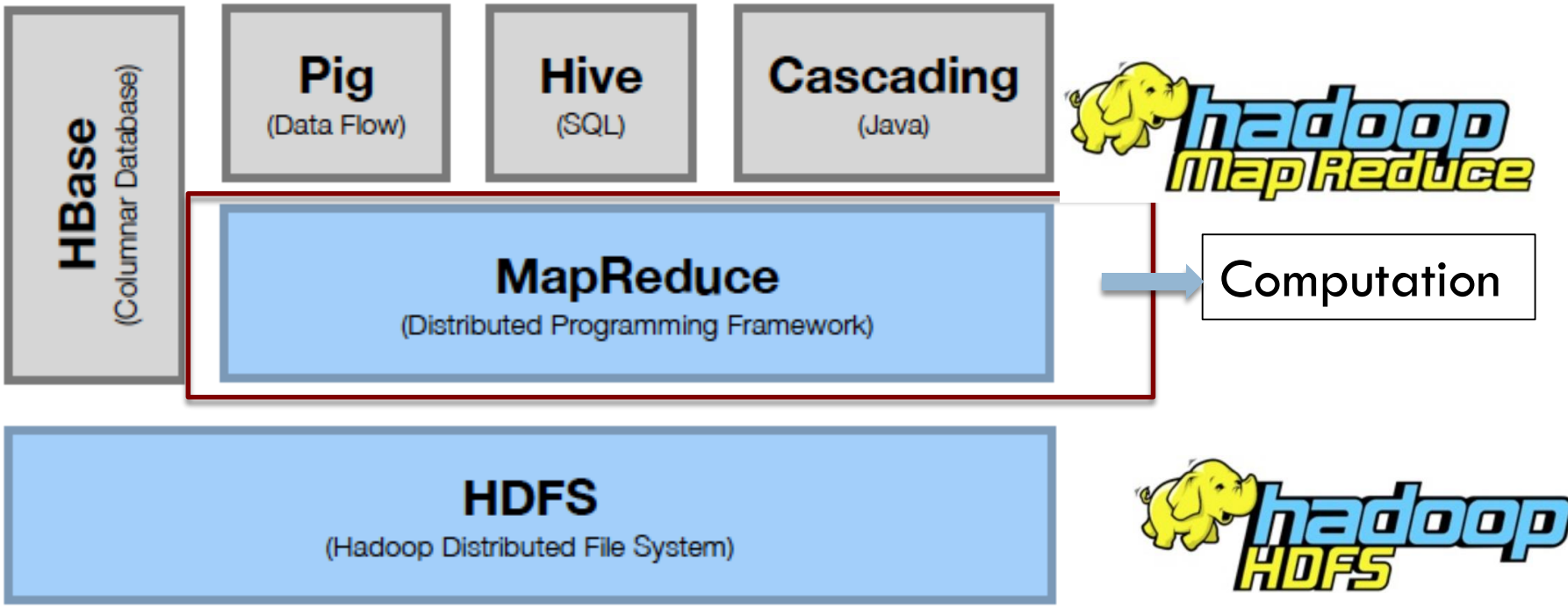
MAPREDUCE

DS8003 – MGT OF BIG DATA AND TOOLS

Ryerson University

Instructor: Kanchana Padmanabhan

Hadoop CORE



MapReduce is a computing model that decomposes large data manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers

MapReduce

- It is a software framework
- Process data-sets in-parallel
- Scale across thousands of nodes on commodity hardware
- Reliable and Fault Tolerant
- Complex details are abstracted away
  - No I/O
  - No networking code
  - No synchronization

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

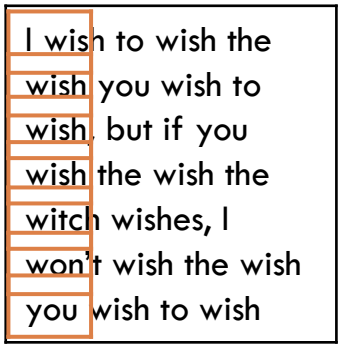
MapReduce

- Each node processes data stored on that node
- Consists of three (main) phases
  - Map
  - Combiner
  - Reduce
- Operates on <key, value> pairs

(input) -> **map** -> <k2, v2> -> **combiner** -> <k3, v3> -> **reduce** -> (output)

Word Count Example

- The WordCount program reads/scans through the document line by line
- It tokenizes/splits the line by delimiters (space, tab etc.)
- Counts the number of occurrences of each word



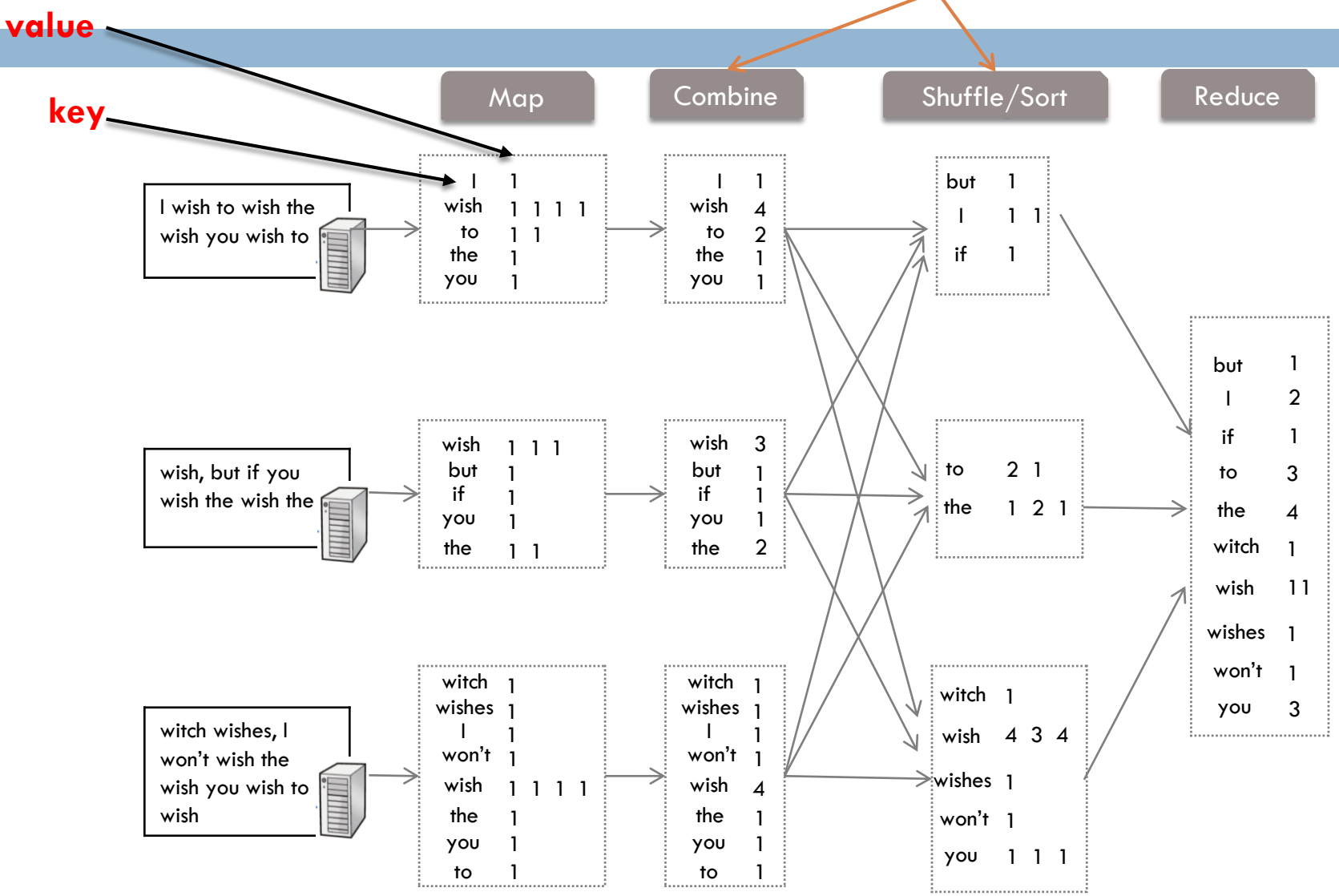
Document

I	--->	1 1
wish	--->	1 1 1 1 1 1 1 1 1 1 1
to	--->	1 1 1
the	--->	1 1 1 1
you	--->	1 1 1
but	--->	1
if	--->	1
witch	--->	1
wishes	--->	1
won't	--->	1

I	--->	2
wish	--->	1 1
to	--->	3
the	--->	4
you	--->	3
but	--->	1
if	--->	1
witch	--->	1
wishes	--->	1
won't	--->	1

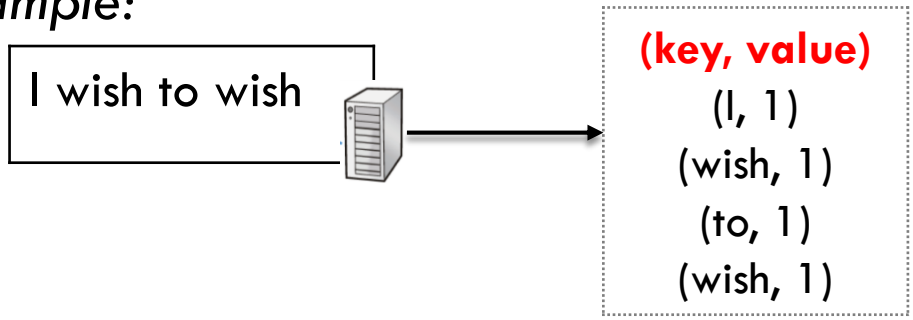
Word Count – MapReduce

MapReduce handles these automatically for you!!



Map Phase

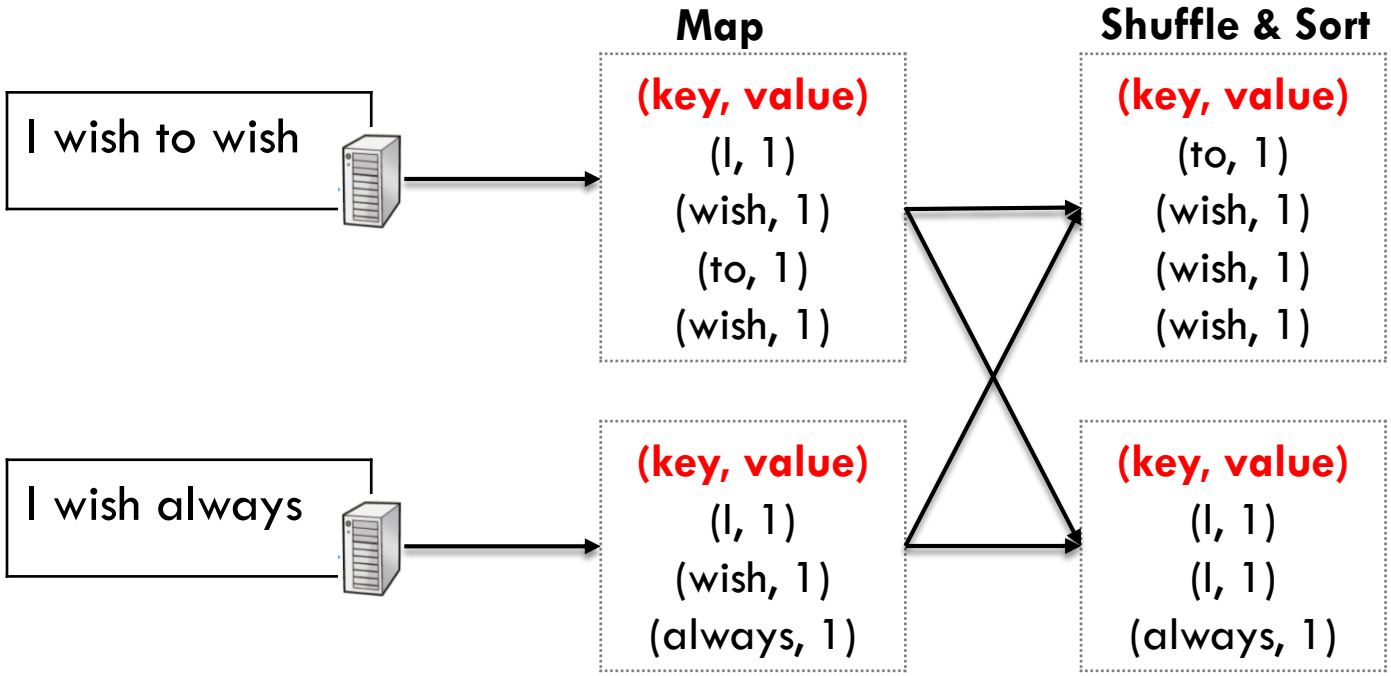
- Each mapper processes a single input split from HDFS
- Each map task process data one line at a time
- Each line is transformed into a *key* and a *value (key-value pair)*
- *Example:*



- The number of maps is usually driven by the total size of the inputs, that is, the total number of blocks of the input files.

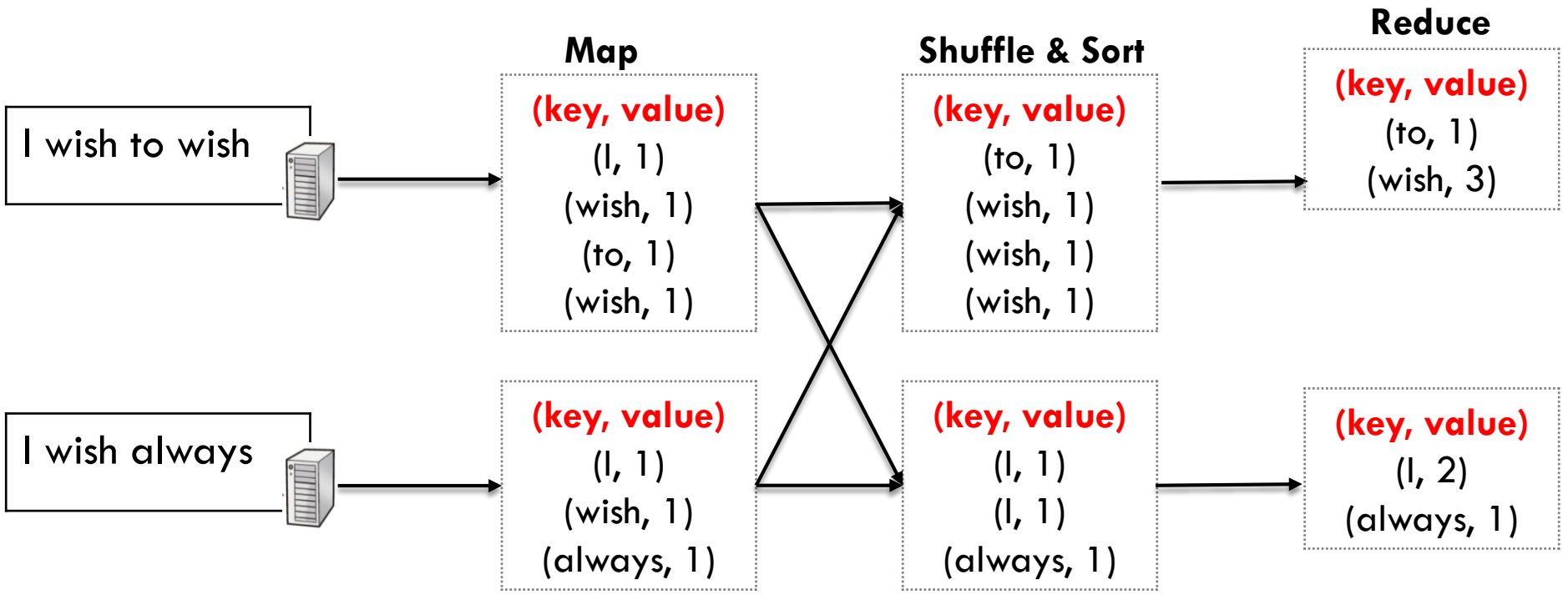
Shuffle & Sort Phase

- Makes sure that all the values associated with the same intermediate key are sent to the same reducer
- Performed automatically; Programmer does not have to code this phase



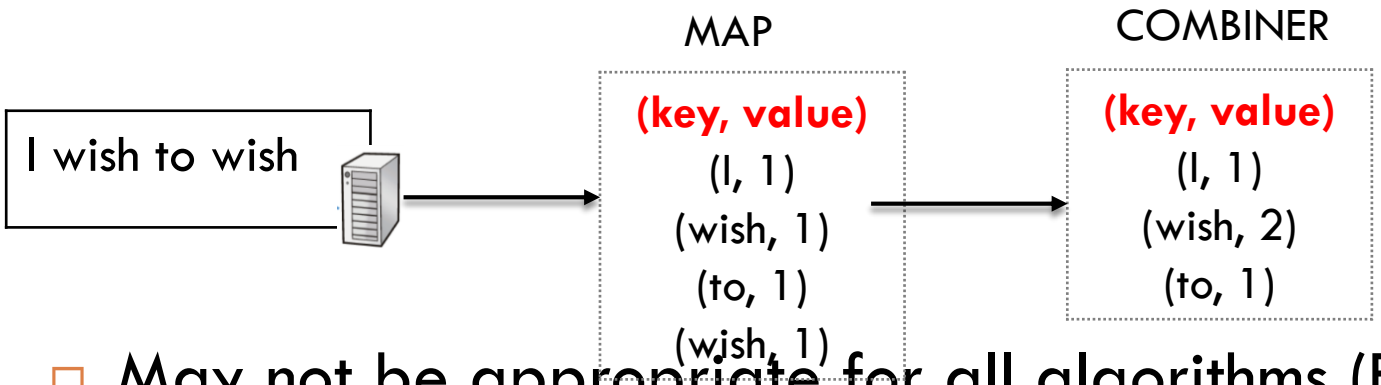
Reduce Phase

- Reducer receives the key and associated list of values and then does the reduce operations
- Reducer writes output to HDFS



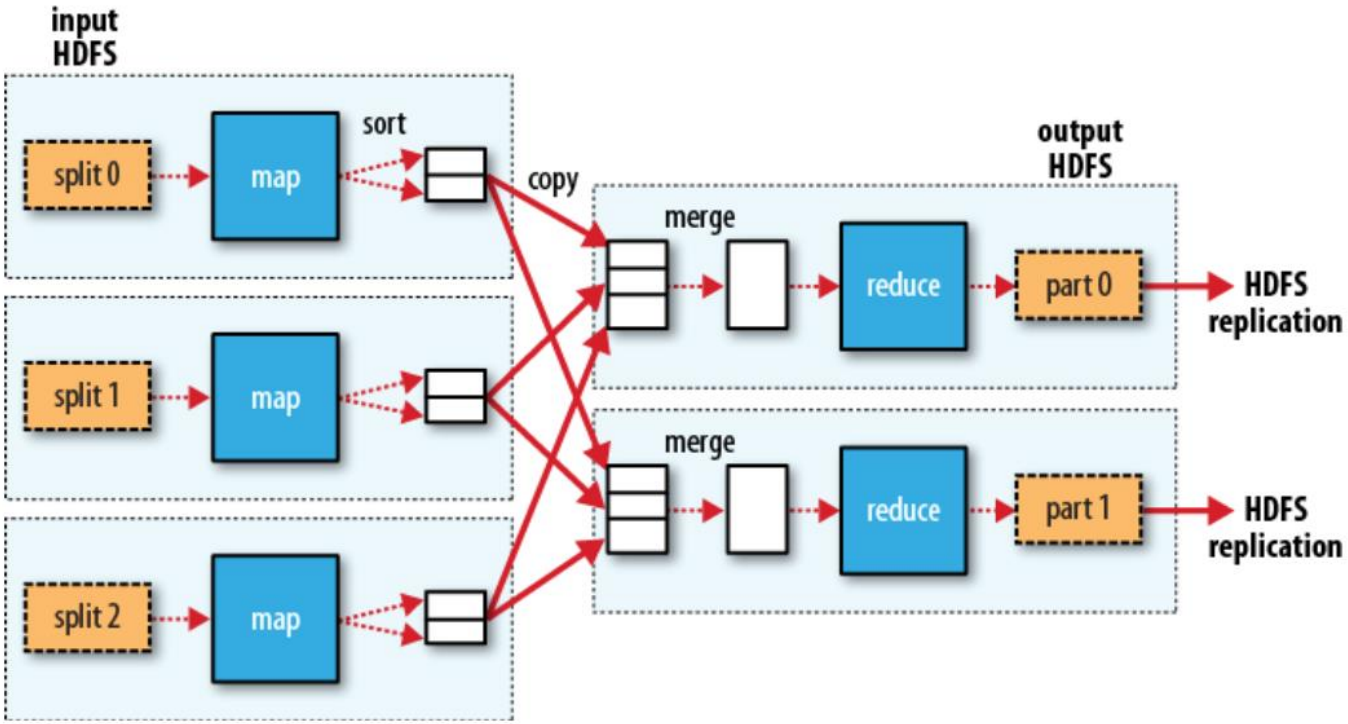
Combiner Phase

- Same as the Reducer
- Occurs on the same node as the Map Process
- Performs local aggregation, after being sorted on the *keys*



- May not be appropriate for all algorithms (Example: It may not be suitable for "mean" calculation)

MapReduce – Map & Reduce

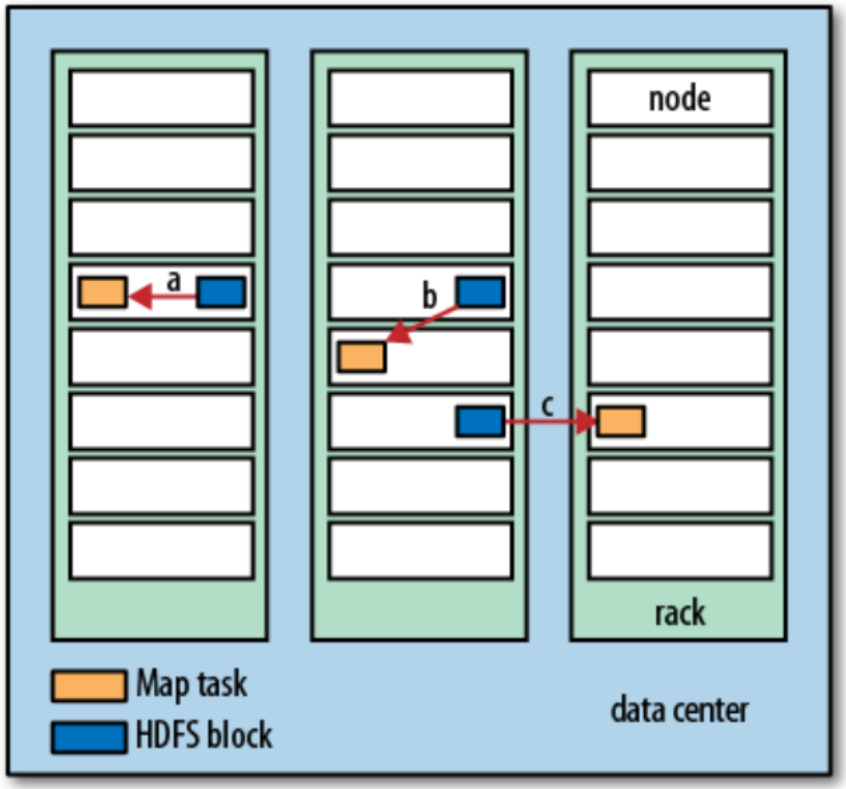


MapReduce (in Summary)

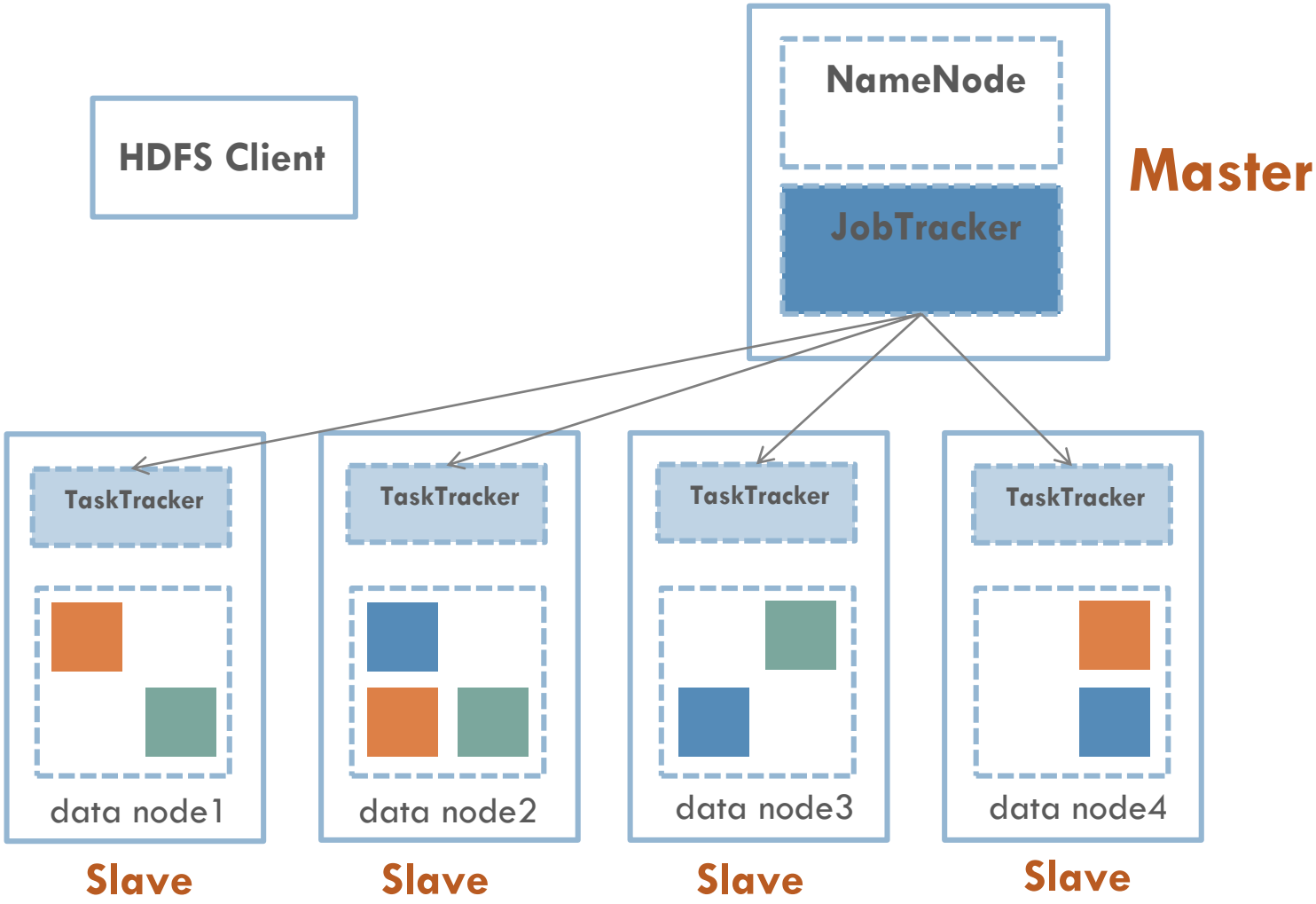
- Automatic parallelization and distribution
  - It makes M/R programming much easier
- Developer simply need to focus on writing the *map* and *reduce* functions
- M/R is written in Java
- It also supports Python Streaming
  - writing *map* and *reduce* function in python
- Typically, compute nodes and the storage nodes are the same
  - Takes Advantage of Data Locality

Data Access - Hadoop

- Data Locality
  - Hadoop tries to process data on the same machine that stores it
  - This improves performance and conserves bandwidth
  - Brings computation to the data



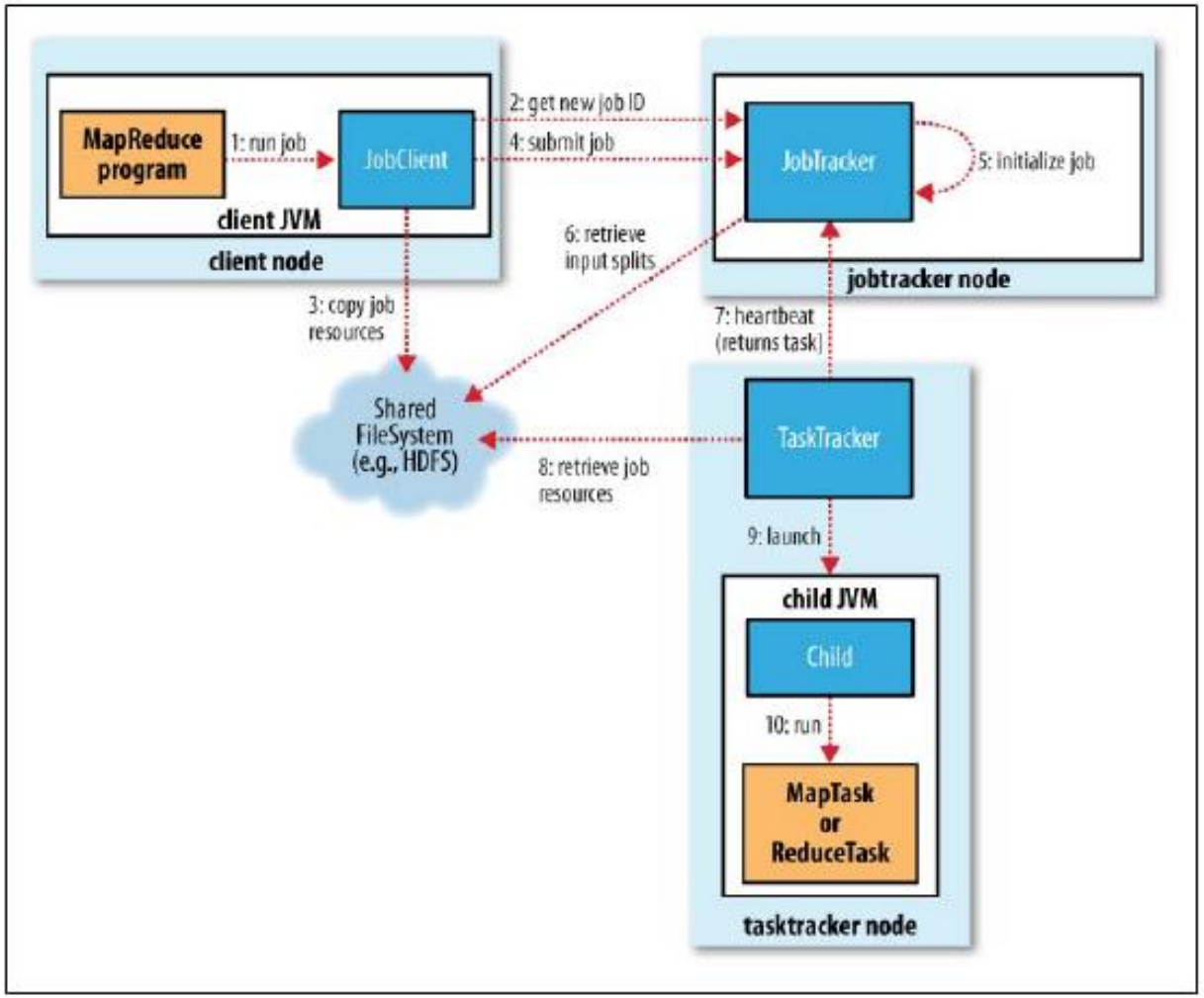
MapReduce Execution Architecture



MapReduce Execution Terminology

- Terminology
  - Job
    - Consists of a Mapper, a Reducer, and a list of input files
  - Task
    - An individual unit of work
    - A job is broken down to many tasks
      - map tasks or reduce tasks
  - Client
    - Machine on which the program runs

MapReduce Architecture



Job Tracker

- Manages the MapReduce execution
- Client contacts Job Tracker to launch jobs
- It communicates with NameNode to get data locations
- Identifies TaskTrackers to executes the job (accounts for data locality or proximity)
- Monitors TaskTrackers; Provides job status updates to the client
- JobTracker is down – no MapReduce (Similar to NameNode failure)

http://hadoopinrealworld.com/jobtracker-and-tasktracker/

Task Tracker

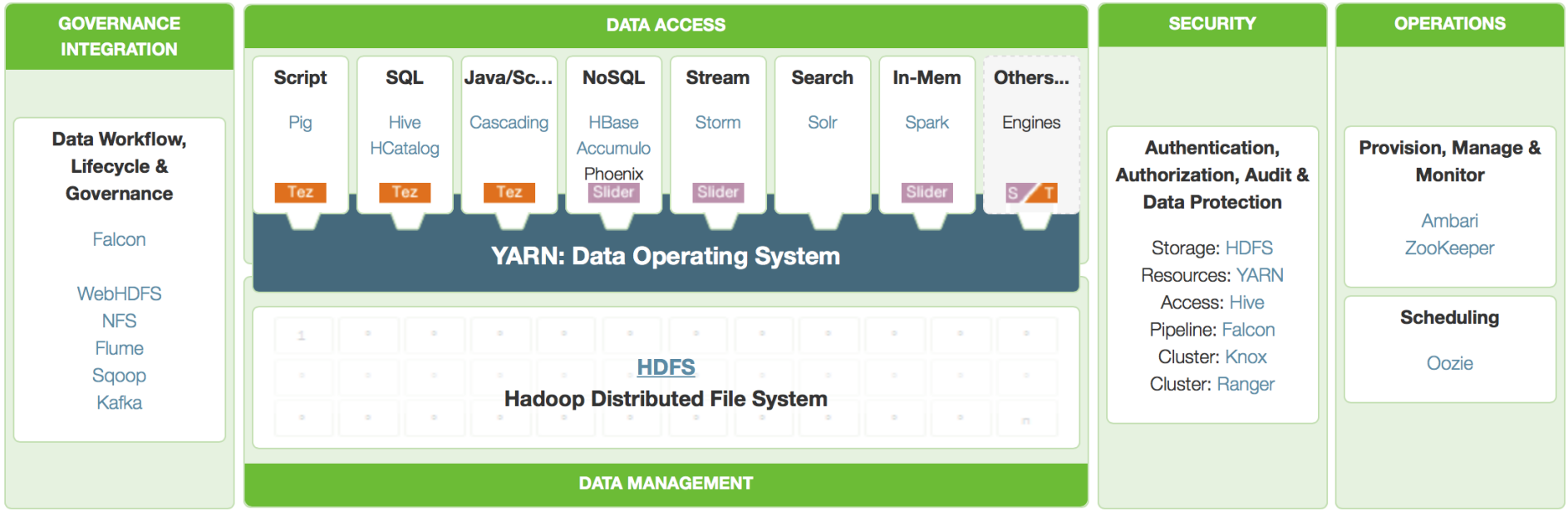
- Runs on Data Nodes (that's how it can take advantage of data locality)
- TaskTracker execute individual Map and Reduce processes
- Sends periodic updates (heartbeats) to JobTracker
- If Task Tracker fails then JobTracker will reassign the job to another Task Tracker

Fault Tolerance - Hadoop

- If a node fails, the master will notice that failure and re-assign the task to a different node on the system
- Recovering a failed node doesn't affect nodes working on other portions of the data
- If a failed node restarts, it is automatically added back to the system and assigned new task
- If a node appears to run slowly, the master can redundantly execute another instance of the same task (speculative execution)

Hadoop Ecosystem

- Data analysis
  - Hive, Pig, Spark
- Machine Learning
  - Mahout, Spark (Mllib)
- Graph processing
  - Giraph, Spark (GraphX)
- Database Integration
  - Sqoop
- Scheduling & Workflow
  - Oozie
- Cluster management
  - Ambari
- Search
  - Solr
- NoSQL
  - Hbase, Cassandra
- Stream Processing
  - Storm



YARN (Hadoop 2.0)

- YARN (**Y**et **A**nother **R**esource **N**egotiator)
- Cluster Resource Management
- Three different processes (JobTracker performed all of these functions on Hadoop 1.0)
  - Resource management (Scheduler)
  - Job and task scheduling (Applications Manager)
  - Job status information (JobHistory Server)
- TaskTracker replaced by **NodeManager**
- Tools apart from MapReduce can also schedule jobs through YARN (Example: Spark)
- JobTracker/TaskTracker were coupled tightly with the MapReduce Framework
- Has a Failover setup with the help of Zookeeper (not a single point of failure)

Recommended Readings

- (Google) Google MapReduce (link)
- (Google) The Google File System (link)
- (Microsoft) Scaling up vs scale out for hadoop: time to rethink? (link)
- (Google) Vision paper: towards an understanding of the limits of mapreduce computation (link)
- (Twitter) MapReduce is good enough? If all you have is a hammer, throw away everything that is not a nail! (link)

https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/ResourceManagerHA.html

http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-operators/

http://saphanatutorial.com/how-yarn-overcomes-mapreduce-limitations-in-hadoop-2-0/

https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html