

# MapReduce Job Processing

Jeffrey Jestes

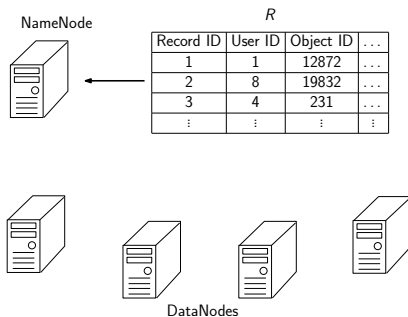
April 17, 2012

# Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).

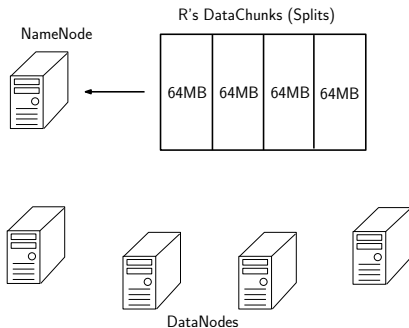
# Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).



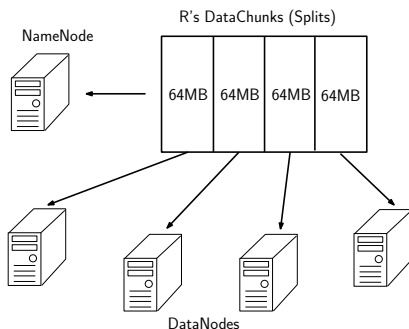
# Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).



# Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).



# Background: Hadoop Core

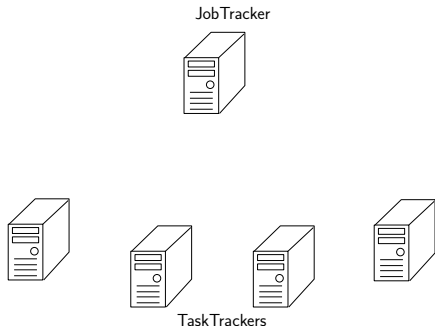
- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.

# Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.

# Background: Hadoop Core

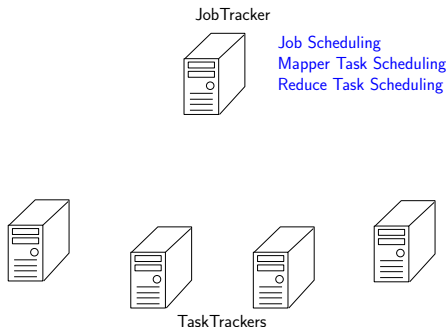
- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.





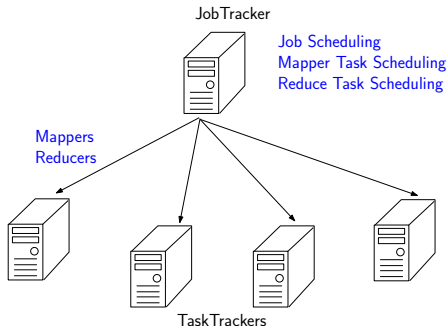
# Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.



# Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.



# Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.

# Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.
- The other machines run DataNode and TaskTracker tasks and are called *slaves*.

# Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.
- The other machines run DataNode and TaskTracker tasks and are called *slaves*.

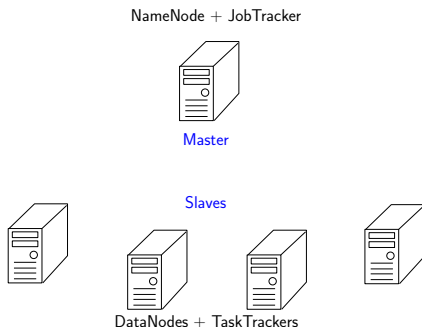
NameNode + JobTracker



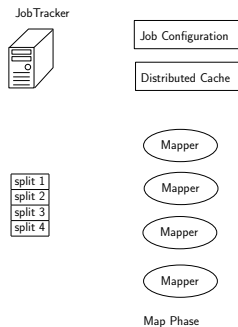
DataNodes + TaskTrackers

# Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.
- The other machines run DataNode and TaskTracker tasks and are called *slaves*.

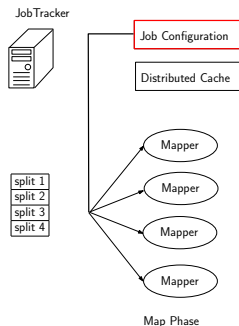


# Background: MapReduce Job Overview



- Next we look at an overview of a typical MapReduce Job.

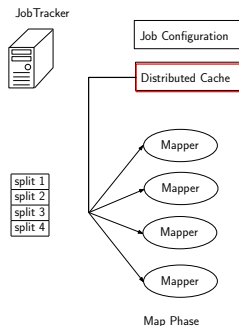
# Background: MapReduce Job Overview



- Job specific variables are first placed in the *Job Configuration* which is sent to each *Mapper Task* by the *JobTracker*.

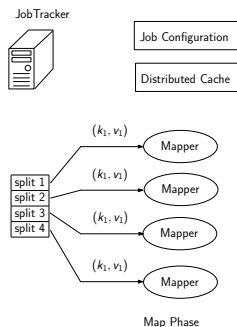


# Background: MapReduce Job Overview



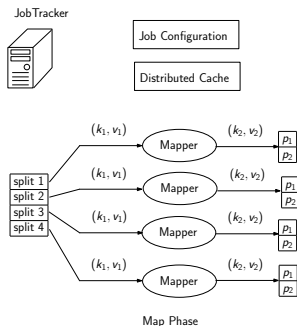
- Large data such as files or libraries are then put in the *Distributed Cache* which is copied to each *TaskTracker* by the *JobTracker*.

# Background: MapReduce Job Overview



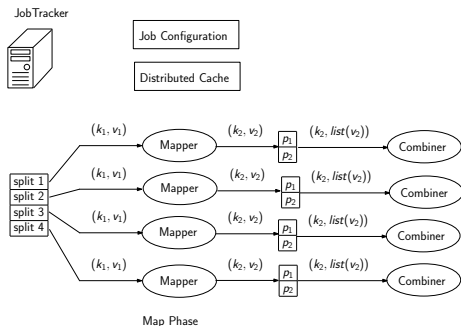
- The JobTracker next assigns each *InputSplit* to a *Mapper* task on a TaskTracker, we assume  $m$  Mappers and  $m$  InputSplits.

# Background: MapReduce Job Overview



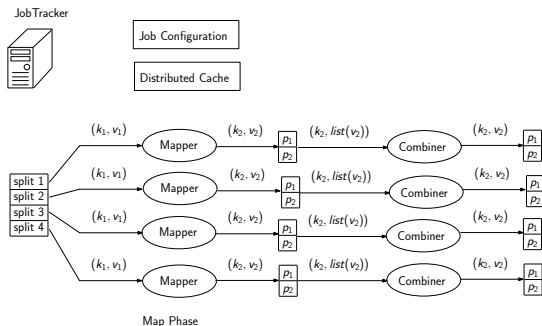
- Each Mapper maps a  $(k_1, v_1)$  pair to an intermediate  $(k_2, v_2)$  pair and partitions by  $k_2$ , i.e.  $hash(k_2) = p_i$  for  $i \in [1, r]$ ,  $r = |reducers|$ .

# Background: MapReduce Job Overview



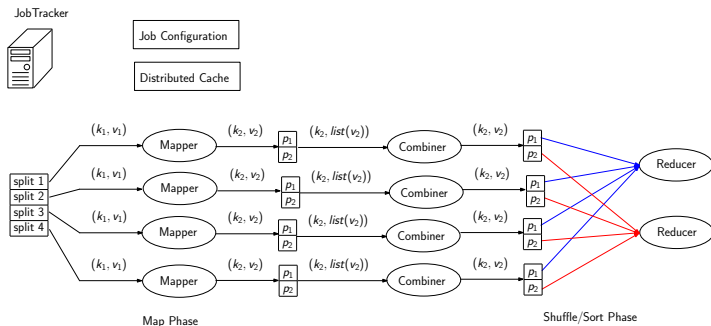
- An optional *Combiner* is executed over  $(k_2, list(v_2))$ .

# Background: MapReduce Job Overview



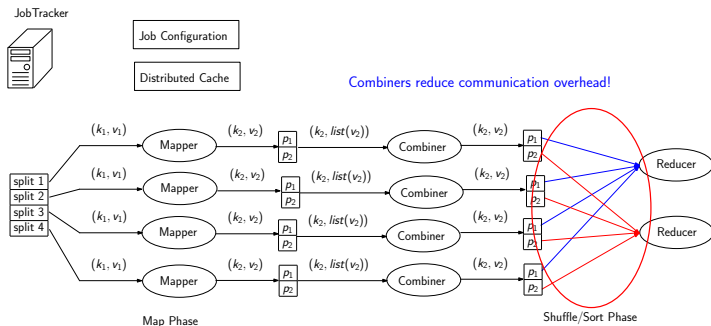
- The *Combiner* aggregates  $v_2$  for a  $k_2$  and a  $(k_2, v_2)$  is written to a partition on disk.

# Background: MapReduce Job Overview



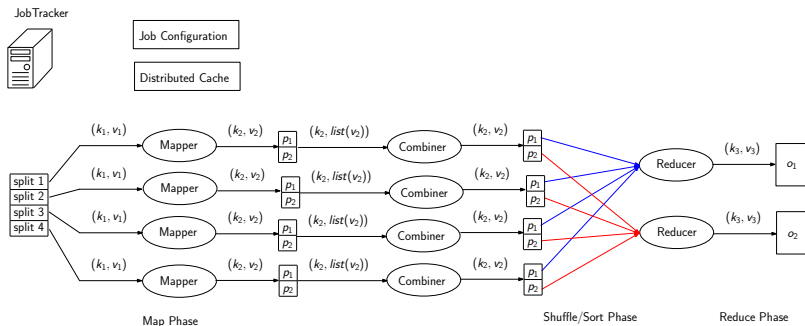
- The JobTracker assigns two TaskTrackers to run the Reducers, each Reducer copies and sorts its inputs from corresponding partitions.

# Background: MapReduce Job Overview



- The JobTracker assigns two TaskTrackers to run the Reducers, each Reducer copies and sorts its inputs from corresponding partitions.

# Background: MapReduce Job Overview



- Each Reducer reduces a  $(k_2, \text{list}(v_2))$  to a single  $(k_3, v_3)$  and writes the results to a DFS file,  $o_i$  for  $i \in [1, r]$ .