Mesos:
The Datacenter Operating System

David Greenberg

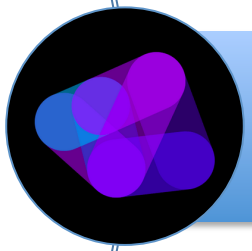
Two Sigma

Who am I?

- Architected project to build a massive Mesos cluster
- Building custom framework and leveraging open source

The Plan

What is Mesos?



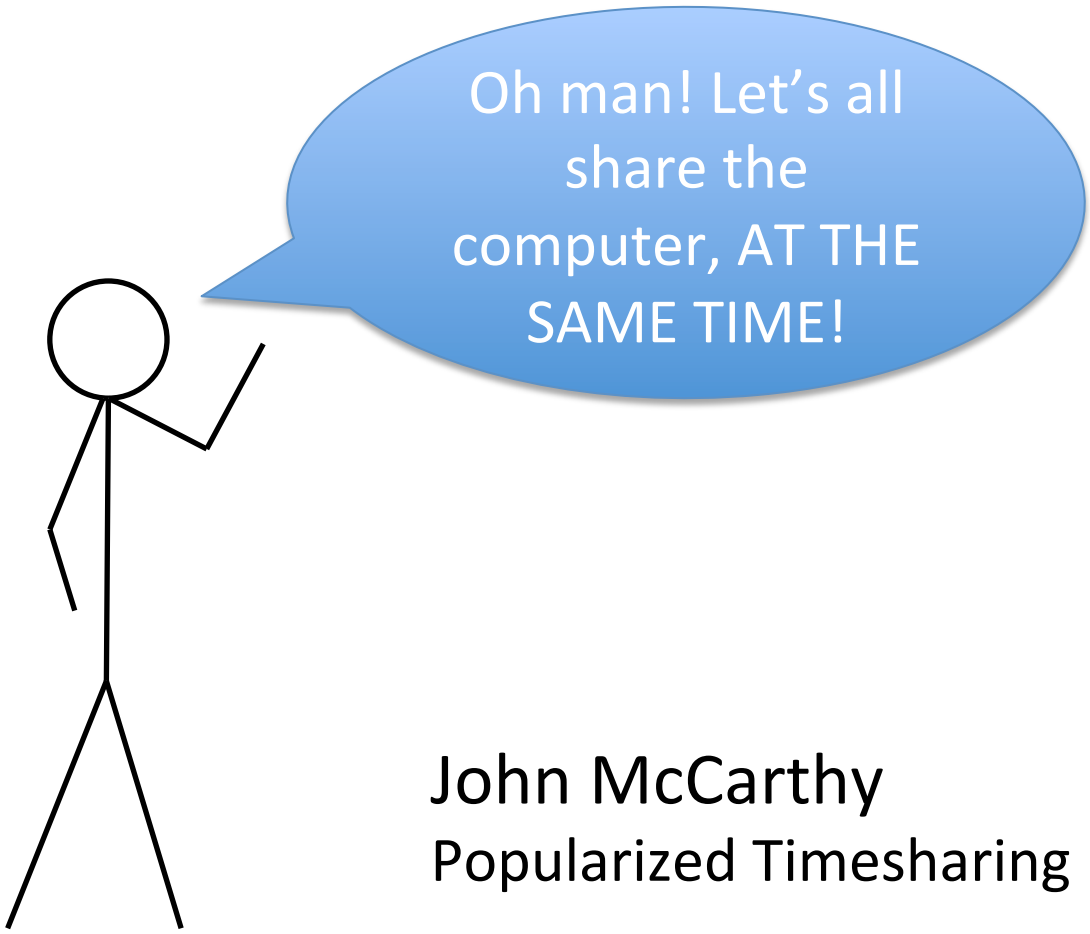
How can I use Mesos?

How can I build on Mesos?

What is Mesos?

A long time ago…

1957



John McCarthy

Popularized Timesharing

A long time ago…

2010

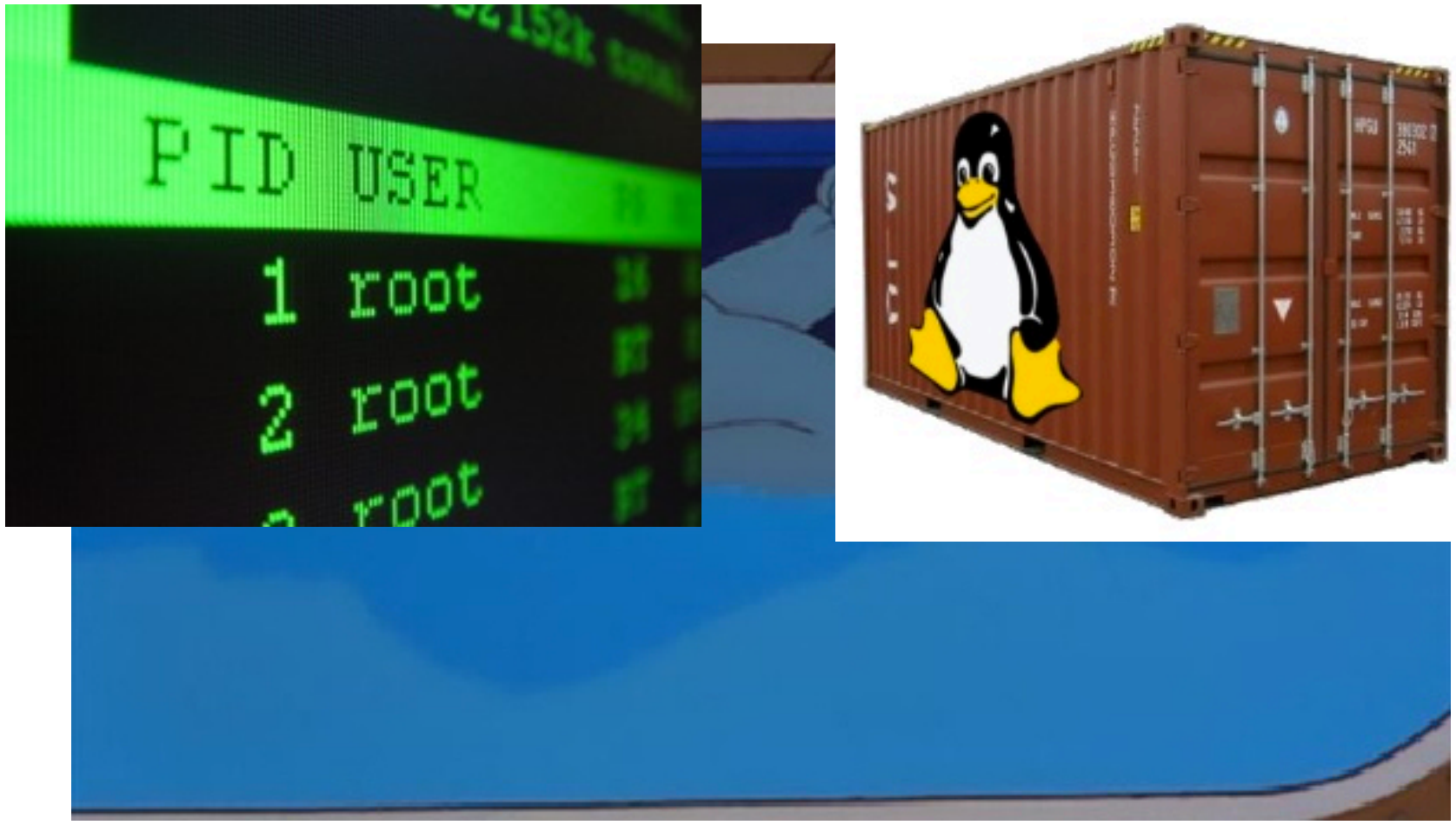
Ben Hindman

Popularized Mesos

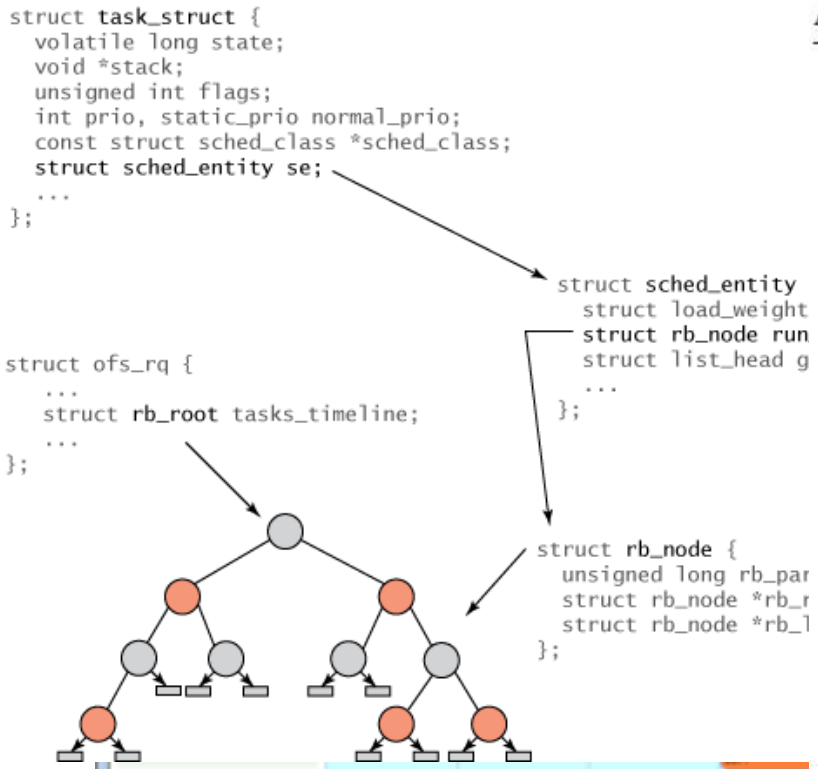
Good ideas today mirror good ideas of yesteryear

Mesos: an Operating System

Isolation



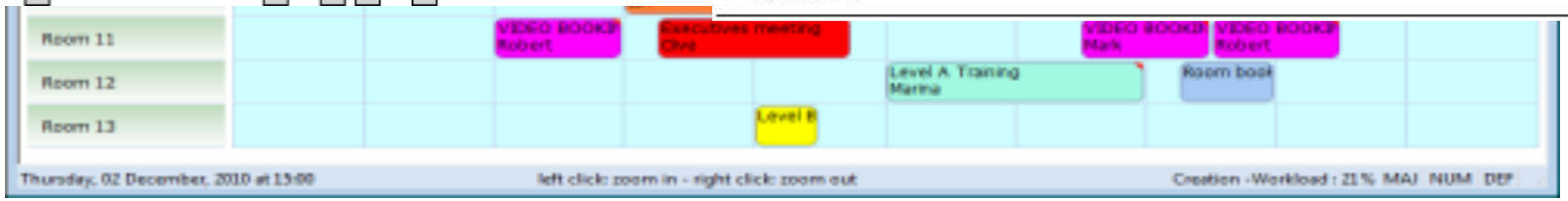
Resource Sharing



Algorithm 1 DRF pseudo-code

$R = \langle r_1, \cdots, r_m \rangle$ ▷ total resource capacities
$C = \langle c_1, \cdots, c_m \rangle$ ▷ consumed resources, initially 0
$s_i \ \ (i = 1..n)$ ▷ user i's dominant shares, initially 0
$U_i = \langle u_{i,1}, \cdots, u_{i,m} \rangle \ \ (i = 1..n)$ ▷ resources given to user i, initially 0

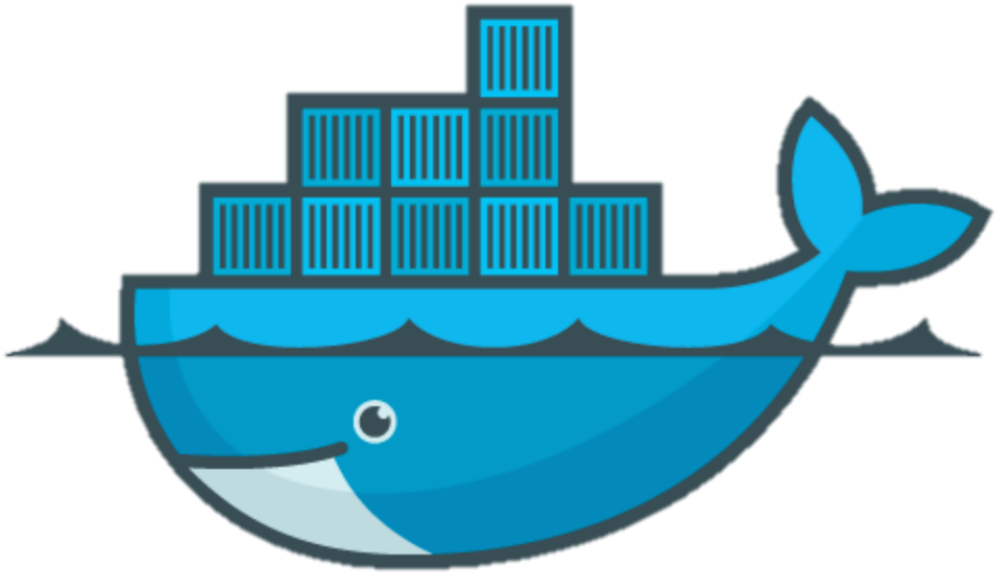
pick user i with lowest dominant share s_i
$D_i \leftarrow$ demand of user i's next task
if $C + D_i \leq R$ **then**
  $C = C + D_i$ ▷ update consumed vector
  $U_i = U_i + D_i$ ▷ update i's allocation vector
  $s_i = \max_{j=1}^{m} \{u_{i,j}/r_j\}$
else
  return ▷ the cluster is full
end if



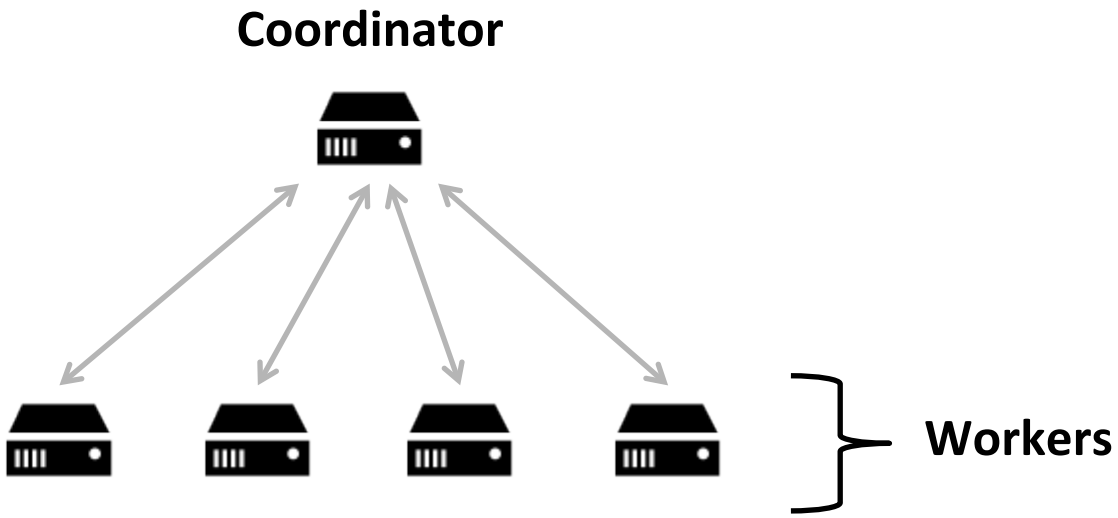
Common Infrastructure

- read(), write(), open()
- bind(), connect()
- apt-get, yum

- launchTask(), killTask(), statusUpdate()
- Docker

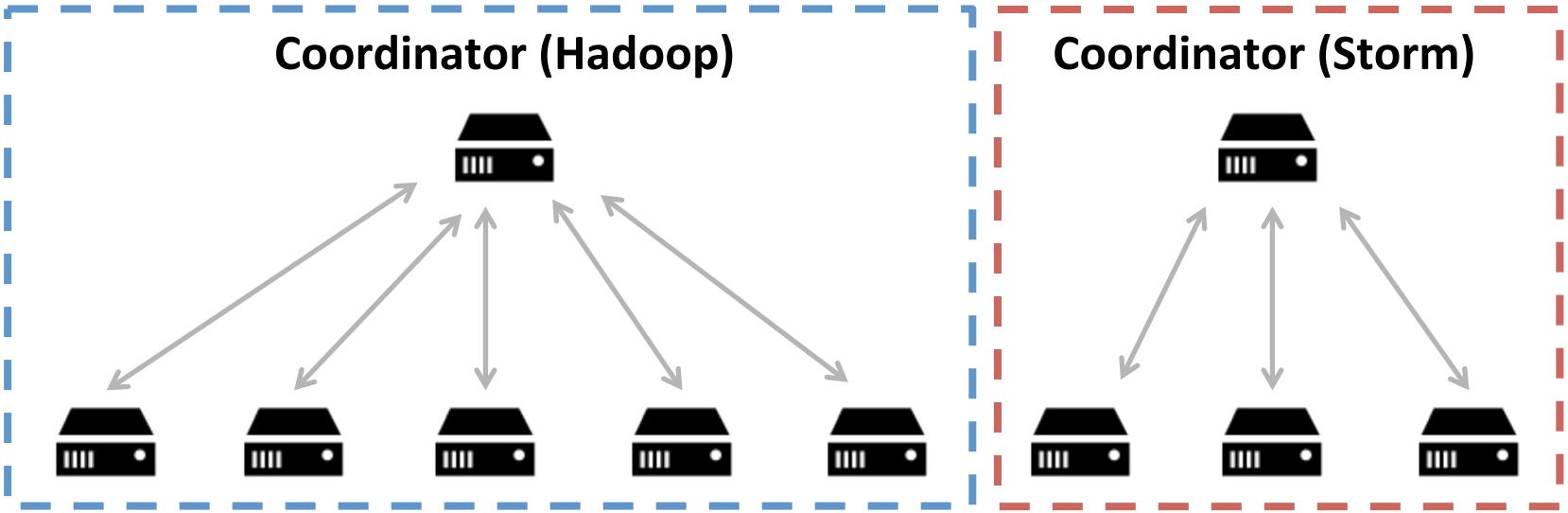


Distributed System* Anatomy

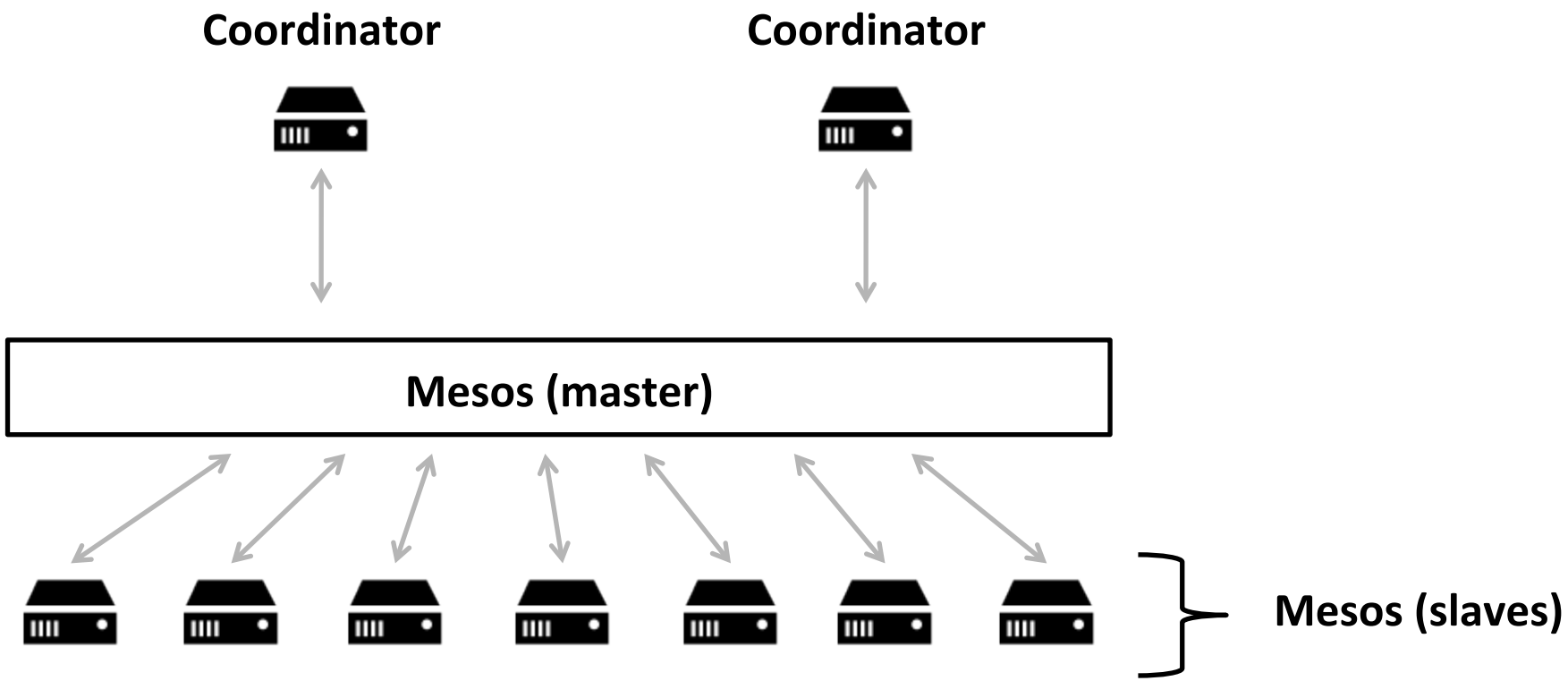


* Excluding peer-to-peer systems

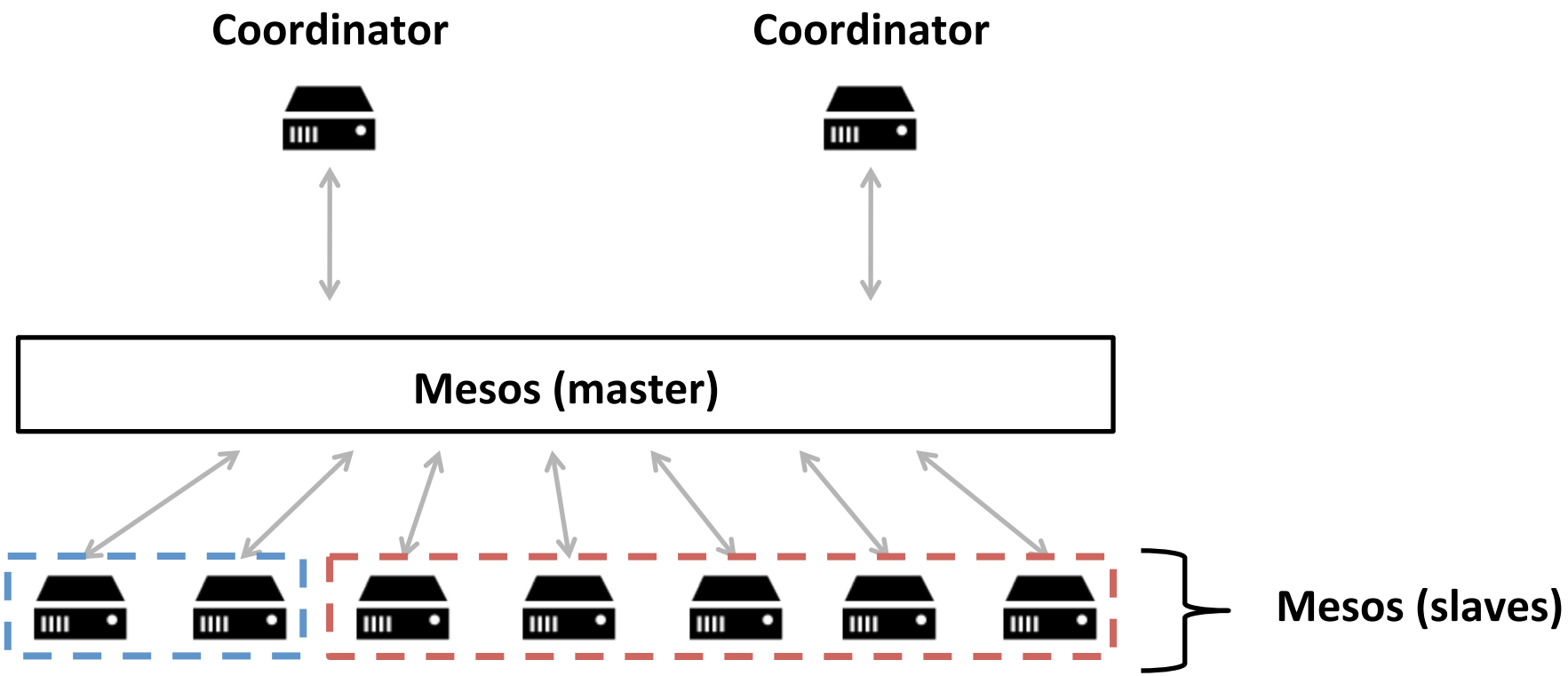
Static Partitioning



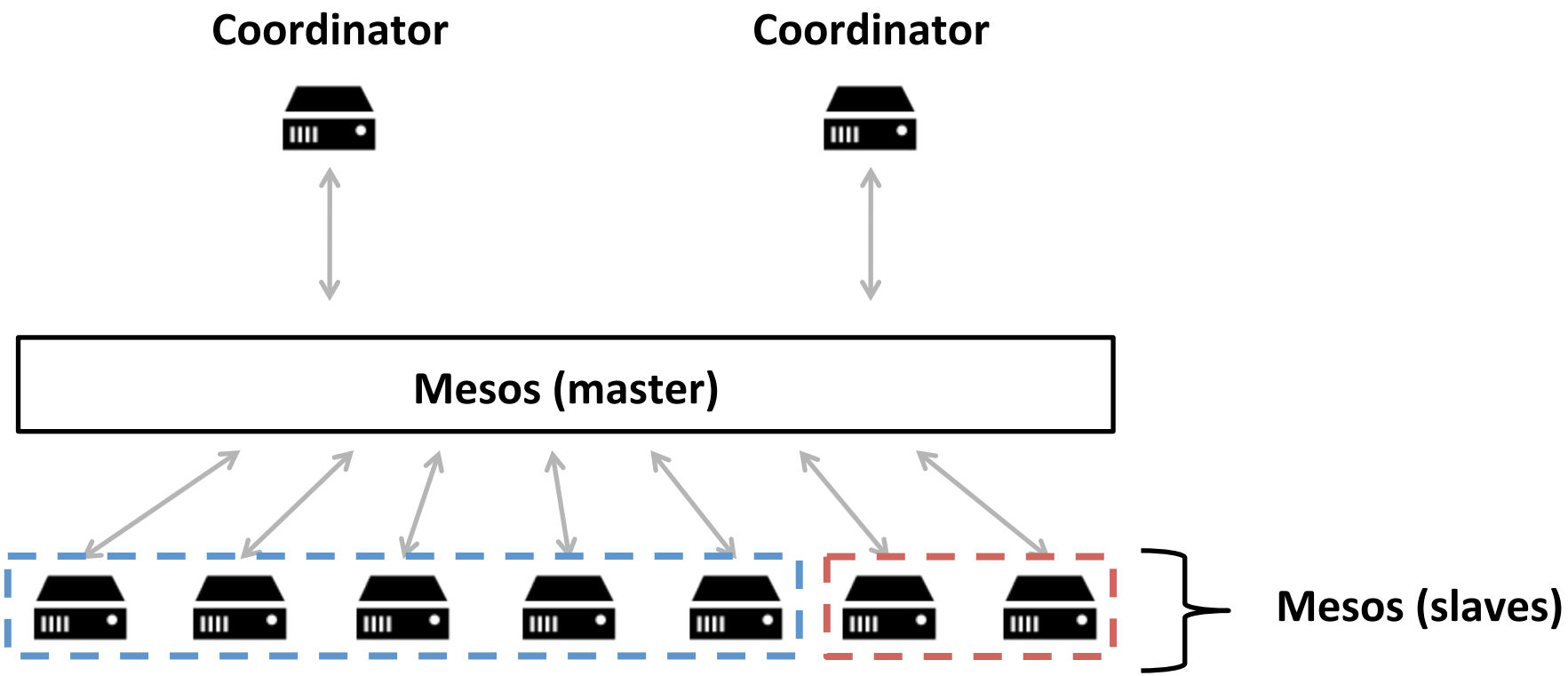
Mesos: a Level of Indirection



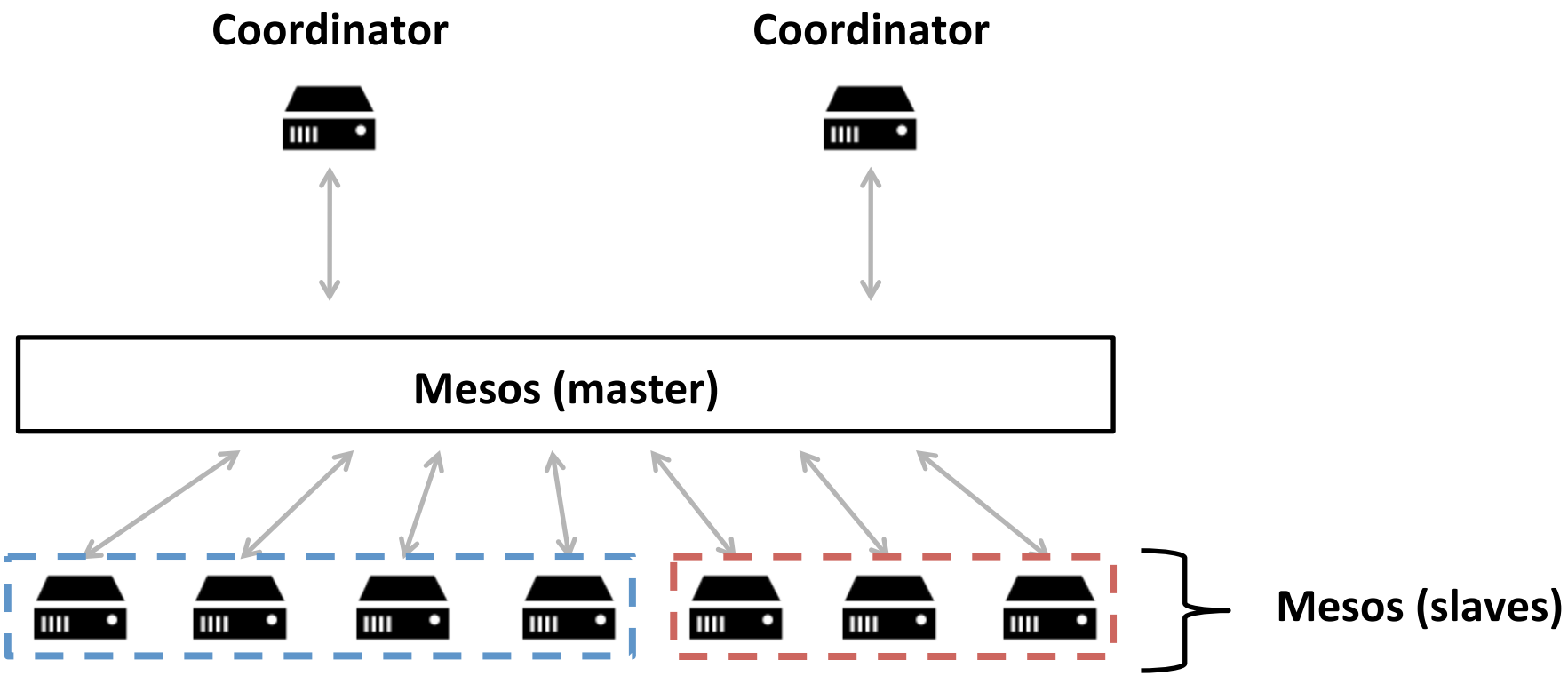
Mesos: a Level of Indirection



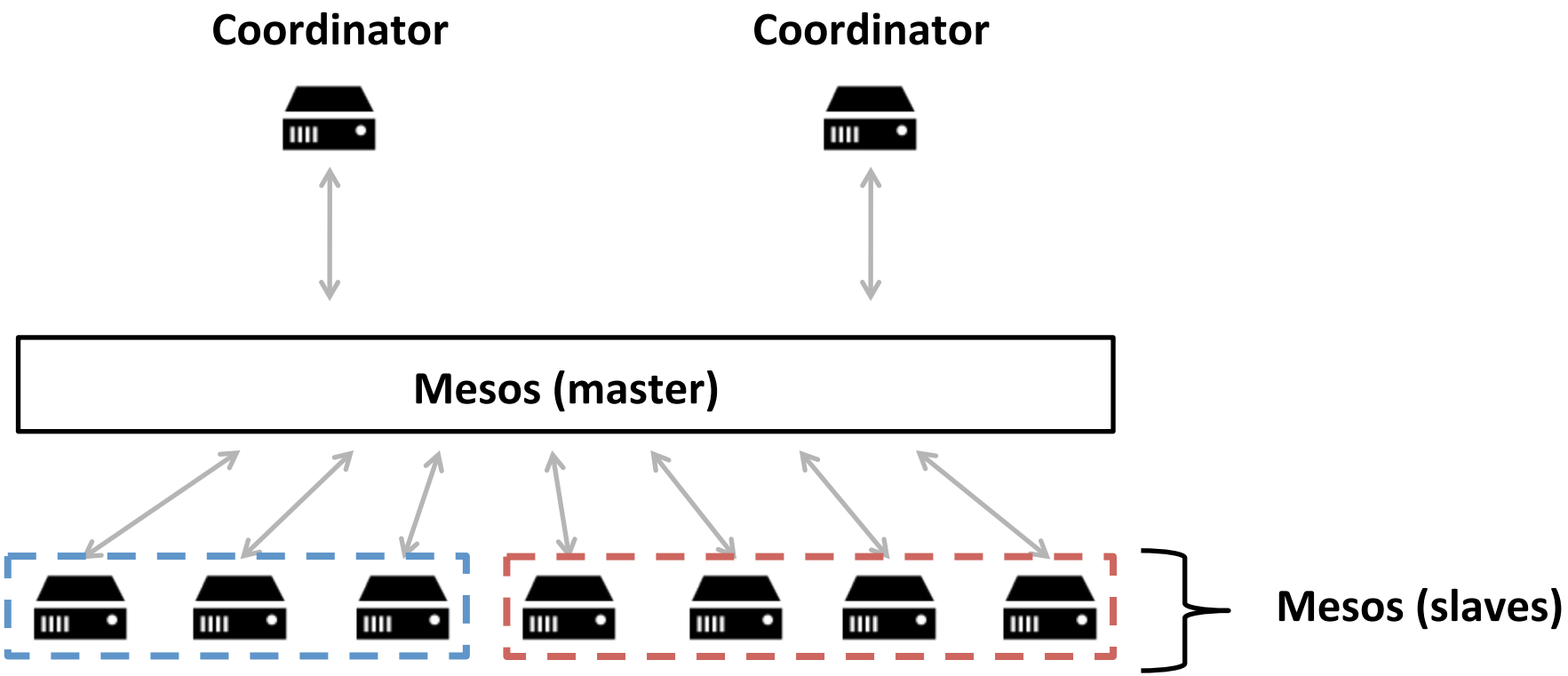
Mesos: a Level of Indirection



Mesos: a Level of Indirection



Mesos: a Level of Indirection

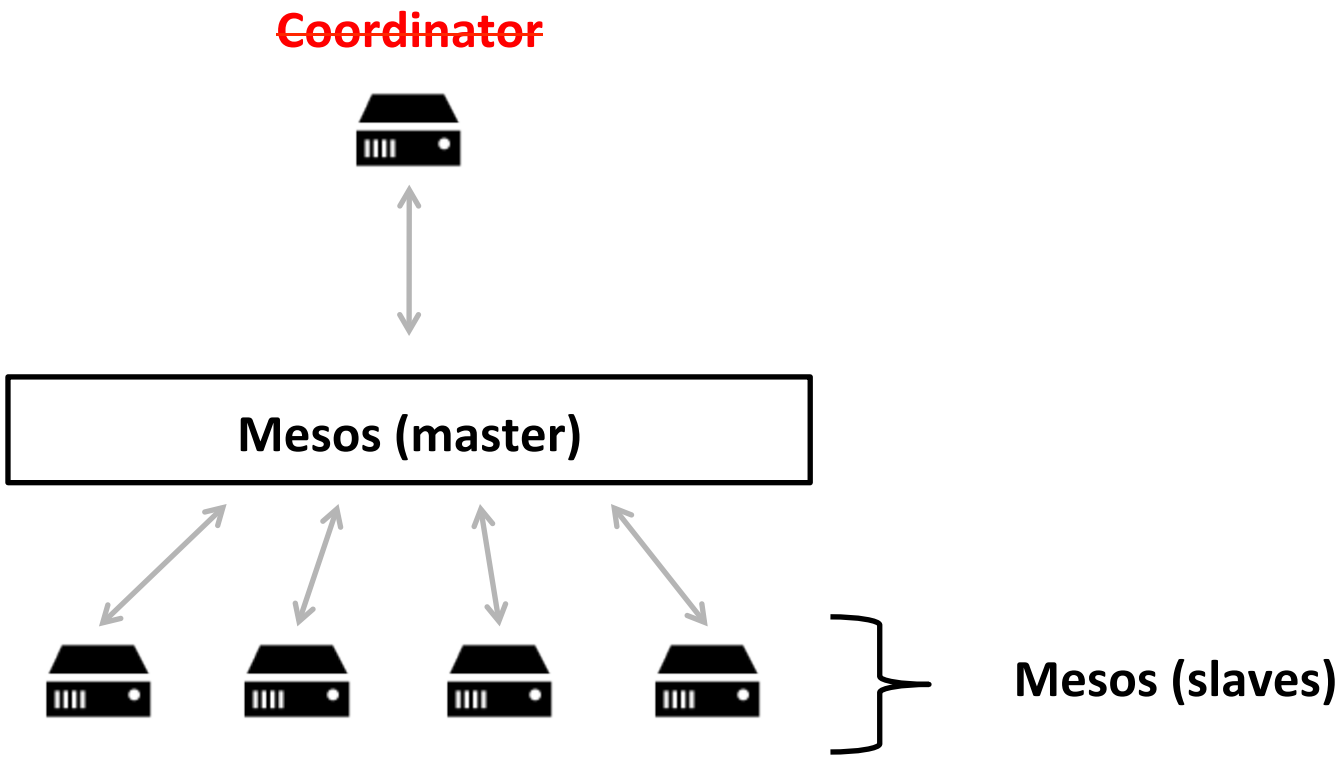


Coordinating Execution

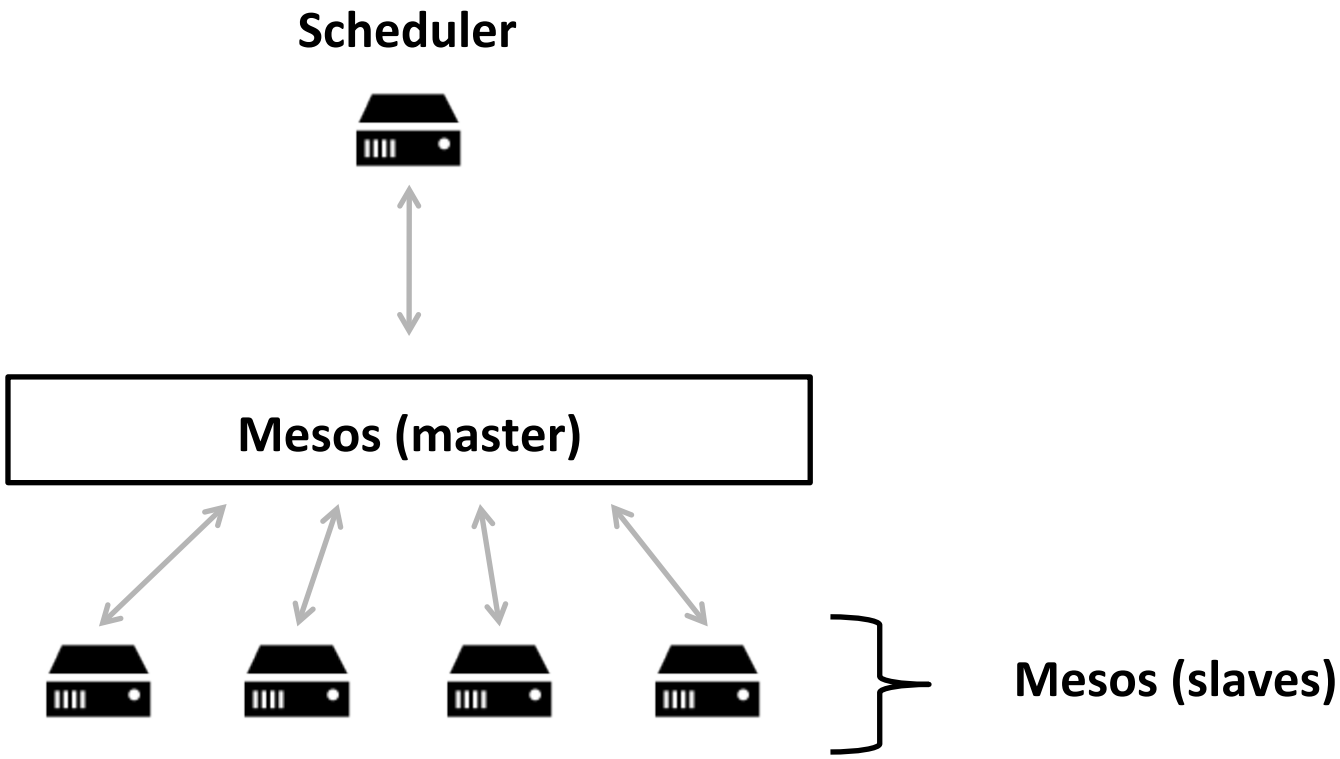
≈

Scheduling

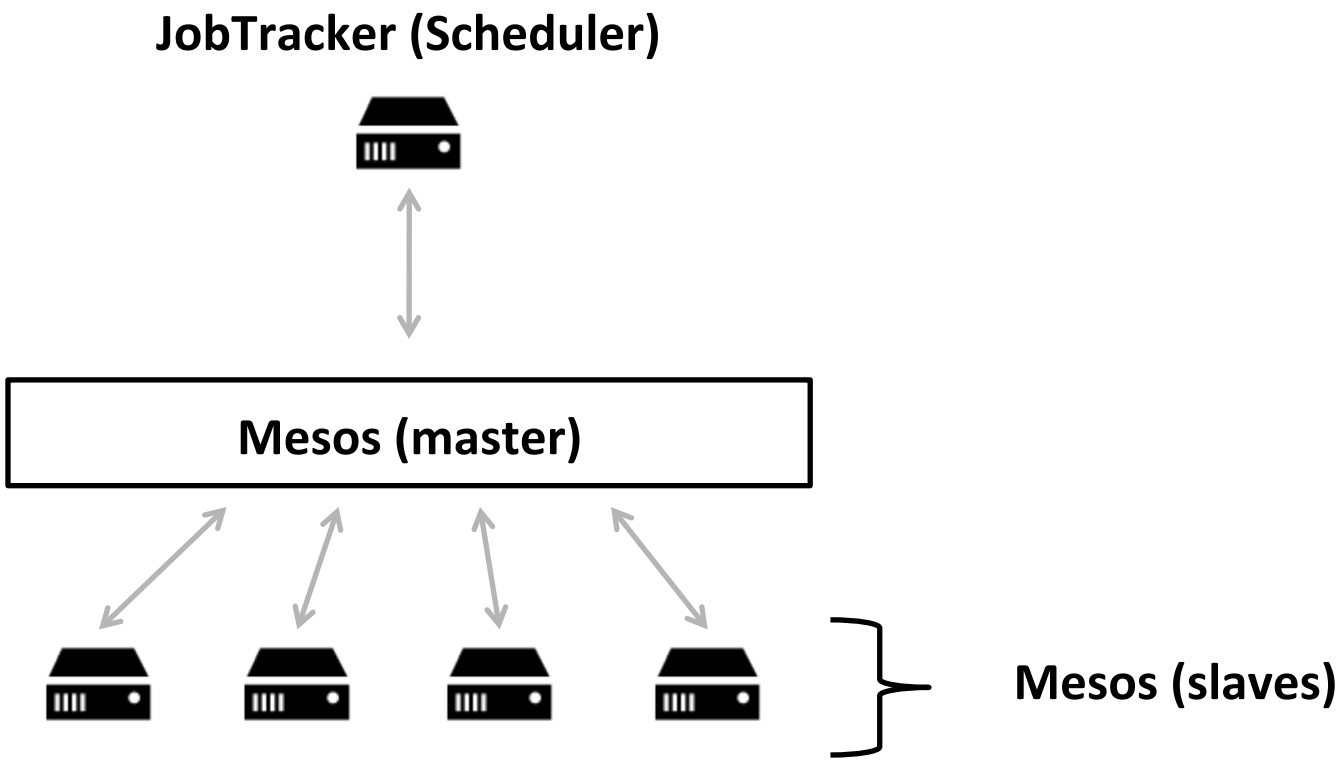
s/Coordinator/Scheduler/



s/Coordinator/Scheduler/



Apache Hadoop



Distributed System

≈

(Mesos) framework

a Mesos framework
is a distributed system
that has a coordinator

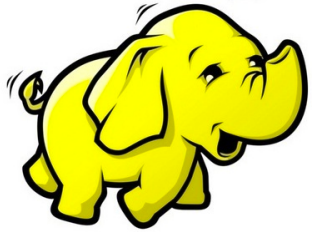
a Mesos framework
is a distributed system
that has a ~~coordinator~~

a Mesos framework
is a distributed system
that has a scheduler

a Mesos framework
is an app for your cluster

How can I use Mesos?

Tons of Flexibility!



Jenkins

- Continuous build server
- Just install a plugin!

Hadoop

- Multi-cluster isolation
- Fast startup

- Just run the repacked Cloudera CDH 4.2.1 MR1 distribution for Mesos

Marathon

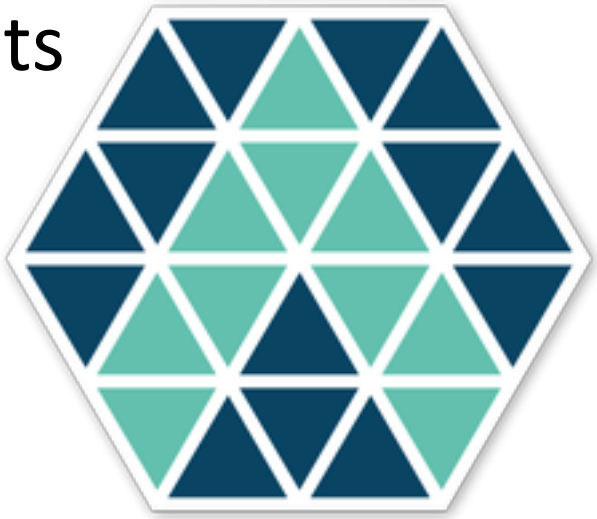
- PaaS on Mesos
- init.d for the cluster
- Docker support
- Scales at the click of a button
- Manages edge routers - HAProxy

Chronos

- Distributed cron
- Supports job dependencies
- REST API

Aurora

- Advanced PaaS on Mesos
- Powers Twitter
- Supports phased rollouts
- Supports complex deployments



Spark

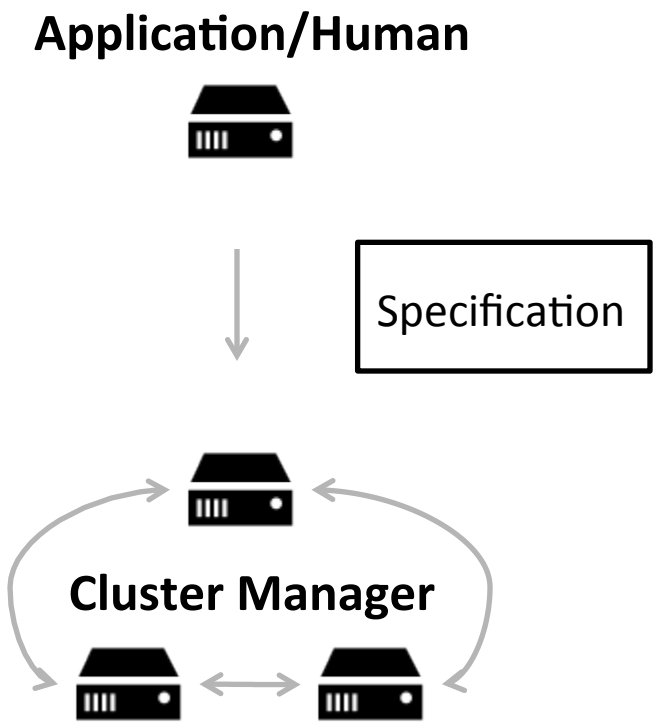
- In memory Map Reduce, built for “Medium Data”
- Supports SQL as well as Java, Python, and Scala
- Designed for interactive analysis via REPL

How do I use these?

- Free online interactive tutorials!
  - http://mesosphere.io/learn
- Covers all of the previously mentioned and many more

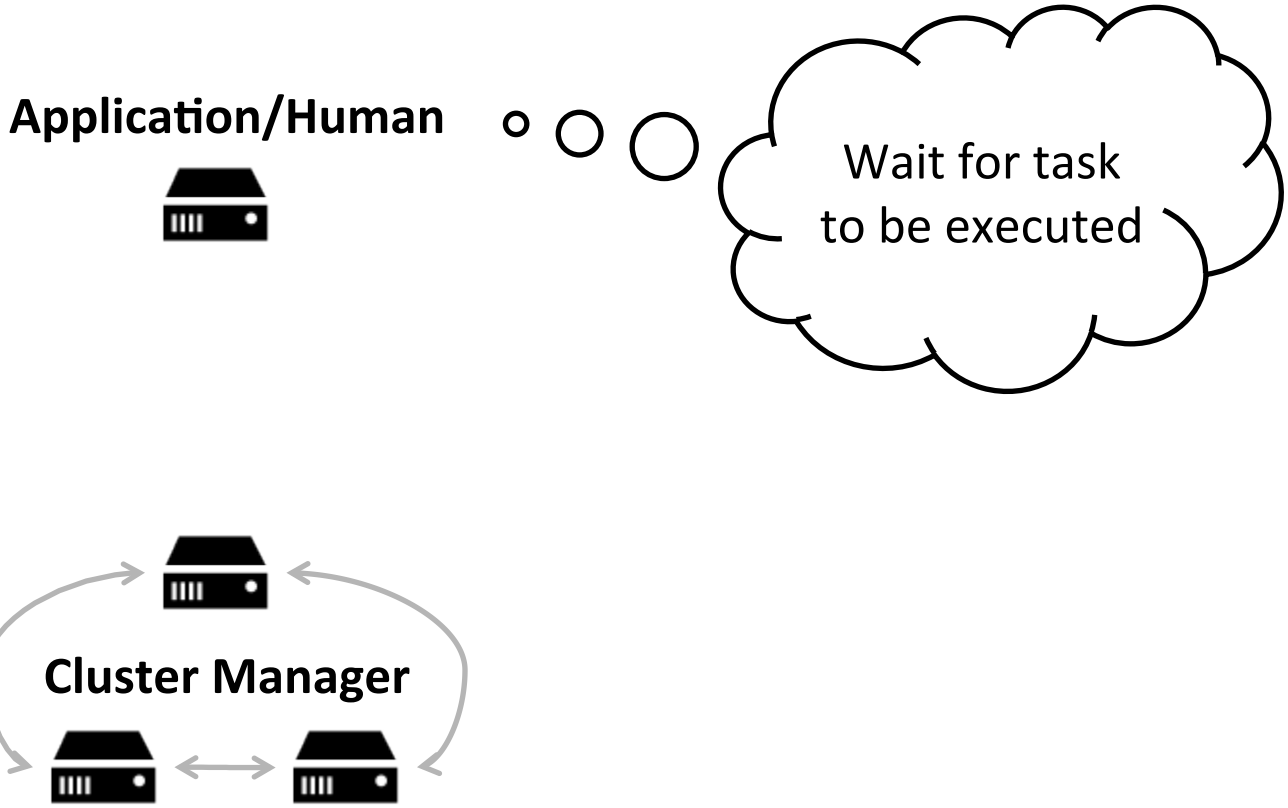
How can I build on Mesos?

Cluster Manager Status Quo

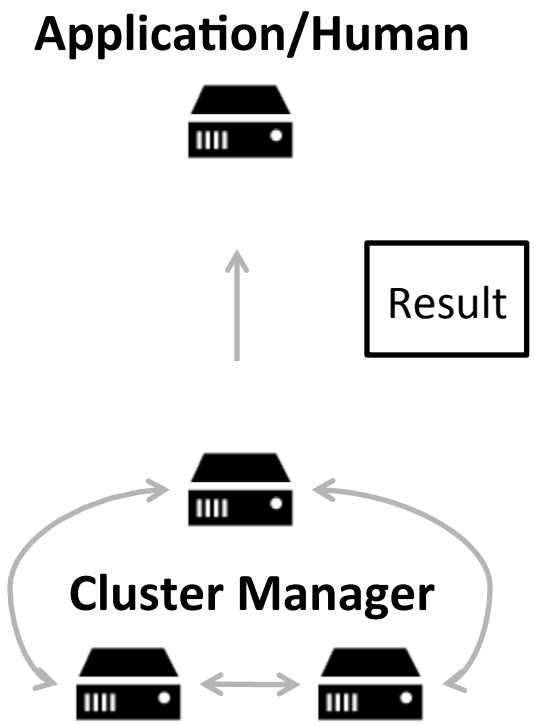


The specification includes as much information as possible to assist the cluster manager in scheduling and execution

Cluster Manager Status Quo



Cluster Manager Status Quo

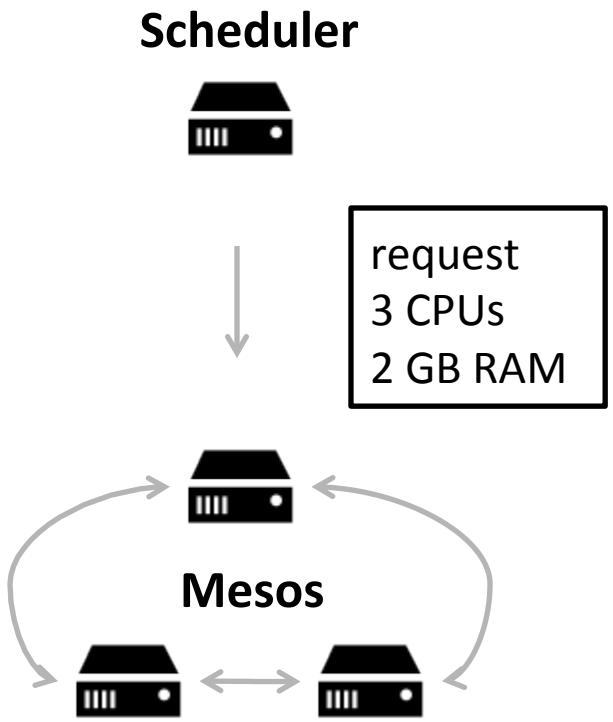


Problems with Specifications

① Hard to specify certain desires or constraints

② Hard to update specifications dynamically as tasks execute and finish/fail

An Alternative Model



- A request is purposely simplified subset of a specification
- It is just the required resources ***at that point in time***

What should you do if you can't satisfy a request?

What should you do if you can't satisfy a request?

① **Wait until you can …**

What should you do if you can't satisfy a request?

① **Wait until you can …**

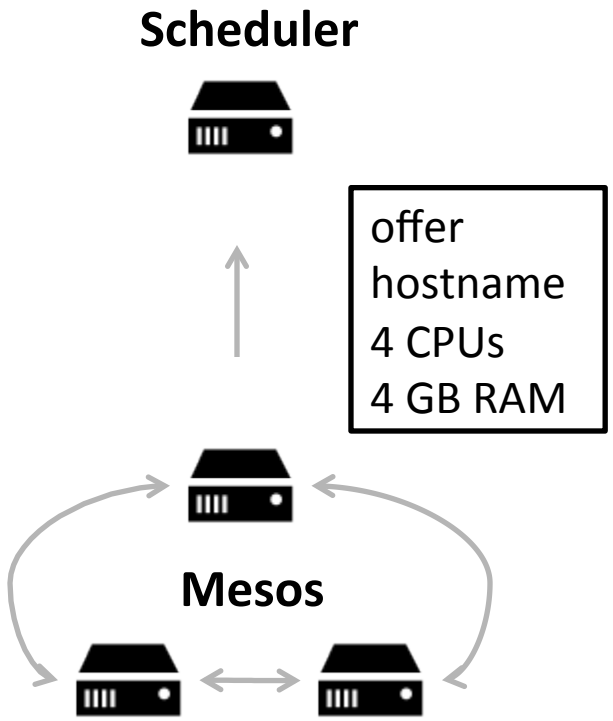
② ***Offer*** **best you can <u>immediately</u>**

What should you do if you can't satisfy a request?

① **Wait until you can …**

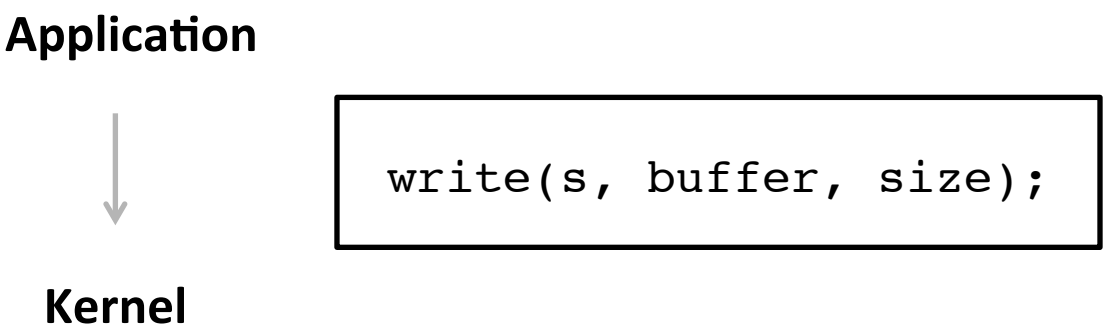
② ***Offer*** **best you can <u>immediately</u>**

Mesos Model

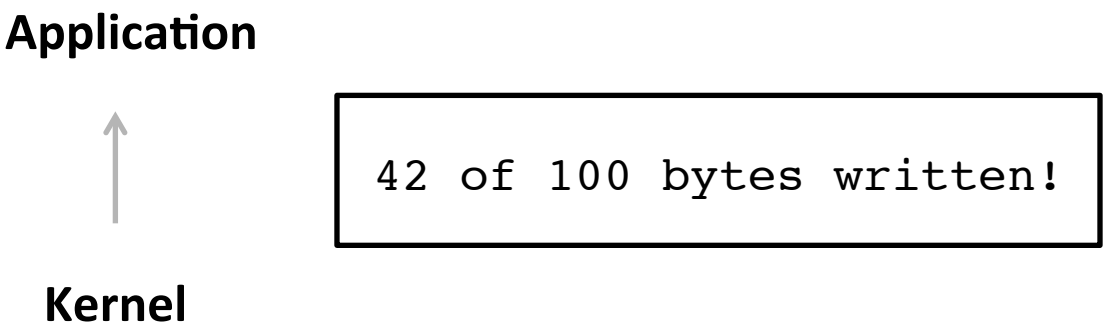


- Resources are allocated via *resource offers*
- A resource offer represents a snapshot of available resources that a scheduler can use to run tasks

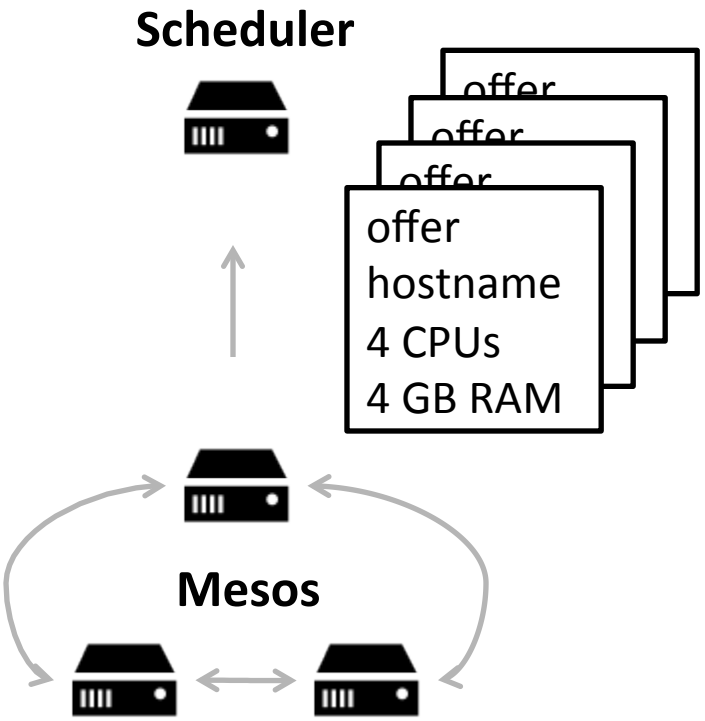
An Analogue: non-blocking sockets



An Analogue: non-blocking sockets

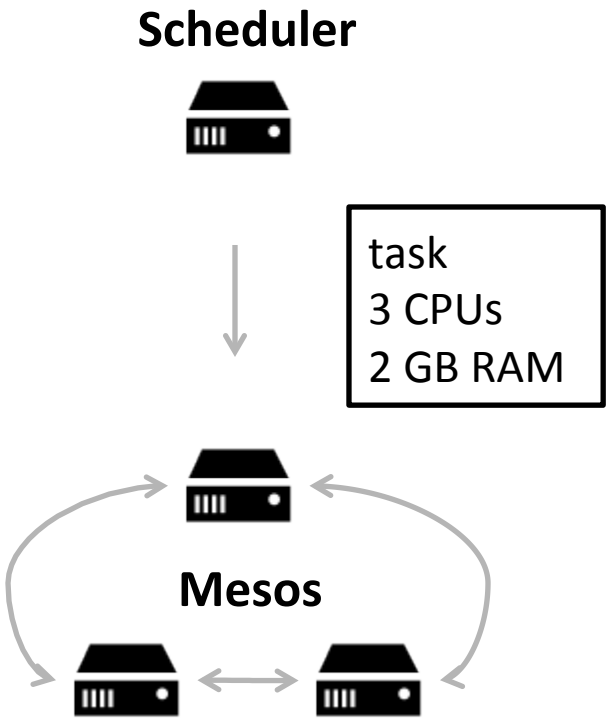


Mesos Model



Scheduler uses the offers to decide what tasks to run

Mesos Model



Scheduler uses the offers to decide what tasks to run

“Two-level scheduling”

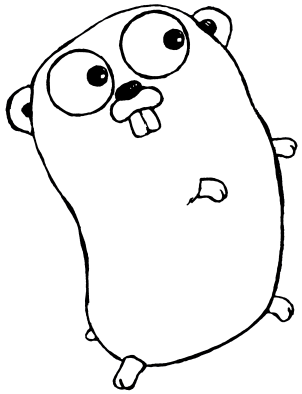
Two-level Scheduling

- Mesos: controls resource *allocations* to schedulers
- Schedulers: make decisions about what tasks to run given allocated resources

Two-level Scheduling Elsewhere

- Mesos influenced by operating system supported *user-space scheduling*
  - E.g. green threads, goroutines
- Mesos is designed less like a "cluster manager" and more like an operating system (or kernel)

Language Bindings



Should I build it on Mesos?

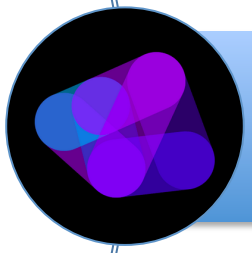
- Theme of MesosCon: it's easy to build frameworks
- Open source and proprietary frameworks are being created all the time
  - Two Sigma
  - Netflix
  - Twitter
  - Hubspot

But should I really build it on Mesos?

- Most users just use Marathon, Hadoop, Spark, and Chronos
- Why did we build our own?
  - Exotic workload

The Plan, redux

What is Mesos?



How can I use Mesos?

How can I build on Mesos?

Questions?

Thank you