DruidProposal

Druid Proposal

Abstract

Druid is a high-performance, column-oriented, distributed data store.

Proposal

Druid is an open source data store designed for real-time exploratory analytics on large data sets. Druid's key features are a column-oriented storage layout, a distributed shared-nothing architecture, and ability to generate and leverage indexing and caching structures. Druid is typically deployed in clusters of tens to hundreds of nodes, and has the ability to load data from Apache Kafka and Apache Hadoop, among other data sources. Druid offers two query languages: a SQL dialect (powered by Apache Calcite) and a JSON-over-HTTP API.

Druid was originally developed to power a slice-and-dice analytical UI built on top of large event streams. The original use case for Druid targeted ingest rates of millions of records/sec, retention of over a year of data, and query latencies of sub-second to a few seconds. Many people can benefit from such capability, and many already have (see http://druid.io/druid-powered.html). In addition, new use cases have emerged since Druid's original development, such as OLAP acceleration of data warehouse tables and more highly concurrent applications operating with relatively narrower queries.

Background

Druid is a data store designed for fast analytics. It would typically be used in lieu of more general purpose query systems like Hadoop MapReduce or Spark when query latency is of the utmost importance. Druid is often used as a data store for powering GUI analytical applications.

The buzzwordy description of Druid is a high-performance, column-oriented, distributed data store. What we mean by this is:

- "high performance": Druid aims to provide low query latency and high ingest rates.
- "column-oriented": Druid stores data in a column-oriented format, like most other systems designed for analytics. It can also store indexes along with the columns.
- "distributed": Druid is deployed in clusters, typically of tens to hundreds of nodes.
- "data store": Druid loads your data and stores a copy of it on the cluster's local disks (and may cache it in memory). It doesn't query your data from some other storage system.

Rationale

Druid is a mature, active project with a large number of production installations, dozens of contributors to each release, and multiple vendors offering professional support. Given Druid's strong community, its close integration with many other Apache projects (such as Kafka, Hadoop, and Calcite), and its pre-existing Apache-inspired governance structure, we feel that Apache is the best home for the project on a long-term basis.

Current Status

Meritocracy

Since Druid was first open sourced the original developers have solicited contributions from others, including through our blog, the project mailing lists, and through accepting GitHub pull requests. We have an Apache-inspired governance structure with a PMC and committers, and our committer ranks include a good number of people from outside the original development team.

Community

The Druid core developers have sought to nurture a community throughout the life of the project. We use GitHub as the focal point for bug reports and code contributions, and the mailing lists for most other discussion. To try to make people feel welcome, we've also spelled this out on a "CONTRIBUTE" link from the project page: http://druid.io/community/. Today we have an active contributor base (a typical release has ~40 contributors) and mailing list.

Core Developers

Druid enjoys good diversity of committer affiliation. The most active developers over the past year are affiliated with four different companies: Imply, Metamarkets, Yahoo, and Hortonworks. Many Druid committers are also committers on other ASF projects as well, including Apache Airflow, Apache Curator, and Apache Calcite. The original developers of Druid remain involved in the project.

Alignment

Druid's current governance structure is Apache-inspired with a PMC and committers chosen by a meritocratic process. Additionally, Druid integrates with a number of other Apache projects, including Kafka, Hadoop, Hive, Calcite, Superset (incubating), Spark, Curator, and ZooKeeper.

Known Risks

Orphaned products

The risk of Druid becoming orphaned is low, due to a diverse committer base that is invested in the future of the project.

Inexperience with Open Source

Druid's core developers have been running it as a community-oriented open source project for some time now, and many of them are committers on other open source projects as well, including Apache Airflow, Apache Curator, and Apache Calcite.

Homogenous Developers

Druid's current diversity of committer affiliation means that we have become accustomed to working collaboratively and in the open. We hope that a transition to the ASF helps Druid's contributor base become even more diverse.

Reliance on Salaried Developers

Druid's user base and contributor base skews heavily towards salaried developers. We believe this is natural since Druid is a technology designed to be deployed on large clusters, and due to this, tends to be deployed by organizations rather than by individuals. Nevertheless, many current Druid developers have continued working on the project even through job changes, which we take to be a good sign of developer commitment and personal interest.

Relationships with Other Apache Products

Druid integrates with a number of other Apache projects. Druid internally uses Calcite for SQL planning, and Curator and ZooKeeper for coordination. Druid can read data in Avro or Parquet format. Druid can load data from streams in Kafka or from files in Hadoop. Druid integrates with Hive as an option for SQL query acceleration. Druid data can be visualized by Superset (incubating).

A Excessive Fascination with the Apache Brand

Druid is a successful project with a diverse community. The main reason for pursuing incubation is to find a stable, long term home for the project with a well known governance philosophy.

Required Resources

Mailing lists

We would like to migrate the existing Druid mailing lists from Google Groups to Apache.

- druid-user@googlegroups -> users@druid.incubator.apache.org
- druid-development@googlegroups -> dev@druid.incubator.apache.org

Source control

Druid development currently takes place on GitHub. We would like to continue using GitHub, if possible, in order to preserve the workflows the community has developed around GitHub pull requests.

Issue tracking

Druid currently uses GitHub issues for issue tracking. We would like to migrate to Apache JIRA at http://issues.apache.org/jira/browse/DRUID.

Documentation

Druid's documentation can be found at http://druid.io/docs/latest/.

Initial Source

Druid was initially open-sourced by Metamarkets in 2012 and has been run in a community-governed fashion since then. The code is currently hosted at https://github.com/druid-io/ and includes the following repositories:

- druid (primary repository)
- druid-console (web console for Druid)
- druid-io.github.io (source for Druid's website at http://druid.io/)
- tranquility (realtime stream push client for Druid)
- docker-druid (Docker image for Druid)
- pydruid (Python library)
- RDruid (R library)
- oss-parent (Maven POM files)

Source and Intellectual Property Submission Plan

A complete set of the open source code needs to be licensed from the owning organization to the Foundation. Commercial legal counsel for the owning organization will review the standard Foundation licensing paperwork and propose any updates as needed. This license will enable Apache to incubate and manage the Druid project moving forward.

Other Druid paraphernalia to be transferred to Apache consists of:

- GitHub organization at https://github.com/druid-io/
- Twitter account at https://twitter.com/druidio
- "druid.io" domain name
- "Druid" trademark assignment per Foundation standard paper. The trademark assignment paperwork shall be reviewed by the owning organization's commercial and IP counsel
- CLAs - all rights in the code licensed above should encompass the CLAs that existed between developers and owning organization

A copyright license to the code, trademark assignment of Druid, and transfer of other paraphernalia to Apache should be sufficient to cover all rights required by Apache to operate the project.

External Dependencies

External dependencies distributed with Druid currently all have one of the following Category A or B licenses: ASL, BSD, CDDL, EPL, MIT, MPL; with one exception: the optional Druid MySQL metadata store extension depends on MySQL Connector/J, which is GPL licensed. Druid currently packages this as a separate download; see our current presentation on: http://druid.io/downloads.html. As part of incubation we intend to determine the best strategy for handling the MySQL extension.

Cryptography

Not applicable.

Initial Committers

The initial committers for incubation are the current set of committers on Druid who have expressed interest in being involved in Apache incubation. Affiliations are listed where relevant. We may seek to add other committers during incubation; for example, we would want to add any current Druid committers who express an interest after incubation begins.

- Charles Allen (charles@allen-net.com) (Snap)
- David Lim (david.clarence.lim@gmail.com) (Imply)
- Eric Tschetter (cheddar@apache.org) (Splunk)
- Fangjin Yang (fj@imply.io) (Imply)
- Gian Merlino (gian@apache.org) (Imply)
- Himanshu Gupta (g.himanshu@gmail.com) (Oath)
- Jihoon Son (jihoonson@apache.org) (Imply)
- Jonathan Wei (jon.wei@imply.io) (Imply)
- Kurt Young (kurt@apache.org)
- Lijin Bin (binlijin@gmail.com) (Alibaba)
- Maxime Beauchemin (maxime.beauchemin@apache.org) (Lyft)
- Mohamed Slim Bouguerra (bslim@apache.org) (Hortonworks)
- Nishant Bangarwa (nishant@apache.org) (Hortonworks)
- Parag Jain (paragjain16@gmail.com) (Oath)
- Roman Leventov (leventov.ru@gmail.com) (Metamarkets)
- Xavier Léauté (xavier@leaute.com) (Confluent)

Sponsors

- Champion: Julian Hyde
- Nominated mentors: Julian Hyde, P. Taylor Goetz, Jun Rao
- Sponsoring entity: Apache Incubator