Building Big Systems with ZooKeeper

Course Summary

Description

This course is unique in that it teaches the best practices and caveats of designing modern concurrent Big Data systems. ZooKeeper is the ideal tool to understand and practice the theory, and to reason about system performance, fault tolerance, and stability.

ZooKeeper is the defacto standard for coordinating multiple components in distributed systems. In this class, we will learn ZooKeeper architecture, design, and implementation. Then we will go through the standard ZooKeeper design patterns and their implementation.

In recent year, most of the design work with ZooKeeper is done through Curator. Curator makes the implementation of the design patterns – called recipes – much easier and more robust. We will work with Elections (such as Leader Latch and Leader Election), Locks, Barriers, and more.

The course includes a balance of theory and lab work.

Topics

- Zookeeper fundamentals
- Zookeeper API in Java & C
- Zookeeper environment
- Curator and Exhibitor
- Curator recipes and use cases
- ZooKeeper internals

Audience

This course is designed for Developers, administrators, architects.

Prerequisites

Prior to taking this class, you should have experience and background in software development and administration.

Duration

Three days

Building Big Systems with ZooKeeper

Course Outline

I. ZooKeeper fundamentals
- A. Distribute coordination system
- B. Design goals and results
- C. Common coordination tasks

II. ZooKeeper Java and C API
- A. Goals and capabilities
- B. Differences, pros and cons
- C. Labs

III. ZooKeeper environment
- A. Track and react to ZooKeeper changes
- B. Handling failures (network, apps)
- C. Concurrency issues

IV. Curator and Exhibitor
- A. Goals and design
- B. Installation and configuration
- C. Advantages and current trends

V. Curator recipes and use cases
- A. Elections
- B. Locks
- C. Barriers
- D. Counters
- E. Caches
- F. Nodes
- G. Queues
- H. Centralized initialization

VI. ZooKeeper internals
- A. Internals
- B. Administration