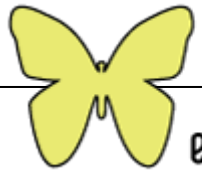
Making Distributed Data Persistent Services Elastic (Without Losing All Your Data)



elodina.net

Joe Stein

CEO of Elodina, Inc. Elodina http://www.elodina.net/ is a startup focusing on the support & maintenance of third party open source software (like Mesos frameworks) and offering SaaS based solutions for those systems. Elodina started as Big Data Open Source Security http://stealth.ly and has been working for the last couple of years on implementing and assisting organizations with their Kafka, Mesos, Hadoop, Cassandra, Accumulo, Storm, Spark, etc, Big Data systems.

Twitter: https://twitter.com/allthingshadoop
LinkedIn: https://www.linkedin.com/in/charmalloc

Overview

- **Mesos Overview**
- **Roles, Attributes, Constraints, Modules, Hooks**
- **Marathon, Kafka, Cassandra, HDFS, MySQL, YARN and more!**

Apache Mesos Overview

Papers

- Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center
  https://www.cs.berkeley.edu/~alig/papers/mesos.pdf

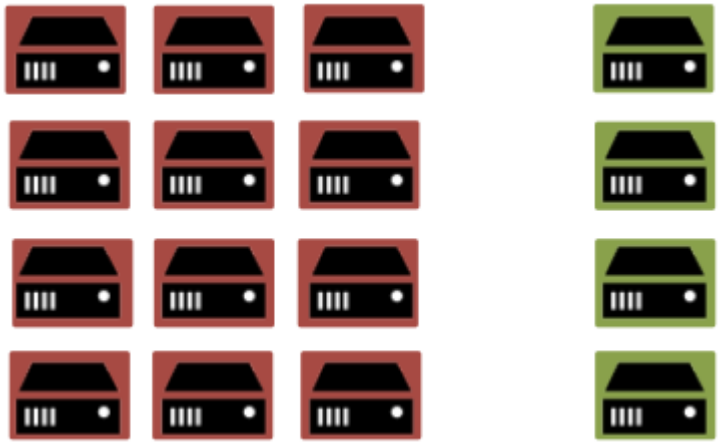
- Omega: flexible, scalable schedulers for large compute clusters
  http://eurosys2013.tudos.org/wp-content/uploads/2013/paper/Schwarzkopf.pdf

- Large-scale cluster management at Google with Borg
  http://research.google.com/pubs/pub43438.html

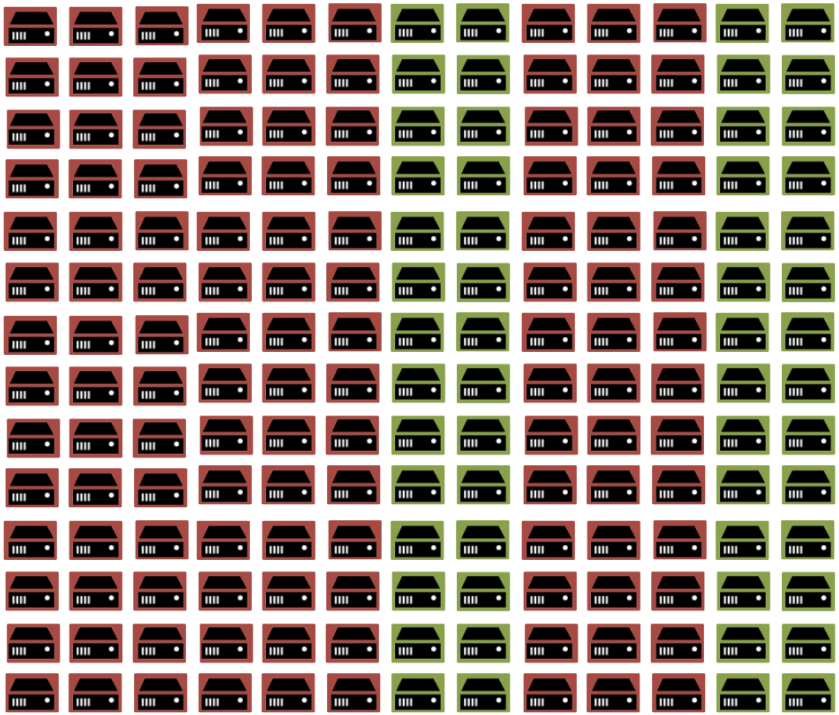
Static partitioning

Static partitioning

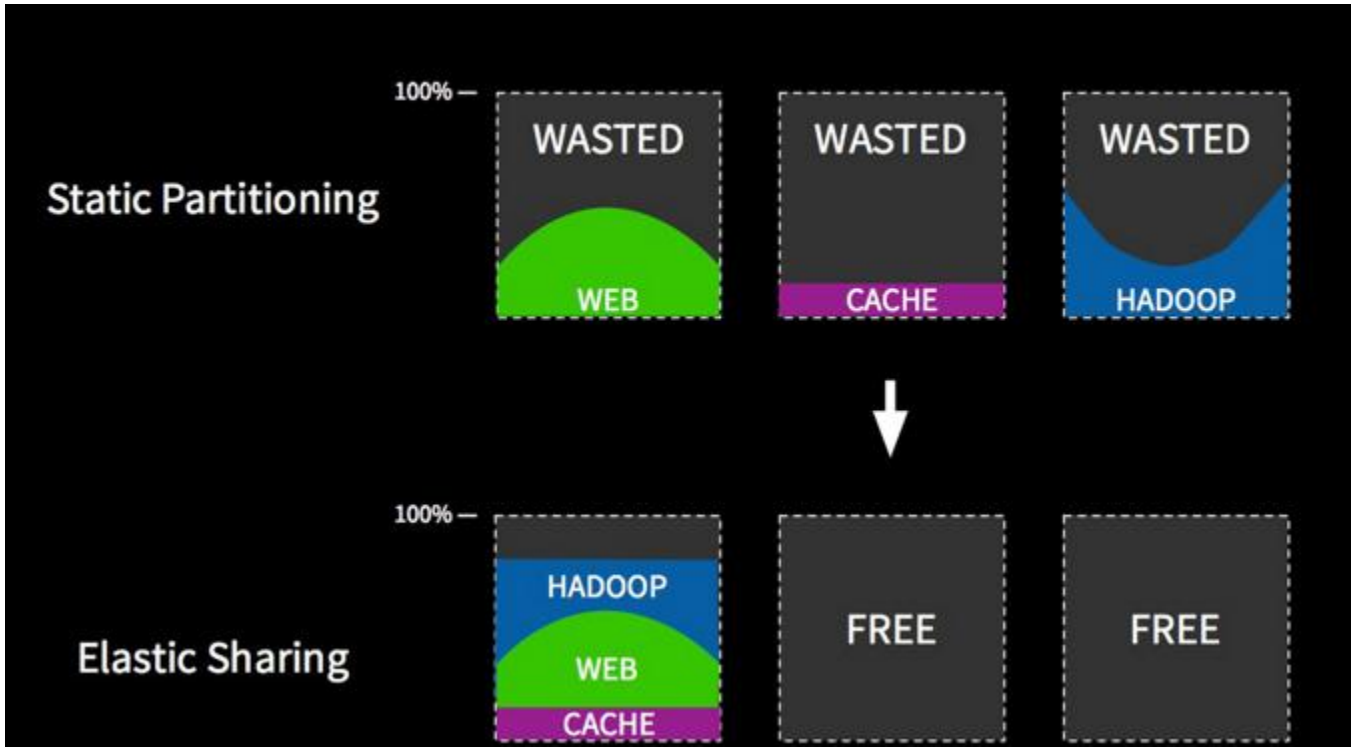
Static partitioning



Static partitioning



Better option



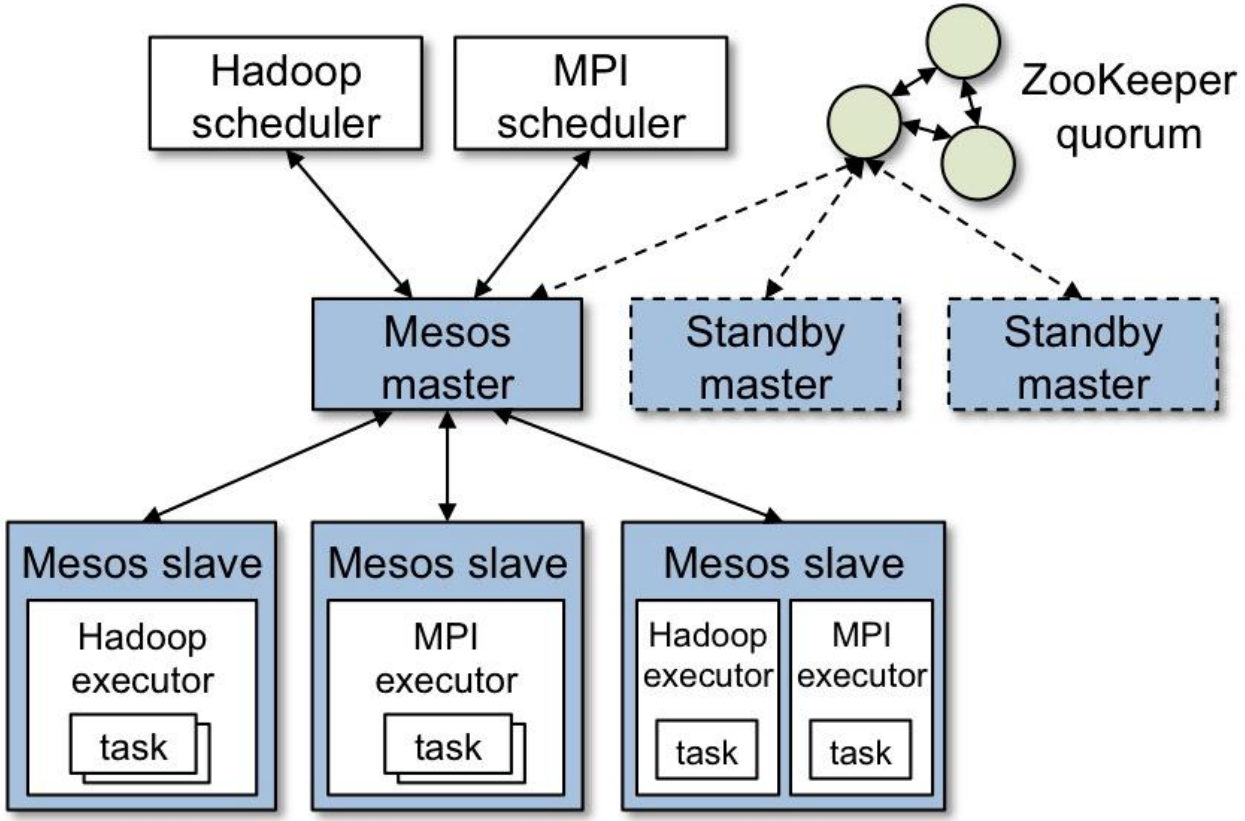
Data Center Operating System

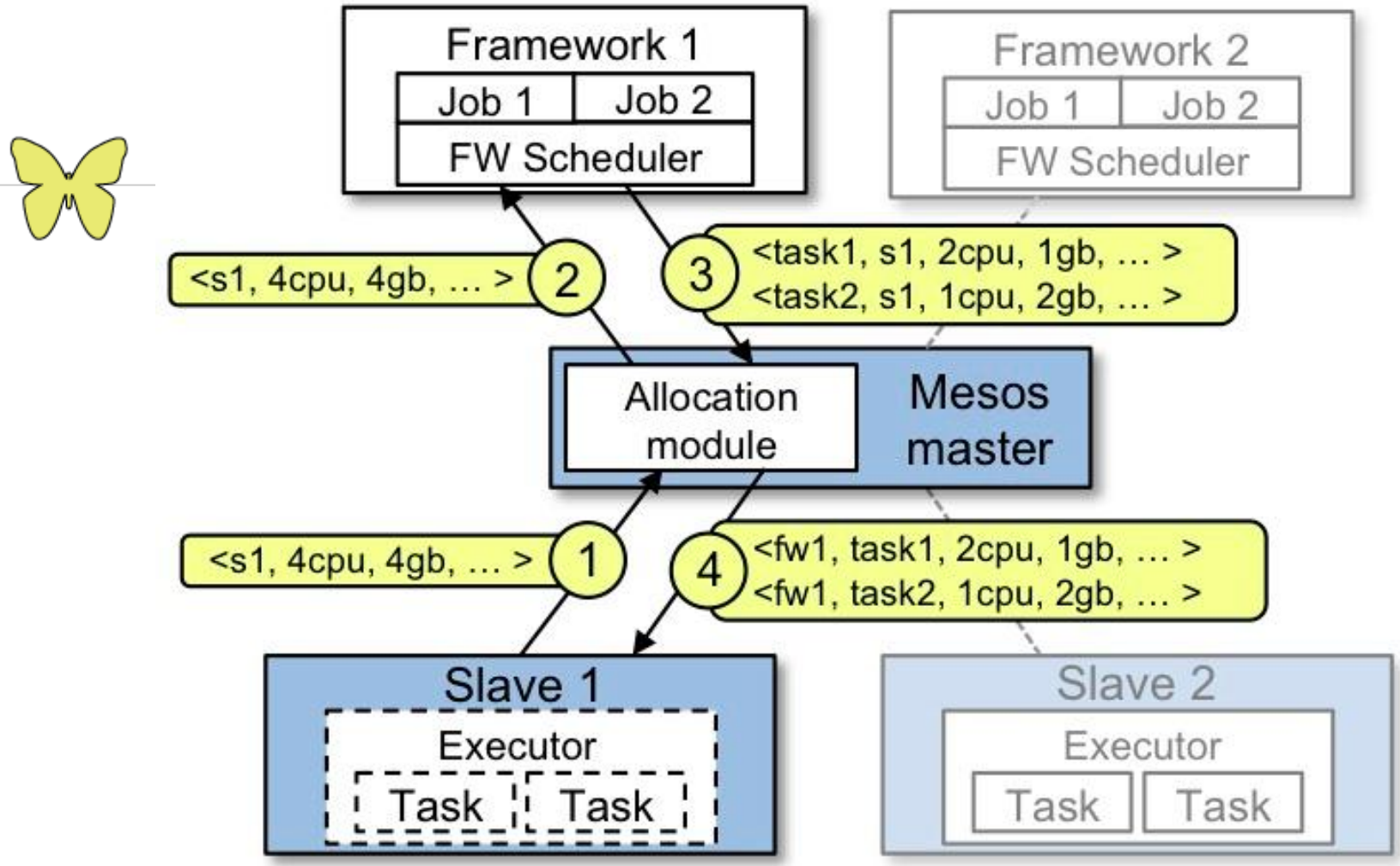
Mesos

- Scalability to 10,000s of nodes
- Fault-tolerant replicated master and slaves using ZooKeeper
- Support for Docker containers
- Native isolation between tasks with Linux Containers
- Multi-resource scheduling (memory, CPU, disk, and ports)
- Java, Python and C++ APIs for developing new parallel applications
- Web UI for viewing cluster state


MESOS
ALL THE THINGS
memegenerator.net

Mesos


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


Framework 1
Job 1
Job 2
FW Scheduler
Framework 2
Job 1
Job 2
FW Scheduler
<s1, 4cpu, 4gb, … >
2
3
<task1, s1, 2cpu, 1gb, … >
<task2, s1, 1cpu, 2gb, … >
Allocation module
Mesos master
<s1, 4cpu, 4gb, … >
1
4
<fw1, task1, 2cpu, 1gb, … >
<fw1, task2, 1cpu, 2gb, … >
Slave 1
Executor
Task
Task
Slave 2
Executor
Task
Task

Sample Frameworks

C++ - https://github.com/apache/mesos/tree/master/src/examples

Java - https://github.com/apache/mesos/tree/master/src/examples/java

Python - https://github.com/apache/mesos/tree/master/src/examples/python

Scala - https://github.com/mesosphere/scala-sbt-mesos-framework.g8

Go - https://github.com/mesosphere/mesos-go

Roles, Attributes, Constraints, Modules, Hooks

Roles

Total consumable resources per slave, in the form 'name(role):value;name(role):value...'. This value can be set to limit resources per role, or to overstate the number of resources that are available to the slave.

```
--resources="cpus(*):8; mem(*):15360; disk(*):710534; ports(*):[31000-32000]"
--resources="cpus(prod):8; cpus(stage):2 mem(*):15360; disk(*):710534; ports(*):[31000-32000]"
```

All * roles will be detected, so you can specify only the resources that are not all roles (*).

```
--resources="cpus(prod):8; cpus(stage)"
```

Frameworks bind a specific roles or any. A default roll (instead of *) can also be configured.

Roles can be used to isolate and segregate frameworks.

Attributes

The Mesos system has two basic methods to describe the slaves that comprise a cluster. One of these is managed by the Mesos master, the other is simply passed onwards to the frameworks using the cluster.

```
--attributes='disks:sata;raid:jbod;dc:1;rack:3'
```

Constraints

Constraints control where apps run to allow optimizing for fault tolerance or locality. Constraints are made up of three parts: a field name, an operator, and an optional parameter. The field can be the slave hostname or any Mesos slave attribute.

- UNIQUE
- CLUSTER
- GROUP_BY
- LIKE
- UNLIKE

Modules & Hooks

- Plugable Allocator
- Plugable Isolator
- Subverting environment variables
- Decoration of label task

Docs http://mesos.apache.org/documentation/latest/modules/

Sample Code https://github.com/mesos/modules

Future release(s) to make things even better!

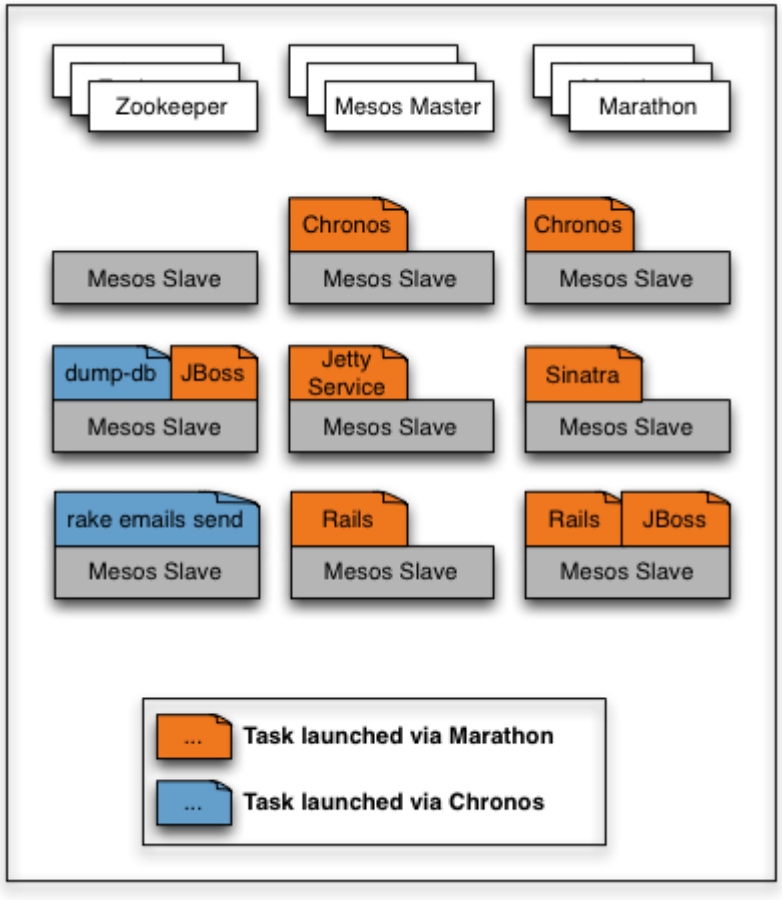
MESOS-2018 Dynamic Reservations
MESOS-1554 Persistent resources support for storage-like services
MESOS-1279 Add resize task primitive
MESOS-1607 Optimistic Offers

Marathon

Marathon

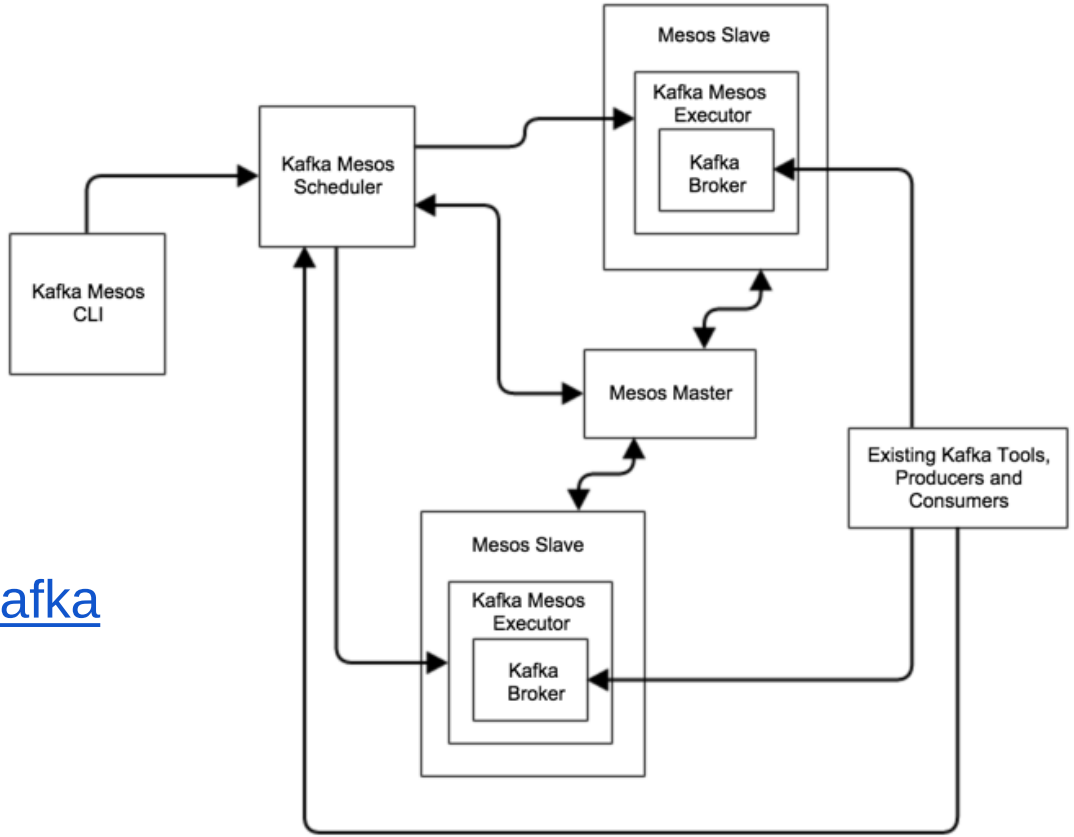
https://github.com/mesosphere/marathon
Cluster-wide init and control system for services in cgroups or docker based on Apache Mesos

- REST API
- Supports Constraints
- Discovery
- Health checks
- Deployments



Apache Kafka with Apache Mesos

mesos/kafka



https://github.com/mesos/kafka

Goals we set out with

- smart broker.id assignment.
- preservation of broker placement (through constraints and/or new features).
- ability to-do configuration changes.
- rolling restarts (for things like configuration changes).
- scaling the cluster up and down with automatic, programmatic and manual options.
- smart partition assignment via constraints visa vi roles, resources and attributes.

Scheduler & Executor

Scheduler

- Provides the operational automation for a Kafka Cluster.
- Manages the changes to the broker's configuration.
- Exposes a REST API for the CLI to use or any other client.
- Runs on Marathon for high availability.

Executor

- The executor interacts with the kafka broker as an intermediary to the scheduler

CLI & REST API

- scheduler - starts the scheduler.
- add - adds one more more brokers to the cluster.
- update - changes resources, constraints or broker properties one or more brokers.
- remove - take a broker out of the cluster.
- start - starts a broker up.
- stop - this can either a graceful shutdown or will force kill it (./kafka-mesos.sh help stop)
- rebalance - allows you to rebalance a cluster either by selecting the brokers or topics to rebalance. Manual assignment is still possible using the Apache Kafka project tools. Rebalance can also change the replication factor on a topic.
- help - ./kafka-mesos.sh help || ./kafka-mesos.sh help {command}

Launch 20 brokers in seconds

./kafka-mesos.sh add 1000..1019 --cpus 0.01 --heap 128 --mem 256 --options num.io.threads=1

./kafka-mesos.sh start 1000..1019

Mesos Frameworks Slaves Offers

Master 20150501-230424-84414636-5050-26136

Cluster: (Unnamed)
Server: 172.16.8.5:5050
Version: 0.21.1
Built: 4 months ago by *root*
Started: 53 minutes ago
Elected: 53 minutes ago

LOG

Slaves

Activated	1
Deactivated	0

Tasks

Staged	20
Started	0
Finished	5
Killed	0
Failed	0
Lost	1

Resources

Active Tasks

Find...

ID	Name ▲	State	Started	Host	
broker-1000-cdccbdb2-73b5-466b-9834-e5d439f88ce3	broker-1000	RUNNING	3 minutes ago	172.16.8.20	Sandbox
broker-1001-029d82fc-3ad5-436d-9180-3c205f3cac7f	broker-1001	RUNNING	3 minutes ago	172.16.8.20	Sandbox
broker-1002-76072b68-43f1-4a68-97ea-82a86ef70300	broker-1002	RUNNING	3 minutes ago	172.16.8.20	Sandbox
broker-1003-d356de34-82f7-4012-a492-d39b5dbf44f8	broker-1003	RUNNING	3 minutes ago	172.16.8.20	Sandbox
broker-1004-19b4de5d-9463-4ccf-a43c-d633403297e9	broker-1004	RUNNING	3 minutes ago	172.16.8.20	Sandbox
broker-1005-7da42bba-f4b6-4b5e-b046-895bed089a28	broker-1005	RUNNING	3 minutes ago	172.16.8.20	Sandbox
broker-1006-3586bf59-0791-48b0-9c59-78c1f690a39d	broker-1006	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1007-31b363de-3a97-47e7-8ab4-6f092a663dad	broker-1007	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1008-d5897c2b-1e9c-48bf-b7af-fff6f4e3bc1c	broker-1008	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1009-e59510da-6b07-4af4-b9f8-7bc992f8b38e	broker-1009	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1010-9850c3ce-57ae-477c-ac5f-e87e94b70840	broker-1010	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1011-3665ac48-4db4-4d9f-85cb-a1b73033a00a	broker-1011	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1012-cbeef2cd-1db7-4ce9-adf5-d7610db7fbf0	broker-1012	RUNNING	2 minutes ago	172.16.8.20	Sandbox
broker-1013-60499bbf-3032-4118-aa2e-c3c85041ef73	broker-1013	RUNNING	2 minutes ago	172.16.8.20	Sandbox

Mesosphere DCOS

Kafka is available on DCOS
https://docs.mesosphere.com/services/kafka/

Apache Cassandra with Apache Mesos

Cassandra on Mesos

https://github.com/mesosphere/cassandra-mesos

The Mesos scheduler is the component with the most high-level intelligence in the framework. It will need to possess the ability to bootstrap a ring and distribute the correct configuration to all subsequently started nodes. The Scheduler will also be responsible for orchestrating all tasks with regard to restarting nodes and triggering and monitoring periodic administrative tasks required by a node.

Cassandra Scheduler

- Bootstrapping a ring
- Adding nodes to a ring
- Restarting a node that has crashed
- Providing configuration to nodes
  - Seed nodes, Snitch Class, JVM OPTS
- Scheduling and running administrative utilities
  - nodetool repair
  - nodetool cleanup
- Registers with a failover timeout

- Supports framework authentication
- Declines offers to resources it doesn't need
- Only use necessary fraction of offers
- Deal with lost tasks
- Does not rely on in-memory state
- Verifies supported Mesos Version
- Supports roles
- Able to provide set of ports to be used by Nodes
- Initial implementation will be for a static set of ports with a potential for longer term dynamic port usage.

Cassandra Executor

- Monitor health of running node
- Use JMX Mbeans for interfacing with Cassandra Server Process
- Communicate results of administrative actions via StatusUpdates to scheduler when necessary
- Does not rely on file system state outside sandbox
- Pure libprocess communication with Scheduler leveraging StatusUpdate
- Does not rely on running on a particular slave node
- Data directories will be created and managed by Mesos leveraging the features provided in MESOS-1554

Apache HDFS with Apache Mesos

HDFS on Mesos

https://github.com/mesosphere/hdfs

- 3 journal nodes
- 2 name nodes (active/standby)
- data nodes, lots of them!

- Fault tolerance more than just what Hadoop gives.
- Ease of configuration and distributing nodes.
- Elastic DFS
- Run multiple frameworks at a time for new solutions

MySQL with Apache Mesos

MySQL on Mesos (Apache Incubating)

- Open sourced by Twitter https://github.com/twitter/mysos
- Moving to Apache https://twitter.com/ApacheMysos
- Dramatically simplifies the management of a MySQL cluster:
  - Efficient hardware utilization through multi-tenancy (in performance-isolated containers)
  - High reliability through preserving the MySQL state during failure and automatic backing up to/restoring from HDFS
  - An automated self-service option for bringing up new MySQL clusters
  - High availability through automatic MySQL master failover
  - An elastic solution that allows users to easily scale up and down a MySQL cluster by changing the number of slave instances

Questions?

Joe Stein
http://www.elodina.net