SamzaProposal

Abstract

Samza is a stream processing system for running continuous computation on infinite streams of data.

Proposal

Samza provides a system for processing stream data from publish-subscribe systems such as Apache Kafka. The developer writes a stream processing task, and executes it as a Samza job. Samza then routes messages between stream processing tasks and the publish-subscribe systems that the messages are addressed to.

Background

Samza was developed at LinkedIn to enable easier processing of streaming data on top of Apache Kafka. Current use cases include content processing pipelines, aggregating operational log data, data ingestion into distributed database infrastructure, and measuring user activity across different aggregation types.

Samza is focused on providing an easy to use framework to process streams. It uses Apache YARN to provide a mechanism for deploying stream processing tasks in a distributed cluster. Samza also takes advantage of YARN to make decisions about stream processor locality, co-partition of streams, and provide security. Apache Kafka is also leveraged to provide a mechanism to pass messages from one stream processor to the next. Apache Kafka is also used to help manage a stream processor's state, so that it can be recovered in the event of a failure.

Samza 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 Samza. While our use case of processing events from a large website like LinkedIn has driven the design of Samza, its uses are varied and we expect many new use cases to emerge. Samza provides a generic API to process messages from streaming infrastructure and will appeal to many users.

Current Status

Meritocracy

Our intent with this incubator proposal is to start building a diverse developer community around Samza following the Apache meritocracy model. Since Samza was initially developed in late 2011, we have had fast adoption and contributions by multiple teams at LinkedIn. We plan to continue support for new contributors and work with those who contribute significantly to the project to make them committers.

Community

Samza is currently being used internally at LinkedIn. We hope to extend our contributor base significantly and invite all those who are interested in building large-scale distributed systems to participate.

Core Developers

Samza is currently being developed by four engineers at LinkedIn: Jay Kreps, Jakob Homan, Sriram Subramanian, and Chris Riccomini. Jakob is an ASF Member, Incubator PMC member and PMC member on Apache Hadoop, Kafka and Giraph. Jay is a member of the Apache Kafka PMC and contributor to various Apache projects. Chris has been an active contributor for several projects including Apache Kafka and Apache YARN. Sriram has contributed to Samza, as well as Apache Kafka.

Alignment

The ASF is the natural choice to host the Samza project as its goal of encouraging community-driven open-source projects fits with our vision for Samza. Additionally, many other projects with which we are familiar with and expect Samza to integrate with, such as Apache ZooKeeper, YARN, HDFS 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 Samza being abandoned as it is part of LinkedIn's internal infrastructure.

Inexperience with Open Source

All of the core developers have experience with open source development. Jay and Chris has been involved with several open source projects released by LinkedIn, and Jay is a committer on Apache Kafka. Jakob has been actively involved with the ASF as a full-time Hadoop committer and PMC member. Sriram is a contributor to Apache Kafka.

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 Samza.

Reliance on Salaried Developers

Currently, the developers are paid to do work on Samza. 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 Samza internally, the reliance on salaried developers is unlikely to change.

Relationships with Other Apache Products

Samza is deeply integrated with Apache products. Samza uses Apache Kafka as its underlying message passing system. Samza also uses Apache YARN for task scheduling. Both YARN and Kafka, in turn, rely on Apache ZooKeeper for coordination. In addition, we hope to integrate with Apache HDFS in the near future.

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 Samza 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

http://wiki.apache.org/incubator/SamzaProposal

Initial Source

Available upon request.

External Dependencies

The dependencies all have Apache compatible licenses.

- metrics (Apache 2.0)
- zkclient (Apache 2.0)
- zookeeper (Apache 2.0)
- jetty (Apache 2.0)
- jackson (Apache 2.0)
- commons-httpclient (Apache 2.0)
- slf4j (MIT)
- avro (Apache 2.0)
- hadoop (Apache 2.0)
- junit (Common Public License)
- grizzled-slf4j (BSD)
- scalatra (https://github.com/scalatra/scalatra/blob/develop/LICENSE)
- scala (http://www.scala-lang.org/node/146)
- joptsimple (MIT)
- kafka (Apache 2.0)
- scalate (Apache 2.0)
- leveldb jni (BSD)

Cryptography

Samza will depend on secure Hadoop, which can optionally use Kerberos.

Required Resources

Mailing Lists

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

Subversion Directory

Git is the preferred source control system: git://git.apache.org/samza

Issue Tracking

JIRA Samza (SAMZA)

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
- Jakob Homan
- Chris Riccomini
- Sriram Subramanian

Affiliations

- Jay Kreps (LinkedIn)
- Jakob Homan (LinkedIn)
- Chris Riccomini (LinkedIn)
- Sriram Subramanian (LinkedIn)

Sponsors

Champion

Jakob Homan (Apache Member)

Nominated Mentors

- Arun C Murthy <acmurthy at apache dot org>
- Chris Douglas <cdouglas at apache dot org>
- Roman Shaposhnik <rvs at apache dot org>

Sponsoring Entity

We are requesting the Incubator to sponsor this project.