Aufgabenstellung Masterarbeit

Name, Vorname:	Sun, Jun
Studiengang:	Distributed Systems Engineering
Matrikelnummer:	3933532
Thema:	***Sliding Window Joins over Out-of-order Data Streams***

Description of Work

Driven by the increasing need for processing large volumes of data that is continuously generated by financial markets, sensors embedded in environments, network infrastructures, etc., *data stream processing* has now become an important data management technology, and many stream processing engines (SPE) have been developed. In many scenarios where data streams are produced, data elements in a stream are associated with timestamps at their generation, which determine a temporal order on the data stream. However, in many applications, especially those involving distributed and network data, data may not arrive at SPEs in the strict time-order due to network delays and asynchronous communications [CKT08, KFD+10]. Such streams where data is not ordered according to timestamps are referred to as *out-of-order* streams. The correctness of results produced by many data analyzing operations relies on the order in which the incoming data is processed, therefore, SPEs must be able to deal with out-of-order data streams.

A commonly used approach for dealing with out-of-order arrivals is the so-called K*-Slack* approach [BSW04]. Essentially, it is a buffering-based approach, where a buffer of size K time units or K stream elements is used to temporarily keep and sort incoming data in timestamp-order before sending them to actual data-analysis operations. Since the first proposal of K-Slack in [BSW04], many improvements have been proposed (e.g. [MP13a, MP13b]), which try to minimize the K-Slack buffer size while still keeping a good quality of out-of-order handling. Motivated by the observation that many stream processing applications do not require 100% correct results, an on-going research in our lab is a new variant of the K-Slack approach, which adjusts the buffer size at runtime to minimize the latency introduced by the K-Slack buffer while guaranteeing that errors in the produced results do not exceed a user-specified bound. Different types of stream processing operations might use different error models for their results. For instance, for sliding window aggregations, we can use the relative error of the produced aggregate results, in relation to the actual aggregates results if all data had arrived in strict time-order. We refer to this new variant of K-Slack as *Adaptive Quality-Driven* K*-Slack*.

We have applied the adaptive quality-driven K-Slack approach for processing sliding window aggregations over out-of-order data streams. The core of this thesis is to extend our existing work to support processing sliding-window joins over out-of-order data streams as well. Sliding-window join is a fundamental operation in data stream processing [GO03]. Joining out-of-order data streams poses new challenges in terms of the trade-off between the latency of join results and the correctness of the ordering of results [Ham05]. Within this thesis, a proper model for measuring the quality of the sliding-window join results should be developed. The developed quality model should be integrated into the adaptive quality-driven K-Slack approach and applied to process sliding window joins. The solution should be implemented in a prototype SPE, which is an extension of a state-of-the-art commercial SPE. In addition, the SPE prototype currently supports only system-time based join semantics. Therefore, an extension to support application-time based join semantics is required.

Backlog

Based on the above description of the thesis scope, a following list of features should be developed:

1. Extend the existing SPE prototype which currently supports only system-time based processing semantics for sliding window joins to support application-time based processing semantics as well.
2. Develop a model for measuring the quality of the sliding-window join results and integrate the developed model with the existing adaptive quality-driven K-slack approach.
3. Integrate the overall solution for sliding window joins over out-of-order data streams into the SPE prototype. This involves making design decisions like where to hook of the disorder handling component to best fit the current architecture and implementation of the prototype.
4. The solution should be evaluated using both synthetic and real-world out-of-order data streams.

References

[BSW04] Shivnath Babu, Utkarsh Srivastava, and Jennifer Widom. Exploiting k-constraints to reduce memory overhead in continuous queries over data streams. *ACM Trans. Database Syst.*, 29(3):545–580, September 2004.

[CKT08] Graham Cormode, Flip Korn, and Srikanta Tirthapura. Time-decaying aggregates in out-of-order streams. In *Proceedings of the Twenty-seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '08, pages 89–98, New York, NY, USA, 2008. ACM.

[GO03] Lukasz Golab and M. Tamer Özsu. Issues in data stream management. *SIGMOD Rec.*, 32(2):5–14, June 2003.

[Ham05] Moustafa A. Hammad. Optimizing in-order execution of continuous queries over streamed sensor data. In *In Proc. Int. Conf. on Scientific and Statistical Database Management (SSDBM)*, pages 143–146, 2005.

[KFD+10] Sailesh Krishnamurthy, Michael J. Franklin, Jeffrey Davis, Daniel Farina, Pasha Golovko, Alan Li, and Neil Thombre. Continuous analytics over discontinuous streams. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 1081–1092, New York, NY, USA, 2010. ACM.

[MP13a] C. Mutschler and M. Philippsen. Distributed low-latency out-of-order event processing for high data rate sensor streams. In *Parallel Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, pages 1133–1144, 2013.

[MP13b] Christopher Mutschler and Michael Philippsen. Reliable speculative processing of out-of-order event streams in generic publish/subscribe middlewares. In *Proceedings of the 7th ACM International Conference on Distributed Event-based Systems*, DEBS '13, pages 147–158, New York, NY, USA, 2013. ACM.

Betreuer:	M.Sc. Yuanzhen Ji
Verantwortlicher Hochschullehrer:	Prof. Christof Fetzer
Institut:	Systemarchitektur
Lehrstuhl:	Systems Engineering
Beginn am:	1.10.2014
Einzureichen am:	15.03.2015

Verantwortlicher Hochschullehrer