Job Scheduling with the Fair and Capacity Schedulers

Matei Zaharia

Wednesday, June 10, 2009

Santa Clara Marriott

Motivation

- Provide fast response times to small jobs in a shared Hadoop cluster
- Improve utilization and data locality over separate clusters and Hadoop on Demand

Hadoop at Facebook

- 600-node cluster running Hive
- 3200 jobs/day
- 50+ users
- Apps: statistical reports, spam detection, ad optimization, …

Facebook Job Types

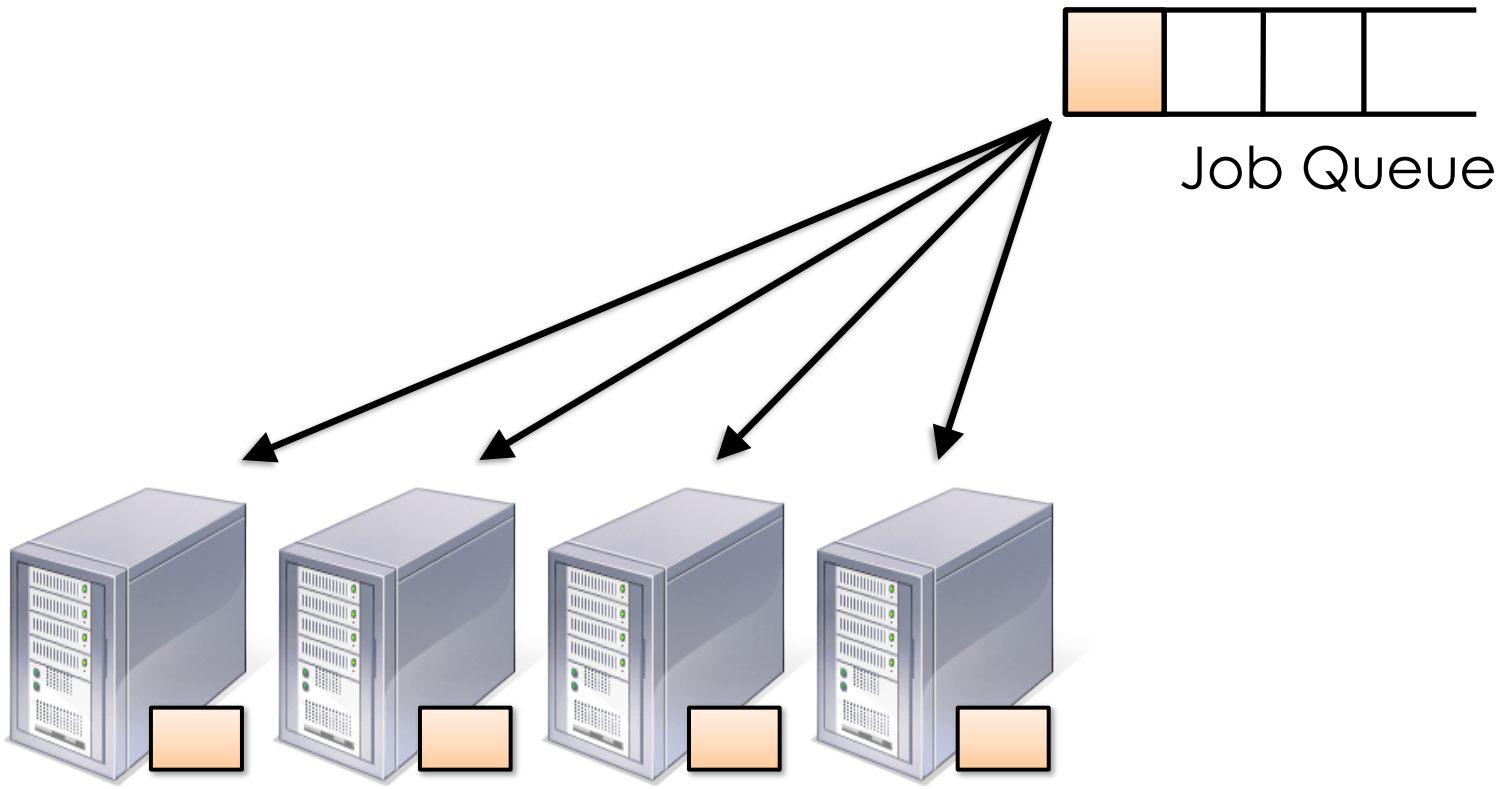
- **Production jobs:** data import, hourly reports, etc
- **Small ad-hoc jobs:** Hive queries, sampling
- **Long experimental jobs:** machine learning, etc

➢ GOAL: fast response times for small jobs, guaranteed service levels for production jobs

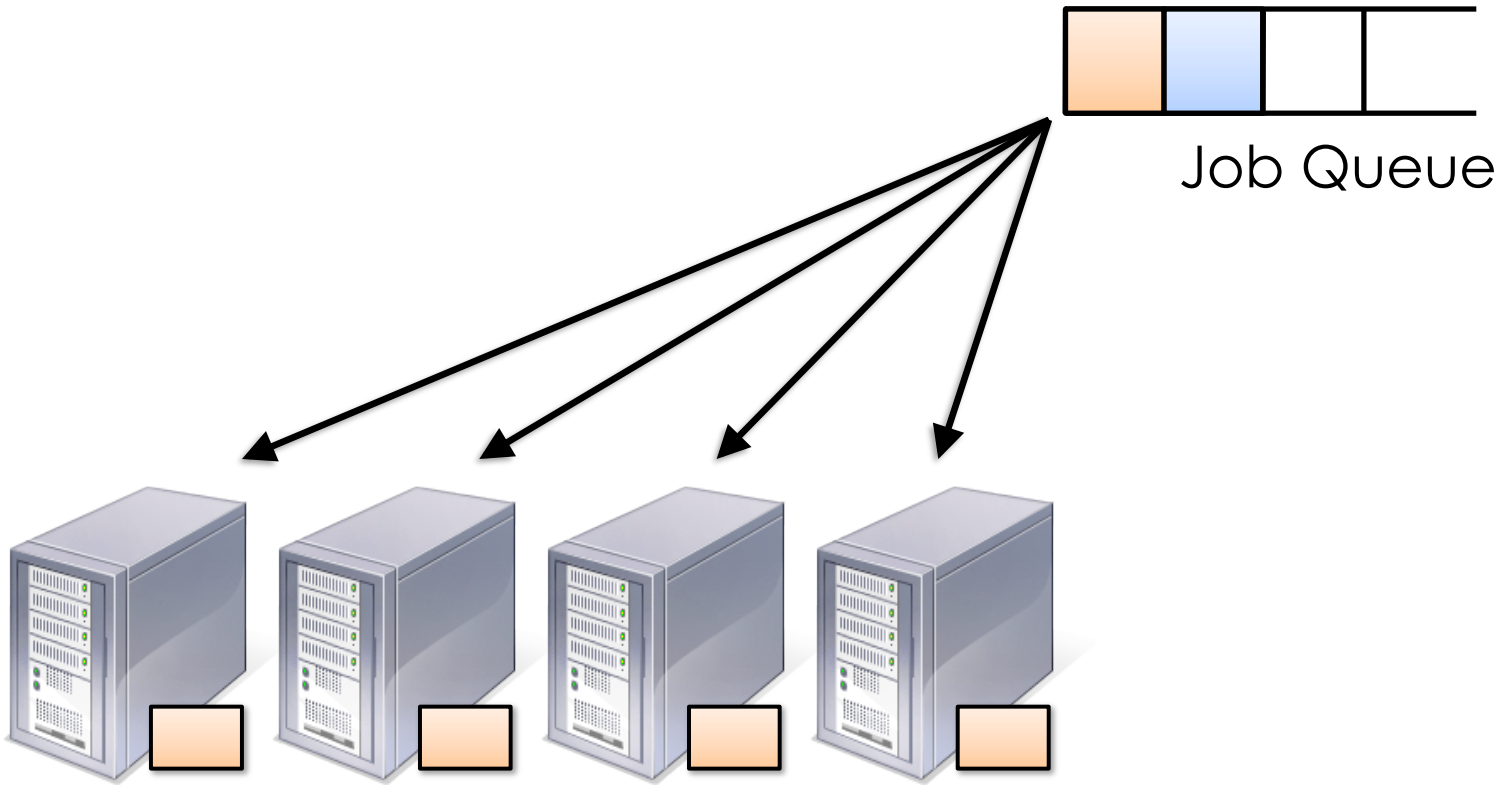
Outline

- Fair scheduler basics
- Configuring the fair scheduler
- Capacity scheduler
- Useful links

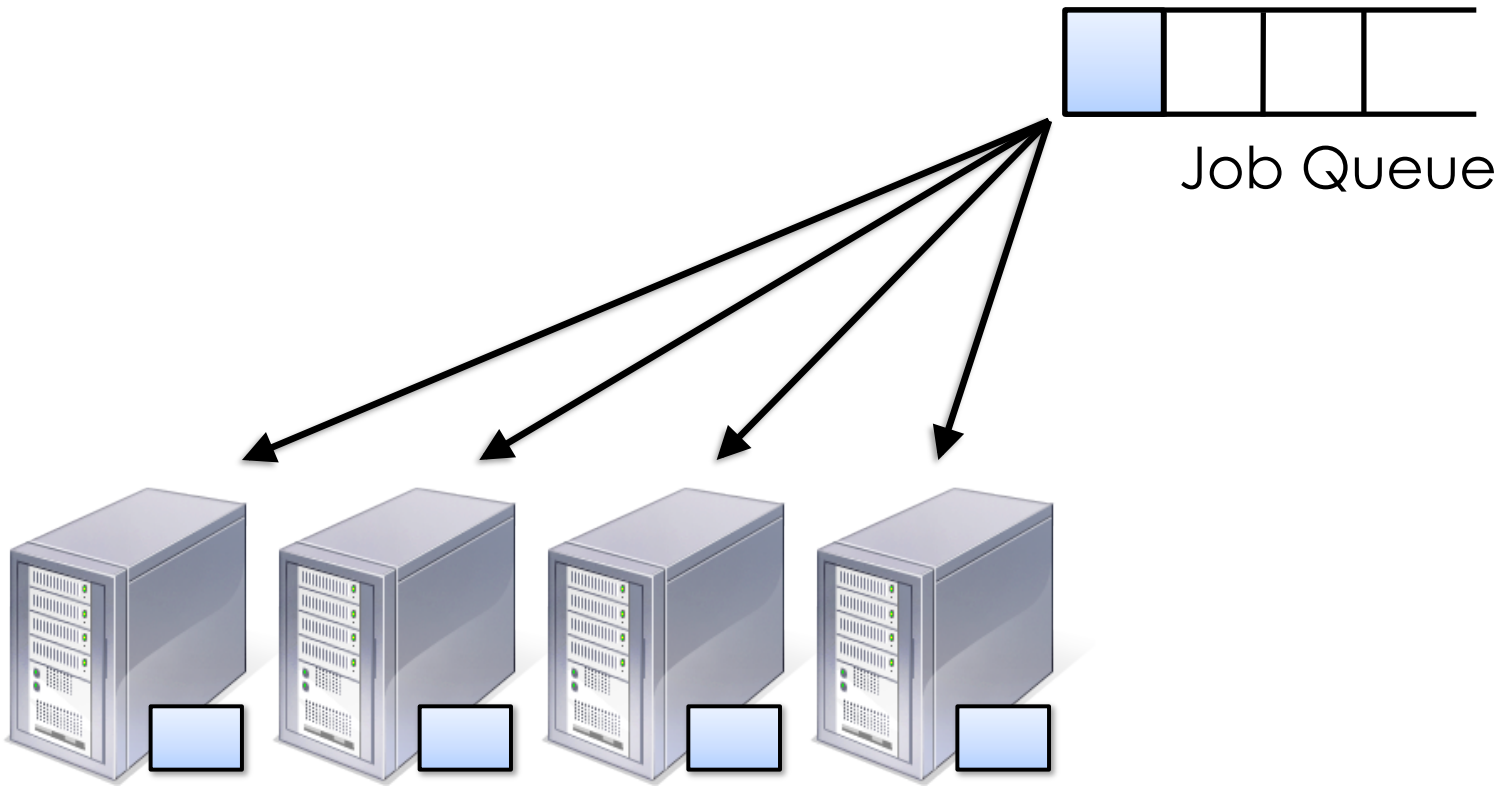
FIFO Scheduling



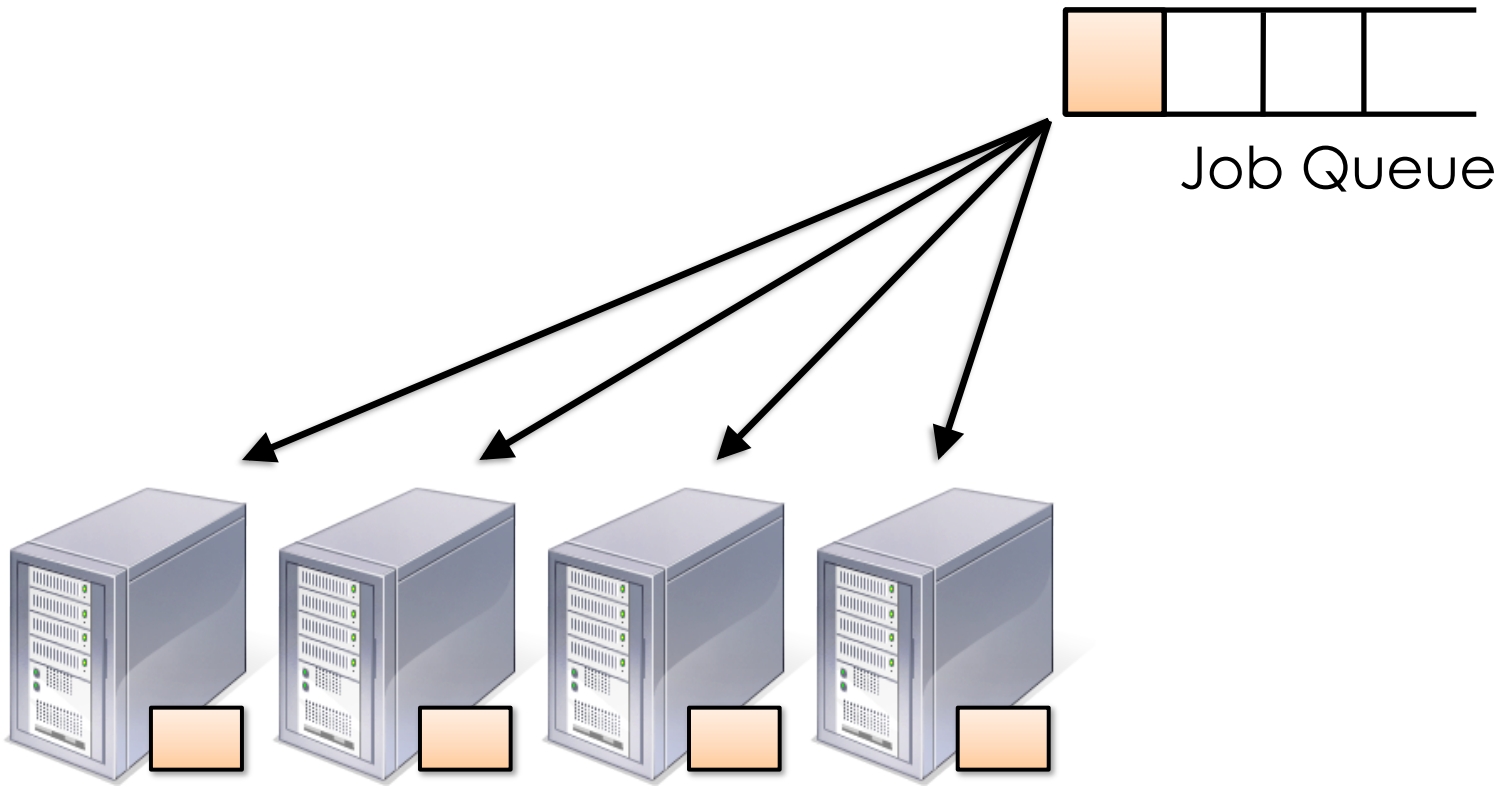
FIFO Scheduling



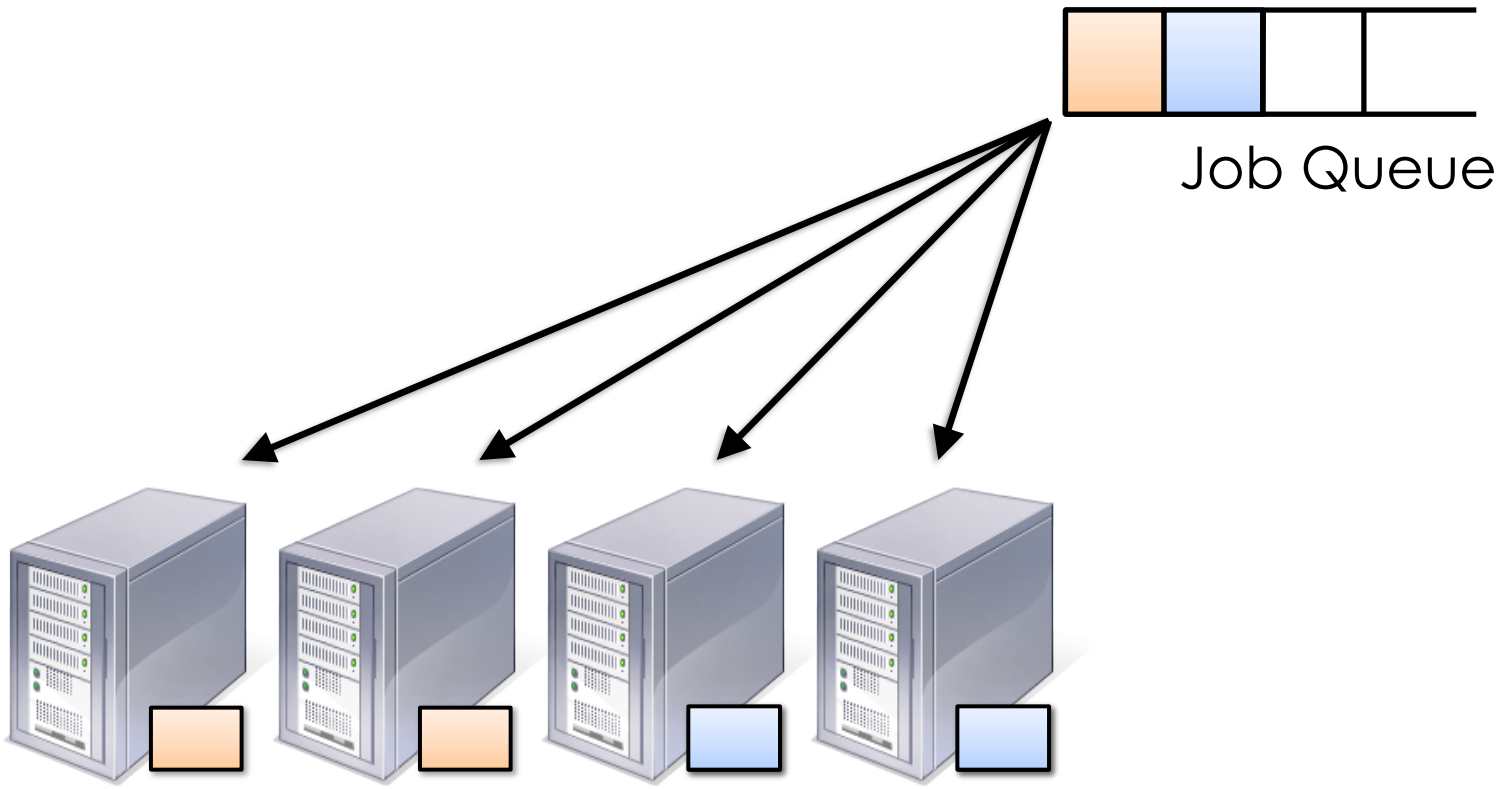
FIFO Scheduling



Fair Scheduling



Fair Scheduling



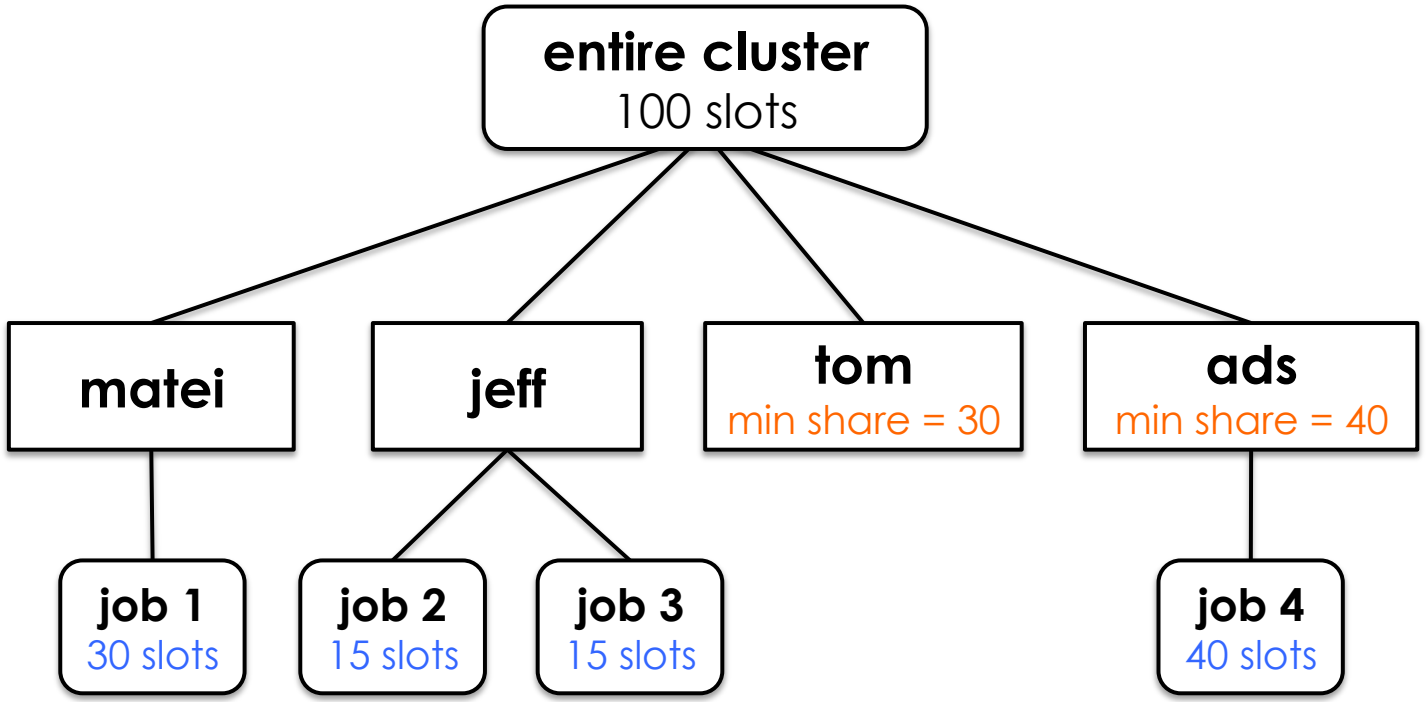
Fair Scheduler Basics

- Group jobs into "*pools*"
- Assign each pool a guaranteed *minimum share*
- Divide excess capacity evenly between pools

Pools

- Determined from a configurable job property
  - Default in 0.20: user.name (one pool per user)
- Pools have properties:
  - Minimum map slots
  - Minimum reduce slots
  - Limit on # of running jobs

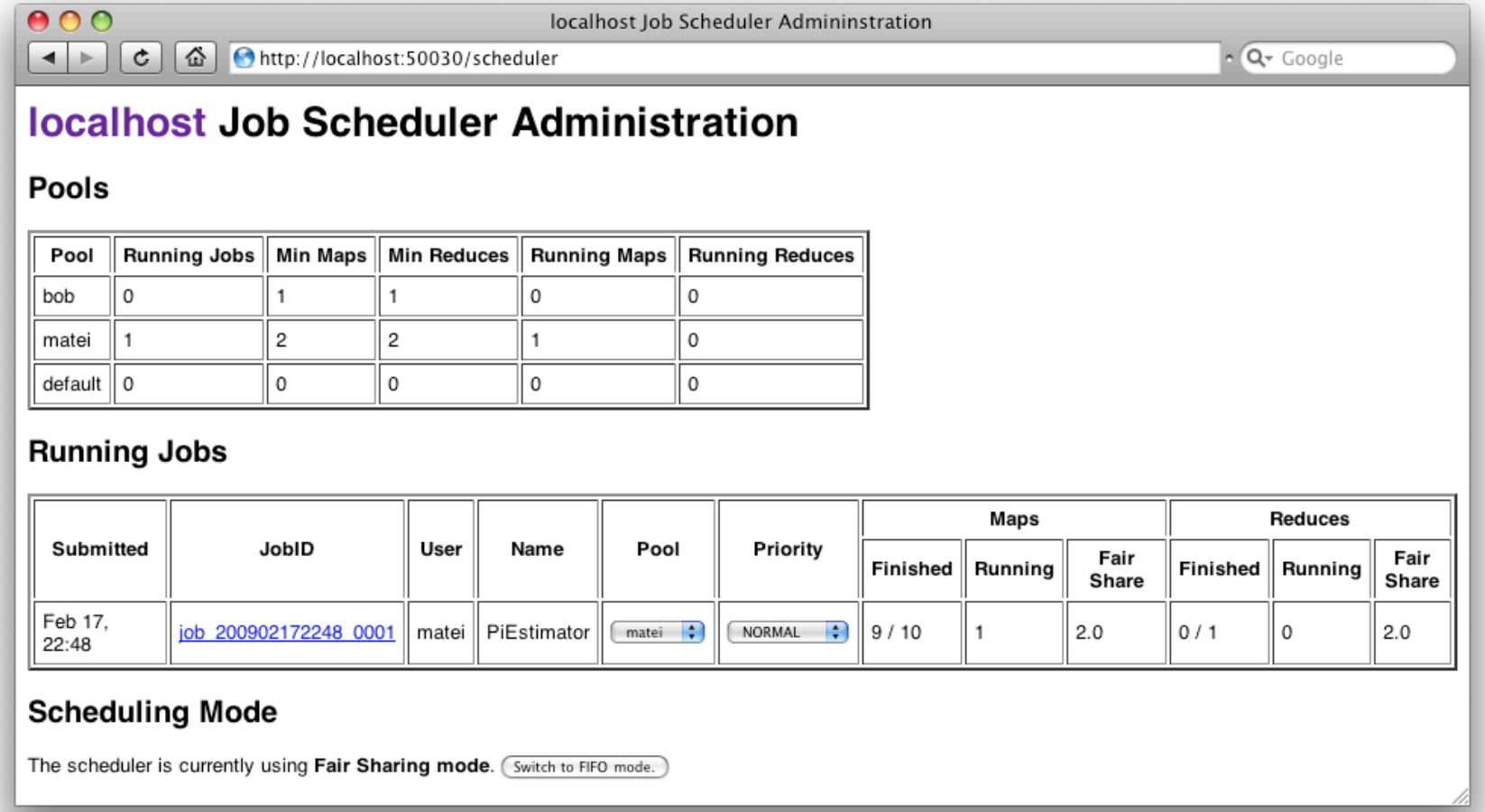
Example Pool Allocations



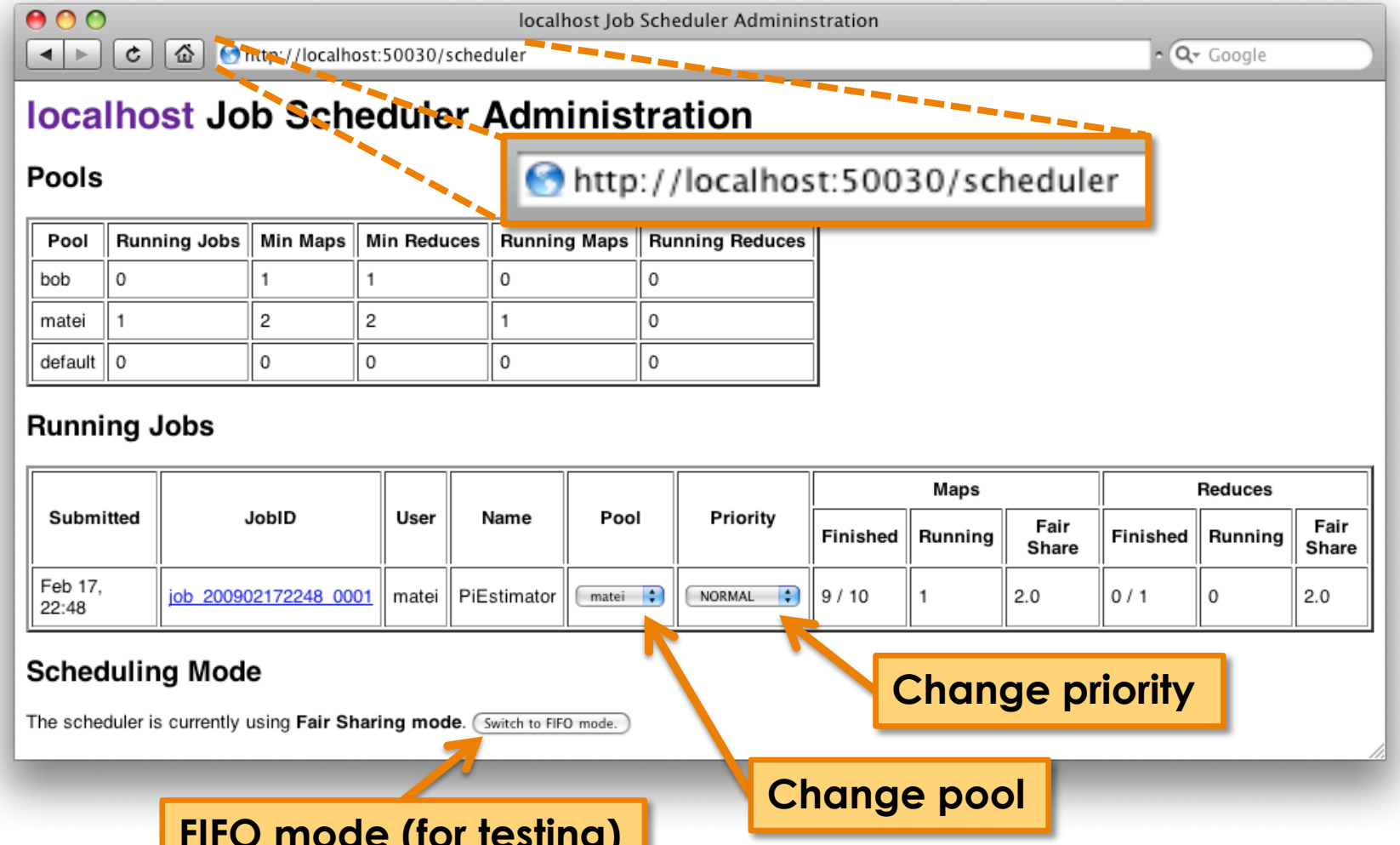
Scheduling Algorithm

- Split each pool's min share among its jobs
- Split each pool's total share among its jobs
- When a slot needs to be assigned:
  - If there is any job below its min share, schedule it
  - Else schedule the job that we've been most unfair to (based on "deficit")

Scheduler Dashboard



Scheduler Dashboard



Additional Features

- Weights for unequal sharing:
  - Job weights based on priority (each level = 2x)
  - Job weights based on size
  - Pool weights

- Limits for # of running jobs:
  - Per user
  - Per pool

Installing the Fair Scheduler

- **Build it:**
  - `ant package`
- **Place it on the classpath:**
  - `cp build/contrib/fairscheduler/*.jar lib`

Configuration Files

- Hadoop config (conf/mapred-site.xml)
  - Contains scheduler options, pointer to pools file
- Pools file (pools.xml)
  - Contains min share allocations and limits on pools
  - Reloaded every 15 seconds at runtime

Minimal hadoop-site.xml

```
<property>
  <name>mapred.jobtracker.taskScheduler</name>
  <value>org.apache.hadoop.mapred.FairScheduler</value>
</property>

<property>
  <name>mapred.fairscheduler.allocation.file</name>
  <value>/path/to/pools.xml</value>
</property>
```

Minimal pools.xml

```
<?xml version="1.0"?>
<allocations>
</allocations>
```

Configuring a Pool

```
<?xml version="1.0"?>
<allocations>
  <pool name="ads">
    <minMaps>10</minMaps>
    <minReduces>5</minReduces>
  </pool>
</allocations>
```

Setting Running Job Limits

```
<?xml version="1.0"?>
<allocations>
  <pool name="ads">
    <minMaps>10</minMaps>
    <minReduces>5</minReduces>
    <maxRunningJobs>3</maxRunningJobs>
  </pool>
  <user name="matei">
    <maxRunningJobs>1</maxRunningJobs>
  </user>
</allocations>
```

Default Per-User Running Job Limit

```
<?xml version="1.0"?>
<allocations>
  <pool name="ads">
    <minMaps>10</minMaps>
    <minReduces>5</minReduces>
    <maxRunningJobs>3</maxRunningJobs>
  </pool>
  <user name="matei">
    <maxRunningJobs>1</maxRunningJobs>
  </user>
  <userMaxJobsDefault>10</userMaxJobsDefault>
</allocations>
```

Other Parameters

mapred.fairscheduler.assignmultiple:

- **Assign a map and a reduce on each heartbeat; improves ramp-up speed and throughput; recommendation: set to true**

Other Parameters

mapred.fairscheduler.poolnameproperty:

- **Which JobConf property sets what pool a job is in**
  - **Default: user.name (one pool per user)**
  - **Can make up your own, e.g. "pool.name", and pass in JobConf with conf.set("pool.name", "mypool")**

Useful Setting

```
<property>
  <name>mapred.fairscheduler.poolnameproperty</name>
  <value>pool.name</value>
</property>

<property>
  <name>pool.name</name>
  <value>${user.name}</value>
</property>
```

Future Plans

- Preemption (killing tasks) if a job is starved of its min or fair share for some time (HADOOP-4665)
- Global scheduling optimization (HADOOP-4667)
- FIFO pools (HADOOP-4803, HADOOP-5186)

Capacity Scheduler

- Organizes jobs into queues
- Queue shares as %'s of cluster
- FIFO scheduling within each queue
- Supports preemption
- http://hadoop.apache.org/core/docs/current/capacity_scheduler.html

Thanks!

- Fair scheduler included in Hadoop 0.19+ and in Cloudera's Distribution for Hadoop
- Fair scheduler for Hadoop 0.17 and 0.18: http://issues.apache.org/jira/browse/HADOOP-3746
- Capacity scheduler included in Hadoop 0.19+
- Docs: http://hadoop.apache.org/core/docs/current
- My email: matei@cloudera.com