HDFS and Map Reduce: an introduction

Get used to HDFS

Write your first MapReduce project

Big Data, a scaling problem

Scaling

Vertical scale:

- more powerful machine → Exponential cost
- Monolithic softwares

Horizontal scale:

- More machines → Linear cost
- Distributed softwares

How to distribute ?

- Not an easy task
- Not always feasible
- Difficult to implement from scratch

MapReduce Algorithm

- Very simple batch-oriented data processing model
- Highly scalable (20PB/day at Google in 2008)
- A reusable pattern → a generic framework can be implemented

Hadoop MapReduce

- Framework on top of HDFS
- Only need to implement Map() and Reduce()
- Execution is handled by the Framework
- Fault tolerant

MapReduce Principles

- Input reader → list(k1,v1)
  >1TB file(s) → indexed splits (64MB)
- Map(k1,v1) → list(k2,v2)
- Combine (local aggregation)
- Shuffle/sort
- Reduce(k2, list(v2)) → list(k3,v3)
- Output writer

MapReduce Example

- Word Count (MR Hello World)
- Input file :

  Hadoop uses MapReduce.

  There is a Map phase

  There is a Reduce phase

MapReduce Example

- Input reader will split by line:

  1. Hadoop uses MapReduce
  2. There is a Map phase
  3.
  4. There is a Reduce phase

MapReduce Example

- 4 Mapper will be called:

1. (Hadoop,1)

   (uses,1)

   (MapReduce,1)
2. ...
3. //Generates nothing
4. ...

MapReduce Example

- Shuffle and sort:

  (a,[1,1])

  (Hadoop,1)

  (is,1)

  (Map,1)

  (MapReduce,1)

  (phase,[1,1])

  (Reduce,1)

  (There,[1,1])

  (uses,1)

MapReduce Example

- 9 Reduces:

  (a,2)

  (Hadoop,1)

  (is,1)

  (Map,1)

  (MapReduce,1)

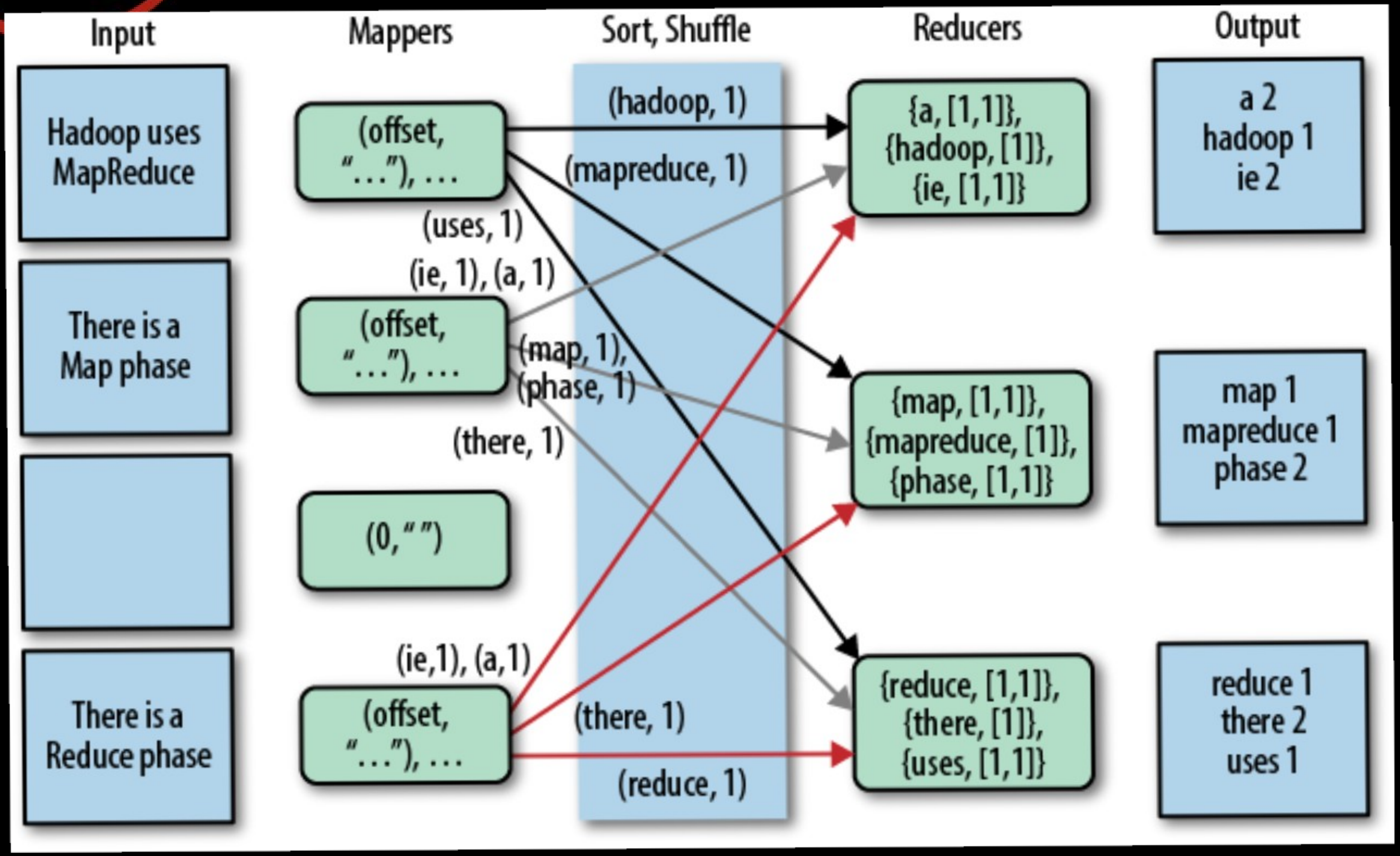
  (phase,2)

  (Reduce,1)

  (There,2)

  (uses,2)

MapReduce Example



MapReduce Example

- 9 Reduces but not 9 reducers !!!
- Number of reducers can be specified
- The partitioner class handle distribution over reducers

MapReduce Example

- Output Writer:

a	2
Hadoop	1
is	1
Map	1
MapReduce	1
phase	2
Reduce	1
There	2
uses	2

MapReduce Example

- Same with a combiner
- Input file :

  Hadoop uses Hadoop MapReduce.

  There is a Hadoop Map phase

  There is a Hadoop Reduce phase

MapReduce Example

- 4 Mapper will be called:
  1. (Hadoop,1)

     (uses,1)

     (Hadoop,1)

     (MapReduce,1)
  2. ...
  3. //Generates nothing
  4. ...

MapReduce Example

- Combiner may be used:

  1. (Hadoop,2)

     (uses,1)

     (MapReduce,1)

- Running the combiner function makes for a more compact map output, so there is less data to write to local disk and to transfer to the reducer

MapReduce Example

- Shuffle and sort:

  (a,[1,1])

  (Hadoop,[2,1,1])

  (is,1)

  (Map,1)

  (MapReduce,1)

  (phase,[1,1])

  (Reduce,1)

  (There,[1,1])

  (uses,1)

MapReduce Example

- 9 Reduces:

  (a,2)

  (Hadoop,4)

  (is,1)

  (Map,1)

  (MapReduce,1)

  (phase,2)

  (Reduce,1)

  (There,2)

  (uses,2)

MapReduce Example

- Custom Output Writer:

  2 “a” found

  4 “Hadoop” found

  1 “is” found

  1 “Map” found

  1 “MapReduce” found

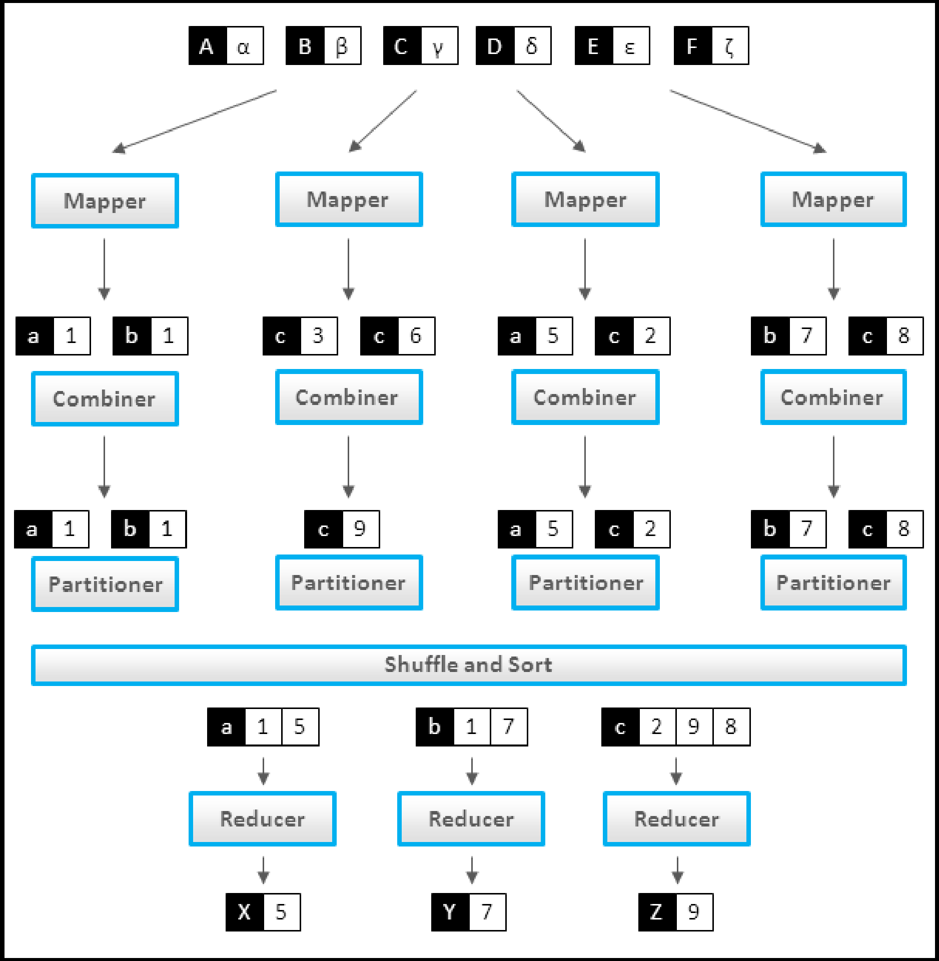
  2 “phase” found

  1 “Reduce” found

  2 “There” found

  2 “uses” found

Workflow of MapReduce Job



http://lintool.github.io/MapReduceAlgorithms/

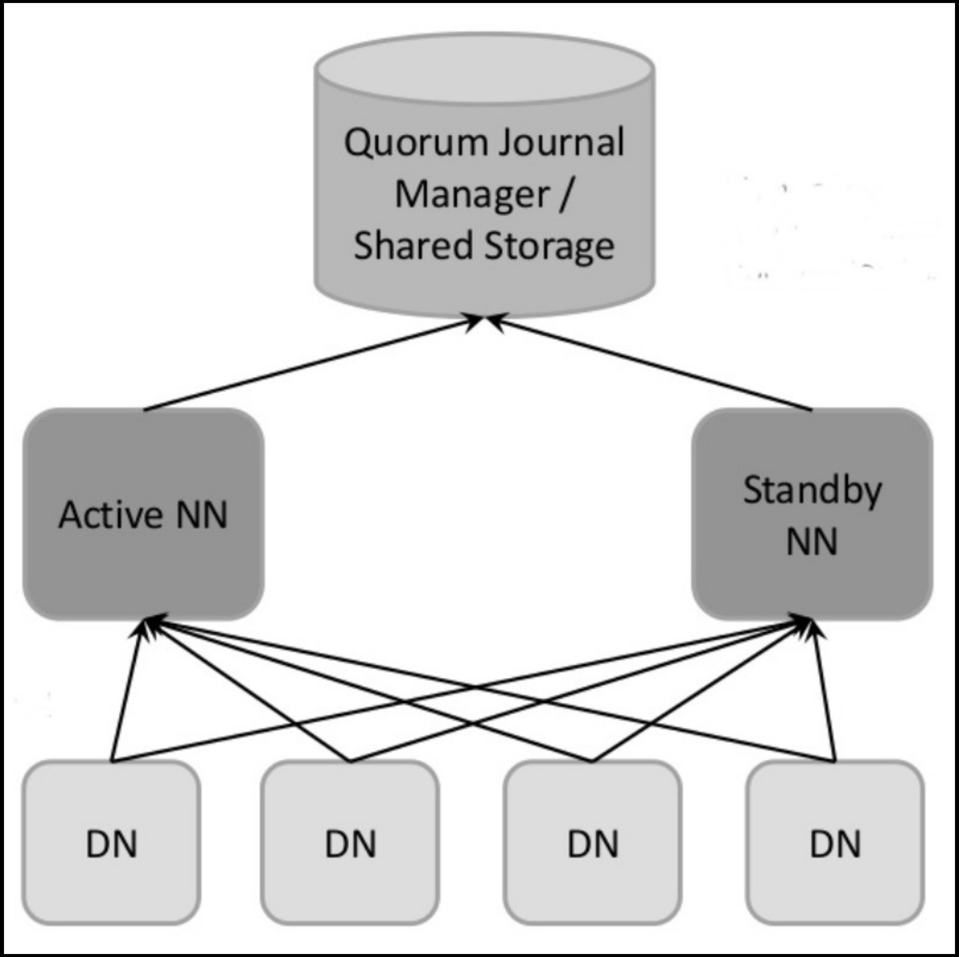
HDFS

What Hadoop Distributed FileSystem Does ?

HDFS

- Provide distributed, replicated file system
- Handle disk and/or host crash
- Rack awareness
- WORM → do not support updates

HDFS Architecture



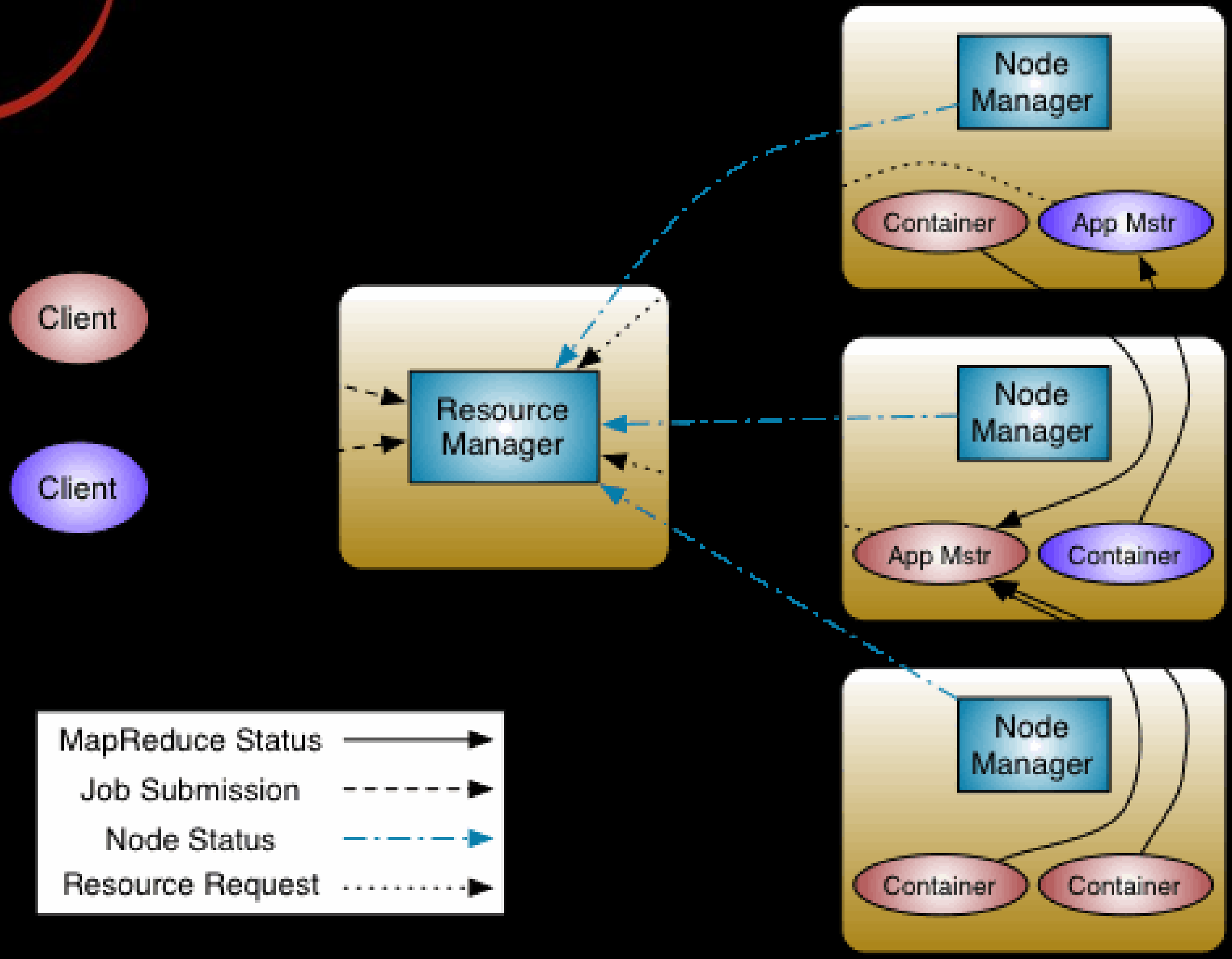
YARN

What Yet Another Resource Negotiator (often called MapReduce 2.0) Does ?

YARN

- Dispatch Hadoop application
- Load-balancing (policy can be customized)
- Handle disk and/or host crash
- Can launch non-MR application
- Support Docker Container

HDFS Architecture


Client
Client
Resource Manager
Node Manager
Container
App Mstr
Node Manager
App Mstr
Container
Node Manager
Container
Container
MapReduce Status
Job Submission
Node Status
Resource Request