HoyaProposal

Hoya Proposal

Abstract

Hoya is an application to deploy and manage existing distributed applications in a YARN cluster.

The core concept is : *Hoya understands YARN so your application doesn't need to*.

Proposal

Hoya allows users to deploy distributed applications across a Hadoop cluster, using the YARN Resource Manager to allocate and distribute parts of an application across the cluster.. Hoya monitors the health of these deployed containers, and react to their failure by creating new instances of them. It supports dynamic updates to cluster size, and "freezing" and "thawing" the application.

In this way, Hoya can take a classic distributed application ~~such as Apache HBase and Apache Accumulo~~ and make it a dynamic YARN application.

Background

Hoya was developed at Hortonworks to support the deployment of HBase clusters in YARN. It was done both to showcase this possibility, and to drive YARN development towards the needs of long-lived applications.

It has evolved to provide an extension model: providers. These allow Hoya to support different applications - it now supports Apache Accumulo, and can easily support other suitable applications

Hoya is now capable of running long-lived, dynamic applications in YARN clusters. It is already being used in internal Proof of Concept applications and in testing the applications and YARN itself; other people and organisations are experimenting with it and providing feedback.

Rationale

Hadoop YARN's cluster manager makes it possible to convert static, one-per-node, cluster-wide services, into dynamic, user-specific applications, where each project can run their own version. Resource allocation is more flexible, failure and scalability improved, and new applications can be developed more easily.

Hoya's goal is to take existing Hadoop applications and host them inside a YARN cluster, giving them all these benefits - without rewriting the applications.

Initial Goals

1. Donate the Hoya source code and documentation to the Apache Software Foundation. 2. Setup and standardize the open governance of the Hoya project. 3. Build a user and developer community 4. Tie in better with HBase, Accumulo and other projects ~~both ASF and external~~ that can be deployed in a YARN cluster without any code changes. 5. Help migrate more distributed applications into YARN clusters - such as Apache HAMA.

Longer Term Options

There are some longer term possibilities that could improve Hoya

1. Implement a management API for managing Hoya applications by tools such as Apache Ambari. 2. Add a web UI for tracking and redirecting to the (changing) location of deployed services. 3. Provide a Java API to ease creation and manipulation of Hoya-deployed clusters by other programs. 4. Explore/Research sophisticated placement and failure tracking algorithms. Its precisely because this is a less mature product that we can experiment here. 5. Explore load-driven cluster sizing. 6. Explore leveraging Twill internally in Hoya.

Hoya is driving YARN service support via YARN-896. We intend to evolve features and get practical experience using them before merging them into the Hadoop codebase.

Current Status

Meritocracy

The core of the code was originally written by one person: Steve Loughran, who has long-standing experience in Apache projects, with colleagues with HBase experience (Ted Yu, Devaraj Das) and Accumulo (Billie Rinaldi, Josh Elser).

Community

We are happy to report that there are folks in Accumulo, HBase and some users outside Hortonworks 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 Hoya, growing the community and add committers following the Apache meritocracy model

Alignment

The project is completely aligned with Apache, from its build process up. It depends on Apache Hadoop, and it currently deploys HBase and Accumulo.

Hoya and Samza are driving the work of supporting long-lived services in YARN. While many of these relate to service longevity, there is also the challenge of having low-latency table lookups co-exist with CPU-and-IO intensive analytics workloads.

Relationship with Apache Twill

Twill is a library, a convenience library, that one can use to write YARN applications. Hoya aims to provide a general framework using which one can take existing applications (HBase/ & Accumulo to start with), and make them run well in a Yarn cluster, without intruding at all into their internals.

The key differentiators are

- **Long lived static applications:** the application's containers are expected to be relatively stable, with their termination being an unexpected event to which Hoya must react.
- **no application code-changes**: The only glue between the App and Hoya is a Hoya interface that the App needs to implement for it to be deployable/manageable by Hoya.

Twill and Hoya are therefore very different. One is a platform for new YARN applications, the other a YARN application to adapt existing applications to YARN.

Although one could argue that Hoya can be written using Twill libraries (which is something we should pursue as part of long/medium-term collaboration between the two projects), The goals of the two projects are different - Twill would continue to make YARN application developers' lives easier, while Hoya is a tool that could deploy distributed-frameworks easily in a YARN cluster, and be able to later do basic management . Things like dynamic patching of the application's configuration to run in the YARN cluster, failure detection reacting to failures, storing some state about applications to facilitate better application restart behavior in a YARN cluster, etc. would be in the purview of Hoya. Management frameworks could use Hoya as a tool to start/stop/shrink /expand an instance of an application.

Relationship with Apache Helix

Hoya shares some common goals with Apache Helix. Helix is more sophisticated and is designed to work standalone. Hoya is designed to work only in the context of a YARN cluster, and focuses on that YARN integration.

We have discussed Hoya with the Helix team, and feel that the work we are doing in YARN integration -and driving YARN changes, will be of direct benefit to Helix. We plan to collaborate on features which can be shared across both projects.

Relationship with Apache Accumulo and Apache HBase

We offer these projects the flexibility of operation in a YARN cluster. As such, it should expand the uses of the applications, and their user base.

There may be some changes that the applications can make to help them live more easily in a YARN cluster, and to be managed by Hoya. To date, all changes have been limited to supporting dynamic port allocations and some exit code changes -minor improvements of benefits to all.

It may be in future that we encounter situations where other changes to the applications can help them work even better in Hoya-managed deployments. If these arise we would hope to work with the relevant teams to get the changes adopted --knowing up front that neither of these project teams would countenance any changes that interfered with classic static application deployments.

The initial Hoya committer list includes committers for both Accumulo and HBase, who can maintain cross-project collaboration.

Known Risks

The biggest risk is getting the critical mass of use needed to build a broad development team. We don't expect to have or need many full-time developers, but active engagement from the HBase and Accumulo developers would significantly aid adoption and governance.

The other risk is YARN not having the complete feature set needed for long lived services: restarting, security token renewal, log-capture and other issues. We are working with the YARN developers to address these issues, issues shared with other long-lived services on YARN.

Orphaned Products

Steve Loughran plans to work full time on the project; the others splitting their time between Hoya and the applications that it already deploys. Closer integration with these applications will increase adoption, broaden the developer base and reduce any risks of orphanage.

Inexperience with Open Source

All of the core developers have long-standing experience in open source, Two of them are accumulo committers, two HBase committers. Steve Loughran has been a committer on various ASF projects since 2001, (Ant, Axis), a mentor to Incubated projects, a Hadoop committer since 2008, and full-time developer on HP's openSource SmartFrog project from 2005-2012.

Homogeneous Developers

The current core developers are all from Hortonworks. However, we hope to establish a developer community that includes users of Hoya and developers on the applications themselves -HBase, Accumulo, etc)

Reliance on Salaried Developers

Currently, the developers are paid to do work on Hoya. A key goal for the incubation process will be to broaden the developer base.

Relationships with Other Apache Products

This is covered in the Alignment section

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 Hoya a solid home as an open source project with a broad developer base -and to encourage adoption by the related ASF projects.

Documentation

All Hoya documentation is currently in markdown-formatted text files in the source repository; they will be delivered as part of the initial source donation.

Initial Source

The initial source ~~all ASF-licensed~~ can be found at https://github.com/hortonworks/hoya

Hoya is written in Java. Its source tree is entirely self-contained and relies on Apache Maven as its build system. Alongside the application and HBase and Accumulo providers, it contains unit, localhost and functional tests, the latter for use with remote clusters.

Source and IP Submission Plan

1. All source will be moved to Apache Infrastructure 2. All outstanding issues in our in-house JIRA infrastructure will be replicated into the Apache JIRA system. 3. We have a currently-unused twitter handle `@hoyaproject` which would be passed to the PMC.

External Dependencies

Hoya has no external dependencies except for some Java libraries that are considered ASF-compatible (JUnit, SLF4J, jcommander, groovy), and Apache artifacts : Hadoop, HBase, Accumulo, Log4J and the transient dependencies of all these artifacts.

Required Resources

Mailing Lists

1. hoya-dev 2. hoya-commits 3. hoya-private

Infrastructure

1. Git repository 2. JIRA Hoya (HOYA) 3. Gerrit for reviewing patches

The existing code includes local host integration tests, so we would like a Jenkins instance to run them whenever a new patch is submitted.

Initial Committers

1. Steve Loughran (stevel at a.o) 2. Billie Rinaldi (billie at a.o) 3. Ted Yu (tedyu at a.o) 3. Josh Elser (elserj at a.o) 4. Devaraj Das (ddas at a.o) 5. Larry McCay (lmccay at a.o) 6. Abhishek Kapoor

Sponsors

Champion: Vinod Kumar Vavilapalli

Nominated Mentors

1. Jean-Baptiste Onofré
2. Enis Söztutar
3. Vinod Kumar Vavilapalli
4. Ashutosh Chauhan
5. Arun Murthy

Sponsoring Entity

Incubator PMC