HudiProposal

Hudi Proposal

Abstract

Hudi is a big-data storage library, that provides atomic upserts and incremental data streams.

Hudi manages data stored in Apache Hadoop and other API compatible distributed file systems/cloud stores.

Proposal

Hudi provides the ability to atomically upsert datasets with new values in near-real time, making data available quickly to existing query engines like Apache Hive, Apache Spark, & Presto. Additionally, Hudi provides a sequence of changes to a dataset from a given point-in-time to enable incremental data pipelines that yield greater efficiency & latency than their typical batch counterparts. By carefully managing number of files & sizes, Hudi greatly aids both query engines (e.g: always providing well-sized files) and underlying storage (e.g: HDFS NameNode memory consumption).

Hudi is largely implemented as an Apache Spark library that reads/writes data from/to Hadoop compatible filesystem. SQL queries on Hudi datasets are supported via specialized Apache Hadoop input formats, that understand Hudi's storage layout. Currently, Hudi manages datasets using a combination of Apache Parquet & Apache Avro file/serialization formats.

Background

Apache Hadoop distributed filesystem (HDFS) & other compatible cloud storage systems (e.g: Amazon S3, Google Cloud, Microsoft Azure) serve as longer term analytical storage for thousands of organizations. Typical analytical datasets are built by reading data from a source (e.g: upstream databases, messaging buses, or other datasets), transforming the data, writing results back to storage, & making it available for analytical queries--all of this typically accomplished in batch jobs which operate in a bulk fashion on partitions of datasets. Such a style of processing typically incurs large delays in making data available to queries as well as lot of complexity in carefully partitioning datasets to guarantee latency SLAs.

The need for fresher/faster analytics has increased enormously in the past few years, as evidenced by the popularity of Stream processing systems like Apache Spark, Apache Flink, and messaging systems like Apache Kafka. By using updateable state store to incrementally compute & instantly reflect new results to queries and using a "tailable" messaging bus to publish these results to other downstream jobs, such systems employ a different approach to building analytical dataset. Even though this approach yields low latency, the amount of data managed in such real-time data-marts is typically limited in comparison to the aforementioned longer term storage options. As a result, the overall data architecture has become more complex with more moving parts and specialized systems, leading to duplication of data and a strain on usability.

Hudi takes a hybrid approach. Instead of moving vast amounts of batch data to streaming systems, we simply add the streaming primitives (upserts & incremental consumption) onto existing batch processing technologies. We believe that by adding some missing blocks to an existing Hadoop stack, we are able to a provide similar capabilities right on top of Hadoop at a reduced cost and with an increased efficiency, greatly simplifying the overall architecture in the process.

Hudi was originally developed at Uber (original name "Hoodie") to address such broad inefficiencies in ingest & ETL & ML pipelines across Uber's data ecosystem that required the upsert & incremental consumption primitives supported by Hudi.

Rationale

We truly believe the capabilities supported by Hudi would be increasingly useful for big-data ecosystems, as data volumes & need for faster data continue to increase. A detailed description of target use-cases can be found at https://uber.github.io/hudi/use_cases.html.

Given our reliance on so many great Apache projects, we believe that the Apache way of open source community driven development will enable us to evolve Hudi in collaboration with a diverse set of contributors who can bring new ideas into the project.

Initial Goals

- Move the existing codebase, website, documentation, and mailing lists to an Apache-hosted infrastructure.
- Integrate with the Apache development process.
- Ensure all dependencies are compliant with Apache License version 2.0.
- Incrementally develop and release per Apache guidelines.

Current Status

Hudi is a stable project used in production at Uber since 2016 and was open sourced under the Apache License, Version 2.0 in 2017. At Uber, Hudi manages 4000+ tables holding several petabytes, bringing our Hadoop warehouse from several hours of data delays to under 30 minutes, over the past two years. The source code is currently hosted at github.com (https://github.com/uber/hudi), which will seed the Apache git repository.

Meritocracy

We are fully committed to open, transparent, & meritocratic interactions with our community. In fact, one of the primary motivations for us to enter the incubation process is to be able to rely on Apache best practices that can ensure meritocracy. This will eventually help incorporate the best ideas back into the project & enable contributors to continue investing their time in the project. Current guidelines (https://uber.github.io/hudi/community.html#becoming-a-committer) have already put in place a meritocratic process which we will replace with Apache guidelines during incubation.

Community

Hudi community is fairly young, since the project was open sourced only in early 2017. Currently, Hudi has committers from Uber & Snowflake. We have a vibrant set of contributors (~46 members in our slack channel) including Shopify, DoubleVerify and Vungle & others, who have either submitted patches or filed issues with hudi pipelines either in early production or testing stages. Our primary goal during the incubation would be to grow the community and groom our existing active contributors into committers.

Core Developers

Current core developers work at Uber & Snowflake. We are confident that incubation will help us grow a diverse community in a open & collaborative way.

Alignment

Hudi is designed as a general purpose analytical storage abstraction that integrates with multiple Apache projects: Apache Spark, Apache Hive, Apache Hadoop. It was built using multiple Apache projects, including Apache Parquet and Apache Avro, that support near-real time analytics right on top of existing Apache Hadoop data lakes. Our sincere hope is that being a part of the Apache foundation would enable us to drive the future of the project in alignment with the other Apache projects for the benefit of thousands of organizations that already leverage these projects.

Known Risks

Orphaned products

The risk of abandonment of Hudi is low. It is used in production at Uber for petabytes of data and other companies (mentioned in community section) are either evaluating or in the early stage for production use. Uber is committed to further development of the project and invest resources towards the Apache processes & building the community, during incubation period.

Inexperience with Open Source

Even though the initial committers are new to the Apache world, some have considerable open source experience - Vinoth Chandar (Linkedin voldemort, Chromium), Prasanna Rajaperumal (Cloudera experience), Zeeshan Qureshi (Chromium) & Balaji Varadarajan (Linkedin Databus). We have been successfully managing the current open source community answering questions and taking feedback already. Moreover, we hope to obtain guidance and mentorship from current ASF members to help us succeed with the incubation.

Length of Incubation

We expect the project be in incubation for 2 years or less.

Homogenous Developers

Currently, the lead developers for Hudi are from Uber. However, we have an active set of early contributors/collaborators from Shopify, DoubleVerify and Vungle, that we hope will increase the diversity going forward. Once again, a primary motivation for incubation is to facilitate this in the Apache way.

Reliance on Salaried Developers

Both the current committers & early contributors have several years of core expertise around data systems. Current committers are very passionate about the project and have already invested hundreds of hours towards helping & building the community. Thus, even with employer changes, we expect they will be able to actively engage in the project either because they will be working in similar areas even with newer employers or out of belief in the project.

Relationships with Other Apache Products

To the best of our knowledge, there are no direct competing projects with Hudi that offer all of the feature set namely - upserts, incremental streams, efficient storage/file management, snapshot isolation/rollbacks - in a coherent way. However, some projects share common goals and technical elements and we will highlight them here. Hive ACID/Kudu both offer upsert capabilities without storage management/incremental streams. The recent Iceberg project offers similar snapshot isolation/rollbacks, but not upserts or other data plane features. A detailed comparison with their trade-offs can be found at https://uber.github.io/hudi/comparison.html.

We are committed to open collaboration with such Apache projects and incorporate changes to Hudi or contribute patches to other projects, with the goal of making it easier for the community at large, to adopt these open source technologies.

Excessive Fascination with the Apache Brand

This proposal is not for the purpose of generating publicity. We have already been doing talks/meetups independently that have helped us build our community. We are drawn towards Apache as a potential way of ensuring that our open source community management is successful early on so hudi can evolve into a broadly accepted ~~and used~~ method of managing data on Hadoop.

Documentation

[1] Detailed documentation can be found at https://uber.github.io/hudi/

Initial Source

The codebase is currently hosted on Github: https://github.com/uber/hudi . During incubation, the codebase will be migrated to an Apache infrastructure. The source code already has an Apache 2.0 licensed.

Source and Intellectual Property Submission Plan

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

External Dependencies

Non apache dependencies are listed below

- JCommander (1.48) Apache-2.0
- Kryo (4.0.0) BSD-2-Clause
- Kryo (2.21) BSD-3-Clause
- Jackson-annotations (2.6.4) Apache-2.0
- Jackson-annotations (2.6.5) Apache-2.0
- jackson-databind (2.6.4) Apache-2.0
- jackson-databind (2.6.5) Apache-2.0
- Jackson datatype: Guava (2.9.4) Apache-2.0
- docker-java (3.1.0-rc-3) Apache-2.0
- Guava: Google Core Libraries for Java (20.0) Apache-2.0
- bijection-avro (0.9.2) Apache-2.0
- com.twitter.common:objectsize (0.0.12) Apache-2.0
- Ascii Table (0.2.5) Apache-2.0
- config (3.0.0) Apache-2.0
- utils (3.0.0) Apache-2.0
- kafka-avro-serializer (3.0.0) Apache-2.0
- kafka-schema-registry-client (3.0.0) Apache-2.0
- Metrics Core (3.1.1) Apache-2.0
- Graphite Integration for Metrics (3.1.1) Apache-2.0
- Joda-Time (2.9.6) Apache-2.0
- JUnit CPL-1.0
- Awaitility (3.1.2) Apache-2.0
- jersey-connectors-apache (2.17) GPL-2.0-only CDDL-1.0
- jersey-container-servlet-core (2.17) GPL-2.0-only CDDL-1.0
- jersey-core-server (2.17) GPL-2.0-only CDDL-1.0
- htrace-core (3.0.4) Apache-2.0
- Mockito (1.10.19) MIT
- scalatest (3.0.1) Apache-2.0
- Spring Shell (1.2.0.RELEASE) Apache-2.0

All of them are Apache compatible

Cryptography

No cryptographic libraries used

Required Resources

Mailing lists

- private@hudi.incubator.apache.org (with moderated subscriptions)
- dev@hudi.incubator.apache.org
- commits@hudi.incubator.apache.org
- user@hudi.incubator.apache.org

Git Repositories

Git is the preferred source control system: git://git.apache.org/incubator-hudi

Issue Tracking

We prefer to use the Apache gitbox integration to sync Github & Apache infrastructure, and rely on Github issues & pull requests for community engagement. If this is not possible, then we prefer JIRA: Hudi (HUDI)

Initial Committers

- Vinoth Chandar (vinoth at uber dot com) (Uber)
- Nishith Agarwal (nagarwal at uber dot com) (Uber)
- Balaji Varadarajan (varadarb at uber dot com) (Uber)
- Prasanna Rajaperumal (prasanna dot raj at gmail dot com) (Snowflake)
- Zeeshan Qureshi (zeeshan dot qureshi at shopify dot com) (Shopify)
- Anbu Cheeralan (alunarbeach at gmail dot com) (DoubleVerify)

Sponsors

Champion

Julien Le Dem (julien at apache dot org)

Nominated Mentors

- Luciano Resende (lresende at apache dot org)
- Thomas Weise (thw at apache dot org
- Kishore Gopalakrishna (kishoreg at apache dot org)
- Suneel Marthi (smarthi at apache dot org)

Sponsoring Entity

The Incubator PMC