Master Thesis

Integrating Big Data Streams with PipelineDB

Prof. Dr. Burkhard Freitag
Chair of Information Management, University of Passau
http://www.im.uni-passau.de/

Tutor: Alexander Stenzer (Alexander.Stenzer@uni-passau.de)

April 30, 2019

Keywords: Database Systems, Sensor Data, Data Streams, Big Data, *PipelineDB*,

1 Introduction

The majority of today's Big Data is generated in the form of data streams [2], some of them delivering data in short bursts, i.e. at irregular intervals, others providing data in a more continuous way. These data streams have to be stored, managed and processed appropriately. In particular, the continuous computation of various aggregates, such as peak and average values or conspicuous anomalies, is highly relevant.

Pure relational database systems and data manipulation languages are usually not well-prepared for handling big data streams. However, the *PiplineDB* technology [3, 4, 6] which extends the PostgreSQL database software, provide some support of big data streams that can be exploited for database systems.

2 Objectives

The objective of this thesis is to **make viable the technology of big data streams** for information systems which are based on relational databases. To this end, a workbench supporting the integration of big data stream into the PostgreSQL database management system is to be designed and implemented that allows to combine stream-based queries and traditional relational queries.

In particular, stream data aggregation shall be supported, among others sliding max, min and average values, as well as time series functionality like group-by-hour, average by hour etc. Moreover, alerts shall be supported, based on data point outliers, e.g. data point above or below some threshold, and time series outliers, e.g. gradient above or below threshold. If time allows, also the visualization of the integrated stream data can be considered a sub-objective.

The thesis consists of a conceptual part and an implementation part. The implementation shall prototypically implement the described workbench. The required data aggregation and time series functionalities have to be verified based on realistic data such as the Open Data Server of the German National Weather Service [1], an artificial market orders stream [5], and data generated by the

Jmeter framework. The combination of stream-based queries and traditional relational queries shall be verified based on a structural representation of buildings and generated data that demonstrate the ability of the workbench to assign temperature data streams to specific locations within the building. To this end, an appropriate scenario will be provided. Beyond merely demonstrating that the required functionality has been successfully implemented its limits - e.g. how many streams can be processed in parallel, what is the maximum data rate per stream? - have to be explored and documented.

3 Tasks

Among the subtasks that have to be performed are:

- Study the literature (see below) as well as the *PiplineDB* and *Jmeter* frameworks;
- Design essential aggregation functionality such as sliding max, min and average values, time series functionality like group-by-hour, average by hour, and alert functionality based on data and gradient outliers;
- Design a software workbench supporting the integration of big data streams into the PostgreSQL database system;
- Implement the proposed solutions;
- Confirm the correctness and efficiency of the proposed solutions using the available data and scenarios (see above);
- Explore the limits of the proposed solution in terms of maximum number of parallel streams and maximum data rate per stream;
- Document the proposed solution;
- Demonstrate the proposed solution.

References

[1] Deutscher Wetterdienst (DWD). Open Data Server. Website, last visited 18 January 2019. https://www.dwd.de/EN/ourservices/opendata/opendata.html.

[2] R. Lax, S. Chernyak, and T. Akidau. *Streaming Systems*. O'Reilly Media, Inc., July 2018.

[3] PibelineDB. High-performance time-series aggregation for postgresql, last visited 01 March 2019. https://www.pipelinedb.com/.

[4] PipelineDB.com. *PipelineDB Documentation*, last visited 21 January 2019. http://docs.pipelinedb.com/.

[5] PubNub. Market orders. Website, last visited 18 January 2019. https://www.pubnub.com/developers/realtime-data-streams/financial-securities-market-orders/.

[6] W. Wingerath, N. Ritter, and F. Gessert. *Data Stream Management*, pages 43–55. Springer International Publishing, Cham, 2019.