Introduction to Mesos and Marathon

mlowicki@opera.com
medium.com/@mlowicki

Share commodity cluster between many computing frameworks

Clusters of commodity servers are major computing platforms

Polyglot Persistence

http://martinfowler.com/bliki/PolyglotPersistence.html

Options

- Statically partition cluster (one framework per partition)
- Set of VMs to each framework

Requirements

- Support different frameworks
- Scalability
- Fault-tolerant and highly available

Design philosophy

Define a minimal interface that enables efficient resource sharing across frameworks and otherwise push control of task scheduling and execution to the frameworks.

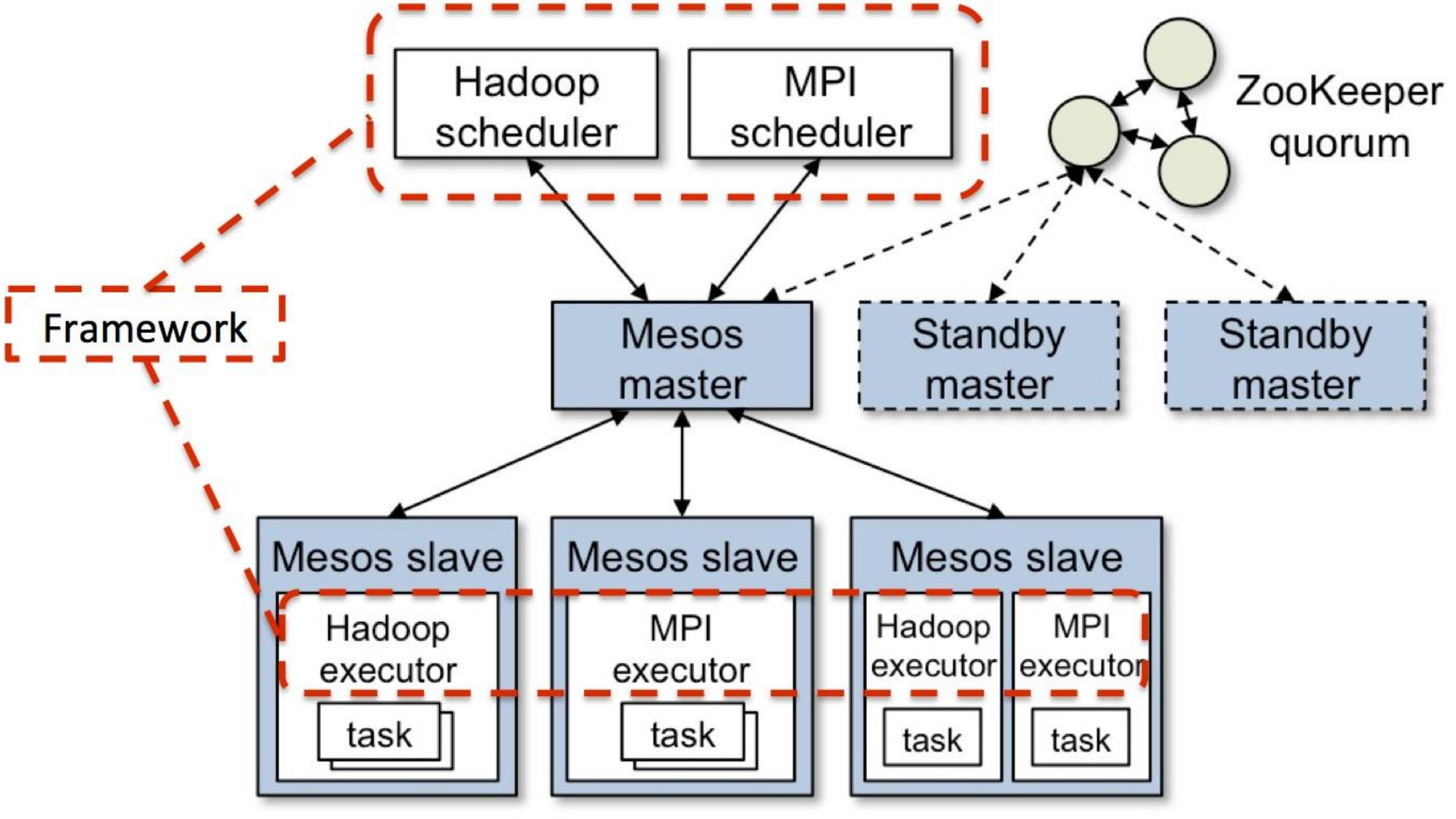
Centralized scheduler

- Complex
- Need for expressive API to capture all frameworks' requirements
- Burden load on core part
- Decentralized approach works well in practice

Many frameworks per cluster

- Multiple version for single framework (roll out upgrades)
- Separate production and experiments workloads


MESOS


Hadoop scheduler
MPI scheduler
ZooKeeper quorum
Framework
Mesos master
Standby master
Standby master
Mesos slave
Mesos slave
Mesos slave
Hadoop executor
MPI executor
Hadoop executor
MPI executor
task
task
task
task

Two-level scheduling

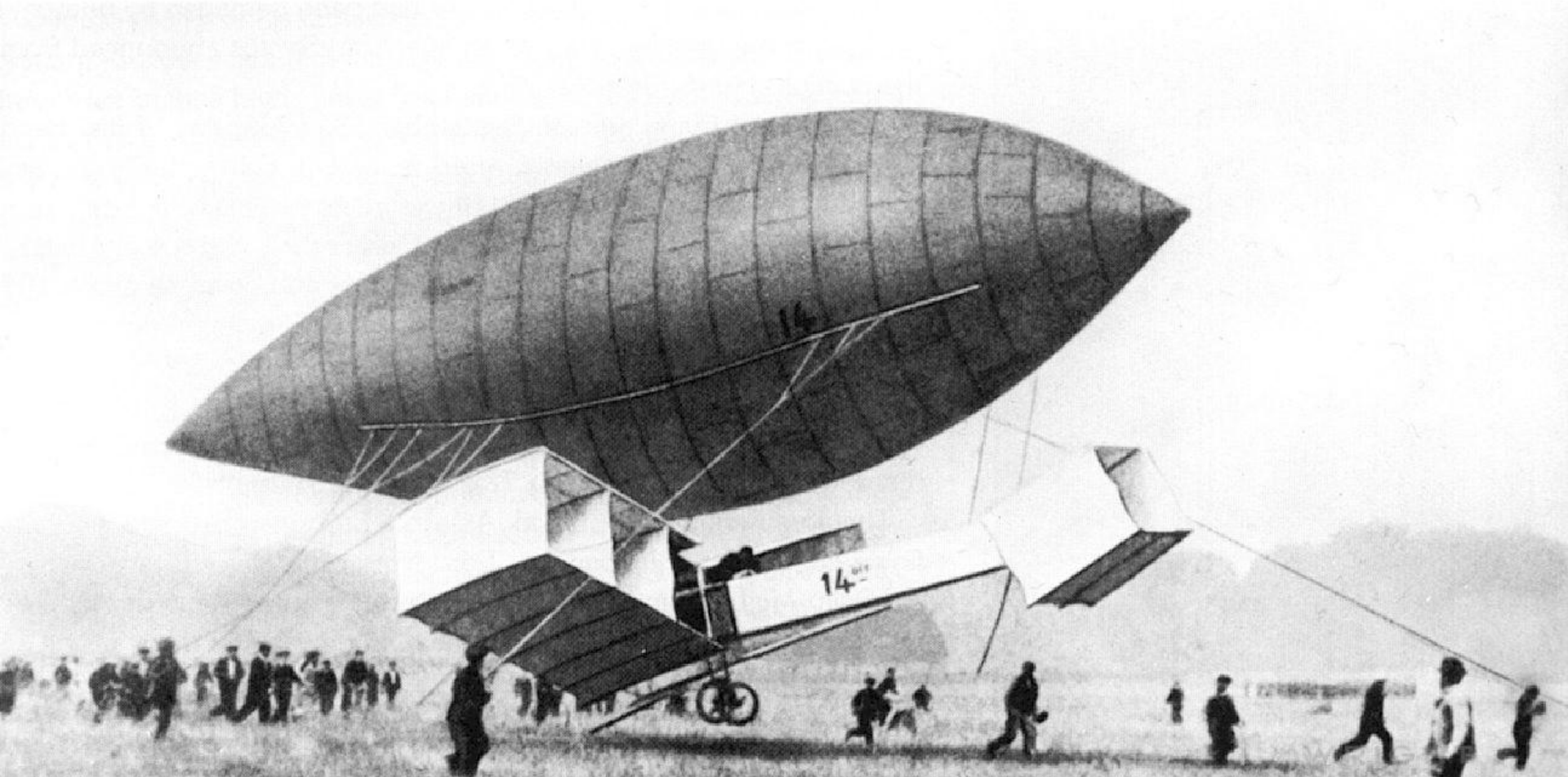
- Master decides how many resources to offer each framework (org policy)
- Framework decides which resource offers to accept (run computations on offered resources)

Fault tolerance

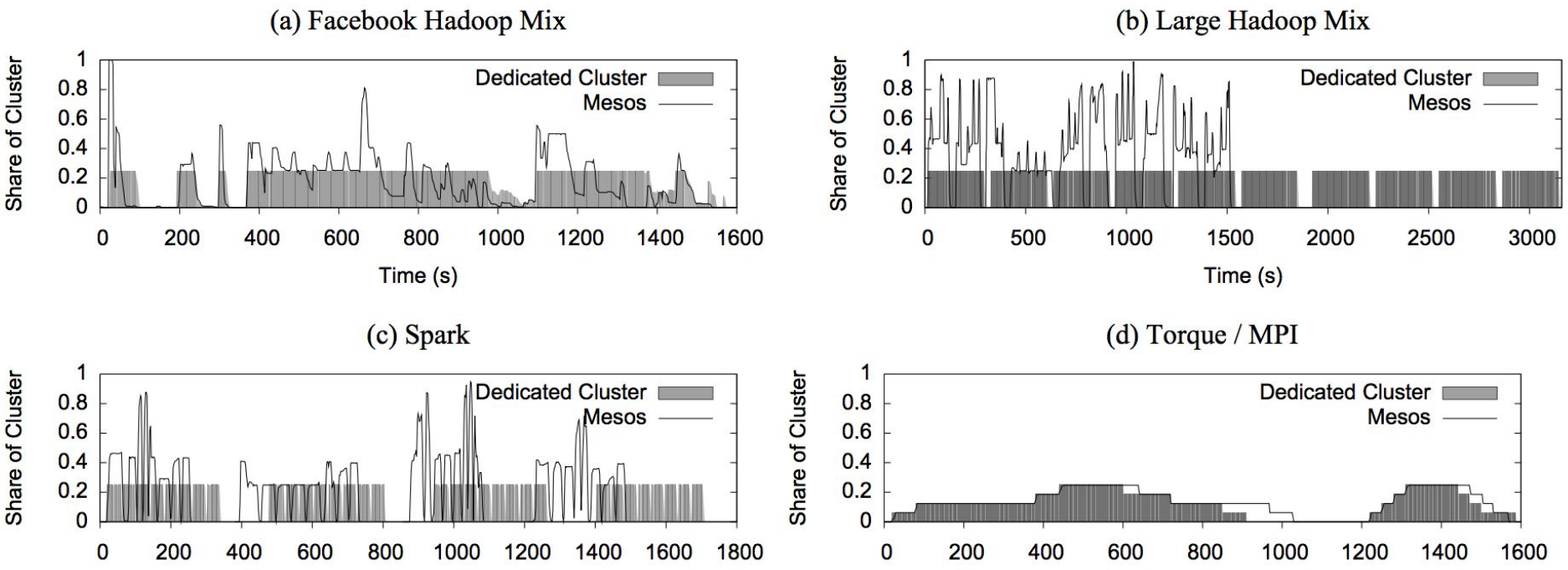
- soft state
- hot-standby (ZooKeeper is used to select new master)

Pluggable

- allocation modules (fair sharing, priorities, Dominant Resource Fairness, ...)
- isolation modules


14
14

Proven scalability up to 50k nodes


(a) Facebook Hadoop Mix
Share of Cluster
Time (s)
Dedicated Cluster
Mesos
0
0.2
0.4
0.6
0.8
1
0
200
400
600
800
1000
1200
1400
1600
(b) Large Hadoop Mix
Share of Cluster
Time (s)
Dedicated Cluster
Mesos
0
0.2
0.4
0.6
0.8
1
0
500
1000
1500
2000
2500
3000
(c) Spark
Share of Cluster
Dedicated Cluster
Mesos
0
0.2
0.4
0.6
0.8
1
0
200
400
600
800
1000
1200
1400
1600
1800
(d) Torque / MPI
Share of Cluster
Dedicated Cluster
Mesos
0
0.2
0.4
0.6
0.8
1
0
200
400
600
800
1000
1200
1400
1600

Powered by Mesos

- Twitter
- Apple (Siri)
- Netflix
- Uber
- and more

MARATHON

A cluster-wide init and control system for services in cgroups or Docker containers

Marathon REST

```
curl -X POST https://m3.mesos.services.ams.osa:8080/v2/apps
-d @app.json -H "Content-type: application/json"
```

```
{
  "id": "hello",
  "cmd": "python3 -m http.server 8080",
  "cpus": 0.5,
  "mem": 32.0,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "python:3",
      "network": "BRIDGE",
      "portMappings": [
        { "containerPort": 8080, "hostPort": 0, "servicePort": 8001 }
      ]
    }
  },
  "healthChecks": [
    {
      "path": "/",
      "portIndex": 0,
      "protocol": "HTTP",
      "gracePeriodSeconds": 10,
      "intervalSeconds": 10,
      "timeoutSeconds": 20,
      "maxConsecutiveFailures": 3,
      "ignoreHttp1xx": false
    }
  ],
  "upgradeStrategy": {
    "minimumHealthCapacity": 0.5
  }
}
```

```
curl -X PUT
https://m3.mesos.services.ams.osa:8080/v2/apps/hello -d
'{"instances": 5}' -H "Content-type: application/json"
```

```
curl -X PUT
https://m3.mesos.services.ams.osa:8080/v2/apps/hello -d
@app.json -H "Content-type: application/json"
```

Health checks

Types:

- TCP
- HTTP
- COMMAND

https://mesosphere.github.io/marathon/docs/health-checks.html

marathon-lb*

Issues before introducing Mesos

- Configuration drifts
- Underutilized nodes
- Stability / performance issues with OpenStack

Pets vs cattle

Service Model

Borrowed from
@randybias at Cloudscaling
http://www.slideshare.net/randybias/the-cloud-revolution-cyber-press-forum-philippines

- Pets are given names like pussinboots.cern.ch
- They are unique, lovingly hand raised and cared for
- When they get ill, you nurse them back to health

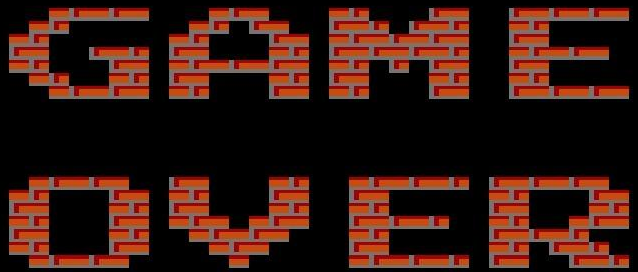
- Cattle are given numbers like vm0042.cern.ch
- They are almost identical to other cattle
- When they get ill, you get another one

- Future application architectures should use Cattle but Pets with strong configuration management are viable and still needed

http://www.slideshare.net/gmccance/cern-data-centre-evolution

Resources

- Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center
- Return of the Borg: How Twitter Rebuilt Google's Secret Weapon


GAME
OVER