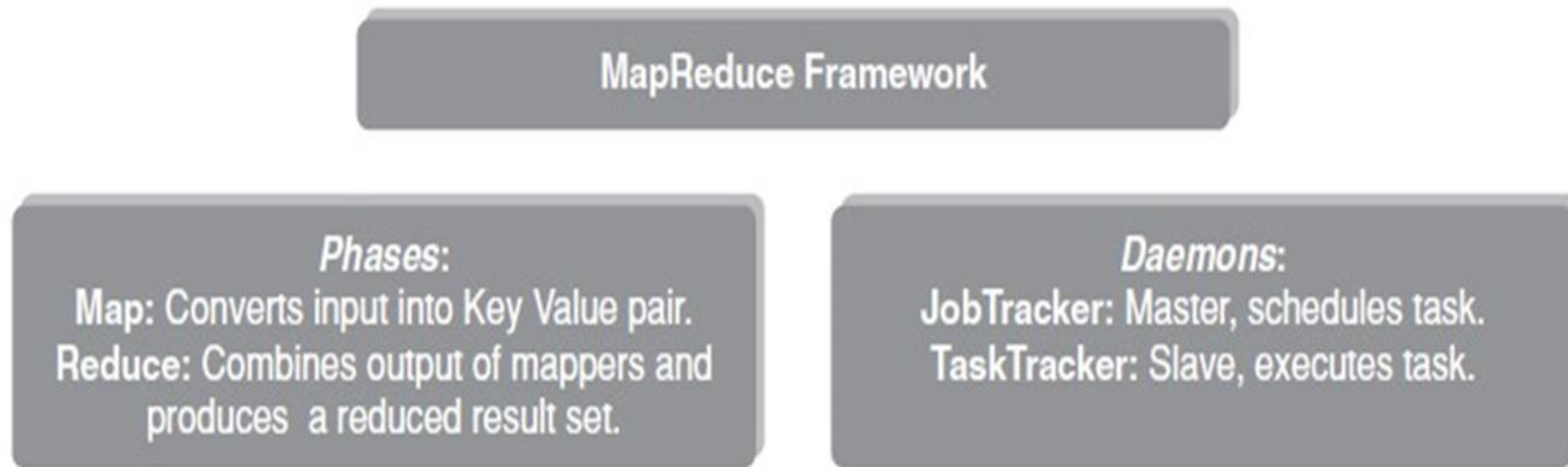


# **Data Processing with Hadoop- MapReduce Concepts**

A.Baskar

# Data Processing with Hadoop

- MapReduce Programming is a software framework.
- MapReduce Programming helps you to process massive amounts of data in parallel.



# MapReduce

- In MR programming the input data is split into smaller chunks (64MB)
- **Map Tasks** process these chunks in parallel.
- The output produced by this task is <Key, Value> which will be intermediate output and is stored in local disk of the server.
- The output of the mapper is automatically shuffled and sorted based on the key.
- The final output becomes input to the Reducer Phase.

# MapReduce

- **Reduce task** provides the output by combining various mapper output.
- Job Input and Output stored in file systems.
- MR tasks also take care of other functionalities like ,
  - Monitoring
  - Scheduling
  - Re-executing failed tasks

HDFS and MR framework run on same set of nodes. This configurations allows effective scheduling of task for the data present. This is called **Data Locality**. This results in very high throughput.

# MapReduce Daemons

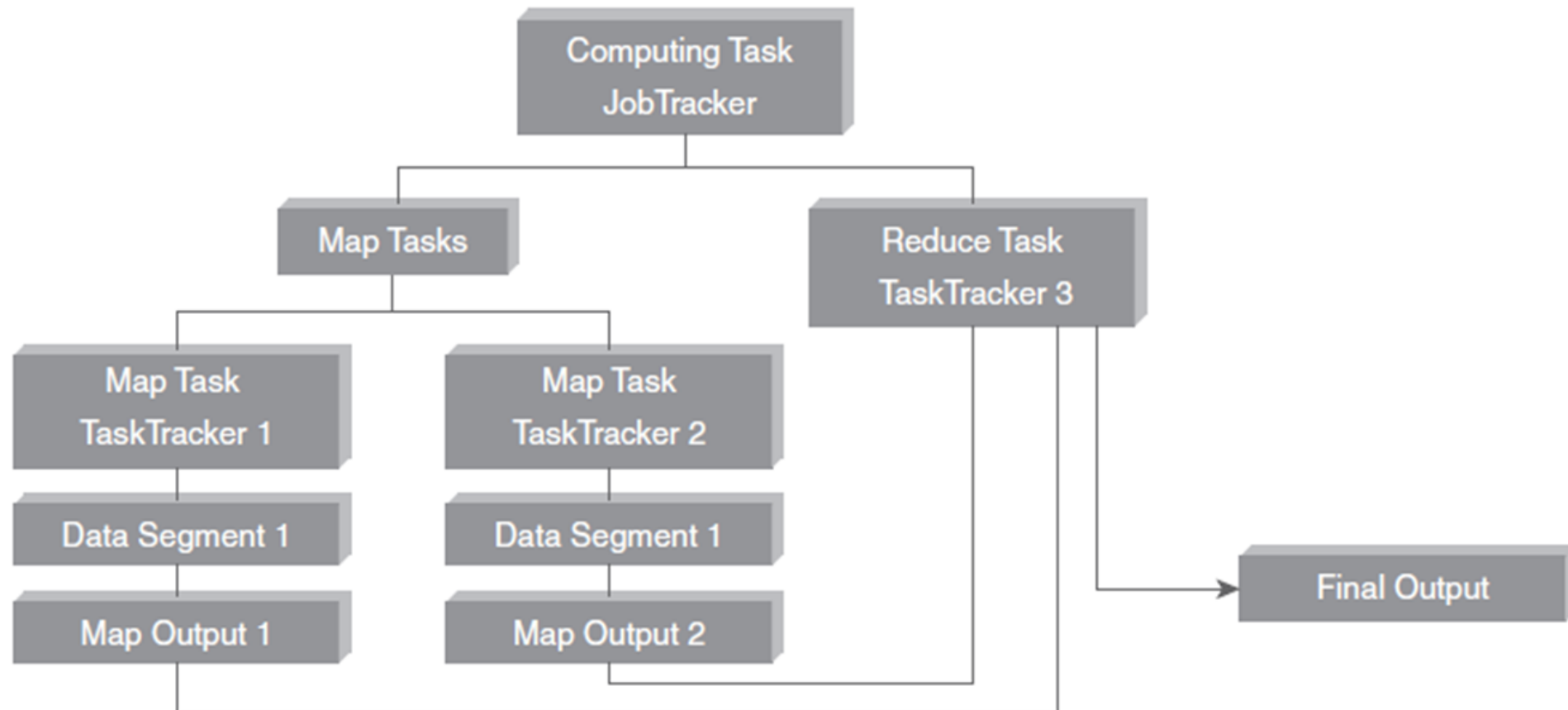
- **Job Tracker**
  - A single Job tracker per cluster.
  - Job tracker is responsible for scheduling the jobs to task tracker and monitor the tasks and re-executing the failed task.
  - It creates a connection between Hadoop and your application. When you submit the code the job tracker creates a execution plan.
  - Job tracker is responsible for over all MR task.

# MapReduce Daemons

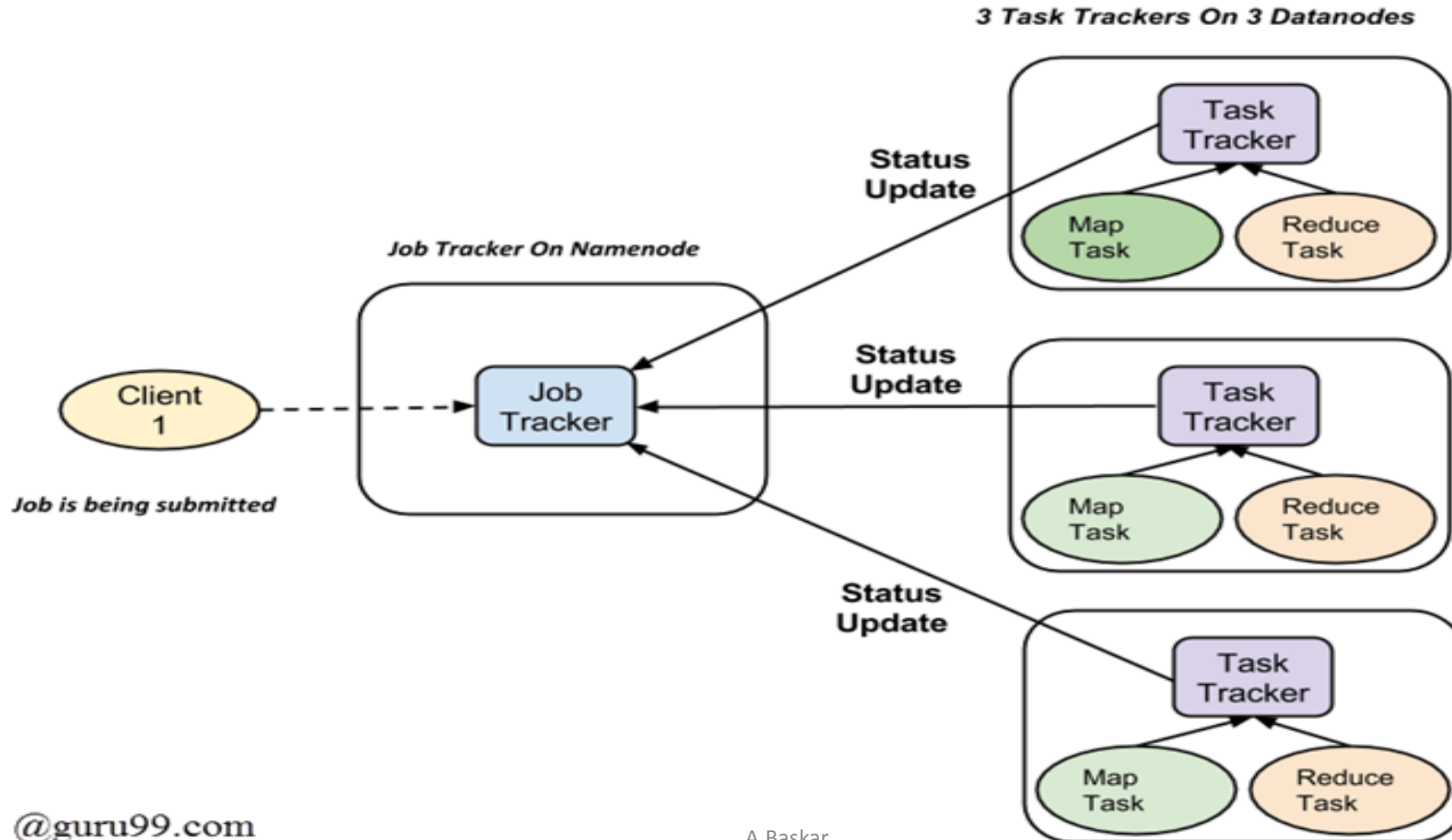
- **Task Tracker**

- Task tracker is responsible for executing the task assigned by Job tracker.
- There is a single task tracker per slave and spawns multiple JVM to execute multiple map and reduce.
- Task tracker continuously send heartbeat to job tracker.
- If Job tracker failed to receive heartbeat then it assumes TT is failed then re assign the job to another TT which holds the same data.

# How MapReduce Programming Workflow



# How MapReduce Organizes Work?





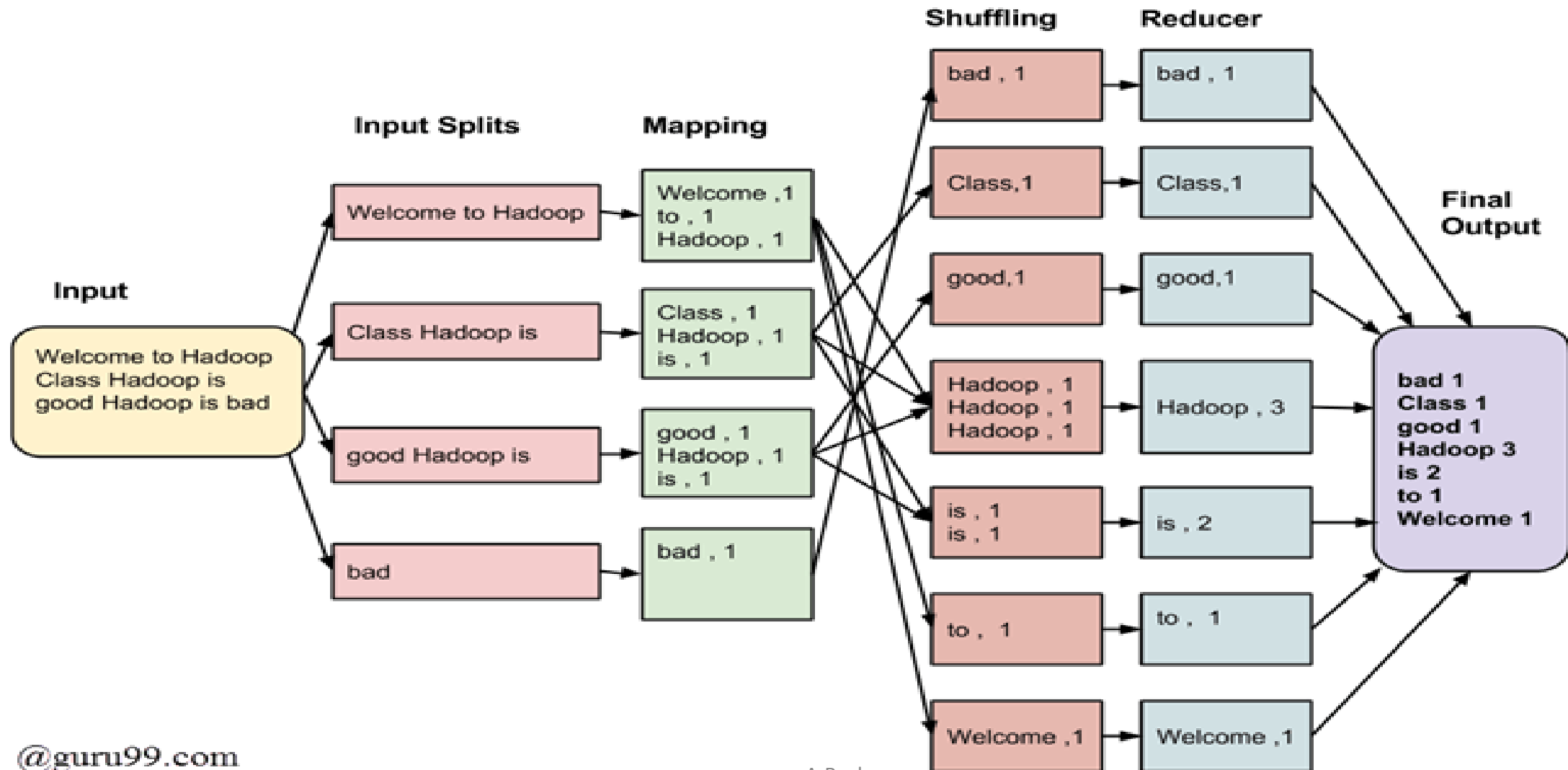
# How MapReduce Organizes Work?

- The complete execution process (execution of Map and Reduce tasks, both) is controlled by two types of entities called a
- **Jobtracker** : Acts like a **master** (responsible for complete execution of submitted job)
- **Multiple Task Trackers** : Acts like **slaves**, each of them performing the job
- For every job submitted for execution in the system, there is one **Jobtracker** that resides on **Namenode** and there are **multiple tasktrackers** which reside on **Datanode**.

# How MapReduce Organizes Work?

- A job is divided into multiple tasks which are then run onto multiple data nodes in a cluster.
- It is the responsibility of job tracker to coordinate the activity by scheduling tasks to run on different data nodes.
- Execution of individual task is then look after by task tracker, which resides on every data node executing part of the job.
- Task tracker's responsibility is to send the progress report to the job tracker.
- In addition, task tracker periodically sends '**heartbeat**' signal to the Job tracker so as to notify him of current state of the system.
- Thus job tracker keeps track of overall progress of each job. In the event of task failure, the job tracker can reschedule it on a different task tracker.

# MapReduce – Word Count Example



# MapReduce Programming

- In MapReduce Programming, Jobs (Applications) are split into a set of map tasks and reduce tasks. Then these tasks are executed in a distributed fashion on Hadoop cluster.
- Each task processes small subset of data that has been assigned to it. This way, Hadoop distributes the load across the cluster.
- MapReduce job takes a set of files that is stored in HDFS (Hadoop Distributed File System) as input.
- Map task takes care of loading, transforming, parsing and filtering
- Reduce task is responsible for grouping and aggregation.

# MapReduce Programming- Mapper

- A mapper maps the input key–value pairs into a set of intermediate key–value pairs. Maps are individual tasks that have the responsibility of transforming input records into intermediate key–value pairs.
- Mapper Consists of following phases:
  - **RecordReader**
  - **Map**
  - **Combiner**
  - **Partitioner**

# MapReduce Programming-Mapper

## Record Reader:

- 1.It converts a byte- oriented view of the input into record –oriented view and presents to mapper tasks
- 2.It presents task with key- value pair
3. Key: positional information
4. Value is chunk of data that constitutes the record.

## Map:

1. works on the key- value pair
- 2.generates zero or more intermediate key- value pairs

## Combiner:

1. It is an optional function , but it provides high performance.(bandwidth and disk space)
2. It takes intermediate key- value pair and applies user specified aggregate function only on that mapper
3. It is also known as local reducer.

# MapReduce Programming-Mapper

- **Partitioner:**

- 1.It takes the intermediate key-value pair and splits them into shard, sends the shard to the particular reducer as per the user specific code.
- 2.Key with same value goes to the same reducer.

# MapReduce Programming-Reducer

- The primary chore of the Reducer is to reduce a set of intermediate values (the ones that share a common key) to a smaller set of values.
- The Reducer has three primary phases:
  - Shuffle
  - Sort
  - Reduce
  - Output Format.
- Hadoop assigns map tasks to Data Node where the actual data to be processed.
- Hadoop ensures data locality.(moving code not data).



# MapReduce Programming-Reducer

- **Shuffle and Sort:**

1. This phase takes the output of all practitioners and downloads them into the local machine where the reducer is running.
2. These individual data pipes are sorted by key.
3. Main purpose of sort is grouping similar words so that their values can be easily iterated by reduce task.

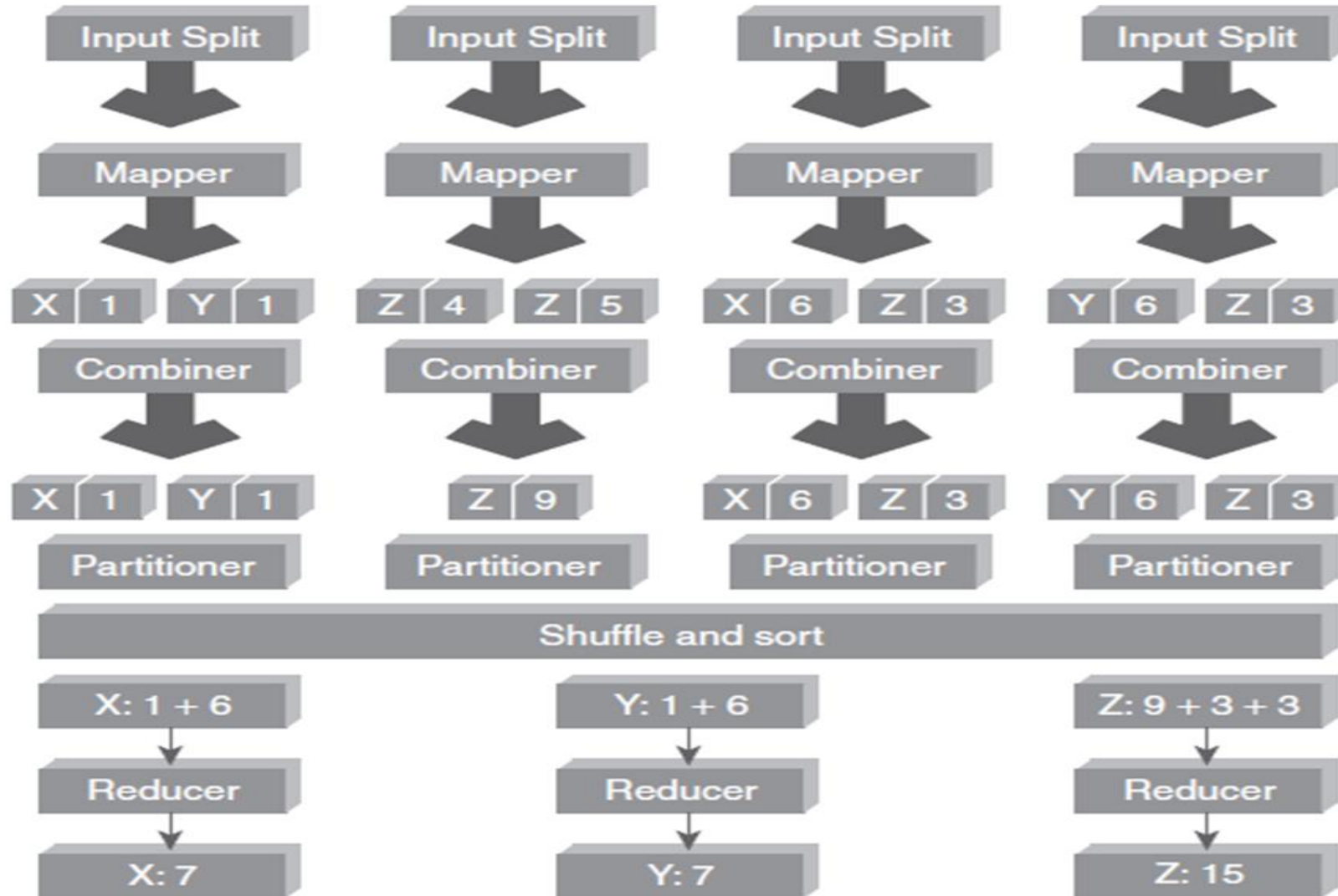
- **Reduce:**

1. It takes the grouped data f, applies the reduce function, and processes one group at a time .
2. The reduce function iterates all the values associated with that key.
- 3.It provides various operations like aggregation, filtering and combining data.
4. Once it is done is sent to output format.

- **Output Format:**

- 1.It separates key- value pair with tab and writes it out to a file using record write.

# The chores of Mapper, Combiner, Partitioner, and Reducer



# Combiner

- It is an optimization technique for MapReduce Job. Generally, the reducer class is set to be the combiner class. The difference between combiner class and reducer class is as follows:
- Output generated by combiner is intermediate data and it is passed to the reducer.
- Output of the reducer is passed to the output file on disk.

# Partitioner

- The partitioning phase happens after map phase and before reduce phase. Usually the number of partitions are equal to the number of reducers. The default Partitioner is hash Partitioner.

# Programming Demo