

KafkaProposal

Abstract

Kafka is a distributed publish-subscribe system for processing large amounts of streaming data.

Proposal

Kafka provides an extremely high throughput distributed publish/subscribe messaging system. Additionally, it supports relatively long term persistence of messages to support a wide variety of consumers, partitioning of the message stream across servers and consumers, and functionality for loading data into Apache Hadoop for offline, batch processing.

Background

Kafka was developed at [LinkedIn](#) to process the large amounts of events generated by that company's website and provide a common repository for many types of consumers to access and process those events. Kafka has been used in production at [LinkedIn](#) scale to handle dozens of types of events including page views, searches and social network activity. Kafka clusters at [LinkedIn](#) currently process more than two billion events per day.

Kafka fills the gap between messaging systems such as Apache ActiveMQ, which provide low latency message delivery but don't focus on throughput, and log processing systems such as Scribe and Flume, which do not provide adequate latency for our diverse set of consumers. Kafka can also be inserted into traditional log-processing systems, acting as an intermediate step before further processing. Kafka focuses relentlessly on performance and throughput by not introspecting into message content, nor indexing them on the broker. We also achieve high performance by depending on Java's `sendFile/transferTo` capabilities to minimize intermediate buffer copies and relying on the OS's pagecache to efficiently serve up message contents to consumers. Kafka is also designed to be scalable and it depends on Apache [ZooKeeper](#) for coordination amongst its producers, brokers and consumers.

Kafka is written in Scala. It was developed internally at [LinkedIn](#) to meet our particular use cases, but will be useful to many organizations facing a similar need to reliably process large amounts of streaming data. Therefore, we would like to share it the ASF and begin developing a community of developers and users within Apache.

Rationale

Many organizations can benefit from a reliable stream processing system such as Kafka. While our use case of processing events from a very large website like [LinkedIn](#) has driven the design of Kafka, its uses are varied and we expect many new use cases to emerge. Kafka provides a natural bridge between near real-time event processing and offline batch processing and will appeal to many users.

Current Status

Meritocracy

Our intent with this incubator proposal is to start building a diverse developer community around Kafka following the Apache meritocracy model. Since Kafka was open sourced we have solicited contributions via the website and presentations given to user groups and technical audiences. We have had positive responses to these and have received several contributions and clients for other languages. We plan to continue this support for new contributors and work with those who contribute significantly to the project to make them committers.

Community

Kafka is currently being used by developed by engineers within [LinkedIn](#) and used in production in that company. Additionally, we have active users in or have received contributions from a diverse set of companies including [MediaSift](#), [SocialTwist](#), Clearspring and Urban Airship. Recent public presentations of Kafka and its goals garnered much interest from potential contributors. We hope to extend our contributor base significantly and invite all those who are interested in building high-throughput distributed systems to participate. We have begun receiving contributions from outside of [LinkedIn](#), including clients for several languages including Ruby, PHP, Clojure, .NET and Python.

To further this goal, we use [GitHub](#) issue tracking and branching facilities, as well as maintaining a public mailing list via Google Groups.

Core Developers

Kafka is currently being developed by four engineers at [LinkedIn](#): Neha Narkhede, Jun Rao, Jakob Homan and Jay Kreps. Jun has experience within Apache as a Cassandra committer and PMC member. Neha has been an active contributor to several projects [LinkedIn](#) has open sourced, including Bobo, Sensei and Zoie. Jay has experience with open source software as the originator of the Project Voldemort project, as well as being active within the Hadoop ecosystem community. Jakob is an Apache Hadoop committer and PMC and previous Apache [ZooKeeper](#) contributor.

Alignment

The ASF is the natural choice to host the Kafka project as its goal of encouraging community-driven open-source projects fits with our vision for Kafka. Additionally, many other projects with which we are familiar with and expect Kafka to integrate with, such as Apache Hadoop, Pig, [ZooKeeper](#) and log4j are hosted by the ASF and we will benefit and provide benefit by close proximity to them.

Known Risks

Orphaned Products

The core developers plan to work full time on the project. There is very little risk of Kafka being abandoned as it is a critical part of [LinkedIn](#)'s internal infrastructure and is in production use.

Inexperience with Open Source

All of the core developers have experience with open source development. [LinkedIn](#) open sourced Kafka several months ago and has been receiving contributions since. Jun is an Apache Cassandra committer and PMC member. Jay and Neha have been involved with several open source projects released by [LinkedIn](#). Jakob has been actively involved with the ASF as a full-time Hadoop committer and PMC member.

Homogeneous Developers

The current core developers are all from [LinkedIn](#). However, we hope to establish a developer community that includes contributors from several corporations and we actively encouraging new contributors via the mailing lists and public presentations of Kafka.

Reliance on Salaried Developers

Currently, the developers are paid to do work on Kafka. However, once the project has a community built around it, we expect to get committers, developers and community from outside the current core developers. However, because [LinkedIn](#) relies on Kafka internally, the reliance on salaried developers is unlikely to change.

Relationships with Other Apache Products

Kafka is deeply integrated with Apache products. Kafka uses Apache [ZooKeeper](#) to coordinate its state amongst the brokers, consumers, and soon, the producers. Kafka provides input formats to allow Hadoop [MapReduce](#) to load data directly from Kafka. Kafka provides an appender to allow consuming data directly from Apache log4j.

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 Kafka 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

Information about Kafka can be found at [<http://sna-projects.com/kafka/>] The following links provide more information about the project:

- Kafka roadmap and goals: [<http://sna-projects.com/kafka/projects.php>]
- The [GitHub] site: [<https://github.com/kafka-dev/kafka>]
- Kafka overview from Jay Kreps: [<http://www.slideshare.net/ydn/hug-january-2011-kafka-presentation>]
- Kafka overview from Jakob Homan: [<http://bit.ly/fLmoZz>]
- Kafka paper at NetDB 2011: [<http://research.microsoft.com/en-us/um/people/srikanth/netdb11/netdb11papers/netdb11-final12.pdf>]

Initial Source

Kafka has been under development at [[LinkedIn](#)] since November 2009. It was open sourced by [[LinkedIn](#)] in January 2011. It is currently hosted on github under the Apache license at [<https://github.com/kafka-dev/kafka>]

Kafka is mainly written in Scala with some performance testing code in Java. Several clients have been contributed in other languages, including Ruby, PHP, Clojure, .NET and Python. Its source tree is entirely self contained and relies of simple build tool (sbt) as its build system and dependency resolution mechanism.

External Dependencies

The dependencies all have Apache compatible licenses.

Cryptography

Not applicable.

Required Resources

Mailing Lists

- kafka-private for private PMC discussions (with moderated subscriptions)
- kafka-dev
- kafka-commits
- kafka-user

Subversion Directory

[<https://svn.apache.org/repos/asf/incubator/kafka>]

Issue Tracking

JIRA Kafka (KAFKA)

Other Resources

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

Initial Committers

- Jay Kreps
- Jun Rao
- Neha Narkhede
- Jakob Homan
- Phillip Rhodes
- Henry Saputra
- Chris Burroughs

Affiliations

- Jay Kreps (LinkedIn)
- Jun Rao (LinkedIn)
- Neha Narkhede (LinkedIn)
- Jakob Homan (LinkedIn)
- Phillip Rhodes (Fogbeam Labs)
- Henry Saputra (Cisco Systems)
- Chris Burroughs (Clearspring Technologies)

Sponsors

Champion

Chris Douglas (Apache Member)

Nominated Mentors

- Alan Cabrera (Apache Member)
- Geir Magnusson, Jr. (Apache Member and Director)
- Owen O'Malley (Apache Member)

Sponsoring Entity

We are requesting the Incubator to sponsor this project.