

CS 744: MESOS

Shivaram Venkataraman

Spring 2024

ADMINISTRIVIA

- Assignment 2: Due tomorrow!
- Project details
 - Create project groups
 - Bid for projects/Propose your own (Piazza, after class)
 - List of project ideas ~20
 - Come up with your own ideas!
 - Submit by Feb 27th Tue at 10pm

Applications

Machine Learning

SQL

Streaming

Graph

Computational Engines

Scalable Storage Systems

Resource Management



Datacenter Architecture





MapReduce

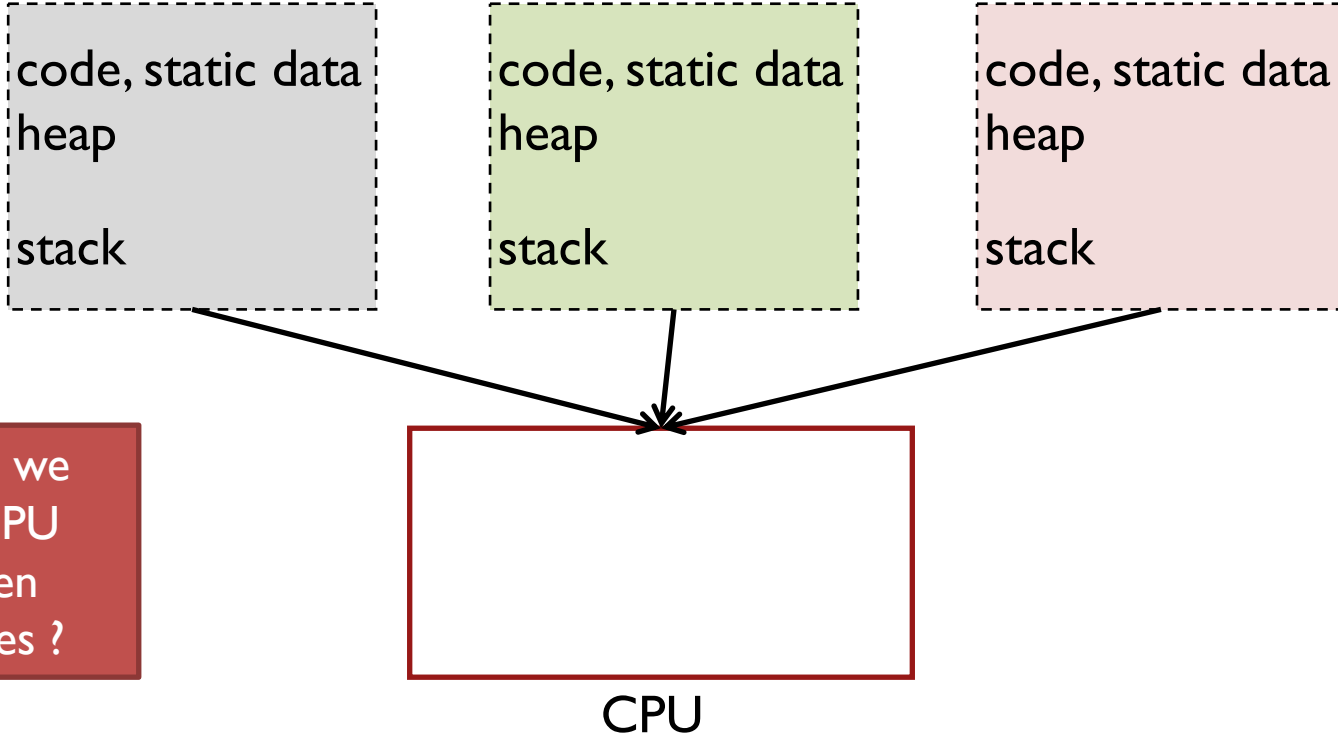
GFS

Spark

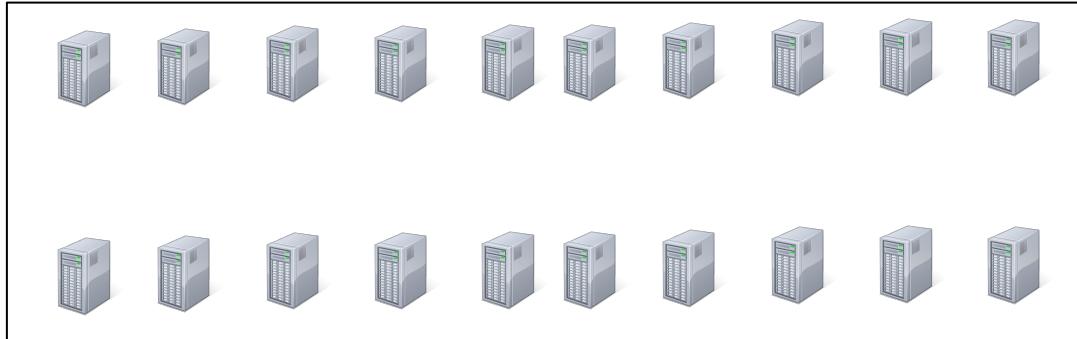
PyTorch

MPI

BACKGROUND: OS SCHEDULING



CLUSTER SCHEDULING



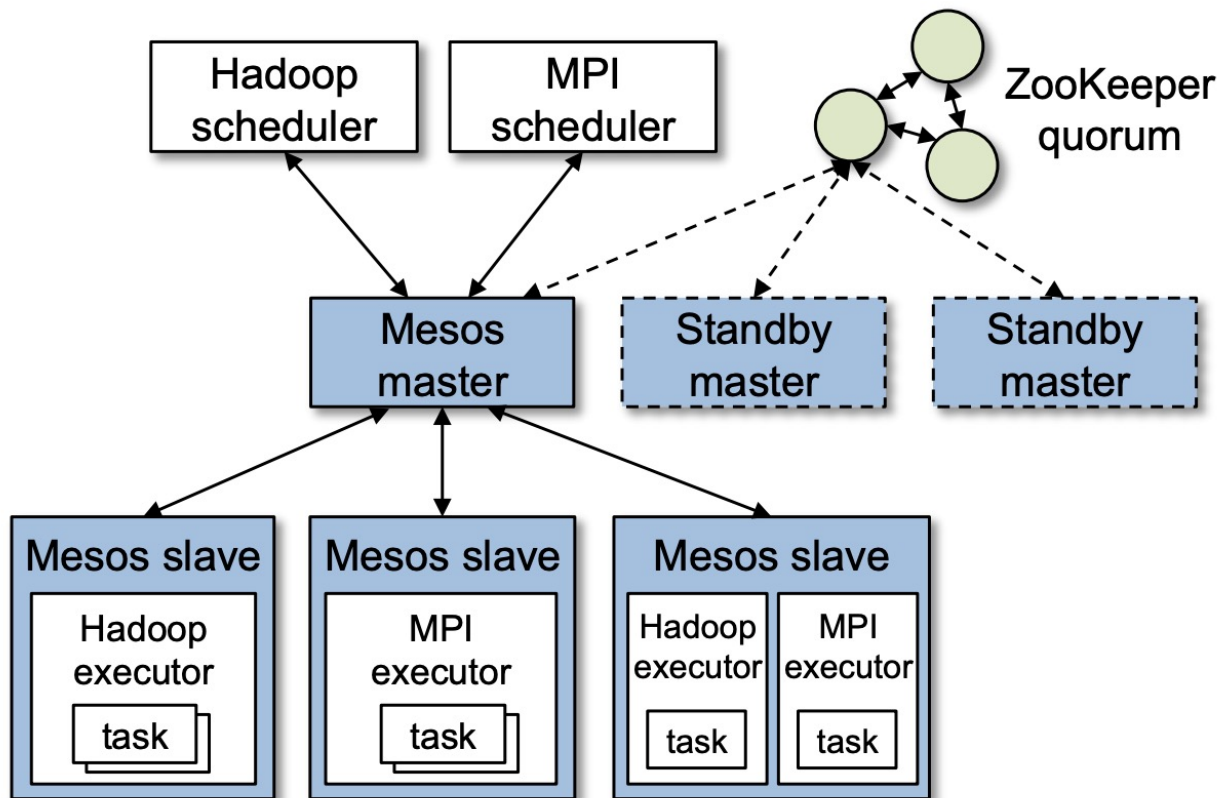
TARGET ENVIRONMENT

Multiple MapReduce versions

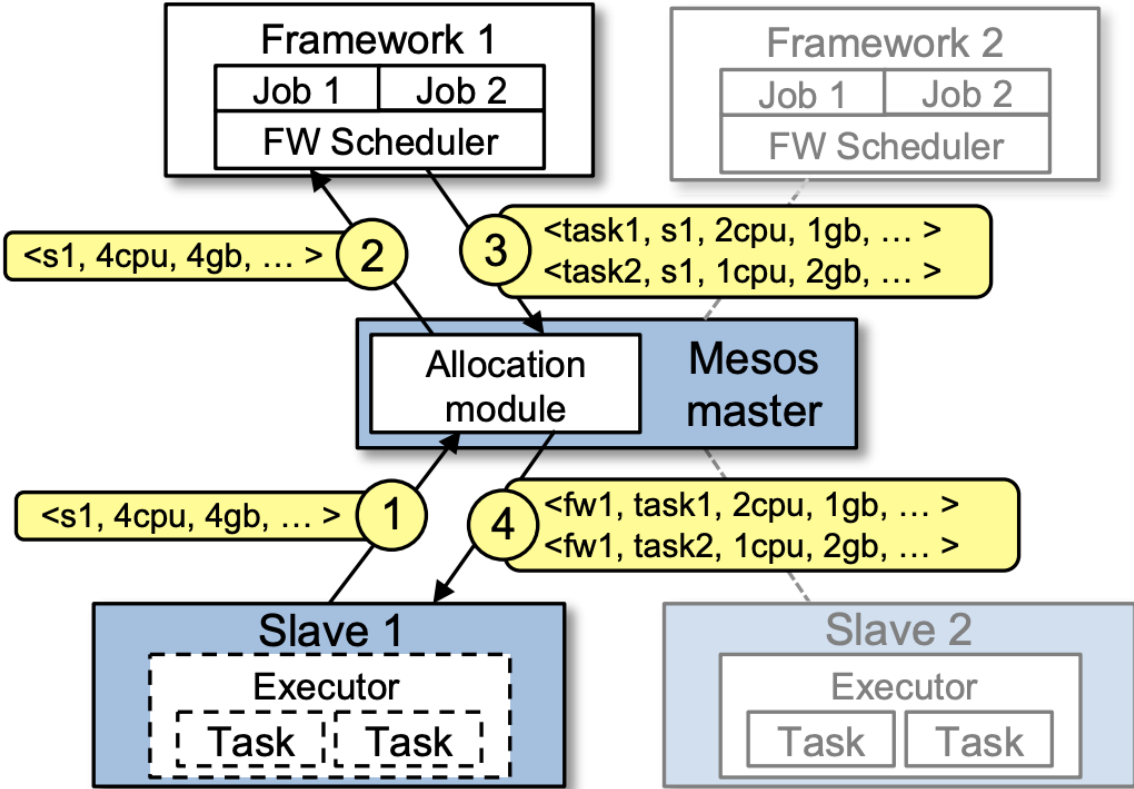
Mix of frameworks: MPI, Spark, MR

Avoid per-framework clusters. Why?

DESIGN



RESOURCE OFFERS



CONSTRAINTS

Examples of constraints

Data locality → soft constraint

GPU machines → hard constraint

Constraints in Mesos:

Applications can reject offers

Optimization: Filters

DESIGN DETAILS

Allocation:

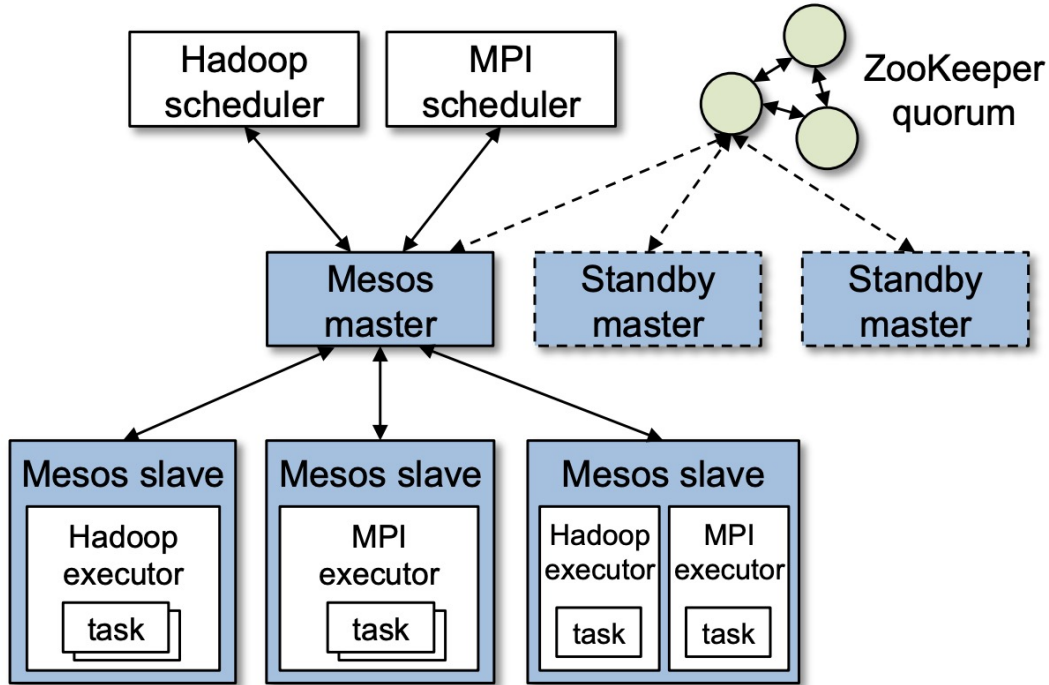
Tasks are short, allocate when they finish

Long tasks? Revocation beyond guaranteed allocation

Isolation

Containers (Docker)

FAULT TOLERANCE



HANDLING PLACEMENT PREFERENCES

What is the problem?

More frameworks have preferred nodes than available

Who gets the offers?

How do we do allocations?

Lottery scheduling – offers weighted by num allocations

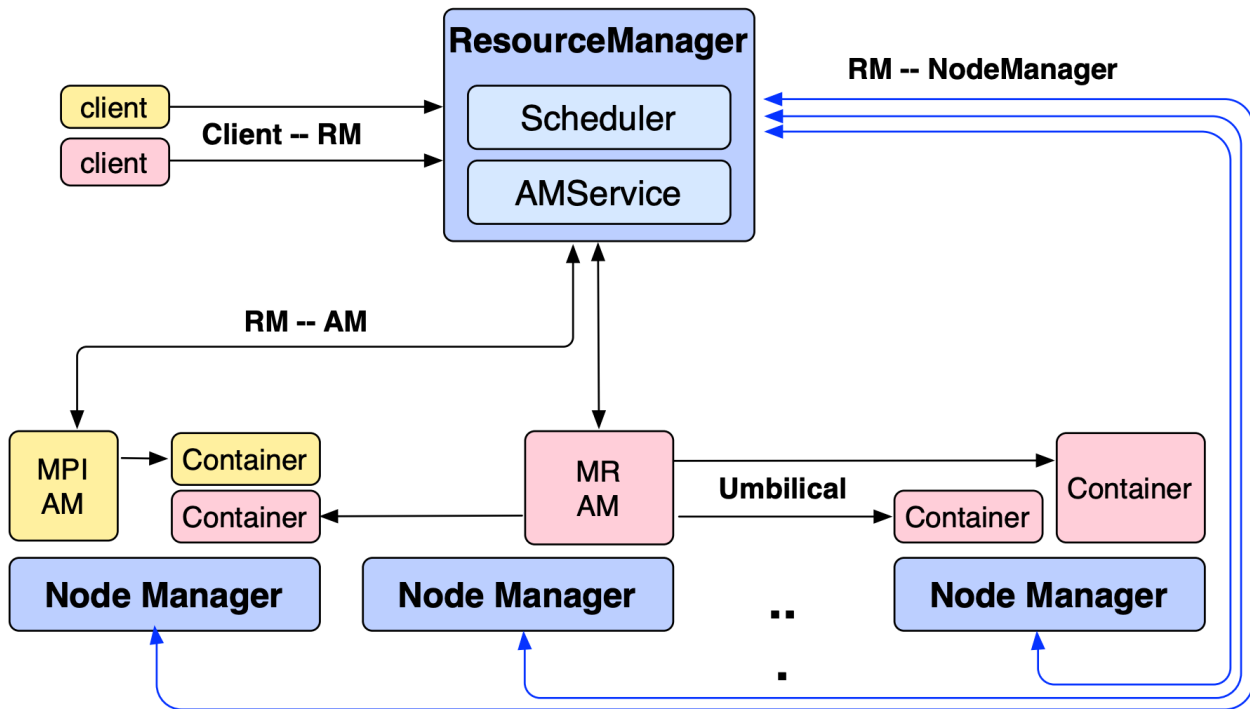
CENTRALIZED VS DISTRIBUTED

Framework complexity

Fragmentation, Starvation

Inter-dependent framework

COMPARISON: YARN



Per-job scheduler

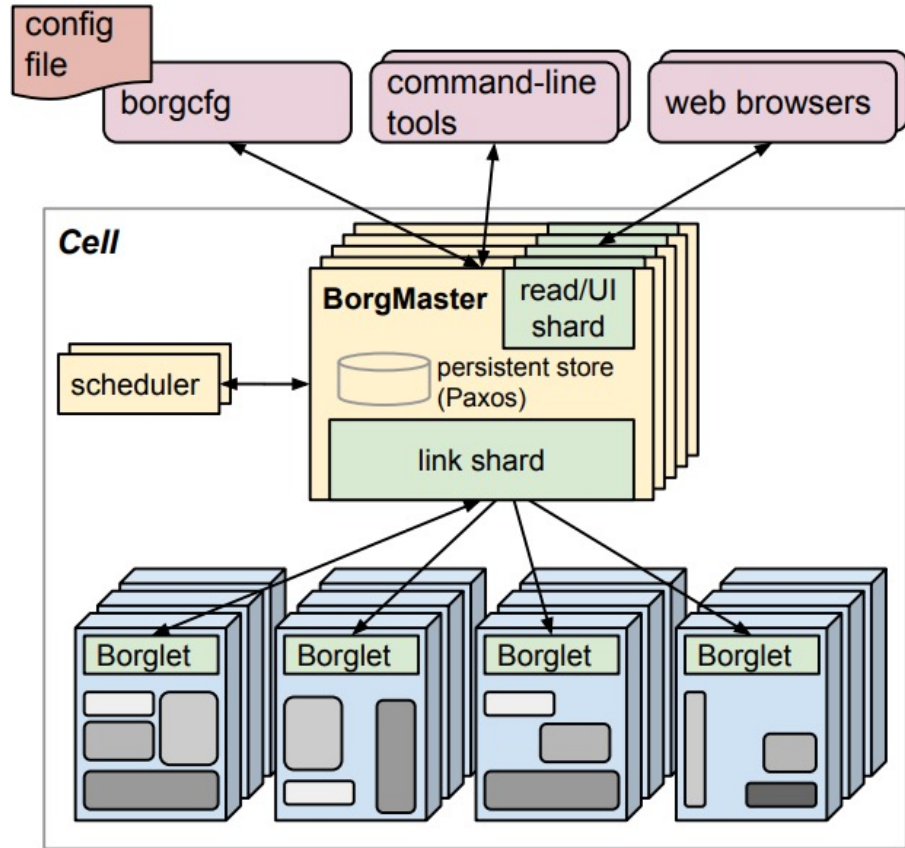
AM asks for resource
RM replies

COMPARISON: BORG (KUBERNETES!?)

Single centralized scheduler

Requests mem, cpu in cfg
Priority per user / service

Support for quotas / reservations



SUMMARY

- Mesos: Scheduler to share cluster between Spark, MR, etc.
- Two-level scheduling with app-specific schedulers
- Provides scalable, decentralized scheduling
- Pluggable Policy ? Next class!

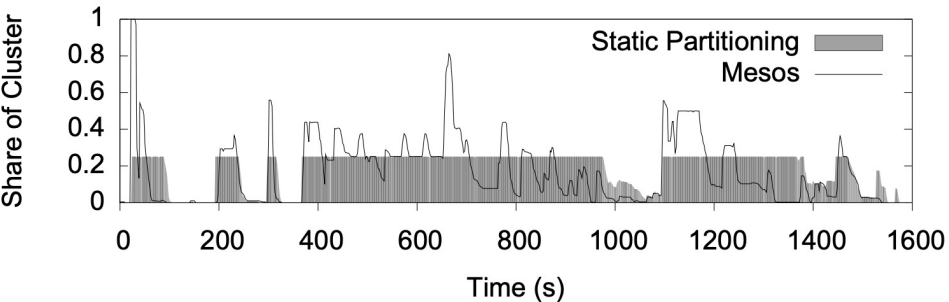


DISCUSSION

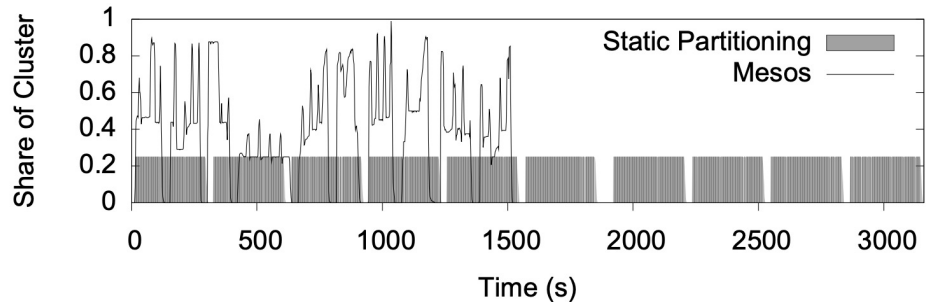
<https://forms.gle/hWQeyW6om3XhnqDS8>

What are some problems that might arise if you wanted to use Mesos with frameworks that had very low latency tasks (e.g., for interactive analytics)

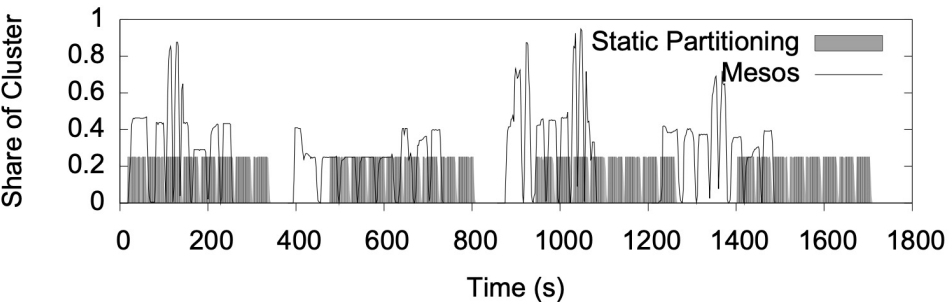
(a) Facebook Hadoop Mix



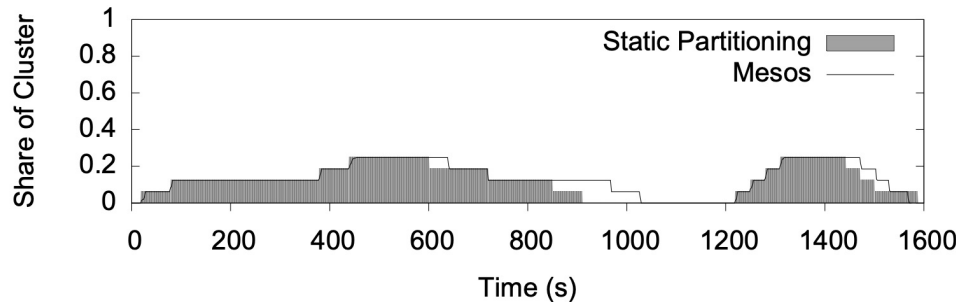
(b) Large Hadoop Mix



(c) Spark



(d) Torque / MPI



NEXT STEPS

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>