Mesos

Study Notes

Richard Kuo

References & Resources

- Mesos - A Platform for Fine-Grained Resource Sharing in the Data Center , University of California, Berkeley, most of material in this slide deck are from this main article/slides ☺
- Dominant Resource Fairness: Fair Allocation of Multiple Resource Types, University of California, Berkeley
- To download Apache Mesos.
- For documentation and training materials Mesosphere.
- Datacenter as a Computer book 1st edition.
- apacheconeu-141118111331-conversion-gate02.pdf from RedHat.

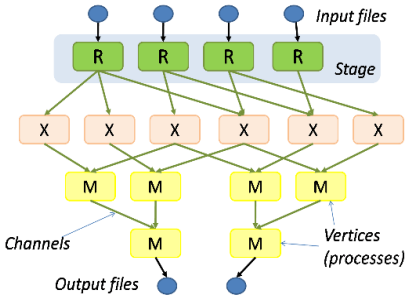
- Richard's study notes - learning mesos demo screen shots.

Outlines

- References and Resources
- Background
- Architecture
- Implementation
- Interface
- Support Frameworks
- Demo

Why?

- New applications need to be:
  - Fault tolerant (Withstand failure)
  - Scalable, Elastic
  - Multi-tenant (work with other apps)



Dryad

- Many different requirements and strategies for various types applications (big data, web, …)
- Infrastructure cost structures are different (public cloud, private cloud,…)

Static Partitioning

Issues with Statically Partitioned Data Centers

- **Complex**
  Machine sprawl, manual resize/scale
- **Limited**
  No software failure handling, “black box”
- **Inefficient**
  Static partitioning, overhead
- **Not Developer-Friendly**
  Long time to roll out software, development starts at the machine level

101 Unpacking: Meta-Scheduling?

*At its core, Mesos is a **focused, scalable, meta-scheduler** that provides **primitives** to express a wide variety of scheduling patterns and use cases.*

- It means that Mesos enables other distributed applications to define their own scheduling policies, with an core algorithm (DRF) to share resources across those applications.
- Mesos can run on 1-O(10^4) nodes.

So, it is a scheduler... that allows distributed applications to share resources in a cluster.

101 Summary : Distributed Systems Kernel Cluster Manager (google-ism)

- Mesos is built using the same principles as the Linux kernel, only at a different level of abstraction.

“We wanted people to be able to program for the data-center just like they program for their laptop” ~ Ben Hindman

  - Computer : Data-center
  - Kernel : Mesos
  - Application : Distributed application
  - Operating System : (Mesos+Frameworks+ Ecosystem)

Mesos is...

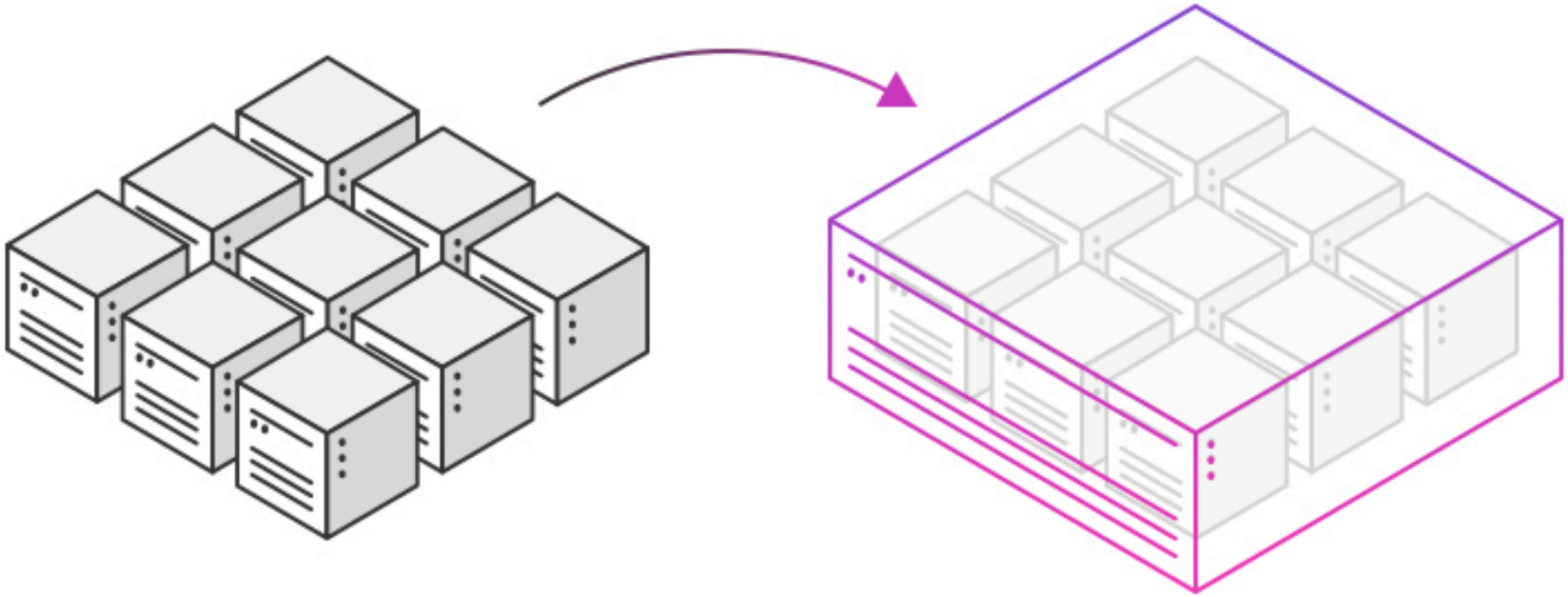
- A cluster resource broker
- Scalable, fault-tolerant, battle-tested an SDK for distributed apps, scales to 10,000s of nodes
- Top-level Apache project
- Twitter and Airbnb are major users and contributors
- APIs for C++, Python, JVM, Go
- Pluggable CPU, memory, IO isolation
- Packages and commercial support through Mesosphere
- Highly available, scalable, elastic

- Some call it “Datacenter Operating System” ?? Maybe


Mesos Users Today
vimeo
OpenTable
NETFLIX
Atigeo
airbnb
sharethrough
salesforce
MediaCrossing
CommonwealthBank
UCSF
DueDil
HubSpot
xogito
radical thinking
PayPal
CONVIVA
shopify
Sigmoid Analytics
BEST BUY
CLOUD PHYSICS
device scape
CATEGORIZE
pb pinkbike
medidata
ignidata
iQIY 爱奇艺

Warehouse Scale Datacenter

Datacenter OS (DCOS)



Datacenter or Cloud

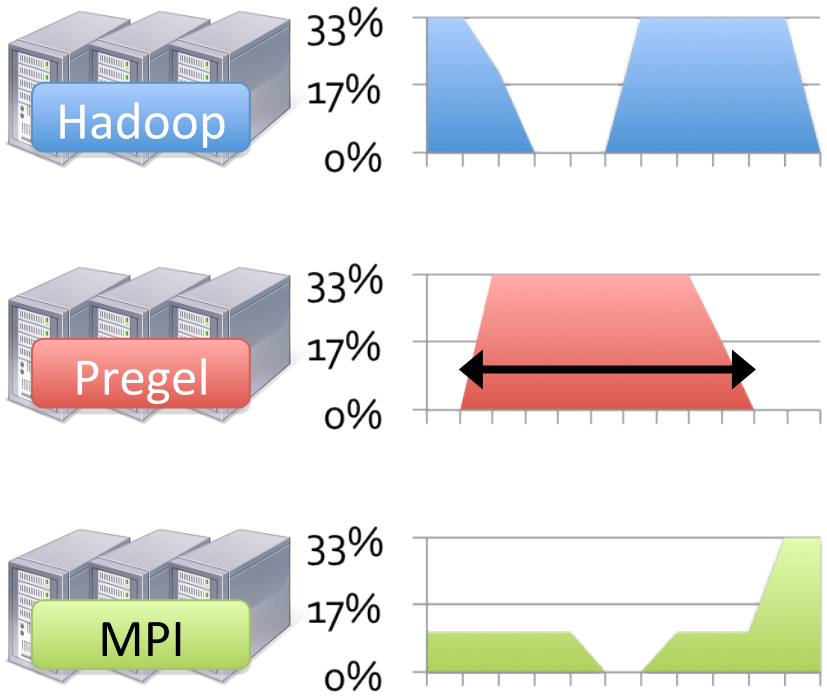
Gone are the days where writing and deploying new applications means managing individual machines and static partitions.

With Mesosphere

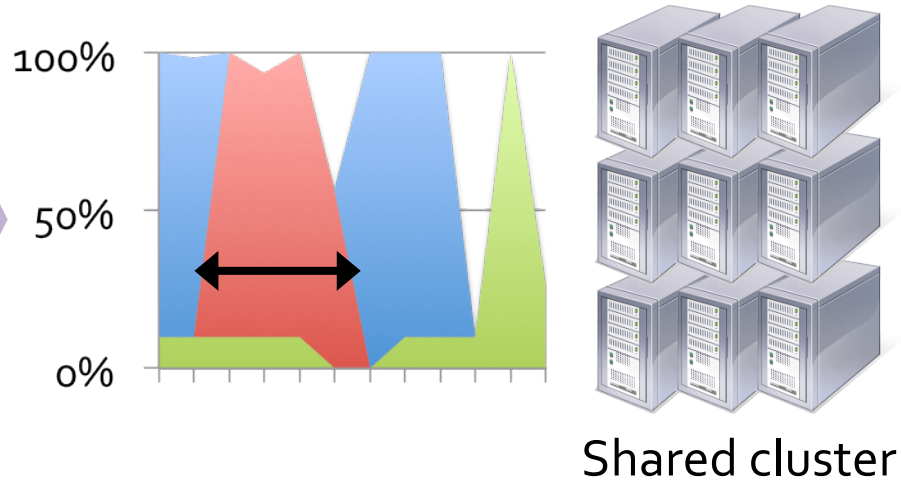
Pool your datacenter and cloud resources, so all your apps run together on the same machines —reducing complexity and waste.

Partitioning

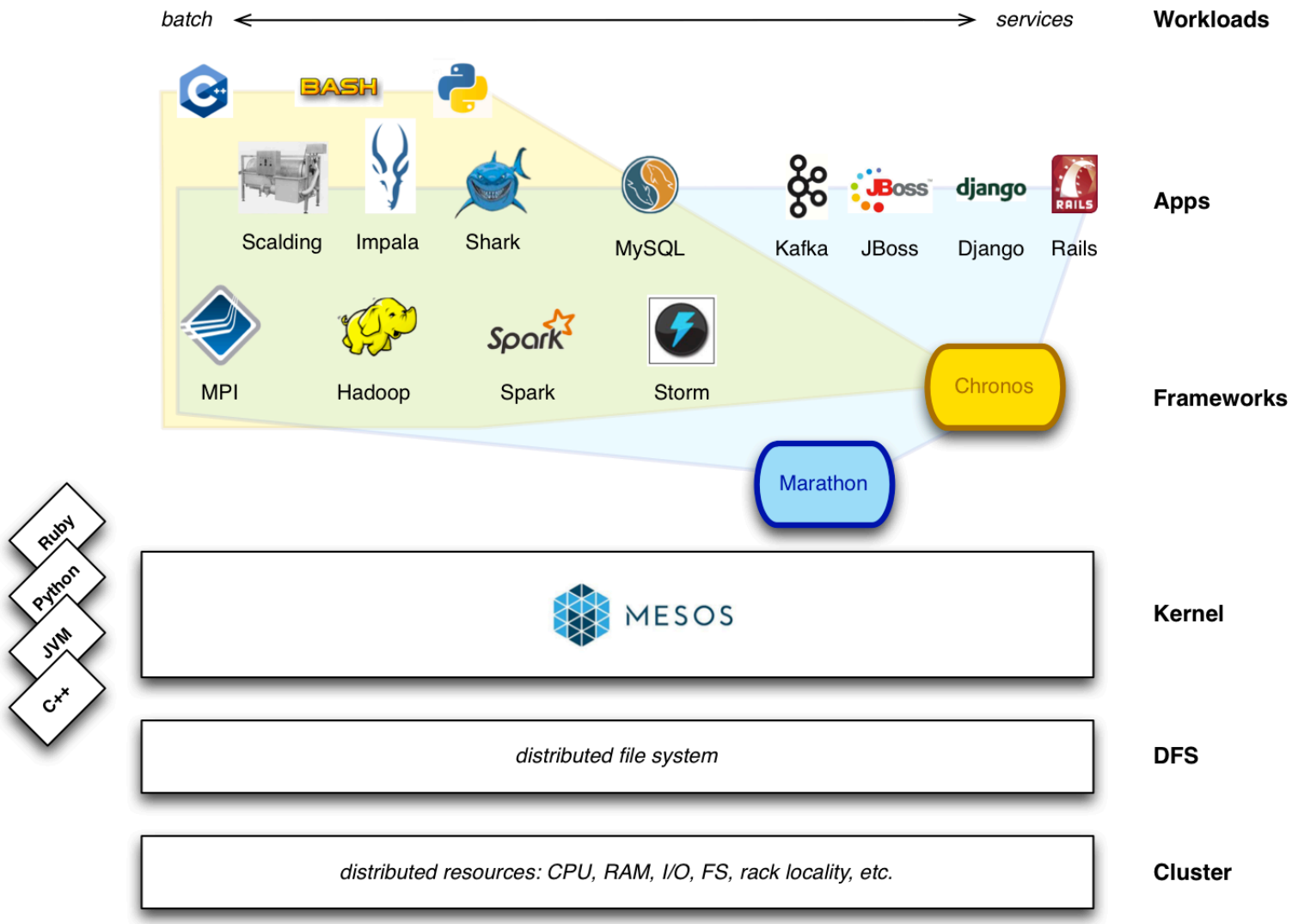
Today: static partitioning



Mesos: dynamic sharing



Mesos Architecture



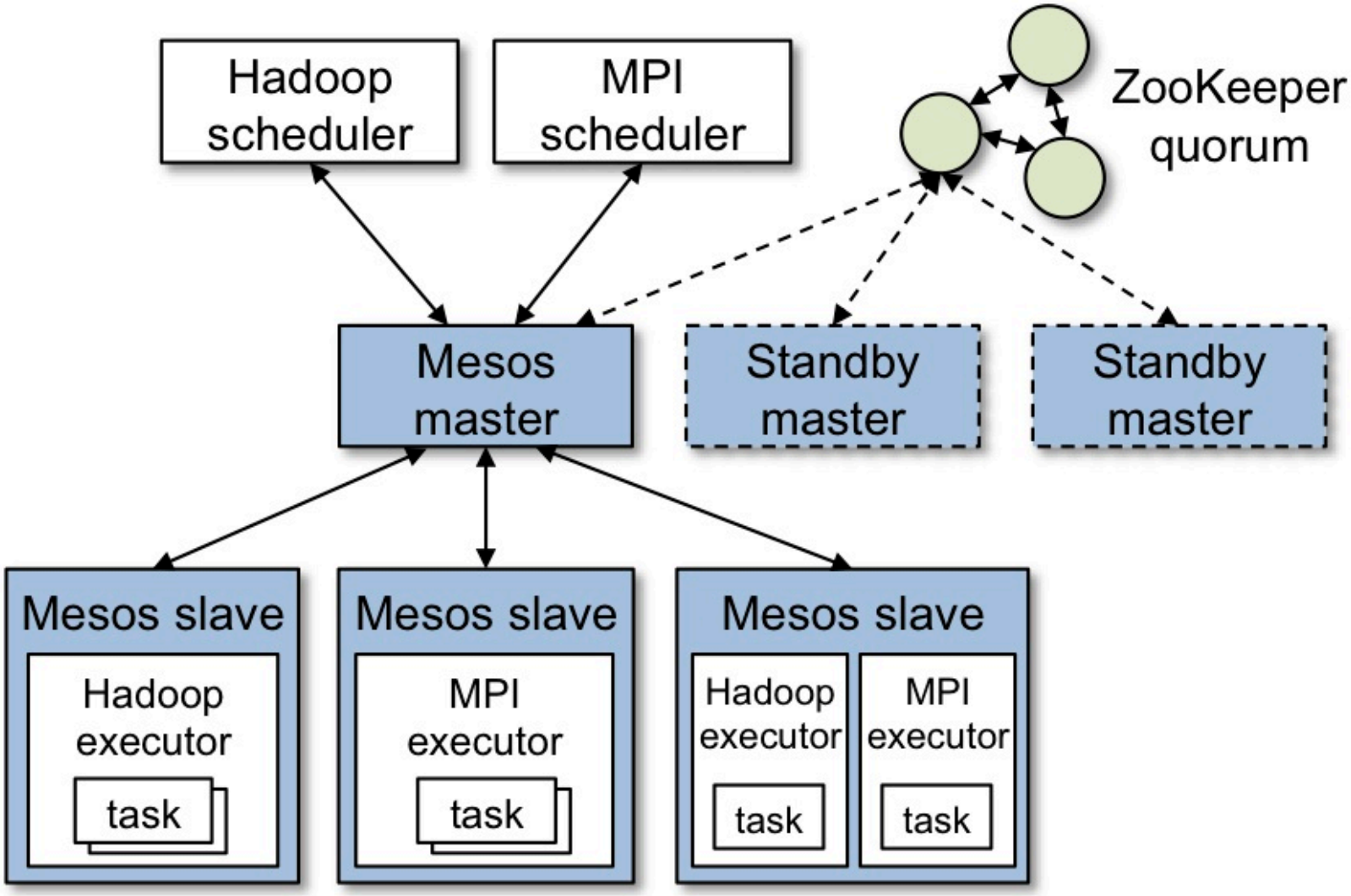
Framework

A framework is an application that runs distributed applications on Mesos.

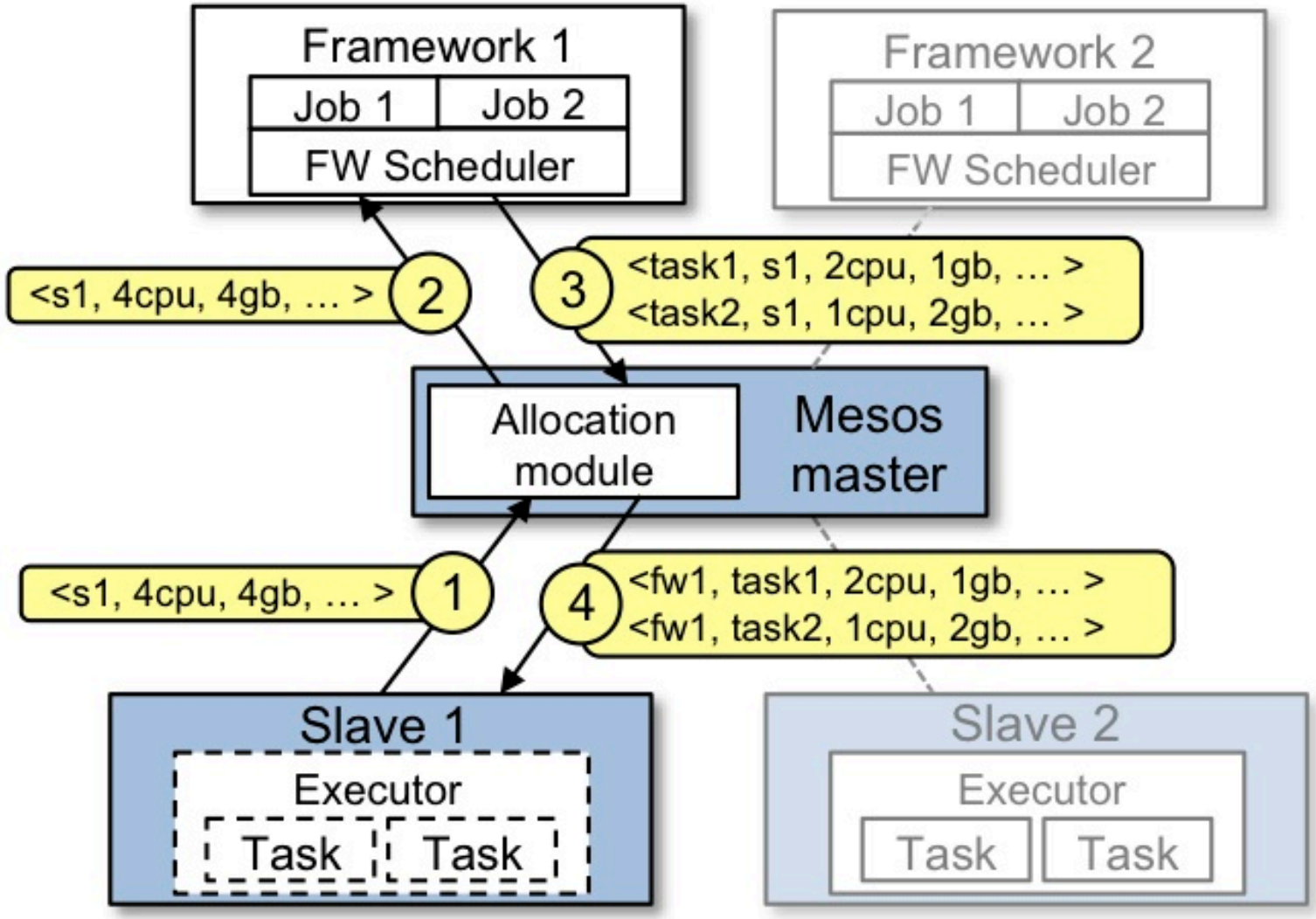
- Framework = scheduler + executor
- Schedulers get resource offers; Executors run tasks

API/SDK in multiple languages to build its own frameworks.

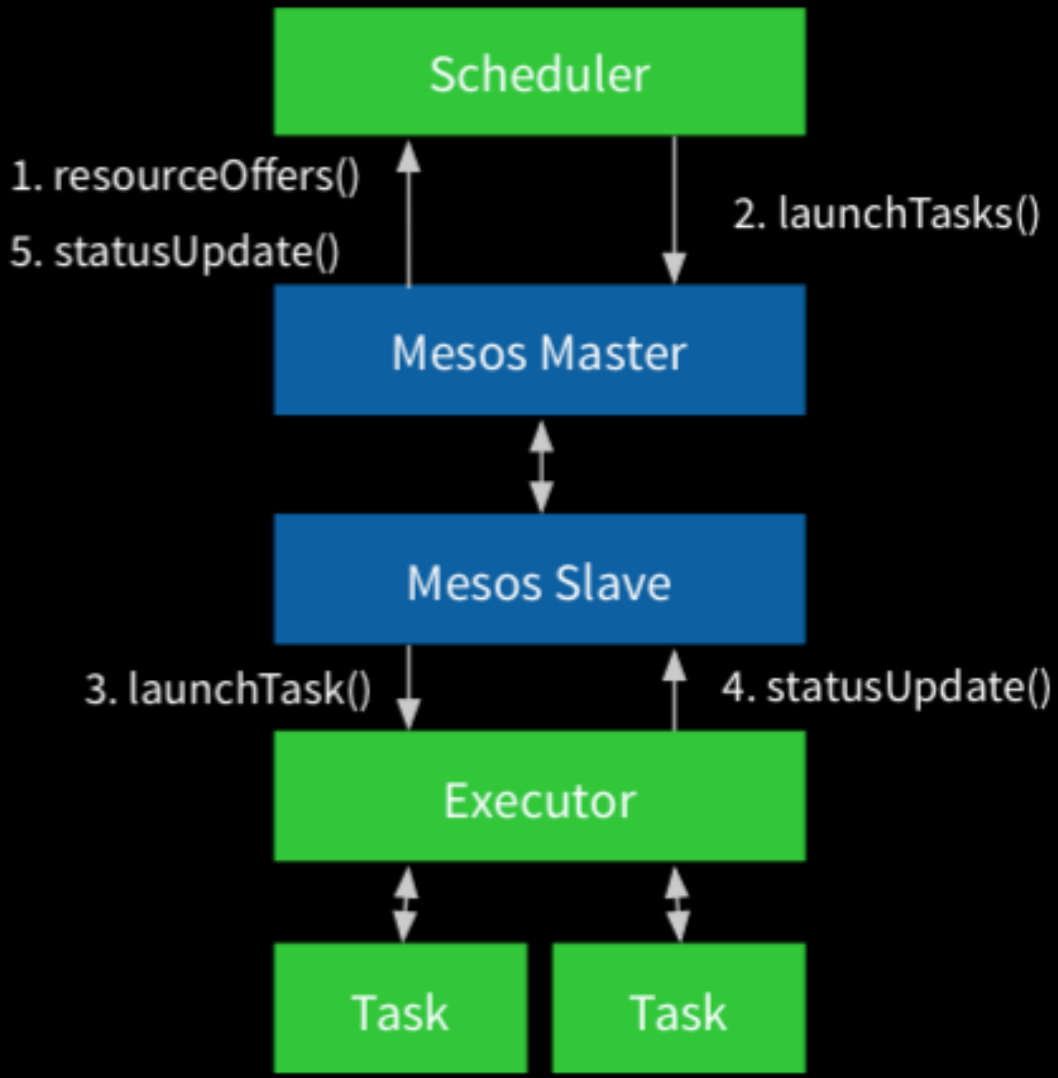
Mesos Key Components



A Framework Gets Scheduled to Run a Task



Resource Offers and Launching a Task



mesosphere

Design

- Small microkernel-like core that pushes scheduling logic to frameworks
- Fine-grained sharing:
  - Allocation at the level of *tasks* within a job
  - Improves utilization, latency, and data locality
- Resource offers:
  - Simple, scalable application-controlled scheduling mechanism

Implementation

- 20,000 lines of C++
- Master failover using ZooKeeper
- Frameworks ported: Hadoop, MPI, Torque, Spark,…
- Open source in Apache

Mesos API

Scheduler Callbacks	Scheduler Actions
resourceOffer(offerId, offers) offerRescinded(offerId) statusUpdate(taskId, status) slaveLost(slaveId)	replyToOffer(offerId, tasks) setNeedsOffers(bool) setFilters(filters) getGuaranteedShare() killTask(taskId)

Executor Callbacks	Executor Actions
launchTask(taskDescriptor) killTask(taskId)	sendStatus(taskId, status)

Popular Frameworks

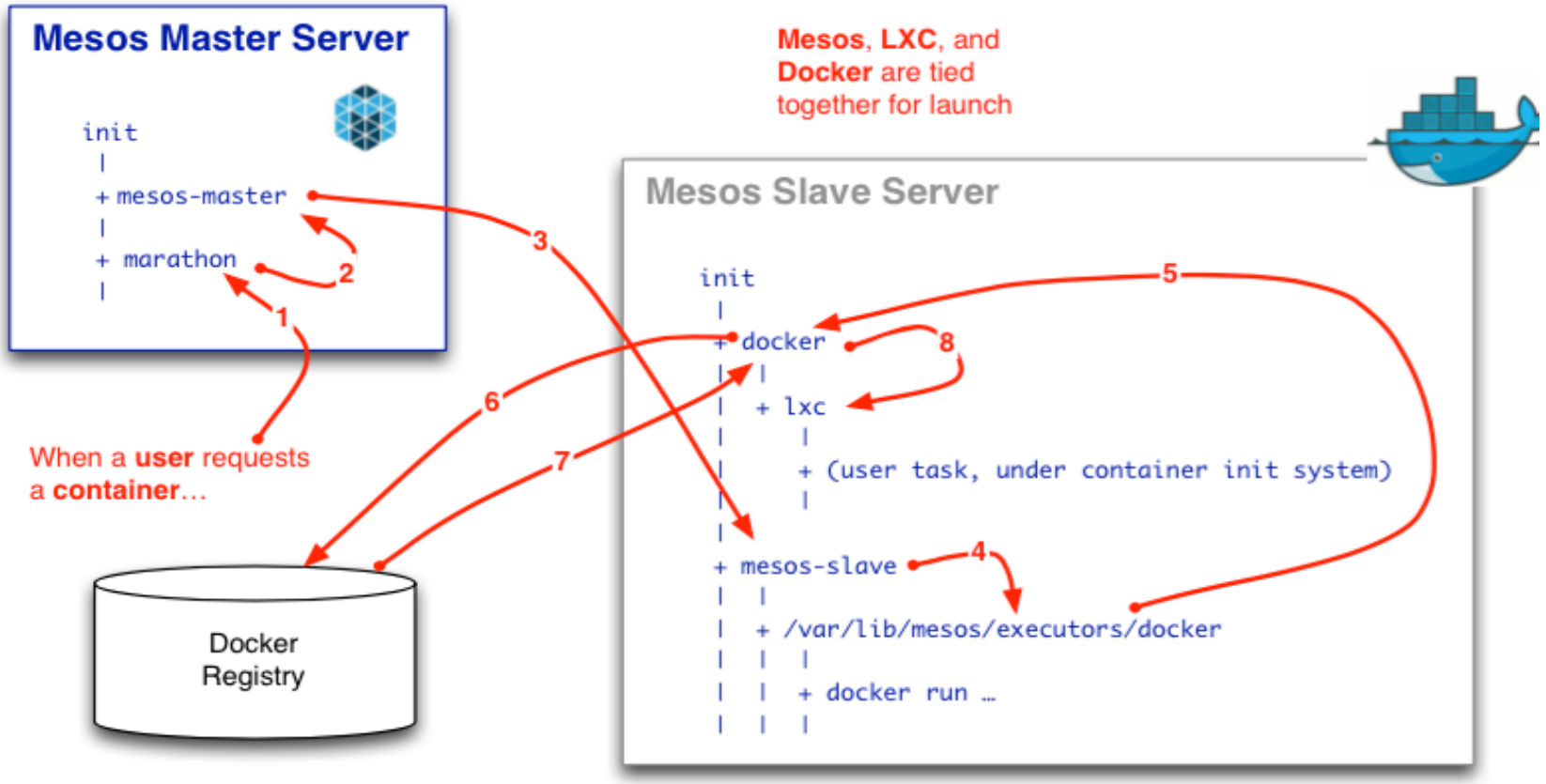
- Continuous Integration: Jenkins, GitLab
- Big Data: Hadoop, Spark, Storm, Kafka, Cassandra, HyperTable, MPI
- Python workloads: DPark, Exelixi
- Meta-Frameworks / HA Services: Aurora, Marathon
- Distributed Cron: Chronos
- Containers: Docker

- More …

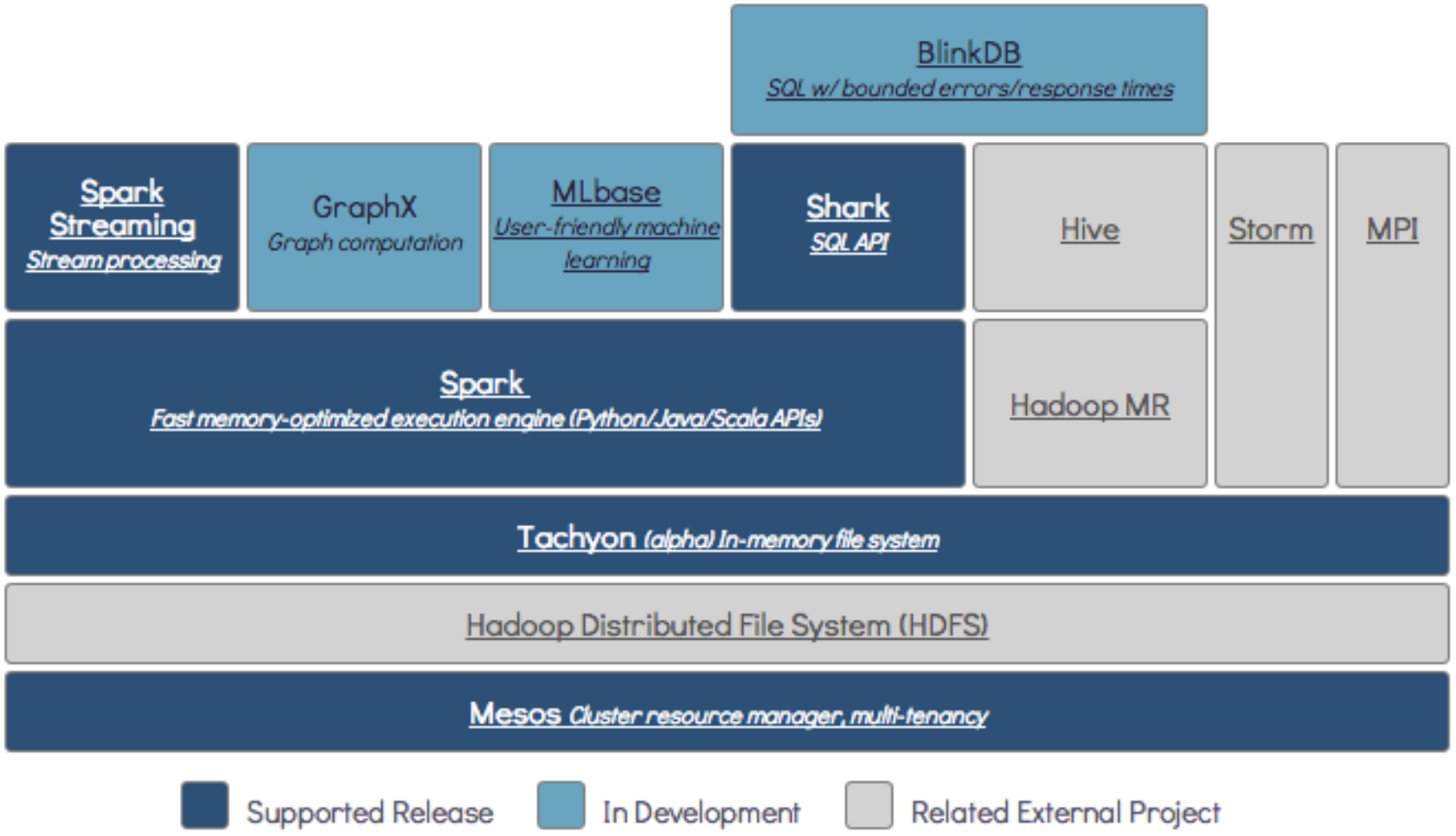
Framework Isolation

- Mesos uses OS isolation mechanisms, use Linux container for isolation.
- Containers currently support CPU, memory, IO and network bandwidth isolation.

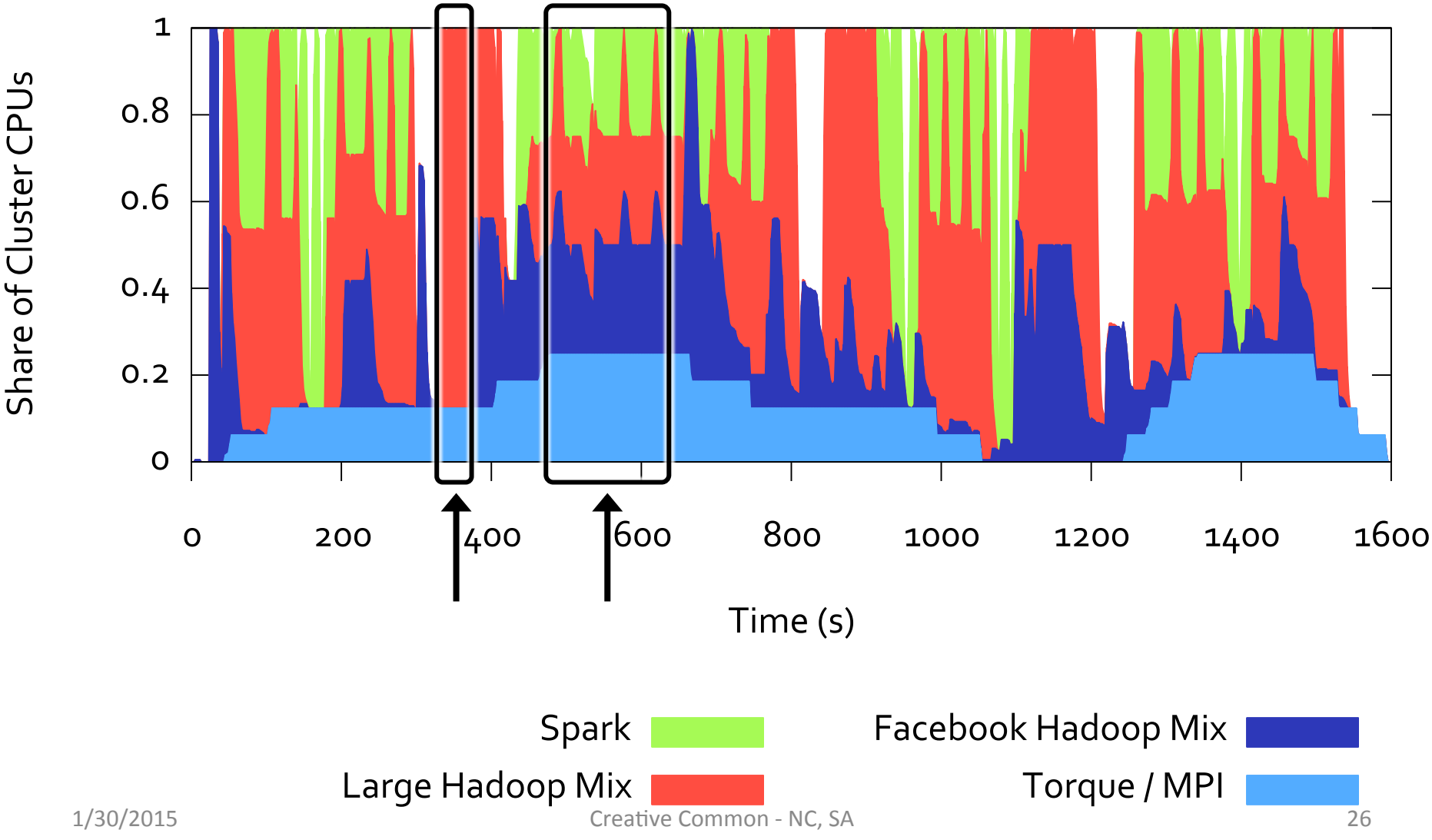
Docker on Mesos



Spark and Hadoop on Mesos



Results – Dynamic Resources Sharing



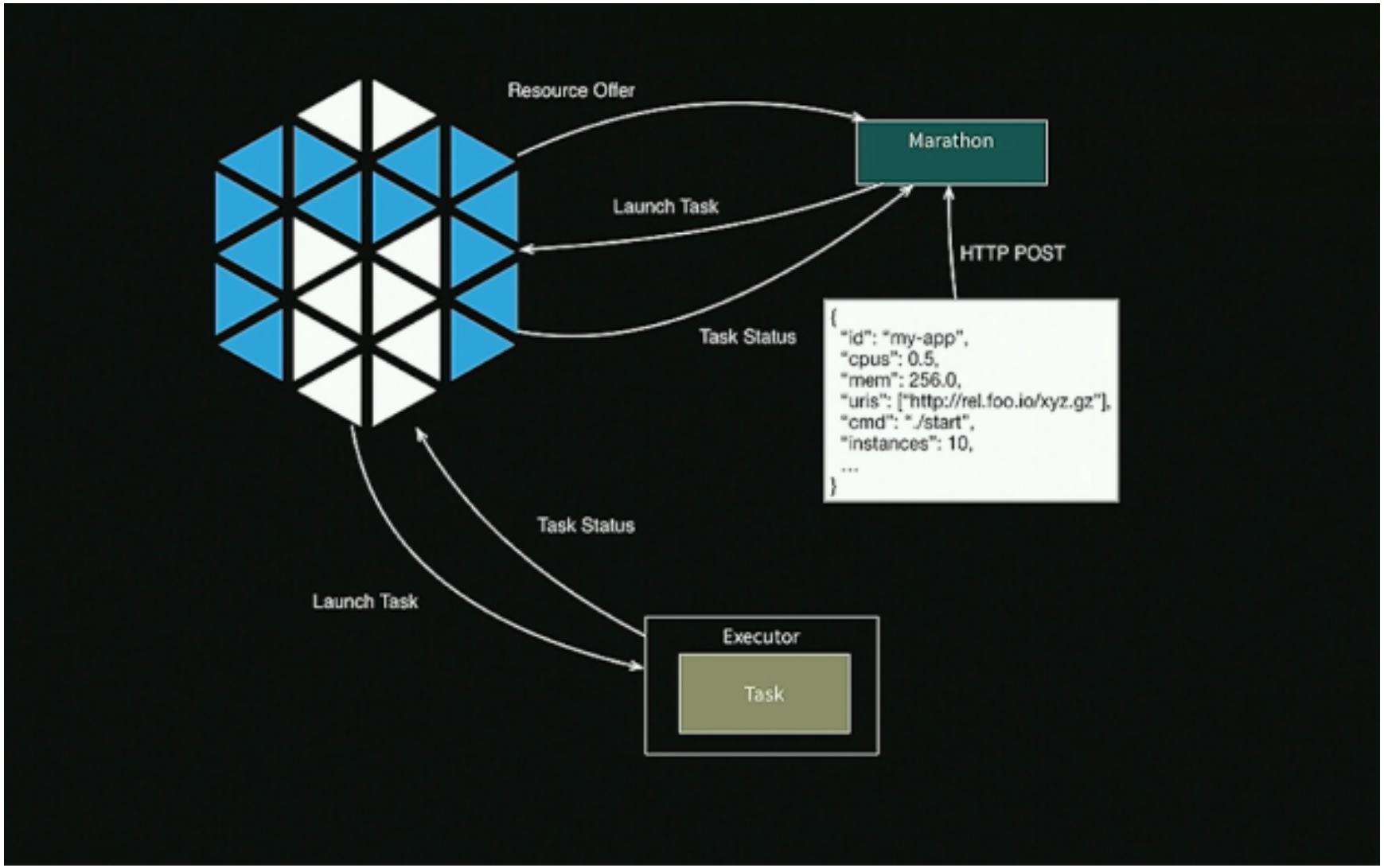
Marathon

- A generic mesos framework to run long running services (web apps, etc …)
- A distributed Init.d for the cluster
  - Service discovery
  - Runs any Linux binary without modification (Tomcat, Play, …)
  - Cluster wide process supervisor
- Private PaaS
  - Service discovery
  - Provide a self service Rest API for deployment
  - Authentication & SSL
  - Placement constraint (nodes, racks,…)
- Service discovery and load balancing via HAProxy

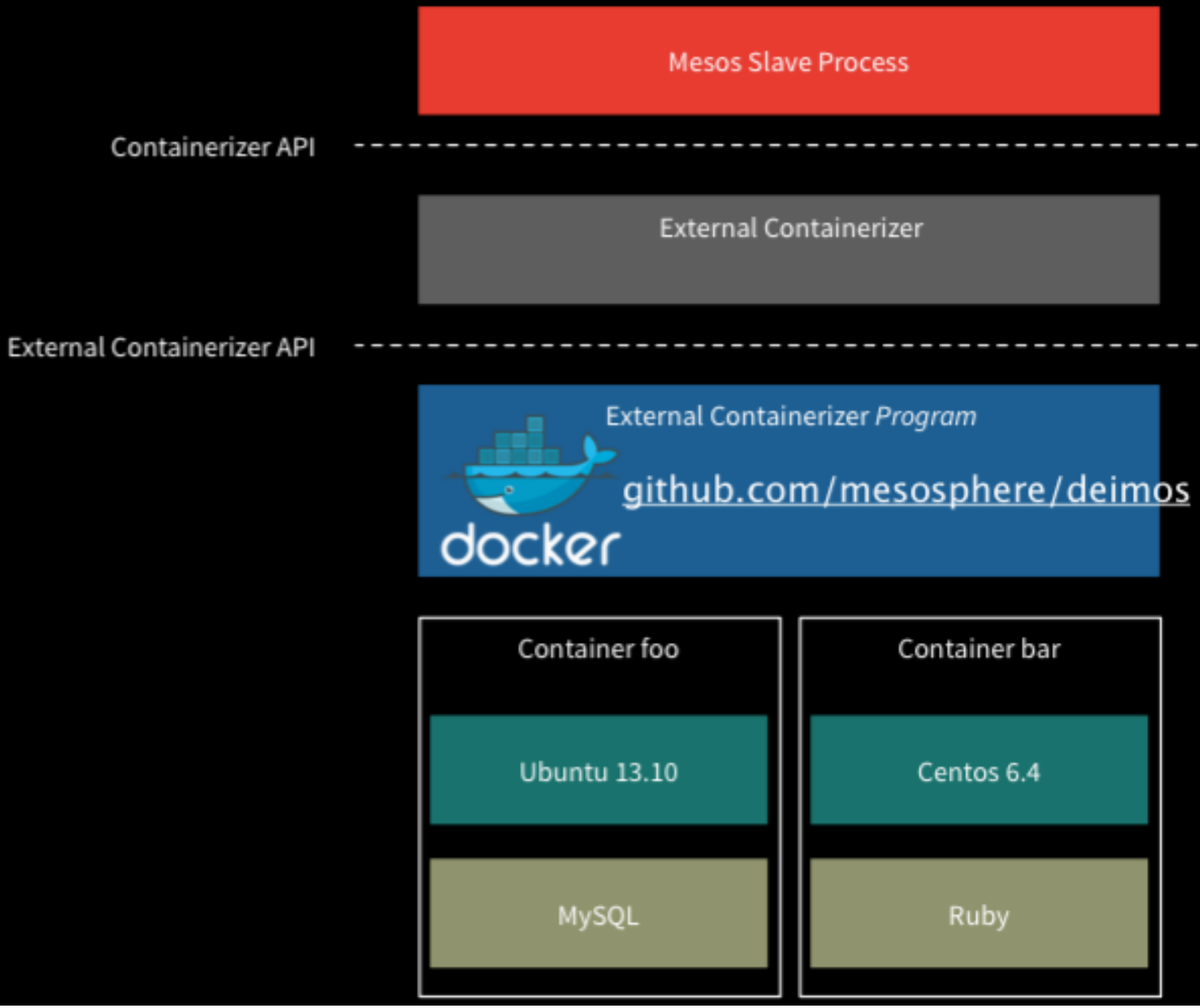
Marathon Concepts

- An app describes a task
- A task is an instance of an app
- Marathon creates tasks for apps

Interfaces


Resource Offer
Marathon
Launch Task
HTTP POST
Task Status
{
"id": "my-app",
"cpus": 0.5,
"mem": 256.0,
"uris": ["http://rel.foo.io/xyz.gz"],
"cmd": "./start",
"instances": 10,
...
}
Task Status
Launch Task
Executor
Task

Mesos provides pluggable resource isolation



Marathon API - Launching Dockers

- Starting with Mesos 0.19 containers are 1st class citizens
- Deimos is the Docker containerizer

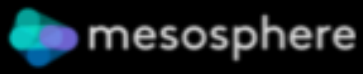
```
POST /v2/apps
{
    “id”: “Cassandra”,
    “container”: {
        “image”: “docker:///mesosphere/cassandra:2.0.6”,
        “options”: [“-v”, “/mnt:/mnt:rw”, “-e”,
“CLUSTER_NAME=prod”]
    }
}
```

mesosphere

Marathon API - Scaling Apps

- Just tell Marathon how many you want!

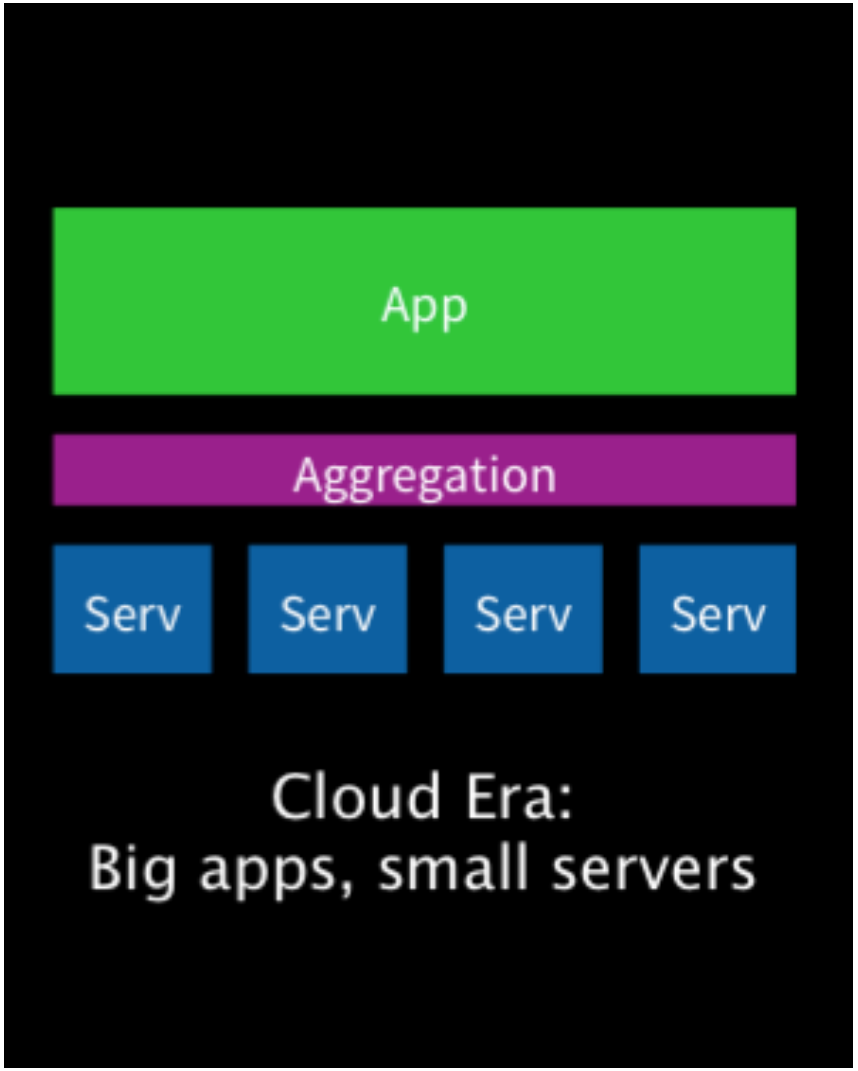
```
PATCH /v2/apps/Play
{
    “instances”: 4
}
```



Aggregation

- Mesosphere aggregates resources, makes a data center look like one big computer

- Mesosphere runs on top of a VM or on bare metal

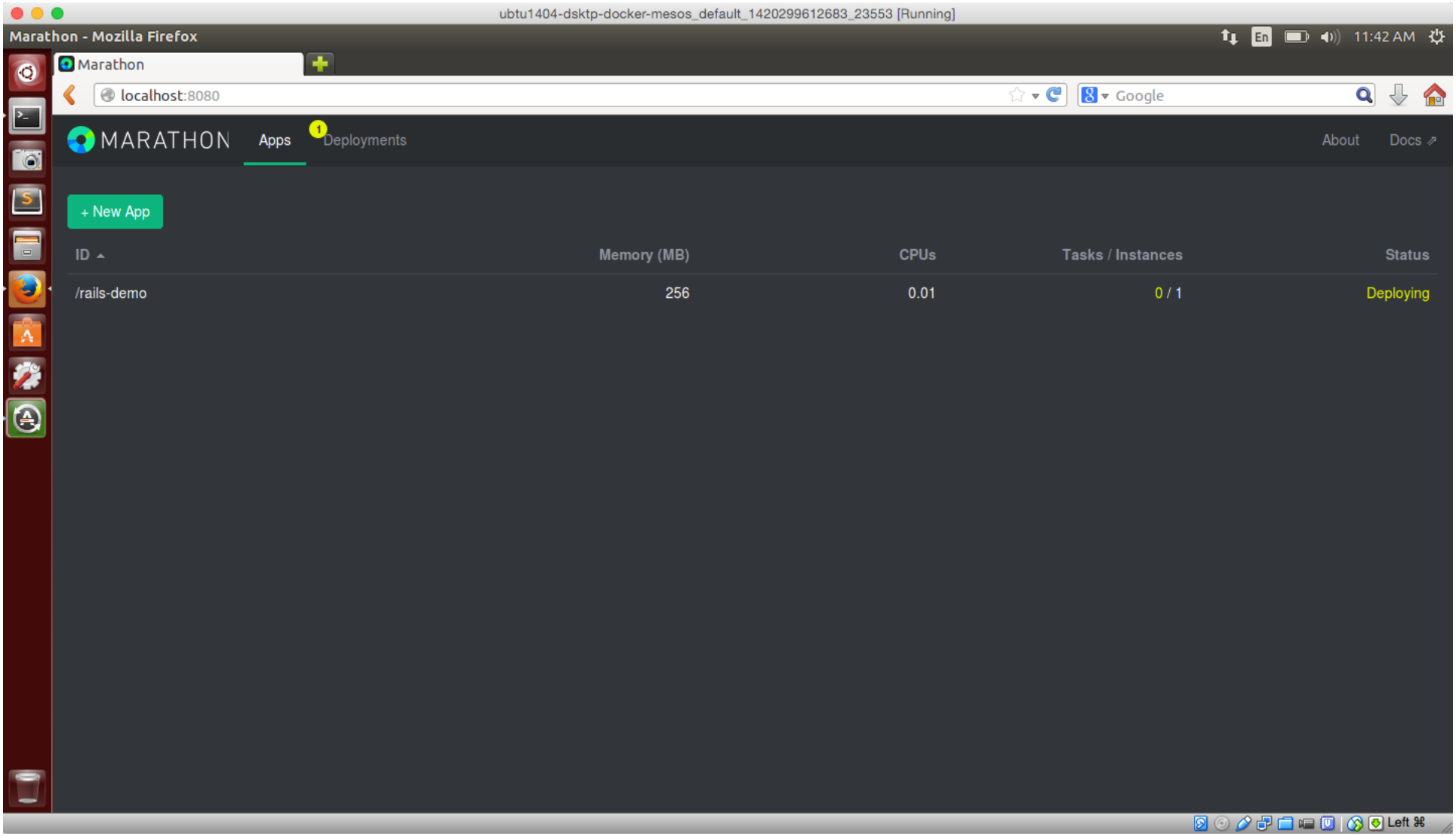


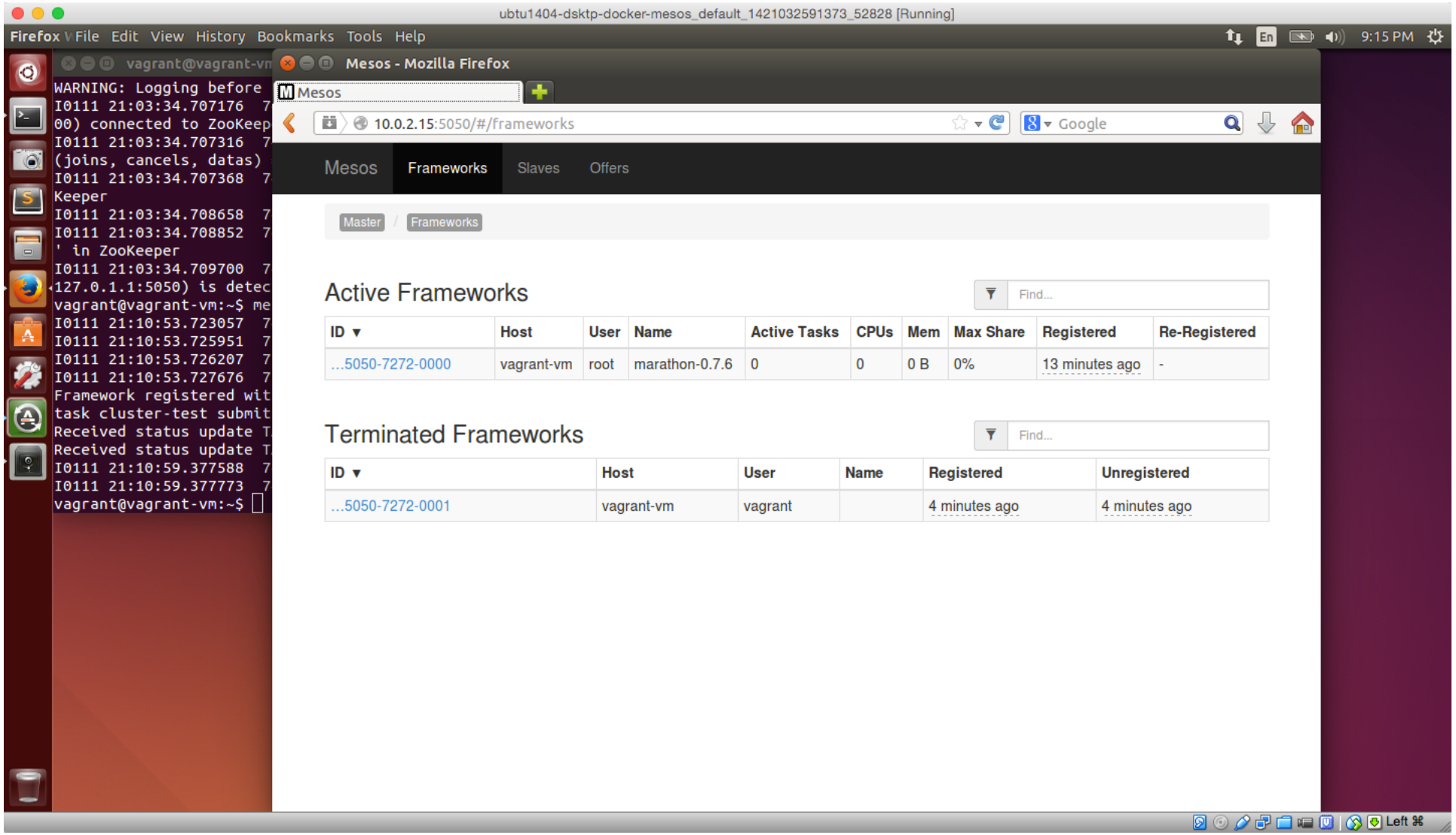
Conclusion

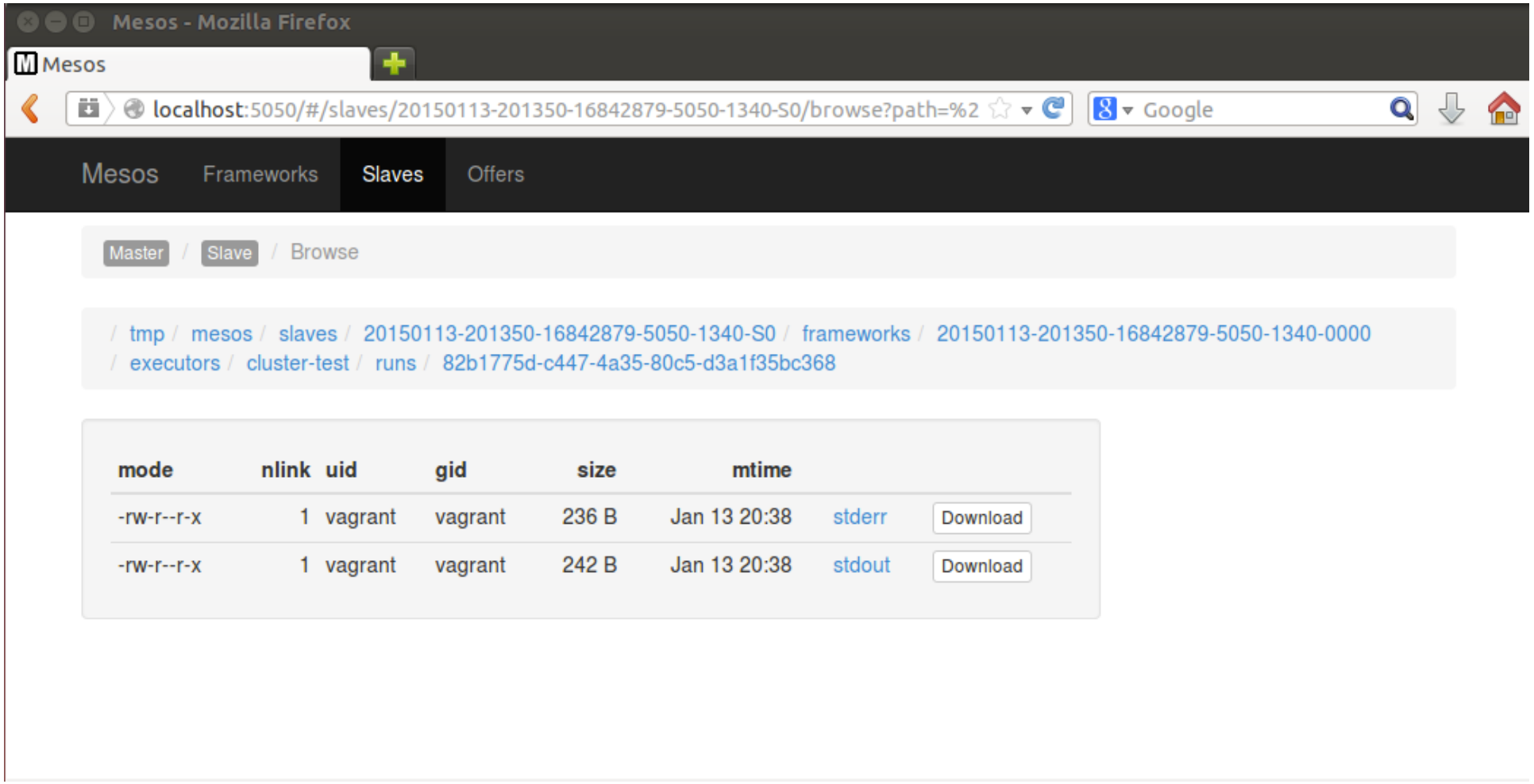
- Mesos shares clusters efficiently among diverse frameworks:
  - **Fine-grained sharing** at the level of tasks
  - **Resource offers**, a scalable mechanism for application-controlled scheduling
- Two levels resource allocation.
- Allows co-existence of current frameworks and development of new version.
- Runs on top of bare metal and VM.
- Supports containers.

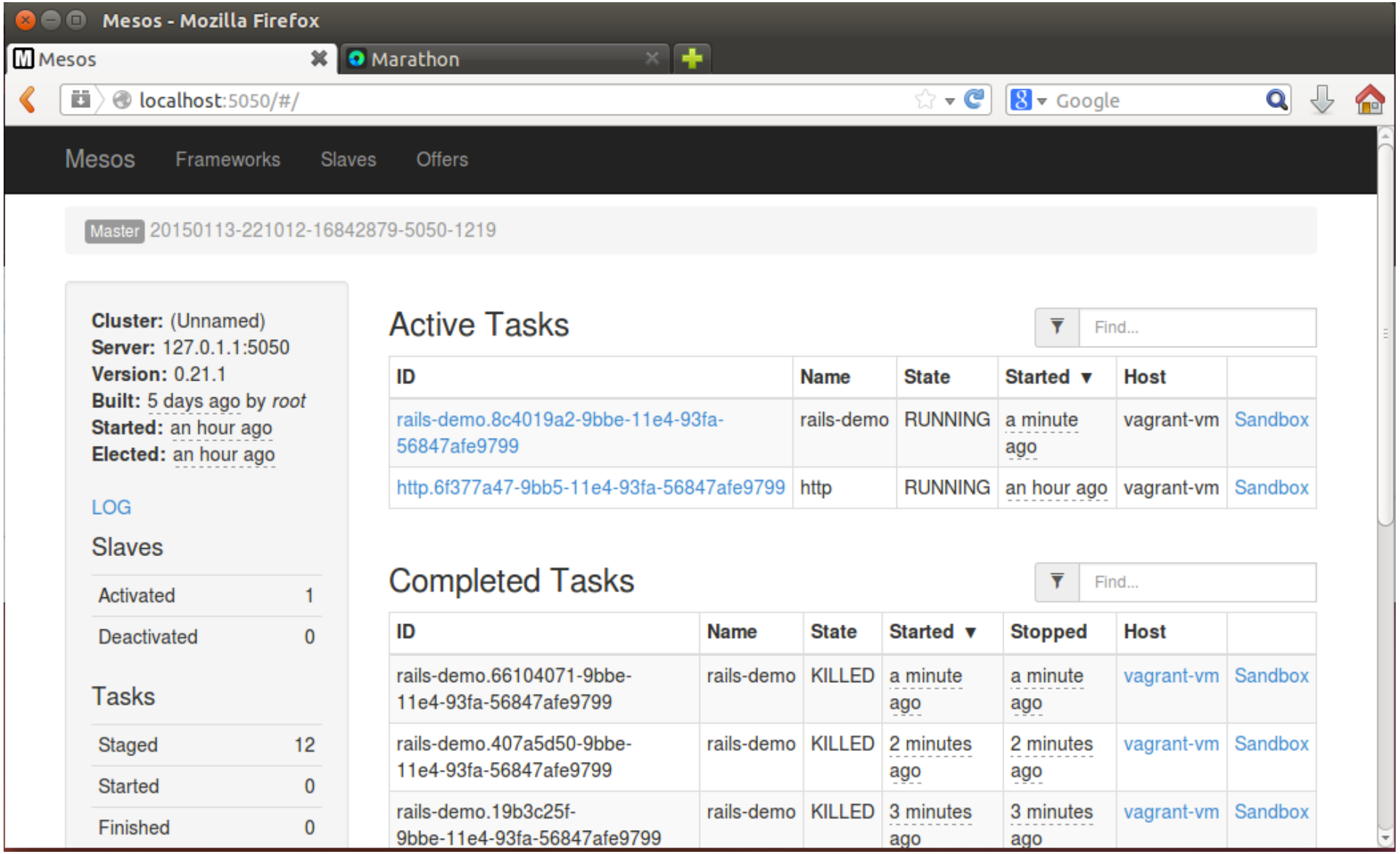
DEMO

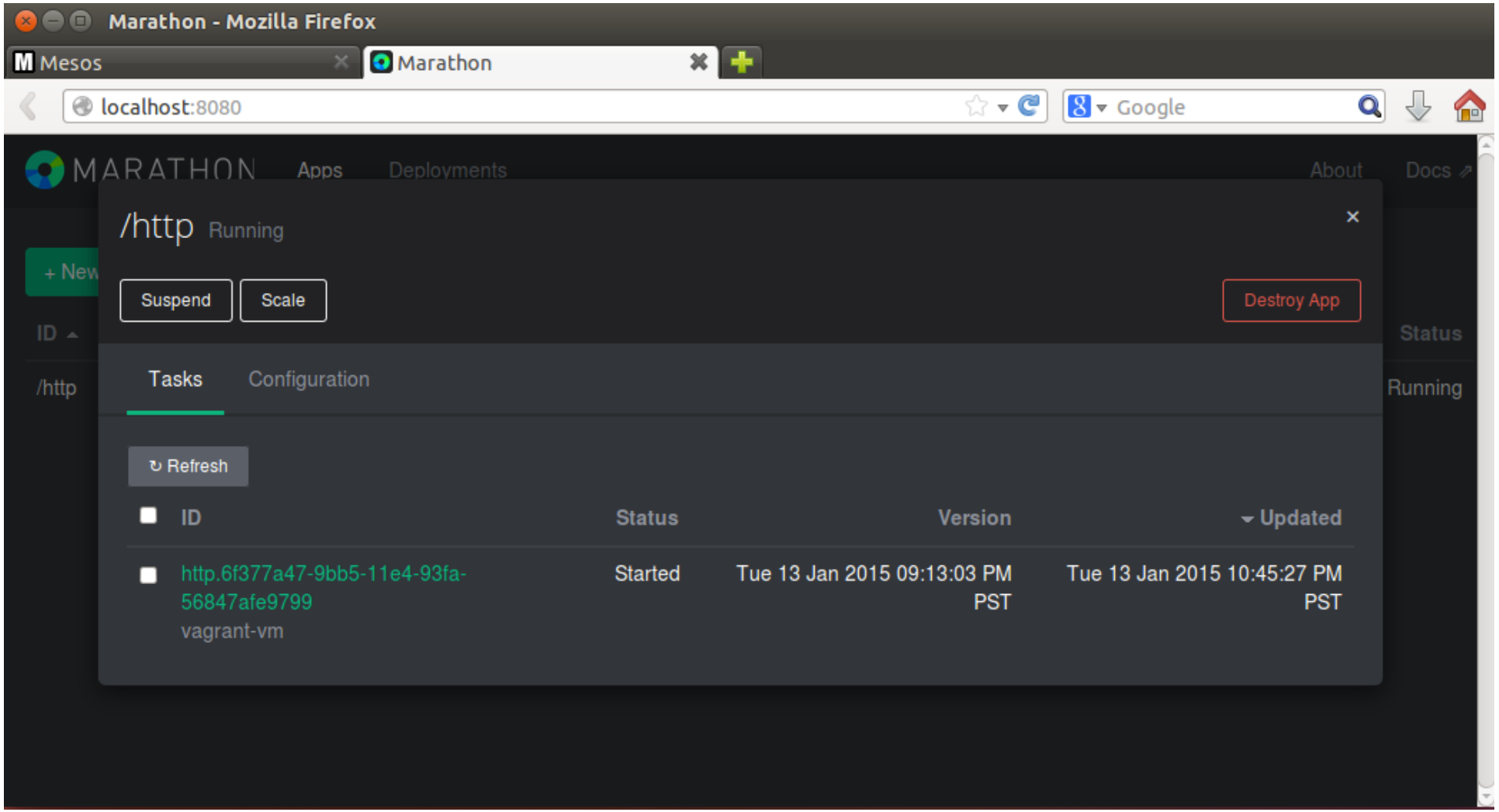
Backup demo screen shots Richard's study notes - learning mesos

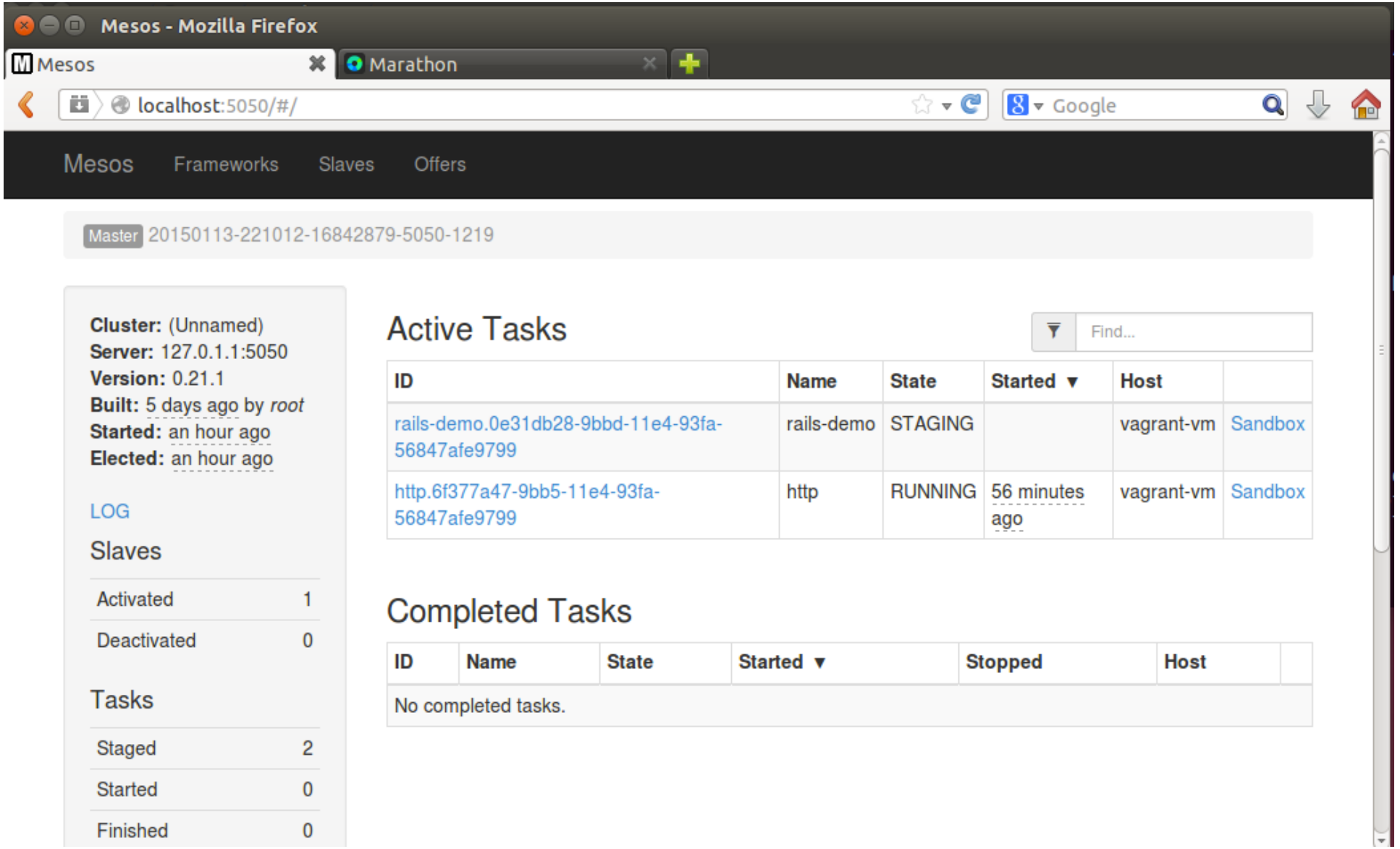

ubtu1404-dsktp-docker-mesos_default_1420299612683_23553 [Running]
Marathon - Mozilla Firefox
11:42 AM
Marathon
localhost:8080
Google
MARATHON
Apps
1
Deployments
About
Docs
+ New App
ID
Memory (MB)
CPUs
Tasks / Instances
Status
/rails-demo
256
0.01
0 / 1
Deploying
Left ⌘

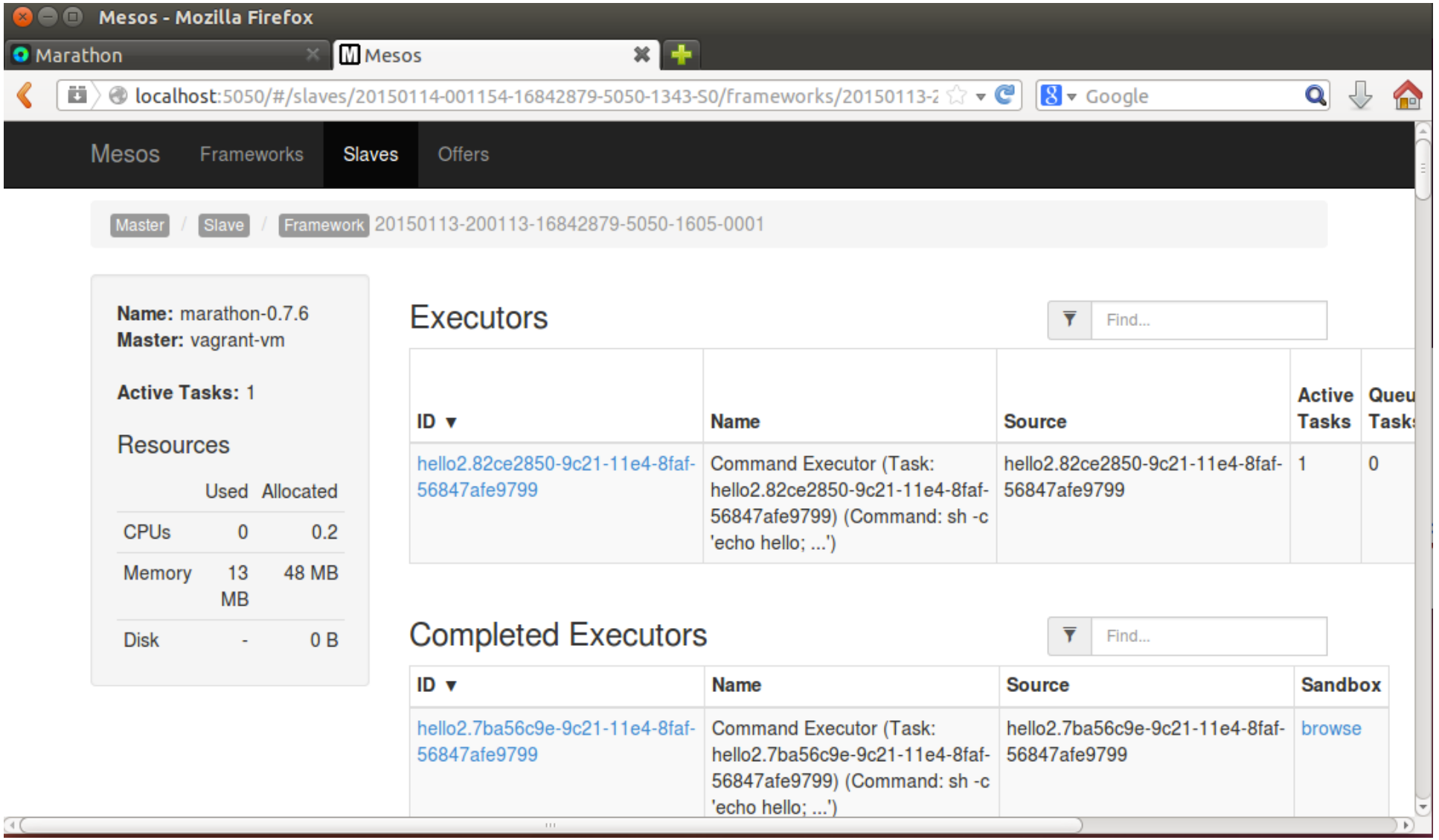

ubtu1404-dsktp-docker-mesos_default_1421032591373_52828 [Running]
Firefox File Edit View History Bookmarks Tools Help
9:15 PM
vagrant@vagrant-vm
WARNING: Logging before
I0111 21:03:34.707176 7
00) connected to ZooKeep
I0111 21:03:34.707316 7
(joins, cancels, datas)
I0111 21:03:34.707368 7
Keeper
I0111 21:03:34.708658 7
I0111 21:03:34.708852 7
' in ZooKeeper
I0111 21:03:34.709700 7
127.0.1.1:5050) is detec
vagrant@vagrant-vm:~$ me
I0111 21:10:53.723057 7
I0111 21:10:53.725951 7
I0111 21:10:53.726207 7
I0111 21:10:53.727676 7
Framework registered wit
task cluster-test submit
Received status update T
Received status update T
I0111 21:10:59.377588 7
I0111 21:10:59.377773 7
vagrant@vagrant-vm:~$
Mesos - Mozilla Firefox
Mesos
10.0.2.15:5050/#/frameworks
Google
Mesos
Frameworks
Slaves
Offers
Master
Frameworks
Active Frameworks
Find...
ID
Host
User
Name
Active Tasks
CPUs
Mem
Max Share
Registered
Re-Registered
...5050-7272-0000
vagrant-vm
root
marathon-0.7.6
0
0
0 B
0%
13 minutes ago
-
Terminated Frameworks
Find...
ID
Host
User
Name
Registered
Unregistered
...5050-7272-0001
vagrant-vm
vagrant
4 minutes ago
4 minutes ago
Left ⌘


Mesos - Mozilla Firefox
Mesos
localhost:5050/#/slaves/20150113-201350-16842879-5050-1340-S0/browse?path=%2
Google
Mesos
Frameworks
Slaves
Offers
Master
Slave
Browse
/ tmp / mesos / slaves / 20150113-201350-16842879-5050-1340-S0 / frameworks / 20150113-201350-16842879-5050-1340-0000 / executors / cluster-test / runs / 82b1775d-c447-4a35-80c5-d3a1f35bc368
mode nlink uid gid size mtime
-rw-r--r-x 1 vagrant vagrant 236 B Jan 13 20:38 stderr Download
-rw-r--r-x 1 vagrant vagrant 242 B Jan 13 20:38 stdout Download


Mesos - Mozilla Firefox
Mesos
Marathon
localhost:5050/#/
Google
Mesos
Frameworks
Slaves
Offers
Master 20150113-221012-16842879-5050-1219
Cluster: (Unnamed)
Server: 127.0.1.1:5050
Version: 0.21.1
Built: 5 days ago by root
Started: an hour ago
Elected: an hour ago
LOG
Slaves
Activated 1
Deactivated 0
Tasks
Staged 12
Started 0
Finished 0
Active Tasks
Find...
ID | Name | State | Started | Host
rails-demo.8c4019a2-9bbe-11e4-93fa-56847afe9799 | rails-demo | RUNNING | a minute ago | vagrant-vm | Sandbox
http.6f377a47-9bb5-11e4-93fa-56847afe9799 | http | RUNNING | an hour ago | vagrant-vm | Sandbox
Completed Tasks
Find...
ID | Name | State | Started | Stopped | Host
rails-demo.66104071-9bbe-11e4-93fa-56847afe9799 | rails-demo | KILLED | a minute ago | a minute ago | vagrant-vm | Sandbox
rails-demo.407a5d50-9bbe-11e4-93fa-56847afe9799 | rails-demo | KILLED | 2 minutes ago | 2 minutes ago | vagrant-vm | Sandbox
rails-demo.19b3c25f-9bbe-11e4-93fa-56847afe9799 | rails-demo | KILLED | 3 minutes ago | 3 minutes ago | vagrant-vm | Sandbox


Marathon - Mozilla Firefox
Mesos
Marathon
localhost:8080
Google
MARATHON
Apps
Deployments
About
Docs
/http Running
Suspend
Scale
Destroy App
Tasks
Configuration
Refresh
ID
Status
Version
Updated
http.6f377a47-9bb5-11e4-93fa-56847afe9799
vagrant-vm
Started
Tue 13 Jan 2015 09:13:03 PM PST
Tue 13 Jan 2015 10:45:27 PM PST


Mesos - Mozilla Firefox
Mesos
Marathon
localhost:5050/#/
Google
Mesos
Frameworks
Slaves
Offers
Master 20150113-221012-16842879-5050-1219
Cluster: (Unnamed)
Server: 127.0.1.1:5050
Version: 0.21.1
Built: 5 days ago by root
Started: an hour ago
Elected: an hour ago
LOG
Slaves
Activated 1
Deactivated 0
Tasks
Staged 2
Started 0
Finished 0
Active Tasks
Find...
ID Name State Started ▼ Host
rails-demo.0e31db28-9bbd-11e4-93fa-56847afe9799 rails-demo STAGING vagrant-vm Sandbox
http.6f377a47-9bb5-11e4-93fa-56847afe9799 http RUNNING 56 minutes ago vagrant-vm Sandbox
Completed Tasks
ID Name State Started ▼ Stopped Host
No completed tasks.


Mesos - Mozilla Firefox
Marathon
Mesos
localhost:5050/#/slaves/20150114-001154-16842879-5050-1343-S0/frameworks/20150113-2
Google
Mesos
Frameworks
Slaves
Offers
Master / Slave / Framework 20150113-200113-16842879-5050-1605-0001
Name: marathon-0.7.6
Master: vagrant-vm
Active Tasks: 1
Resources
Used Allocated
CPUs 0 0.2
Memory 13 MB 48 MB
Disk - 0 B
Executors
Find...
ID ▼ Name Source Active Tasks Queu Task
hello2.82ce2850-9c21-11e4-8faf-56847afe9799 Command Executor (Task: hello2.82ce2850-9c21-11e4-8faf-56847afe9799) (Command: sh -c 'echo hello; ...') hello2.82ce2850-9c21-11e4-8faf-56847afe9799 1 0
Completed Executors
Find...
ID ▼ Name Source Sandbox
hello2.7ba56c9e-9c21-11e4-8faf-56847afe9799 Command Executor (Task: hello2.7ba56c9e-9c21-11e4-8faf-56847afe9799) (Command: sh -c 'echo hello; ...') hello2.7ba56c9e-9c21-11e4-8faf-56847afe9799 browse