

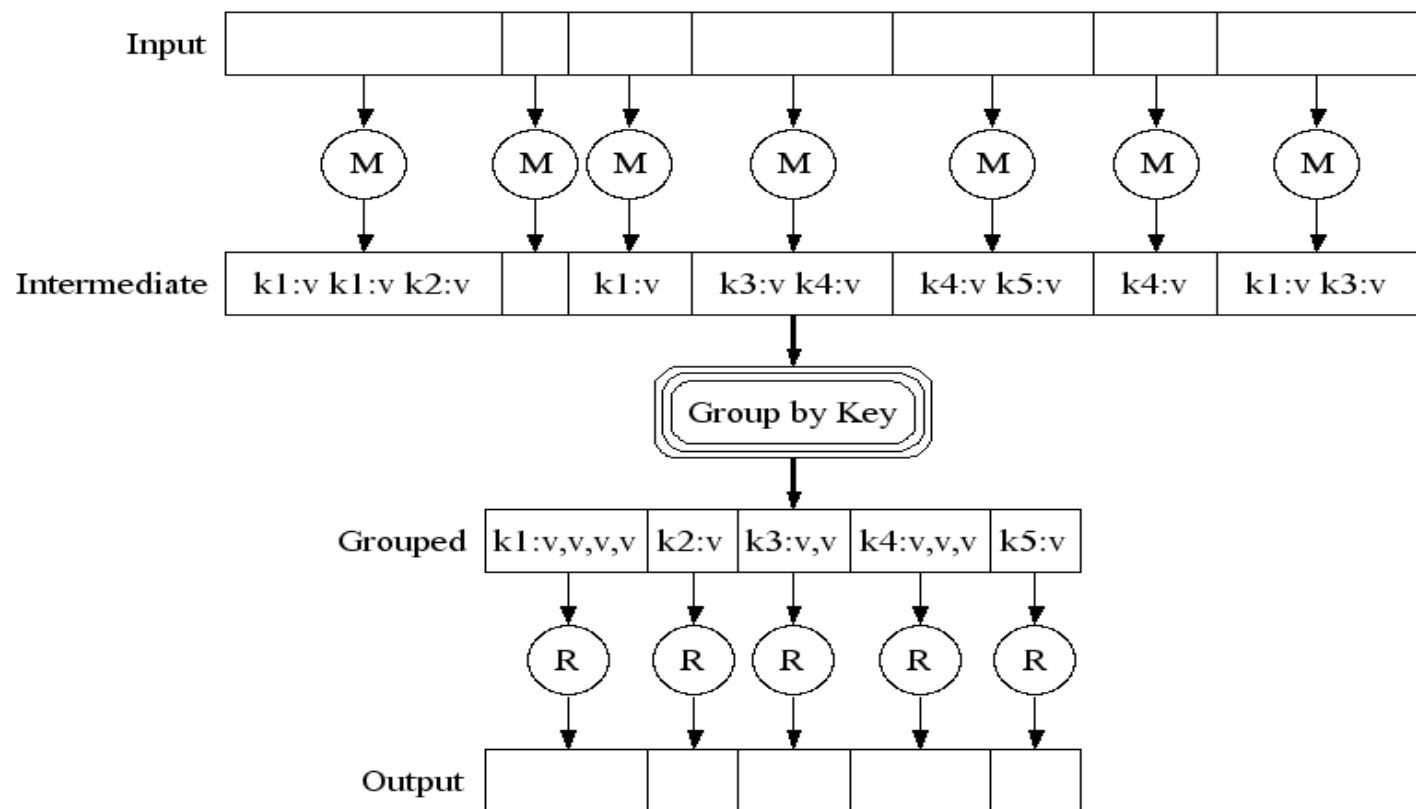


CLOUD COMPUTING APPLICATIONS

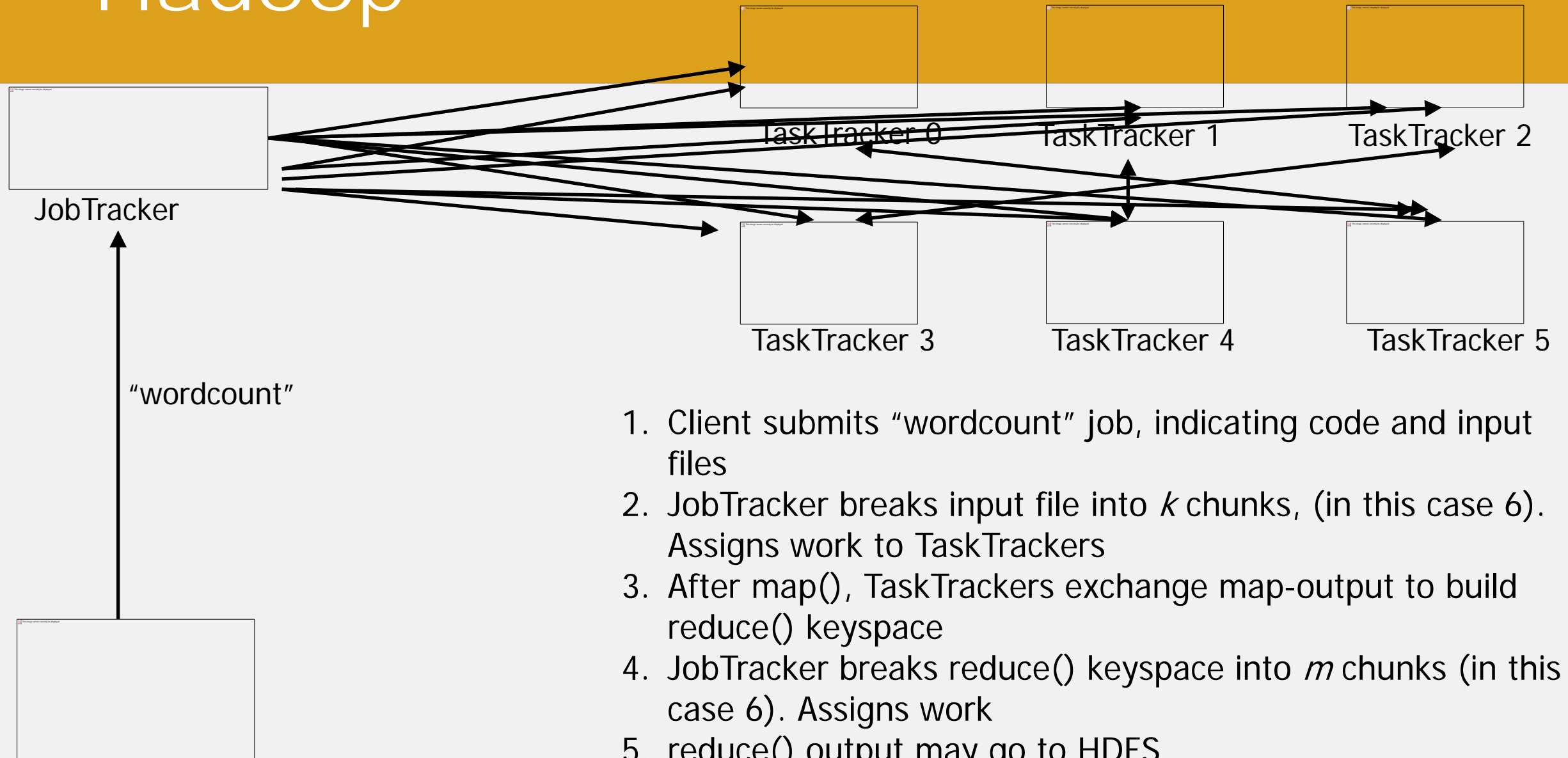
Hadoop Introduction

Roy Campbell & Reza Farivar

Execution



Hadoop



Execution Initialization

- Split input file into 64MB sections (GFS)
 - Read in parallel by multiple machines
- Fork off program onto multiple machines
- One machine is Master
- Master assigns idle machines to either Map or Reduce tasks
- Master coordinates data communication between map and reduce machines

Partition Function

- Inputs to map tasks are created by contiguous splits of input file
- For reduce, we need to ensure that records with the same intermediate key end up at the same worker
- System uses a default partition function
e.g., $\text{hash}(\text{key}) \bmod R$
- Sometimes useful to override
e.g., $\text{hash}(\text{hostname}(\text{URL})) \bmod R$ ensures URLs from a host end up in the same output file

Map-Machine

- Reads contents of assigned portion of input file
- Parses and prepares data for input to map function (e.g., read `<a />` from HTML)
 - Classes implementing `InputFormat`
- Passes data into map function and saves result in memory (e.g., `<target, source>`)
- Periodically writes completed work to local disk
- Notifies Master of this partially completed work (intermediate data)

Reduce-Machine

- Receives notification from Master of partially completed work
- Retrieves intermediate data from Map-Machine via remote-read
- Sorts intermediate data by key (e.g., by target page)
- Iterates over intermediate data
 - For each unique key, sends corresponding set through reduce function
- Appends result of reduce function to final output file (GFS)

Data Flow

- Input, final output are stored on a distributed file system
 - Scheduler tries to schedule map tasks “close” to physical storage location of input data
- Intermediate results are stored on local FS of map and reduce workers
- Output is often input to another map reduce task