

HTraceProposal

Abstract

HTrace is a tracing framework intended for use with distributed systems written in java.

Proposal

HTrace is an aid for understanding system behavior and for reasoning about performance issues in distributed systems. HTrace is primarily a low impedance library that a java distributed system can incorporate to generate 'breadcrumbs' or 'traces' along the path of execution, even as it crosses processes and machines. HTrace also includes various tools and glue for collecting, processing and 'visualizing' captured execution traces for analysis ex post facto of where time was spent and what resources were consumed.

Background

Distributed systems are made up of multiple software components running on multiple computers connected by networks. Debugging or profiling operations run over non-trivial distributed systems – figuring execution paths and what services, machines, and libraries participated in the processing of a request – can be involved.

Rationale

Rather than have each distributed system build its own custom 'tracing' libraries, ideally all would use a single project that provides necessary primitives and saves each project building its own visualizations and processing tools anew.

Google described "...[a] large-scale distributed systems tracing infrastructure" in Dapper, a Large-Scale Distributed Systems Tracing Infrastructure. The paper tells a compelling story of what is possible when disparate systems standardize on a single tracing library and cooperate, 'passing the baton', filling out trace context as executions cross systems.

HTrace aims to provide a rough equivalent in open source of the described core Dapper tools and library. As it is adopted by more projects, there will be a 'network effect' as HTrace will provide a more comprehensive view of activity on the cluster. For example, as HDFS gets HTrace support, we can connect this with the HTrace support in HBase to follow HBase requests as they enter HDFS.

Given the success of HTrace depends on its being integrated by many projects, HTrace should be perceived as unhampered, free of any commercial, political, or legal 'taint'. Being an Apache project would help in this regard.

Initial Goals

HTrace is a small project of narrow scope but with a grand vision:

- Move the HTrace source and repository to Apache, a vendor-neutral location. Currently HTrace resides at a Cloudera-hosted repository.
- Add past contributors as committers and institute Apache governance.
- Evangelize and encourage HTrace diffusion. Initially we will continue a focus on the Hadoop space since that is where most of the initial contributors work and it is where HTrace has been initially deployed.
- Building out the standalone visualization tool that ships with HTrace.
- Build more community and add more committers

Current Status

Currently HTrace has a viable Java trace library that can be interpolated to create 'traces'. The work that needs to be done on this library is mostly bug fixes, ease-of-use improvements, and performance tweaks. In the future, we may add libraries for other languages besides Java.

HTrace has means of dumping traces to the filesystem, Twitters' Zipkin (a tracing sink and visualization system developed by Twitter <https://github.com/twitter/zipkin>), or Apache HBase. Executions can be viewed either in Zipkin or in pygraph (<https://code.google.com/p/python-graph/>).

Since the initial sprint in the summer of 2012 which saw HTrace patches proposed for Apache HDFS and committed to Apache HBase, development has been sporadic; mostly a single developer or two adding a feature or bug fixing. HTrace is currently undergoing a new "spurt" of development with the effort to get HTrace added to Apache HDFS revived and a new standalone viewing facility being added in to HTrace itself.

HTrace has been integrated by Apache Phoenix.

Meritocracy

HTrace, up to this, has been run by Apache committers and PMC members. We want to build out a diverse developer and user community and run the HTrace project in the Apache way. Users and new contributors will be treated with respect and welcomed; they will earn merit in the project by tendering quality patches and support that move the project forward. Those with a proven support and quality patch track record will be encouraged to become committers.

Community

There are just a few developers involved at the moment. If our project is accepted by incubator, building community would be a primary initial goal.

Core Developers

Core developers include Apache members and members of the Hadoop and HBase PMCs. Of those listed, all have contributed to HTrace. Half are from Cloudera. The remainder are Hortonworks, NTTData, Google, and Facebook employees.

Alignment

HTrace has been integrated into Apache HBase and Apache Phoenix. Integration into Apache HDFS is currently being worked on. Approaching the Apache YARN project would be a likely next integration.

Known Risks

As noted above, development has been sporadic up to this. It may continue so.

For HTrace to tell a compelling story, it needs to be taken up by significant projects that make up a traced distributed system. For example, say YARN and HBase take on HTrace but HDFS does not, then the HDFS portions of an end-to-end operation will render opaque compromising our being able to tell a good story around an execution. Because the picture painted has gaps, HTrace may be left aside as ineffective.

Orphaned products

The proposers have a vested interest in making HTrace succeed, driving its development and its insertion into projects we all work on. Its dispersion will shine light on difficult to understand interactions amongst the various systems we all work on. A working, integrated HTrace will add a useful debugging mechanism to the Apache projects we all work on.

Inexperience with Open Source

The majority of the proposers here have day jobs that has them working near full-time on (Apache) open source projects. A few of us have helped carry other projects through incubator. HTrace to date has been developed as an open source project.

Homogenous Developers

The initial group of committers is small but already we have a healthy diversity of participating companies. We are bay-area challenged but a Japanese contributor makes for a good counter balance.

Reliance on Salaried Developers

Most of the contributors are paid to work in the Hadoop ecosystem. While we might wander from our current employers, we probably won't go far from the Hadoop tree. Whoever the Hadoop employer, it is plain a successful HTrace project is in everyone's interest. At least one of the developers has already changed employers but his interest in seeing HTrace succeed prevails.

Relationships with Other Apache Products

For HTrace to succeed, it is critical we build good relations with other distributed systems projects. We intend to initially build on relations we already have in place, mostly in the Hadoop space.

The HTrace project has been incorporated by Apache HBase and Apache Phoenix. It is currently being actively integrated into Apache HDFS.

We do not know of any equivalent or near-equivalent project in the Apache space.

The Dapper paper notes precedent, in particular, the Berkeley Rad Lab X-Trace project.

How HTrace relates to Zipkin

Zipkin is an Apache Licensed project from Twitter. It is a complete tracing tool with trace collectors, trace viewers and tools to help you generate traces. It is written in Scala. If your project is not Scala or if it is Java and you cannot afford a Scala dependency, at a minimum, you need an alternate means of generating traces. HTrace provides this facility for Java as well as bridging tools to feed traces to Zipkin for query and display.

The projects complement each other.

A Excessive Fascination with the Apache Brand

While we intend to leverage the Apache 'branding' when talking to other projects as testament of our project's 'neutrality', we have no plans for making use of Apache brand in press releases nor posting billboards advertising acceptance of HTrace into Apache Incubator.

Documentation

See htrace.org for the current state of the HTrace project and documentation.

How to enable tracing in [HBase using HTrace](#) Elliott Clark on [tracing in HBase](#)

Initial Source

Jonathan Leavitt and Todd Lipcon built the first versions of HTrace in the summer of 2012. Jonathan was Todd's summer intern at Cloudera.

Source and Intellectual Property Submission Plan

We know of no legal encumbrments in the way of transfer of source to Apache.

External Dependencies

HTrace includes third party libs. These include guava, jetty, junit, protobuf, hbase, and thrift. All dependencies are Apache licensed or licenses that are palatable: e.g. junit is EPL (Eclipse Public License v1.0) and [ProtoBufs](#) are BSD licensed.

Cryptography N/A

Required Resources

Mailing lists

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

Git Repository

<https://git-wip-us.apache.org/repos/asf/incubator-htrace.git>

Issue Tracking

JIRA HTrace (HTRACE)

Other Resources

Means of setting up regular builds for htrace on builds.apache.org

Initial Committers

- Colin [McCabe](#) (cmccabe@apache.org)
- Elliott Clark (eclark@apache.org)
- Jonathan Leavitt (jon.s.leavitt@gmail.com) – CLA being submitted
- Masatake Iwasaki (iwasakims@gmail.com) – CLA being submitted
- Michael Stack (stack@apache.org)
- Nick Dimiduk (ndimiduk@apache.org)
- Todd Lipcon (todd@apache.org)

Affiliations

- Colin [McCabe](#) - Cloudera
- Elliott Clark - Facebook
- Jonathan Leavitt - Google
- Masatake Iwasaki - NTTData
- Michael Stack - Cloudera
- Nick Dimiduk - Hortonworks
- Todd Lipcon - Cloudera

Sponsors

Champion

Roman Shaposhnik

Nominated Mentors

- Michael Stack - Apache Member
- Todd Lipcon - Apache Member
- Jake Farrell - Apache Member
- Billie Rinaldi - Apache Member
- Andrew Purtell - Apache Member
- Lewis John [McGibbney](#) - Apache Member

We will be soliciting more mentors as part of the proposal process.

Sponsoring Entity

We would like to propose Apache incubator to sponsor this project.