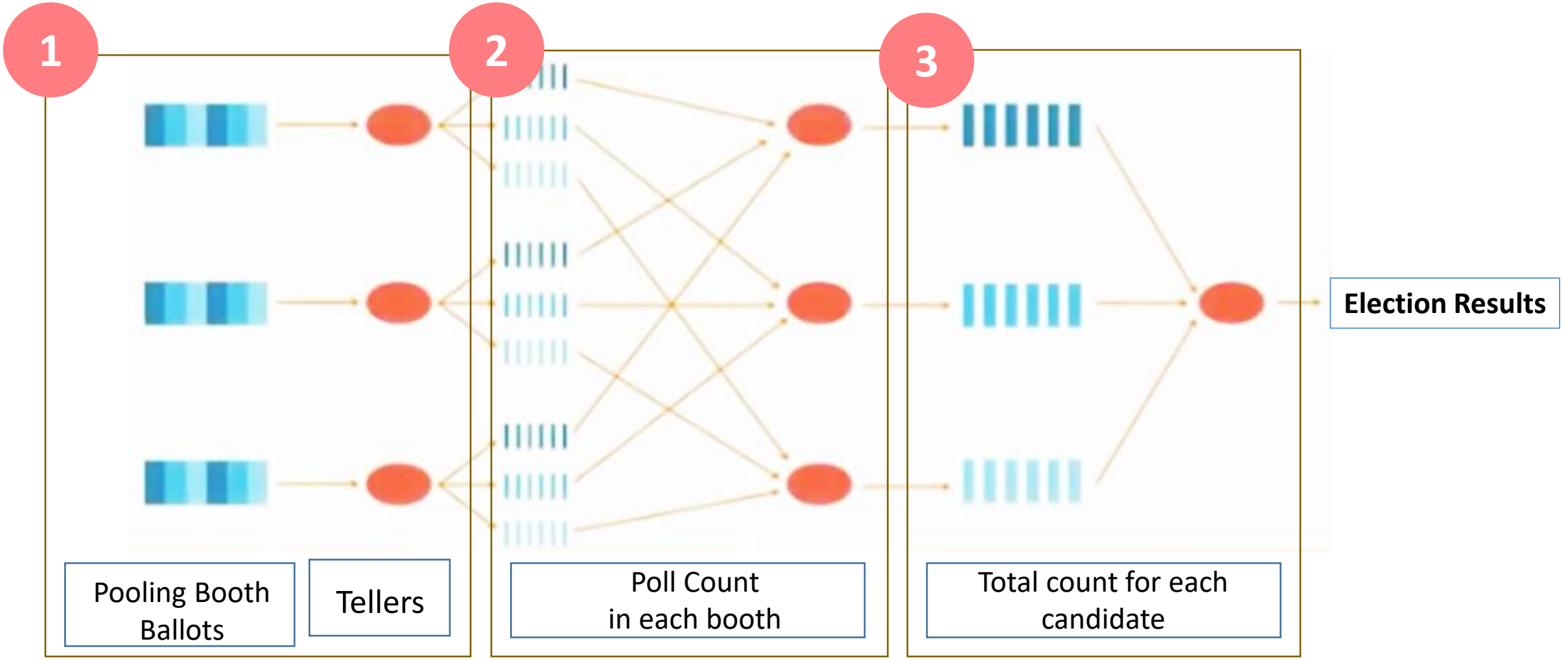
BIG DATA

HADOOP

BIG DATA

MAPREDUCE PROCESS

MAPREDUCE ANALOGY



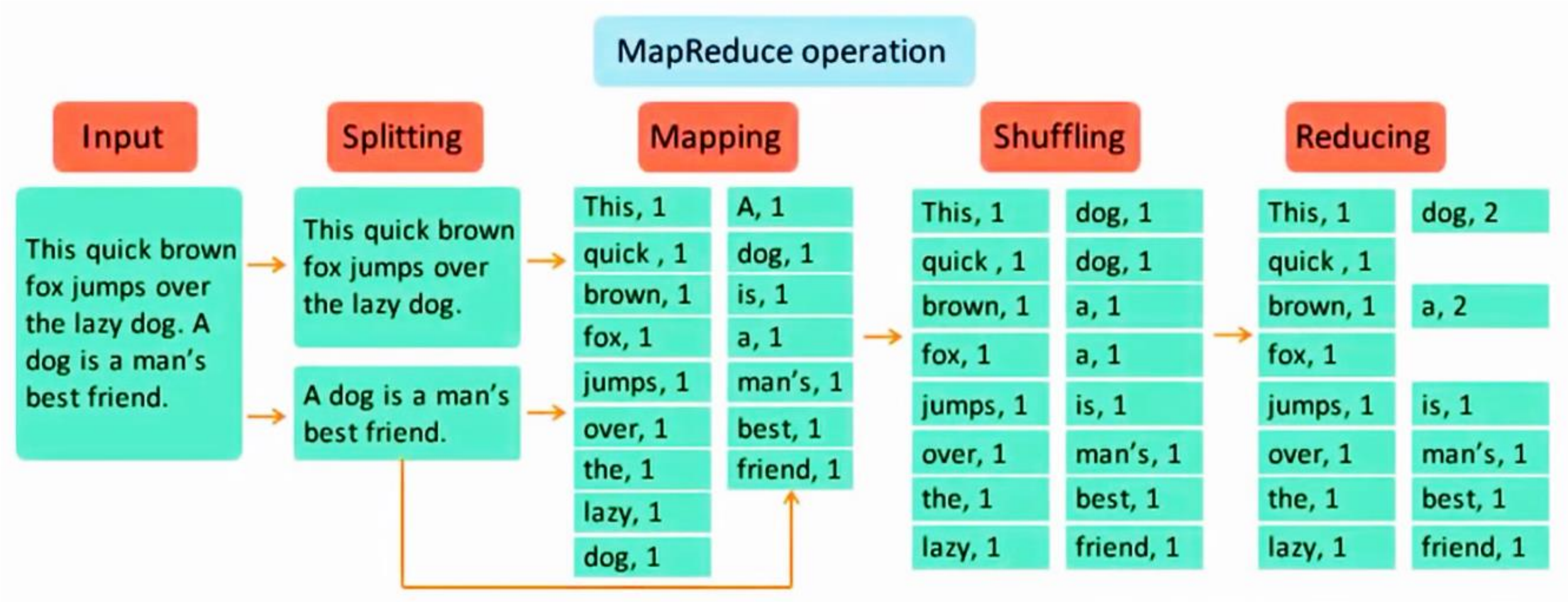
MAPREDUCE PARALLELISM

The Key Reason to perform mapping and reducing is to speed up the execution of a specific process by splitting a process into number of tasks, thus it encourages parallelism in job flow.

Individual Work

Parallel Work

MAPREDUCE ANALOGY



MAPREDUCE PERFORMANCE TUNING

Mapper task is the first phase of processing that processes each input record (from RecordReader) and generates an intermediate key-value pair. Hadoop Mapper store intermediate-output on the local disk.

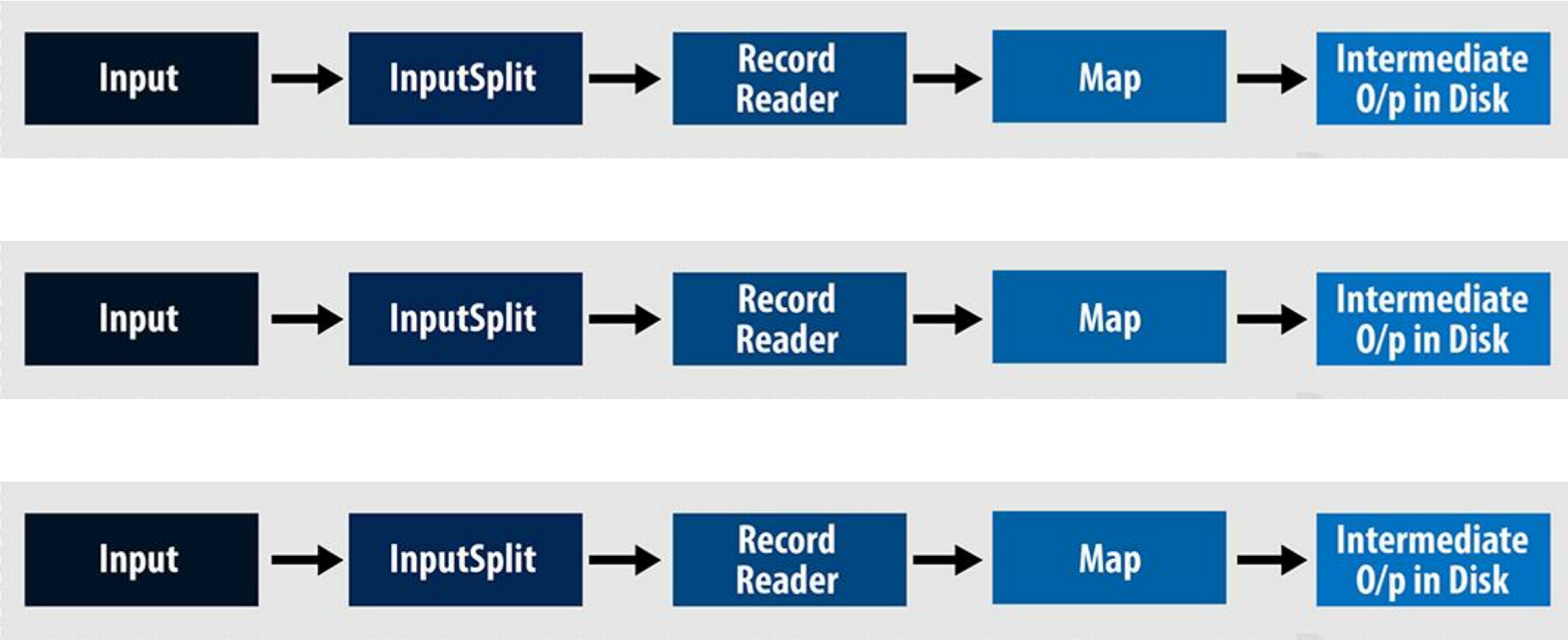
Hadoop Mapper task processes each input record and it generates a new <key, value> pairs. The <key, value> pairs can be completely different from the input pair. In mapper task, the output is the full collection of all these <key, value> pairs.

Key-Value PAIR GENERATION IN HADOOP

Input Split – It is the logical representation of data. It describes a unit of work that contains a single map task in a MapReduce program.

Record Reader – It communicates with the InputSplit and it converts the data into key-value pairs suitable for reading by the Mapper. By default, it uses TextInputFormat for converting data into the key-value pair. RecordReader communicates with the Inputsplit until the file reading is not completed.

INPUT SPLIT & RECORD READER

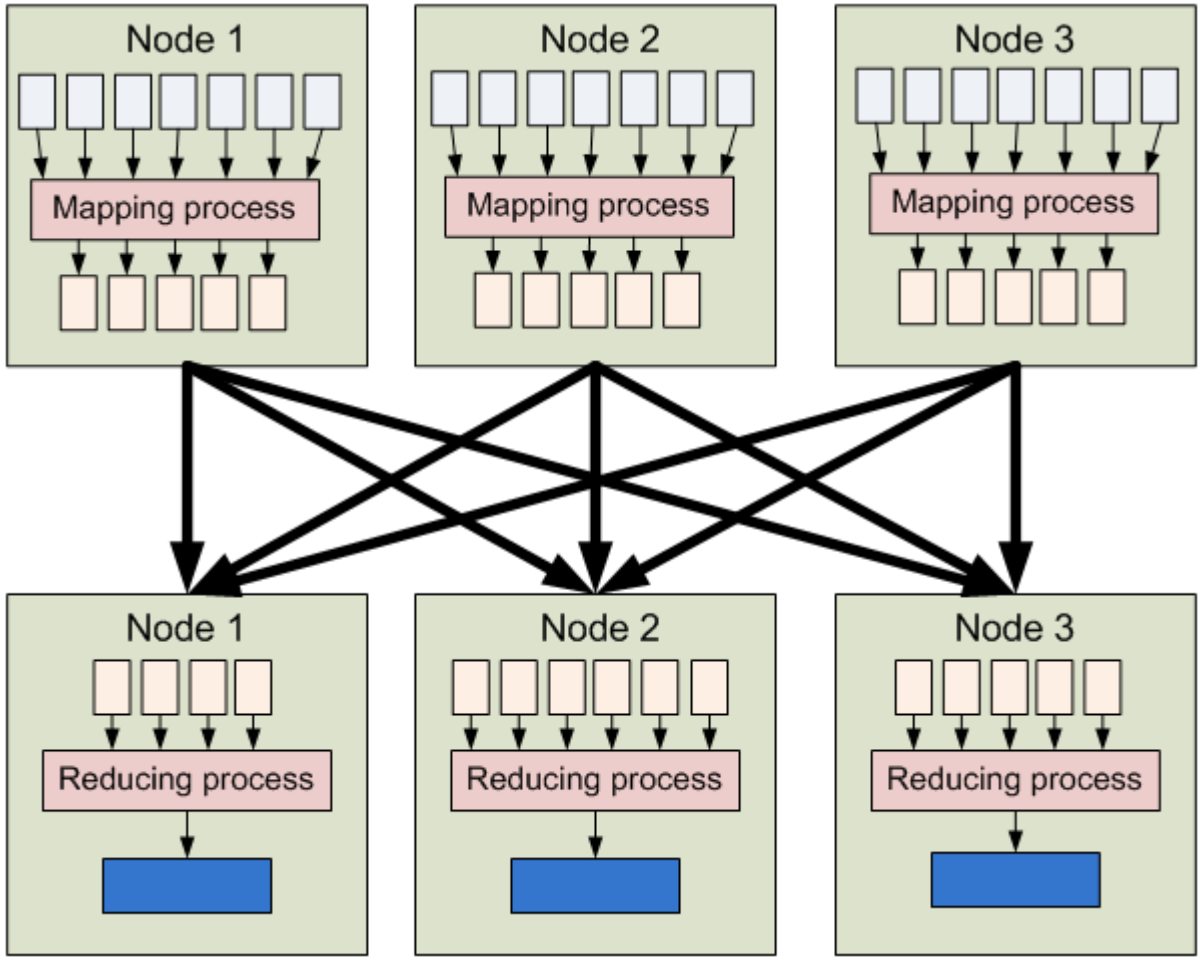


INPUT SPLIT & RECORD READER

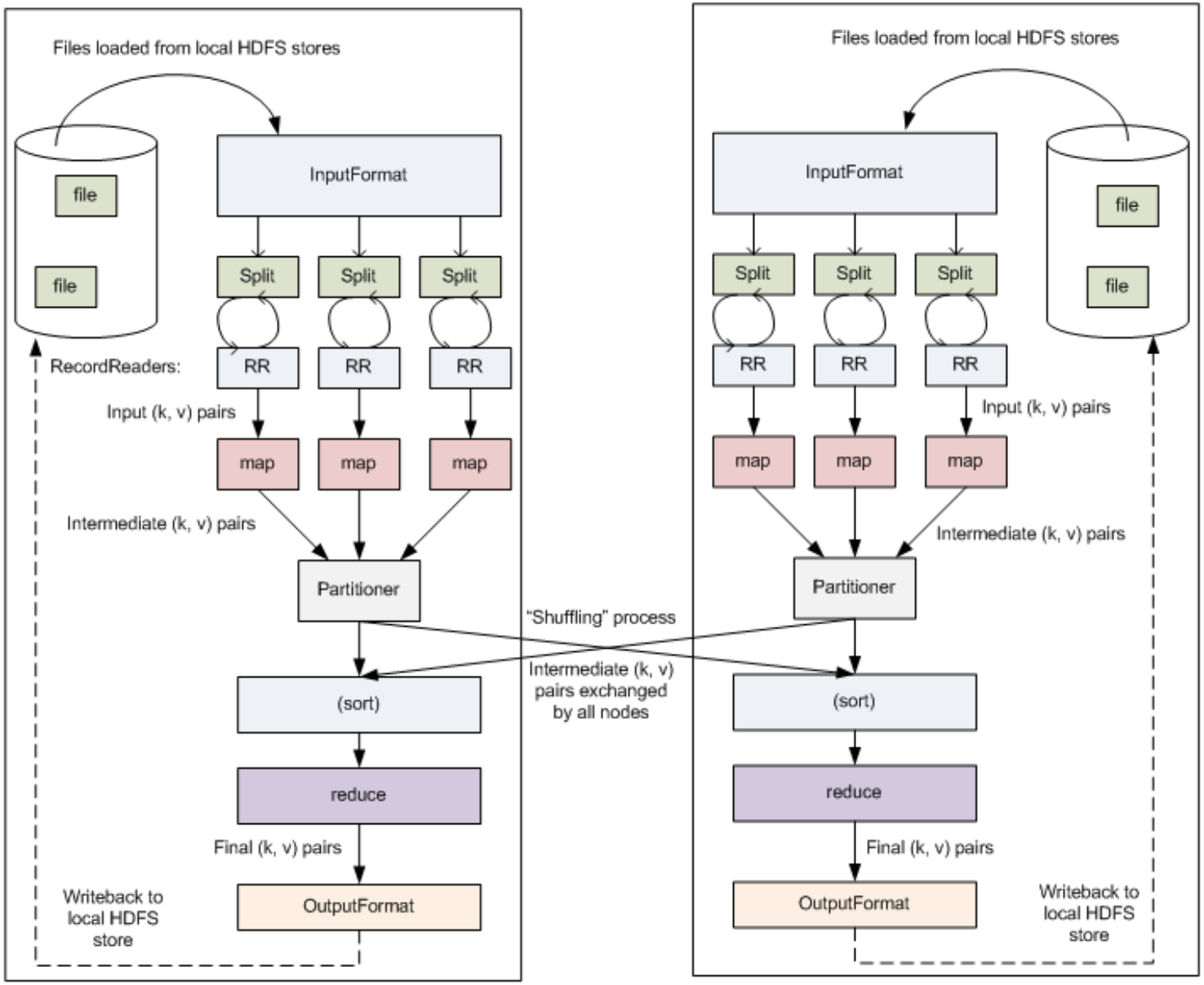
InputSpits do not always depend on the number of blocks, we can customize the number of splits for a particular file by setting ***mapred.max.split.size*** property during job execution.

RecordReader's responsibility is to keep reading/converting data into key-value pairs until the end of the file. Byte offset (unique number) is assigned to each line present in the file by RecordReader. Further, this key-value pair is sent to the mapper. The output of the mapper program is called as intermediate data

MAPREDUCE PROCESS



MAPREDUCE PROCESS CLOSER LOOK



DISABLE REDUCER

We can achieve this by setting ***job.setNumreduceTasks(0)*** in the configuration in a driver. This will make a number of reducer as 0 and thus the only mapper will be doing the complete task.

Advantages of Map only job in Hadoop:

In between map and reduces phases there is key, sort and shuffle phase. Sort and shuffle are responsible for sorting the keys in ascending order and then grouping values based on same keys. This phase is very expensive and if reduce phase is not required we should avoid it, as avoiding reduce phase would eliminate sort and shuffle phase as well. This also saves network congestion as in shuffling, an output of mapper travels to reducer and when data size is huge, large data needs to travel to the reducer.

MODIFY HDFS Block Size

To configure the data block at Cluster level we need to specify in the **hdfs.site.xml**, eg. For 128 mb,value will be 64*1024*1024

<property>
<name>dfs.block.size</name>
<value>134217728</value>
<description>Block size</description>
</property>

For multinode, in a Cluster. we need to update the same in the node(Name Node and Data Node) and restart the daemons.

This change doesn't affect the existing files in Hadoop HDFS.

To change a block size for specific file in a cluster:
hadoop fs -Ddfs.blocksize=134217728 -put /home/hduser/test/test.text /hdfs

ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/daily01

HAPPY LEARNING