Mesos

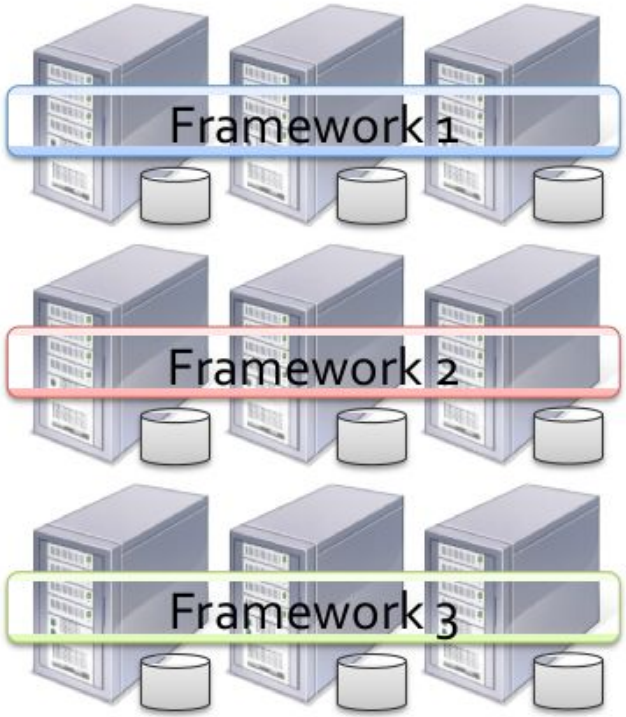
Problem

- Different applications need different frameworks
- How can we share a cluster among multiple frameworks?
  - Statically partitioning the cluster
  - Centralized task scheduler

Key ideas

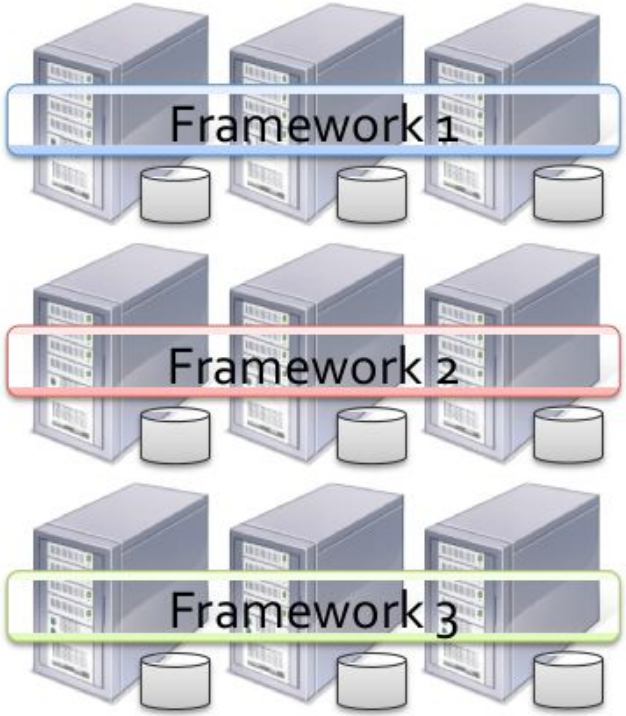
- Fine-grained sharing

Coarse-Grained Sharing (HPC):



Storage System (e.g. HDFS)

Coarse-Grained Sharing (HPC):



Storage System (e.g. HDFS)

Fine-Grained Sharing (Mesos):

Storage System (e.g. HDFS)

Key ideas

- Fine-grained sharing
- Decentralized scheduling
  - Mesos decides *resource offers*
  - Frameworks can *reject*

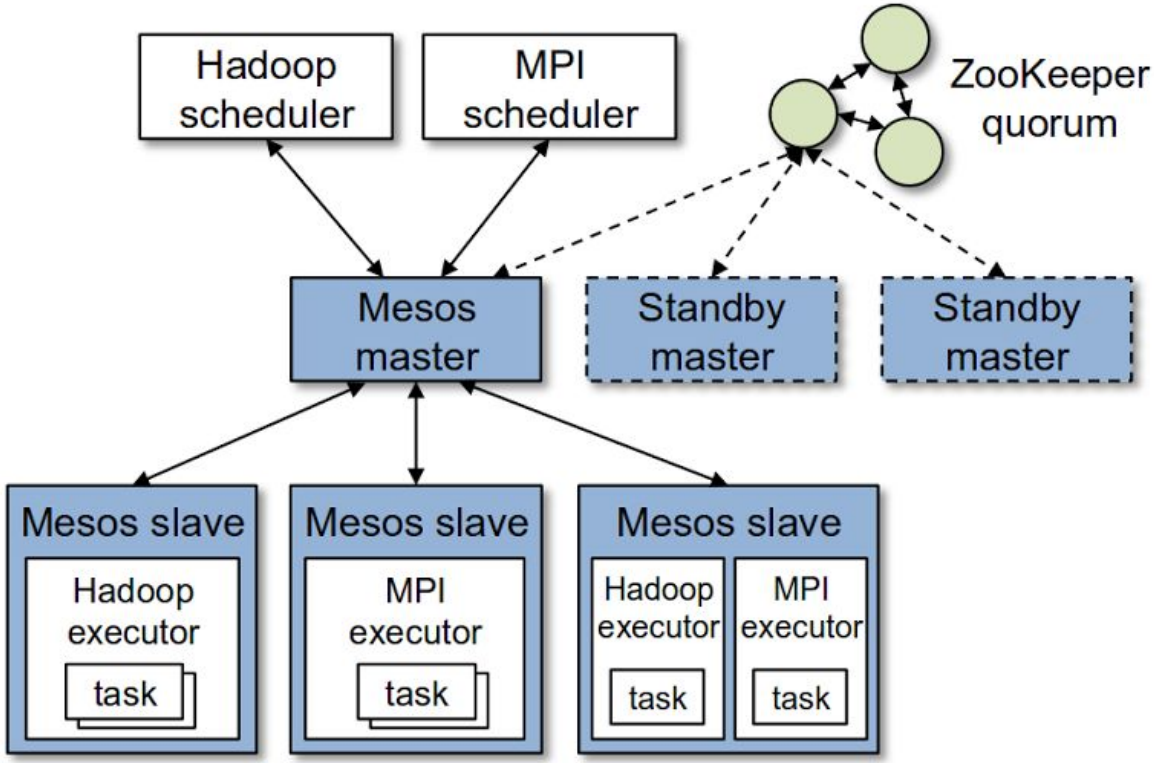


Figure 2: Mesos architecture diagram, showing two running frameworks (Hadoop and MPI).

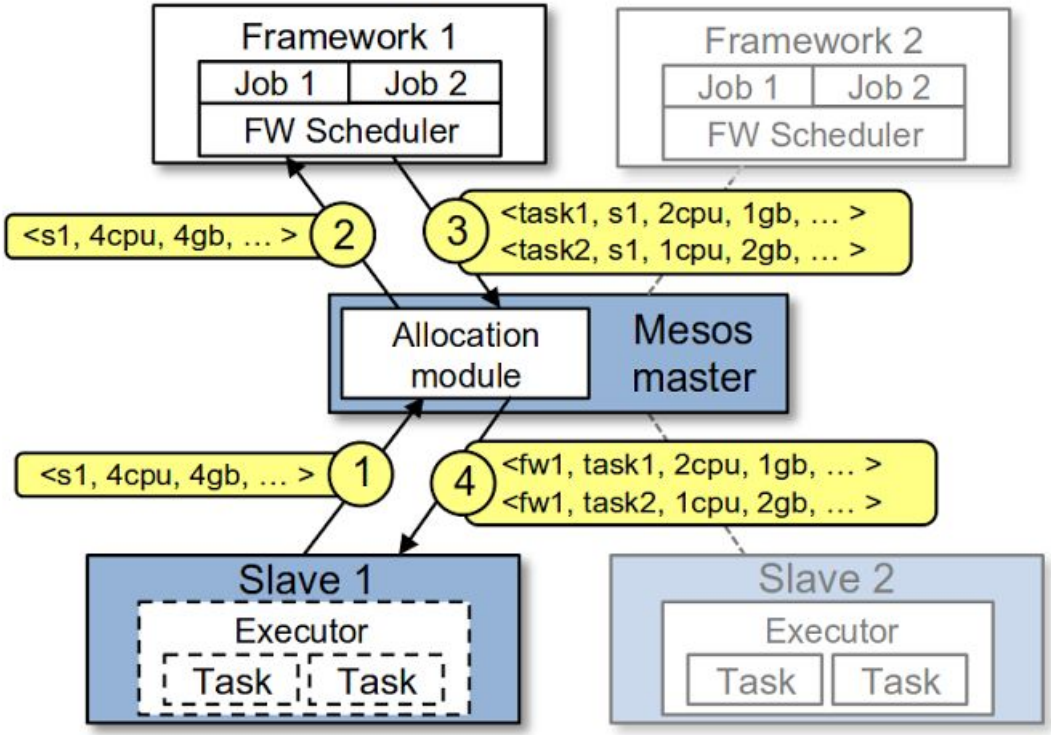


Figure 3: Resource offer example.

Optimizations

- Frameworks can set resource filters
- Master can revoke tasks
  - Master can set guaranteed allocation
  - Frameworks can call `setNeedsOffers(bool)`

Frameworks should behave

- Resources offered count as resources allocated
- Mesos can rescind offers after a timeout
- Short tasks
- Elastic scaling

Scalability and fault tolerance

- Master has soft state
  - Active slaves, active frameworks, running tasks
- Multiple masters with leader election
- Frameworks deal with own failures

Use case: Best with...

- Elastic frameworks
- Homogeneous task durations
- Frameworks that prefer nodes equally

Use case: Frameworks prefer nodes

- If each framework can get preferred slots, they will
- Else, lottery scheduling
  - Frameworks will probably get proportionate numbers of preferred slots

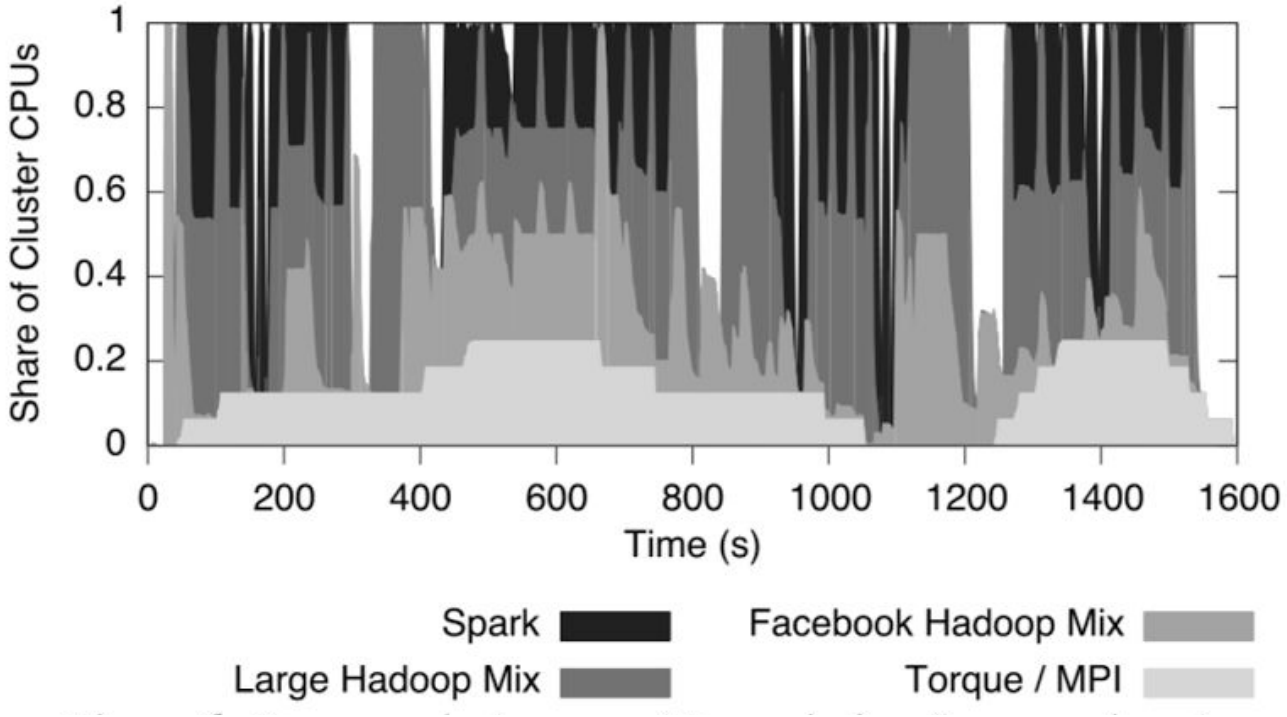
- Delay scheduling → data locality

Use case: Heterogeneous task durations

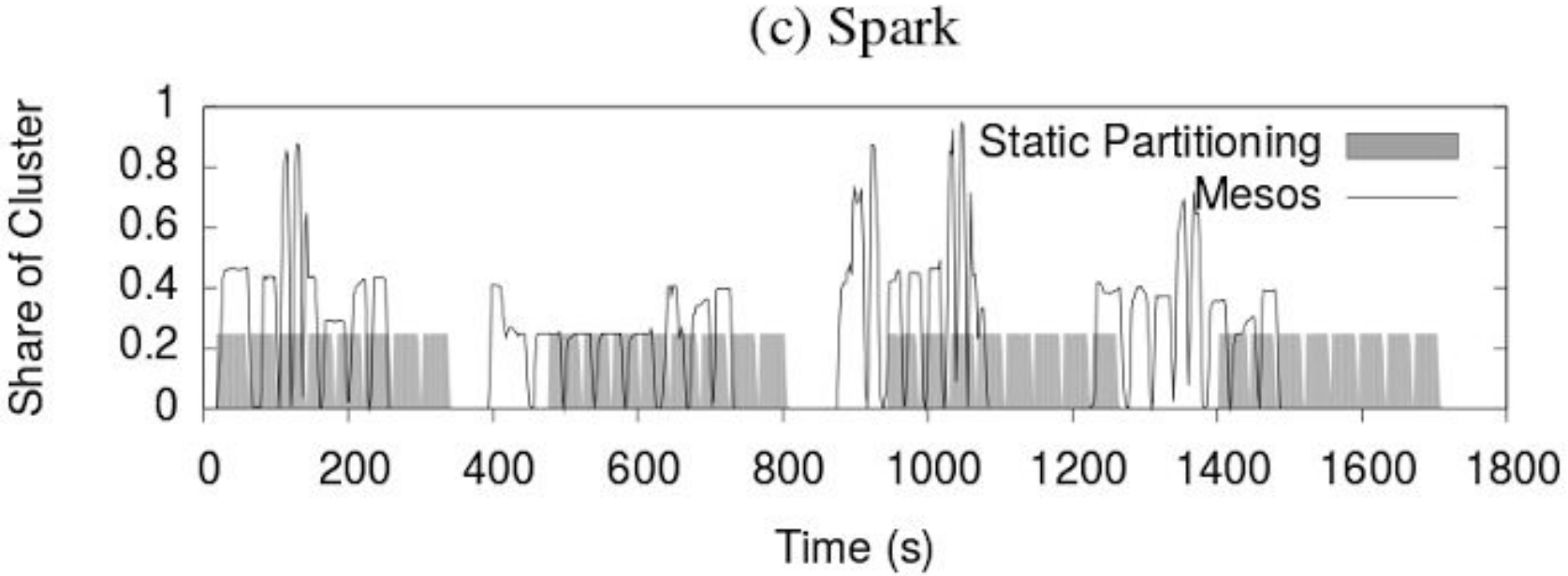
- Okay when there are many slots or not many long tasks
- Master can reserve space for short tasks
- Master can set minimum offer size for long tasks

Limitations

- Fragmentation (bounded)
- Framework interdependence
- Framework schedulers required to use resource offers



Jobs have higher utilization than static partitioning



Jobs finish at least as fast as in static partitioning

Questions?