CS 696 Intro to Big Data: Tools and Methods
Fall Semester, 2016
Doc 19 HDFS, YARN
Nov 3, 2016

Installing Hadoop - Linux & mac

Download Hadoop tar file

http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/SingleCluster.html

Extract files from tar file

Hadoop Commands

To make it easier to access hadoop command line commands

Define HADOOP_HOME in your shell to be top level directory of hadoop you downloaded

Add to path

$HADOOP_HOME/bin
$HADOOP_HOME/sbin

bin commands	sbin scripts
hadoop	scripts to start/stop
hdfs	
yarn	
rcc	
mapred	

Hadoop Commands

Textbook assumes in your path

$HADOOP_HOME/bin

$HADOOP_HOME/sbin

Hadoop on-line documentation

Sometimes assumes current working directory is $HADOOP_HOME

bin/hdfs namenode -format

Sometimes assumes added bin & sbin to your path

Setup passphraseless ssh

If you don’t do this each time you start HDFS and YARN daemon you will

Have to login in multiple times

Hadoop Command

```
pro 18->hadoop

Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]

CLASSNAME            run the class named CLASSNAME
 or
  where COMMAND is one of:
  fs                   run a generic filesystem user client
  version              print the version
  jar <jar>            run a jar file
  classpath            prints the class path needed to get the
```

(Not showing some commands)

Running a Sample Program

grep: A map/reduce program that counts the matches of a regex in the input.

Change directory to your hadoop installation

```
$ mkdir input
$ cp etc/hadoop/*.xml input
$ ls input
```

capacity-scheduler.xml hadoop-policy.xml httpfs-site.xml kms-site.xml
core-site.xml hdfs-site.xml kms-acls.xml yarn-site.xml

```
$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
    grep input output 'dfs[a-z.]+'
$ ls output
```

_SUCCESS part-r-00000

```
$ cat output/part-r-00000
    1       dfsadmin
```

Provided Hadoop Examples

Al pro 22->hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar

Lists the examples

bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
dbcount: An example job that count the pageview counts from a database.
distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
grep: A map/reduce program that counts the matches of a regex in the input.
join: A job that effects a join over sorted, equally partitioned datasets
pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
pi
randomtextwriter, randomwriter: writes 10GB of random data per node.
secondarysort: An example defining a secondary sort to the reduce.
sort
sudoku
teragen, terasort, teravalidate
wordcount, wordmean, wordmedian, wordstandarddeviation, multifilewc,
aggregatewordcount, aggregatewordhist

Finding

General command

```
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
    exampleName exampleArgs
```

Finding arguments

```
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
    exampleName
```

Native Libraries

For performance some components of hadoop have native libraries

- Compression (bzip2, lz4, snappy, zlib)
- Native io utilities
- CRC32 checksum

Only on GNU/Linux

- RHEL4/Fedora
- Unbuntu
- Gentoo

On other systems uses Java implementation

16/11/02 09:12:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Tesing for native support

hadoop checknative -a

16/11/02 09:28:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Native library checking:

hadoop: false

zlib: false

snappy: false

lz4:   false

bzip2:   false

openssl: false

16/11/02 09:28:08 INFO util.ExitUtil: Exiting with status 1

Compiling Hadoop Java Programs

Need Jar files in the following directories

{HADOOP_HOME}/share/hadoop/common
{HADOOP_HOME}/share/hadoop/common/lib
{HADOOP_HOME}/share/hadoop/mapreduce
{HADOOP_HOME}/share/hadoop/yarn
{HADOOP_HOME}/share/hadoop/hdfs

Using Hadoop to compile

Setup

```
export JAVA_HOME=/path/to/your/java/install
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

Compiling

```
hadoop com.sun.tools.javac.Main ListOfJavaClassFiles
```

Packaging into Jar file - jar command

jar

Program in java distribution

Compresses class files & adds manifest

Usage: jar {ctxui}[vfmn0PMe] [jar-file] [manifest-file] [entry-point] [-C dir] files ...

jar cf yourJarFileName.jar listOfClassFiles

Follow the Example at

http://hadoop.apache.org/docs/r2.7.3/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

https://goo.gl/D9KxE1

Running a Program

hadoop jar yourJarFileName.jar ClassWithMain arg0 arg1 ...

You need a main in a class that configures the job

Arguments (arg0, arg1) in the above command are passed to main

Often input and output directories

Maven & Hadoop

For maven users

hadoop jar files are in the standard maven repository

Chapter 6 of the textbook

Eclipse & Hadoop

http://hdt.incubator.apache.org

Does not seem like an active project

There are several third party eclipse hadoop plugins

HDFS

Hadoop Distributed Filesystem HDFS

Parts of a file are distributed on different machine

Large files - 100 MB, GB or TB

- File block size - 128MB or larger for efficient transfer

Streaming data access

- Copy to HDSF once
- Read many times

Handles node failure

High-latency access

Single Writer, append only

Namenode & Datanodes

- Namenode
  - master
  - Manages filesystem
    - Filesystem tree & metadata for files * directories
  - Clients interact with namenode
  - Cluster may contain multiple namenodes
    - Federation
      - Divide namespace up if too many files
    - High Availability
      - Backup if main namenode fails

- Datanode
  - worker
  - Reads file blocks
  - Reports to name node which blocks it contains

Things that can go wrong

Datanode fails

Namenode fails

Network partition

Network/name node pause

Datanode fails

Each block of a file is stored on multiple machines

This is set in conf file

For standalone & Pseudo distributed set to 1

hdfs-site.xml

HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

Namenode

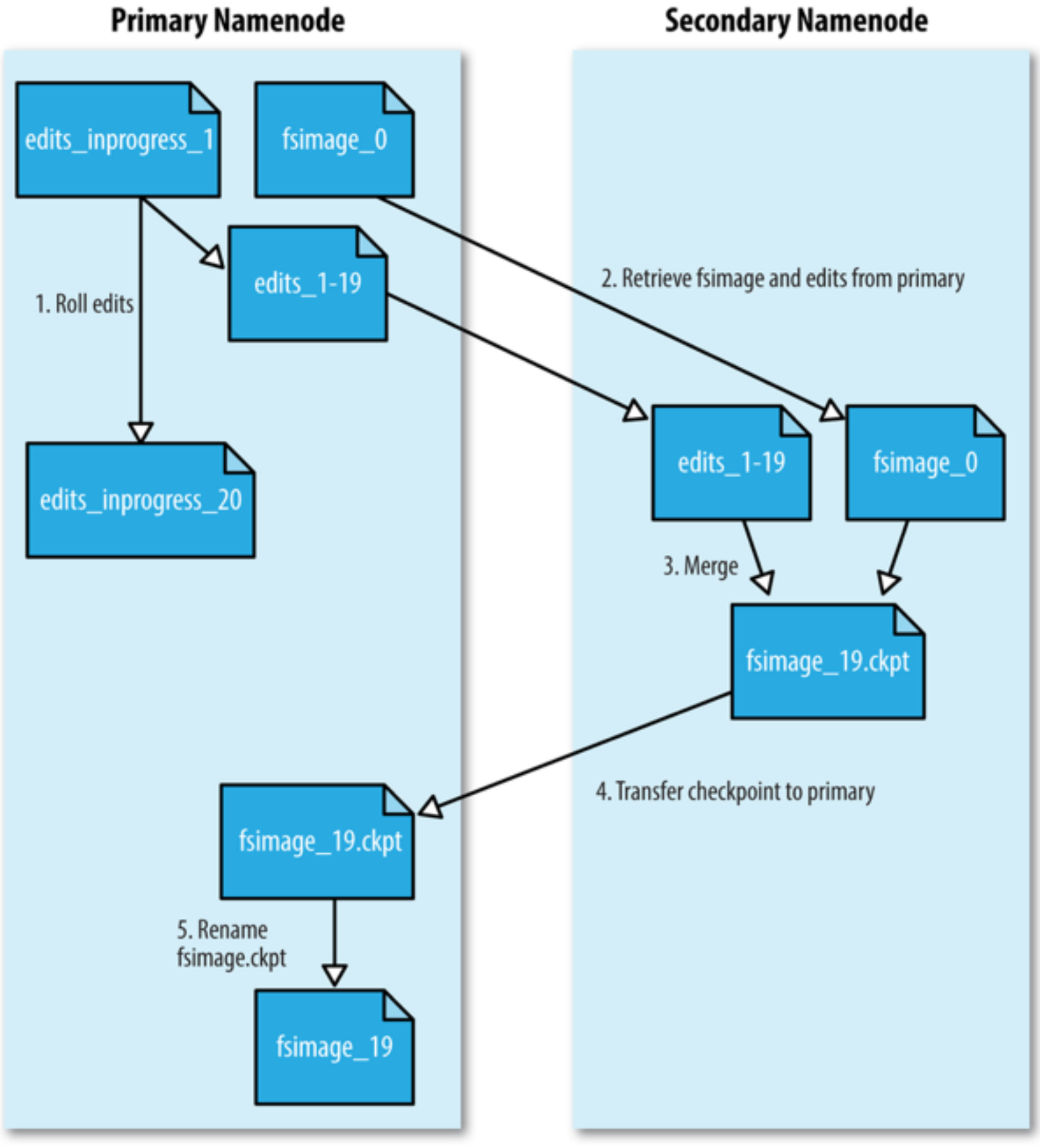
Single point of failure

Keeps filesystem data in memory

Writes current state
- Local disk
- Shared filesystem
  - NSF or Quorum journal mangager (QJM)

Writes log of changes
- Local disk
- Shared filesystem

Second namenode
- Periodically requests snapshot of state and rest of log


Primary Namenode
Secondary Namenode
edits_inprogress_1
fsimage_0
edits_1-19
1. Roll edits
edits_inprogress_20
2. Retrieve fsimage and edits from primary
edits_1-19
fsimage_0
3. Merge
fsimage_19.ckpt
4. Transfer checkpoint to primary
fsimage_19.ckpt
5. Rename
fsimage.ckpt
fsimage_19

Interfaces to HDFS

Java

Command line

HTTP Interface

HDFS, Standalone & Pseudo-Distributed

Standalone

- Does not use HDFS
- Uses local file system

Pseudo-Distributed

- HDFS files stored in /tmp
- Only one datanode

Hadoop fs options

appendToFile
cat
checksum
chgrp
chmod
chown
copyFromLocal
copyToLocal
count
cp
createSnapshot
deleteSnapshot
df
du
expunge
find
get
getfattr
getmerge
help
ls
mkdir
moveFromLocal
moveToLocal
mv
put
renameSnapshot
rm
rmdir
setfacl
setfattr
setrep
stat
tail
test
text
touchz
truncate
usage

http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/FileSystemShell.html

HDFS Configuration

etc/hadoop/core-site.xml

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

Not clear why 9000

Some commands when accessing remote sites default to 8020

etc/hadoop/hdfs-site.xml

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

HDFS Setup & Use

Format filesystem

hdfs namenode -format

Start NameNode & DataNode daemons

start-dfs.sh

Create HDSF directories needed

hdfs dfs -mkdir /user
hdfs dfs -mkdir /user/yourNameHere

View NameNode info

http://localhost:50070/

hadoop support dfs commands

hdfs dfs -mkdir

hadoop fs -mkdir

http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/SingleCluster.html

WebView - http://localhost:50070

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/user/whitney/input Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	whitney	supergroup	4.33 KB	11/1/2016, 1:05:25 PM	1	128 MB	capacity-scheduler.xml
-rw-r--r--	whitney	supergroup	908 B	11/1/2016, 1:05:26 PM	1	128 MB	core-site.xml
-rw-r--r--	whitney	supergroup	9.46 KB	11/1/2016, 1:05:26 PM	1	128 MB	hadoop-policy.xml
-rw-r--r--	whitney	supergroup	893 B	11/1/2016, 1:05:26 PM	1	128 MB	hdfs-site.xml
-rw-r--r--	whitney	supergroup	620 B	11/1/2016, 1:05:26 PM	1	128 MB	httpfs-site.xml
-rw-r--r--	whitney	supergroup	3.44 KB	11/1/2016, 1:05:26 PM	1	128 MB	kms-acls.xml
-rw-r--r--	whitney	supergroup	5.38 KB	11/1/2016, 1:05:26 PM	1	128 MB	kms-site.xml
-rw-r--r--	whitney	supergroup	690 B	11/1/2016, 1:05:26 PM	1	128 MB	yarn-site.xml

Web Access Ports Used by HDFS

Namenode	50070	dfs.http.address
Datanodes	50075	dfs.datanode.http.address
Secondarynamenode	50090	dfs.secondary.http.address

Pseudo-Distributed

When HDFS daemon is running hadoop reads input from HDFS

Grep Example Again

Change directory to your hadoop installation

Copy files to HDFS

```
$ hdfs dfs -put etc/hadoop input
```

Run the example

```
$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
        grep input output 'dfs[a-z.]+'
```

View the output

```
hdfs dfs -cat output/*
16/11/02 10:38:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library fo
platform... using builtin-java classes where applicable
1       dfsadmin
1       dfs.replication
```

Grep Example Again

Copy the output to local file system

```
hdfs dfs -get output output
```

Run the program again

```
$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
        grep input output 'dfs[a-z.]+'
```

Get Exception

```
6/11/02 10:26:02 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with processName=JobTracker,
sessionId= - already initialized
org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://localhost:9000/user/whitn
output already exists
        at
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:14
```

Grep Example Again

Deleting the output

```
$ hadoop fs -rm output/*

$ hadoop fs -rmdir output
```

Getting command options

```
$ hadoop fs -rm
-rm: Not enough arguments: expected 1 but got 0
Usage: hadoop fs [generic options] -rm [-f] [-r|-R] [-skipTrash] <src> ...
```

Easier deleting

```
$ hadoop fs -rm -R output
```

hadoop fs -cat

Usage: hadoop fs -cat URI [URI ...]

Local files

hadoop fs -cat file:///Java/hadoop-2.7.3/README.txt

HDFS files

hadoop fs -cat hdfs://localhost/user/whitney/README
hadoop fs -cat hdfs://localhost:9000/user/whitney/README
hadoop fs -cat /user/whitney/README
hadoop fs -cat README

Tutorial sets hdfs port to 9000 in conf
cat assumes port 8020 so need :9000 using full URI
Without hdfs:// cat using setting in conf file

Snapshots

Read-only copies of the filesystem

- O(1) creation time
- Blocks are not copied
- Store modifications

input, foo
top level HDFS directories
Added for example

Make a directory snapshotable

```
hdfs dfsadmin -allowSnapshot input
```

Take a snapshot

```
hdfs dfs -createSnapshot input firstSnap
```

List snapshots of a directory

```
hdfs dfs -ls input/.snapshot
```

Contents of a snapshot

```
hdfs dfs -ls input/.snapshot/firstSnap
```

Copying a shapshot file

```
hdfs dfs -cp input/.snapshot/firstSnap/core-site.xml foo
```

Advanced HDFS features

balancer

- Rebalances the data across datanodes
  - Adding new nodes or existing node runs out of space

fsck

- Find inconsistencies in filesystem

Upgrade & Rollback

DataNode Hot Swap Drive

Details on Hadoop fs commands

http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/FileSystemShell.html

Accessing HDFS

Java

Classes to interface with HDFS and other filesystem

WebHDFS

Allows non-Java clients to access HDFS remotely

```
curl -i "http://127.0.0.1:50070/webhdfs/v1/user/whitney/input?
                                        user.name=whitney&op=GETFILESTATUS
```

Hadoop Filesystems

Local	file
HDFS	hdfs
WebHDFS	webhdfs
Secure WebHDFS	swebhdfs
HAR	har
S3	s3a
Azure	wasb
Swift	swift

YARN

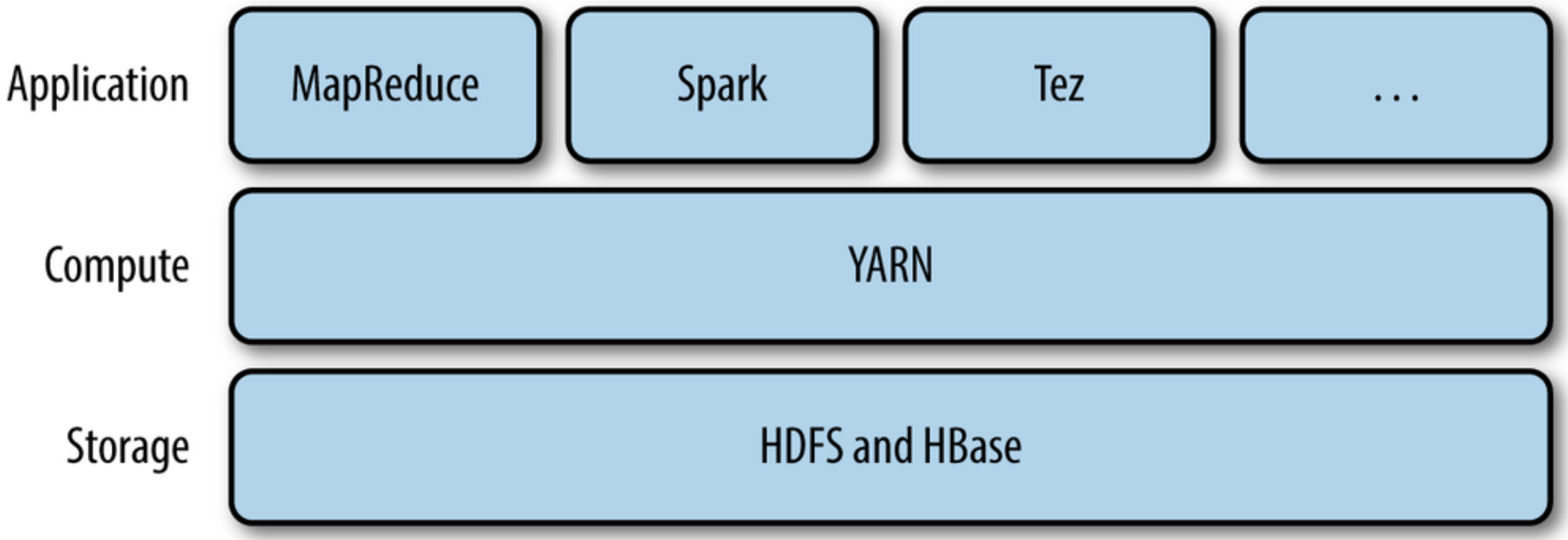
YARN

How to schedule jobs on a cluster

- Multiple requests at same time

Each request requires

- Different amount/type of resources
- Runs different length of time

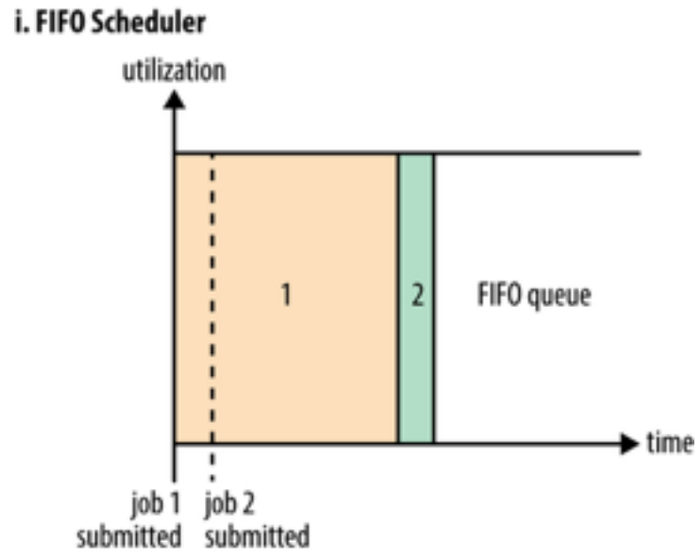


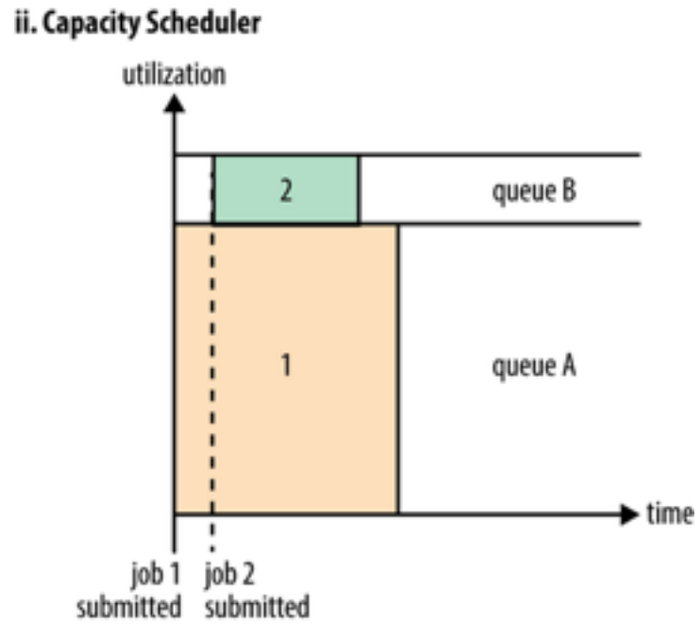
Yarn Scheduling Algorithms

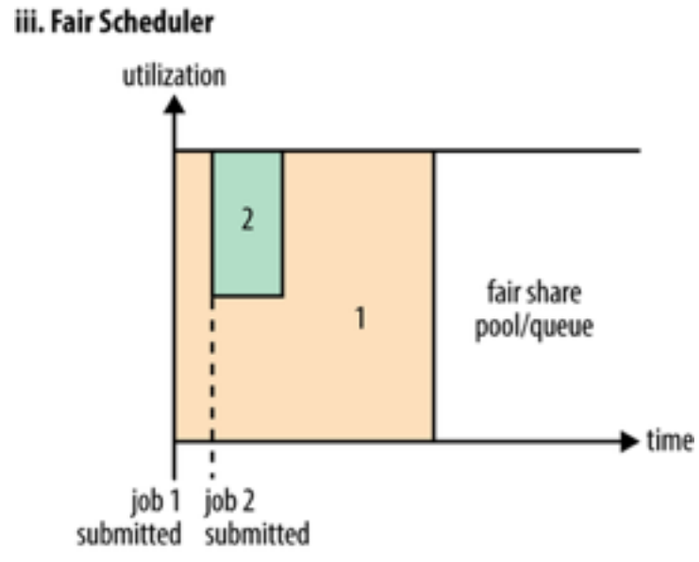
FIFO

Capacity

Fair







Yarn FIFO Scheduler

Jobs are run in the order they are submitted

YARN Capacity Scheduler

Each group

- Assigned a part of the cluster
- Has separate queue for jobs with quota of resources available

Queue elasticity

- If parts of cluster are idle a queue may be assigned more than its quota
- When demand increases wait until jobs are finished to return resources to proper queue

YARN Fair Scheduler

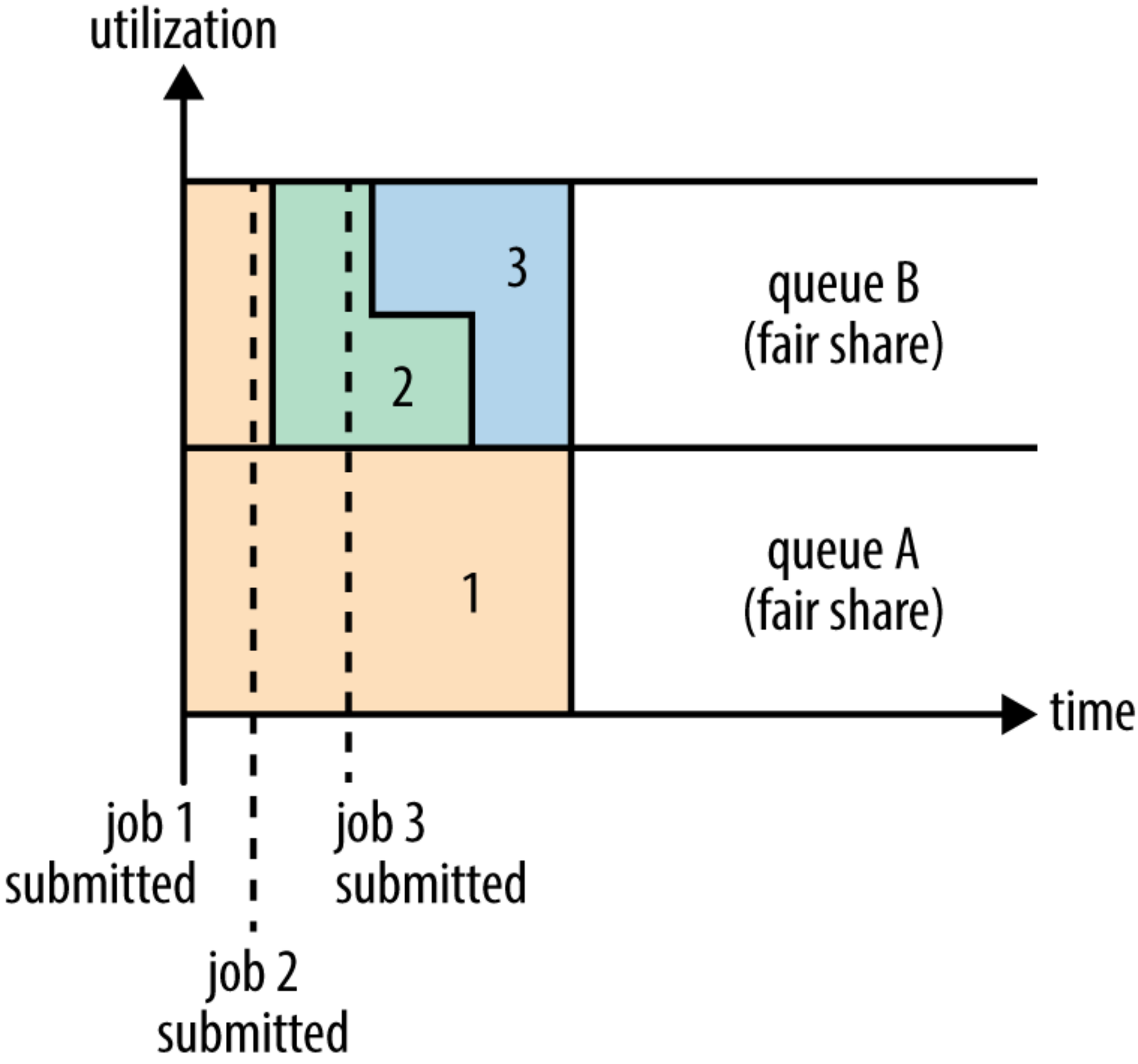
Each user/group has separate job queue

Configure what amount or resource is fair for each user

When new requests arrive

- Wait until resources are freed up
- Preempt running jobs

Each queue can have different scheduling algorithms


utilization
3
2
1
queue B
(fair share)
queue A
(fair share)
time
job 1
submitted
job 3
submitted
job 2
submitted

Delay Scheduling

What happens when a job requests a node that is busy

- Mode resources on given node to another node

Each node sends heartbeat to YARN resource manager

- Current status
- Each heartbeat is scheduling opportunity

Delay scheduling

- When requested node in busy
- Wait a given number of heartbeats before scheduling the job

Which Resource

Each job request

- CPUs
- Memory

Which resource rquirement to use to determine how much of cluster is needed?

- Default is memory

Dominant Resource Fairness

- Uses the dominant resource
- YARN can be configured to use

Running YARN

start-yarn.sh

ResourceManager - http://localhost:8088/