RocketMQProposal

RocketMQ Proposal

Abstract

RocketMQ is a fast, low latency, reliable, scalable, distributed, easy to use message-oriented middleware, especially for processing large amounts of streaming data.

Proposal

RocketMQ provides a message model including both pub/sub and P2P and it supports both reliable FIFO and strict sequential message queues. It also has the ability to accumulate a billion messages in a single queue, provides mobile, internet-friendly protocols such as MQTT and HTTP. RocketMQ also supports the ability to load data into Apache Hadoop for offline storage or to handle stream processing for Apache Storm.

Background

RocketMQ was developed at Alibaba in 2011 and has been used in production there since that time. It can process the large amounts of events generated by various systems and provides a common repository for many types of consumers to access and process those events. RocketMQ also handles dozens of types of events including trade order process, search, social network activity stream and data pipeline. Every day at Alibaba, RocketMQ clusters process more than 500 billion events. The Alibaba Group also uses RocketMQ to provide message services for more than 3000 core applications.

RocketMQ was developed to meet Alibaba's particular use cases to provide low latency message delivery and high throughput message sending. Alibaba has also created its cornerstone product derived from RocketMQ, a Platform as a Service (PaaS) product named the Alibaba Cloud Platform (https://intl.aliyun.com/). More than 100 companies use the RocketMQ open source version today. We believe RocketMQ can benefit more people so, we would like to share it via the ASF and begin developing a community of developers and users via The Apache Way.

Rationale

As background description, many organizations can benefit from a low latency, reliable, high throughput, distributed platform. Its usage is varied and we expect many new use cases to emerge. RocketMQ provides many features to support many use cases from enterprise application integration, to web applications to the flourishing of IoT applications.

Current Status

Meritocracy

The intent of this proposal is to start building a diverse developer and user community around RocketMQ following the ASF meritocracy model. Since RocketMQ was open sourced, we have solicited contributions via the website and presentations given to user groups and technical audiences and have received positive feedback and contributions including clients for C++ and .NET. We plan to continue this support for new contributors and work with those who contribute significantly to the project to encourage them to become committers.

Community

RocketMQ is currently being developed by engineers working for Alibaba where it is highly used in a production environment. We also have active users in or have received contributions from a diverse set of companies including CMBC(China Minsheng Bank), Schneider Electric(http://www.schneider-electric.com/), the China Railway Ministry official ticketing website, China Union, Sina, Umei (http://sh.jumei.com), Chinese Academy of Sciences and many more. We hope to grow the base of contributors by inviting all those who offer significant contributions and excel through the use of The Apache Way. Contributions from outside of Alibaba are now being received by the RocketMQ project, including a dashboard, the flume-rocketmq module, the storm-rocketmq and more.

To further this goal, the project currently makes use of GitHub project features as well as a public mailing list via Google Groups.

Core Developers

RocketMQ is currently being developed by engineers from Alibaba and Yeahmobi: Xiaorui Wang, Von Gosling, Jiangwei Jiang, Xinyu Zhou, Zhanhui Li. Xiaorui Wang, one of Alibaba MOM project owners is also the originator of the RocketMQ project. He has rich experience with open source software, as well as being active within the RocketMQ community. Von Gosling, another MOM project owner at Alibaba and co-creator of the RocketMQ project, is an active open source software committer and has been an active contributor to several projects in Alibaba, Apache community and Google Code. Von also has deep experience with performance tuning, distributed system design and coding. Xinyu Zhou, Wei Zhou and Jinjixiang, They have rich experience in distributed system design and performance tuning, especially in message queue, big data, etc. Zhanhui Li, is a developer at Yeahmobi who has a great passion for software engineering, especially in fields of distributed system design and development. Longda, has been involved with several open source projects released by Alibaba,especially jstorm project, which has donated to apache.

Aligment

The ASF is the natural choice to host the RocketMQ project as its goal of encouraging community-driven open source projects fits with our vision for RocketMQ. The ASF is also the home to many other projects with which we are familiar with and hope to integrate with RocketMQ including Apache Storm, Flume and Hadoop. We believe that there will be mutual benefit by close proximity to these and other projects.

Known Risks

Orphaned products

The core developers currently work full-time on the RocketMQ project for Alibaba. RocketMQ provides a critical internal infrastructure and has been in production use at Alibaba since 2011, so there is no concern that it will become an orphaned project.

Inexperience with Open Source

The core developers are all active contributors, users and followers of open source. They are all already committers and contributors to the RocketMQ Github project and several of them also have experience actively contributing to open source projects. Though the core set of developers do not have experience at the ASF, there are plans to onboard individuals to the project who have ASF open source experience.

Homogenous Developers

The current core developers are from Alibaba and Yeahmobi but the goal is to establish a vibrant developer community and we will actively encourage new contributors.

Reliance on Salaried Developers

Currently, the developers are paid to work on RocketMQ for both Alibaba and Yeahmobi. We look forward to building a strong community around the project in order to encourage more contributors to join the project.

Relationships with Other Apache Products

RocketMQ is not currently integrated with any other ASF projects except for several Apache Commons products, such as commons-lang, commons-cli, etc. As a messaging server, it is mainly used to traditional publish/subscribe scenario and demandingly high volume realtime transaction system that tolerates no message loss. It uses java primitive file api and some jni technology to custom-build storage desgin. More details about its storage, please see the RocketMQ Storage Design) The table below providing some context for simple comparison to some other Apache messaging projects at the ASF such as ActiveMQ and Kafka (apologies for the terrible MoinMoin table formatting, its syntax is limited. For a much better formatted table, please see the RocketMQ Github page).

Messaging Product	**Protocol and Specification**	**Order Message**	**Message Filter**	**Server Triggered Redelivery**	**Persistent Messaging**	**Retroactive Consumers**	**Message Priority**	**High Availability and Failover**	**Message Tracking**	**Configuration**	**Management and Operation Tools**
ActiveMQ	Push model, supports Open Wire, Stomp, AMQP, HTTP, JMS, MQTT	Exclusive consumer or exclusive queues will ensure ordering	Supported	Not Supported	Supports very fast persistence using JDBC along with a high performance journalsuch as LevelDB and KahaDB	Supported	Supported	Supported, depending on storage, if Kahadb storage a ZooKeeper server is required, maybe trigger split brain after rebooting half of a two-node cluster	Not Supported	The default configuration is low level, user need to optimize the configuration	Supported
Kafka	Pull model, supports TCP	Ensure ordering of messages within a partition	Supported, you can use Kafka Streams to filter messages	Not Supported	High performance file storage	Supported offset indicate	Not Supported	Supported, requires a ZooKeeper server	Not Supported	Kafka uses key-value pairs format for configuration. These values can be supplied either from a file or programmatically.	Supported, use terminal commands to expose core metrics
RocketMQ	Pull model, supports TCP, JMS	Ensure strict ordering of messages, have no hot spot problem and can scale out gracefully	Supported, you can even upload yourself custom-built filter code snippets	Supported	High performance and low latency file storage	Supported timestamp and offset 2 indicates	Not Supported	Supported,Master-Slave model, without anther kit	Supported	All configurations are off the shelf, user only needs to pay attention to a few configurations	Supported, rich web and terminal command to expose core metrics

A Excessive Fascination with the Apache Brand

While we respect the reputation of the Apache brand and have no doubt that it will attract new contributors and users, our interest is primarily to give RocketMQ a solid home as an open source project following an established development model. More reason are provided in the Rationale and Alignment sections.

Documentation

Information about RocketMQ can be found on the Github project wiki [https://github.com/alibaba/RocketMQ]

Initial Source

RocketMQ has been under development at Alibaba since 2011. The source code was opened up in 2012. It is currently hosted on Github using the Apache License ([https://github.com/alibaba/RocketMQ/blob/master/LICENSE]).

External Dependencies

RocketMQ depends on some Apache projects:

- Commons Lang
- Commons CLI
- Commons Codec
- Maven

and other open source projects (organized by license):

- ALv2:
  - JNA
  - Netty
  - FastJson
  - Jackson
- MPL
  - Javassist
- MIT
  - Mockito
  - SLF4J
  - Bridge-method-annotation
- EPL 1.0
  - JUnit
  - Logback

As all dependencies are managed using Apache Maven, none of the external libraries need to be packaged in a source distribution.

Required Resources

Mailing lists

- rocketmq-private (PMC discussion)
- rocketmq-dev (developer discussion)
- rocketmq-user (user discussion)
- rocketmq-commits (SCM commits)
- rocketmq-issues (JIRA issue feed)

Subversion Directory

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

Issue Tracking

JIRA RocketMQ (RocketMQ)

Other Resources

The existing source code already has unit tests so we will make use of existing Apache continuous testing infrastructure. The resulting load should not be very large.

Initial Committers

- Xiaorui Wang <vintage.wang at gmail dot com>
- Von Gosling <fengjia10 at gmail dot com>
- Xinyu Zhou <zhou at xinyu dot im>
- Zhanhui Li <lizhanhui at gmail dot com>
- Wei Zhou <zhouwei198732 at gmail dot com>
- Jinjixiang <canna1989115 at gmail dot com>
- Longda <hustjackie at gmail dot com>

Affiliations

- Xiaorui Wang: Alibaba
- Von Gosling: Alibaba
- Xinyu Zhou: Alibaba
- Zhanhui Li: Yeahmobi
- Wei Zhou: Alibaba
- Jinjixiang: Alibaba

- Longda: Alibaba

Sponsors

Champion

Bruce Snyder bsnyder@apache.org

Nominated Mentors

- Justin McLean jmclean@apache.org
- Bruce Snyder bsnyder@apache.org
- Brian McCallister brianm@apache.org
- Willem Ning Jiang ningjiang@apache.org
- Luke Han lukehan@apache.org

Sponsoring Entity

We are requesting the Incubator to sponsor this project.