

# CS 744: TWINE

Shivaram Venkataraman

Spring 2025

# ADMINISTRIVIA

- Assignment grading
- Project groups
- Next week plan

# Applications

Machine Learning

SQL

Streaming

Graph

Computational Engines

Scalable Storage Systems

Resource Management



Datacenter Architecture





MapReduce

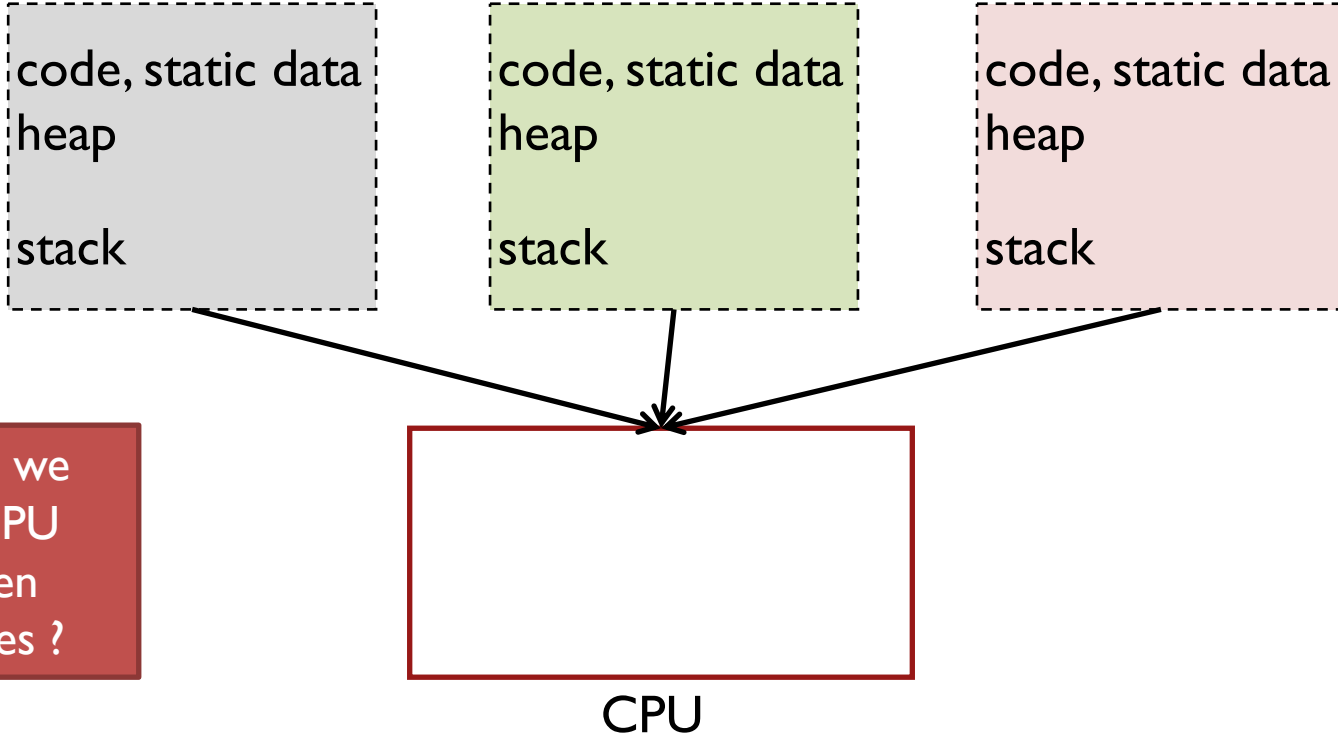
GFS

Spark

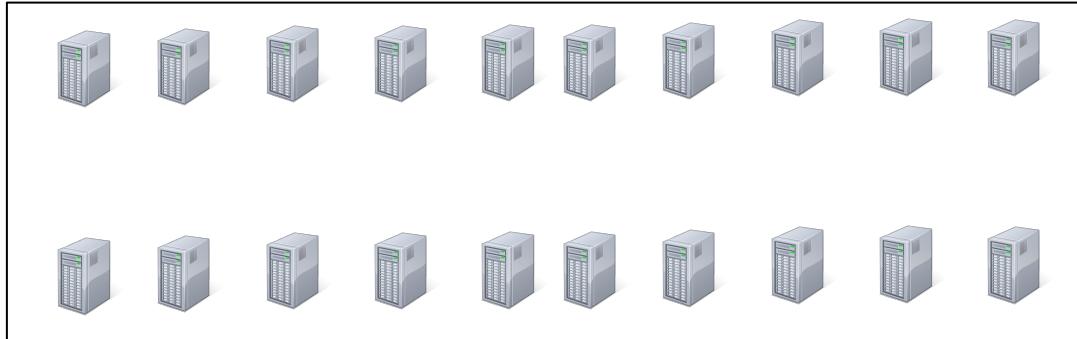
PyTorch

MPI

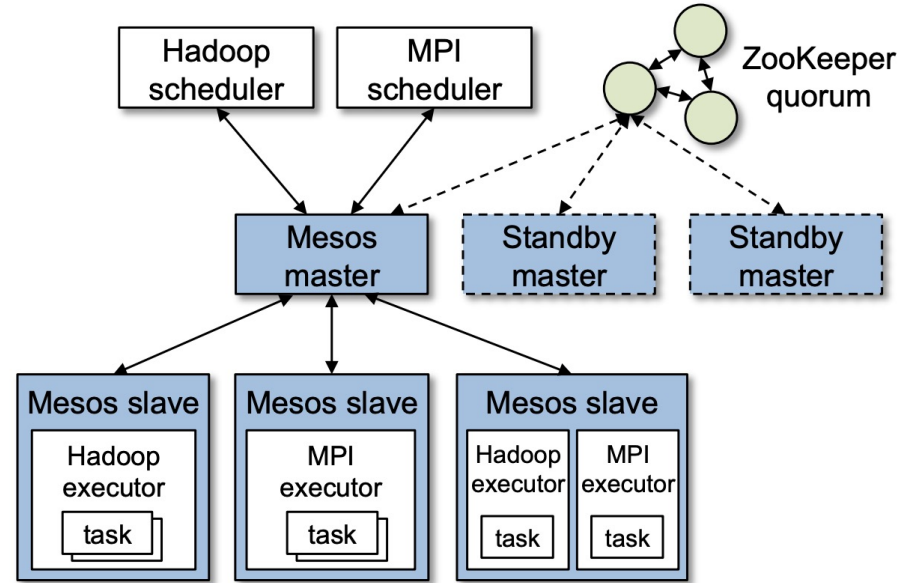
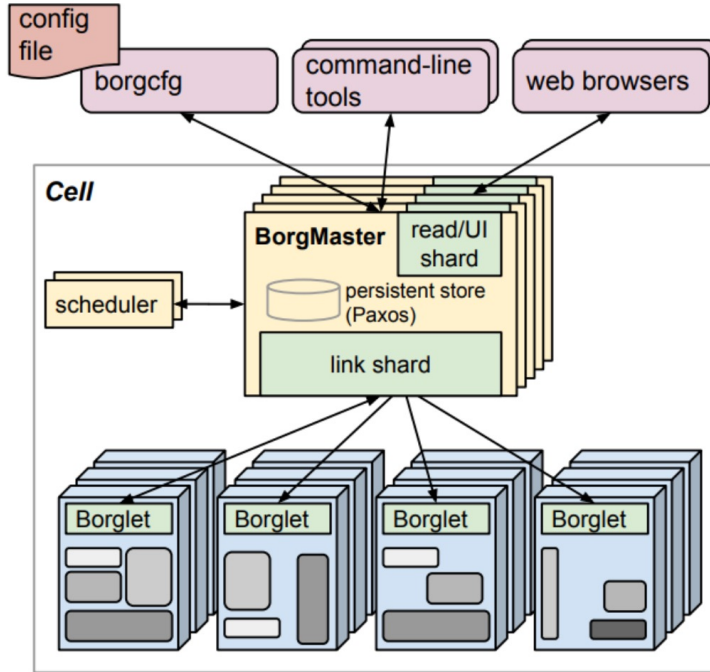
# BACKGROUND: OS SCHEDULING



# CLUSTER SCHEDULING



# BORG, MESOS ETC



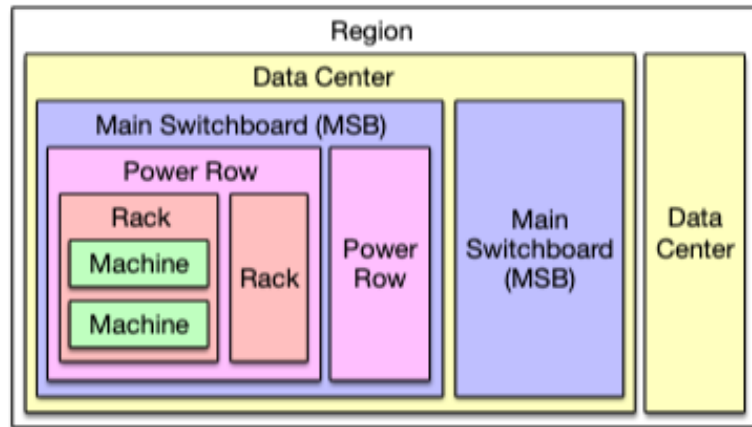
# TWINE

Goals:

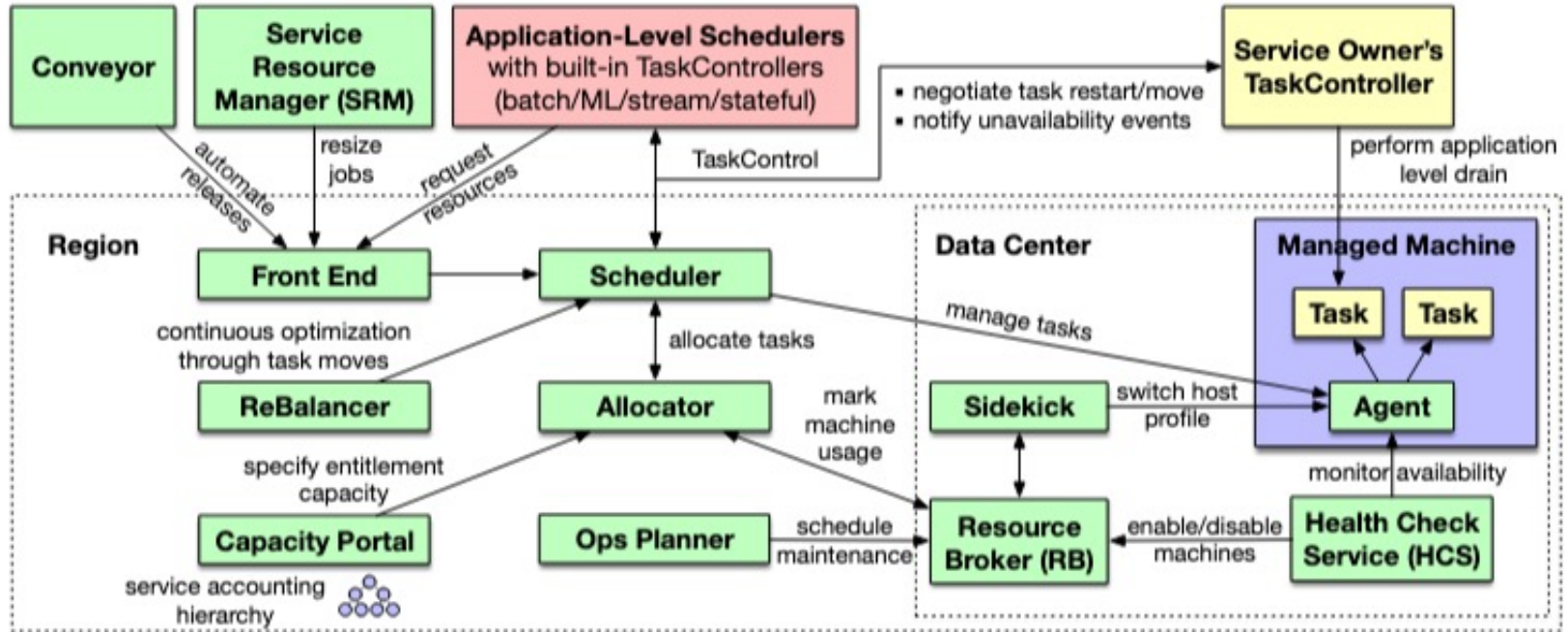
Share resources across clusters

Improve scheduler → application lifecycle

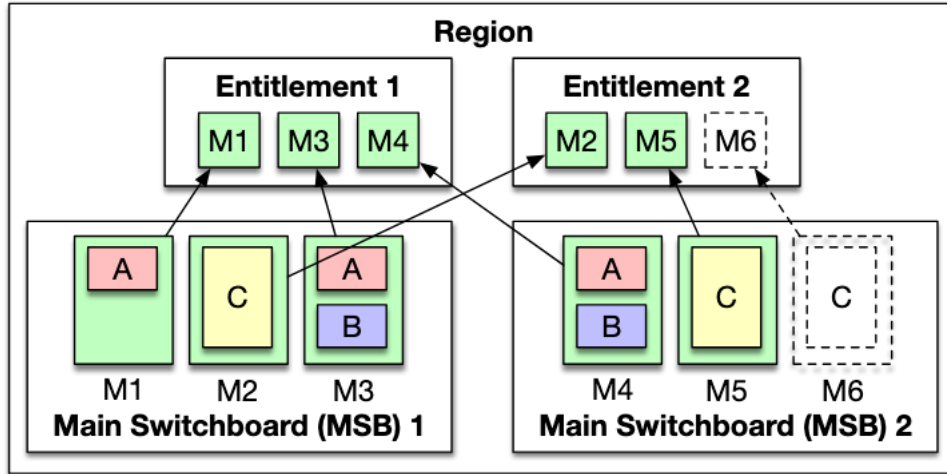
Allow hardware customization



# TWINE: ARCHITECTURE



# ALLOCATOR



In-memory index of all machines

Optimistic concurrency control to resolve conflicts

# TASK CONTROL, APP LEVEL SCHEDULERS

Application knows how to handle lifecycle (Mesos)

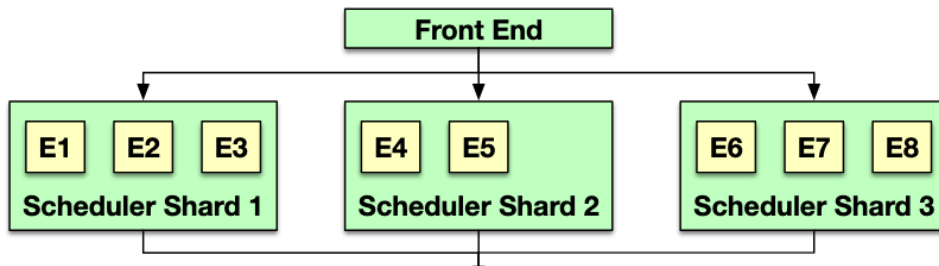
Containers managed by  
application-level schedulers

```
service TaskController {
    TaskControlResponse process(TaskControlRequest request);
}
struct TaskControlRequest {
    string jobHandle;
    list<> request; // Pending task operations to be approved.
    list<> completed; // Completed task operations.
    list<> advanceNotices; // Upcoming planned maintenance.
    list<> allUnhealthyTasks; // Tasks unhealthy due to any reason.
    int sequenceNumber; // Increase after each call.
}
struct TaskControlResponse {
    list<> ack; // Approved task operations.
}
```

# SCALING TO 1M MACHINES

Sharding vs. federation

Shard frontend, scheduler,  
allocator, resource broker...



Use separate persistent external stores!

Move machines vs. move jobs

Separation of concerns

App level scheduler (outside Twine)

# FAULT TOLERANCE

Risks? Failures, network partitions

Shard, replicate components

Scheduler failure doesn't affect running tasks

Network redundancy

Twine manages itself?

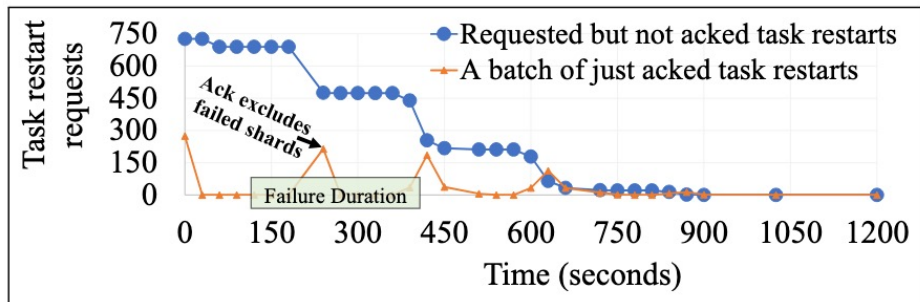
# SUMMARY

- Twine: Scheduler across a region (multiple datacenters)
- Shard all components to ensure scalability
- App-level schedulers, rebalancer for expensive decisions

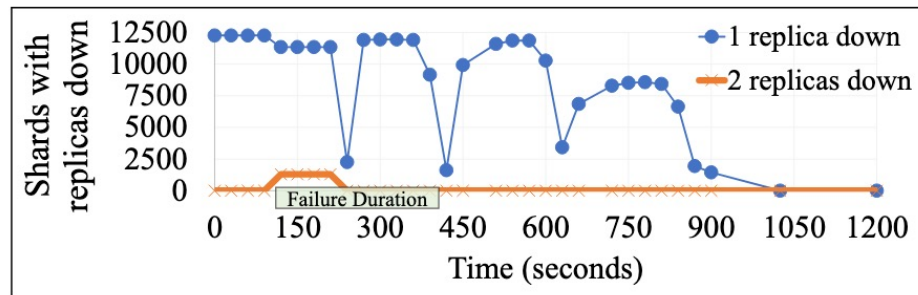


# DISCUSSION

<https://forms.gle/TzaxuHu3kvVM673M9>



(a) Acknowledged and pending task-restart requests.



(b) Shards with some replicas down.

What are some features you might implement if you were writing an application level scheduler for ML training jobs?

# NEXT STEPS

Next class: ML Scheduling with Blox