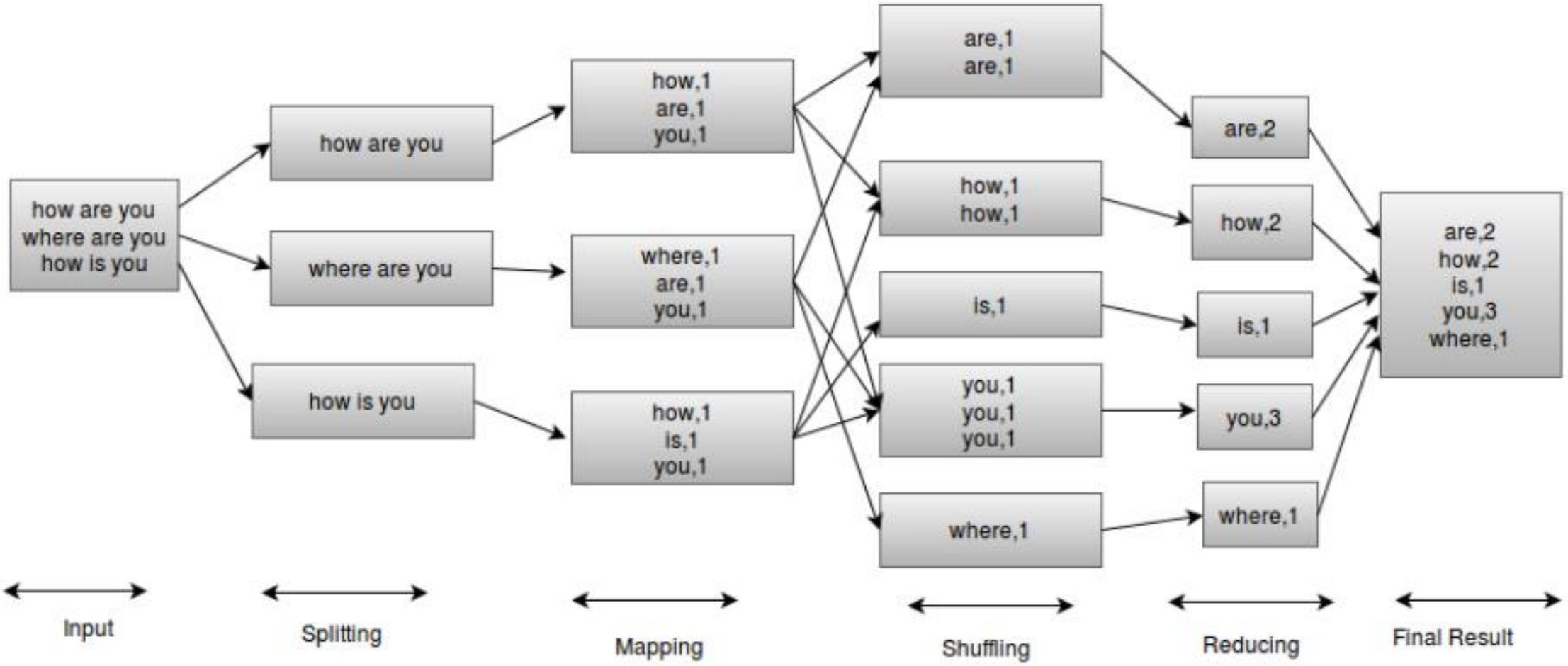
  1. Input Splitting – Divide 1000 numbers in 10 peoples.

  2. Map Method – Each member perform addition of 100 numbers.

  3. Reduce Method – After addition performed by all 10 members, Collect this addition to single person and again perform addition of this 10 collected numbers and we get final output.


1
2
3
.
.
.
500
.
.
.
.
1000
1+2+3+....+100
101+102+.....+200
201+202+....+300
801+802+....+900
901+902+.....+1000
5050
15150
25250
85850
95950
500500
Input Split
Mapper
Reducer

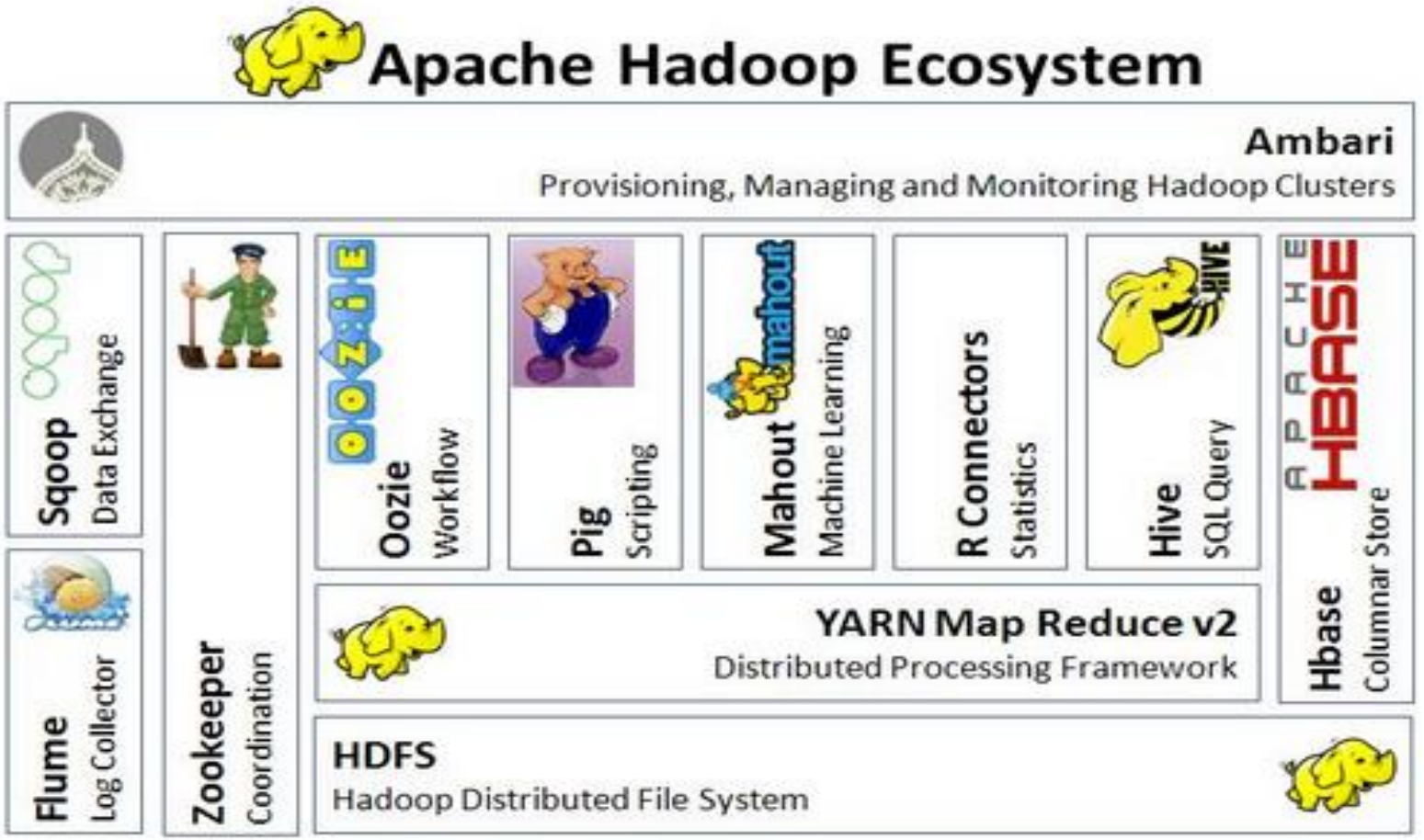
Word Count example in Map-Reduce format



Hadoop Distributions

Hadoop Ecosystem

- The Hadoop ecosystem contains different sub-projects (tools) such as Sqoop, Pig, and Hive that are used to help Hadoop modules.



Hadoop Running Modes

- **Standalone Mode**
  - Default mode of Hadoop
  - HDFS is not utilized in this mode.
  - Local file system is used for input and output
  - Used for debugging purpose
  - No Custom Configuration is required in 3 hadoop (mapred-site.xml,core-site.xml, hdfs-site.xml) files.
  - Standalone mode is much faster than Pseudo-distributed mode.

- **Pseudo Distributed Mode (Single Node Cluster)**
  - Configuration is required in given 3 files for this mode
  - Replication factory is one for HDFS.
  - Here one node will be used as Master Node / Data Node / Job Tracker / Task Tracker
  - Used for Real Code to test in HDFS.
  - Pseudo distributed cluster is a cluster where all daemons are running on one node itself.

- **Fully distributed mode (or multiple node cluster)**
- This is a Production Phase
- Data are used and distributed across many nodes.
- Different Nodes will be used as Master Node / Data Node / Job Tracker / Task Tracker

Accessing HDFS

- We can access HDFS in 3 different ways.
  1. CLI mode
  2. GUI mode using thick clients
  3. Browser

Cloudera Quick VM Download

- http://www.cloudera.com/downloads/quickstart_vms/5-8.html

Local File System Commands on Linux

Adobe Acrobat
Document

HDFS Commands on Linux

Adobe Acrobat
Document