Curator's History

Curator's History

Curator was initially developed by Jordan Zimmerman at 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.

The Curator Incubator Proposal was presented in February 2013 and was formally accepted into the Incubator in March 2013.

Curator graduated to a Top Level Project in September, 2013.

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.