SOFTWARE AND HARDWARE INFRASTRUCTURE FOR DATA STREAM PROCESSING

V.I. Protsenko[1,2], P.G. Seraphimovich[1,2], S.B. Popov[1,2], N.L. Kazanskiy[1,2]

[1] Image Processing Systems Institute - Branch of the Federal Scientific Research Centre "Crystallography and Photonics" of Russian Academy of Sciences, Samara, Russia
[2]Samara National Research University, Samara, Russia

Abstract. In this paper state-of-the-art hardware and software technologies for stream data processing are reviewed. IBM InfoSphere Streams and Apache Spark are among of the most popular software products that alleviates burden of distributed program development for data analysis tasks. Capabilities of these systems are considered in application to the time series analysis. IBM InfoSphere Streams turns to be more suitable for online processing, whereas Apache Spark time series library focuses on a bulk processing of big collections of the time series.

Keywords: data stream processing, high performance computing, time series analysis.

Citation: Protsenko VI, Seraphimovich PG, Popov SB, Kazanskiy NL. Firmware and hardware infrastructure for data stream processing. CEUR Workshop Proceedings, 2016; 1638: 782-787. DOI: 10.18287/1613-0073-2016-1638-782-787

1 Introduction

In the recent years business and scientific organizations faced the problem of development of analytic pipelines that could process large amounts of data in real-time and be able to seamlessly incorporate new data sources and new queries. Previous approach, that suggest to use conventional data bases, was not suited for real-time data analysis because of the need to store data before the processing. Priority in ACID principle in databases also constrains its ability to scale well on clusters of tenths to thousands of nodes. As a consequence, it is hard or impossible to process massive amounts of data in a fixed time. New approaches were proposed: Hadoop framework for bulk processing and dataflow graph based stream processing systems. Still a lot of research results in the data base field is now used in data stream processing systems: random sampling [1], aggregations [2, 3], join techniques [4, 5], query plan optimizers [6] and schedulers [7, 8, 9].

A trend of using commodity hardware started with projects Beowulf, Berkley NOW and HPVM is still present today. 10 Gigabit Ethernet is already wide spread and 100 Gigabit Ethernet is emerging [10], top x86_x64 processors for PC include up to 8 cores on one chip. Also according to performance benchmarks [11] commodity versions of GPU have similar computational power as HPC variants, but lack durability and support for high performance computing transport mechanisms. It turns out that clusters could achieve a remarkable performance gain by using GPU, moving responsibility of fault tolerance to the software framework like Apache Spark and IBM InfoSphere Streams. This is true also to recently introduced coprocessor Intel Phi [12] and FPGA units [13].

2 Data stream processing systems

2.1 Stream model

Data stream can be viewed as an sequence of elements $x_{1,}x_2, \ldots, x_n$ that has following properties:

- elements arrive continuously,
- number of the elements in the stream could be infinite.

Depending on the application processing can be done per element of in terms of windows. Windows are the rules according to which elements in the sequence are divided in subsequences that should be analyzed as a whole. Windows are commonly divided into two types: sliding window and tumbling window.

Tumbling windows are divided into 4 types according to the trigger policy that launches processing of the current window:

- count-based policy,
- delta-based policy depending on changing attribute,
- time based policy, according to the local or global time stamp,
- punctuation based policy.

Sliding windows have more types than tumbling windows. They have three processing trigger types and three evicting policy types, that together form nine variants. Evicting policy could be one of the following:

- count based,
- delta-based,
- time based.

Window semantics fits well time series analysis tasks and naturally matches wide range of applications. Syntax and semantics of stream query language is still a topic of the research [14-18].

2.2 IBM InfoSphere Streams

One of the most popular commercial system for data stream analysis is IBM InfoSphere Streams. Research on this system could be traced to 2008 when the first engine was presented [19]. Since then the project matured into the robust system that can be used in production for real-time text analysis, data extraction and financial insights. System also offers good capabilities in the time series analysis.

Time series could be represented in the system in two ways: like a sequence stream of elements or like a vector element of data stream. Moreover, each vector can be interpreted as a time slice of several time series, and in this case data stream multiplexes several time series.

The system bundle contains com.ibm.streams.timeseries toolkit with number of implemented algorithms for time series analysis. Among its capabilities are:

- anomaly detection,
- cross-correlation of two streams,
- times series normalization,
- DFT and DWT transforms,
- stream statistics evaluation,
- application of DSP to input sequence.

For short and long term predictions package has implementation of ARIMA model, Holt-Winters model, Kalman filter and multivariate autoregression model. Fig 1 shows an example of online predictions according to the ARIMA model.

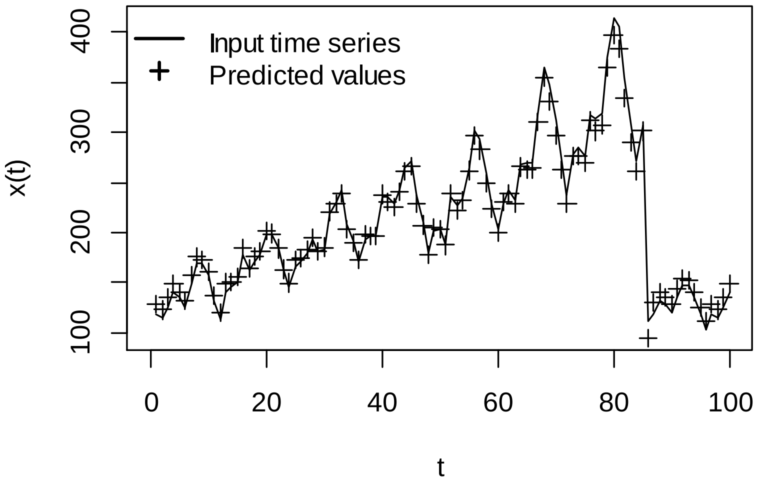

Fig. 1. Online ARIMA predictions one time unit ahead after the initial period of training

2.3 Apache Spark

Apache Spark is a new programming framework for distributed data processing. It has a good integration with Hadoop ecosystem. Basic part of the project offers an ability for bulk and stream processing in the terms of RDD [20] and DStream [21] abstractions. There are also additional projects: MLlib - a scalable machine learning library, graph processing library GraphX, Spark SQL module for structured data processing, SparkR module that enables integration with language R.

Time series analysis in Apache Spark is enabled by SparkTS library. Current implementation of the library includes following models: ARGARCH, ARIMA, EGARCH, EWMA, GARCH and AR. Also a number of statistical tests: augmented Dickey-Fuller, Breusch-Godfrey, Breusch-Pagan, Durbin-Watson, KPSS and Ljung-Box. Apache Spark makes available bulk and stream distributed processing over thousands of independent time series. However, it has no online processing algorithms as it is the case with IBM InfoSphere Streams.

An experiment conducted on 2 nodes with 32 cores (four E5-2450 processors) connected by 10Gb Ethernet shows that processing big amount of time series, 40000 in our case, can be reduced from 27 minutes to 1.2 minutes. Results are shown in the fig 2 and table 1.

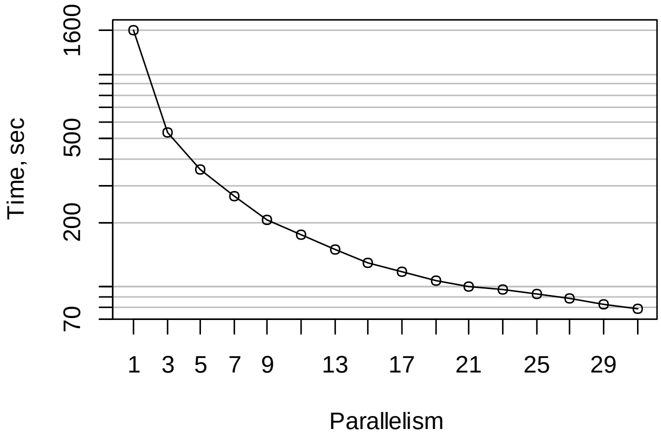

Fig. 2. Performance of Apache Spark on testing 40000 time series to be stationary with KPSS test

Table 1. Performance of Apache Spark on testing 40000 time series to be stationary with KPSS test

Parallelism	Time, seconds
1	1600
9	207
17	118
31	78

3 Conclusion

Data analysis is an interactive task. Because of increasing speed of accumulating data it start to be more a more important to use a lot of computational power to match sufficient response time of analytic queries. Moreover, a subset of queries could be answered in online fashion that can also decrease wait time. Apache Spark and IBM InfoSphere Stream reviewed systems achieves these by managing large number of nodes and are able to run data analysis tasks in distributed fashion. After the fixed period of ARIMA learning time IBM InfoSphere Streams was able to make short-term predictions in online fashion. Bulk processing time for test that time series is stationary in Apache Spark was decreased from 27 minutes to 1.2 minutes. This way data processing systems help to reveal the potential of commodity hardware and bring the ability of time series analysis on large amounts of data. Short-term and long-term predictions play important role in finance [22-23] and automatic control [24-26]. Elaborate design of IBM InfoSphere Streams system also allows to transform multimedia data [27,28] which result can be further piped into time series analysis operators.

References

1. Olken F. Random sampling from databases. University of California at Berkeley, 1993.
2. Hellerstein JM, Haas PJ, Wang HJ. Online aggregation. ACM SIGMOD Record. ACM, 1997; 26(2): 171-182.
3. Fang M et al. Computing Iceberg Queries Efficiently. Internaational Conference on Very Large Databases (VLDB'98), New York, August 1998. Stanford InfoLab, 1999.
4. Haas PJ, Hellerstein JM. Ripple joins for online aggregation. ACM SIGMOD Record. ACM, 1999; 28(2): 287-298.
5. Luo G et al. A scalable hash ripple join algorithm. Proceedings of the 2002 ACM SIGMOD international conference on Management of data. ACM, 2002: 252-262.
6. Graefe G, McKenna WJ. The Volcano optimizer generator: Extensibility and efficient search. Data Engineering, 1993. Proceedings Ninth International Conference on IEEE, 1993: 209-218.
7. Armbrust M et al. Spark sql: Relational data processing in spark. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM, 2015: 1383-1394.
8. Abadi DJ et al. Aurora: a new model and architecture for data stream management. The VLDB Journal – the International Journal on Very Large Data Bases, 2003; 12(2): 120-139.
9. Isard M et al. Dryad: distributed data-parallel programs from sequential building blocks. ACM SIGOPS Operating Systems Review. ACM, 2007; 41(3): 59-72.
10. Renaudier J et al. Transmission of 100Gb’s Coherent PDM-QPSK over 16x100km of standard fiber with allerbium amplifiers. Optics Express, 2009; 17(7): 5112-5119.
11. Bergdorf M et al. Desmond. GPU Performance as of October, 2015.
12. Reinders J. An overview of programming for Intel Xeon processors and Intel Xeon Phi coprocessors. Intel Corporation, Santa Clara, 2012.

13. Hauck S, DeHon A. Reconfigurable computing: the theory and practice of FPGA-based computation. Morgan Kaufmann, 2010.
14. Maier D et al. Semantics of data streams and operators. Database Theory-ICDT 2005. Springer Berlin Heidelberg, 2005: 37-52.
15. Krämer J, Seeger B. Semantics and implementation of continuous sliding window queries over data streams. ACM Transactions on Database Systems (TODS), 2009; 34(1): 4 p.
16. Li J et al. Semantics and evaluation techniques for window aggregates in data streams. Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM, 2005: 311-322.
17. Arasu A, Babu S, Widom J. The CQL continuous query language: semantic foundations and query execution. The VLDB Journal – the International Journal on Very Large Data Bases, 2006; 15(2): 121-142.
18. Beck H et al. LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. AAAI, 2015: 1431-1438.
19. Gedik B et al. SPADE: the system s declarative stream processing engine. Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008: 1123-1134.
20. Zaharia M et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012; 2 p.
21. Zaharia M et al. Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. Presented as part of the, 2012.
22. Bollerslev T, Chou RY, Kroner KF. ARCH modeling in finance: A review of the theory and empirical evidence. Journal of econometrics, 1992; 52(1-2): 5-59.
23. Franses PH, Van Dijk D. Non-linear time series models in empirical finance. Cambridge University Press, 2000.
24. Karpenko S et al. Visual navigation of the UAVs on the basis of 3D natural landmarks. Eighth International Conference on Machine Vision. International Society for Optics and Photonics, 2015: 98751I-98751I-10.
25. Shvets EA, Nikolaev DP. Complex approach to long-term multi-agent mapping in low dynamic environments. Eighth International Conference on Machine Vision. International Society for Optics and Photonics, 2015: 98752A-98752A-10.
26. Miller A, Miller B. Stochastic control of light UAV at landing with the aid of bearing-only observations. Eighth International Conference on Machine Vision. International Society for Optics and Photonics, 2015: 987529-987529-10.
27. Protsenko VI, Kazanskiy NL, Serafimovich PG. Real-time analysis of parameters of multiple object detection systems. Computer Optics, 2015; 39(4): 582-591. DOI: 10.18287/0134-2452-2015-39-4-582-591.
28. Kazanskiy NL, Protsenko VI, Serafimovich PG. Comparison of system performance for streaming data analysis in image processing tasks by sliding window. Computer Optics, 2014; 38(4): 804-810.