Universität Potsdam

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

# Distributed Dynamic Stream Processing
# Master Project WS 2020

Tilmann Rabl

Data Engineering Systems

# This Session

1. Intro Data Engineering Systems

2. Motivation Master Project

3. Milestones

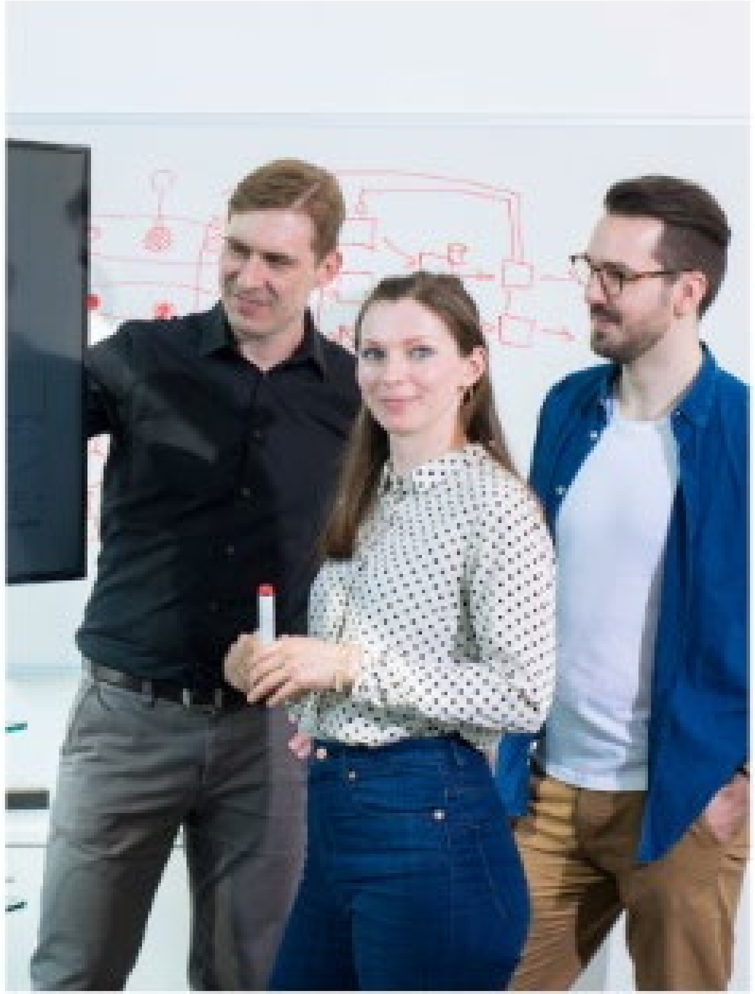
# Data Engineering Systems

Systems that enable data engineering

- Database systems
- Stream processing systems
- Graph processing systems
- Machine learning systems
- …

Systems that support data science

- Experiment databases
- Optimizers
- Deployment systems
- End-to-end ML



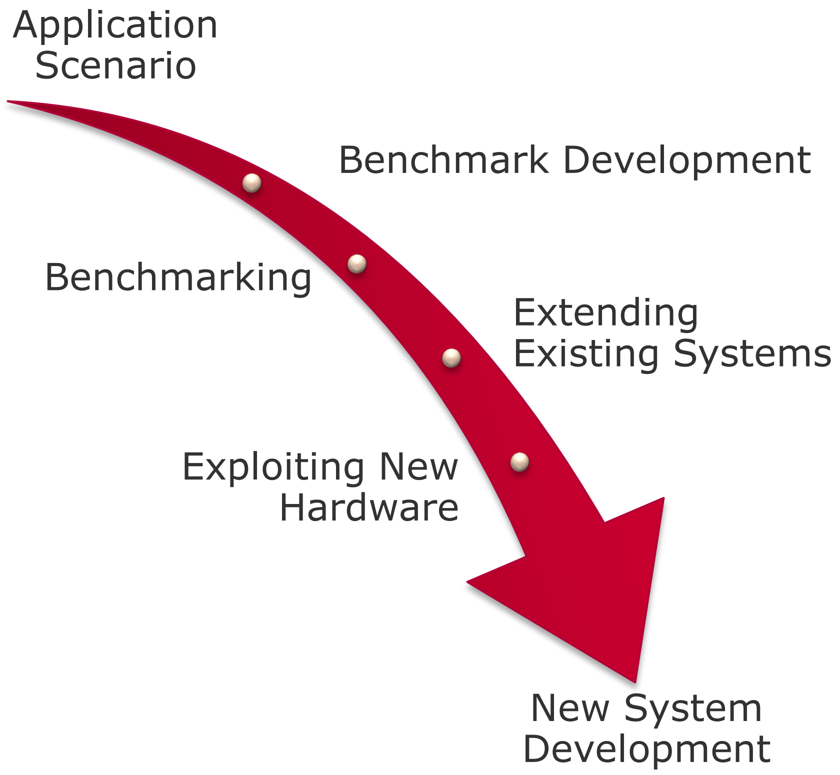
# Data Engineering Systems Group

# Research Topics

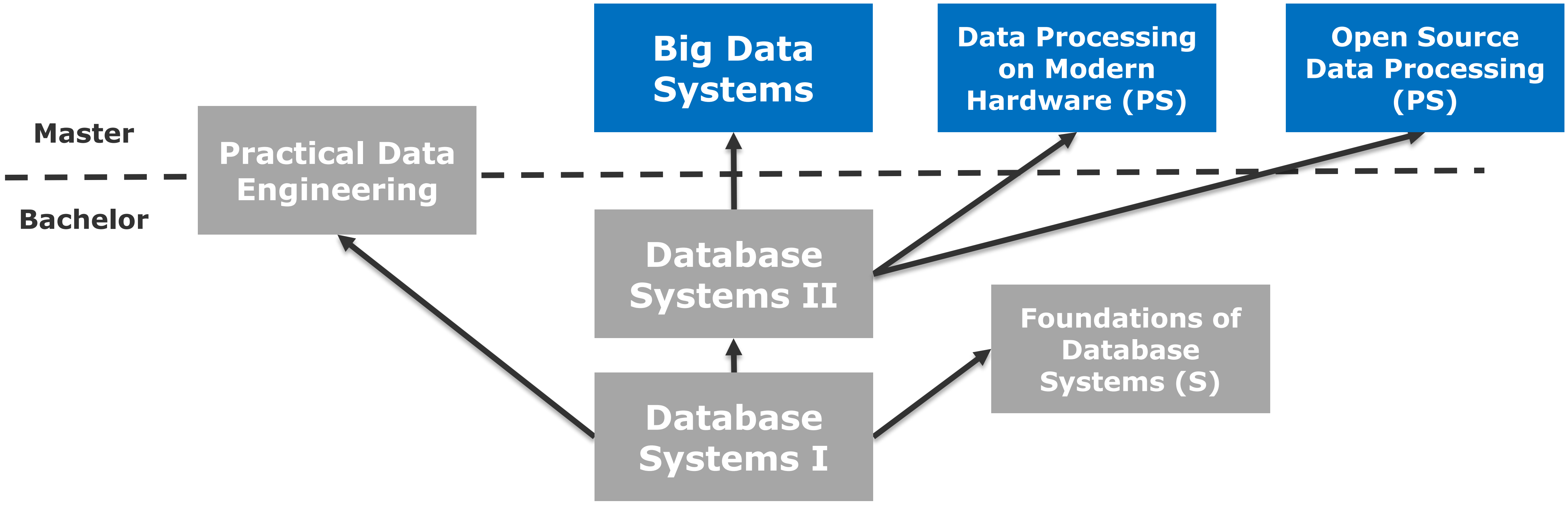
- Database Management
  - SIGMOD 17, VLDBJ 18, SIGMOD 20
- Machine Learning Systems
  - PVLDB 17,  SOCC 18, EDBT 20, SIGMOD 20
- Stream Processing
  - SIGMOD 19, PVLDB 19, SIGMOD 20, PVLDB 20
- Benchmarking
  - SOCC 17, ICDE 18, ICPE 18, PVDLB 20

Interested in a thesis? Write us an e-mail.

## Research Approach



# Data Engineering Courses


Big Data Systems
Data Processing on Modern Hardware (PS)
Open Source Data Processing (PS)
Master
Practical Data Engineering
Bachelor
Database Systems II
Foundations of Database Systems (S)
Database Systems I

# Teaching

This semester only Master level courses.

- Big Data Systems
  - Lecture on Implementation and Use of Big Data Systems
  - Tuesdays/Thursdays, 11:00 – 12:30
  - Master, 6 ECTS
- Open Source Data Processing
  - Learn how to do open source development with practitioners
  - Wednesdays, 11:00 – 12:30
  - Master, 6 ECTS
- B & M Projects
  - Bachelor Project – ML System Benchmarking
  - Master Project – Distributed Stream Processing
- Data Management on Modern Storage Technologies
  - Project Seminar on DB + PMem
  - Wednesdays, 13:30 – 15:00
  - Master, 6 ECTS
- GPU Accelerated Data Processing
  - Project Seminar on GPU + DP
  - Tuesdays, 13:30 – 15:00
  - Master, 6 ECTS
- Projects & Theses
  - Topics on Data Engineering Systems available
  - Guidance in Competitions and Challenges
  - SIGMOD SRC / Programming C / DEBS GC

# Motivation

# Big Fast Data

Data is growing

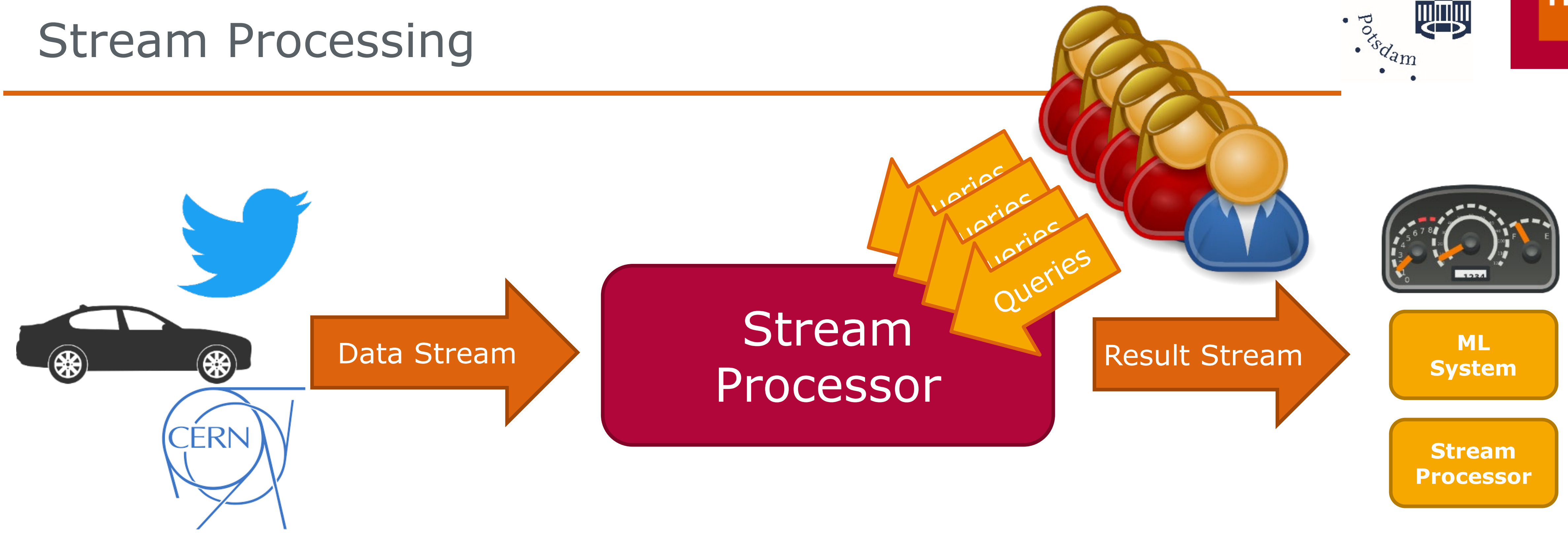
Messages, tweets, social networks (statuses, check-ins, shared content), blogs, click streams, various logs, …

- *Facebook: > 1,5B active users, > 60B messages/day*
- *Twitter: > 300M active users, > 500M tweets/day*

Everyone is interested!

**The value of data is decreasing with its age!**

# Stream Processing



Data Stream

Stream Processor

Queries

Result Stream

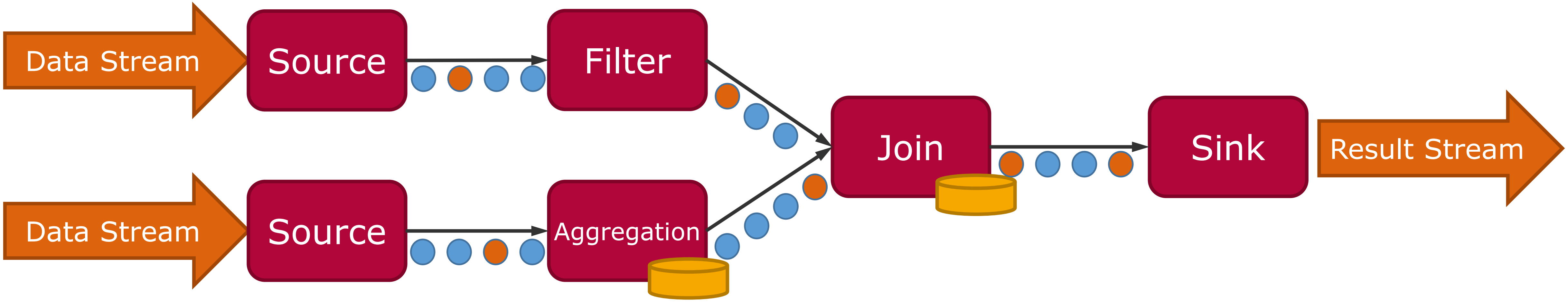
ML System

Stream Processor

Challenge

- Potentially unlimited data set
- Many different queries
- Continuous results

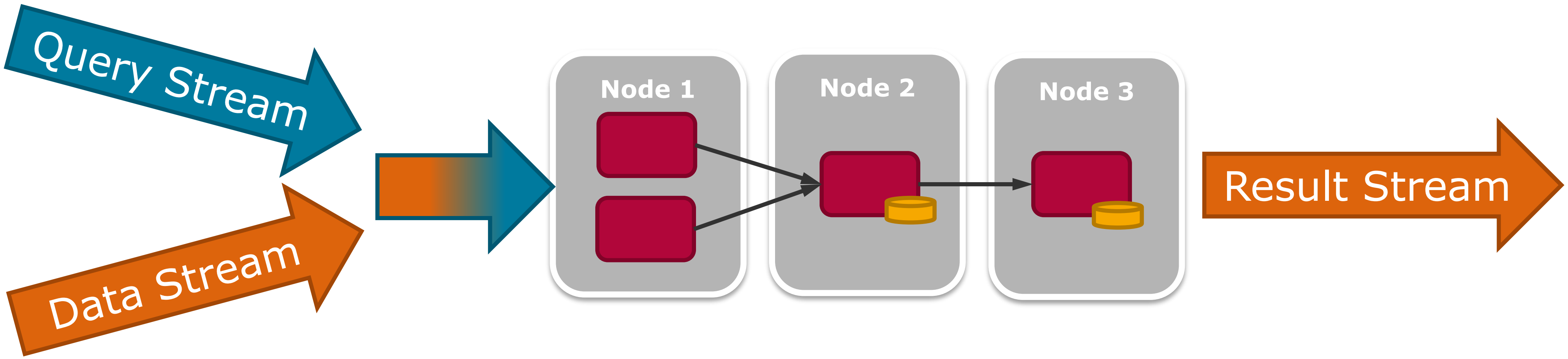
# Streaming Processing Job



## Dataflow

- 🟥 Operators
- 🔵 Records
- 🟠 Control events
- 🛢 State

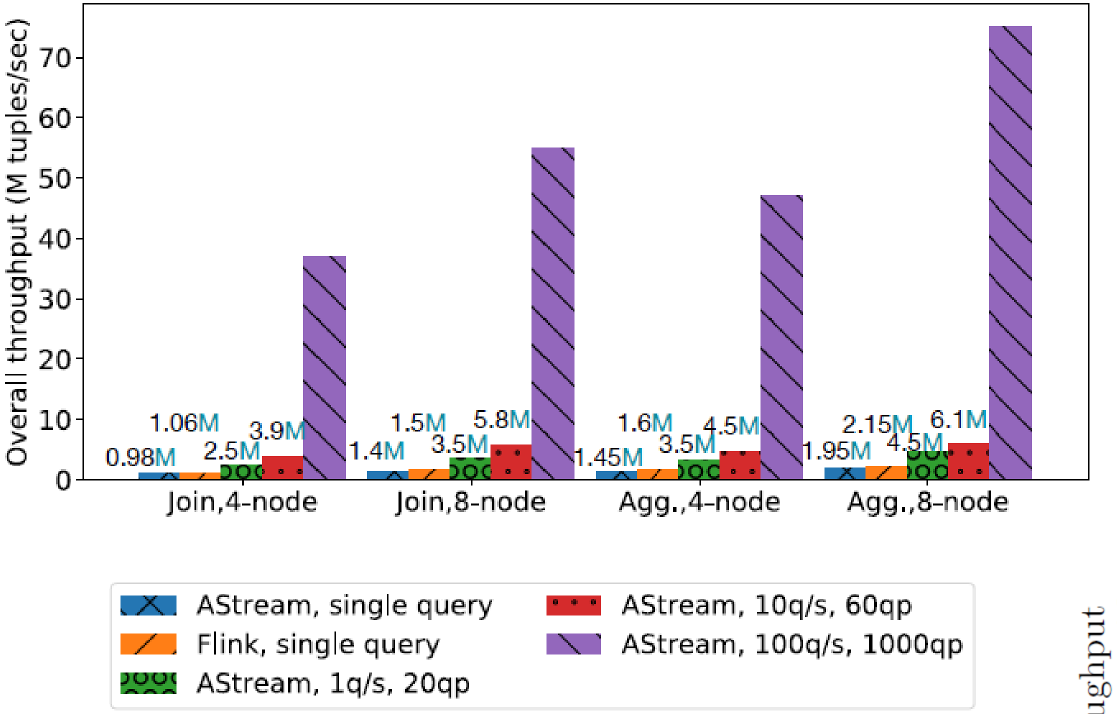
# Distributed Stream Processing



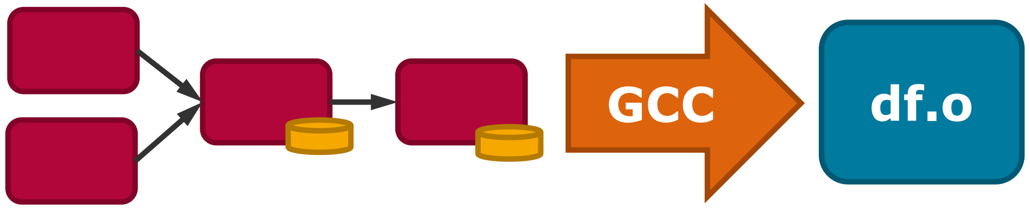
- Distribute query topology on multiple nodes
- Employ efficient network communication

Hendrik Makait: Rethinking Message Brokers on RDMA and NVM. SIGMOD 2020.

# Make it fast!



- Adhoc query processing is much more efficient!
- But it could be even faster – through code generation!
- Modern networking can help



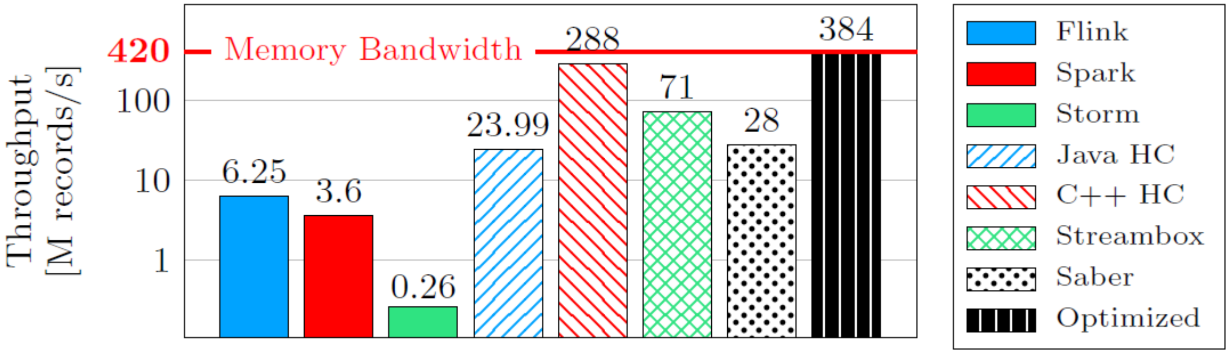


**Figure 1: Yahoo! Streaming Benchmark (1 Node).**

# Thank you!

Distributed Dynamic Stream Processing

Build

- Distributed stream processing engine prototype
- Query compilation pipeline
- ~~Multiquery optimization~~

Learn

- Distributed systems engineering
- Stream processing
- Query compilation

Supervising Team

- Pedro Silva
- Tilmann Rabl

# Meeting Time

- Proposal:
  - Wednesday 3:15 pm

# Code of Conduct

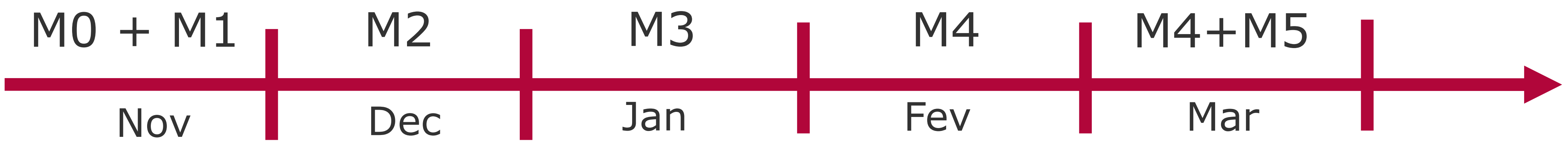
- This course should be fun for everyone
  - Needs to be a safe environment

- Communication
  - Learn how to write professional emails: https://medium.com/@lportwoodstacer/how-to-email-your-professor-without-being-annoying-af-cf64ae0e4087
  - Use netiquette in forum, email, chats, etc.

- Generally
  - Treat everyone with respect and consideration
  - HPI should be a safe place for everyone

# Project Overview and Milestones

# Overview



# Milestone 0: Work Environment (1 week)

- Main interests / expectations of each member
- Start reading the recommended literature
- Outline of group dynamics: e.g. hours per week, workdays of the week, etc.
- Weekly meeting time slot
- Software engineering approach: e.g. XP, Scrum, Waterfall, etc.
- Communication tools of the group: e.g. Slack, Discord, IRC, etc.
- IDEs: C Lion, Visual Studio, Eclipse, Emacs, etc.
- Repository solution: Github, Gitlab, Bitbucket, etc.
- File sharing: Google drive, Owncloud, Slack, etc.
- Quick presentation on the weekly meeting

# Milestone 1: Setting up the baseline (1-2 weeks)

- Defining an **use case/workload**: what the data will look like?
- Defining a **simple workflow on Flink** using only filter, join and window (aggregation) operators;
- Getting the code from Jeyhun's Flink benchmarking tool: https://github.com/streamline-eu/StreamBenchmarks
- Defining which hat **metrics** could be interesting to understand the performance of the execution.
- Running Jeyhun's benchmark on the application and **plotting the data**.
- **Short report and quick presentation** on the weekly meeting.

# Milestone 2: Proto 1, a first prototype (2-3 weeks)

- Defining a **hard-coded** “query compilation” for the workflow defined in Milestone 1 and implementing it.
- Defining **quick hardcoded procedures** to get the metrics from the query-compiled application
- Getting the metrics and **plotting the data** (using quick hardcoded scripts)
- **Comparing the results of benchmarking** the baseline (Milestone 1) and Proto 1
- Short report and quick presentation on the weekly meeting

# Milestone 3: Proto 2, more generic Proto 1 (4 weeks)

- Defining an approach for making the **workflow definition** more generic and **having parallel queries**
- Working on the **insights from Proto 1 benchmark results**: are there bottlenecks, or improvement opportunities? How hard it would be to implement them?
- Checking current considered metrics and planning and **defining new metrics** to help evaluating the impact of parallel queries and genericity
- **Improving the procedures** to get the metrics from Proto 2. Making them more reusable?
- **Improving the procedures** to plot the data of Proto 2's benchmark. Making them more reusable?
- **Comparing the results from benchmark** of Milestone 1 to Flink's and Proto 1's. How making it more generic affected the metrics?
- Short report and presentation on the weekly meeting

# Milestone 4: Proto 3, going distributed (4 weeks):

- **Analyzing the challenges** of distributing Proto 2 and defining a plan of attack
- **Implementing** the plan
- Checking current considered metrics and **defining new metrics** to help evaluating the impact of distribution on performance
- **Benchmarking** and comparing results to Flink's and Proto 2's
- Short report and presentation

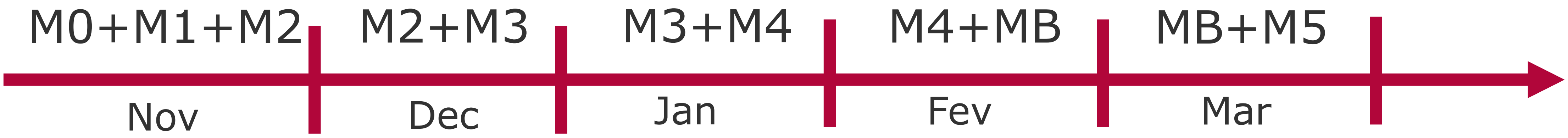
# Bonus Milestone: Proto 4, the modern hardware one

- **Adding RDMA to the soup**. What are the benefits that RDMA could bring to Proto 3? And what are the challenges?
- **Designing** a plan of attack
- **Implementing** the plan
- **Benchmarking** and comparing results to Flink's, Proto 2's and Proto 3's
- Short report and presentation

# Milestone 5: Reporting in, team

- Time to wrap up and write the final report!

# An idea of the daring overview



M0+M1+M2 M2+M3 M3+M4 M4+MB MB+M5

Nov Dec Jan Fev Mar

# Thank you for your attention!

- Questions?
  - pedro.silva@hpi.de
  - tilmann.rabl@hpi.de