Abstract

Event Mesh is a dynamic plug-in cloud-native basic service layer used to decouple the application and middleware layer. It provides flexible, reliable and fast event distribution, and can be managed.

- EventMesh codebase: https://github.com/WeBankFinTech/EventMesh

Proposal

EventMesh is an event-driven micro-service infrastructure that implements RPC and event capability. EventMesh not only provides event sourcing and event filtering, event security and event audit, event governance and management, but also provides common service invocation and message modes such as RPC capability and Messaging asynchronous event notification, event multicast and broadcast. In addition, it also provides high-availability capabilities in distributed scenarios such as multi-centre and multi-active applications, nearby service calls, and grey release. The enhancement of fault tolerance for machine failures also makes EventMesh more stable and reliable, providing 7x24 services for business.

We propose to contribute the EventMesh code base and related auxiliary tools to the Apache Software Foundation, and look forward to establishing an open community for the continuous development of EventMesh in the "Apache Way".

Background

In the past, the Enterprise Service Bus(ESB) was unable to cope with the challenges of large service invocations in an multi-center distributed architecture. Specifically, with the growth of the business, the number of distributed system services has increased sharply; the inter-system relationships are diversified and complicated, and service governance is more complicated; the number of services and Data Center Nodes(DCN) is constantly growing; due to the limited single node capacity, capacity expansion operations become more frequent, which improves the lossless capacity expansion requirements of the business; the system's disaster tolerance becomes weaker, the system switching becomes complicated, and the business lossless requirements for machine failures become higher, etc.

EventMesh supports diversified service invocation methods to meet online, batch, and big data processing scenarios. It supports multi-centre and multi-active disaster tolerance and higher machine utilization, flexible and smooth upgrade and expansion methods, unlimited capacity, full-process service management and tool assistance to achieve fast and automatic isolation of faulty applications, the framework itself or Internet connection. EventMesh has technical advantages such as millisecond delay and 100 million message processing capabilities and helps financial companies to meet challenges in all directions.

Rationale

EventMesh is an event-driven microservice infrastructure that implements RPC and event capability. EventMesh not only provides RPC capability but also provides common service invocation and message modes such as Messaging asynchronous event notification, event multicast and broadcast. In addition, it also provides high-availability capabilities in distributed scenarios such as multi-centre multi-active applications, nearby service calls, and grey release. The enhancement of fault tolerance for machine failures also makes EventMesh more stable and reliable, providing 7x24 services for business.

We propose to contribute the EventMesh code base and related auxiliary tools to the Apache Software Foundation, and look forward to establishing an open community for the continuous development of EventMesh in the "Apache Way".

Initial Goal

- Migrate the existing codebase, website, and documentation to Apache-hosted infrastructure.
- Work with the infrastructure team to implement and approve our code review, build, and testing workflows in the context of the ASF.
- Incremental development and releases along with Apache guidelines.

Current Status

EventMesh has been an open-source project on GitHub https://github.com/WeBankFinTech/EventMesh since September 2019.

Meritocracy

This proposal intends to start building a community around EventMesh following the ASF meritocracy model. We plan to invite more people as committers if they want to contribute to this project.

Releases

EventMesh has undergone multiple public releases, listed here: https://github.com/WeBankFinTech/EventMesh/releases

These old releases were not performed in the typical ASF fashion. We will adopt the ASF source release process upon joining the incubator.
Code Reviews

EventMesh's code reviews are currently public on Github: [https://github.com/WeBankFinTech/EventMesh/pulls](https://github.com/WeBankFinTech/EventMesh/pulls).

Community

EventMesh seeks to develop developer and user communities during incubation.

Core Developers

Currently, most of the core developers of EventMesh are working in the middleware Team of WeBank.

- Guangsheng Chen ([https://github.com/ggeasonchen](https://github.com/ggeasonchen))
- Weiqiang Liang ([https://github.com/wqliang](https://github.com/wqliang))
- Yang Jun ([https://github.com/yangjun781120](https://github.com/yangjun781120))
- Mengfei Xiong ([https://github.com/iNanos](https://github.com/iNanos))

Alignment

EventMesh is aligned with several other ASF projects.

EventMesh uses RocketMQ as the default implementation of event storage. Event producers send events to RocketMQ through EMesher, and event consumers consume events from RocketMQ through EMesher.

Known Risks

Orphaned Products

The risk of EventMesh becoming an orphan product is very low because it is the core architecture product of at least three companies (such as WeBank, Alibaba, Oppo) and is widely used in production environments, reaching a scale of more than 100 clusters. In addition, since EventMesh was open-sourced on Github in September 2019, it has received 250 stars and been forked more than 70 times in a few months and has received many feedbacks from researchers and users outside WeBank. We hope to further build the EventMesh community through Apache and diversify it.

Inexperience with Open Source

EventMesh's core developers are not only committers and contributors to the EventMesh Github project, but also active users and followers of the open-source community. Most developers are involved in work related to source code released under an open-source license, and some of them also have experience in developing code in an open-source environment.

WeBank attaches great importance to work in the field of open source. The leader in the WeBank middleware team is the core committer of other ASF projects, and the other members are contributors. They will continue to invest in guiding others to practice the Apache Way with other incubator mentors.

Homogenous Developers

The EventMesh project has begun to attract community users, received many peer feedback, and contributions from developers outside WeBank are very welcome. We hope to work with these developers and community members to build the EventMesh community, encourage more users to participate in the community, and develop them into committers when they make important contributions.

An Excessive Fascination with the Apache Brand

EventMesh is proposing to enter incubation at Apache in order to help efforts to diversify the committer-base, not so much to capitalize on the Apache brand. The EventMesh project is in production use already inside WeBank, but is not expected to be a WeBank product for external customers. As such, the EventMesh project is not seeking to use the Apache brand as a marketing tool.

Documentation

Information about EventMesh can be found at [https://github.com/WeBankFinTech/EventMesh](https://github.com/WeBankFinTech/EventMesh). The following links provide more information about EventMesh in open source:

- Codebase at Github: [https://github.com/WeBankFinTech/EventMesh](https://github.com/WeBankFinTech/EventMesh)
- Issue Tracking: [https://github.com/WeBankFinTech/EventMesh/issues](https://github.com/WeBankFinTech/EventMesh/issues)
- Releases: [https://github.com/WeBankFinTech/EventMesh/releases](https://github.com/WeBankFinTech/EventMesh/releases)
### Initial Source

Besides the core codebase, EventMesh also hosts its side projects on Github under WeBankFinTech Group. Specifically, the initial source includes:

#### Components of EventMesh:

- **eventmesh-store**: the store layer of EventMesh which implemented with DeFiBus (custom develop on RocketMQ) by default. We wish the store layer is a general solution and can use any store implement such as kafka, redis etc.

- **eventmesh-registry**: automatically routes events between applications and services connected to separate event meshers

- **eventmesh-emesh**: an middleware to transmit events between event producers and consumers, support cloud native apps and microservices

- **eventmesh-governance**: governance layer for event producers and consumers

- **eventmesh-acl**: security at various level of authentication, authorization and topic/channel access control

- **eventmesh-sdk**: support for popular open standard protocols and APIs, including REST/HTTP, AMQP, MQTT, Websocket and JMS, gRPC etc.

### External Dependencies

EventMesh has the following external dependencies.

- RocketMQ (Apache 2.0)
- Netty (Apache 2.0)
- tomcat (Apache 2.0)
- Guava (Apache 2.0)
- slf4j (MIT License)
- commons-lang (Apache 2.0)
- commons-collections (Apache 2.0)
- commons-io (Apache 2.0)
- httpclient (Apache 2.0)
- fastjson (Apache 2.0)
- metrics (Apache 2.0)
- log4j (Apache 2.0)
- disruptor (Apache 2.0)
- mybatis (Apache 2.0)
- jackson (Apache 2.0)
- spring-boot (Apache 2.0)
- logback (GNU 2.1)

Build and test dependencies:

- Apache Maven (Apache 2.0)
- Gradle (Apache 2.0)
- Junit (Eclipse Public License - v 1.0)
- mockito (MIT License)
- powermock (Apache 2.0)
- assertj (Apache 2.0)

### Required Resources

### Mailing List

There are currently no mailing lists. The usual mailing lists are expected to be set up when entering incubation:
Git Repositories

Upon entering incubation, we want to move the existing repository from https://github.com/WeBankFinTech/EventMesh to Apache infrastructure like https://github.com/apache/incubator-eventmesh.

Issue Tracking

EventMesh currently uses Github to track issues. Would like to continue to do so while we discuss migration possibilities with the ASF Infra committee.

Other Resources

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

Source and Intellectual Property Submission Plan

Most of the current code is Apache 2.0 licensed and the copyright is assigned to WeBank. If the project enters incubator, WeBank will transfer the source code & trademark ownership to ASF via a Software Grant Agreement.

Initial Committers

- Guangsheng Chen (chenguangsheng@apache.org)
- Jun Yang (yangjun1120@gmail.com)
- Weiqiang Liang (wqliang98@gmail.com)
- Mengfei Xiong (xiong_mengfei@163.com)
- Yiliu Chen (yiliusange@gmail.com)
- Fan He (601023364@qq.com)
- Hongbing Yan (397294722@qq.com)
- Yanqing Zhong (185682649@qq.com)
- Heng Du (duhengforever@apache.org)
- Jaskey (linjunjie@apache.org)
- Suri (942576560@qq.com)
- Le Liu (liulefirst@163.com)
- Yi Chen (409696597@qq.com)
- Weiming Xue (mike_xwm@126.com)
- Huan Li (lihuansice@sina.com)
- Ronghua Liang (liangrh7@163.com)
- Lei Ding (dinglie@apache.org)

Affiliations

Guangsheng Chen, Jun Yang, Weiqiang Liang, Mengfei Xiong, Yiliu Chen, Fan He, Hongbing Yan, Yanqing Zhong, Suri, Le Liu, Yi Chen, Weiming Xue, Huan Li and Ronghua Liang of the initial committers are employees of WeBank.

Heng Du, Lei Ding of the initial committers are employees of Alibaba.

Jaskey of initial committers are employees of Oppo.

Sponsors
Champion
Von Gosling (ASF Member, IPMC Member), vongosling@apache.org

Nominated Mentors
Justin Mclean (ASF Member, IPMC VP), jmclean@apache.org
Von Gosling (ASF Member, IPMC Member), vongosling@apache.org
Junping Du (ASF Member, IPMC Member), junping_du@apache.org
Francois Papon (ASF Member, IPMC Member), fpapon@apache.org
Jean-Baptiste Onofre (ASF Member, IPMC Member), jbonofre@apache.org

Informal Mentors
Heng Du (duhengforever@apache.org)

Sponsoring Entity
We are requesting the Incubator to sponsor this project.