PES UNIVERSITY ONLINE

BIG DATA

Job Management and YARN

K V Subramaniam
Computer Science and Engineering

What we have learnt so far..

- Data processing distributed over a cluster – Map Reduce
- Job Submission Flow
- How does job management actually happen?
- How is failure management addressed?

… handled by **YARN**

Lecture Overview

- Need for YARN - history
- YARN Architecture
- Job submission lifecycle – YARN
- Scheduling
- Failure Handling
- Benefits of YARN

Big Data: The need for YARN

Motivation

- Recall
  - Job – the entire map reduce application
  - Task – Individual mappers/reducers

- How do we
  - Allocate resources – determine which nodes will run the jobs
  - Monitor the tasks – start new tasks or restart failed/slow tasks
  - Monitor the overall state of the job?

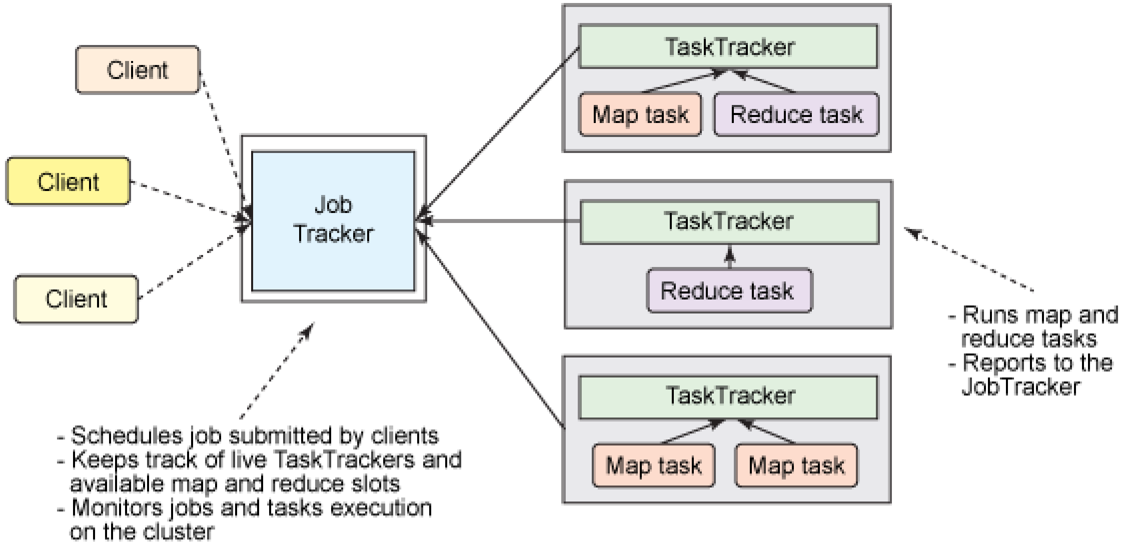
BIG DATA

Hadoop 1.0 Job Management

- Job Tracker
  - Manage Cluster resources
  - Job scheduling

- Task Tracker
  - One per task
  - Manage the task

- Fault Tolerance, Cluster resource management and scheduling handled by JobTracker



- Schedules job submitted by clients
- Keeps track of live TaskTrackers and available map and reduce slots
- Monitors jobs and tasks execution on the cluster

- Runs map and reduce tasks
- Reports to the JobTracker

Hadoop 1.0 Issues

Limits scalability

- Job tracker runs on a single machine and is responsible for cluster management, scheduling and monitoring

Availability

- JobTracker is the single point of availability/failure

Resource utilization problems

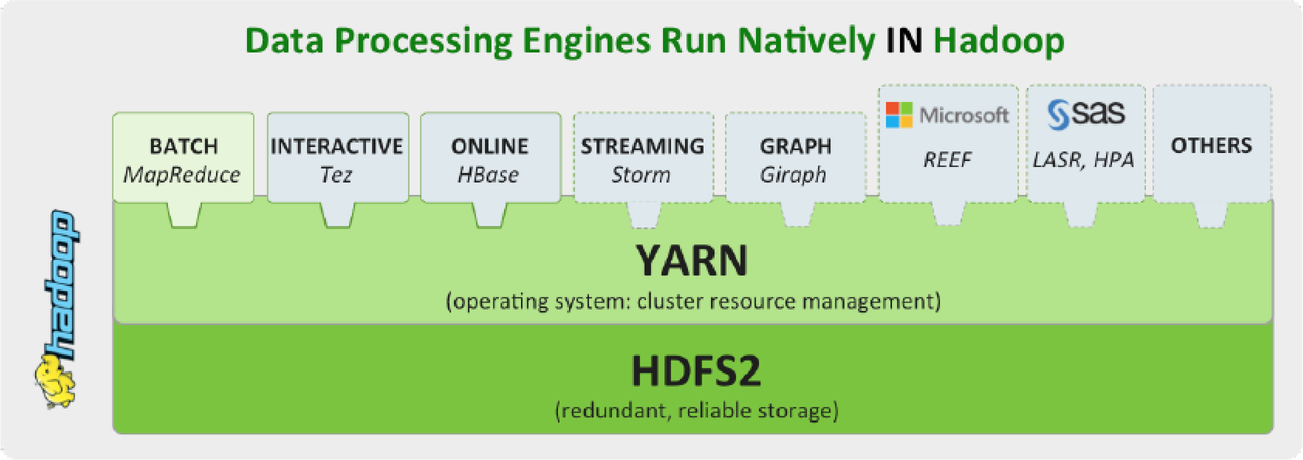
- Predefined #map/reduce slots. Utilization issues because map slots may be full but reduce slots are free.

Limitation in running MR applications

- Tightly integrated with Hadoop. Only MR apps can run. Can’t coexist with other applications.

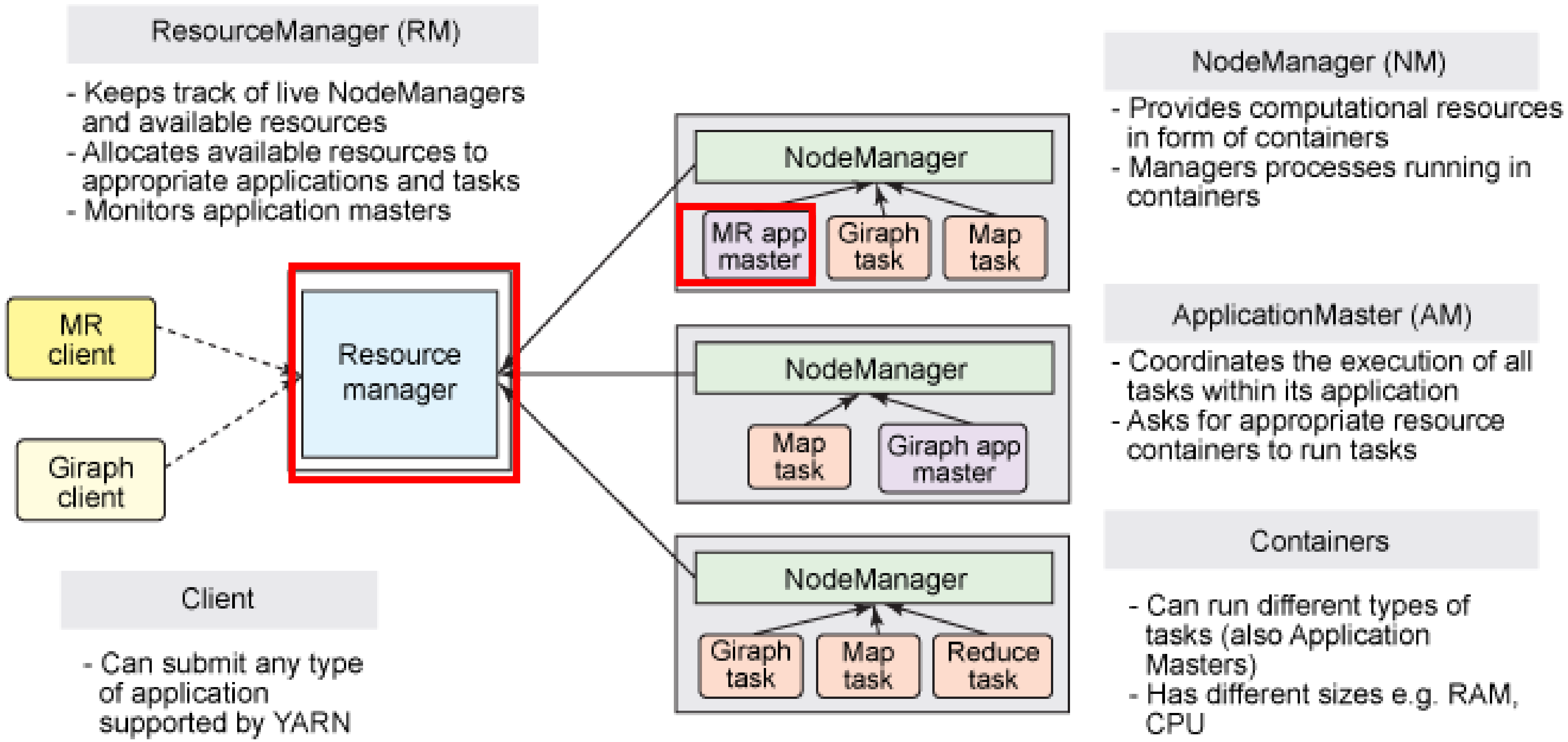
Big Data: YARN Architecture

Map Reduce - Motivation


Data Processing Engines Run Natively IN Hadoop
BATCH
MapReduce
INTERACTIVE
Tez
ONLINE
HBase
STREAMING
Storm
GRAPH
Giraph
Microsoft
REEF
sas
LASR, HPA
OTHERS
YARN
(operating system: cluster resource management)
HDFS2
(redundant, reliable storage)
hadoop

- Issues in managing clusters > 4000 nodes
- 2010 – MapReduce v2 with YARN
  - Yet Another Resource Negotiator
  - YARN Application Resource Negotiator!!

YARN Architecture



ResourceManager (RM)

- Keeps track of live NodeManagers and available resources
- Allocates available resources to appropriate applications and tasks
- Monitors application masters

NodeManager (NM)

- Provides computational resources in form of containers
- Managers processes running in containers

ApplicationMaster (AM)

- Coordinates the execution of all tasks within its application
- Asks for appropriate resource containers to run tasks

Containers

- Can run different types of tasks (also Application Masters)
- Has different sizes e.g. RAM, CPU

Client

- Can submit any type of application supported by YARN

- Split responsibility of Job Tracker
- Resource Manager – manage cluster wide resources
- Application Master – manage lifecycle of application

YARN Components

Resource Manager

- Arbitrates resources amongst all applications of the system

Node Manager

- Per machine slave
- Responsible for launching application containers
- Monitors resource usage

Application Master

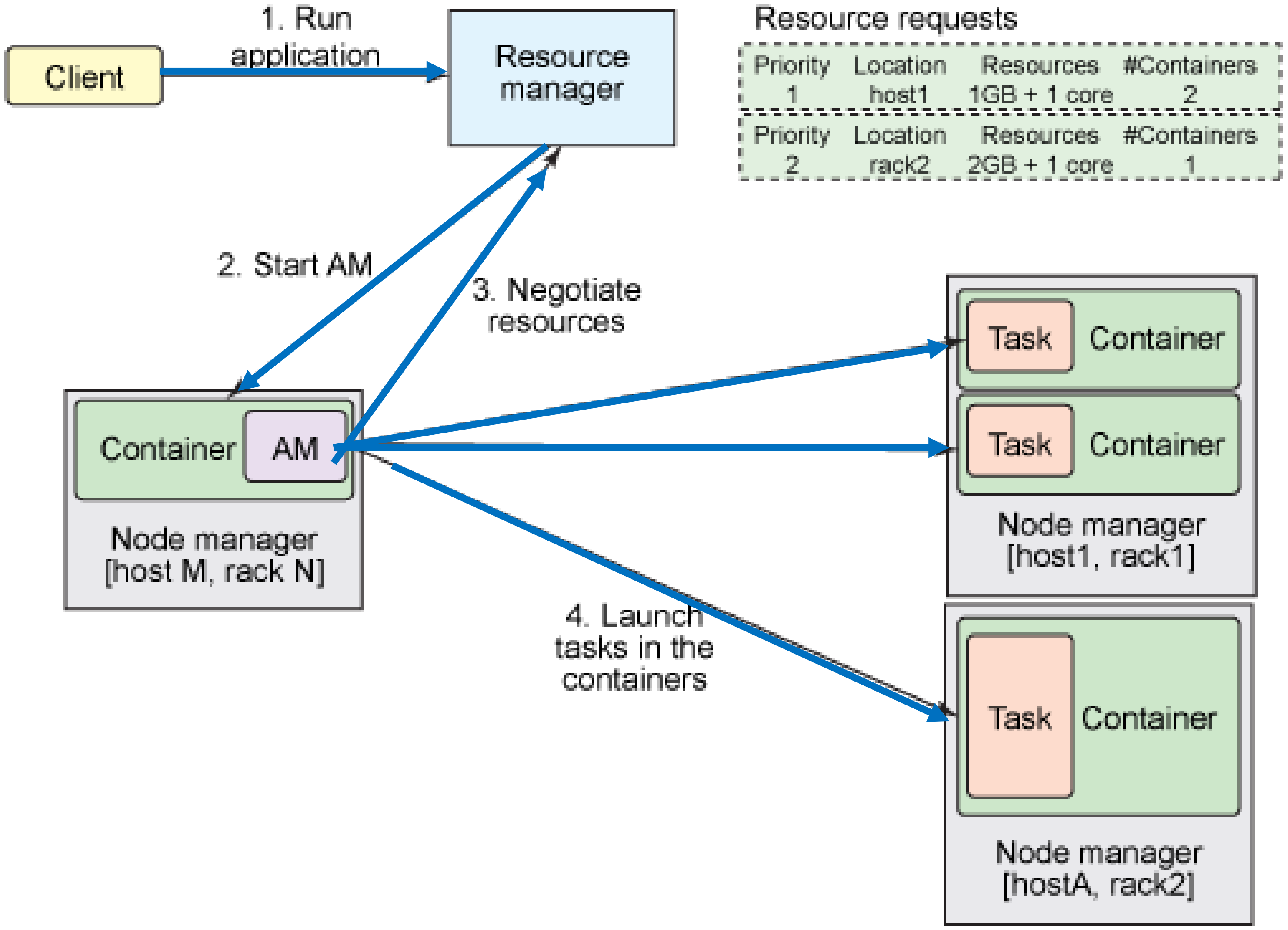
- Negotiate appropriate resource containers from the scheduler
- Track and monitor the progress of the containers

Container

- Unit of allocation incorporating resources such as memory, CPU, disk

Big Data: Job Submission - YARN

YARN Working



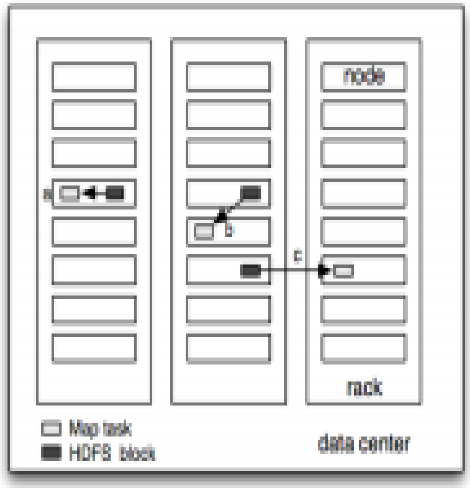
Resource requests

Priority	Location	Resources	#Containers
1	host1	1GB + 1 core	2
2	rack2	2GB + 1 core	1

Data Locality in Map Reduce

- Attempts to run the map task on a node where the input data resides in HDFS.
  - *data locality optimization* - it doesn't use valuable cluster bandwidth.
- What happens when all nodes hosting the block replicas are busy?
  - look for a free map slot on a node in the same rack as one of the blocks.
- Very occasionally even this is not possible, so an off-rack node is used, which results in an inter-rack network transfer.

.



Scheduling in YARN

Schedulers in Hadoop

- Early Hadoop versions → simplistic FIFO scheduler
  - In order of submission
  - each job would use the whole cluster
  - so jobs had to wait their turn.
- How to share resources fairly?
- Balance between
  - Production jobs
  - Ad-hoc jobs

Fair Scheduler

- Aims to give every user a fair share of the cluster capacity over time.
- Jobs are placed in pools,
  - Default → each user gets their own pool.
- If a single job is running, it gets all of the cluster.
- As more jobs are submitted,
  - free task slots are given to the jobs in such a way as to give each user a fair share of the cluster.
- Short job – completes in reasonable time
- Long job – can continue making progress.

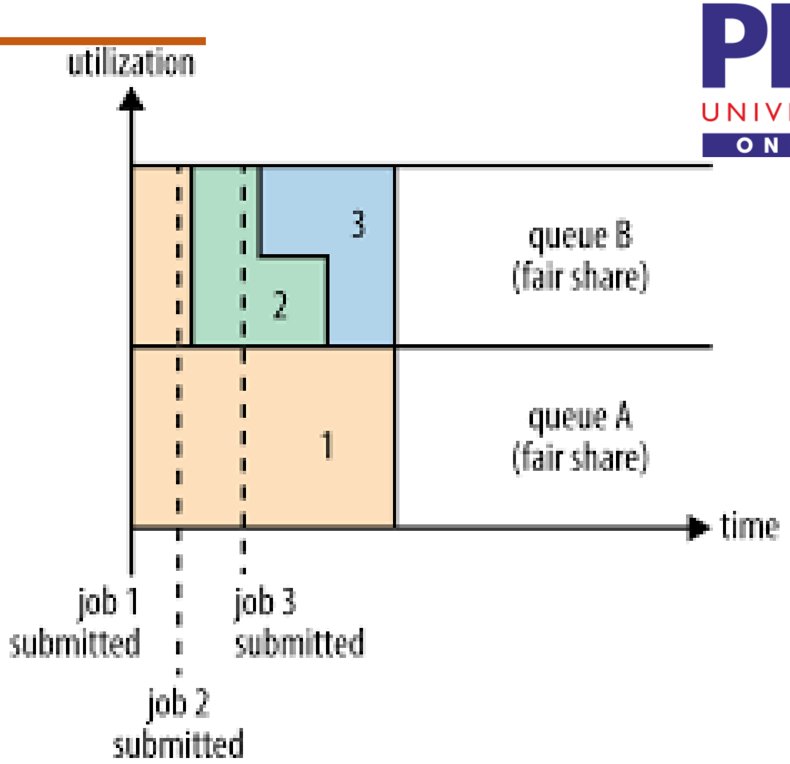


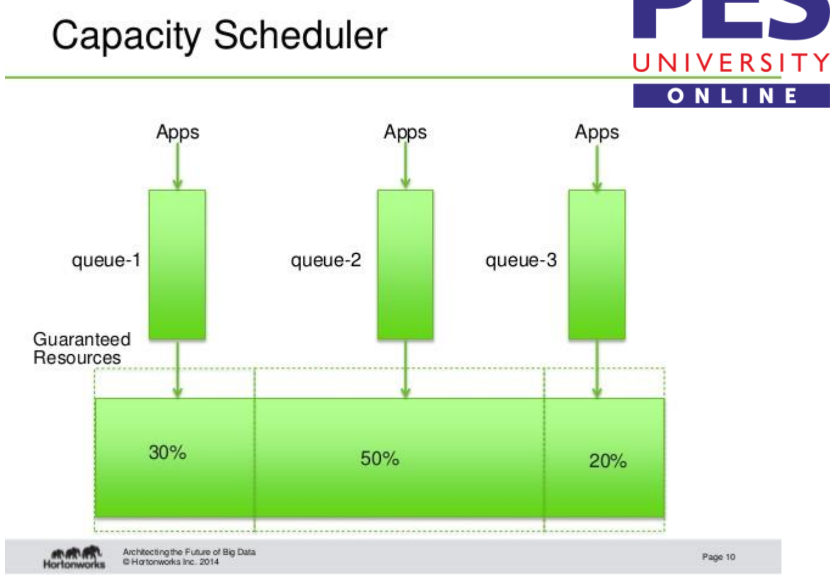
Image courtesy: Tom White, "Hadoop the definitive guide"

Fair Scheduler

- Consider a user who submits more jobs
  - Scheduler ensures → that user does not hog the cluster
- Custom pools
  - Guaranteed minimum capacities with map/reduce slots
  - It is also possible to define custom pools with guaranteed minimum capacities defined in terms of the number of map
- The Fair Scheduler supports preemption
  - If pool not received its fair share over certain time
  - scheduler will kill tasks in pools running over capacity

Capacity Scheduler

- Different approach
- number of queues (like the Fair Scheduler’s pools),
  - Has an allocated capacity
  - Can be hierarchical
  - Within each queue → scheduled using FIFO (with priorities)
- Cannot use free spare capacity even if it exists
- Like breaking up cluster into smaller clusters



Capacity Scheduler

Architecting the Future of Big Data
© Hortonworks Inc. 2014

Page 10

Image: https://www.slideshare.net/Hadoop_Summit/w-525hall1shenv2

Handling Failures

What can fail?

- Task
- Application Manager
- Resource Manager
- Node Manager

Task Failure

Due to runtime exceptions	• JVM reports error back to parent application master
Hanging tasks	• Progress updates not happening for 10 mins • Timeout value can be set.
Killed tasks	• Speculative duplicates can be killed
Recovery	• AM tries restarting task on a different node

Application Master Failure

When can failure occur?

- Due to hardware or network failures

How to detect for failures?

- AM sends periodic heartbeats to Resource Manager

Restart

- Max-attempts to restart application
    - Default = 2

Node Manager Failure

When can failure occur?	• Hardware, crashing, slow network
How to detect for failures?	• When a heartbeat is not received by RM for 10mins
Restart	• Tasks of incomplete jobs will be rerun – maybe on different node

Resource Manager Failure

How is failure handled?

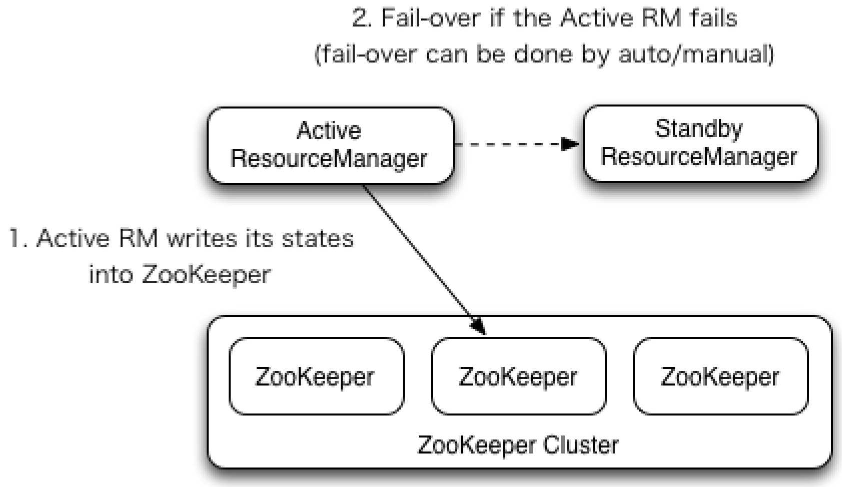
- Active Standby configuration

Impact

- More serious as all tasks fail

Restart

- Handled by failover controller


2. Fail-over if the Active RM fails
(fail-over can be done by auto/manual)
Active ResourceManager
Standby ResourceManager
1. Active RM writes its states into ZooKeeper
ZooKeeper
ZooKeeper
ZooKeeper
ZooKeeper Cluster

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

Benefits of YARN

YARN Benefits – Case Study @Yahoo

- YARN manages a very large cluster at Yahoo
  - Scalability – to over 40,000 servers with 100,000 CPUs, 455 PB of data
    - Runs over 850,000 jobs per day
  - Flexibility
    - Same cluster has Hadoop, Storm and Spark (100 node cluster) sharing resources using YARN

https://www.techrepublic.com/article/why-the-worlds-largest-hadoop-installation-may-soon-become-the-norm/

Review Exercises

Review Exercises

- All problems listed in T1 as part of LO2.5

Solved Exercises

- A 1000 node YARN cluster has no jobs running. Two pools are configured with max of 50% of the resources. A new job requiring 600 nodes is submitted and on starting consumes all 600 nodes. Which YARN scheduler is active?
  - Either FIFO or Fair because they will use the entire cluster if there is no other job.
- Will the failure of task result in failure of the entire job?
  - No. Task will be restarted
- What are speculative duplicates?
  - Tasks that are started when AM determines that there is a slow running task.

Additional Notes, Reference Material and Notes

YARN Further Reading

- Chapter 2.5 of T1
- Chapter 4 in T2
- https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html
- There is a good description of YARN in the Tom White book.
- Also follow links from slides given before

PES UNIVERSITY ONLINE

THANK YOU

K V Subramaniam

Dept. of Computer Science and Engineering

subramaniamkv@pes.edu