



YARN OVERVIEW

Bin Jiang

02/11/2017

Objective

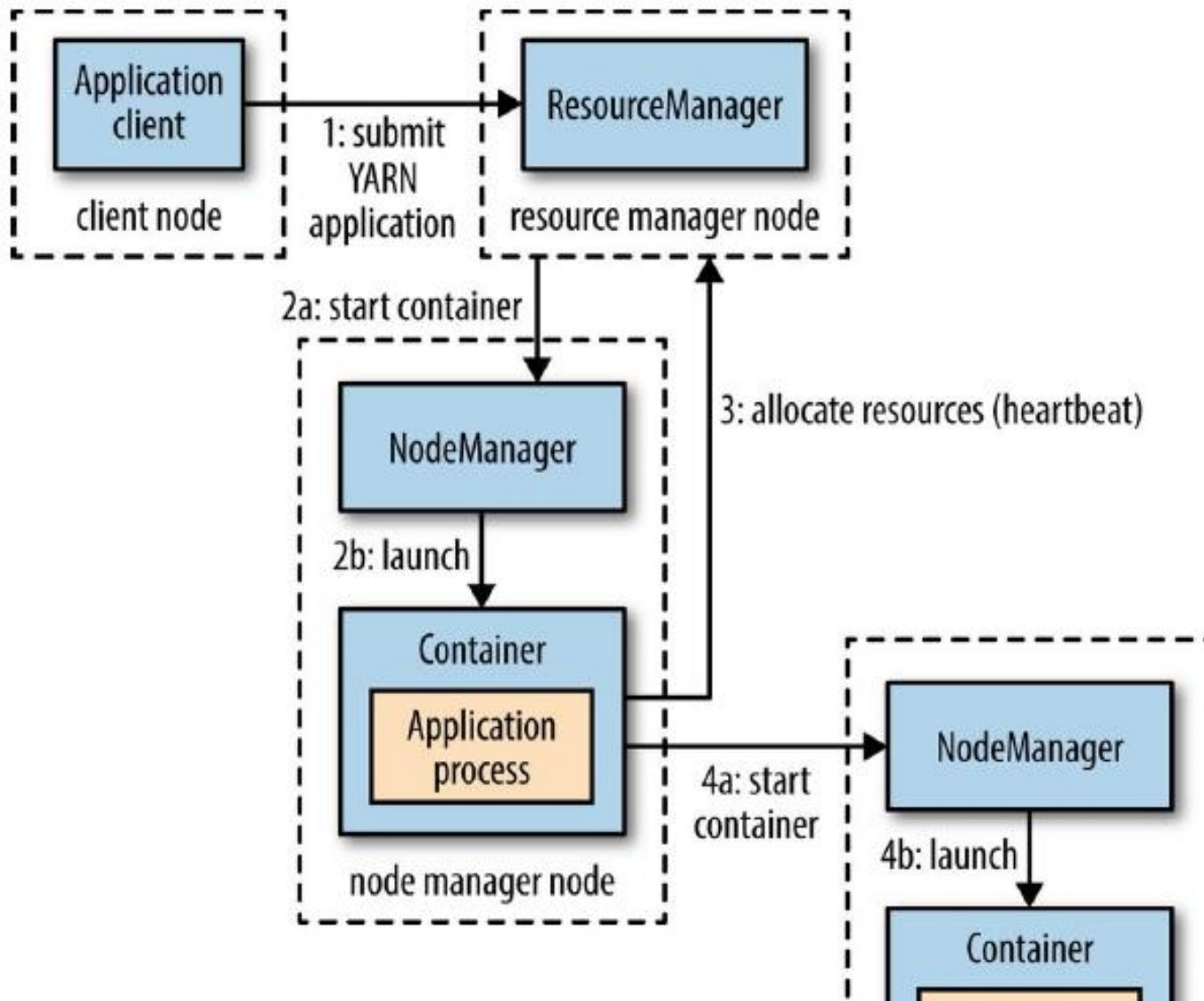
Participants will learn about

- Anatomy of a YARN Application Run
- Scheduling in YARN

YARN Core Concepts

- Resource Manager
- Node Manager
- Application Master
- Container
- Capacity Scheduler
- Fair Scheduler
- FIFO Scheduler

Anatomy of a YARN Application Run



Anatomy of a YARN Application Run

- Resource Request
 - Locality constraint
 - Make all requests at run time (Spark)
 - Make dynamic requests (MapReduce)

Anatomy of a YARN Application Run

- Application Lifespan
 - Short live or long live application
 - One application per one user job
 - One application per workflow (multiple jobs)
 - Long live application shared by different users (Apache Slider and Impala <proxy application, low latency, daemon>)

Anatomy of a YARN Application Run

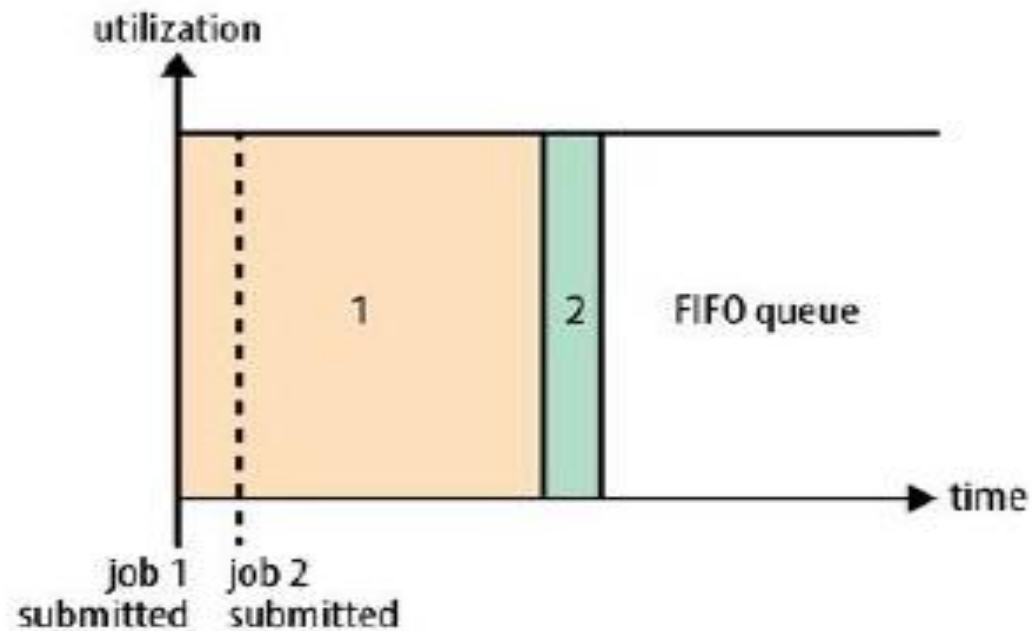
- Building YARN Application
 - Existing YARN application (spark, storm and mapreduce)
 - Apache Slider (run existing distributed application on YARN)
 - Apache Twill
 - Distributed shell application (Example)
 - Long live application shared by different users (Apache Slider and Impala <proxy application, low latency, daemon>)

Scheduling in YARN

- Scheduler Options
 - FIFO
 - Capacity
 - Fair

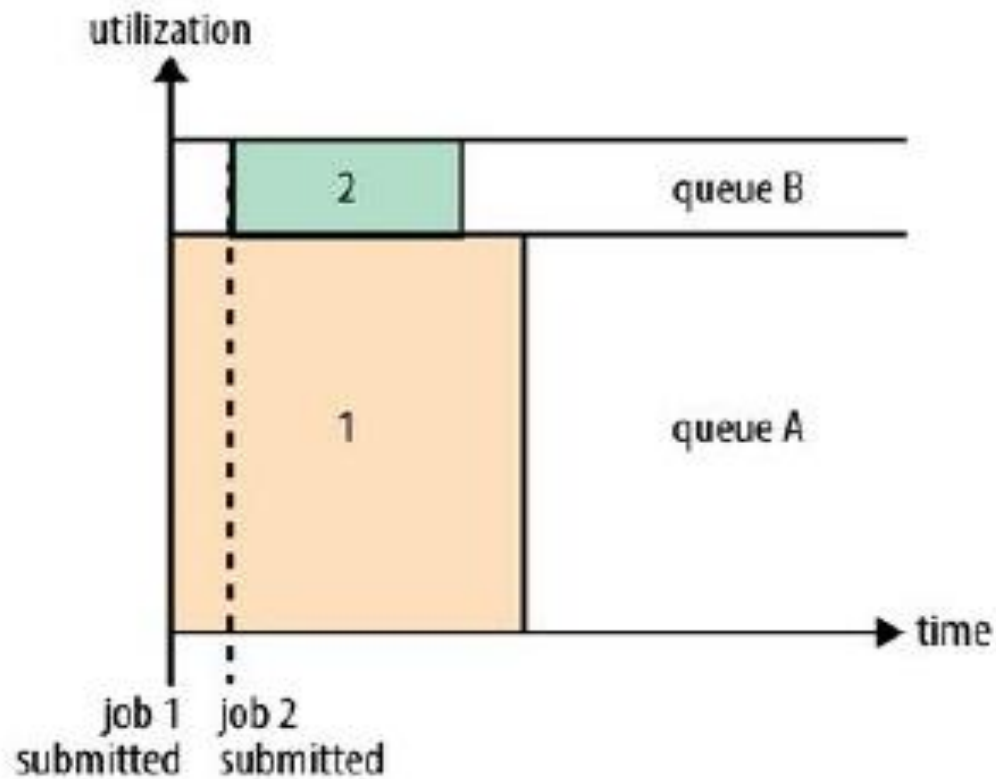
Scheduling in YARN

ii. FIFO Scheduler



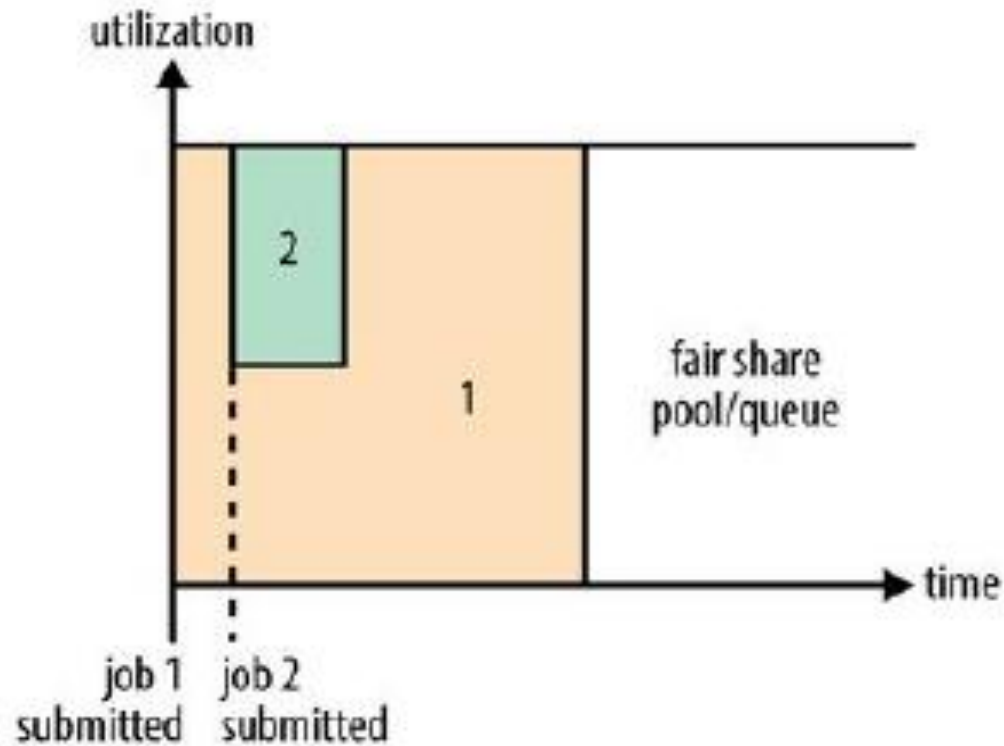
Scheduling in YARN

ii. Capacity Scheduler



Scheduling in YARN

iii. Fair Scheduler



Scheduling in YARN

- Capacity Scheduler Configuration
 - capacity-scheduler.xml
 - yarn.scheduler.capacity.<queue-path>.<sub-property>

```
<configuration>
  <property>
    <name>yarn.scheduler.capacity.root.queues</name>
    <value>prod,dev</value>
  </property>
  <property>
    <name>yarn.scheduler.capacity.root.dev.queues</name>
    <value>eng,science</value>
  </property>
  <property>
    <name>yarn.scheduler.capacity.root.prod.capacity</name>
    <value>40</value>
  </property>
  <property>
    <name>yarn.scheduler.capacity.root.dev.capacity</name>
    <value>60</value>
  </property>
</configuration>
```

Scheduling in YARN

- Fair Scheduler Configuration
 - fair-scheduler.xml
 - yarn.resourcemanager.scheduler.class

```
<allocations>
  <defaultQueueSchedulingPolicy>fair</defaultQueueSchedulingPolicy>

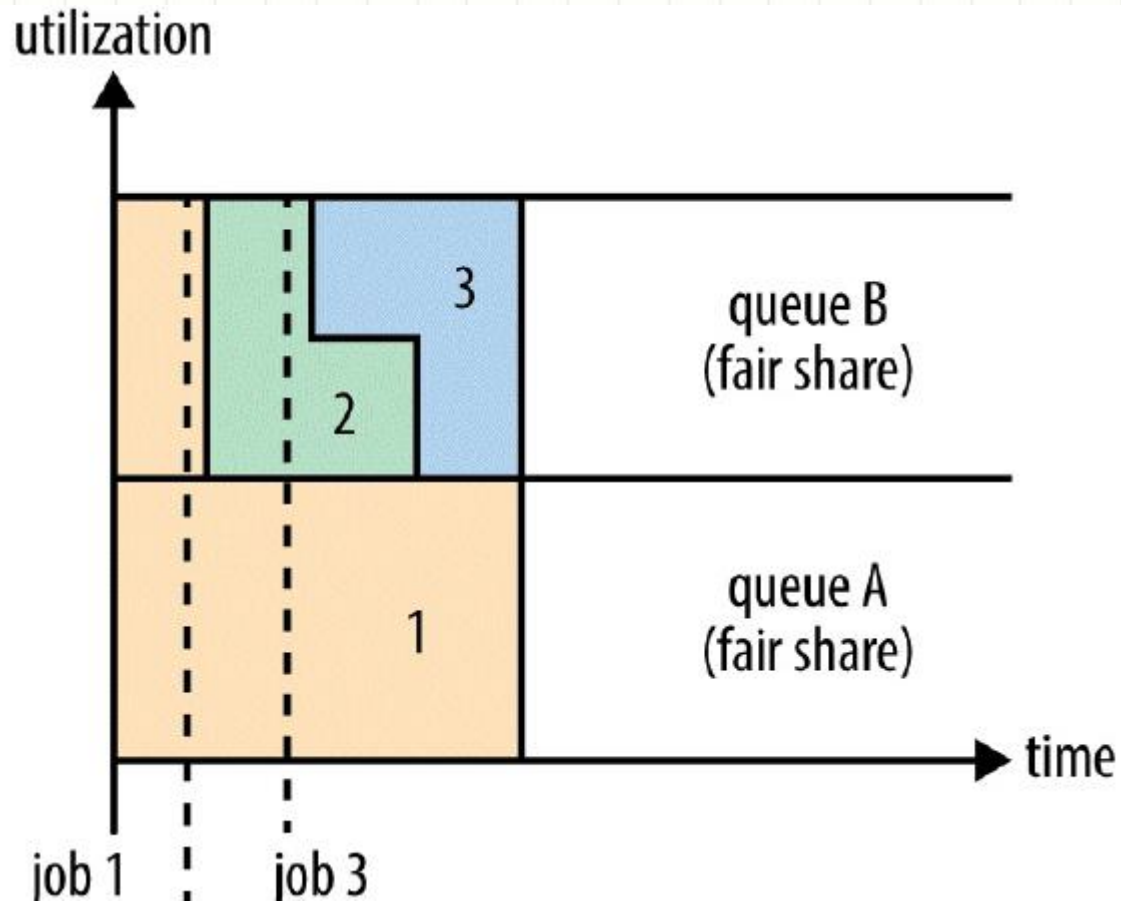
  <queue name="prod">
    <weight>40</weight>
    <schedulingPolicy>fifo</schedulingPolicy>
  </queue>

  <queue name="dev">
    <weight>60</weight>
    <queue name="eng" />
    <queue name="science" />
  </queue>

  <queuePlacementPolicy>
    <rule name="specified" create="false" />
    <rule name="primaryGroup" create="false" />
    <rule name="default" queue="dev.eng" />
  </queuePlacementPolicy>
</allocations>
```

Scheduling in YARN

- Fair Scheduler Configuration



Scheduling in YARN

- Fair Scheduler Configuration
 - fair-scheduler.xml
 - yarn.resourcemanager.scheduler.class

```
<allocations>
  <defaultQueueSchedulingPolicy>fair</defaultQueueSchedulingPolicy>

  <queue name="prod">
    <weight>40</weight>
    <schedulingPolicy>fifo</schedulingPolicy>
  </queue>

  <queue name="dev">
    <weight>60</weight>
    <queue name="eng" />
    <queue name="science" />
  </queue>

  <queuePlacementPolicy>
    <rule name="specified" create="false" />
    <rule name="primaryGroup" create="false" />
    <rule name="default" queue="dev.eng" />
  </queuePlacementPolicy>
</allocations>
```

Scheduling in YARN

- Delay Scheduling
 - Data Locality
 - waiting a short time
 - `yarn.scheduler.capacity.node-locality-delay`