CuratorProposal

Curator - ZooKeeper client wrapper and rich ZooKeeper framework

Abstract

Curator is a set of Java libraries that make using Apache ZooKeeper much easier. While ZooKeeper comes bundled with a Java client, using the client is non-trivial and error prone.

Proposal

Curator is a set of Java libraries that make using Apache ZooKeeper much easier. While ZooKeeper comes bundled with a Java client, using the client is non-trivial and error prone. It consists of three components that build on each other. Curator Client is a replacement for the bundled ZooKeeper class that takes care of some low-level housekeeping and provides some useful utilities. Curator Framework is a high-level API that greatly simplifies using ZooKeeper. It adds many features that build on ZooKeeper and handles the complexity of managing connections to the ZooKeeper cluster and retrying operations. Curator Recipes consists of implementations of some of the common ZooKeeper "recipes". Additionally, Curator Test is included which includes utilities to help with unit testing ZooKeeper-based applications.

Background

Curator was initially developed by Netflix to make writing ZooKeeper-based applications easier and more reliable. Curator was open-sourced by Netflix on GitHub as an Apache 2.0 licensed project in July 2011. During this time Curator has been formally released many times and has gained widespread adoption.

Rationale

New users of ZooKeeper are surprised to learn that a significant amount of connection management must be done manually. For example, when the ZooKeeper client connects to the ensemble it must negotiate a new session, etc. This takes some time. If you use a ZooKeeper client API before the connection process has completed, ZooKeeper will throw an exception. These types of exceptions are referred to as "recoverable" errors. Curator automatically handles connection management, greatly simplifying client code. Instead of directly using the ZooKeeper APIs you use Curator APIs that internally check for connection completion and wrap each ZooKeeper API in a retry loop. Curator uses a retry mechanism to handle recoverable errors and automatically retry operations. The method of retry is customizable. Curator comes bundled with several implementations (ExponentialBackoffRetry, etc.) or custom implementations can be written.

The ZooKeeper documentation describes many possible uses for ZooKeeper calling each a "recipe". While the distribution comes bundled with a few implementations of these recipes, most ZooKeeper users will need to manually implement one or more of the recipes. Implementing a ZooKeeper recipe is not trivial. Besides the connection handling issues, there are numerous edge cases that are not well documented that must be considered. For example, many recipes require that an ephemeral-sequential node be created. New users of ZooKeeper will not know that there is an edge case in ephemeral-sequential node creation that requires you to put a special "marker" in the node’s name so that you can search for the created node if an I/O failure occurs. This is but one of many edge cases that are not well documented but are handled by Curator.

Current Status

Meritocracy

Curator was initially developed by Jordan Zimmerman in 2011 at Netflix. Developers external to Netflix provided feedback, suggested features and fixes and implemented extensions of Curator. Netflix's engineering team has since maintained the project and has been dedicated towards its improvement. Contributors to Curator include developers from multiple organizations around the world. Curator will be a meritocracy as it enters the Incubator and beyond.

Community

Curator is currently used by a number of organizations all over the world. Curator has an active and growing user and developer community with active participation in the http://groups.google.com/group/curator-users mailing list and at its GitHub home: https://github.com/Netflix/curator.

Since open sourcing the project, there have been fifteen individuals from various organizations who have contributed code.

Core Developers

The core developers for Curator are:

- Jordan Zimmerman
- Jay Zarfoss

Jordan has contributed towards Apache ZooKeeper and both Jordan and Jay are familiar with Apache principles and philosophy for community driven software development.

Alignment

Curator is a natural complement for Apache ZooKeeper. Java users of ZooKeeper will naturally want to use Curator. When Curator graduates from Incubator it may be useful to distribute Curator artifacts as part of ZooKeeper releases as the preferred/recommended client side library. Further, at graduation a determination can be made as to whether Curator should become a Top Level Project or be merged into ZooKeeper itself. That being said, some IPMC members feel that Curator should grow into its own TLP rather than be a part of ZooKeeper.

Known Risks

Orphaned Products

Curator is already deployed in production at multiple companies and they are actively participating in creating new features. Curator is getting traction with developers and thus the risks of it being orphaned are minimal.

Inexperience with Open Source

All code developed for Curator has been open sourced by Netflix under Apache 2.0 license. All committers to Curator are intimately familiar with the Apache model for open-source development and are experienced with working with new contributors.

Homogeneous Developers

The initial committers are from a single organization. However, we expect that once approved for incubation, the project will attract new contributors from diverse organizations and will thus grow organically. The submission of patches from developers from several different organizations is a strong indication that Curator will be widely adopted.

Reliance on Salaried Developers

It is expected that Curator will be developed on salaried and volunteer time, although all of the initial developers will work on it mainly on salaried time.

Relationships with Other Apache Products

Curator depends upon other Apache Projects: Apache ZooKeeper, Apache Log4J, and multiple Apache Commons components. Its build depends upon Apache Maven. Notably, there is interest from other Apache Projects such as HBase in adopting Curator as the client library for ZooKeeper. Apache James Mailbox has already incorporated Curator.

An Excessive Fascination with the Apache Brand

We would like Curator to become an Apache project to further foster a healthy community of contributors and consumers around the project. Since Curator directly interacts with Apache ZooKeeper and solves an important problem of many ZooKeeper users, residing in the Apache Software Foundation will increase interaction with the larger community.

Documentation

- Curator wiki at GitHub: https://github.com/Netflix/curator/wiki
- Curator issues at GitHub: https://github.com/Netflix/curator/issues
- Curator javadoc at GitHub: http://netflix.github.com/curator/doc/

Initial Source

- git://github.com/Netflix/curator.git

Source and Intellectual Property Submission Plan

- The initial source is already licensed under the Apache License, Version 2.0. https://github.com/Netflix/curator/blob/master/LICENSE.txt

External Dependencies

The required external dependencies are all Apache License or compatible licenses. Following components with non-Apache licenses are enumerated:

- org.slf4j: MIT-like License
- org.mockito: MIT-like License

Cryptography

Curator contains no known cryptography.

Required Resources

Mailing lists

- curator-private (with moderated subscriptions)
- curator-dev
- curator-commits
- curator-user

GitHub Repositories

http://github.com/apache/curator git://git.apache.org/curator.git

Issue Tracking

JIRA Curator (CURATOR)

Other Resources

The existing code already has unit and integration tests so we would like a Jenkins instance to run them whenever a new patch is submitted. This can be added after project creation.

Initial Committers

- Jordan Zimmerman (jzimmerman at netflix dot com)
- Jay Zarfoss (jzarfoss at netflix dot com)

Affiliations

- Jordan Zimmerman, Netflix
- Jay Zarfoss, Netflix

Sponsors

Champion

- Patrick Hunt

Nominated Mentors

- Patrick Hunt
- Enis Söztutar
- Mahadev Konar
- Luciano Resende

Sponsoring Entity

- Apache Incubator PMC