Real-Time Processing of Geo-Distributed Financial Data

Antonios Kontaxakis
Antonios Deligiannakis
{akontaxakis,adeli}@athenarc.gr
Athena Research Centre & Technical University of Crete

Holger Arndt
Stefan Burkard
{h.arndt,s.burkard}@springtechno.com
Spring Techno GmbH & Co. KG

Claus-Peter Kettner
Elke Pelikan
{cp.kettner,e.pelikan}@springtechno.com
Spring Techno GmbH & Co. KG

Kathleen Noack
k.noack@springtechno.com
Spring Techno GmbH & Co. KG

ABSTRACT

Enabling real-time processing of financial data streams is extremely challenging, especially considering that typical operations that interest investors often require combining data across (a potentially quadratic number of) different pairs of stocks. In this paper we present the architecture and the components of our system for the real-time processing of geo-distributed financial data at scale. Our system can scale to larger resources and utilizes a Synopses Data Engine in order to efficiently handle complex cross-stock queries, such as the ones required to detect systemic risk or to help forecast the value of some stock. The rich set of supported operations is depicted at the Visual Analytics component of our system.

CCS CONCEPTS

• **Computer systems organization → Real-time systems**.

KEYWORDS

Real-Time Processing, Financial Data, Flink

ACM Reference Format:
Antonios Kontaxakis, Antonios Deligiannakis, Holger Arndt, Stefan Burkard, Claus-Peter Kettner, Elke Pelikan, and Kathleen Noack. 2021. Real-Time Processing of Geo-Distributed Financial Data. In *The 15th ACM International Conference on Distributed and Event-based Systems (DEBS '21), June 28-July 2, 2021, Virtual Event, Italy*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3465480.3467842

1 INTRODUCTION

Processing Financial data, in order to provide real-time alerts and suggestions to users, poses immense challenges. The volume of data generated daily by NYSE alone reaches several terabytes, and these involve trades of thousands of stocks, and also views of the stocks/trades at different granularities (Level 1, Level 2 and Level 3 stock data). This data is produced at different locations around the world and needs to be processed in an interactive, online fashion for timely market surveillance or investment risk/opportunity identification purposes. Typical operations that interest investors often require combining data across different pairs of stocks. For instance, to detect systemic risks in the financial scenario, i.e., stock level events that could trigger instability or collapse of an entire industry or economy, requires discovering and interactively digging into correlations among tens of thousands of stock streams. Moreover, to forecast the future behavior of a given stock X often requires, as an initial step, the identification of stocks and past time windows over which these stocks exhibited similar behavior to the current trends of stock X. The problem with identifying the highly correlated pairs of stock data streams under various statistical measures, such as Pearson's correlation over N distinct, high speed data streams, is that it requires computing at real time $\Theta(N^2)$ correlation pairs.

In this work we describe our system for processing geo-distributed financial data streams. Our system is used by Spring Techno, an IT company that provides custom-tailored trading software and trading algorithms for the financial industry. In our description we place particular emphasis on the execution of data intensive and complex operations that combine data across different data streams/stocks, as single-stock queries (such as displaying the price of particular stocks over a time window) are trivial to handle.

2 OUR APPROACH

Figure 1 showcases the architecture of our system. Stock data streams are ingested through Kafka in our geo-distributed clusters, and forwarded to two components. The *Event-Detection* component processes the input simple events to detect and produce complex events, mainly involving individual stocks. Examples of such complex events include the detection of price swings (i.e., the value of a stock sharply increases or declines), or simple statistics over time windows (maximum or minimum stock price, total volume, etc) which can easily be computed using CEP.

To facilitate the computation of more complex events at scale, the input events are also forwarded and summarized using the Synopses Data Engine [2] (SDE) of the INFORE project (https://www.infore-project.eu/). The use of the INFORE SDE, implemented at Apache Flink [1], offers us, due to its design, the ability to achieve both horizontal scalability (i.e., the ability to scale the computation across multiple machines), vertical scalability (i.e., the ability to scale the computation over an increased number of input data streams), and federated scalability by maintaining the appropriate data synopses at each cluster ingesting data and answering queries of interest by

DEBS '21, June 28-July 2, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8555-8/21/06…$15.00
https://doi.org/10.1145/3465480.3467842

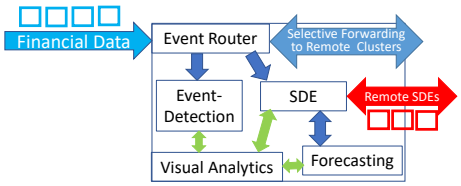

Figure 1: Architecture of our System - View of a Single Cluster

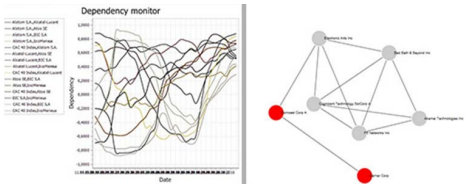

Figure 2: Real-Time Visual Analytics - Stock Correlation Monitoring for Systemic Risk Analysis

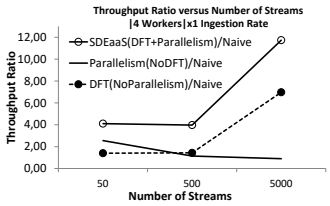

Figure 3: Comparative Analysis for the correlation computation using the DFT synopsis.

processing and communicating synopses data, rather than the actual (much larger) relevant data. The types of synopses maintained by the SDE is determined by queries that are registered to it. For example, pairs of stocks with strong correlations can be detected by either Locality-Sensitive Hashing (LSH) techniques or Discrete Fourier Transform (DFT)-based schemes [2] - both approaches reduce the number of examined stock pairs by focusing on stock pairs whose synopses are not distant in an appropriate metric space.

To enable fault-tolerance and to improve query response times, the data streams that are input to a cluster are also selectively forwarded to a subset of other clusters, based on the queries that clients have registered. This is especially useful for single-stock related queries. Data forwarded to other clusters are tagged, to ensure that the SDE does not process them multiple times at different clusters (this also ensures correctness in cases of event aggregations).

The Forecasting component is used to forecast the occurrence of complex events of interest, such as price swings, or the value of a stock over the short future. When required by the posed query, the Forecasting component may communicate with the SDE component, for example to collect the set of stocks whose behavior in the past closely matched the current behavior of a stock of interest and to then use this set in its forecasting algorithms.

Users register either one-shot or continuous queries through the *Visual Analytics* interface of Spring Techno. One-shot queries are executed once and their result is displayed immediately after their computation. Continuous queries are posed once and then executed continuously over specified time intervals (i.e., every 10 seconds) until their termination. Depending on the nature of each query, queries are forwarded to either the Event-Detection, the SDE or the Forecasting component. The output of any posed query is depicted at the Visual Analytics component of Spring Techno, which supports a variety of complex (one-shot or continuous) analytics. An example is shown in Figure 2, where the correlation between some stocks over time is monitored (on the left), and the strongly correlated stock pairs are also visualized (on the right).

3 EXPERIMENTS

We now provide a brief experimental evaluation of the scalability of our approach. We use a real dataset composed of 5000 stocks contributing a total of 10 TB of Level 1 and Level 2 data. We utilize a Kafka cluster with 3 Dell PowerEdge R320 Intel Xeon E5-2430 v2 2.50GHz machines with 32GB RAM each and one Dell PowerEdge R310 Quad Core Xeon X3440 2.53GHz machine with 16GB RAM. We evaluate the benefits of integrating into our system the virtues of data synopsis and parallel processing, using as a base scenario the identification of pairs of strongly correlated stocks, an operation used both at identifying systemic risk at the stock markets, as well as for forecasting the performance of a stock. In Figure 3 we measure the performance of our synopses-based approach against three alternative approaches. The compared approaches are:

- **Naive**: This is the baseline approach which involves sequential processing of incoming tuples without parallelism or any synopsis, and testing pairwise similarities among all stocks.
- **SDEaaS(DFT+Parallelism)**: This is the approach employed in this work which combines the virtues of parallel processing (using 4 workers in Figure 3) and stream summarization (DFT synopsis) towards delivering interactive analytics at extreme scale.
- **Parallelism(NoDFT)**: This approach performs parallel processing (4 workers), but does not utilize any synopses to bucketize time series or reduce their dimensionality.
- **DFT(NoParallelism)**: The DFT(NoParallelism) approach utilizes DFT synopses to bucketize time series, but no parallelism is used. Pairwise similarity checks are restricted to adjacent buckets and thus comparisons can be pruned, but the computation of similarities is not performed in parallel for each bucket.

Figure 3 displays the ratio of throughputs of each examined approach over the Naive approach varying the amount of monitored stock streams. It is obvious that our synopses-based approach achieves very significant benefits (over a 12-fold speedup) over the Naive approach for just 5000 stocks, even using just 4 workers.

ACKNOWLEDGMENTS

This work has received funding from the EU Horizon 2020 research and innovation program INFORE under grant agreement No 825070.

4 CONCLUSIONS

We presented the architecture of our system for the real-time processing of financial data. Our system consists of several components that interact with each other. A key feature of our system is the ability to scale to larger resources, by typically implementing several components at Flink, as well as to use a Synopses Data Engine in order to efficiently handle complex cross-stock queries, such as the ones required to detect systemic risk or to help forecast the value of some stock. The rich set of supported operations is depicted at the Visual Analytics component of our system.

REFERENCES

[1] Flink. [n.d.]. https://flink.apache.org/. [Online; accessed 13-April-2021].
[2] Antonis Kontaxakis, Nikos Giatrakos, and Antonios Deligiannakis. 2020. A Synopses Data Engine for Interactive Extreme-Scale Analytics. In *CIKM*.