Are we solving the core problems in stream processing?

Jonas Traub
Technische Universität Berlin / DFKI IAM
www.dima.tu-berlin.de | jonas.traub@tu-berlin.de
Are we solving the core problems in stream processing?

Yes, we do!
Are we solving the core problems in stream processing?

Yes, we do!

Examples

Apache Flink and its success story

What are the core problems and how are we solving them
Apache Flink Timeline

- 2008: Initial Vision for a Big Data Analytics Platform
- 2009: DFG Proposal for Stratosphere I
- 2010: Grant Award Start of Stratosphere I
- 2010: Nephele/PACTs Paper Published
- 2011: Stratosphere System Paper Published
- 2012: Vldb Paper Published
- 2014: Berlin Big Data Center Founded
- 2014: Flink Incubator Project
- 2014: Flink Top Level Project
- 2015: DataArtisans Founded
- 2015: Flink Forward 1st Flink Forward Conference
- 2016: Flink Community Groups Across Europe
- 2016: Flink Forward 2nd Flink Forward Conference
- 2017: Flink Forward Conference in Berlin
- 2017: Premier Flink Forward Conference in San Francisco

21 Meetups Worldwide
250 Contributors
516 Members
26 Cities
14 Countries

Berlin 830
Paris 520
Madrid 457
Stockholm 337
Brussels 325
London 315
Munich 171
Amsterdam 124
Istanbul 67
Apache Flink

https://www.meetup.com/topics/apache-flink/
https://flink.apache.org/poweredby.html

Flink Community

- 21,000+ Meetup Members Worldwide
- 400+ Open Source Contributors/Developers
- 41 Meetup Groups Worldwide

Flink Contributors

- 17 Countries that Regularly Hold Meetups
- 36+ Companies using Apache Flink
- Startup data Artisans founded in 2014
Apache Flink - Stateful Computations over Data Streams

- Event-driven Applications
- Stream & Batch Analytics
- Data Pipelines & ETL
- Exactly-once state consistency
- Event-time processing
- Sophisticated late data handling
- Scale-out architecture
- Support for very large state
- Incremental checkpointing

source: flink.apache.org
Examples:
What are core problems and how are we solving them?
Examples:

Expressiveness: Event-time processing and sophisticated late data handling

The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing (Akidau et al.)
Examples:

**Expressiveness:** Event-time processing and sophisticated late data handling

The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing (Akidau et al.)

**Service:** Common APIs and feature sets

Apache Beam: An advanced unified programming model

“Implement batch and streaming data processing jobs that run on any execution engine. (beam.apache.org)”
Examples:

Expressiveness: Event-time processing and sophisticated late data handling

The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing (Akidau et al.)

Service: Common APIs and feature sets

Apache Beam: An advanced unified programming model
“Implement batch and streaming data processing jobs that run on any execution engine. (beam.apache.org)”

Consistency: Exactly-once state consistency

Lightweight asynchronous snapshots for distributed dataflows
P Carbone, G Fóra, S Ewen, S Haridi, K Tzoumas

State management in Apache Flink: consistent stateful distributed stream processing
P Carbone, S Ewen, G Fóra, S Haridi, S Richter, K Tzoumas