Introduction to Hadoop and MapReduce Programming - Session 3

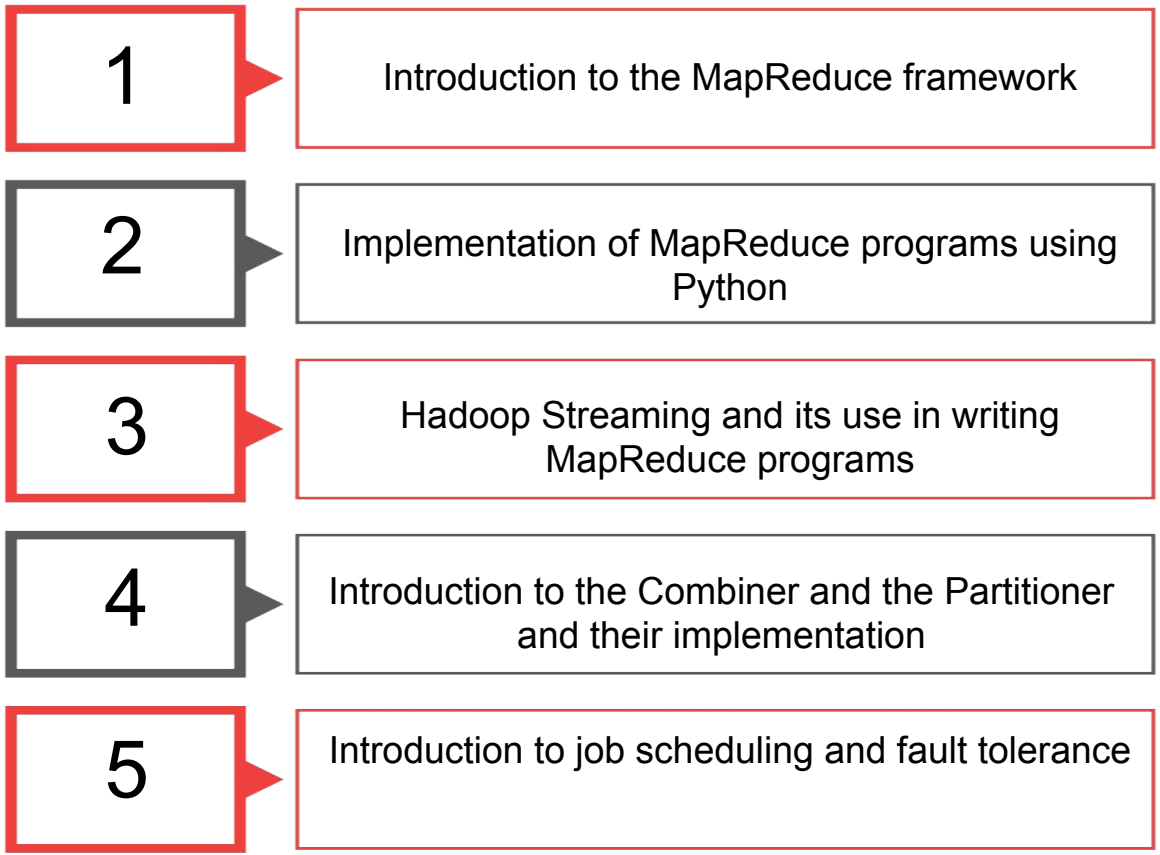
Course: Data Engineering - I

Lecture On: MapReduce Programming

Instructor: Vishwa Mohan

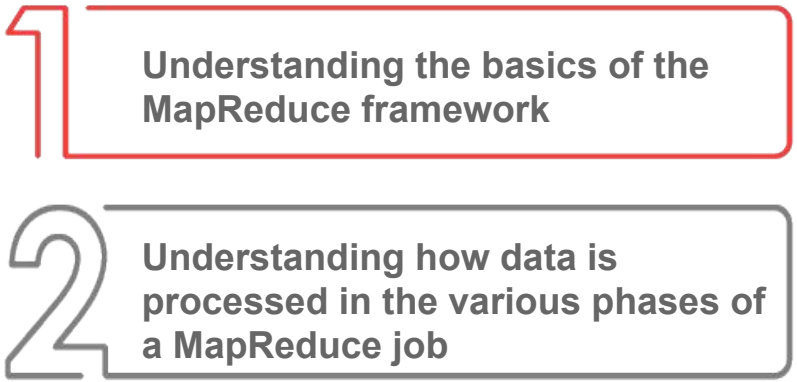
Segment - 01
Introduction: MapReduce Programming

Session Overview



Segment - 02
Introduction to the MapReduce Framework

Segment Overview



Introduction to the MapReduce Framework

MapReduce is a programming model used for processing a large amount of data. It is the **data-processing layer of Hadoop.**

A MapReduce program consists of two phases, and it writes separate scripts for these phases.

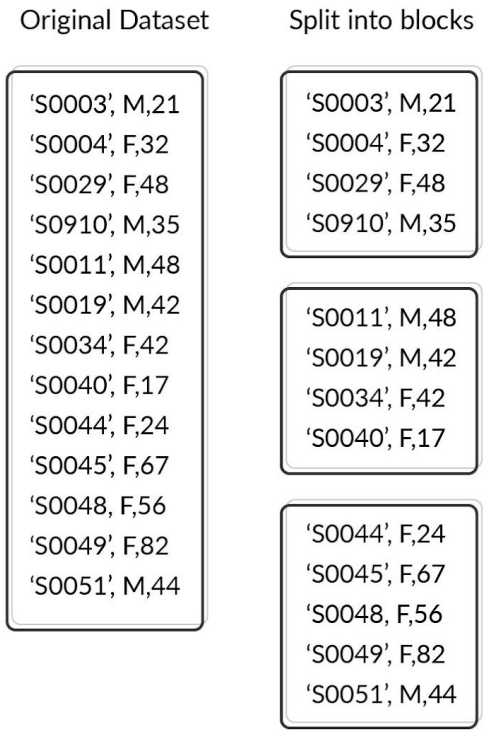
- **The Map phase:** In this phase, the script, also known as the Mapper, transforms the data into a key-value pair.
- **The Reduce phase:** In this phase, the script, also known as the Reducer, aggregates the processed data and yield the output as specified by the user.

Introduction to the MapReduce Framework

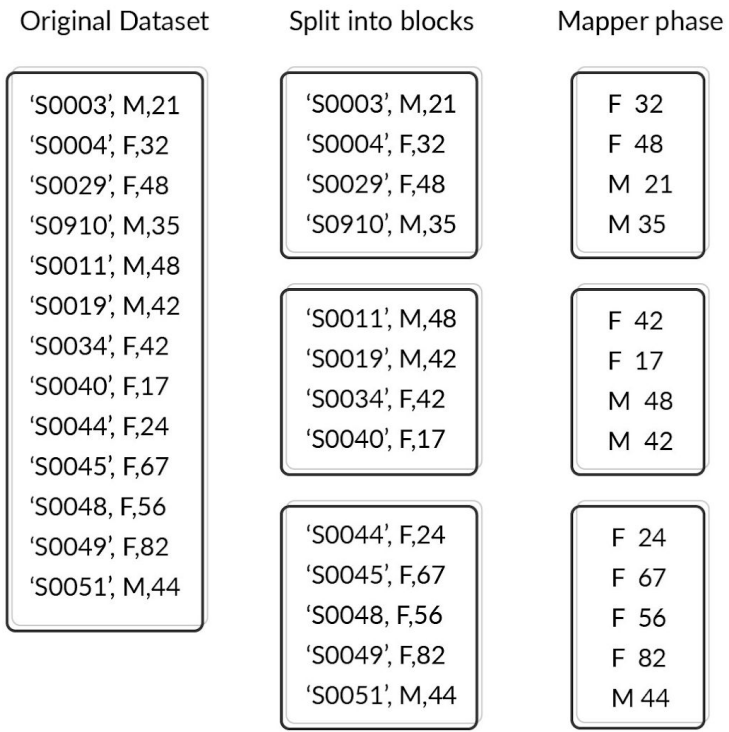
Original Dataset

‘S0003’, M,21
‘S0004’, F,32
‘S0029’, F,48
‘S0910’, M,35
‘S0011’, M,48
‘S0019’, M,42
‘S0034’, F,42
‘S0040’, F,17
‘S0044’, F,24
‘S0045’, F,67
‘S0048, F,56
‘S0049’, F,82
‘S0051’, M,44

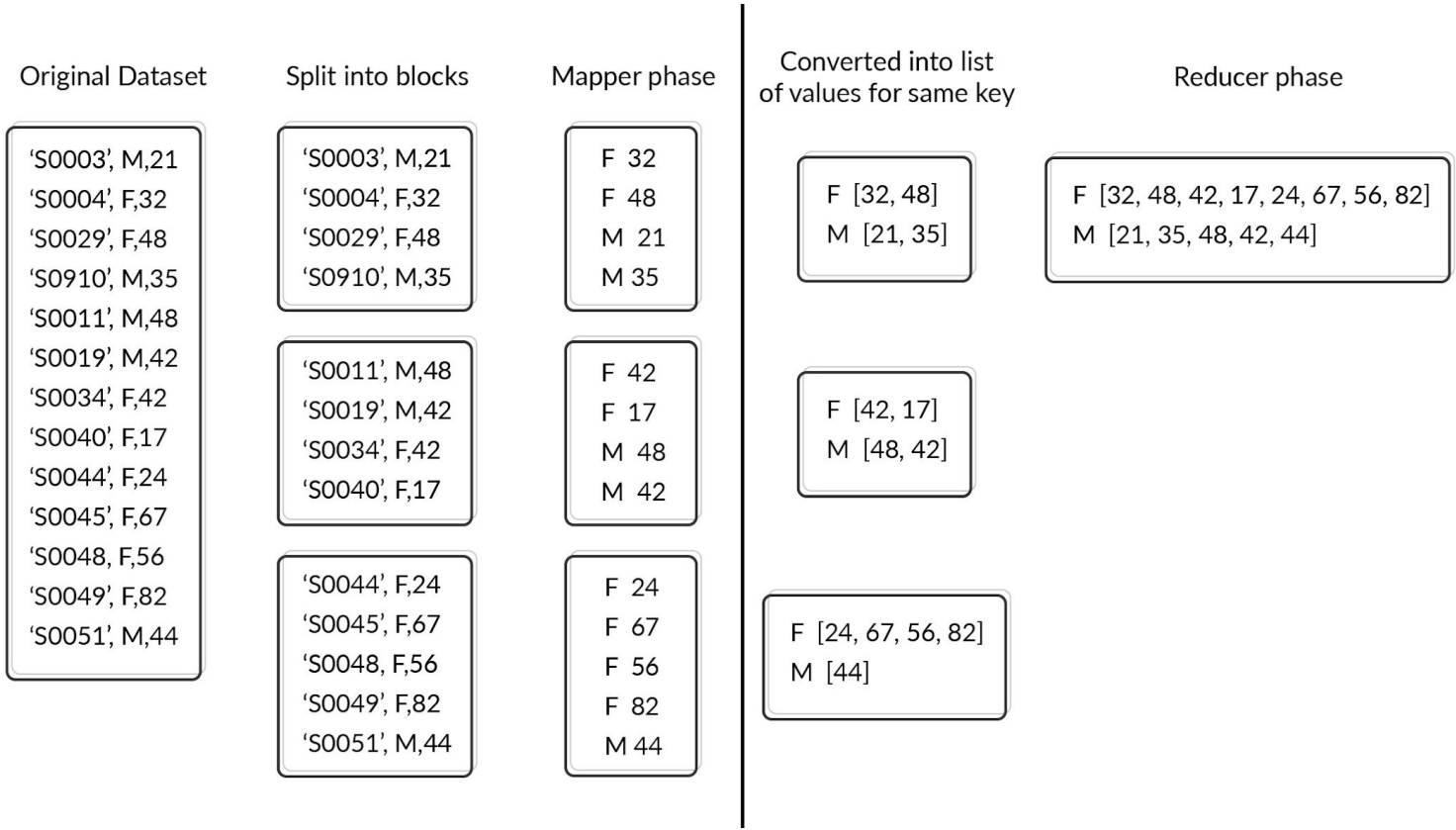
Introduction to the MapReduce Framework



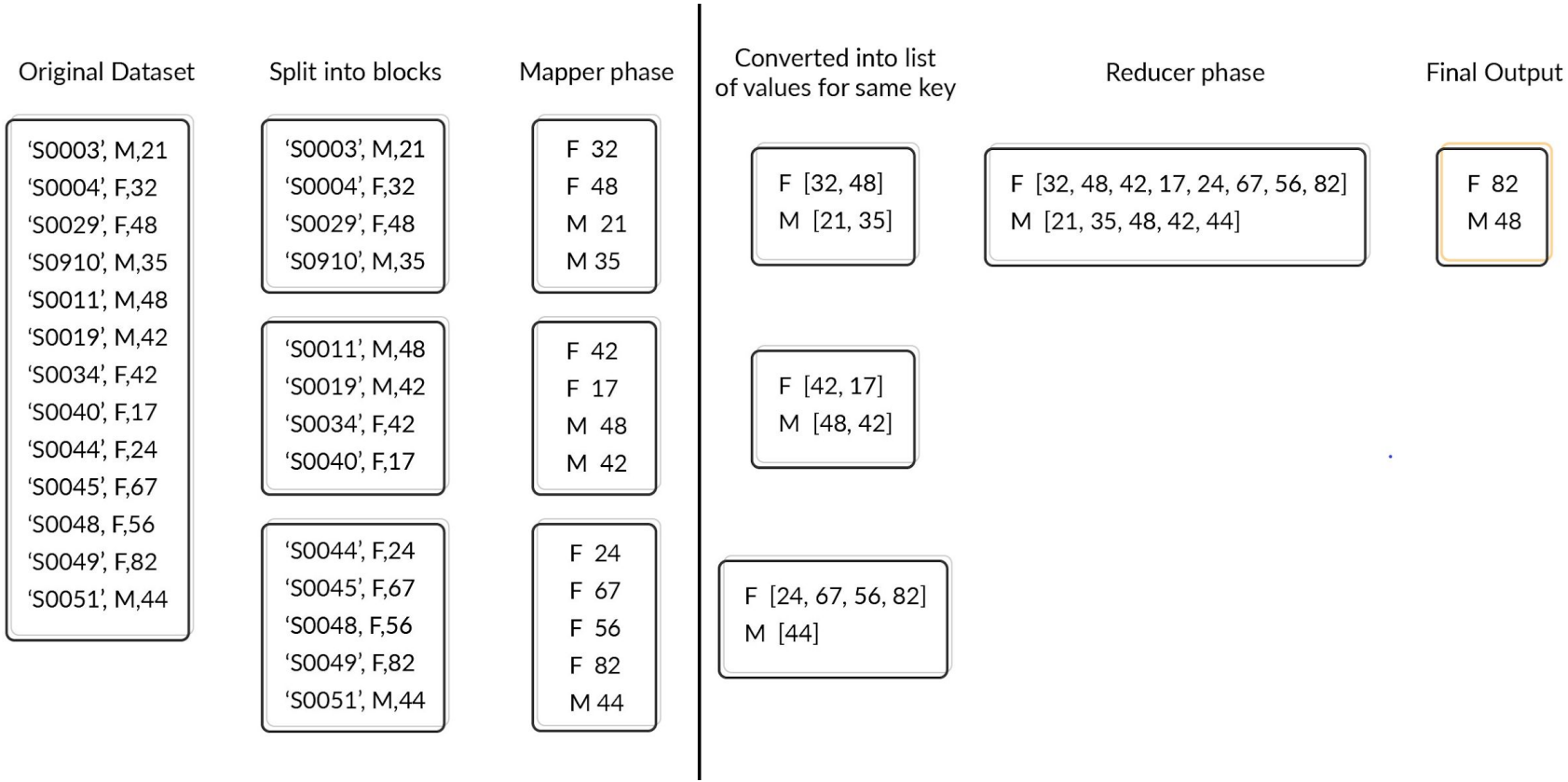
Introduction to the MapReduce Framework



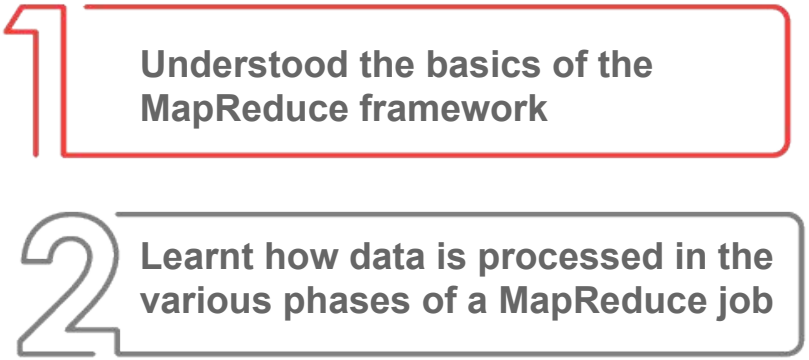
Introduction to the MapReduce Framework



Introduction to the MapReduce Framework

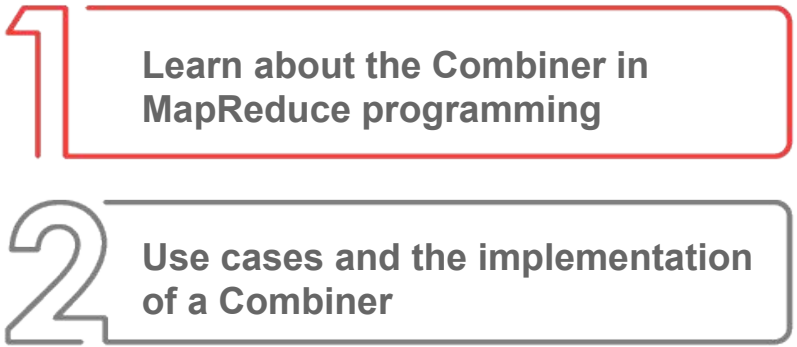


Segment Summary

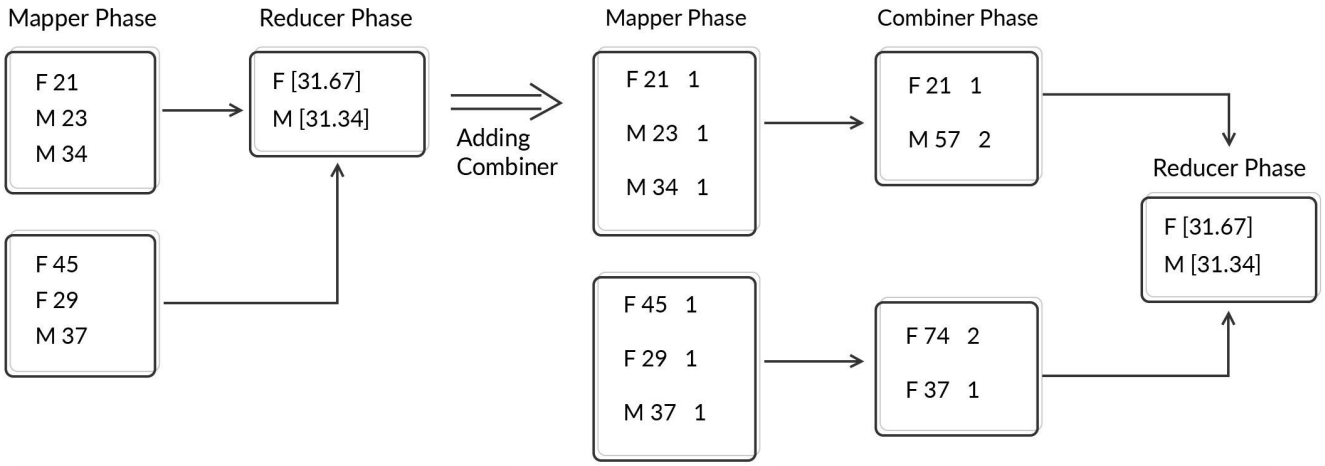


Segment - 05
The Combiner

Segment Overview



The Combiner



Note: Here Reducer will take more time as it first has to take all 6 records and then aggregate them

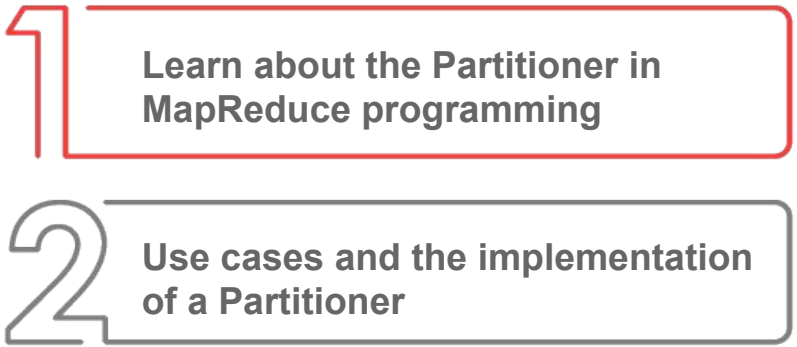
Note: Here Reducer will take lesser time to aggregate as Combiner already calculates the sum from the Mapper Phase and sends it to the Reducer

Segment Summary

1. Learnt about the Combiner in MapReduce programming
2. Implemented an example and learnt when a combiner can be used

Segment - 06
The Partitioner

Segment Overview



The Partitioner

The Partitioner can be used to partition key-value pairs in such a way that the values for each key are partitioned together. This helps in reducing the amount of time taken by the Reducer and allows faster processing of MapReduce jobs.

Suppose we need to partition the data given below such that the addresses with the first 16 bits are processed by the same Reducer.

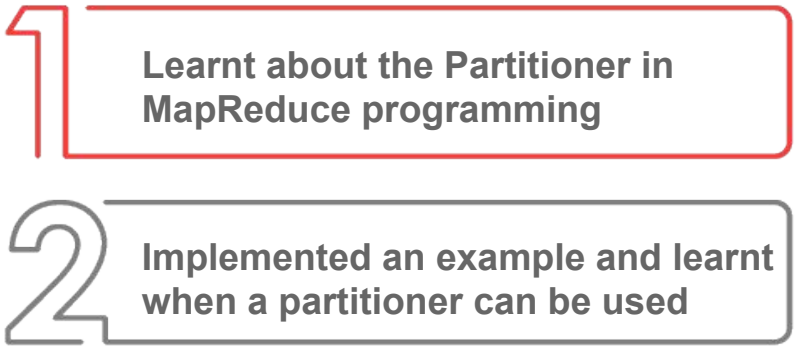
- 192.168.3.1
- 190.192.21.30
- 191.53.75.111
- 192.168.1.7

The Partitioner

Here, we can use the partitioner class ‘KeyFieldBasedPartitioner’ to perform the partition.

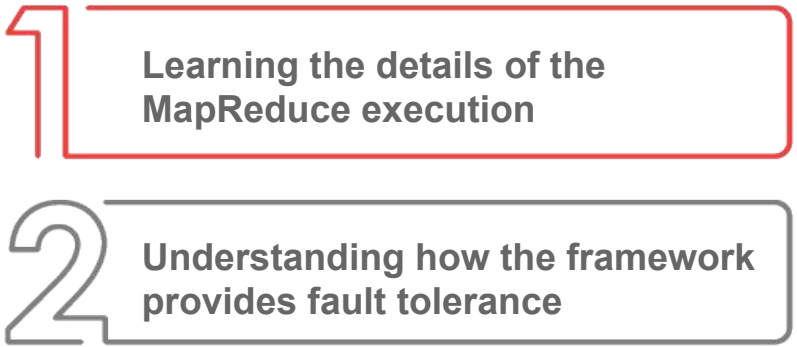
```
hadoop jar \
/lib/hadoop-mapreduce/hadoop-streaming-2.8.5-amzn-6.jar \
-file mapper.py -mapper 'python mapper.py' \
-file reducer.py -reducer' python reducer.py' \
-input <Input> \
-output <Output> \
-D mapreduce.map.output.key.field.separator=. \
-D num.key.fields.for.partition=2 \
-partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner
```

Segment Summary



Segment - 07
Job Scheduling and Fault Tolerance

Segment Overview

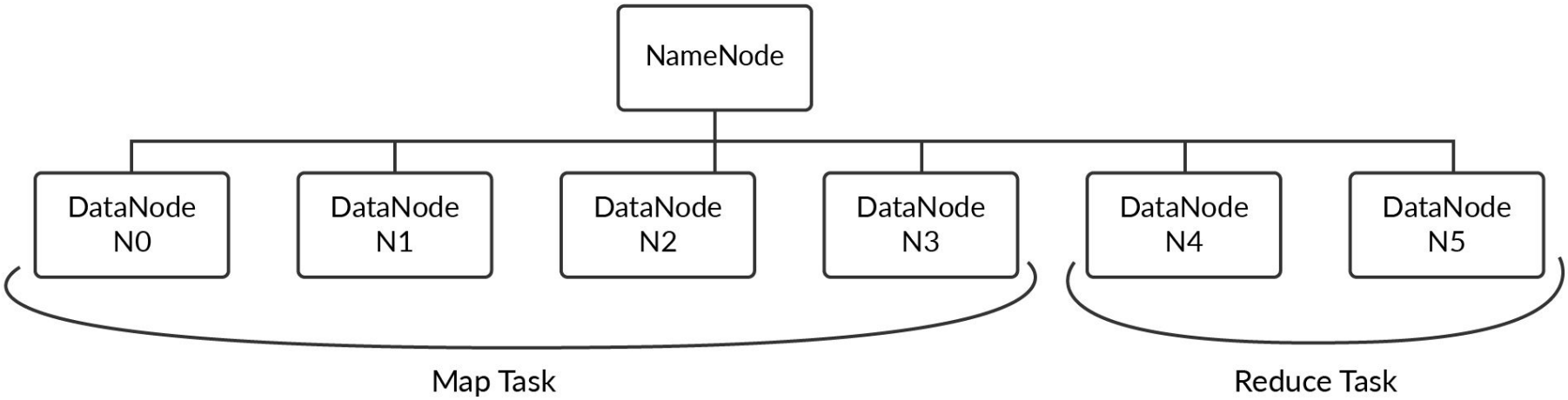


Job Scheduling and Fault Tolerance

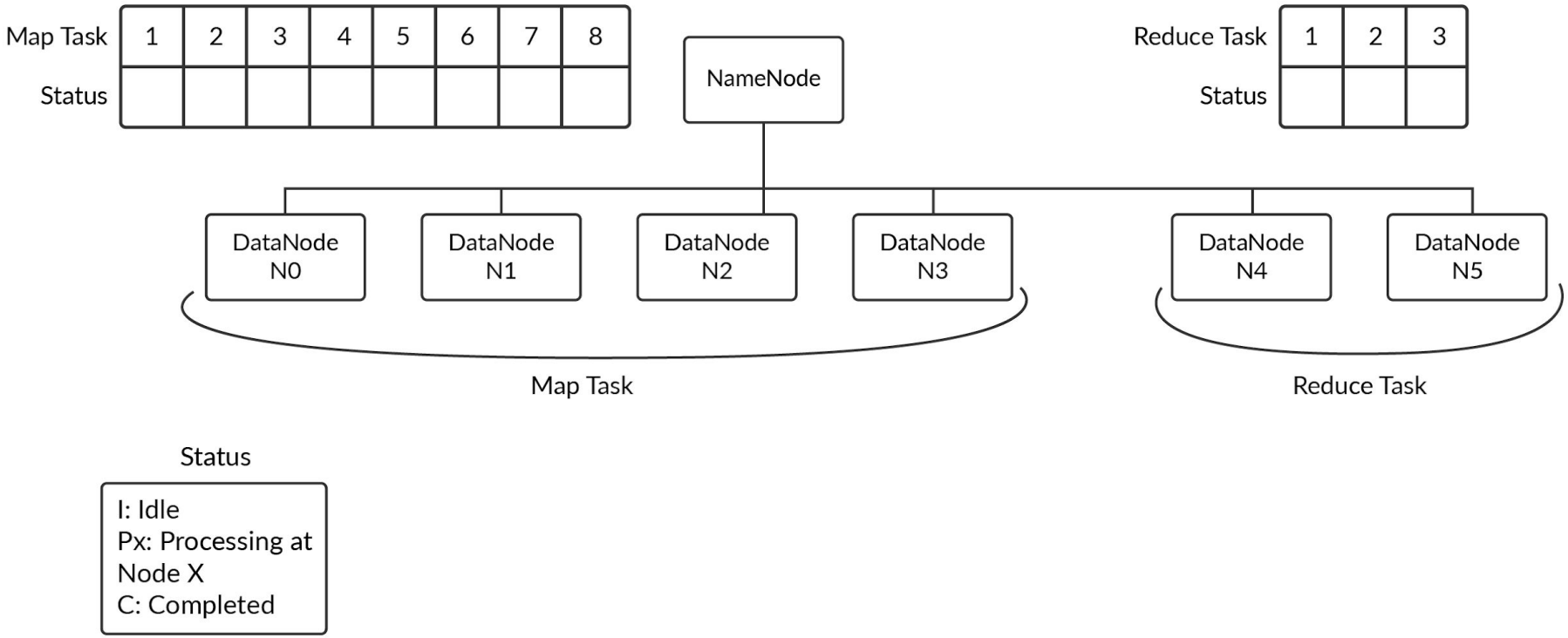
Recall YARN and its components. In Hadoop 2.x, the execution process is handled in its context.

In the previous sessions, you learnt that in an **HDFS**, there are **multiple DataNodes** and a **single NameNode** that manages these DataNodes. The **Map** and **Reduce tasks** are first specified by the **user** and then created by the NameNode.

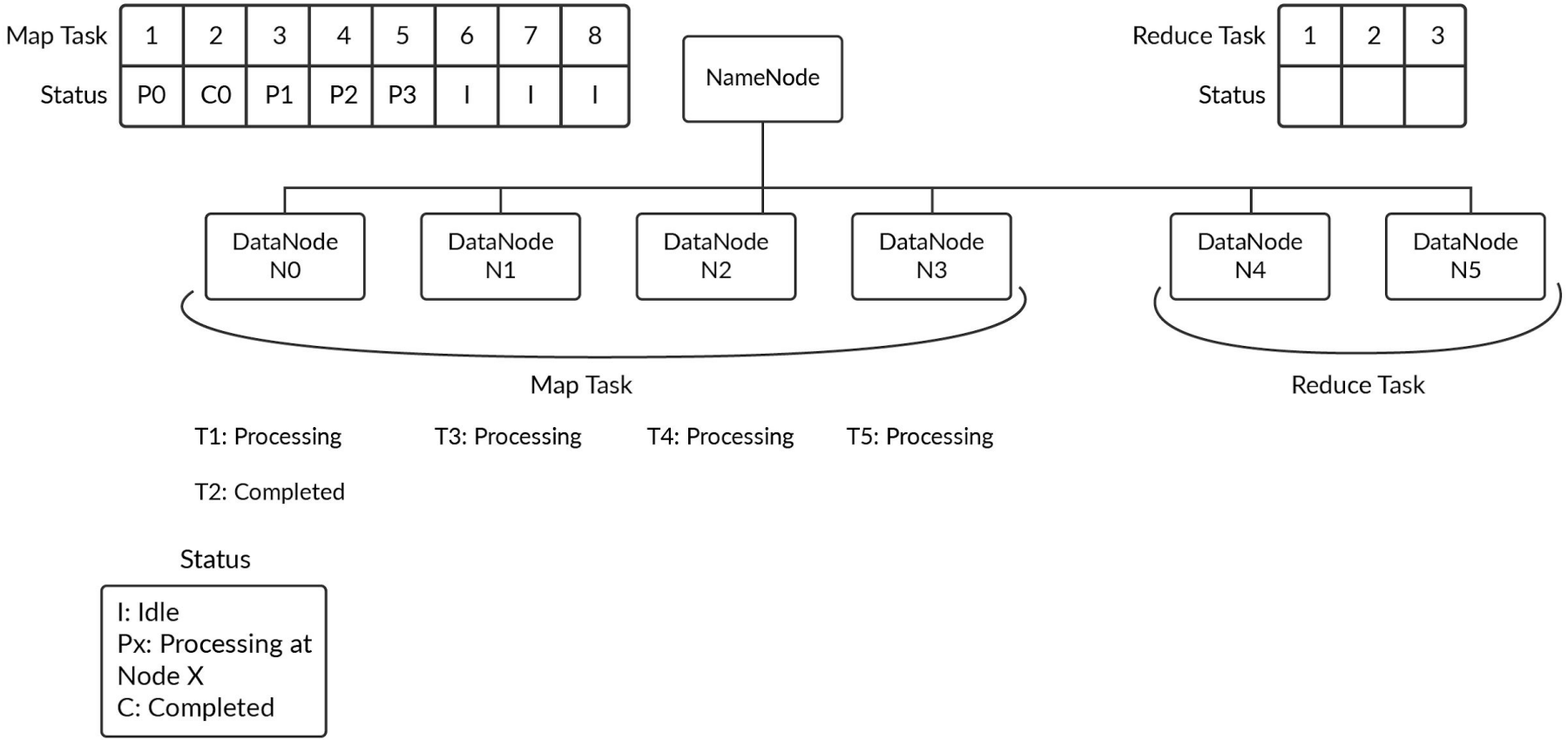
Job Scheduling and Fault Tolerance



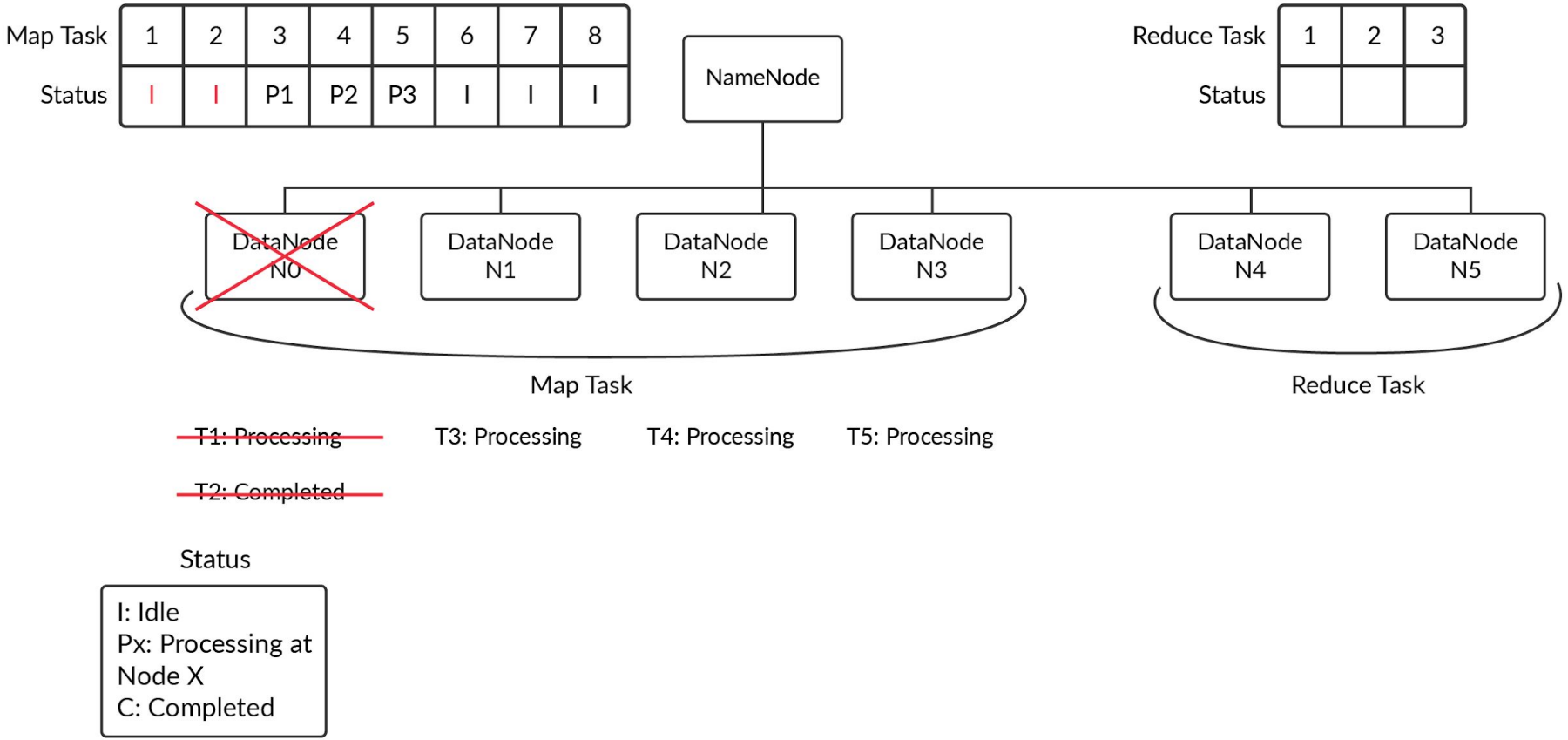
Job Scheduling and Fault Tolerance



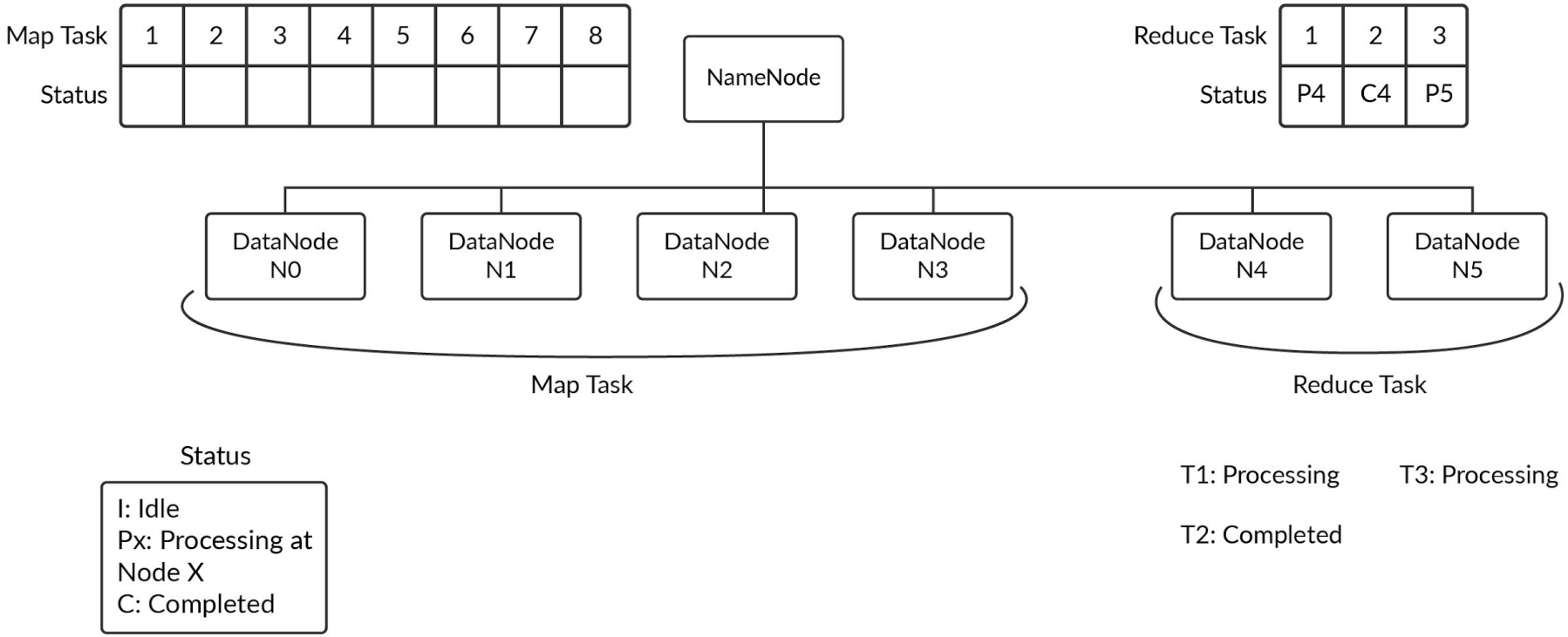
Job Scheduling and Fault Tolerance



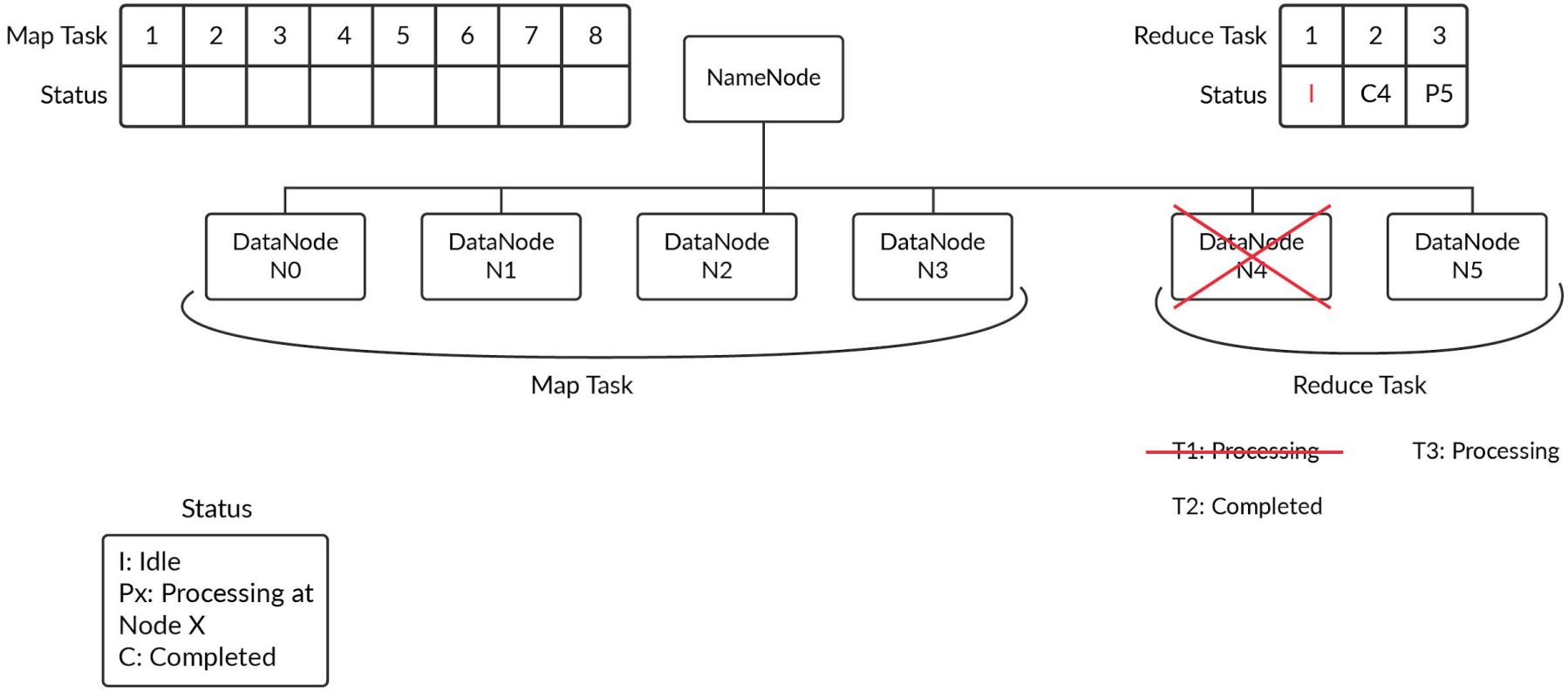
Job Scheduling and Fault Tolerance



Job Scheduling and Fault Tolerance



Job Scheduling and Fault Tolerance



Job Scheduling and Fault Tolerance

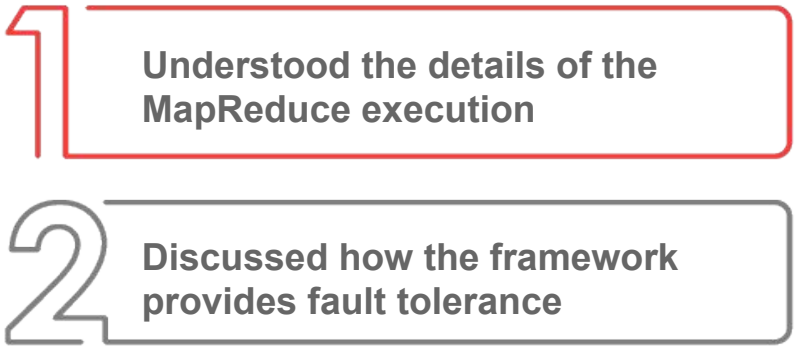
Consider the points given below:

The number of Map tasks is typically set higher than the number of DataNodes.

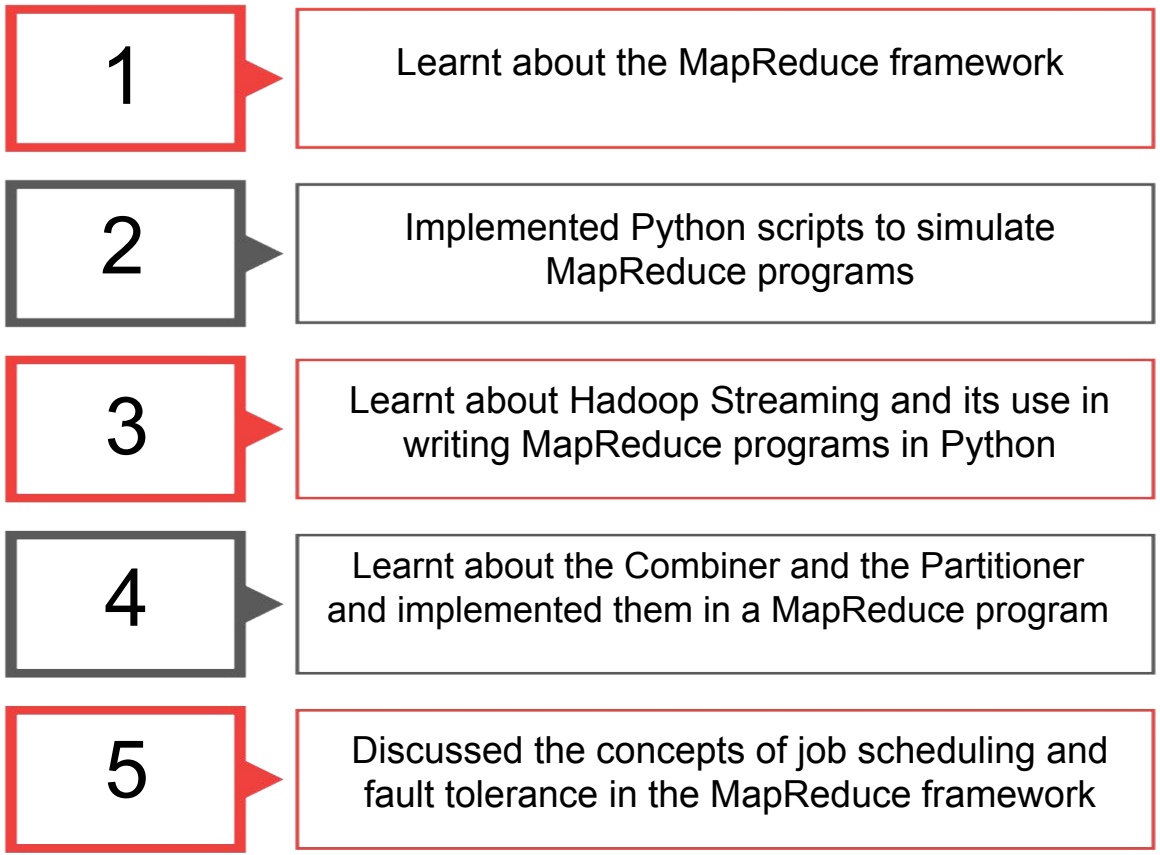
The number of Reduce tasks is usually kept low.

The program is moved to the DataNodes rather than moving the data from the DataNodes to the program.

Segment Summary



Session Summary



Thank you