

ZipkinProposal

*_Follow the Apache guide: [<https://incubator.apache.org/guides/proposal.html>]|Guide *

`\${renderedContent}`

Abstract

Zipkin is a distributed tracing system. It helps gather timing data needed to troubleshoot latency problems in microservice architectures. It manages both the collection and lookup of this data. Zipkin's design is based on the Google Dapper paper.

Proposal

Zipkin provides a defined data model and payload type for distributed trace data collection. It also provides an UI and http api for querying the data. Its server implements this api and includes abstractions for storage and transport of trace payloads. The combination of these parts avoid lock-in to a specific tracing backend. For example, Zipkin includes integration with different open source storage mechanisms like Apache Cassandra and Elasticsearch. It also includes bridges to convert collected data and forward it to service offerings such as Amazon X-Ray and Google Stackdriver. Ecosystem offering extend this portability further.

While primarily focused on the system, Zipkin also includes tracing libraries which applications use to report timing information. Zipkin's core organization includes tracer libraries written in Java, Javascript, Go, PHP and Ruby. These libraries use the formats mentioned above to report data, as well "B3" which is a header format needed to send trace identifiers along with production requests. Many Zipkin libraries can also send data directly to other services such as Amazon X-Ray and Google Stackdriver, skipping any Zipkin infrastructure. There are also more Zipkin tracing libraries outside the core organization than inside it. This is due to the "OpenZipkin" culture of promoting ecosystem work.

Background

Zipkin began in 2012 at Twitter during a time they were investigating performance problems underlying the "fail whale" seen by users. The name Zipkin is from the Turkish word for harpoon: the harpoon that will kill the failures! Incidentally, Zipkin was not the first tracing system, it had roots in a former system at Twitter named [BigBrotherBird](#). It is due to [BigBrotherBird](#) that the de-facto tracing headers we still use today include the prefix "X-B3".

In 2015, a community of users noticed the project was not healthy in so far as it hadn't progressed and often didn't accept pull requests, and the Cassandra backend was stuck on an unmaintained library. For example, the Apache Incubator H-Trace project started in some ways as a reaction to the inability to customize the code. The root cause of this was Twitter moving to internal storage (Manhattan) and also the project not being managed as a product. By mid 2015, the community regrouped as [OpenZipkin](#) and the codebase moved from Twitter to an org also named [OpenZipkin](#). This led to fast progress on concerns including initially a server rewrite and Docker based deployment.

In 2018, the second version of the data model completed, and along the way, many new libraries became standard, including javascript, golang and PHP. The community is dramatically larger than 2015, and Zipkin remains the most popular tracing system despite heavy competition.

Rationale

Zipkin is a de-facto distributed tracing system, which is more important as architectures become more fine grained due to popularity of microservice or even serverless architectures. Applications transition to use more complex communication including asynchronous code and service mesh, increasing the need for tools that visualize the behavior of requests as they map across an architecture.

Zipkin's server is focused only on distributed tracing. It is meant to be used alongside existing logging and metrics systems. Generally, the community optimizes brown field concerns such as interop over breaking changes such as experimental features. The combination of code and community make Zipkin a safe and easier choice for various sites to introduce or grow their observability practice.

Initial Goals

The initial goals are to mature [OpenZipkin](#)'s community process. For example, while [OpenZipkin](#) has a good collaborative process, it lacks formality around project management functions defined in the Apache Software Foundation (ASF). We also seek out help with brand abuse which is becoming common practice in the competitive landscape, yet demotivates volunteers. Towards volunteers, help with on boarding summer of code and funding for those who cannot afford to get to conferences on their own would be nice. Finally, we occasionally have organizations who are constrained to only work with foundation projects: ASF is often mentioned, and being in the ASF removes this collaboration roadblock.

Zipkin will not move all existing code into Apache. In fact, most Zipkin ecosystem exists outside our org! The goal is to start with the data formats and server code. Possibly the java client-side libraries can move initially as well, depending on community feedback.

Current Status

Meritocracy

Zipkin is an active community of contributors who are encouraged to become committers. A Zipkin committer understands the importance of seeking community feedback, and the gravity of brown field concerns. Committers express diverse interest by contributing beyond their sites immediate needs and acknowledging features require diverse need before being merged into the core repositories. A camaraderie between committers and not yet committers exists and is re-inforced with face to face meetups where possible. We expect this to continue and build with incubation and ideally acceptance into the Apache Software Foundation (ASF).

Zipkin encourages involvement from its community members, and the issues are open and available to any developers who wish to contribute to the project. The Zipkin team currently seeks help and asks for suggestions utilizing zipkin-user and zipkin-dev Google groups and Gitter chat on <https://gitter.im/openzipkin/zipkin>. While all contributions are reviewed, generally a "rule of three" policy on diverse need must be met before a feature is considered standard.

Community

Zipkin has a highly active and growing community of users and developers. The community is currently fostered on chat <https://gitter.im/openzipkin/zipkin> and issues in their respective [GitHub](#) repositories, notably the main server: <https://github.com/openzipkin/zipkin>

There are well over 1000 users in the chat room and hundreds who contributed code to code in the main [OpenZipkin GitHub](#) org. Interest metrics have grown dramatically: For example, in three years and a month from when Zipkin began until the time [OpenZipkin](#) formed, its main repository accumulated 2400 [GitHub](#) stars. In the same time after, it accumulated over 6700. Other metrics such as blog count and community meetings have similarly gone way up. We expect further growth as more learn about Zipkin and can engage with Zipkin through the guidance of the Apache Software Foundation (ASF).

Core Developers

The core contributors are a diverse group comprised of both unaffiliated developers and those hailing from small to large companies. They are scattered geographically, and some are highly experienced industry as well as open source developers. Though their backgrounds may be diverse, the contributors are united in their belief in community driven software development.

More detailed information on the core developers and contributors in general can be found under the section on homogeneous developers.

Alignment

Zipkin adoption is growing, and it is no longer feasible for it to remain as an isolated project. Apache is experienced in dealing with software that is very widely accepted and has a growing audience. The proposers believe that the Zipkin team can benefit from the ASF's experience and its broad array of users and developers.

Zipkin supports several Apache projects and options exist for integration with others. Apache CXF, Apache Camel, Apache Incubator [SkyWalking](#) and Apache Incubator HTrace all utilize Zipkin APIs in their core repositories. Many more do via community extensions. Apache Maven is primarily use by Zipkin, and can be used by projects who build upon Zipkin projects.

Known Risks

Orphaned products

Zipkin is already being utilized at multiple companies that are actively participating in improving the code. The thriving community centered around Zipkin has seen steady growth, and the project is gaining traction with developers. The risks of the code being abandoned are minimal.

Inexperience with Open Source

Zipkin rebooted its community in July 2015 and grown there for over three years. Additionally, many of the committers have extensive experience with other open source projects. Zipkin fosters a collaborative and community-driven environment.

In the interest of openly sharing technology and attracting more community members, several of our developers also regularly attend conferences in North America and Europe to give talks about Zipkin. Zipkin meetups are also planned every few months for developers and community members to come together in person and discuss ideas.

Homogenous Developers

At the time of the writing, [OpenZipkin's](#) core 12 developers all work at different companies around the globe. Most operate their own tracing sites, but some no longer operate sites at all: staying for the community we've built. Our ASF champion, Mick Semb Wever, is both a committer and an experienced ASF member.

The Zipkin developers thrive upon the diversity of the community. The Zipkin gitter channel is always active, and the developers often collaborate on fixes and changes in the code. They are always happy to answer users' questions as well.

Zipkin is interested in continuing to expand and strengthen its network of developers and community members through the ASF.

Reliance on Salaried Developers

Zipkin has one full time salaried developer, Adrian Cole. Though some of the developers are paid by their employer to contribute to Zipkin, many Zipkin developers contribute code and documentation on their own time and have done so for a lengthy period. Given the current stream of development requests and the committers' sense of ownership of the Zipkin code, this arrangement is expected to continue with Zipkin' induction into the ASF.

Relationships with Other Apache Products

Zipkin, Apache Incubator Skywalking and Apache Incubator HTrace address similar use cases. Most similarities are between Zipkin and HTrace: Zipkin hopes to help serve the community formerly served by HTrace, but understands the data services focus of HTrace may require different tooling. [SkyWalking](#) addresses more feature surface than Zipkin. For example, metrics collection is not a goal of Zipkin, yet it is a goal of [SkyWalking](#). [SkyWalking](#) accepts Zipkin formats and can be used as a replacement server. [SkyWalking](#) PPMC member, Sheng Wu, has been a routine member of Zipkin design discussions and has offered to help Zipkin through ASF process.

While Zipkin does not directly rely upon any Apache project, zipkin supports several Apache projects. Apache CXF, Apache Camel, Apache Incubator [SkyWalking](#), Apache Incubator Dubbo, Apache Incubator [ServiceComb](#) and Apache Incubator HTrace all utilize Zipkin APIs in their core repositories. Many more do via community extensions. Apache Maven is primarily used by Zipkin, and can be used by projects who build upon Zipkin projects.

A Excessive Fascination with the Apache Brand

Zipkin recognizes the fortitude of the Apache brand, but the motivation for becoming an Apache project is to strengthen and expand the Zipkin community and its user base. While the Zipkin community has seen steady growth over the past several years, association with the ASF is expected to expedite this pattern of growth. Development is expected to continue on Zipkin under the Apache license whether or not it is supported by the ASF.

Documentation

The Zipkin project documentation is publicly available at the following sites:

- <https://zipkin.io>: project overview
- <http://zipkin.io/zipkin-api/#/>: swagger specification
- <https://github.com/openzipkin/b3-propagation>: header formats
- <https://zipkin.io/zipkin/>: Javadocs for the Zipkin server

Initial Source

The initial source is located on [GitHub](#) in the following repositories:

- [git://github.com/OpenZipkin/zipkin.git](https://github.com/OpenZipkin/zipkin.git)
- [git://github.com/OpenZipkin/zipkin-dependencies.git](https://github.com/OpenZipkin/zipkin-dependencies.git)
- [git://github.com/OpenZipkin/zipkin-api.git](https://github.com/OpenZipkin/zipkin-api.git)
- [git://github.com/OpenZipkin/b3-propagation.git](https://github.com/OpenZipkin/b3-propagation.git)
- [git://github.com/OpenZipkin/docker-zipkin.git](https://github.com/OpenZipkin/docker-zipkin.git)
- [git://github.com/OpenZipkin/docker-zipkin-dependencies.git](https://github.com/OpenZipkin/docker-zipkin-dependencies.git)
- [git://github.com/openzipkin/zipkin-reporter-java](https://github.com/openzipkin/zipkin-reporter-java)
- [git://github.com/openzipkin/brave](https://github.com/openzipkin/brave)
- [git://github.com/openzipkin/zipkin-aws](https://github.com/openzipkin/zipkin-aws)
- [git://github.com/openzipkin/docker-zipkin-aws](https://github.com/openzipkin/docker-zipkin-aws)
- [git://github.com/openzipkin/zipkin-azure](https://github.com/openzipkin/zipkin-azure)
- [git://github.com/openzipkin/docker-zipkin-azure](https://github.com/openzipkin/docker-zipkin-azure)
- [git://github.com/openzipkin/zipkin-gcp](https://github.com/openzipkin/zipkin-gcp)
- [git://github.com/openzipkin/docker-zipkin-gcp](https://github.com/openzipkin/docker-zipkin-gcp)
- [git://github.com/openzipkin/brave-cassandra](https://github.com/openzipkin/brave-cassandra)
- [git://github.com/openzipkin/docker-jre-full](https://github.com/openzipkin/docker-jre-full)
- [git://github.com/openzipkin/brave-karaf](https://github.com/openzipkin/brave-karaf)

Depending on community progress, other repositories may be moved as well

Source and Intellectual Property Submission Plan

Zipkin's initial source is licensed under the Apache License, Version 2.0. <https://github.com/openzipkin/zipkin/blob/master/LICENSE>

All source code is copyrighted to 'The [OpenZipkin](#) Authors', to which the existing core community (members list in Initial Committers) has the rights to re-assign to the ASF.

External Dependencies

This is a listing of Maven coordinates for all of the external dependencies Zipkin uses. All of the dependencies are in Sonatype and their licenses should be accessible.

Cryptography

Zipkin contains no cryptographic algorithms.

Required Resources

Mailing Lists

- Zipkin-dev: for development discussions
- Zipkin-user: for community discussions
- Zipkin-private: for PPMC discussions
- Zipkin-commits: for code changes

Git Repositories

The Zipkin team is experienced in git and requests to transfer [GitHub](#) repositories(list in Initial Source) to Apache.

Issue Tracking

The community would like to continue using [GitHub](#) Issues.

Initial Committers

- Zoltán Nagy
- Adrian Cole, Pivotal
- Bas van Beek
- Brian Devins
- Eirik Sletteberg
- Jeanneret Pierre-Hugues
- Jordi Polo Carres
- José Carlos Chávez
- Kristof Adriaenssens
- Lance Linder
- Mick Semb Wever,
- Tommy Ludwig

Champion

- Michael Semb Wever, mck@apache.org

Mentors

- Michael Semb Wever, mck@apache.org
- Andriy Redko, reta@apache.org
- John D. Ament, johndament@apache.org
- Willem Ning Jiang, ningjiang@apache.org

Sponsoring Entity

We are requesting the Apache Incubator to sponsor this project.