LOCK-FREE DATA STRUCTURES FOR DATA STREAM PROCESSING

Alexander Baumstark

MOTIVATION [1]

- Key requirements of Data Stream Processing are high throughput and low latency
  - e.g., sensor data processing, IoT, data analytics
- Require efficient parallelization of algorithms
- Conventional techniques use blocking mechanisms which limit the possible parallelization
- Primary objectives:
  1. Lock-free design principles
  2. Data Structure Implementations
  3. Effect on performance in Data Stream Processing

LOCK-FREE SYNCHRONIZATION

- Non-blocking synchronization
- Achieved with atomic operations like compare and swap (CAS) or fetch and add (FAA)
- Guarantees that at least one thread is doing progress
- Results in practice with a higher degree of parallelism

BENCHMARKS

- Lock-free synchronization can improve the throughput of algorithms in Data Stream Processing
- The Pipefabric framework is used for benchmarks in order to compare the blocking and lock-free approaches

1. Tuple exchanging process with Reader-Write Queues
2. Symmetric Hash Join with (Multi-)Hashmaps

PIPEFABRIC [2]

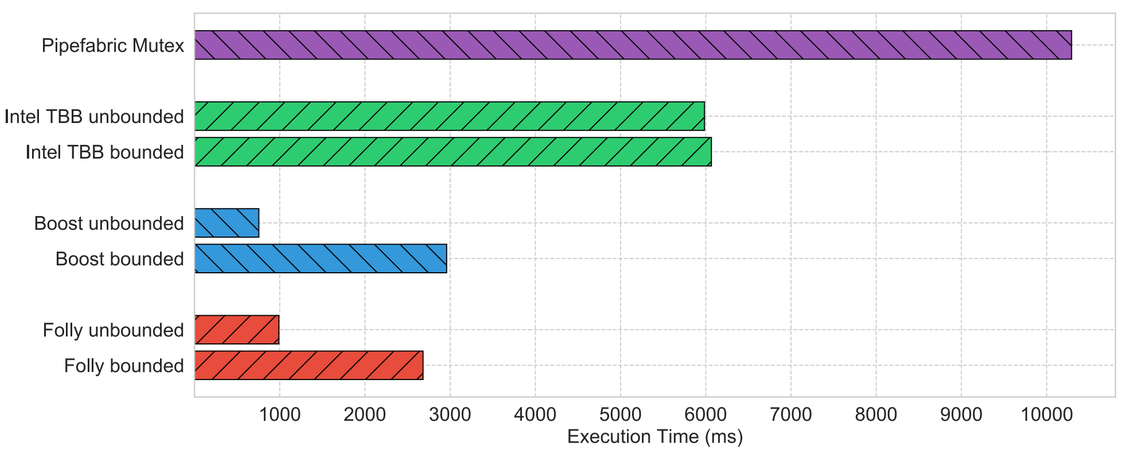
- C++ framework for processing streams of tuples
- Provides a set of operators and utility classes:
  - publish-subscribe framework
  - operators for data stream processing (aggregates, grouping, joins and complex event processing)

CONTRIBUTION

- Lock-free (Multi-)Hashmap design
- Improved performance for tuple exchange process and Symmetric Hash Join algorithm
- Benchmarks (Pipefabric)

TUPLE EXCHANGE

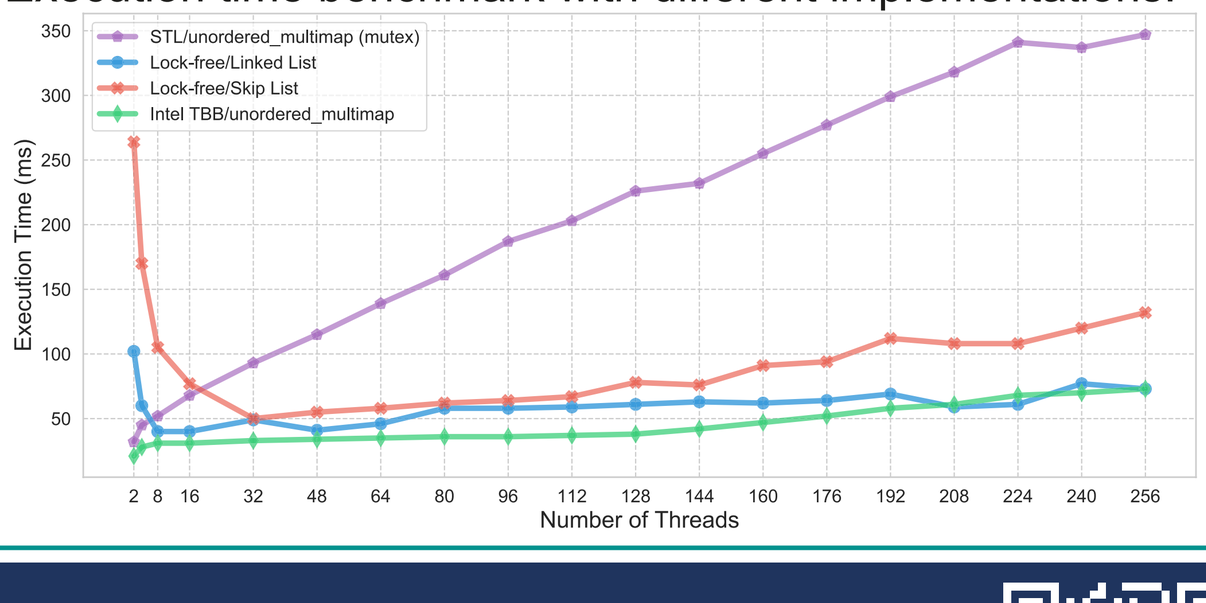
- Tuple exchange process between two threads
  - based on single reader and single writer queue
  - current implementation locks the entire queue for each push/pop
- Benchmark of different implementations with performance libraries:



SYMMETRIC HASH JOIN

- Continuously generates results while tuples arrive from a stream
- Used data structure: STL (Multi-)Hashmap
  - supports multiple elements with the same key
  - current implementation locks entire hashmap
- Lock-free implementations enable a higher scalability

Execution time benchmark with different implementations:



Symmetric Hash Join Algorithm

[1] Alexander Baumstark. Lock-free Data Structures for Data Stream Processing. Bachelor's Thesis, TU Ilmenau, Aug. 17, 2018.
[2] https://github.com/dbis-ilm/pipefabric

Technische Universität Ilmenau

Alexander Baumstark
contact: alexander.baumstark@tu-ilmenau.de

TECHNISCHE UNIVERSITÄT ILMENAU