BeamProposal

Apache Beam

Abstract

Apache Beam is an open source, unified model and set of language-specific SDKs for defining and executing data processing workflows, and also data ingestion and integration flows, supporting Enterprise Integration Patterns (EIPs) and Domain Specific Languages (DSLs). Dataflow pipelines simplify the mechanics of large-scale batch and streaming data processing and can run on a number of runtimes like Apache Flink, Apache Spark, and Google Cloud Dataflow (a cloud service). Beam also brings DSL in different languages, allowing users to easily implement their data integration processes.

Proposal

Beam is a simple, flexible, and powerful system for distributed data processing at any scale. Beam provides a unified programming model, a software development kit to define and construct data processing pipelines, and runners to execute Beam pipelines in several runtime engines, like Apache Spark, Apache Flink, or Google Cloud Dataflow. Beam can be used for a variety of streaming or batch data processing goals including ETL, stream analysis, and aggregate computation. The underlying programming model for Beam provides MapReduce-like parallelism, combined with support for powerful data windowing, and fine-grained correctness control.

Background

Beam started as a set of Google projects (Google Cloud Dataflow) focused on making data processing easier, faster, and less costly. The Beam model is a successor to MapReduce, FlumeJava, and Millwheel inside Google and is focused on providing a unified solution for batch and stream processing. These projects on which Beam is based have been published in several papers made available to the public:

- MapReduce - http://research.google.com/archive/mapreduce.html
- Dataflow model - http://www.vldb.org/pvldb/vol8/p1792-Akidau.pdf
- FlumeJava - http://research.google.com/pubs/pub35650.html
- MillWheel - http://research.google.com/pubs/pub41378.html

Beam was designed from the start to provide a portable programming layer. When you define a data processing pipeline with the Beam model, you are creating a job which is capable of being processed by any number of Beam processing engines. Several engines have been developed to run Beam pipelines in other open source runtimes, including a Beam runner for Apache Flink and Apache Spark. There is also a “direct runner”, for execution on the developer machine (mainly for dev/debug purposes). Another runner allows a Beam program to run on a managed service, Google Cloud Dataflow, in Google Cloud Platform. The Dataflow Java SDK is already available on GitHub, and independent from the Google Cloud Dataflow service. Another Python SDK is currently in active development.

In this proposal, the Beam SDKs, model, and a set of runners will be submitted as an OSS project under the ASF. The runners which are a part of this proposal include those for Spark (from Cloudera), Flink (from data Artisans), and local development (from Google); the Google Cloud Dataflow service runner is not included in this proposal. Further references to Beam will refer to the Dataflow model, SDKs, and runners which are a part of this proposal (Apache Beam) only. The initial submission will contain the already-released Java SDK; Google intends to submit the Python SDK later in the incubation process. The Google Cloud Dataflow service will continue to be one of many runners for Beam, built on Google Cloud Platform, to run Beam pipelines. Necessarily, Cloud Dataflow will develop against the Apache project additions, updates, and changes. Google Cloud Dataflow will become one user of Apache Beam and will participate in the project openly and publicly.

The Beam programming model has been designed with simplicity, scalability, and speed as key tenants. In the Beam model, you only need to think about four top-level concepts when constructing your data processing job:

- Pipelines - The data processing job made of a series of computations including input, processing, and output
- PCollections - Bounded (or unbounded) datasets which represent the input, intermediate and output data in pipelines
- PTransforms - A data processing step in a pipeline in which one or more PCollections are an input and output
- I/O Sources and Sinks - APIs for reading and writing data which are the roots and endpoints of the pipeline

Rationale

With Google Dataflow, Google intended to develop a framework which allowed developers to be maximally productive in defining the processing, and then be able to execute the program at various levels of latency/cost/completeness without re-architecting or re-writing it. This goal was informed by Google’s past experience developing several models, frameworks, and tools useful for large-scale and distributed data processing. While Google has previously published papers describing some of its technologies, Google decided to take a different approach with Dataflow. Google open-sourced the SDK and model alongside commercialization of the idea and ahead of publishing papers on the topic. As a result, a number of open source runtimes exist for Dataflow, such as the Apache Flink and Apache Spark runners.

We believe that submitting Beam as an Apache project will provide an immediate, worthwhile, and substantial contribution to the open source community. As an incubating project, we believe Dataflow will have a better opportunity to provide a meaningful contribution to OSS and also integrate with other Apache projects.

In the long term, we believe Beam can be a powerful abstraction layer for data processing. By providing an abstraction layer for data pipelines and processing, data workflows can be increasingly portable, resilient to breaking changes in tooling, and compatible across many execution engines, runtimes, and open source projects.

Initial Goals

We are breaking our initial goals into immediate (< 2 months), short-term (2-4 months), and intermediate-term (> 4 months).

Our immediate goals include the following:

- Plan for reconciling the Dataflow Java SDK and various runners into one project
- Plan for refactoring the existing Java SDK for better extensibility by SDK and runner writers
- Validating all dependencies are ASL 2.0 or compatible
- Understanding and adapting to the Apache development process

Our short-term goals include:

- Moving the newly-merged lists, and build utilities to Apache
- Start refactoring codebase and move code to Apache Git repo
- Continue development of new features, functions, and fixes in the Dataflow Java SDK, and Dataflow runners
- Cleaning up the Dataflow SDK sources and crafting a roadmap and plan for how to include new major ideas, modules, and runtimes
- Establishment of easy and clear build/test framework for Dataflow and associated runtimes; creation of testing, rollback, and validation policy
- Analysis and design for work needed to make Beam a better data processing abstraction layer for multiple open source frameworks and environments

Finally, we have a number of intermediate-term goals:

- Roadmapping, planning, and execution of integrations with other OSS and non-OSS projects/products
- Inclusion of additional SDK for Python, which is under active development

Current Status

Meritocracy

Dataflow was initially developed based on ideas from many employees within Google. As an ASL OSS project on GitHub, the Dataflow SDK has received contributions from data Artisans, Cloudera Labs, and other individual developers. As a project under incubation, we are committed to expanding our effort to build an environment which supports a meritocracy. We are focused on engaging the community and other related projects for support and contributions. Moreover, we are committed to ensure contributors and committers to Dataflow come from a broad mix of organizations through a merit-based decision process during incubation. We believe strongly in the Beam model and are committed to growing an inclusive community of Beam contributors.

Community

The core of the Dataflow Java SDK has been developed by Google for use with Google Cloud Dataflow. Google has active community engagement in the SDK GitHub repository (https://github.com/GoogleCloudPlatform/DataflowJavaSDK), on Stack Overflow (http://stackoverflow.com/questions/tagged/google-cloud-dataflow) and has had contributions from a number of organizations and individuals.

Everyday, Cloud Dataflow is actively used by a number of organizations and institutions for batch and stream processing of data. We believe acceptance will allow us to consolidate existing Dataflow-related work, grow the Dataflow community, and deepen connections between Dataflow and other open source projects.

Core Developers

The core developers for Dataflow and the Dataflow runners are:

- Frances Perry
- Tyler Akidau
- Davor Bonaci
- Luke Cwik
- Ben Chambers
- Kenn Knowles
- Dan Halperin
- Daniel Mills
- Mark Shields
- Craig Chambers
- Maximilian Michels
- Tom White
- Josh Wills
- Robert Bradshaw

Alignment

The Beam SDK can be used to create Beam pipelines which can be executed on Apache Spark or Apache Flink. Beam is also related to other Apache projects, such as Apache Crunch. We plan on expanding functionality for Beam runners, support for additional domain specific languages, and increased portability so Beam is a powerful abstraction layer for data processing.

Known Risks

Orphaned Products

The Dataflow SDK is presently used by several organizations, from small startups to Fortune 100 companies, to construct production pipelines which are executed in Google Cloud Dataflow. Google has a long-term commitment to advance the Dataflow SDK; moreover, Dataflow is seeing increasing interest, development, and adoption from organizations outside of Google.

Inexperience with Open Source

Google believes strongly in open source and the exchange of information to advance new ideas and work. Examples of this commitment are active OSS projects such as Chromium (https://www.chromium.org) and Kubernetes (http://kubernetes.io/). With Dataflow, we have tried to be increasingly open and forward-looking; we have published a paper in the VLDB conference describing the Dataflow model (http://www.vldb.org/pvldb/vol8/p1792-Akidau.pdf) and were quick to release the Dataflow SDK as open source software with the launch of Cloud Dataflow. Our submission to the Apache Software Foundation is a logical extension of our commitment to open source software.

Homogeneous Developers

The majority of committers in this proposal belong to Google due to the fact that Dataflow has emerged from several internal Google projects. This proposal also includes committers outside of Google who are actively involved with other Apache projects, such as Hadoop, Flink, and Spark. We expect our entry into incubation will allow us to expand the number of individuals and organizations participating in Dataflow development. Additionally, separation of the Dataflow SDK from Google Cloud Dataflow allows us to focus on the open source SDK and model and do what is best for this project.

Reliance on Salaried Developers

The Dataflow SDK and Dataflow runners have been developed primarily by salaried developers supporting the Google Cloud Dataflow project. While the Dataflow SDK and Cloud Dataflow have been developed by different teams (and this proposal would reinforce that separation) we expect our initial set of developers will still primarily be salaried. Contribution has not been exclusively from salaried developers, however. For example, the contrib directory of the Dataflow SDK (https://github.com/GoogleCloudPlatform/DataflowJavaSDK/tree/master/contrib) contains items from free-time contributors. Moreover, seperate projects, such as ScalaFlow (https://github.com/darkjh/scalaflow) have been created around the Dataflow model and SDK. We expect our reliance on salaried developers will decrease over time during incubation.

Relationship with other Apache products

Dataflow directly interoperates with or utilizes several existing Apache projects.

- Build
  - Apache Maven
- Data I/O, Libraries
  - Apache Avro
  - Apache Commons
- Dataflow runners
  - Apache Flink
  - Apache Spark

Beam when used in batch mode shares similarities with Apache Crunch; however, Beam is focused on a model, SDK, and abstraction layer beyond Spark and Hadoop (MapReduce.) One key goal of Beam is to provide an intermediate abstraction layer which can easily be implemented and utilized across several different processing frameworks.

An excessive fascination with the Apache brand

With this proposal we are not seeking attention or publicity. Rather, we firmly believe in the Beam model, SDK, and the ability to make Beam a powerful yet simple framework for data processing. While the Dataflow SDK and model have been open source, we believe putting code on GitHub can only go so far. We see the Apache community, processes, and mission as critical for ensuring the Beam SDK and model are truly community-driven, positively impactful, and innovative open source software. While Google has taken a number of steps to advance its various open source projects, we believe Beam is a great fit for the Apache Software Foundation due to its focus on data processing and its relationships to existing ASF projects.

Documentation

The following documentation is relevant to this proposal. Relevant portion of the documentation will be contributed to the Apache Beam project.

- Dataflow website: https://cloud.google.com/dataflow
- Dataflow programming model: https://cloud.google.com/dataflow/model/programming-model
- Codebases
  - Dataflow Java SDK: https://github.com/GoogleCloudPlatform/DataflowJavaSDK
  - Flink Dataflow runner: https://github.com/dataArtisans/flink-dataflow
  - Spark Dataflow runner: https://github.com/cloudera/spark-dataflow
- Dataflow Java SDK issue tracker: https://github.com/GoogleCloudPlatform/DataflowJavaSDK/issues
- google-cloud-dataflow tag on Stack Overflow: http://stackoverflow.com/questions/tagged/google-cloud-dataflow

Initial Source

The initial source for Beam which we will submit to the Apache Foundation will include several related projects which are currently hosted on the GitHub repositories:

- Dataflow Java SDK (https://github.com/GoogleCloudPlatform/DataflowJavaSDK)
- Flink Dataflow runner (https://github.com/dataArtisans/flink-dataflow)
- Spark Dataflow runner (https://github.com/cloudera/spark-dataflow)

These projects have always been Apache 2.0 licensed. We intend to bundle all of these repositories since they are all complimentary and should be maintained in one project. Prior to our submission, we will combine all of these projects into a new git repository.

Source and Intellectual Property Submission Plan

The source for the Dataflow SDK and the three runners (Spark, Flink, Google Cloud Dataflow) are already licensed under an Apache 2 license.

- Dataflow SDK - https://github.com/GoogleCloudPlatform/DataflowJavaSDK/blob/master/LICENSE
- Flink runner - https://github.com/dataArtisans/flink-dataflow/blob/master/LICENSE
- Spark runner - https://github.com/cloudera/spark-dataflow/blob/master/LICENSE

Contributors to the Dataflow SDK have also signed the Google Individual Contributor License Agreement (https://cla.developers.google.com/about/google-individual) in order to contribute to the project.

With respect to trademark rights, Google does not hold a trademark on the phrase “Dataflow.” Based on feedback and guidance we receive during the incubation process, we are open to renaming the project if necessary for trademark or other concerns.

External Dependencies

All external dependencies are licensed under an Apache 2.0 or Apache-compatible license. As we grow the Beam community we will configure our build process to require and validate all contributions and dependencies are licensed under the Apache 2.0 license or are under an Apache-compatible license.

Required Resources

Mailing Lists

We currently use a mix of mailing lists. We will migrate our existing mailing lists to the following:

- dev@beam.incubator.apache.org
- user@beam.incubator.apache.org
- private@beam.incubator.apache.org
- commits@beam.incubator.apache.org

Source Control

The Dataflow team currently uses Git and would like to continue to do so. We request a Git repository for Beam with mirroring to GitHub enabled.

- https://git-wip-us.apache.org/repos/asf/incubator-beam.git

Issue Tracking

We request the creation of an Apache-hosted JIRA. The Dataflow project is currently using both a public GitHub issue tracker and internal Google issue tracking. We will migrate and combine from these two sources to the Apache JIRA.

- Jira ID: BEAM

Initial Committers

- Aljoscha Krettek [aljoscha@apache.org]
- Amit Sela [amitsela33@gmail.com]
- Ben Chambers [bchambers@google.com]
- Craig Chambers [chambers@google.com]
- Dan Halperin [dhalperi@google.com]
- Davor Bonaci [davor@google.com]
- Frances Perry [fjp@google.com]

- James Malone [jamesmalone@google.com]
- Jean-Baptiste Onofré [jbonofre@apache.org]
- Josh Wills [jwills@apache.org]
- Kostas Tzoumas [kostas@data-artisans.com]
- Kenneth Knowles [klk@google.com]
- Luke Cwik [lcwik@google.com]
- Maximilian Michels [mxm@apache.org]
- Stephan Ewen [stephan@data-artisans.com]
- Tom White [tom@cloudera.com]
- Tyler Akidau [takidau@google.com]
- Robert Bradshaw [robertwb@google.com]

Additional Interested Contributors

- Debo Dutta [dedutta@cisco.com]
- Henry Saputra [hsaputra@apache.org]
- Taylor Goetz [ptgoetz@gmail.com]
- James Carman [james@carmanconsulting.com]
- Joe Witt [joewitt@apache.org]
- Vaibhav Gumashta [vgumashta@hortonworks.com]
- Prasanth Jayachandran [pjayachandran@hortonworks.com]
- Johan Edstrom [seijoed@gmail.com]
- Hugo Louro [hmclouro@gmail.com]
- Krzysztof Sobkowiak [krzys.sobkowiak@gmail.com]
- Jeff Genender [jgenender@apache.org]
- Edward J. Yoon [edward.yoon@samsung.com]
- Hao Chen [hao@apache.org]
- Byung-Gon Chun [bgchun@gmail.com]
- Charitha Elvitigala [charithcc@apache.org]

- Alexander Bezzubov [bzz@apache.org]
- Tsuyoshi Ozawa [ozawa@apache.org]
- Mayank Bansal [mabansal@gmail.com]
- Supun Kamburugamuve [supun@apache.org]
- Matthias Wessendorf [matzew@apache.org]
- Felix Cheung [felixcheung@apache.org]
- Ajay Yadava [ajay.yadav@inmobi.com]
- Liang Chen [chenliang613@huawei.com]
- Renaud Richardet [renaud (at) apache (dot) org]
- Bakey Pan [bakey1985@gmail.com]
- Andreas Neumann [anew@apache.org]
- Suresh Marru [smarru@apache.org]
- Hadrian Zbarcea [hzbarcea@gmail.com]

Affiliations

The initial committers are from six organizations. Google developed Dataflow and the Dataflow SDK, data Artisans developed the Flink runner, and Cloudera (Labs) developed the Spark runner.

- Cloudera
  - Tom White
- Data Artisans
  - Aljoscha Krettek
  - Kostas Tzoumas
  - Maximilian Michels
  - Stephan Ewen
- Google
  - Ben Chambers
  - Dan Halperin
  - Davor Bonaci
  - Frances Perry
  - James Malone
  - Kenneth Knowles
  - Luke Cwik
  - Tyler Akidau
  - Robert Bradshaw
- PayPal
  - Amit Sela
- Slack
  - Josh Wills
- Talend
  - Jean-Baptiste Onofré

Sponsors

Champion

- Jean-Baptiste Onofre [jbonofre@apache.org]

Nominated Mentors

- Jean-Baptiste Onofre [jbonofre@apache.org]
- Jim Jagielski [jim@apache.org]
- Venkatesh Seetharam [venkatesh@apache.org]
- Bertrand Delacretaz [bdelacretaz@apache.org]
- Ted Dunning [tdunning@apache.org]

Sponsoring Entity

The Apache Incubator