

KuduProposal

Kudu Proposal

Abstract

Kudu is a distributed columnar storage engine built for the Apache Hadoop ecosystem.

Proposal

Kudu is an open source storage engine for structured data which supports low-latency random access together with efficient analytical access patterns. Kudu distributes data using horizontal partitioning and replicates each partition using Raft consensus, providing low mean-time-to-recovery and low tail latencies. Kudu is designed within the context of the Apache Hadoop ecosystem and supports many integrations with other data analytics projects both inside and outside of the Apache Software Foundation.

We propose to incubate Kudu as a project of the Apache Software Foundation.

Background

In recent years, explosive growth in the amount of data being generated and captured by enterprises has resulted in the rapid adoption of open source technology which is able to store massive data sets at scale and at low cost. In particular, the Apache Hadoop ecosystem has become a focal point for such "big data" workloads, because many traditional open source database systems have lagged in offering a scalable alternative.

Structured storage in the Hadoop ecosystem has typically been achieved in two ways: for static data sets, data is typically stored on Apache HDFS using binary data formats such as Apache Avro or Apache Parquet. However, neither HDFS nor these formats has any provision for updating individual records, or for efficient random access. Mutable data sets are typically stored in semi-structured stores such as Apache HBase or Apache Cassandra. These systems allow for low-latency record-level reads and writes, but lag far behind the static file formats in terms of sequential read throughput for applications such as SQL-based analytics or machine learning.

Kudu is a new storage system designed and implemented from the ground up to fill this gap between high-throughput sequential-access storage systems such as HDFS and low-latency random-access systems such as HBase or Cassandra. While these existing systems continue to hold advantages in some situations, Kudu offers a "happy medium" alternative that can dramatically simplify the architecture of many common workloads. In particular, Kudu offers a simple API for row-level inserts, updates, and deletes, while providing table scans at throughputs similar to Parquet, a commonly-used columnar format for static data.

More information on Kudu can be found at the existing open source project website: <http://getkudu.io> and in particular in the Kudu white-paper PDF: <http://getkudu.io/kudu.pdf> from which the above was excerpted.

Rationale

As described above, Kudu fills an important gap in the open source storage ecosystem. After our initial open source project release in September 2015, we have seen a great amount of interest across a diverse set of users and companies. We believe that, as a storage system, it is critical to build an equally diverse set of contributors in the development community. Our experiences as committers and PMC members on other Apache projects have taught us the value of diverse communities in ensuring both longevity and high quality for such foundational systems.

Initial Goals

- Move the existing codebase, website, documentation, and mailing lists 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 per Apache guidelines

Current Status

Releases

Kudu has undergone one public release, tagged here <https://github.com/cloudera/kudu/tree/kudu0.5.0-release>

This initial release was not performed in the typical ASF fashion – no source tarball was released, but rather only convenience binaries made available in Cloudera's repositories. We will adopt the ASF source release process upon joining the incubator.

Source

Kudu's source is currently hosted on [GitHub](https://github.com/cloudera/kudu) at <https://github.com/cloudera/kudu>

This repository will be transitioned to Apache's git hosting during incubation.

Code review

Kudu's code reviews are currently public and hosted on Gerrit at <http://gerrit.cloudera.org:8080/#/q/status:open+project:kudu>

The Kudu developer community is very happy with gerrit and hopes to work with the Apache Infrastructure team to figure out how we can continue to use Gerrit within ASF policies.

Issue tracking

Kudu's bug and feature tracking is hosted on JIRA at: <https://issues.cloudera.org/projects/KUDU/summary>

This JIRA instance contains bugs and development discussion dating back 2 years prior to Kudu's open source release and will provide an initial seed for the ASF JIRA.

Community discussion

Kudu has several public discussion forums, linked here: <http://getkudu.io/community.html>

Build Infrastructure

The Kudu Gerrit instance is configured to only allow patches to be committed after running them through an extensive set of pre-commit tests and code lints. The project currently makes use of elastic public cloud resources to perform these tests. Until this point, these resources have been internal to Cloudera, though we are currently investing in moving to a publicly accessible infrastructure.

Development practices

Given that Kudu is a persistent storage engine, the community has a high quality bar for contributions to its core. We have a firm belief that high quality is achieved through automation, not manual inspection, and hence put a focus on thorough testing and build infrastructure to ensure that bar. The development community also practices review-then-commit for all changes to ensure that changes are accompanied by appropriate tests, are well commented, etc.

Rather than seeing these practices as barriers to contribution, we believe that a fully automated and standardized review and testing practice makes it easier for new contributors to have patches accepted. Any new developer may post a patch to Gerrit using the same workflow as a seasoned contributor, and the same suite of tests will be automatically run. If the tests pass, a committer can quickly review and commit the contribution from their web browser.

Meritocracy

We believe strongly in meritocracy in electing committers and PMC members. We believe that contributions can come in forms other than just code: for example, one of our initial proposed committers has contributed solely in the area of project documentation. We will encourage contributions and participation of all types, and ensure that contributors are appropriately recognized.

Community

Though Kudu is relatively new as an open source project, it has already seen promising growth in its community across several organizations:

- **Cloudera** is the original development sponsor for Kudu.
- **Xiaomi** has been helping to develop and optimize Kudu for a new production use case, contributing code, benchmarks, feedback, and conference talks.
- **Intel** has contributed optimizations related to their hardware technologies.
- **Dropbox** has been experimenting with Kudu for a machine monitoring use case, and has been contributing bug reports and product feedback.
- **Dremio** is working on integration with Apache Drill and exploring using Kudu in a production use case.
- Several community-built Docker images, tutorials, and blog posts have sprouted up since Kudu's release.

By bringing Kudu to Apache, we hope to encourage further contribution from the above organizations as well as to engage new users and contributors in the community.

Core Developers

Kudu was initially developed as a project at Cloudera. Most of the contributions to date have been by developers employed by Cloudera.

Many of the developers are committers or PMC members on other Apache projects.

Alignment

As a project in the big data ecosystem, Kudu is aligned with several other ASF projects. Kudu includes input/output format integration with Apache Hadoop, and this integration can also provide a bridge to Apache Spark. We are planning to integrate with Apache Hive in the near future. We also integrate closely with Cloudera Impala, which is also currently being proposed for incubation. We have also scheduled a hackathon with the Apache Drill team to work on integration with that query engine.

Known Risks

Orphaned Products

The risk of Kudu being abandoned is low. Cloudera has invested a great deal in the initial development of the project, and intends to grow its investment over time as Kudu becomes a product adopted by its customer base. Several other organizations are also experimenting with Kudu for production use cases which would live for many years.

Inexperience with Open Source

Kudu has been released in the open for less than two months. However, from our very first public announcement we have been committed to open-source style development:

- our code reviews are fully public and documented on a mailing list
- our daily development chatter is in a public chat room
- we send out weekly “community status” reports highlighting news and contributions
- we published our entire JIRA history and discuss bugs in the open
- we published our entire Git commit history, going back three years (no squashing)

Several of the initial committers are experienced open source developers, several being committers and/or PMC members on other ASF projects (Hadoop, HBase, Thrift, Flume, et al). Those who are not ASF committers have experience on non-ASF open source projects (Kiji, open-vm-tools, et al).

Homogenous Developers

The initial committers are employees or former employees of Cloudera. However, the committers are spread across multiple offices (Palo Alto, San Francisco, Melbourne), so the team is familiar with working in a distributed environment across varied time zones.

The project has received some contributions from developers outside of Cloudera, and is starting to attract a *user* community as well. We hope to continue to encourage contributions from these developers and community members and grow them into committers after they have had time to continue their contributions.

Reliance on Salaried Developers

As mentioned above, the majority of development up to this point has been sponsored by Cloudera. We have seen several community users participate in discussions who are hobbyists interested in distributed systems and databases, and hope that they will continue their participation in the project going forward.

Relationships with Other Apache Products

Kudu is currently related to the following other Apache projects:

- Hadoop: Kudu provides [MapReduce](#) input/output formats for integration
- Spark: Kudu integrates with Spark via the above-mentioned input formats, and work is progressing on support for Spark Data Frames and Spark SQL.

The Kudu team has reached out to several other Apache projects to start discussing integrations, including Flume, Kafka, Hive, and Drill.

Kudu integrates with Impala, which is also being proposed for incubation.

Kudu is already collaborating on [ValueVector](#), a proposed TLP spinning out from the Apache Drill community.

We look forward to continuing to integrate and collaborate with these communities.

An Excessive Fascination with the Apache Brand

Many of the initial committers are already experienced Apache committers, and understand the true value provided by the Apache Way and the principles of the ASF. We believe that this development and contribution model is especially appropriate for storage products, where Apache’s community-over-code philosophy ensures long term viability and consensus-based participation.

Documentation

- Documentation is written in [AsciiDoc](#) and committed in the Kudu source repository:
- <https://github.com/cloudera/kudu/tree/master/docs>
- The Kudu web site is version-controlled on the ‘gh-pages’ branch of the above repository.
- A [LaTeX](#) whitepaper is also published, and the source is available within the same repository.
- APIs are documented within the source code as [JavaDoc](#) or C++-style documentation comments.
- Many design documents are stored within the source code repository as text files next to the code being documented.

Source and Intellectual Property Submission Plan

The Kudu codebase and web site is currently hosted on [GitHub](#) and will be transitioned to the ASF repositories during incubation. Kudu is already licensed under the Apache 2.0 license.

Some portions of the code are imported from other open source projects under the Apache 2.0, BSD, or MIT licenses, with copyrights held by authors other than the initial committers. These copyright notices are maintained in those files as well as a top-level NOTICE.txt file. We believe this to be permissible under the license terms and ASF policies, and confirmed via a recent thread on general@incubator.apache.org.

The “Kudu” name is not a registered trademark, though before the initial release of the project, we performed a trademark search and Cloudera's legal counsel deemed it acceptable in the context of a data storage engine. There exists an unrelated open source project by the same name related to deployments on Microsoft's Azure cloud service. We have been in contact with legal counsel from Microsoft and have obtained their approval for the use of the Kudu name.

Cloudera currently owns several domain names related to Kudu (getkudu.io, kududb.io, et al) which will be transferred to the ASF and redirected to the official page during incubation.

Portions of Kudu are protected by pending or published patents owned by Cloudera. Given the protections already granted by the Apache License, we do not anticipate any explicit licensing or transfer of this intellectual property.

External Dependencies

The full set of dependencies and licenses are listed in <https://github.com/cloudera/kudu/blob/master/LICENSE.txt>

and summarized here:

- **Twitter Bootstrap**: Apache 2.0
- **d3**: BSD 3-clause
- **epoch JS library**: MIT
- **lz4**: BSD 2-clause
- **gflags**: BSD 3-clause
- **glog**: BSD 3-clause
- **gperftools**: BSD 3-clause
- **libev**: BSD 2-clause
- **squeasel**: MIT license
- **protobuf**: BSD 3-clause
- **rapidjson**: MIT
- **snappy**: BSD 3-clause
- **trace-viewer**: BSD 3-clause
- **zlib**: zlib license
- **llvm**: University of Illinois/NCSA Open Source (BSD-alike)
- **bitshuffle**: MIT
- **boost**: Boost license
- **curl**: MIT
- **libunwind**: MIT
- **nvml**: BSD 3-clause
- **cyrus-sasl**: Cyrus SASL license (BSD-alike)
- **openssl**: OpenSSL License (BSD-alike)
- **Guava**: Apache 2.0
- **StumbleUpon Async**: BSD
- **Apache Hadoop**: Apache 2.0
- **Apache log4j**: Apache 2.0
- **Netty**: Apache 2.0
- **slf4j**: MIT
- **Apache Commons**: Apache 2.0
- **murmur**: Apache 2.0

Build/test-only dependencies:

- **CMake**: BSD 3-clause
- **gcovr**: BSD 3-clause
- **gmock**: BSD 3-clause
- **Apache Maven**: Apache 2.0
- **JUnit**: EPL
- **Mockito**: MIT

Cryptography

Kudu does not currently include any cryptography-related code.

Required Resources

Mailing lists

- private@kudu.incubator.apache.org (PMC)
- commits@kudu.incubator.apache.org (git push emails)
- issues@kudu.incubator.apache.org (JIRA issue feed)
- dev@kudu.incubator.apache.org (Gerrit code reviews plus dev discussion)
- user@kudu.incubator.apache.org (User questions)

Repository

- [git://git.apache.org/kudu](https://git.apache.org/kudu)

Gerrit

We hope to continue using Gerrit for our code review and commit workflow. The Kudu team has already been in contact with Jake Farrell to start discussions on how Gerrit can fit into the ASF. We know that several other ASF projects and podlings are also interested in Gerrit.

If the Infrastructure team does not have the bandwidth to support Gerrit, we will continue to support our own instance of Gerrit for Kudu, and make the necessary integrations such that commits are properly authenticated and maintain sufficient provenance to uphold the ASF standards (e.g. via the solution adopted by the AsterixDB podling).

Issue Tracking

We would like to import our current JIRA project into the ASF JIRA, such that our historical commit messages and code comments continue to reference the appropriate bug numbers.

Initial Committers

- Adar Dembo adar@cloudera.com
- Alex Feinberg alex@strlen.net
- Andrew Wang wang@apache.org
- Dan Burkert dan@cloudera.com
- David Alves dralves@apache.org
- Jean-Daniel Cryans jdcryans@apache.org
- Mike Percy mpercy@apache.org
- Misty Stanley-Jones misty@apache.org
- Todd Lipcon todd@apache.org

The initial list of committers was seeded by listing those contributors who have contributed 20 or more patches in the last 12 months, indicating that they are active and have achieved merit through participation on the project. We chose not to include other contributors who either have not yet contributed a significant number of patches, or whose contributions are far in the past and we don't expect to be active within the ASF.

Affiliations

- Adar Dembo - Cloudera
- Alex Feinberg - Forward Networks
- Andrew Wang - Cloudera
- Dan Burkert - Cloudera
- David Alves - Cloudera
- Jean-Daniel Cryans - Cloudera
- Mike Percy - Cloudera
- Misty Stanley-Jones - Cloudera
- Todd Lipcon - Cloudera

Sponsors

Champion

- Todd Lipcon

Nominated Mentors

- Jake Farrell - ASF Member and Infra team member, Acquia
- Brock Noland - ASF Member, [StreamSets](#)
- Michael Stack - ASF Member, Cloudera
- Jarek Jarcec Cecho - ASF Member, Cloudera
- Chris Mattmann - ASF Member, NASA JPL and USC
- Julien Le Dem - Incubator PMC, Dremio
- Carl Steinbach - ASF Member, [LinkedIn](#)

Sponsoring Entity

