State management for elastic distributed streaming applications

Master Thesis Proposal

Introduction Efficient state management is crucial for long-running distributed streaming applications. On one hand, streaming applications have real-time latency requirements, thus streaming engines need to process incoming events and update their internal state as quickly as possible. On the other hand, streaming applications operate in highly dynamic distributed environments where workload changes and worker failures are common. Therefore, stream processing systems need to support dynamic scaling and quick failure recovery, while ensuring the correctness and balanced re-distribution of state.

To support state management, many stream processors offer high-level state APIs and predefined state types to users [3, 4]. Application state defined with the provided APIs and types is then guaranteed to be guarded against failures and automatically split and/or merged during reconfiguration. However, efficiently managing arbitrary user-defined state is challenging. The provided state API has to be expressive and flexible to support a wide variety of streaming applications, such as Complex Event Processing (CEP) and Graph analytics, while at the same time ensuring that state can be efficiently represented to support low-latency scaling and recovery.

Thesis Goal This thesis will extend Strymon [1] with state management capabilities. Strymon applications are highly dynamic so that they often require reconfiguration while running. When adding or removing resources to/from a running dataflow, state needs to be efficiently redistributed among a dynamic set of workers. At the same time, state management must be designed in a way that enables the implementation of efficient analytical querying and fine-grained recovery in the case of failures.

The goal of this thesis is to define and develop a high-level state API for Strymon and a set of state types for Timely Dataflow [2] operators. The state management mechanism will be designed so that dynamic scaling and reconfiguration of running dataflows can be efficiently supported in the future. The state API needs to allow the definition of complex data types, such as arrays, maps, and arbitrarily nested structures and be easy to integrate with Strymon's in-progress scaling and fault-tolerance mechanisms.

Evaluation The student will first evaluate the performance of existing techniques for streaming operator state representation. In particular, we are interested in understanding the performance of state types in terms of (i) updates, (ii) queries, (iii) repartitioning and merging, and (iv) checkpointing. As a next step, the student will use Strymon's state API to develop CEP and graph analytics streaming applications and evaluate its flexibility to support complex use-cases. Finally, the student will use the developed applications to evaluate the developed state management mechanism in dynamic scaling and reconfiguration scenarios.

Additional Directions If time allows, the student will study approaches based on the PAX-based storage layout [1] and differential updates [5], two techniques that have previously demonstrated superior performance in an industrial use-case [2]. We will explore how to extend these techniques further to support complex data types, such as arrays, maps, and arbitrarily nested structures. Another possible direction is to evaluate the performance of commonly used state backends for, such as RocksDB [3] and HDFS.

[1] Anastassia Ailamaki et al. "Weaving Relations for Cache Performance". In: *Proceedings of the 27th International Conference on Very Large Data Bases*. VLDB '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 169–180.

[2] Lucas Braun et al. "Analytics in Motion: High Performance Event-Processing AND Real-Time Analytics in the Same Database". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD '15. Melbourne, Victoria, Australia: ACM, 2015, pp. 251–264.

[3] Paris Carbone et al. "State Management in Apache Flink&Reg;: Consistent Stateful Distributed Stream Processing". In: *Proc. VLDB Endow.* 10.12 (Aug. 2017), pp. 1718–1729.

[4] Raul Castro Fernandez et al. "Integrating Scale out and Fault Tolerance in Stream Processing Using Operator State Management". In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. SIGMOD '13. New York, New York, USA: ACM, 2013, pp. 725–736.

[5] Jens Krueger et al. "Fast Updates on Read-optimized Databases Using Multi-core CPUs". In: *Proc. VLDB Endow.* 5.1 (Sept. 2011), pp. 61–72.

If you are interested in this project please contact Vasiliki Kalavri (vasiliki.kalavri@inf.ethz.ch). The proposed thesis will be supervised by Prof. Timothy Roscoe.

[1] http://strymon.systems.ethz.ch/
[2] https://github.com/frankmcsherry/timely-dataflow
[3] http://rocksdb.org/