Beam YAML, Kafka and Iceberg User Accessibility

Proposal by Charles Nguyen

Context

Overview

The relatively new Beam YAML SDK was introduced in the spirit of making data processing easy, but it has gained little adoption for ML tasks and hasn't been widely used with Managed IOs such as Kafka and Iceberg. We want to address this gap by introducing new illustrative examples of ML use cases using the YAML SDK, Kafka and Iceberg.

Background

The YAML SDK was only introduced in Spring 2024 as Beam's first no-code SDK. It follows a declarative approach of defining a data processing pipeline using YAML, as opposed to other programming language specific SDKs. In general, it still has very few meaningful examples and documentation to go along with.

One of the key missing examples is ML workflows. The API for the YAML SDK includes MLTransform and RunInference in addition to the flexibility from using UDF and SchemaTransformProvider, and we should therefore be able to write pipelines with the SDK that targets ML tasks. Beam itself is very capable of ML workloads, and the new YAML SDK should be just as well capable in defining ML pipelines despite its simplicity.

There are not many examples of working with IOs for YAML SDK, in particular Kafka and Iceberg which are also Beam's Managed IOs. The idea of having source/sink as Managed IO is to allow pipeline's runner to manage them, and therefore simplifying the management of pipelines that make use of these Managed IOs. Being able to create pipelines with Kafka and Iceberg in YAML emphasizes more clearly Beam's goal of how straightforward it is to author and manage such complex pipelines.

Goal and Theme

The goal of Beam YAML is making it possible to author pipelines with ease without sacrificing the rich features that Beam offers. The YAML SDK should therefore gain a lot of adoption, even for more complex use cases like ML and integration with Kafka and Iceberg. The project deliverables of stronger examples and documentation for the SDK will contribute to that adoption. The benefit extends beyond Beam's community towards the larger big data community, where these illustrative examples can help new data engineers, data scientists and analysts to onboard faster and be more productive when authoring pipelines with YAML.

The project also aligns closely with the theme of Google Summer of Code this year, in particular ML.

Design

The proposal includes the 4 main use cases below. The proposed examples are general enough to allow for flexibility when it comes to choosing datasets to work with, but still demonstrate specific ML tasks and the integration with Kafka and Iceberg.

Streaming Inference

The following is a streaming inference pipeline, demonstrating Beam YAML capability to run inference pipeline on a stream of incoming data from a message queue like Kafka.

None

```
Data ---> Kafka ---> Beam ---> Kafka / Iceberg
```

Before getting into the pipeline, we first train a model using Tensorflow to make a prediction/classification. The Beam pipeline is as follow:

1. Read from Kafka stream
2. A PTransform for preprocessing the data
3. A RunInference to predict or classify data
4. Write back to Kafka (as a new topic) / or to an Iceberg table

The pipeline can be applied to a range of problems with similar inference tasks such as sentiment analysis and forecasting.
For sentiment analysis, the text data can be from social media or Youtube comments. The classification can therefore be generally thought of as either positive or negative. For the forecasting example we can use the NYC taxi fares dataset for fare pricing prediction.

Simple Batch Processing

The following is a simple batch pipeline, demonstrating the use case with Iceberg.

None

```
Iceberg ---> Beam ---> Iceberg
```

The set up for Iceberg will likely use Hadoop catalog and GCP for hosted data files. The dataset can be system logs found in [5], and the data processing logic will probably be simple grouping and aggregation transformations. Where it makes sense, we will aim to demonstrate partition write and dynamic read/write.

Change Data Capture

The following are two streaming inference pipelines, demonstrating Beam YAML capability to be used with Kafka and Iceberg together for CDC.

None

```
Data ---> Kafka ---> Beam (1) ---> Iceberg (1)

Iceberg (1) ---> Beam (2) ---> Kafka
```

The 1st pipeline is as follow:

1. Read from Kafka stream
2. A PTransform for processing
3. Write to Iceberg table

The 2nd pipeline is as follow:

1. Stream read from Iceberg CDC
2. A PTransform for processing
3. Write to another Iceberg table

The CDC example is possible with Beam's pipeline leveraging Iceberg's snapshots and table history features along with Kafka. This was recently made possible in Beam [6], and for Iceberg CDC stream read in YAML SDK it will require using SchemaTransformProvider. The dataset chosen should have the "transactional" characteristic with a lot of updates, such as air traffic or GitHub events.

Feature Engineering and Model Evaluation

The following are 2 batch pipelines, demonstrating Beam YAML capability to do feature engineering which is subsequently used for model evaluation, and its integration with Iceberg as feature store.

```
None
   /---> Beam (1) ---> Iceberg (1)
  /                 Beam (3)     \
Data                        ---> Beam (5) ---> Iceberg (3)
  \                 Beam (4)     /
   \---> Beam (2) ---> Iceberg (2)
```

The workflow starts by reading from the same dataset. Beam (1) and (2) each is as followed:

1. PTransforms for feature engineering
2. Write features to Iceberg table

Beam (1) and (2) are series of PTransforms performing feature engineering after reading the data, but differ in their logics of extracting/generating features. Their outputs (the features) are PCollections which are then written to Iceberg tables.

Before moving on to the 2nd pipeline, we train the two models with the engineered input features queried from Iceberg tables. Beam (3) and (4) each then performs the model evaluation:

1. A PTransform for model evaluation
2. Return result as PCollection for later downstream PTransforms

Beam (5) performs the comparison for which model, and hence which feature set, is better and update it to the main Iceberg table:

1. A Ptransform comparing which model is better
2. Read from the corresponding Iceberg table containing the better feature set, and update to the main Iceberg table
3. Delete the Iceberg table containing the worse-performing feature set

The data for this ML workflow will probably be the NYC taxi fares dataset, or something of similar nature, where it is easy to come up with new features for evaluating models.

Deliverables

Concrete pipeline examples and documentation for each of the use cases:

- Streaming Inference (Sentiment Analysis and Forecasting)
- Simple Batch Pipeline
- Change Data Capture
- Feature Engineering and Model Evaluation

Timeline

My availability is flexible with commitment of approx. 12 weeks, starting mid May 2025. There's currently no other plan other than participation in Google Summer of Code.

Task	**# Weeks**
Project Planning and Investigation - Review project scope and goals - Review relevant existing pipeline examples	1
Streaming Inference Example Part I - Sentiment Analysis	1.5
Streaming Inference Example Part II - Forecasting	1.5
Simple Batch Pipeline Example	1
Change Data Capture Example	2
Feature Engineering Example	1.5
Model Evaluation Example	1.5
Final Report, Documentation and Blog Write-up	1
Buffer Week	1

About Me

I'm currently an undergraduate senior majoring in Computer Science. I'm based in Canada (EDT).

Outside of academia, most of my experience is through internships where I was primarily on platform teams working with systems and infrastructure. Especially for the past year or so I have worked on a lot of data analytics and observability problems and have taken a great interest in these areas. I also have experience in open-source contributions before, adding a few new SQL functions to database projects CockroachDB and QuestDB. I have had an amazing time working with big data technology, and I've always wanted to work in ML space and with Kafka and Iceberg. It's a growth opportunity I'm really looking forward to.

LinkedIn | GitHub | Resume | Email

Community Engagement

My engagement in the Beam community has been in the following:
Mailing list - https://lists.apache.org/list?dev@beam.apache.org:2025-3:Charles%20Nguyen (March and February)
Communication - to dannymccormick@google.com, rosinha@google.com and xqhu@google.com
PR: https://github.com/apache/beam/pull/34142
Issue: https://github.com/apache/beam/issues/34242

References

[1] Beam YAML doc - https://beam.apache.org/releases/yamldoc/current/

[2] Beam YAML examples - https://github.com/apache/beam/tree/master/sdks/python/apache_beam/yaml/examples

[3] Beam Python examples - https://github.com/apache/beam/tree/master/sdks/python/apache_beam/examples

[4] Possible datasets - https://clickhouse.com/docs/getting-started/example-datasets

[5] System log datasets - https://github.com/logpai/loghub

[6] Iceberg CDC in Beam - https://github.com/apache/beam/issues/33092