

MyriadProposal

Myriad Proposal

---- ⚠️ **FINAL** ⚠️ This proposal is now complete and has been submitted for a VOTE. ----

Abstract

Myriad enables co-existence of Apache Hadoop YARN and Apache Mesos together on the same cluster and allows dynamic resource allocations across both Hadoop and other applications running on the same physical data center infrastructure.

Proposal

The vision of Myriad is to provide a comprehensive framework to ensure Apache Hadoop YARN and Apache Mesos can interoperate with minimal changes on either side and prevent the static fragmentation of data center resources.

Background

Project Myriad is the first resource management framework that allows big data developers to run YARN-based Hadoop jobs alongside other applications and services in production. ebay Inc., MapR, and Mesosphere jointly built Myriad (available on Github at <https://github.com/mesos/myriad>) with the vision of freeing big data jobs from siloed clusters and consolidating infrastructure into a single pool of resources for greater utilization and operational efficiency. Several companies including Twitter have expressed interest in Myriad and have begun testing it.

Rationale

Many Hadoop users are building larger clusters (data lake/data hub architectures) that support multiple workloads - made possible by the advent of Apache Hadoop YARN. As the clusters grow in size and importance, they become an important application within the broader datacenter. At the same time, Apache Mesos enables efficient resource isolation and sharing across distributed applications for the broader data center, for instance MPI, Spark, long running web services, build/test infrastructure, traditional linux applications/scripts, and others (including arbitrary docker images).

Myriad aims to enable co-existence of Apache Hadoop YARN and Apache Mesos on the same physical data center resources, reducing fragmentation of data center resources.

Project Goals

Initial Goals

- Run Myriad alongside Apache Hadoop YARN and Apache Mesos to allow policy based allocation of data center resources across Apache Hadoop and other distributed applications
- Ensure YARN based execution frameworks work without any changes when running alongside Myriad. YARN Applications will continue to interact and run on top of YARN and can choose to be unaware of Myriad.
- Ensure Mesos based execution frameworks work without any changes when running alongside Myriad. Mesos applications will continue to interact and run on Mesos and can choose to be unaware of Myriad.
- Provide isolation for multi-tenancy.
 - Use linux cgroups (and optionally Docker-like technologies to ease packaging, deployment and broader isolation) so that multiple YARN clusters can run in their own space and are isolated from each other. YARN's RM and NMs are dockerized.
- Myriad should be able to manage full YARN lifecycle:
 - Bring up YARN (RM, NM)
 - Scale Up/Down YARN
 - Release resources and shut down YARN

Longer Term Goals

- Allow fine-grained dynamic allocation of resources to Hadoop including the ability to scale up and scale down the cluster.
- Provide different policies to allow downsizing running applications on Hadoop when resources are taken away from it.
- Provide a framework so the downsizing policy is pluggable and users can write their own implementations.
- Allow multiple versions of Apache Hadoop to run on the same physical infrastructure
- Allow workload portability - ability to migrate YARN workloads across various cloud infrastructures seamlessly (e.g. GCE, AWS, etc)
- Authentication Requirements:
 - Support basic CRAM-MD5 password authentication between Myriad and Mesos. Additional authentication mechanisms may be supported in the future.
 - Traditional user authentication with Hadoop's HTTP web-consoles should work as usual.
- Authorization:
 - Only authorized users are allowed to launch YARN clusters. Mesos allows to specify which framework principal is allowed to register as a particular role.
- Encryption on wire:
 - All control traffic to/from Myriad/Mesos
- Logs
 - Audits: Log all major activities/events with audit trail - who, what, when, result (where to store them?)

- Launching YARN/RM
 - Launching NM's
 - Downsizing NM's
 - Terminating YARN/RM
 - What to do with old logs?
- Debuggability/Visibility
 - Hooks to identify different YARN cluster lifecycles (yarn-id?)
- GUI: Capability to scale-up and scale-down by selecting nodes and providing a scale-up/scale-down factor.

Architectural Overview

The following diagram illustrates the high level architecture. YARN (with Myriad) is registered as a framework with Mesos master along with possibly other Mesos frameworks. This enables YARN to share cluster resources with other Mesos frameworks providing elasticity of resources between Hadoop workloads and Mesos frameworks.

See <https://github.com/mesos/myriad/blob/phase1/docs/images/high-level-architecture.png>

Current Status

Myriad is under active development. Key components of Myriad are:

Myriad Resource Manager (RM) Plugin

- Plugs into Resource Manager Java process via yarn-site.xml configuration.
- Registers Myriad as a framework with Mesos. Receives resource offers from Mesos.
- Monitors YARN's application pipeline and scheduling events to drive scale-up or scale-down decisions for Hadoop.
- Exposes REST APIs to help admins control Hadoop/YARN's resource consumption. Currently the following APIs are supported:
- Scale Up (e.g. "launch 4 Node Manager instances with 10G/6CPU capacity")
- Scale Down (e.g. "kill 2 Node Manager instances with 10G/6CPU capacity")

Myriad Mesos Executor

- Launched on a Mesos slave node by Myriad RM plugin via Mesos.
- Responsible for launching Node Manager process with appropriate capacities configured in yarn-site.xml.
- Mounts YARN's cgroup hierarchy under Mesos' cgroup hierarchy in case YARN's cgroups are enabled.

Currently, a working prototype/demo had been built for the goals listed under the "Initial Goals" section. Open issues and enhancements are tracked at <https://github.com/mesos/myriad/issues>. Myriad is not yet tested for production use.

Meritocracy

We plan to invest in supporting a meritocracy. We will discuss the requirements in a public forum. Several companies have already expressed interest in this project, and we intend to invite developers to contribute and gain karma. We will encourage and monitor community participation so that privileges can be extended to those that contribute.

Community

We are happy to report that there are existing Apache committers and corporate users who are closely involved in the project already. We hope to extend the user and developer base further in the future and build a solid open source community around Myriad, growing the community and adding committers following the Apache Way.

Core Developers

The initial technology was built independently by ebay and MapR. ebay built the technology in consultation with Ben Hindman. MapR built a working prototype in tight consultation and mentorship with Mesosphere.

Alignment

The initial committers strongly believe that Apache Hadoop YARN and Apache Mesos will gain broad adoption and therefore a framework to allow for a co-existence of these frameworks that is transparent to applications written for YARN and Mesos will serve the needs of the broader community.

Known Risks

Inexperience with Open Source

Initial Myriad committers have varying levels of experience using and contributing to Open Source projects, however by working with our mentors and the Apache community we believe we will be able to conduct ourselves in accordance with Apache Incubator guidelines. The close relationship between the Myriad team and Apache Mesos and Apache Hadoop means there is an awareness of the incubation process and a willingness to embrace The Apache Way.

Homogenous Developers

There is already diversity in the core developer community as they are employed by three different and independent companies viz. ebay inc., MapR, and Mesosphere. However, there will continue to be an emphasis on increasing the diversity of the developer community.

Reliance on Salaried Developers

Currently, the core developers are paid to work on Myriad. However, once the project has a community built around it, we expect to get committers, contributors and community from outside the current participating organizations.

Relationships with Other Apache Products

Myriad implements interfaces from both Apache YARN and Apache Mesos, and requires both to be present so that Myriad can coordinate dynamic resource sharing between the two.

An Excessive Fascination with the Apache Brand

While we respect the reputation of the Apache brand and have no doubts that it will attract contributors and users, our interest is primarily to give Myriad a solid home as an open source project following an established development model. We have also given reasons in the Rationale and Alignment sections.

Documentation

Documentation is included in a docs directory of the repository (See <https://github.com/mesos/myriad/tree/phase1/docs>), and currently details how Myriad works, developing the project, auto-scaling a YARN cluster, the Myriad REST API, and more. We will improve docs at every revision drop.

Initial Source

The Myriad codebase has been posted on [GitHub](#) for review and licensed under an Apache v2 license.

<https://github.com/mesos/myriad>

Source and IP Submission Plan

During incubation, the codebase will be available at <https://github.com/apache/incubator-myriad/> and contributors will commit appropriate contribute license agreements.

External Dependencies

All Myriad dependencies have Apache compatible licenses.

Cryptography

Myriad doesn't use cryptography itself. Hadoop and Mesos projects, however, use standard API's and tools for SSH And SSL communication where necessary.

Required Resources

Mailing Lists

- myriad-private for private PMC conversations
- myriad-dev
- myriad-commits

Version Control

We prefer to use Git as our source control system: [git://git.apache.org/myriad](https://git.apache.org/myriad)

Issue Tracking

JIRA Myriad (MYRIAD)

Initial Committers

- Santosh Marella (smarella at mapr dot com)
- Mohit Soni (mohitsoni1989 at gmail dot com)

- Adam Bordelon (me at apache dot org) *
- Meghdoot Bhattacharya (mbhattacharya at paypal dot com)
- Anoop Dawar (anoopdawar at gmail dot com)
- Jim Scott (jim at 13ways dot com)
- Ken Sipe (kensipe at gmail dot com)

Affiliations

- Santosh Marella, MapR
- Mohit Soni, ebay Inc.
- Adam Bordelon, Mesosphere
- Meghdoot Bhattacharya, ebay Inc.
- Anoop Dawar, MapR
- Jim Scott, MapR
- Ken Sipe, Mesosphere

Sponsors

Champion (Proposal)

- Ben Hindman (benh at apache dot org)

Nominated Mentors

- Ben Hindman (benh at apache dot org) - Mesosphere
- Danese Cooper (danese at apache dot org) - ebay, Inc.
- Ted Dunning (tdunning at apache dot org) - MapR
- Luciano Resende (lresende at apache dot org)

Sponsoring Entity

- Apache Incubator