DistributedLogProposal

Abstract

DistributedLog is a high-performance replicated log service. It offers durability, replication and strong consistency, which provides a fundamental building block for building reliable distributed systems, e.g replicated-state-machines, general pub/sub systems, distributed databases, distributed queues and etc.

See "Building Distributedlog - Twitter's high performance replicated log service" for details: https://blog.twitter.com/2015/building-distributedlog-twitter-s-high-performance-replicated-log-service

Proposal

We propose to contribute DistributedLog codebase and associated artifacts (e.g. documentation, web-site content etc.) to the Apache Software Foundation with the intent of forming a productive, meritocratic and open community around DistributedLog's continued development, according to the 'Apache Way'.

Background

Engineers at Twitter began developing DistributedLog in early 2013. DistributedLog is described in a Twitter engineering blog post and presented at the Messaging Meetup in Sep 2015. It has been released as an Apache-licensed open-source project on GitHub in May 2016.

DistributedLog is a high-performance replicated log service, which provides simple stream-oriented abstractions over log-segments and offers durability, replication and strong consistency for building reliable distributed systems. The features offered by DistributedLog includes:

- Simple high-level, stream oriented interface
- Naming and metadata scheme for managing streams and other entities
- Log data management policies, include data segmentation and data retention
- Fast write pipeline leveraging batching and compression
- Fast read mechanism leveraging long-poll and read-ahead caching
- Service tiers supporting writer fan-in and reader fan-out
- Geo-replicated logs

DistributedLog's most important benefit is high-performance with a strong durability guarantee, making it extremely appropriate for running different workloads from distributed database journaling to real-time stream computing. Its modern, layered architecture makes it easy to run the service tiers in multi-tenant datacenter environments such as Apache Mesos or cloud environments such as EC2.

Rationale

DistributedLog is designed to provide core fundamental features like high-performance, durability and strong consistency to anyone who is building reliable distributed systems, in a simple and efficient way.

We believe that the ASF is the right venue to foster an open-source community around DistributedLog's development. We expect that DistributedLog will benefit from collaboration with related Apache projects, and under the auspices of the ASF will attract talented contributors who will push DistributedLog's development forward at a faster pace.

We believe that the timing is right for DistributedLog's development to move to the ASF: DistributedLog has already run in production at Twitter for 3 years and served various workloads including a distributed database journal, reliable cross datacenter replication, search ingestion, andgeneral pub/sub messaging. The project is stable. We are excited to see where an ASF-based community can take DistributedLog.

Current Status

DistributedLog is a stable project that has been used in production at Twitter for 3 years. The source code is public at github.com/twitter, which will seed the Apache git repository.

Meritocracy

We understand the central importance of meritocracy to the Apache Way. We will work to establish a welcoming, fair and meritocratic community. Several companies have already expressed interest in this project, and we intend to invite additional developers to participate. We look forward to growing a rich user and developer community.

Community

There is a large need for a performant replicated log service for applications such as distributed databases, distributed transactional systems, replicated-state-machines and pub/sub messaging/queuing. We want to attract more developers to the project, and we believe that the ASF's open and meritocratic philosophy will help us with this. We note the success of other similar projects already part of the ASF, like Kafka.

Core Developers

DistributedLog is actively developed within Twitter. Most of the developers are from Twitter. Many of them are committers or PMC members of Apache BookKeeper. Others aren't currently affiliated with ASF so they will require new ICLAs.

Alignment

DistributedLog is related to several other Apache projects:

- DistributedLog stores log segments as Ledgers in Apache BookKeeper.
- DistributedLog uses Apache ZooKeeper for naming and metadata management and tracking the ownership of logs.
- DistributedLog uses Apache Thrift as its RPC and serialization framework.
- In the long-term, DistributedLog's data will be stored in Apache Hadoop clusters powered by HDFS filesystem for archives and backup.

Known Risks

Orphaned Products

DistributedLog is used as the fundamental messaging infrastructure at Twitter. It has been serving production traffic for online database systems, search ingestion and a general pub/sub system. Twitter remains committed to developing and supporting the project. Twitter has a strong track record in standing behind projects that were contributed to the ASF by its employees, including Apache Mesos, Apache Aurora, Apache BookKeeper, Apache Hadoop. There are many companies are interested in using it in production.

Inexperience with Open Source

The core developers of DistributedLog are committers of Apache BookKeeper. Although other committers on the initial list are committers or have less experience with the ASF, they already are active in Apache BookKeeper community. We are confident that the project can be run in accordance with Apache principles on an ongoing basis.

Homogeneous Developers

The initial committers are from Twitter. We hope to encourage contributions from other developers and grow them into committers after they have had time to continue their contributions.

Reliance on Salaried Developers

Many of DistributedLog's initial set of committers work full-time on DistributedLog, and are paid to do so. However, as mentioned elsewhere, we anticipate growth in the developer community which we hope will include people from industry, hobbyists, and academics who have an interested in distributed messaging systems.

Relationships with Other Apache Products

DistributedLog uses Apache BookKeeper to store log segments and Apache ZooKeeper to store log metadata and manage log namespaces. It provides an end-to-end solution for replicated logs, to make building reliable distributed systems much easier. Unlike Kafka or ActiveMQ, DistributedLog is not a full-fledged pub/sub, queuing or messaging system. Instead, it is targeting on providing a fundamental building block for other distributed systems, offering durability, replication and consistency. So it could be used by other distributed systems, such as transactional log for replicated state machines (e.g., HDFS NameNode), WAL for distributed databases (e.g. HBase), Journal for in-memory services (e.g., Kestrel) and even storage backend for a full-fledged messaging system.

An Excessive Fascination with the Apache Brand

DistributedLog builds on two existing top-level projects, Apache BookKeeper and Apache ZooKeeper. Some of the core developers actively participate in both projects and understand well the implications of being hosted by Apache. We would like this project to build on the same core values of ASF and to grow a community based on meritocracy. Also, there are several other projects already hosted by ASF in this space of reliable messaging and that overlap with DistributedLog in interests and scope. Consequently, the combination of all these observations makes us believe that DistributedLog should be hosted by the ASF.

Documentation

Building DistributedLog: Twitter's high performance replicated log service (https://blog.twitter.com/2015/building-distributedlog-twitter-s-high-performance-replicated-log-service)

Documentation located in http://distributedlog.io.

Initial Source

DistributedLog's initial source contribution will come from http://github.com/twitter/distributedlog/.

External Dependencies

DistributedLog depends upon a number of third-party libraries, which we list below.

- Apache BookKeeper (Apache Software License v2.0)
- Apache Commons (Apache Software License v2.0)
- Apache Maven (Apache Software License v2.0)
- Apache Thrift (Apache Software License v2.0)
- Apache ZooKeeper (Apache Software License v2.0)
- Google Guava (Apache Software License v2.0)
- Mockito (MIT License)
- Junit (Eclipse Public License 1.0)
- LZ4-java (Apache Software License v2.0)
- SLF4J (MIT License)
- Twitter Finagle (Apache Software License v2.0)
- Twitter Scrooge (Apache Software License v2.0)
- Twitter Util (Apache Software License v2.0)

Required Resources

We request that following resources be created for the project to use:

Mailing lists

- private@distributedlog.incubator.apache.org (moderated subscriptions)
- commits@distributedlog.incubator.apache.org
- dev@distributedlog.incubator.apache.org
- user@distributedlog.incubator.apache.org

Git repository

https://git.apache.org/distributedlog.git

JIRA instance

JIRA project DLOG (DLOG or DL)

Initial Committers

- Sijie Guo (Apache BookKeeper Committer, Twitter)
- Robin Dhamankar (Apache BookKeeper Committer)
- Leigh Stewart (Twitter)
- Dave Rusek (Twitter)
- Honggang Zhang (Twitter)
- Jordan Bull (Twitter)
- Satish Kotha (Twitter)
- Aniruddha Laud
- Franck Cuny (Twitter)
- Eitan Adler (Twitter)

Affiliations

Most of the initial committers are employees of Twitter, except Robin Dhamankar and Aniruddha Laud.

Sponsors

Champion

Flavio Junqueira

Nominated Mentors

- Flavio Junqueira
- Chris Nauroth
- Henry Saputra

Sponsoring Entity

We ask that the Apache Incubator PMC to sponsor this proposal.