Implementation of a Benchmark Suite for Strymon

Nicolas Hafner

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Systems@ETH Zürich

Motivation

- Performance is important for systems
- Measuring performance can be tough
- Would like to compare to competitors
- Industry benchmarks need to be evaluated

Other Works

- Investigated 9 papers for other systems
- Often very simple benchmarks like Word Count
- Code and data often not published
- No paper used a standardised benchmark

Timely

- Streaming processing based on data parallelism
- Computations expressed via data flow graphs
- Each physical worker runs the entire data flow
- Physical exchange of data handled by Timely
- Data is separated into chunks called epochs
- Computation progress tracked by epoch frontiers

Benchmarks

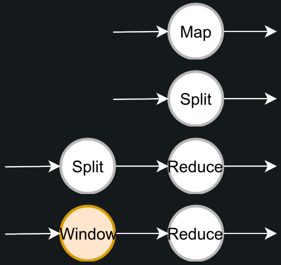
- We implemented three benchmarks:

1. Intel’s HiBench
2. Yahoo’s Streaming Benchmark
3. Apache Beam’s NEXMark

HiBench

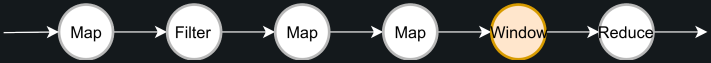
- General Big Data benchmark
- Includes a section for streaming systems
- Only four latency tests:

1. Identity
2. Repartition
3. Word Count
4. Window Reduce



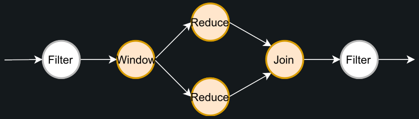
YSB

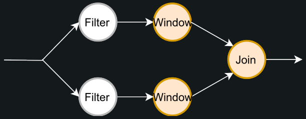
- Initially developed internally at Yahoo:
- Count ad views for ad campaigns
- Original setup requires Redis table lookup
- Only one, relatively simple data flow:



NEXMark

- Based on a paper by Tucker et al.
- An adaptation of XMark for streaming
- Implements an "auctioning system"
- 13 data flows in total
- Uses filter, map, reduce, join, window, session, partition
- Dataflows for Query 5 and 8:

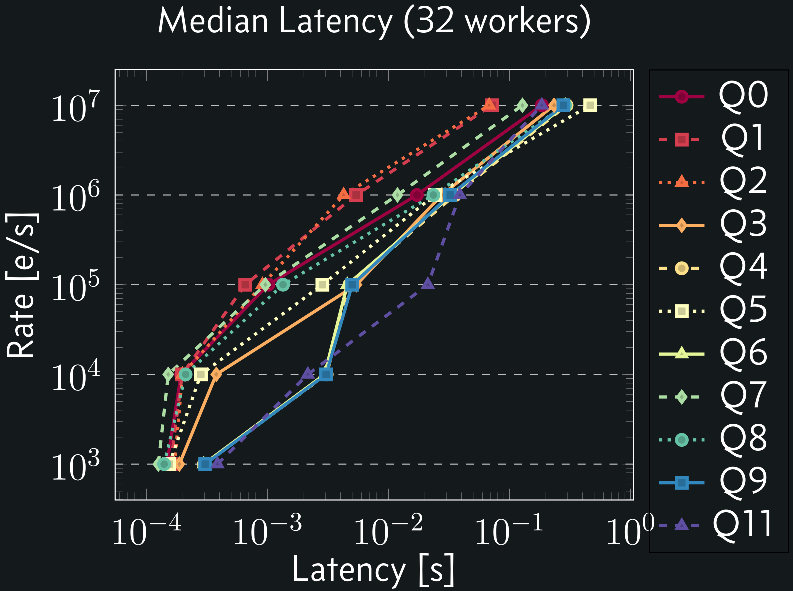




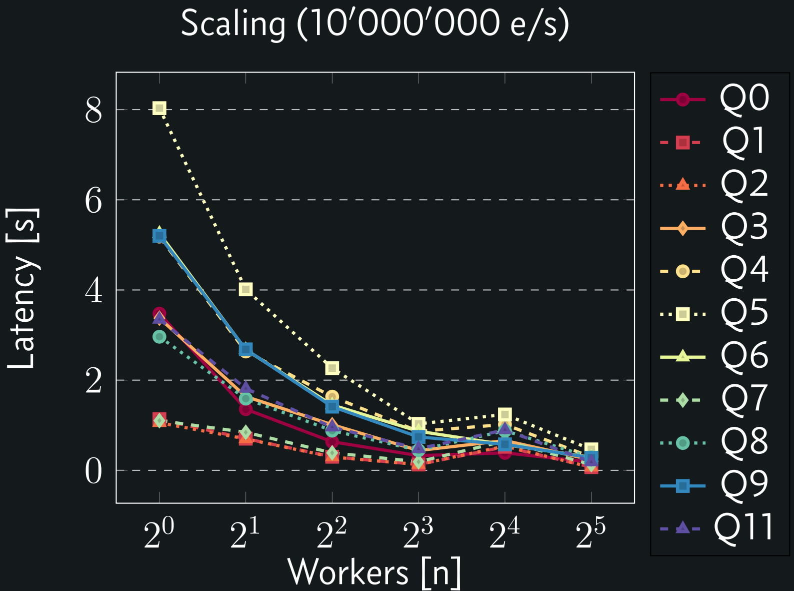
Evaluation

- Run on sgs-r815-03 (32 cores, 2.4GHz)
- Measured closed-loop per-epoch latency
- Data generated directly in memory
- Timely performs very well!

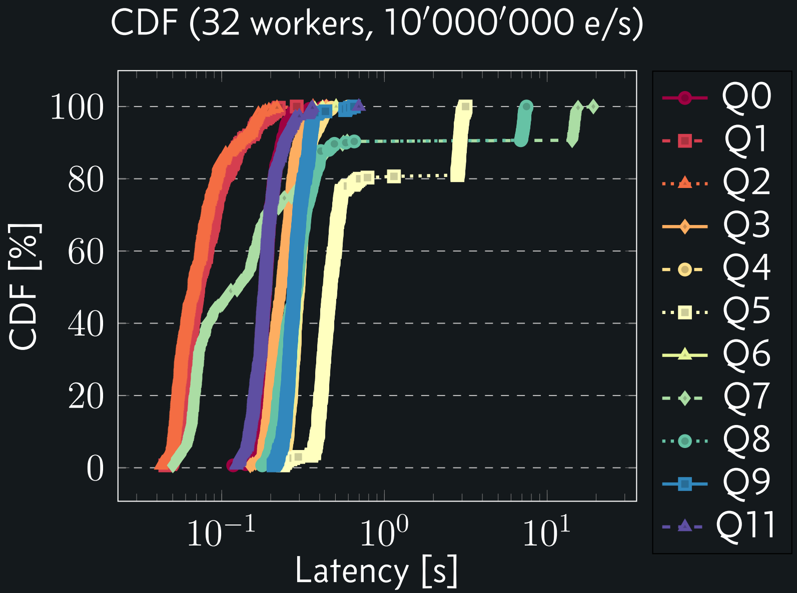
NEXMark Latency Scaling



NEXMark Worker Scaling



NEXMark CDF



CDF (32 workers, 10′000′000 e/s)
CDF [%]
100
80
60
40
20
0
10^{-1}
10^{0}
10^{1}
Latency [s]
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q11

Conclusion

- Timely competes very well with other systems
- Existing benchmarks leave things to be desired
- New benchmarks need:
  - Abstract model definitions for data flows
  - Various short and long data flows
  - Deterministically generated workloads
  - Correctness verification tools

Thanks

Thanks to John Liagouris, Frank McSherry, and the whole DCModel team!