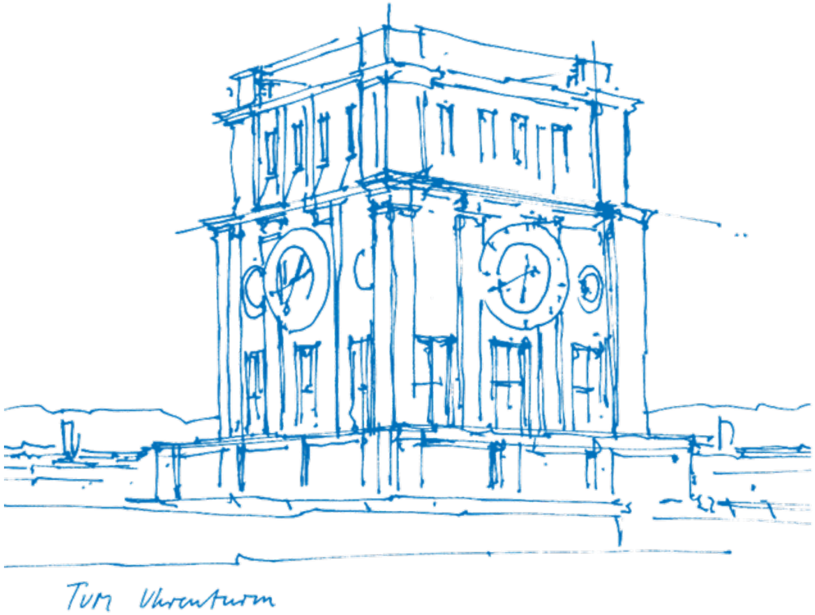
Implementing an Efficient Shuffle Operator for Streaming Database Systems

Bachelor Thesis

Author: Jonas Ladner

Supervisor: Maximilian Rieger, M.Sc.

Examiner: Prof. Dr. Thomas Neumann



Garching, 11.03.2025
Technical University of Munich

Problem Setting

Key Contribution: Efficient, multithreaded shuffle operator implementations.

Shuffle-Simulation Process:

1. **Tuple Generation:** Randomly generated tuples with 32-bit keys and optional data fields.
2. **Data Shuffle:** Tuples stored in partition buckets using slotted pages.
3. **Storing on Slotted Pages:** Thread-local vs. shared (locking/lock-free) write-out strategies.

Implementations

Evaluation

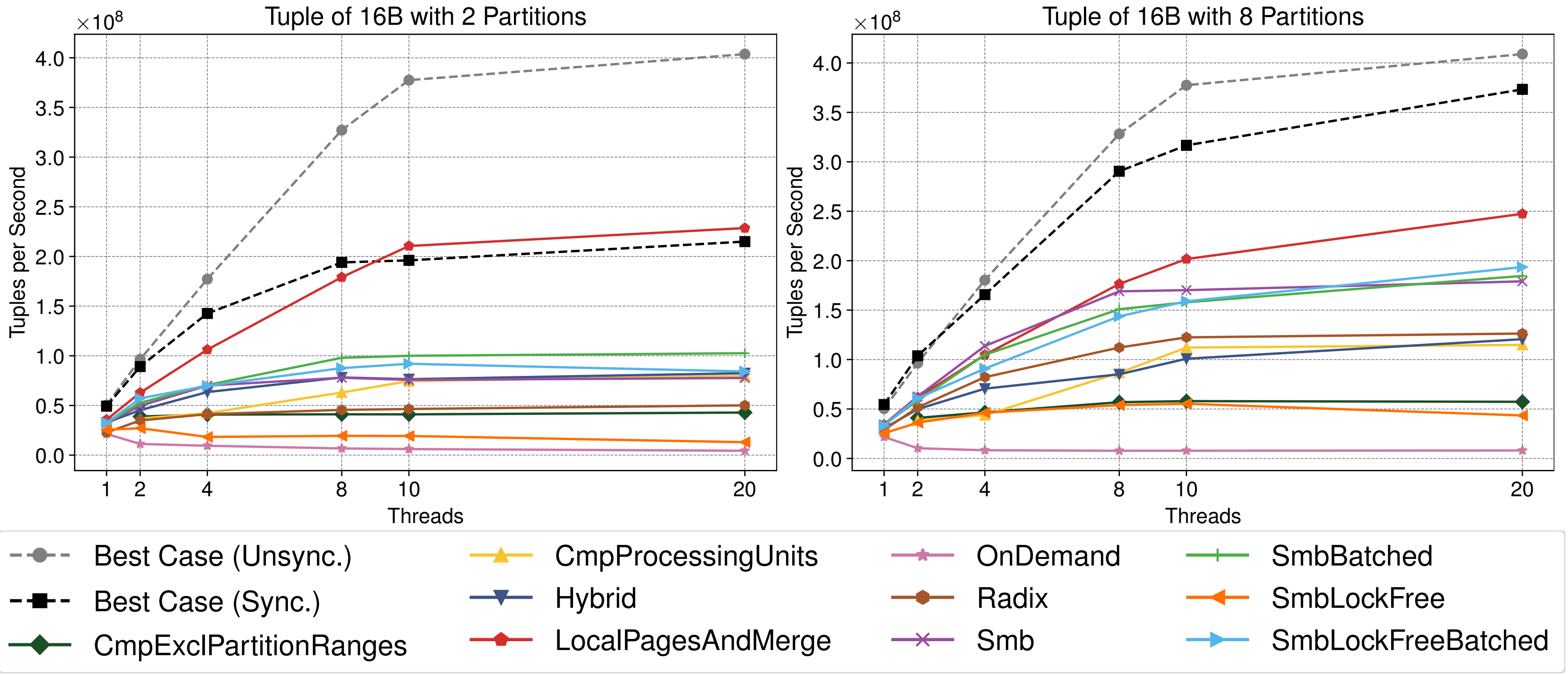


Figure: Benchmark Plots for Tuple of 16B with 2 and 8 Partitions

Evaluation

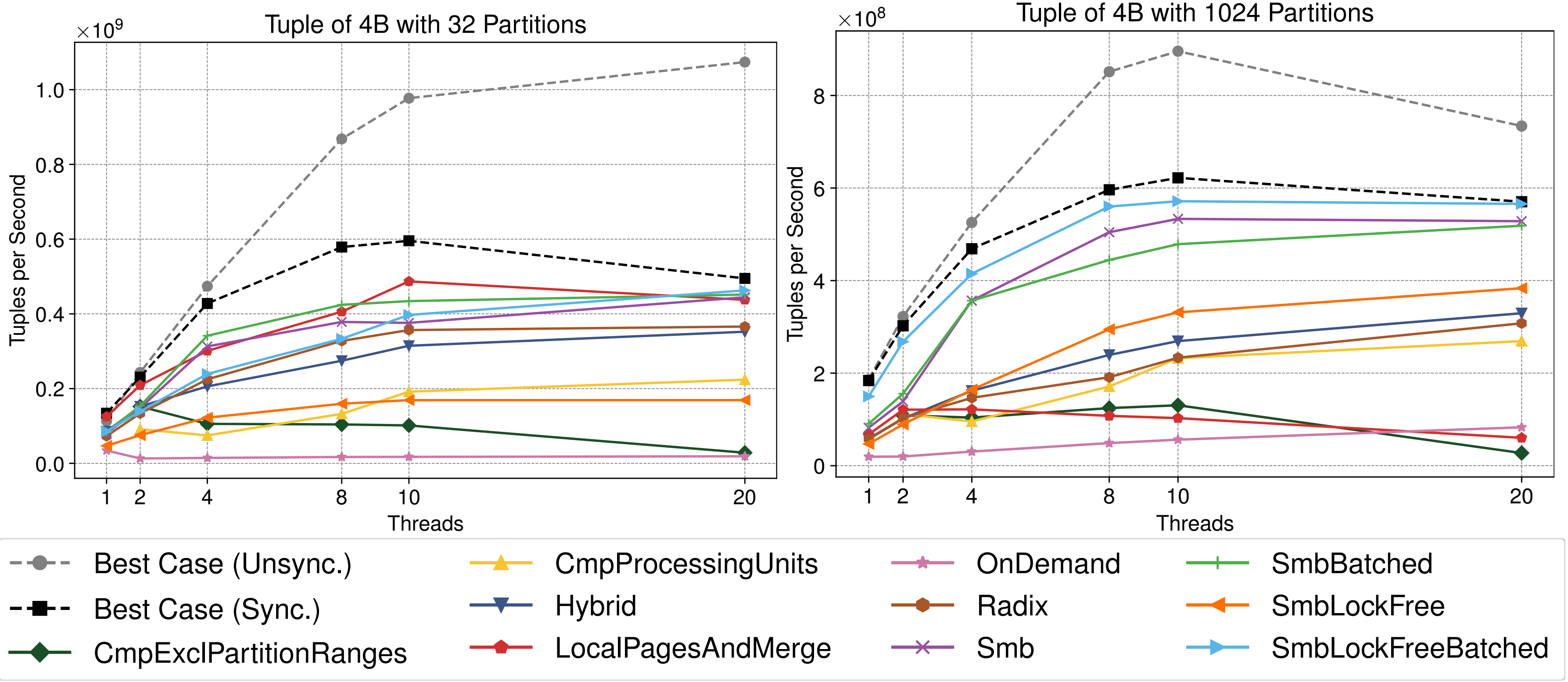


Figure: Benchmark Plots for Tuple of 4B with 32 and 1024 Partitions

Evaluation

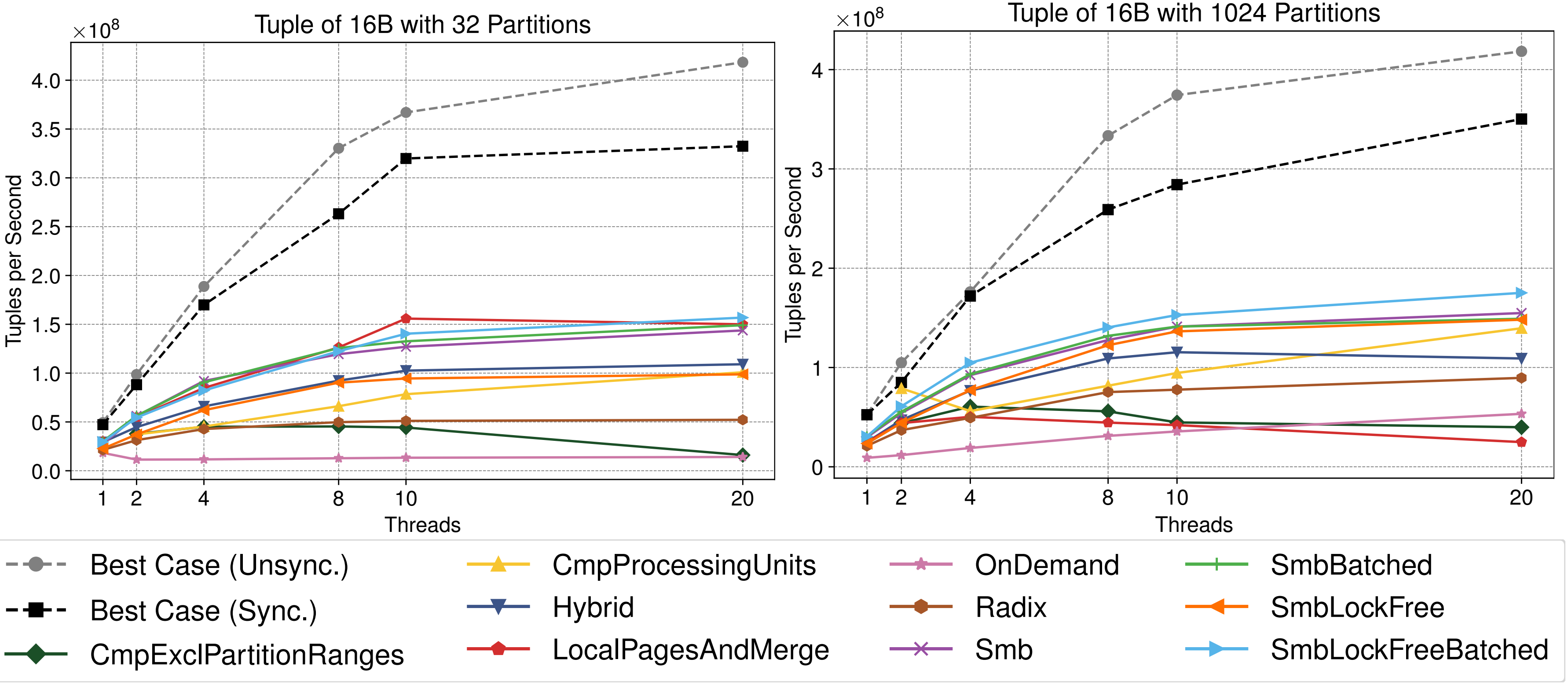


Figure: Benchmark Plots for Tuple of 16B with 32 and 1024 Partitions

Evaluation

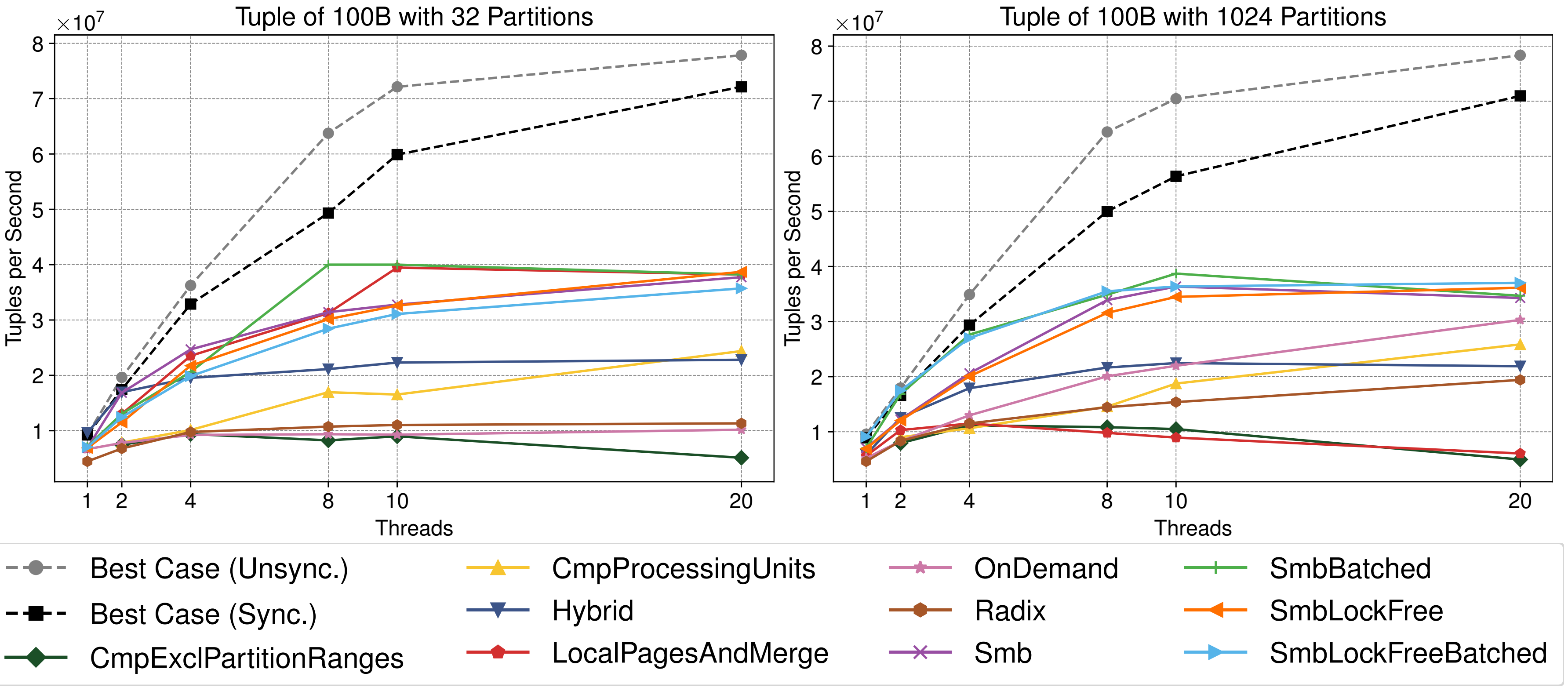


Figure: Benchmark Plots for Tuple of 100B with 32 and 1024 Partitions

Peak Heap Memory

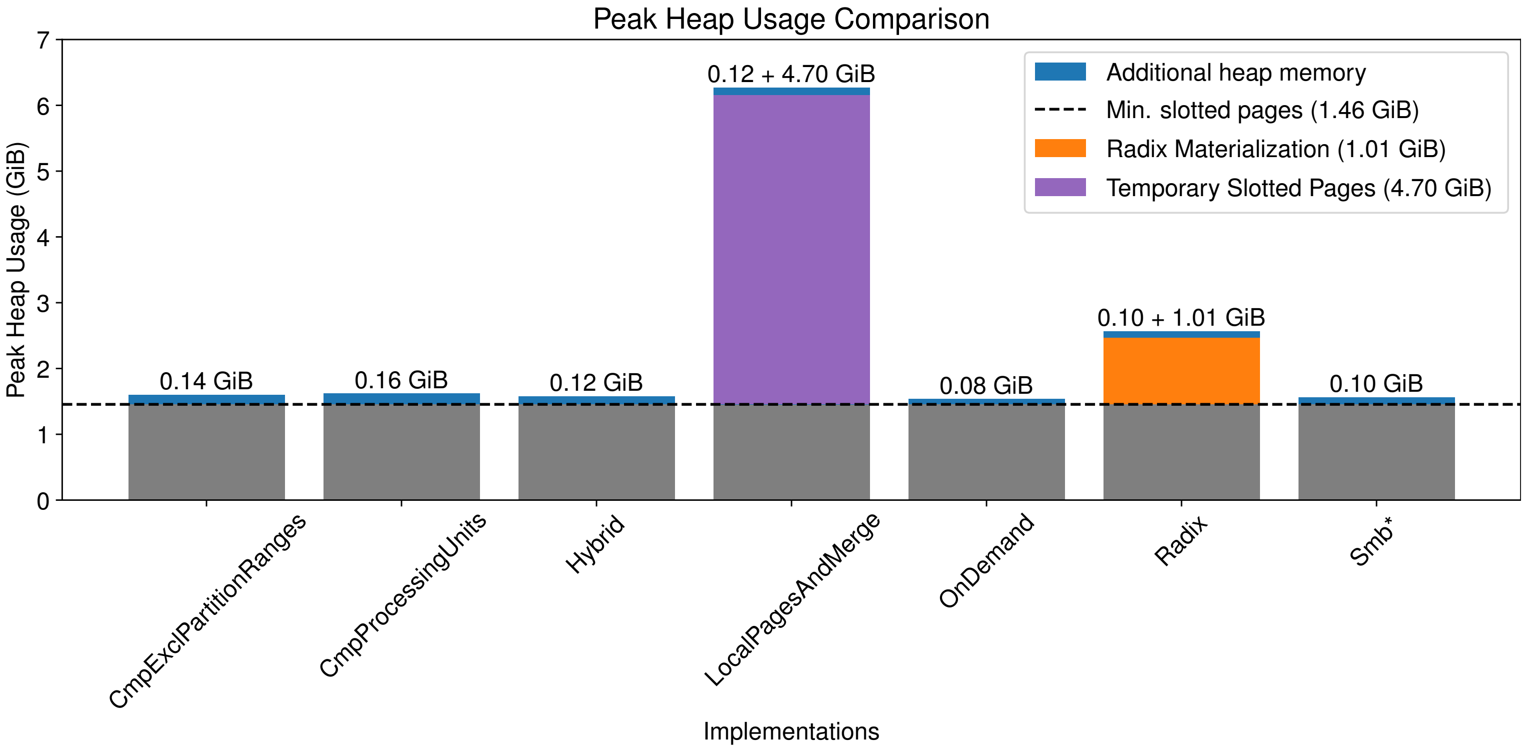


Figure: Peak Heap Usage when using 32 Partitions, 40 Threads and 67.2 Mio. 16B Tuples (1 GiB)

Comparison with Apache Flink

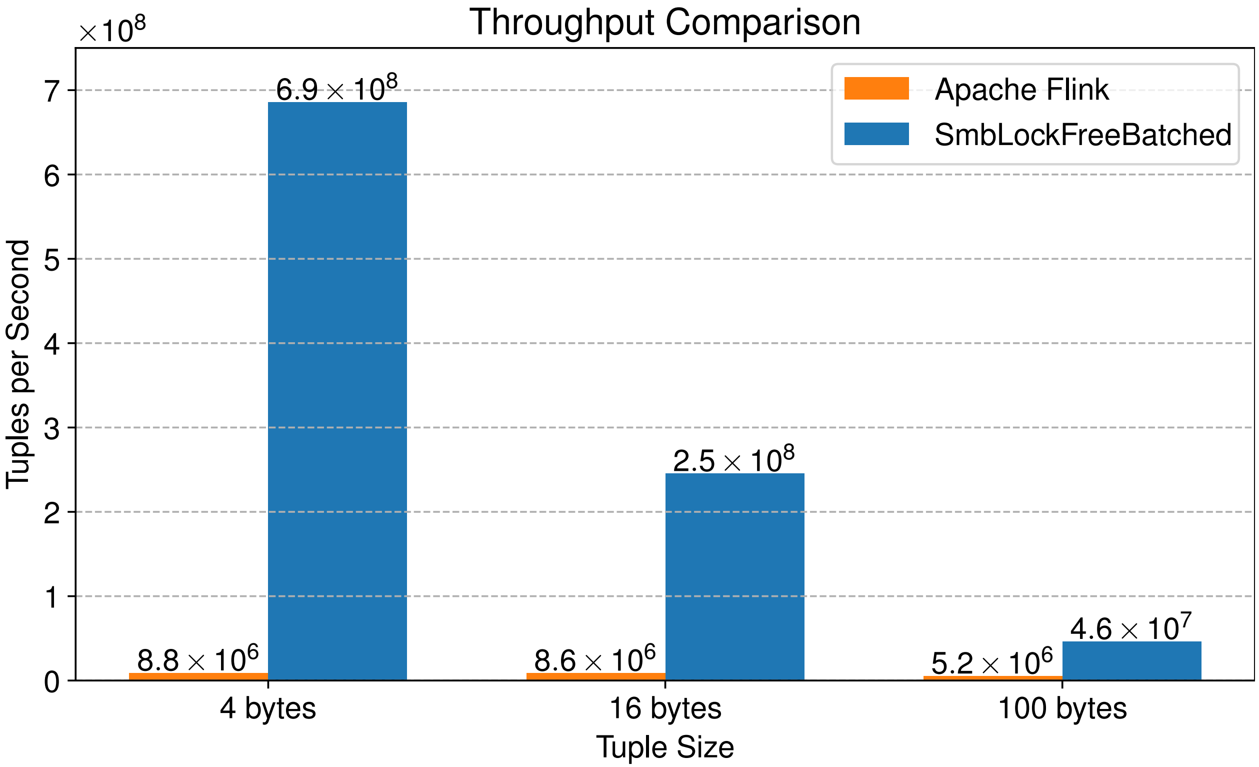


Figure: Tuples per Second Comparision when using 1024 Partitions

Future Work