Turkish Journal of Computer and Mathematics Education Vol.12 No.12 (2021), 3301-3305

Research Article

Improved Optimization and Speed up in Big Stream Data Processing

[1]Vivek Kumar, [2]Vinay K. Mishra, [3]Dilip K. Sharma

[1,2]Assistant Professor, [3]Professor

[1]Teerthanker Mahaveer University, Moradabad

[1]Dr. APJ Abdul Kalam Technical University, Lucknow

[2]Shri Ramswaroop Memorial Group of Professional Colleges, Lucknow

[3]GLA University, Mathura

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract

To provide the low latency, higher throughput & speedup in stream processing, there should be systematic flow design which can accept the continuous incoming stream and provide it tothe different operators to work parallel on the incoming stream. In this paper, we have implemented the proposed pipeline and watermark on Apache Beam along with google cloud dataflow as a runner. The experiments have been carried on stock market dataset, by considering the prices of oil, us dollar and gold as essential dependent parameters. Result of experiments proved that there is a relationship exist between the stock price and those dependent parameters. Now as the prediction of stock market essentially required other dependent parameter to be present which are originally from distributed environment, any parameter delay affect the result of prediction and introduced the good optimization and low latency. To implement the effective stream processing, we have used pipeline and watermark concept to handle and reduced any such delay and increase the speedup in big data stream processing.

Keywords: *Big Data, Optimization, Stream Processing, Speedup, latency*

1. Introduction

Massive and endless stream data processing is essential at the same time multifacetedin big data environment as the businesses process the data having precise time stamp and certainly to be processed in time order. On the arrival of new data, system should be able to cope with it and old data may be updated [1]. The batch processing model is incapable to satisfy the demands of real-time systems as big data streams to be processed with low- latency and in real time. Hence, stream processing model attempts to overcome the batch processing limitations by producing results with low-latency [2]. Data pipeline and watermark are essentially two significant methods to process big data stream. Streaming data pipelines signify a new edge in commercial technology. It handles a large number of events at scale, continuously. Subsequently, we can gather, break down, store and analyzea huge amount of data using pipeline. Watermark is a convenient and valuable technique that helps a stream processing system to deal with latency more specifically data latency. Following section represent real time stream data processing with different types of latency and the concept of pipeline and watermark which helps in reducing latency, and hence to provide higher-throughput.

2. Real Time Big Data Stream Processing

In real world, continuously arriving stream data needs to be processed as soon as it arrives. While attempting to process or investigate data in motion, it is essential to stay aware of the speed at which data is being swallowed into the system [4]. So when partial data which arrived into the system are in processing under processing unit of the system, others are in the arriving phase at the same time. There also exist some of the stream data which are arrived in to the system but yet not processed by the processing unit of the system. Below figure 1 captures the scenario presented here. As a result, in event-time based stream processing system, more precisely the one where prediction depends on multiple parameters which are independent, suffer from different types of latency and low throughput. The different types of latencies which slow down the big data stream processing are as follow:

- **Data Latency:** Data Latency represents the type of latency in system due to latedata arrival.

- **System Latency:** System latency is the current maximum duration that an item of data has been awaiting processing, in seconds.
- **Processing Latency:** Processing latency represents the time required by thesystem to process the data.

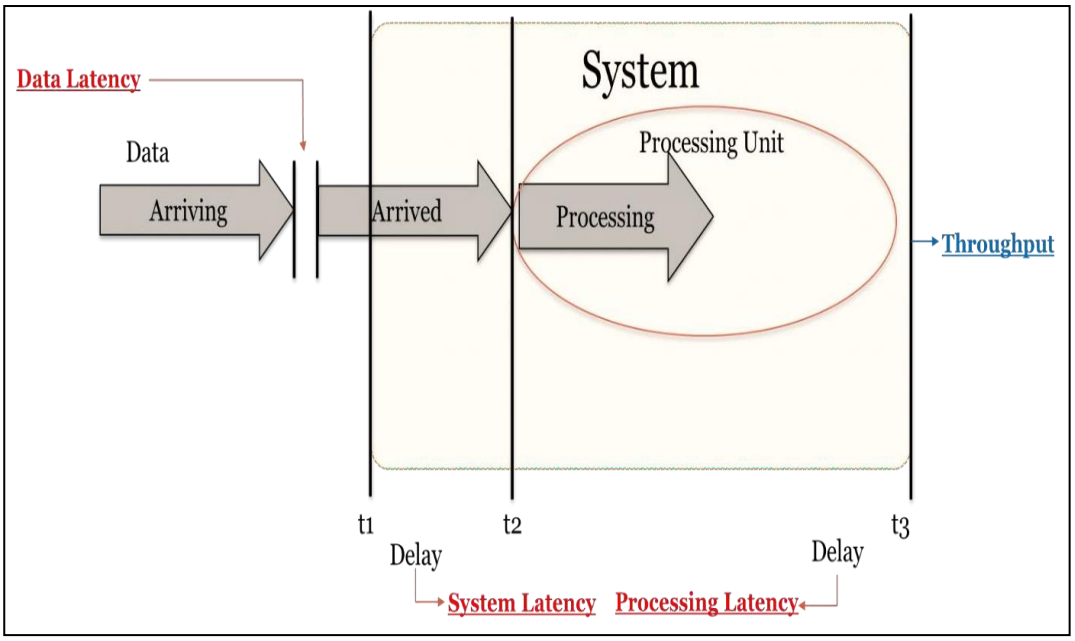


Figure 1. Different types of latency in stream processing system

However, with appropriate pipeline and watermark approach for processing big data stream, described in the following section, we can handle different types of latency and provide higher throughput for the unbounded data.

3. Pipeline in Big Data Stream Processing

A Pipeline captures complete data processing activity from beginning to end such as read and process given input data, converting data and writing output data [7]. A data pipeline sees all data as continuously arriving data. Irrespective of source of data, whetherit arises through static source or through real-time source, the pipeline splits every stream into reduced chunks which it processes simultaneously. Figure 2 shows the proposed pipeline designed to run on Google cloud Dataflow over Apache Beam.

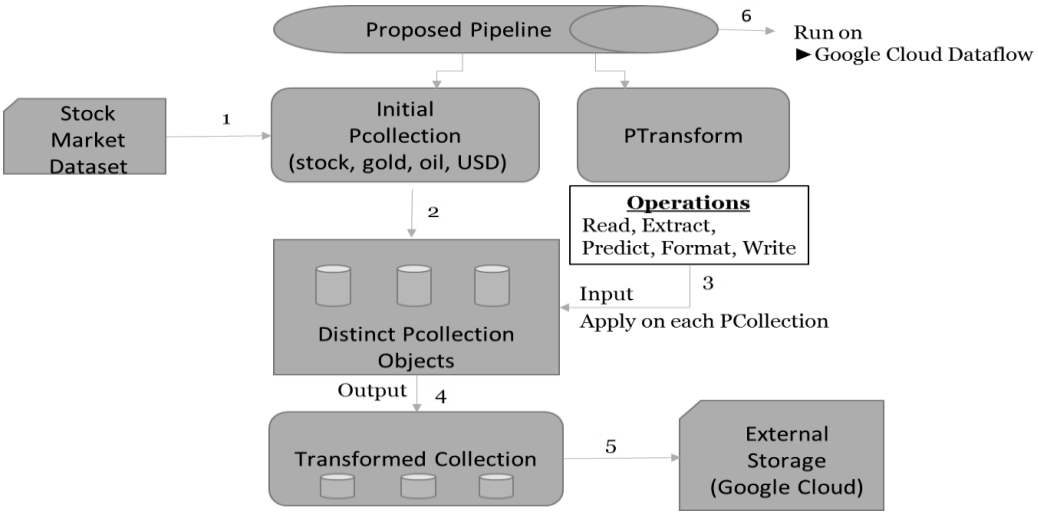


Figure 2. Proposed Pipeline

The Pipeline contains PCollection and PTransform. A PCollection signifies a disseminated input data on which pipeline of apache beam operates. We have conducted experiments on stock market dataset along with the price of gold, oil and USD as there is an interdependencies exist among them [5,6]. Pipeline classically generates a primary PCollection through reading data from an external data source [7]. That source can be bounded, meaning it comes as a static input like a file, or infinite, meaning itcomes from a constantly updated dynamic source. From there, PCollections are the inputs and outputs for each step in the pipeline.

A PTransform signifies functions and operations on data in pipeline. Every PTransform gets PCollection objects as an input, executes a specified operation thatwe would like to apply on the components of the PCollection and generates the PCollection objects as an output. Proposed pipeline has Read, Predict and Write operation under PTransform. As we have considered the input data set as distributed stream data, we have implemented watermark for input completeness in accordance with event time.

4. Watermark in distributed stream processing

Typically, in data processing system, there is a sure measure of slack between the time an event is arise and the real time at which data components are computed at any phase in pipeline. Furthermore, there are no surety that an event will show up in the pipeline in a similar sequence in which they were produced [7].

A watermark is a notion of input completeness with respect to event times [3]. Watermarks based on event times of the data can tell us about overall delay but, if we also compute watermarks based on the processing done by the system then these "system watermarks" allow us to distinguish data delay from system delay.

We have a PCollection that is utilizing fixed size window which is of 1 minute. For every window, Apache Beam should gather every one of the information with an event timestamp in the specified window range (somewhere in the range of 01:00:00 and 01:01:00 in the main window) [7]. Hence data having timestamps not specified in the range (data which is outside 01:01:01) have a place with an alternate window.

Nevertheless, data neither constantly ensured to land in a pipeline in a specified sequence, nor consistently land at predefined range. So, it uses watermark, the concept, which shows when entire data with specific window range can be predictable to land in a pipeline. As soon as the watermark advances the window's boundary, any additional components which landed in the same window will be measured as "late data" which introduce data latency.

5. Experiments

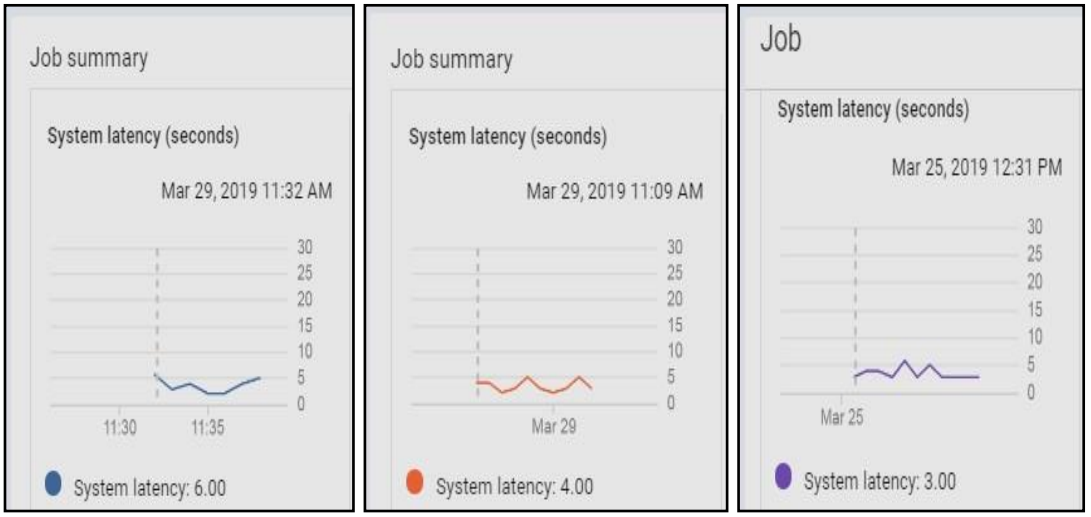
The experiment has been carried out on Apache Beam with Google cloud data flow runner. The following dataset of stock market along with USD price, gold price and Oil Price from [9, 10, 11] have been considered as input dataset to process with proposed pipeline presented in figure-2.

- Data: BSE Stock Data, USD, Gold Price, Oil Price [9, 10, 11]
- Attributes: Date, Open, High, Low, USD, G-Price, O-Price, Close
- Duration (year): 2000-2019.

The below graph shows the experiments on stock data prediction with and without USD, gold price and Oil Price. Comparative analysis proved that stock prediction is better when we consider the different independent parameters such as USD, gold price and Oil Price.

Figure 3. Comparative analysis of stock market prediction

As the independent parameters mentioned above, arriving from distributed environment, they may introduce different latency. We have performed experiments with different window size to deal with the system latency and from the experiments, we can see that as the size of window is changed then system latency will also be changed. There is no more variation in the experiments with window size more than hence window size 30 has been considered here in our case.



Window Size=1 **Window Size=10** **Window Size=30**

Figure 4. System Latency with Different Window Size [on Google CloudDataflow]

So, there is a dependency between the window size and system latency and the latency affected by size of window can be described by the following Polynomial function [8].

$Y = 5.938697318 \cdot 10\text{-}3\ X2 - 2.875478927 \cdot 10\text{-}1\ X + 6.281609195$

Where, Y= System Latency and X = Window Size.

6. Conclusion

In several crucial applications, it is expected to investigate and evaluate such streaming data in real time. One of the basic tasks of any streaming application is processing arriving data from scattered sources and generate an output promptly. The key deliberations for that desired task are: Latency and Throughput. Hence Dealing with stream imperfections such as late data, lost data and out of order data, becomes a significant research in big data stream processing. We have performed experiments for prediction on the stock market data, along with considering the price of US dollar, oil and gold as essential dependent parameters. Since the source of these dependent parameters are distributed, delay in any parameter introduce different types of latency and hence lower down the throughput of stream processing system

References

[1] Salem, Farouk. "Comparative Analysis of Big Data Stream Processing Systems." (2016).

[2] Akidau, Tyler. "The world beyond batch: Streaming 101." oreilly.com 20 (2016).

[3] Dasgupta, Triparna. "Evaluation of two major data stream processing technologies." (2016).

[4] Arfaoui, Mongi, and Aymen Ben Rejeb. "Oil, gold, US dollar and stock market interdependencies: a global analytical insight." European Journal of Management and Business Economics 26.3 (2017): pp. 278-293.

[5] Bedoui, Rihab, et al. "On the study of conditional dependence structure between oil, gold and USD exchange rates." International Review of Financial Analysis 59 (2018): 134-146.

[6] Beam, Apache. "Apache Beam programming guide." (2017).

[7] Polynomial Function, Retrieved from URL: http://xuru.org/rt/PR.asp.

[8] Federal Reserve Economic Data., Retrieved from URL: https://fred.stlouisfed.org.

[9] BSE Sensex 30 Historical Data, Retrieved from URL: https://in.investing.com/indices/sensex-historical-data.

[10] Gold Price Group, Retrieved from URL: https://goldprice.org/.

[11] S. Palaniappan, R. Awang, Intelligent heart disease prediction system using data mining techniques 2008 IEEE/ACS International Conference on Computer Systems and Applications, IEEE (2008), pp. 108-115