Agenda

Understand how MapReduce tasks are scheduled and executed using YARN.
Explore the communication between HDFS and YARN.
Work through examples to solidify these concepts.
What is OLTP?
Why it is not suitable for Analytics
Introduction to Data warehousing
Why is data warehouse a better option?
How to design a Data Warehouse
Dimensional Modeling Approach

Map phase

- Reads assigned input split from HDFS
- Parses input into records as key-value pairs
- Applies map function to each record
- Informs master node of its completion

Partition phase

- Each mapper must determine which reducer will receive each of the outputs
- For any key, the destination partition is the same
- Number of partitions = Number of reducers

Shuffle phase

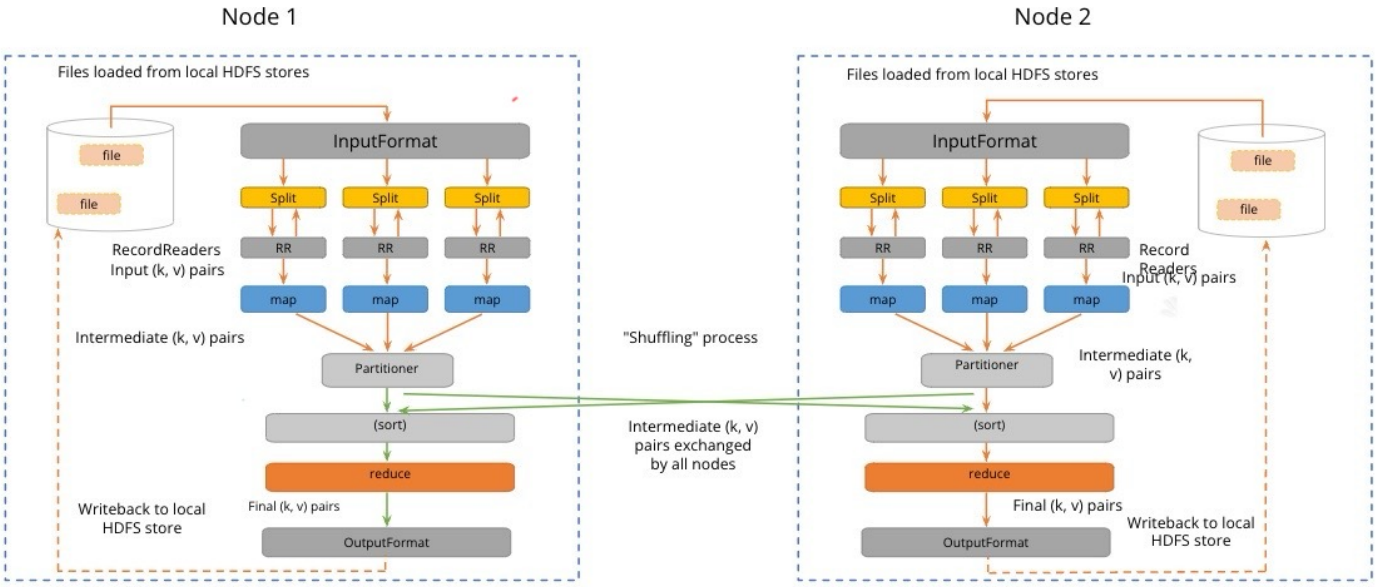
- Fetches input data from all map tasks for the portion corresponding to the reduce tasks bucket

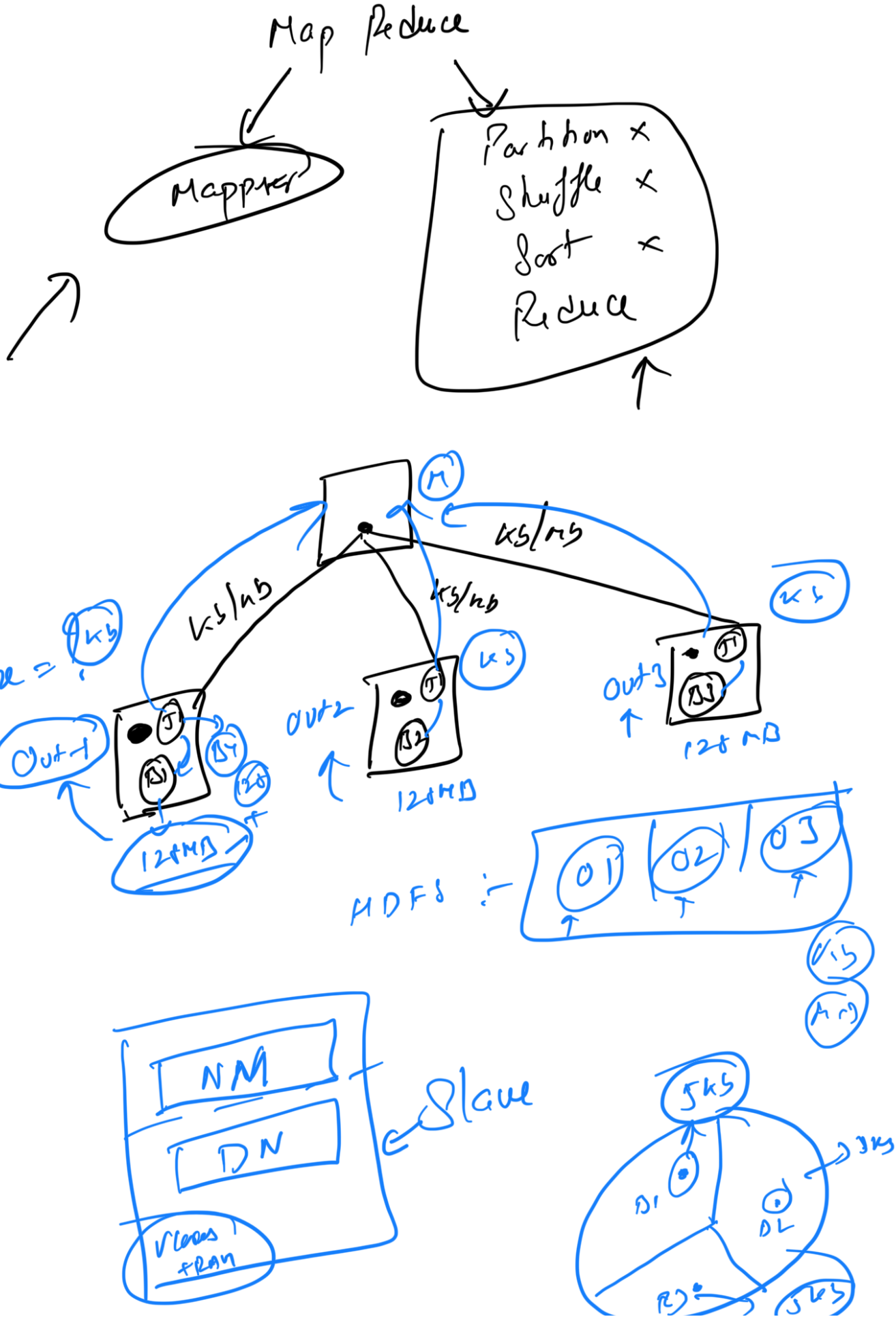
Sort phase

- Merge sorts all map outputs into a single run

Reduce phase

- Applies user-defined reduce function to the merged run




Map Reduce
Mapper
Partition x
Shuffle x
Sort x
Reduce
M
1KB/ms
1KB/ms
1KB/ms
Size = 1KB
Out1
Out2
Out3
128MB
128MB
128MB
HDFS :-
01
02
03
NM
DN
Slave
Yarn
Task
5KB
3KB
D1
D2
D3
5KB

Input format ⇒

① Determines how the input data is divided into chunks, known as Split

② Supplies a 'Record Reader' implementation that reads the data within each split and convert into key value pairs.

→ Common Input format implementations :-

① → TextInputFormat (default)
↳ K = byte offset of line
V = line of text
1:1

② → KeyValueTextInputFormat
↳ used when line here of the input file contains a (K, V) pair separated by a delimeter.

K V
a b
c d
e f

file

4 NLine input format

5 DB input format
↳ JDBC (relational DB)

K = line number (PK)

V = row

Datatypes

Text →
IntWritable →
FloatWritable →
DoubleWritable →
BooleanWritable →
...Writable →

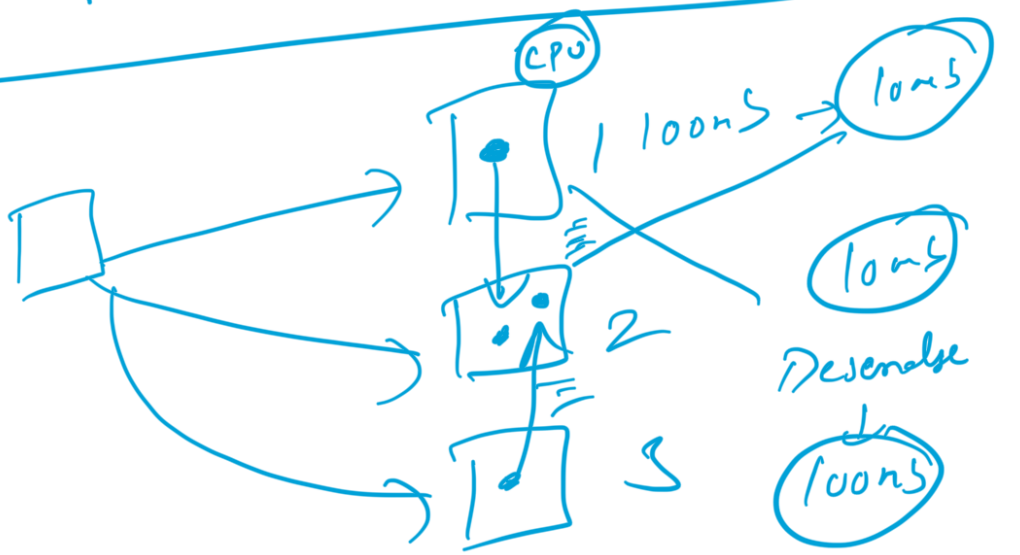
Functions

String
Integer
Float
Double
Boolean
Byte

Byte → placeholder

NullWritable → is not needed.

LongWritable → Long



Writable interface

↳ allows hadoop to read/write data in a Serialized form

100ms → 100ms

↓ Compression

Mapper (LongW, Text, Int, Int)

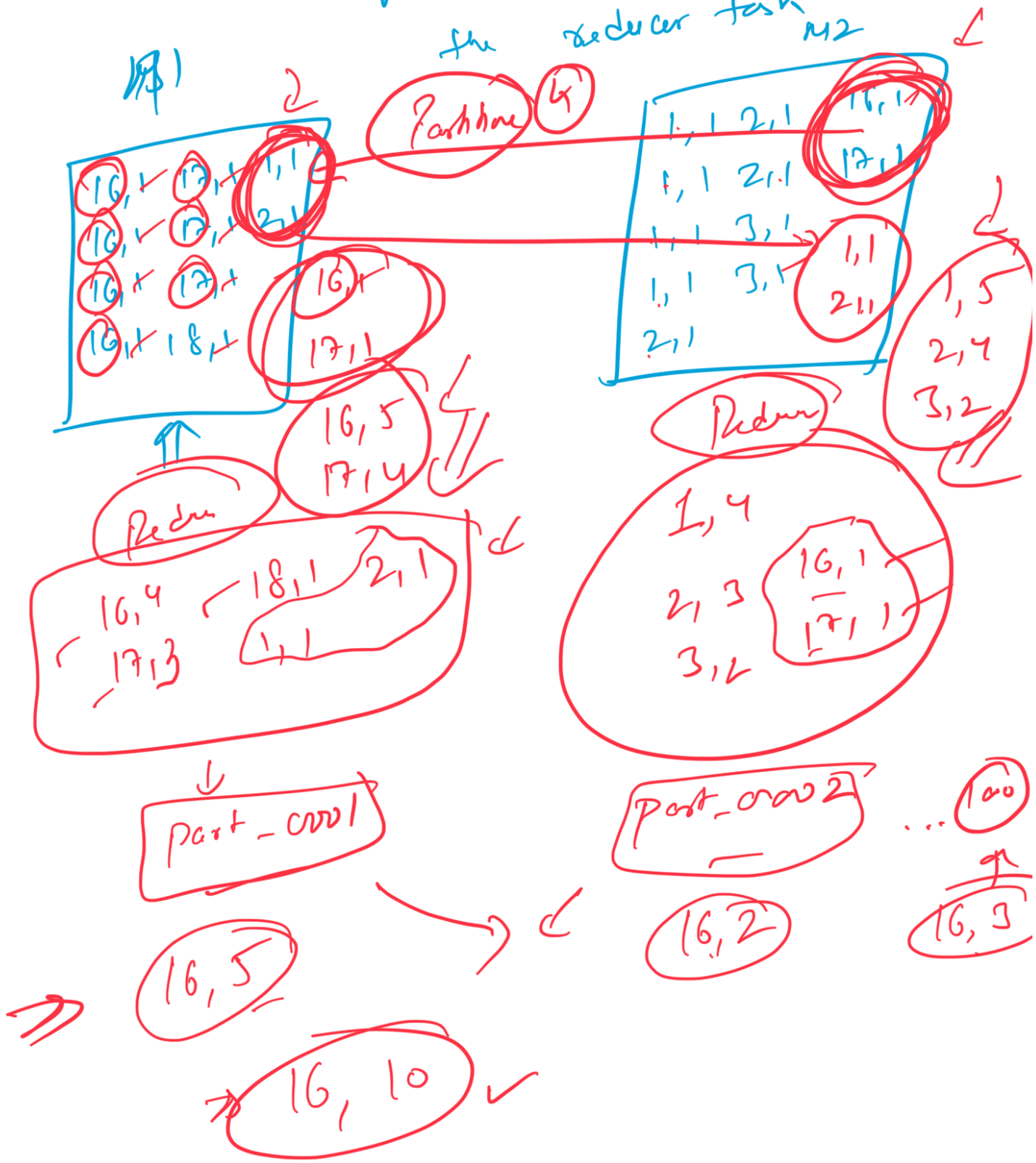
↓ ↓ ↓ ↓

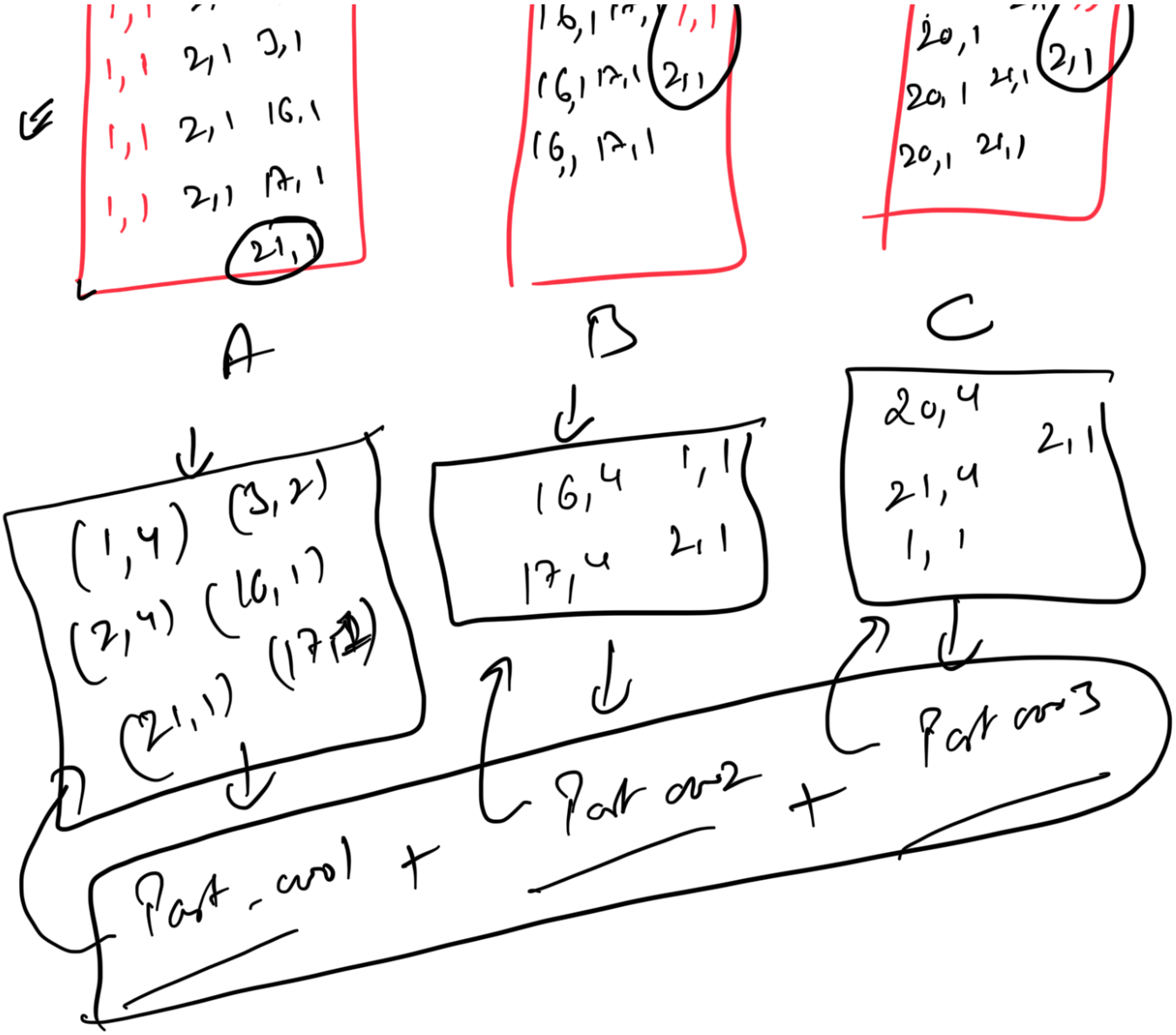
Input output

⇒ Partitioner

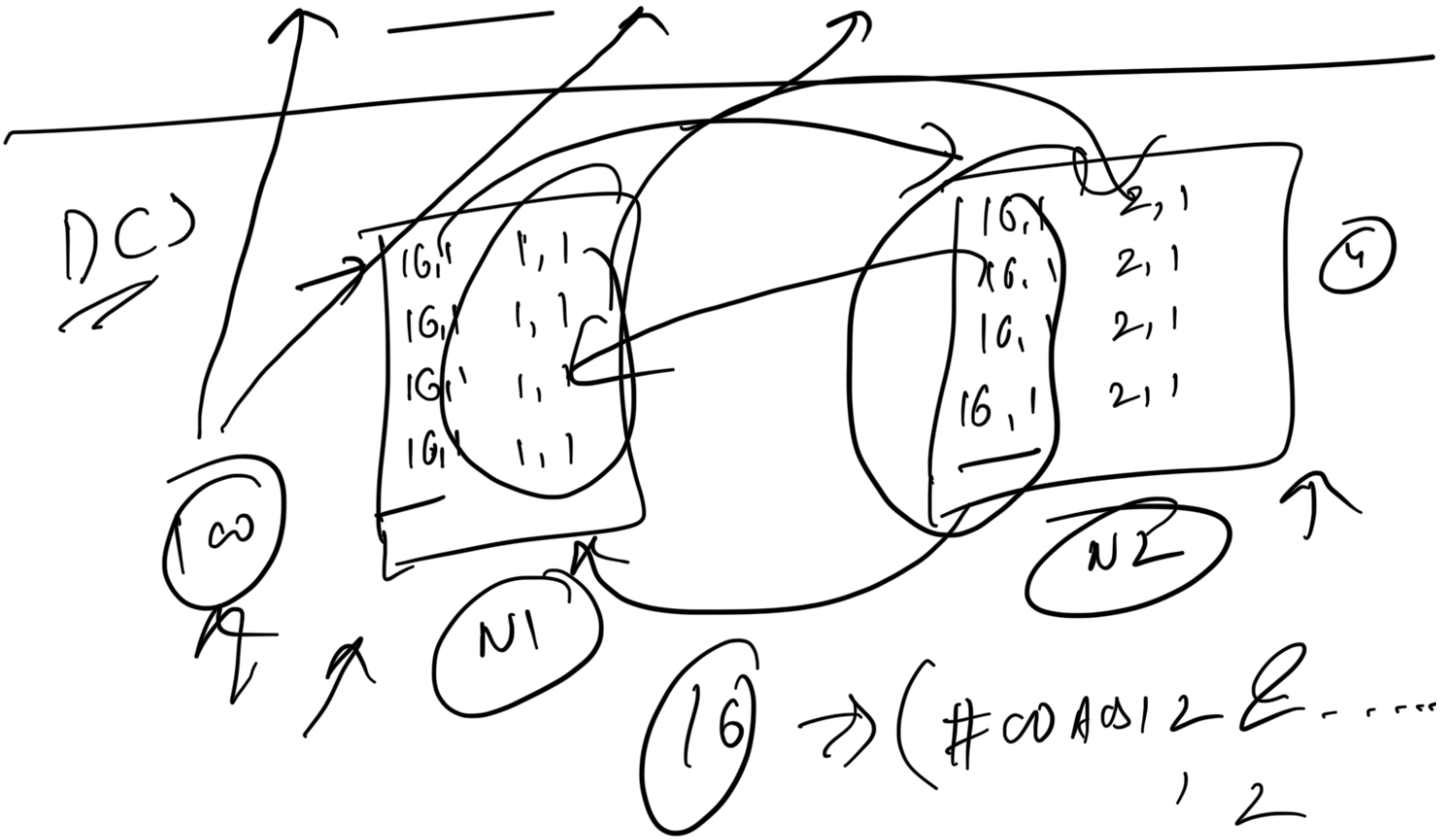
↳ plays a crucial role in the output

Determining how of mapper is distributed among the reducer task





Formula: ReducerNumber = (key.hashCode() & Integer.MAX_VALUE) % numReducers



→ hadoop fs - getmerge

→ Eclipse. Copy merge