Resource Sharing Beyond Boundaries

Mohit Soni    Santosh Marella
A Mesos framework for scaling a YARN cluster (Incubating Into Apache)
https://issues.apache.org/jira/browse/MYRIAD — Edit

262 commits 8 branches 0 releases 15 contributors

Commit activity from 08/24 to 08/18:
Agenda

- What's up with Datacenters these days?
- Apache Mesos vs. Apache Hadoop/YARN?
- Why would you want/need both?
- Resource Sharing with Apache Myriad
What's running on your datacenter?

- Tier 1 services
- Tier 2 services
- High Priority Batch
- Best Effort, backfill
Requirements

- Programming models based on resources, not machines
- Custom resource types
- Custom scheduling algorithms: Fast vs. careful/slow
- Lightweight executors, fast task launch time
- Multi-tenancy, utilization, strong isolation
Hadoop and More

- Support Hadoop/BigData ecosystem
- Support arbitrary (legacy) processes/containers
- Connect Big Data to non-Hadoop apps, share data, resources
Mesos from 10,000 feet

- Open Source Apache project
- Cluster Resource Manager
- Scalable to 10,000s of nodes
- Fault-tolerant, no SPOF
- Multi-tenancy, Resource Isolation
- Improved resource utilization
Mesos is more than
Yet Another Resource Negotiator

- Long-running services; real-time jobs
- Native Docker; cgroups for years; Isolate cpu/mem/disk/net/other
- Distributed systems SDK; ~200 loc for a new app

Core written in C++ for performance, Apps in any language
Why two resource managers?
Static Partitioning sucks

- Hadoop teams fine with isolated clusters, but Ops team unhappy; slow to provision
- Resource silos, no elasticity
- Want to run Hadoop on the same infrastructure, without interrupting Tier-1 services
- Want multi-tenancy, resource sharing/isolation
Introducing Myriad
Myriad Overview

- Mesos Framework for Apache YARN
- Mesos manages DC, YARN manages Hadoop
- Coarse and fine grained resource sharing
Resource Sharing
yarn.resourcemanager.scheduler.class = MyriadFairScheduler
The diagram shows a system architecture with the following components:

- **Mesos Master**
- **Resource Manager**
- **Myriad**
- **HDFS**
- **Mesos Slave**

Interactions are indicated by arrows:

- **Myriad API**: `/api/flexup`, `/api/flexdown`, `/api/config`, `/api/state`

Connections between components are shown with arrows pointing from the Mesos Master to the HDFS and Mesos Slave nodes.
Myriad API
/api/flexup
{"profile":"medium", "instances": 1}
/api/flexup
{
"profile": "zero",
"instances": 1
}
Myriad improves Mesos

- Tighter integration with Hadoop frameworks like HBase, Hive, Pig
- Borrow resources from Hadoop when traffic spikes for tier-1 services
- Backfill unused resource capacity with best-effort Hadoop jobs
- No Mesos code changes necessary
Myriad improves Hadoop

- Elastic scaling
- Fault-tolerant: Maintain NM capacity
- Share resources with other workloads, improve resource utilization
- High SLA hadoop jobs unaffected
- No YARN/Hadoop code changes
Other Features

- RM failover/discovery using Marathon/Mesos-DNS
- Distribution of hadoop binaries
- Web Interface
- Myriad scheduler HA, task reconciliation (in progress)
- Ability to launch Job History Server (in progress)
- Your favorite feature here!
Learn More!
https://github.com/mesos/myriad
dev@myriad.incubator.apache.org
MYRIAD JIRA
Apache Myriad Incubator Proposal
Apache Myriad Incubator Status Page