Pravega OLAP

Integration of Pravega and online analytical processing (OLAP) database with Java

Dell Technologies

Jose Robles, Nate Tsige, Boxiang Lin

9/23/2022

Table Of Contents

I. Introduction

Dell Technologies currently takes charge of an open-source project that is known as Pravega. This is an infrastructure that serves as a storage system that implements data streams to store/serve data [1]. These data streams are made up of sections which contain events. These are sets of bytes in a stream that represent some sort of data – Pravega is effective at storing/ingesting these due to its data streams being consistent, durable, elastic, and append-only [2].

Pravega stores ingested data from many different sources in a row-oriented manner which allows for all data points relating to one object to be stored in the same data block. This is beneficial for queries needing to read and manipulate an entire object, but it is slow to analyze large amounts of data. This is an issue because when we want to process events via big data analytics queries, efficiency is poor due to the row-oriented structure of Pravega. A column-oriented processing engine in which columns store similar data points for distinct objects within a block would allow for a quicker analysis of data points, as well as the compression of columns which is efficient for storing lots of data. Without ingesting Pravega events into a proper big data analytics engine, queries against the events are very slow and not feasible for a system storing as much data as Pravega.

II. Background and Related Work

Businesses are always looking for ways optimize their efficacy and to profit. One way to do that is to analyze data from the data source. In the early days, businesses were having difficulties as they navigated data due to the intense on-the-fly processing needed which resulted in the rise of OLAP databases. OLAPs pre-possess the data obtained from the source and store them. The processed data is instantly available for analysis.

One popular distributed event store and stream-processing platform is Apache Kafka. Kafka is a message-oriented middleware. While Kafka is great for transaction and event streams it lacks many futures that are necessary for modern data-intensive applications. Dell's Pravega further enhances programming models like Kafka and provides a cloud-native streaming infrastructure that enables a wider scope of applications by providing additional features like long-term retention, durability, auto-scale, and ingestion of large data to name a few [3].

However, Pravega is not an analytic engine hence it cannot process the data it ingests. The primary objective is to create plug-ins for either Apache Druid or Apache Pinot to enable integration to perform analysis of events from Pravega stream. This will allow users to make big data analytic queries on data that is passing through their stream.

In order to successfully provide a solution to the problem, the team will need some background knowledge of the problem space. A comprehensive understanding of Pravega and Apache Druid or Pinot is essential to solving the problem. In addition, the team will need to have a fundamental knowledge of data organization such as row-oriented and column-oriented databases, as well as experience developing in Java and familiarity with the SQL language.

III. Project Overview

Compared to similar streaming storage systems like Apache Kafka and Apache Pulsar, Pravega provides a more extended data retention period, auto-scaling of partition, and more [3]. However, since Pravega is a storage engine and not an analytics engine, streams don't get analyzed inside Pravega. Online analytical processing (OLAP) database is a software that enables users to quickly, consistently, and interactively observe information from all aspects to gain a deep understanding of data [4]. Hence, integrating Pravega with the OLAP database will allow users to perform log-based analysis against the stream data they stored in Pravega and therefore strengthen Pravega's ecological system, establishing a bridge between the storage engine to the analytics engine.

Integration of a storage engine and analytics engine for big data streaming is powerful and so beneficial as the needs for the Industrial Internet of Things, Internet of Vehicles, and real-time fraud risk control are developing rapidly. For technology to provide better services and customer experiences, we need applications to respond quickly to customer needs while still learning and adapting to changing behavior patterns. To be able to achieve that, an excellent streaming storage engine like Pravega and a great streaming analytics engine like Apache Druid are needed.

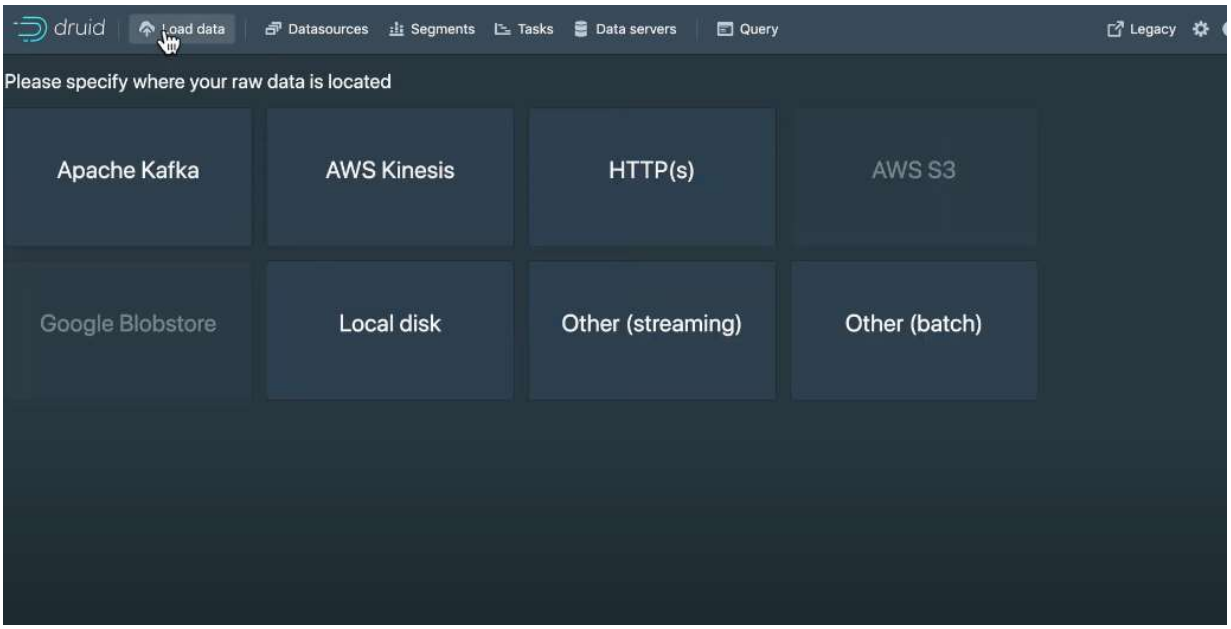
To integrate Pravega with Apache Druid, we will need to develop a plugin for the existing Apache Druid database system. This plugin will serve as a connector that first locates the source stream data in Pravega, then retrieves the stream data and transposes the data from row-oriented format to column-oriented format, and finally ingest them into the Apache Druid database. Since the Apache Druid and its provided client APIs are written in Java. We are expected to develop this plugin primarily using Java. We will also write some SQLs for data transposition and modify some configuration settings for Pravega connection.

We will develop this plugin project in Agile mode that cycles through processes of planning, executing, and evaluating. We are planning to have each sprint last for 3 weeks and each user story or use case to be tracked on GitHub Issue.

We will employ test-driven development (TDD) by having black-box test cases written before actual implementation and then finalizing the test cases after the actual implementation [5]. We might have functional test cases where we will try to validate a small amount of sample data by writing scripts to insert a few rows of sample data into Pravega and test the functionality of the plugin.

We will first develop the plugin in our local environment and version controlled by Git to our course remote repository. Once all test cases and validation are passed, we will then request a pull request to submit our plugin into Apache Druid GitHub repository.

Below is the screenshot of Apache Druid. As you can see on the “Load data” page there are plugins for Apache Kafka and AWS Kinesis stream storage system. We will develop a plugin for Pravega here.



IV. Client and Stakeholder Identification and Preferences

The primary client and stakeholder for this project is Dell Technologies, as they lead the development of the open-source distributed storage system Pravega. Stakeholders also include the JNB team comprised of WSU students that are working to develop this project, as well as the Pravega team at Dell and our Dell software engineer mentors.

The primary need of the Pravega team is for a plugin to be created that allows for an OLAP database to be integrated with Pravega such that the automatic ingestion of Pravega data streams can be enabled to be sent to an OLAP database. A stretch goal that has been highlighted is to produce plugins for both Apache Druid and Pinot (OLAP databases), but the primary goal is to produce just one plugin.

Other stake holders include companies that utilize Pravega to ingest data. Two companies that are public about their adoption of Pravega are Link Labs and Wheels up. Companies such as these will benefit from the new Pravega plugin that will allow for much more efficient big data analytic queries through the integration of an OLAP database.

V. Glossary

Auto scaling: A Pravega concept that allows the number of Stream Segments in a stream to change over time, based on scaling policy.
Big Data – refers to data sets that are too large or complex to be dealth with by traditional data processing software
Data streams – method of organizing data, has a data source and a destination. Comprised of stream segments which contain events
Events – a set of bytes contained within stream segments to represent some data point

OLAP – online analytical processing database
Pravega – open-source storage system implementing streams for storing/serving continuous and unbounded data.
Query – refers to a select query which retrieves data from a database and an action query that applies operations on the data

VI. References

[1] “Pravega Concepts” *Concepts - Exploring Pravega*. [Online]. Available: https://cncf.pravega.io/docs/nightly/pravega-concepts/#introduction. [Accessed: 23-Sep-2022].

[2] “Pravega Overview” *Exploring Pravega*. [Online]. Available: https://cncf.pravega.io/docs/v0.11.0/. [Accessed: 23-Sep-2022].

[3] Pravega, “A reliable stream storage system,” *Pravega*, 21-Mar-2022. [Online]. Available: https://cncf.pravega.io/. [Accessed: 23-Sep-2022].

[4] Foundation, A., 2022. Druid | Database for modern analytics applications. [online] Druid.apache.org. Available at: <https://druid.apache.org> [Accessed 23 September 2022].

[5] Hamilton, T., 2022. What is Test Driven Development (TDD)? Tutorial with Example. [online] Guru99. Available at: <https://www.guru99.com/test-driven-development.html> [Accessed 23 September 2022].