March 22, 2018

DEBUGGING LARGE-SCALE DATA PIPELINES WITH CONSISTENT HASHING

Evgeny Skvortsov

Long Nguyen

DEBUGGING LARGE-SCALE DATA PIPELINES WITH CONSISTENT HASHING

Large-scale data (e.g., "big data") may be processed by a pipeline including a set of data processing elements connected in series. Such pipeline is also referred to as the large-scale data pipeline. The large-scale data pipeline may execute a machine learning model to provide a service, such as an online advertising service. For example, the large-scale data pipeline may process data about user devices (e.g., mobile phones) using the machine learning model to identify audiences that might be interested in an advertisement and provide the advertisement to the audiences. Multiple large-scale data pipelines may execute the same machine learning model to provide a variety of services, such as online advertising services, content streaming services, content sharing services, etc. These pipelines may provide different data sets. These data sets may overlap with each other sometimes. An analytic system may analyze data sets provided by the large-scale data pipelines for various purposes. The analytic system may need to report the same aggregate numbers for overlapping events (e.g., the same number of common users of a content streaming service and a content sharing service that might be interested in certain media content)) and debug the pipelines for correctness. For example, certain users of a content streaming service may also use a content sharing service. The analytic system may need to identify the same subset of the users as potential audiences of a video based on data sets provided by the content streaming service and data sets provided by the content sharing service. Debugging these pipelines may be a challenging task for a couple of reasons. First, input data sets may be different because the pipelines may handle read and write operations in various manners. Second, intermediate input/output sets may be different across the pipelines. Third, different machine learning models may roll out at different rates across the pipelines. Moreover, comparing the whole input/output data sets for every stage of the large-scale data pipelines may

be infeasible because each of the large-scale data pipelines processes a substantial amount of data (e.g., 100TB raw data each day). Such comparison may be time consuming. Random sampling will not work in this case.

Prior solutions have been proposed to debug large-scale data pipelines in ad-hoc ways. For example, the prior solutions trace through from the final output and track toward the beginning of the pipeline. This may be helpful for finding some bugs. However, debugging using these prior solutions may be very time consuming and may require efforts from many platforms providing various services (e.g., advertising services, content streaming services).

To address the above issues, we propose a mechanism for debugging large-scale data pipelines by sampling inputs and outputs of large-scale data with consistent hashing. This proposal solves the debugging and monitoring issues in systematic and efficient ways. The mechanism may ensure that multiple large-scale data pipelines produce the same output set given the same input set and the same machine learning model. As an example, an advertiser can setup a campaign in an advertising platform providing advertising service and serve this campaign using a content streaming platform. An analytical system implementing the mechanism may identify certain users of the content streaming platform as potential audiences that might be interested in the advertising service and may report the same audiences with respect to the campaign on the advertising platform.

The mechanism includes a method of debugging large-scale data pipelines. The method includes sampling inputs and outputs of large-scale data with consistent hashing. For example, the method performs consistent hashing for multiple large-scale data pipelines. Each of the large-scale data pipelines may execute a machine learning model. The method provides one or

more input data sets to the large-scale data pipelines. The large-scale data pipelines may produce output data sets by processing the input data sets using the machine learning model.

The method performs consistent hashing based on input identifiers associated with the input data sets and the output data sets provided by the large-scale data pipelines. The method also produces the same subset (e.g., a sample) of events including inputs/outputs/model for computing alignment. For example, the method computes a hash based on inputs. After computing the hash, the method takes the remainder of the hash value over the total hash slots, which can be a fixed constant (e.g. 10000) . If the ratio of the remainder of the hash value over total slots is less than a threshold (e.g., 1%), then the method produces the same input and output (e.g., produces the sample). The method may also utilize different hash functions and a complete dump of input/output for certain data sets.

The method tracks the alignment of input data sets and output data sets throughout the pipelines to identify any bugs and to determine exactly where misalignment is introduced. The method can ensure the same output set across different pipelines given the same input set and the same machine learning model. Furthermore, the method can monitor a set of alignments of both inputs and outputs across multiple large-scale data pipelines. As such, the mechanism can significantly reduce debugging time (e.g., from weeks to hours) and coordination between different pipelines compared to the prior solutions.

ABSTRACT

A mechanism is provided for debugging large-scale data pipelines by sampling inputs and outputs of the large-scale data with consistent hashing. The mechanism can ensure the same output set across different pipelines given the same input set and the same machine learning model. The mechanism computes consistent hashing based on inputs and produces a consistent sample (e.g., the same subset) of events in the input and output for computing alignment. The mechanism tracks the alignment of input and output sets throughout the pipeline to identify any bugs and to determine exactly where misalignment is introduced.

Keywords: debugging, sampling, consistent hashing, large-scale data, big data

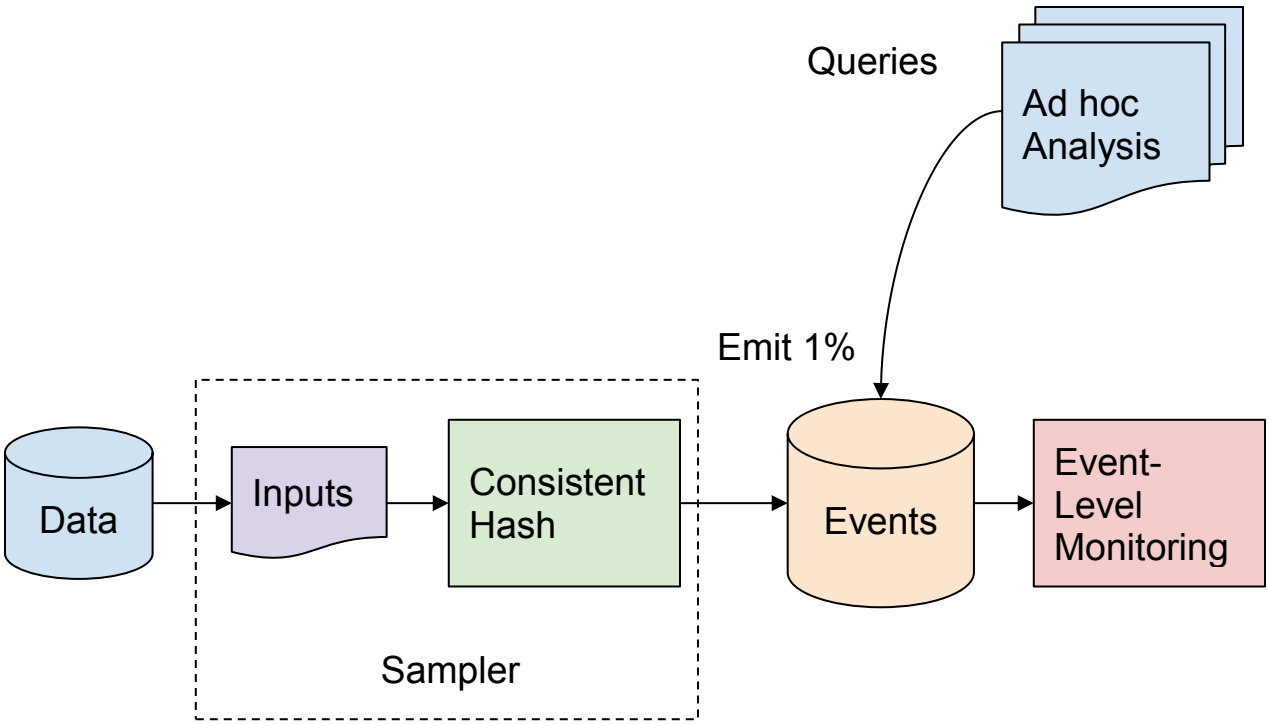

Figure 1 Consistently hash events, and chose the **same** set of 1% events across teams for debugging.