



MAPREDUCE OVERVIEW

Bin Jiang

02/11/2017

Objective

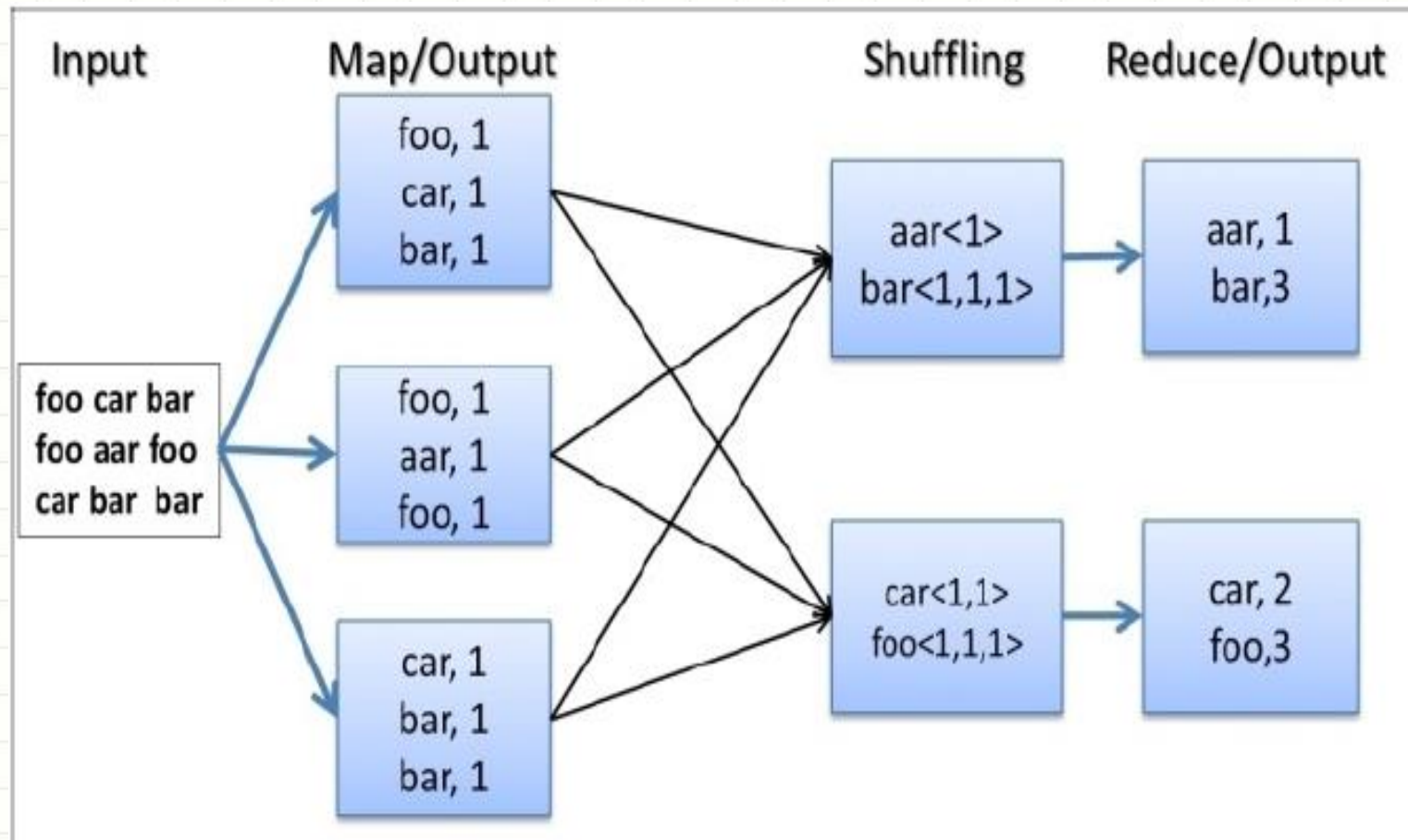
Participants will learn about

- Develop a MapReduce Application
- How does MapReduce work
- MapReduce Type and Format
- MapReduce Features

MapReduce Core Concepts

- Job
- Mapper
- Reducer
- Partitioner
- Combiner
- Sort and Shuffle
- Merge
- InputSplit

Map Reduce Hello World



Background

- Client – Server Application
- Distributed Computing Challenges
 - Process / Hardware failures
 - Data Consistency
 - High Availability
 - Optimization of Distributed Processing
- Google MapReduce

Map Phase

- Input data breaks up into a split of blocks by talking to Name node
- Each block is parsed and processed by one mapper via calling `map()` method
- Mapper output the data as key/value pairs to the local disk and waiting for the reducer to fetch

Reduce Phase

- Reducers fetch the data from the mapper and perform some sort of an aggregation operation by key (e.g. counting, find the max value, calculate the average etc.)
- The output of the reducer can be put into HDFS

Shuffle and Sort Phase

- Sits between the map and reduce phases
- Process the output of the mapper by consolidating and merging the values by key
- Performing the data transfer between nodes
- Sorting the data by key
- Sending the arranged data to the same reducer

Combine Phase

- Sits together with the map phase
- Minimize the data transfer between the map and reduce phase
- Run on the top of map output
- Output of the combiner become the input of the reducer

Summary

- Map stage transform original input into key-value pair
- Combine stage combine the output of the map stage to reduce the data transfer during the shuffle stage
- Shuffle stage merges values with the same key
- Reduce stage processes all the values with the same key

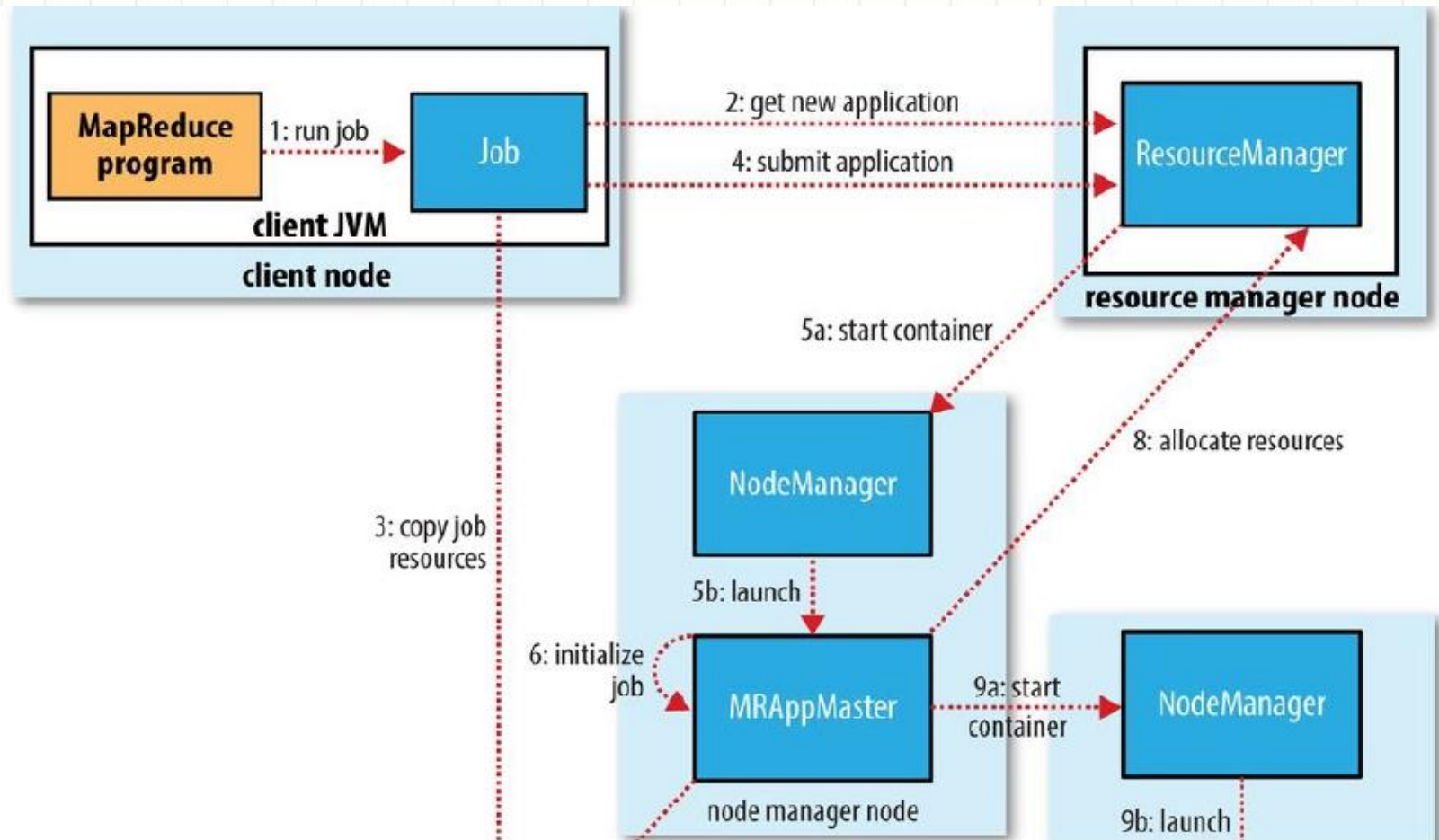
MapReduce Overview

- Shared-Nothing Architecture
- Similarity with SQL Aggregation Operation
 - Aggregation is achieved by using avg, sum, count with the group by clause
 - select name as customer, count(*) as num_of_order
where country = 'canada'
group by name
having count(*) > 10
order by name desc
 - The map and combine phase: select, order by name, where
 - The reduce phase: group by and having

MapReduce Overview

- Fault-tolerance of MapReduce
- Typical MapReduce Jobs
 - Web Crawler
 - PageRanking
 - Processing geographical data
 - The reduce phrase: group by and having
- Dive and conquer pattern
- Data Locality

How does MapReduce work



How does MapReduce work

- Job submission
- Job Initialization
- Task assignment
- Task execution
- Process and status update
- Job completion

How does MapReduce work

- Job submission
 - Ask the Resource Manager to get application ID
 - Check output directory
 - Compute the input splits
 - Copy the resources needed for the job
 - Submit the job

How does MapReduce work

- Job Initialization
 - Resource Manager receive the job request
 - RM checks the YARN scheduler to allocate the container
 - RM asks the node manager to launch the container for the application master
 - Application master decides how to run the mapreduce jobs (uber tasks)
 - AM calls the output committer to create the output folder

How does MapReduce work

- Task assignment
 - AM request to Resource Manager for the container to run the map and reduce tasks
 - Request for the reducer are not made until 5% of mapper tasks have been completed
 - Mapper tasks (data locality)
 - Reducer tasks can be anywhere in the cluster

How does MapReduce work

- Task execution
 - AM contacts the node manager to launch the container to run the tasks
 - Localize the resources, e.g. jars

How does MapReduce work

- Process and status update
 - Long running batch jobs
 - Tasks heartbeat the status to AM
 - NM heartbeats the status to RM
 - AM heartbeats the status to RM

How does MapReduce work

- Job competition
 - AM changes the job status to success once get the notification from last tasks (waitForCompletion)
 - commitJob

How does MapReduce work

- Failure

- Task Failure

- ✓ Notify the application master
 - ✓ NM notify the resource manager and then notify AM to mark the failure and try three attempts
 - ✓ Heartbeat for long running tasks
 - ✓ `mapreduce.map.failures.maxpercent`

How does MapReduce work

- Failure
 - Application Master Failure
 - ✓ AM heartbeat to RM
 - ✓ RM starts new instance of AM
 - ✓ RM uses the job history to recover the state of tasks
 - ✓ Client needs to relocate the new application master by requesting to RM

How does MapReduce work

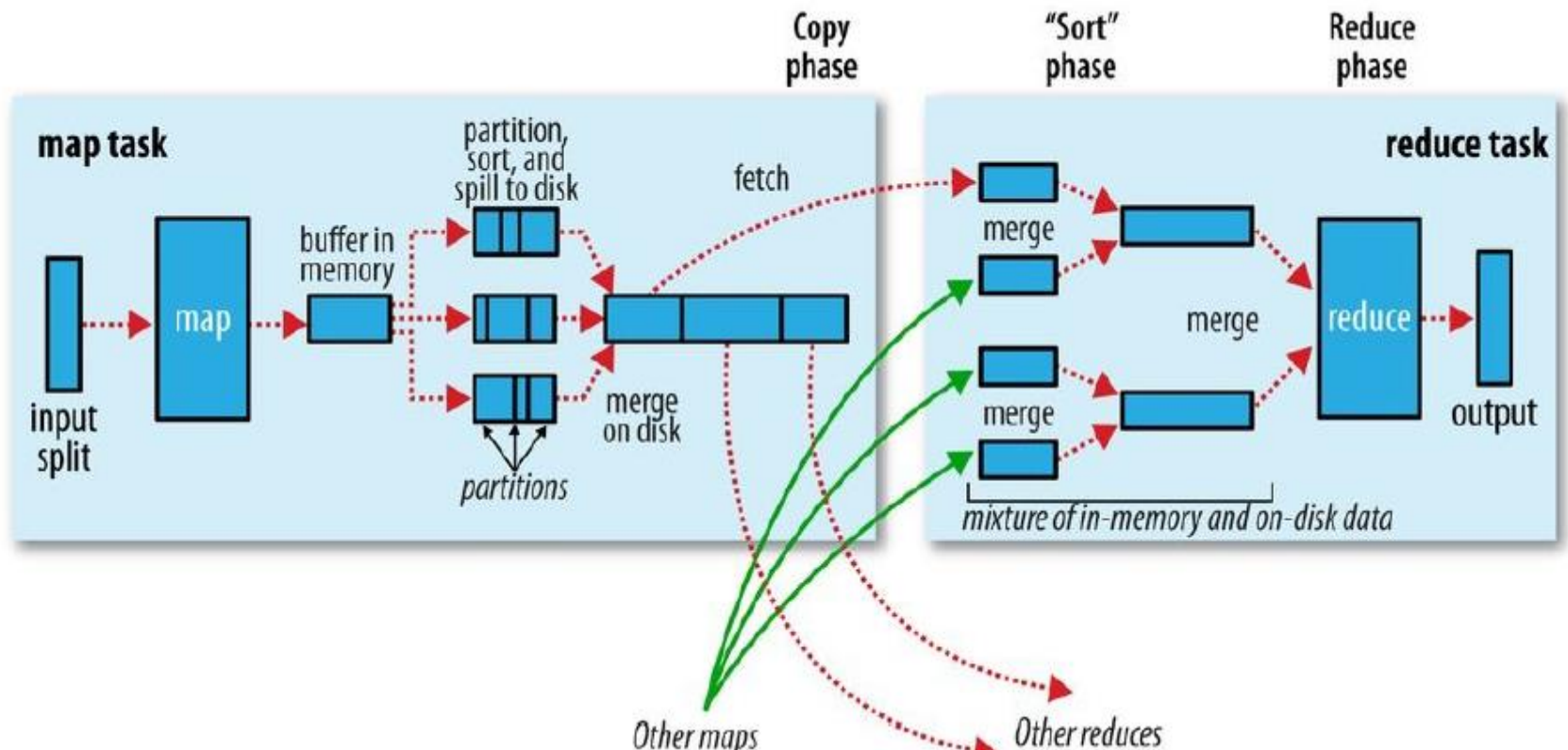
- Failure
 - Node Manager Failure
 - ✓ Stop sending heartbeat to RM
 - ✓ Blacklisted if the number of failure is high
 - ✓ Tasks in the failed node manager will be rerun since the state in the local disk is lost

How does MapReduce work

- Failure
 - Resource Manager Failure
 - ✓ HA
 - ✓ Failure controller (zookeeper)
 - ✓ Restart the application master from the state store
 - ✓ Client and node manager must be configured to handle the RM failover

Shuffle and Sort

- Map Side



Shuffle and Sort

- Reduce Side
 - How do reducers know which machines to fetch map output from?