ADMINISTRIVIA

- Assignment 1: Due Tonight!
- See project list on Piazza
- Assignment 2, Project groups
Scalable Storage Systems

Datacenter Architecture

Resource Management

Computational Engines

Applications

Machine Learning  SQL  Streaming  Graph

MapReduce  Spark

GFS
How do we share CPU between processes?
CLUSTER SCHEDULING

Mechanism

+ Policy

How

What

Space Sharing

Resource Management

Spark

Maffedra
TARGET ENVIRONMENT

Multiple MapReduce versions

Mix of frameworks: MPI, Spark, MR

Data sharing across frameworks

Avoid per-framework clusters
DESIGN

- Hadoop scheduler
- MPI scheduler
- ZooKeeper quorum

Master: single large
Slave: task

Simplicity across frameworks
"Future proof"

Two-level scheduling

GFS/MapReduce:

- Master to slave

Per framework scheduler
RESOURCE OFFERS
CONSTRANTS

Examples of constraints

Constraints in Mesos:

- Reject offers
- Filters: Boolean clauses

→ Specific hardware (GPUs)
→ Large allocation

Data placement → Locality

→ Long tasks

→ Spark

User

Driver

Framework

Spark

Meso
DESIGN DETAILS

Allocation:
Guaranteed allocation, revocation

Isolation
Containers (Docker)

lightweight → fast to start

2 CPU, 1 GB
4 CPU, 4 GB

limit memory, CPU

min limit of how many tasks long tasks

How much progress
FAULT TOLERANCE

- **Soft state**
  - No checkpoint
  - All DS can be repopulated

- **Mesos slaves**
  - Slaves Mesos & framework schedulers
  - Tasks can keep running even if master is down

- **Mesos master**
  - Components: Hadoop scheduler, MPI scheduler, ZooKeeper quorum
  - Heartbeat failure
  - Gfs Master checkpoint
PLACEMENT PREFERENCES

What is the problem?

→ list of frameworks, who gets the offer

How do we do allocations?

→ weighted lottery

< fraction of allocation

scheduler, weighted

random number

[0.0, 0.5]
[0.5, 0.8]
[0.8, 1.0]
COMPARISON: YARN

Per-job scheduler

AM asks for resource
COMPARISON: BORG

Single centralized scheduler

Requests mem, cpu in cfg
Priority per user / service

Support for quotas / reservations
CENTRALIZED VS DECENTRALIZED

Decentralized

- Simple mechanism
- Future frameworks
- Scalability
- "Short tasks"

Centralized

- Global optimum
- Better packing
- Avoid fragmentation
- Avoid Starvation
CENTRALIZED VS DECENTRALIZED

Framework complexity

Fragmentation, Starvation

Low-priority tasks for utilization?
DISCUSSION

https://forms.gle/oYYdvTAcczamnxvT7
What are some problems that could come up if we scale from 10 frameworks to 1000 frameworks in Mesos?

- framework to Mesos master?
- Observation?
- Fault recovery longer?
- Allocation module slower?
- Executor lifecycle?

1 slow sched among 10000 sched

"Pessimistic sched" exclusive
List any one difference between an OS scheduler and Mesos

**OS**
- Process context, preemption
- Time slicing
- Request → new thread → allocate

**Mesos**
- Dependencies across frameworks?
- Space sharing → killing tasks
- Offers
Next class: Scheduling Policy

Further reading
- https://www.umbrant.com/2015/05/27/mesos-omega-borg-a-survey/
- https://queue.acm.org/detail.cfm?id=3173558

Assignment 1 due tonight!
Assignment 2 out Thu