

Apache Flink® Training

System Overview



Apache Flink® Training

dataArtisans

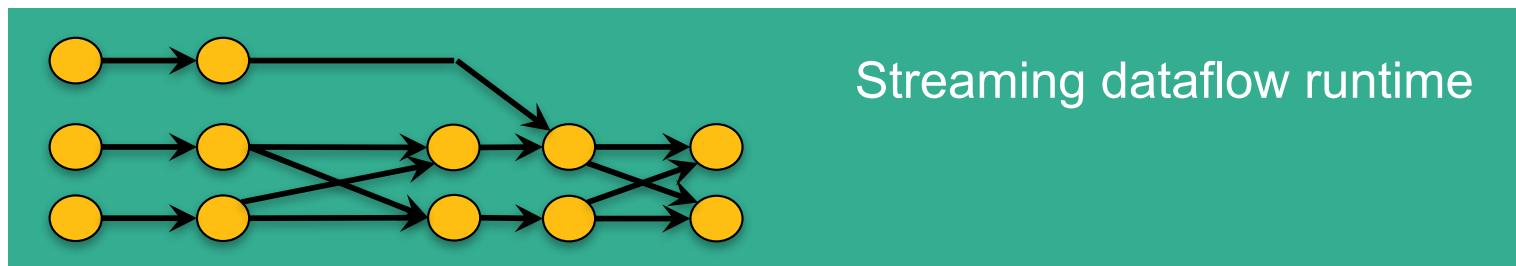
Flink v1.2 – 04.05.2017

What is Flink?



Apache Flink®

*A stream processor
with many applications*



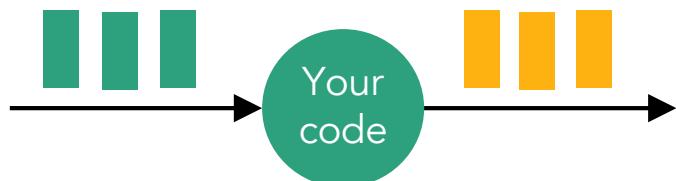
What is Stream Processing?



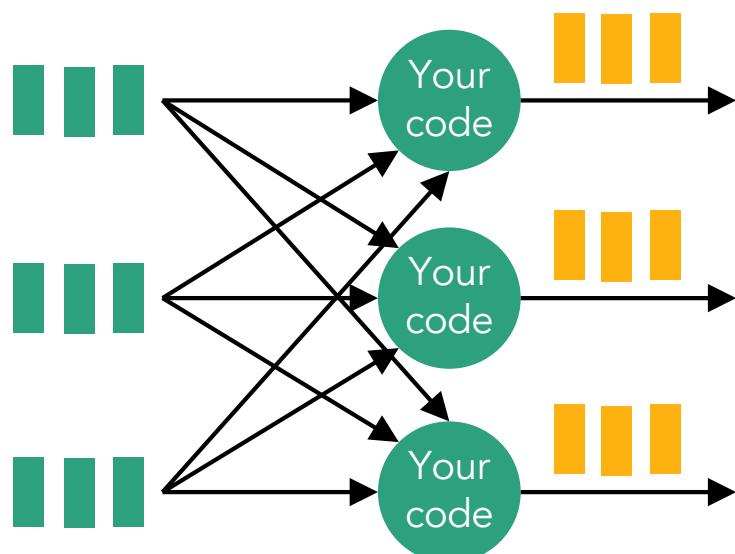
- Today, most data is continuously produced
 - web logs, sensors, database transactions, ...
- The common approach so far
 - Record stream to stable storage (DBMS, HDFS, ...)
 - Analyze data with periodic batch jobs
- Stream processors analyze data as it arrives



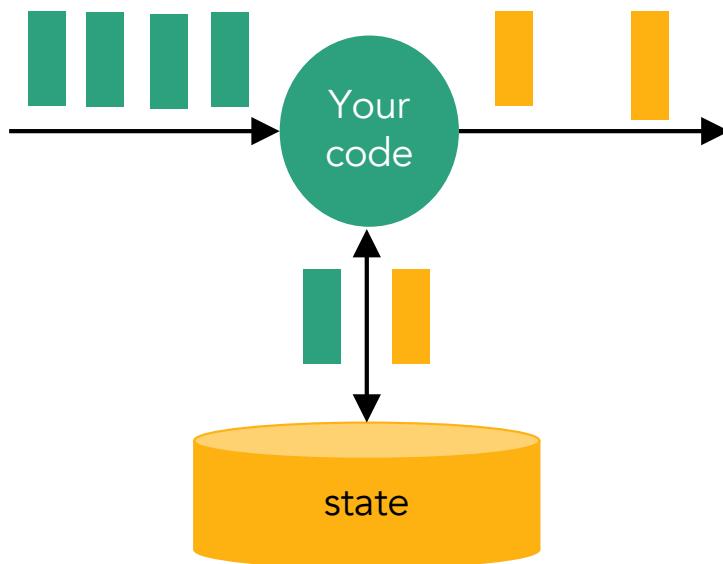
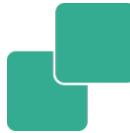
Distributed streaming



- Computations on never-ending “streams” of data records (“events”)
- A stream processor distributes the computation in a cluster
- Low latency, high throughput

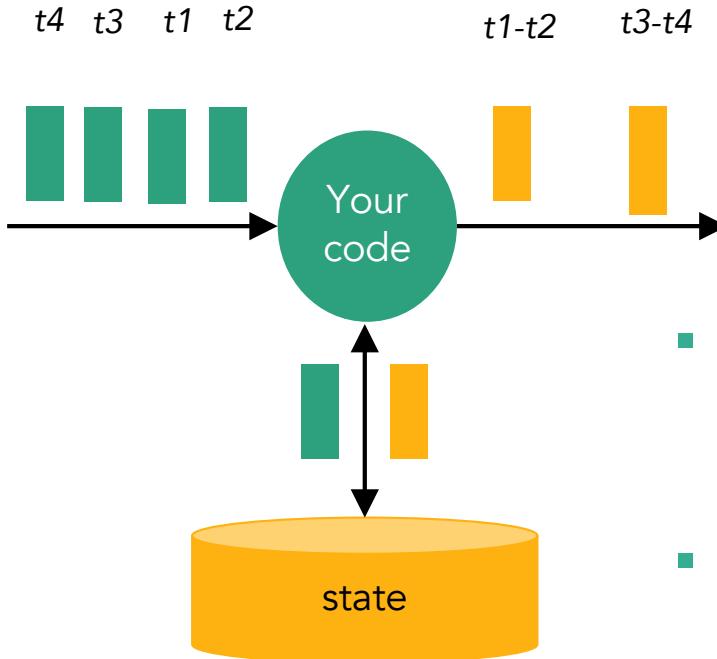


Stateful streaming

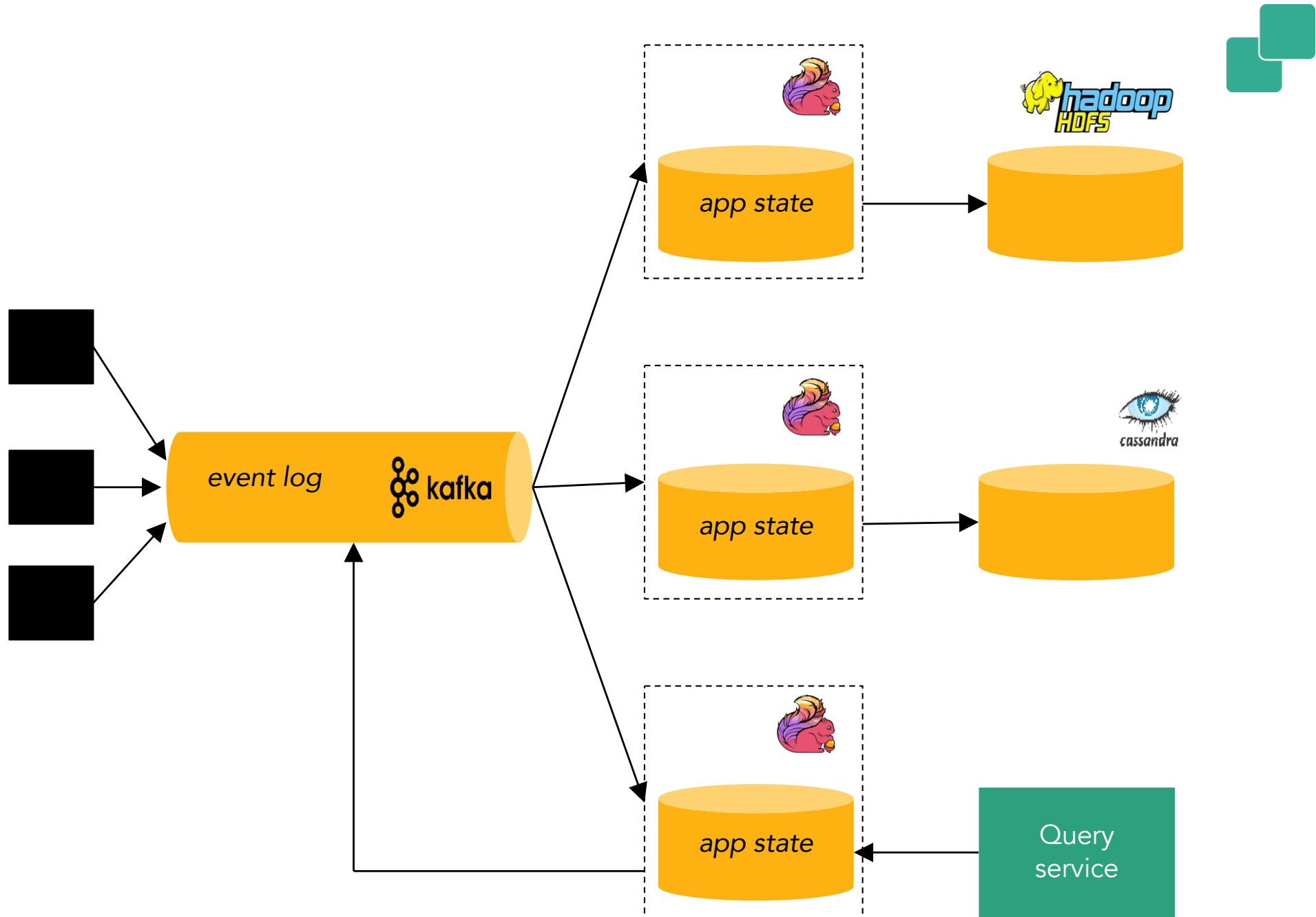


- **Computation and state**
 - E.g., counters, windows of past events, state machines, trained ML models
 - Results depend on history of stream
- **Fault-tolerant, with exactly-once consistency**
- **A stateful stream processor provides tools to manage state**
 - Recover, roll back, version, upgrade, ...

Event-time streaming

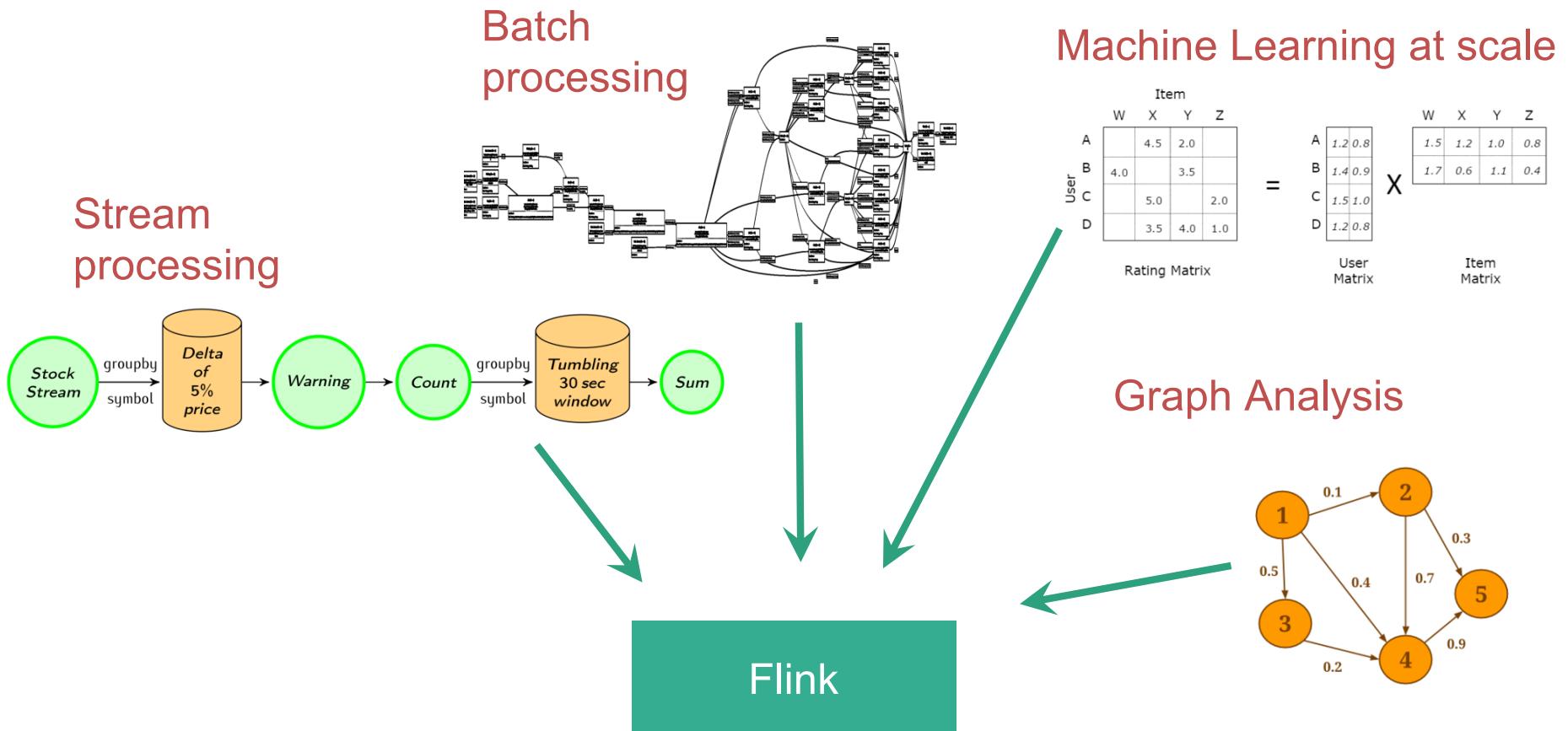


- Makes the time dimension of data explicit
- Processing depends on timestamps
- An event-time stream processor gives you the tools to reason about time
 - E.g., handle streams that are out of order





Native support for various workloads





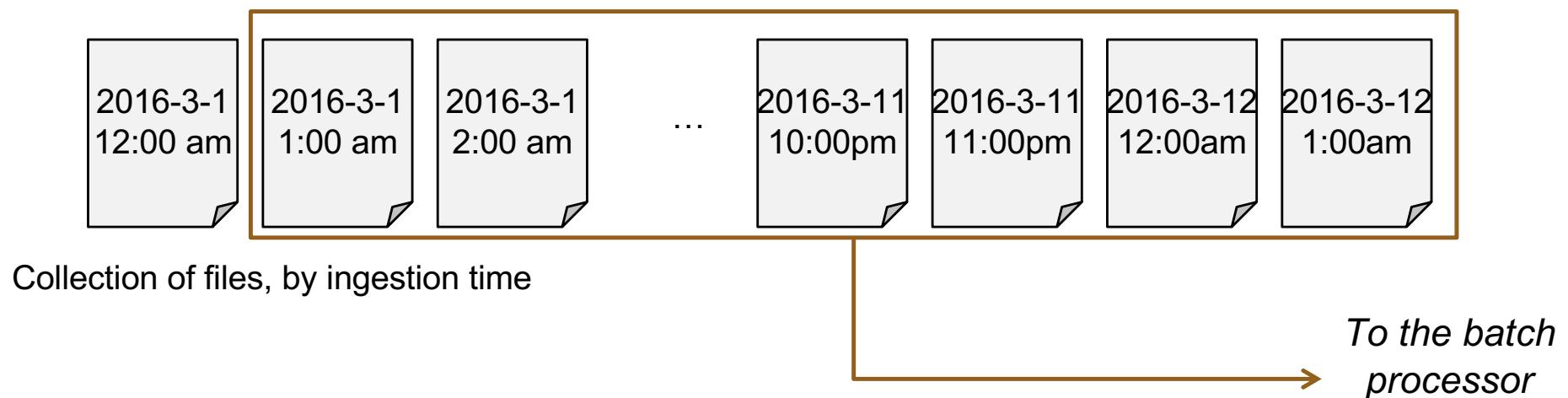
Benefits of a streaming architecture

- More real-time reaction to events
- Robust continuous applications
 - Continuous batch apps are duck-taped together from many tools
- Process both real-time and historical data
 - Using exactly the same application

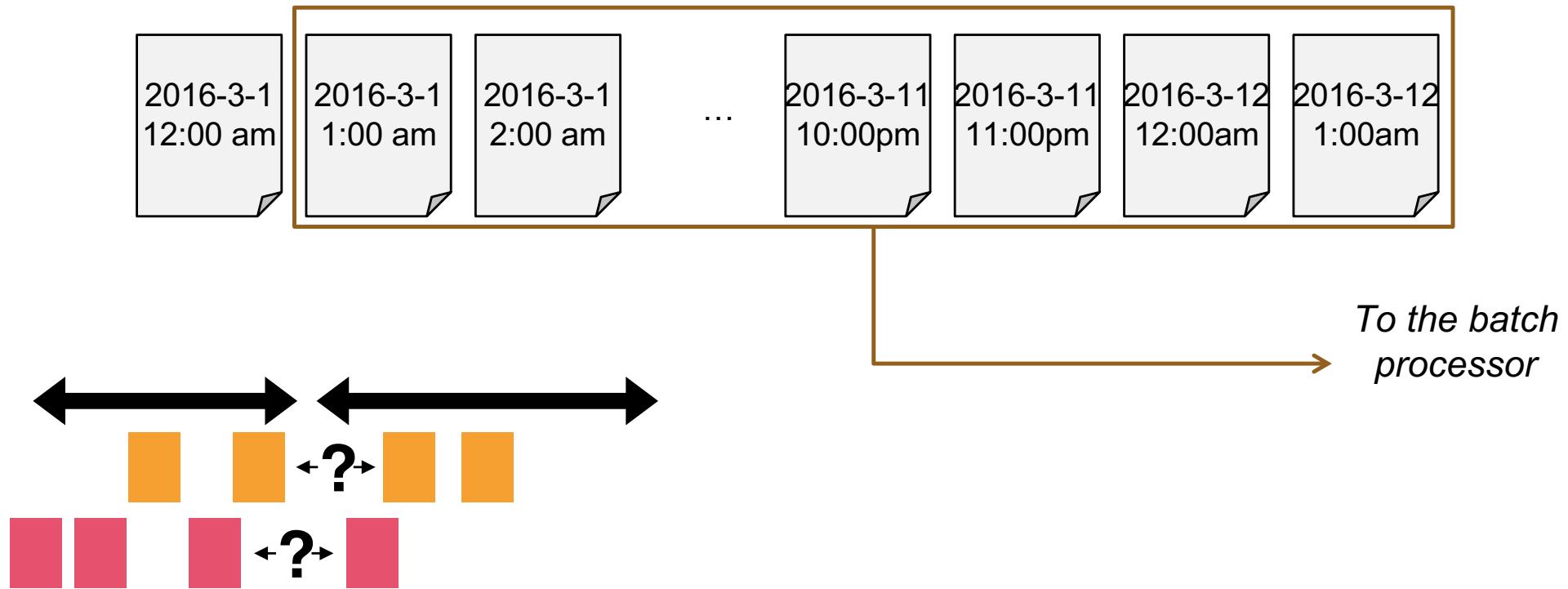


(Re)processing data (in batch)

- Re-processing data (what-if exploration, to correct bugs, etc.)
- Usually by running a batch job with a set of old files
- Tools that map files to times



Unclear Batch Boundaries

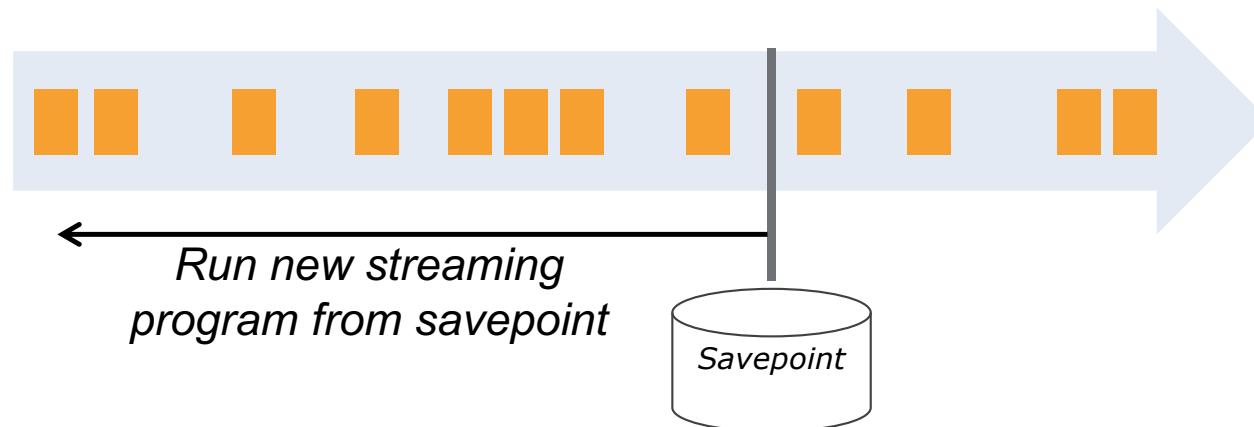


What about **sessions** across batches?

(Re)processing data (streaming)



- Draw savepoints at times that you will want to start new jobs from (daily, hourly, ...)
- Reprocess by starting a new job from a savepoint
 - Defines start position in stream (for example Kafka offsets)
 - Initializes pending state (like partial sessions)



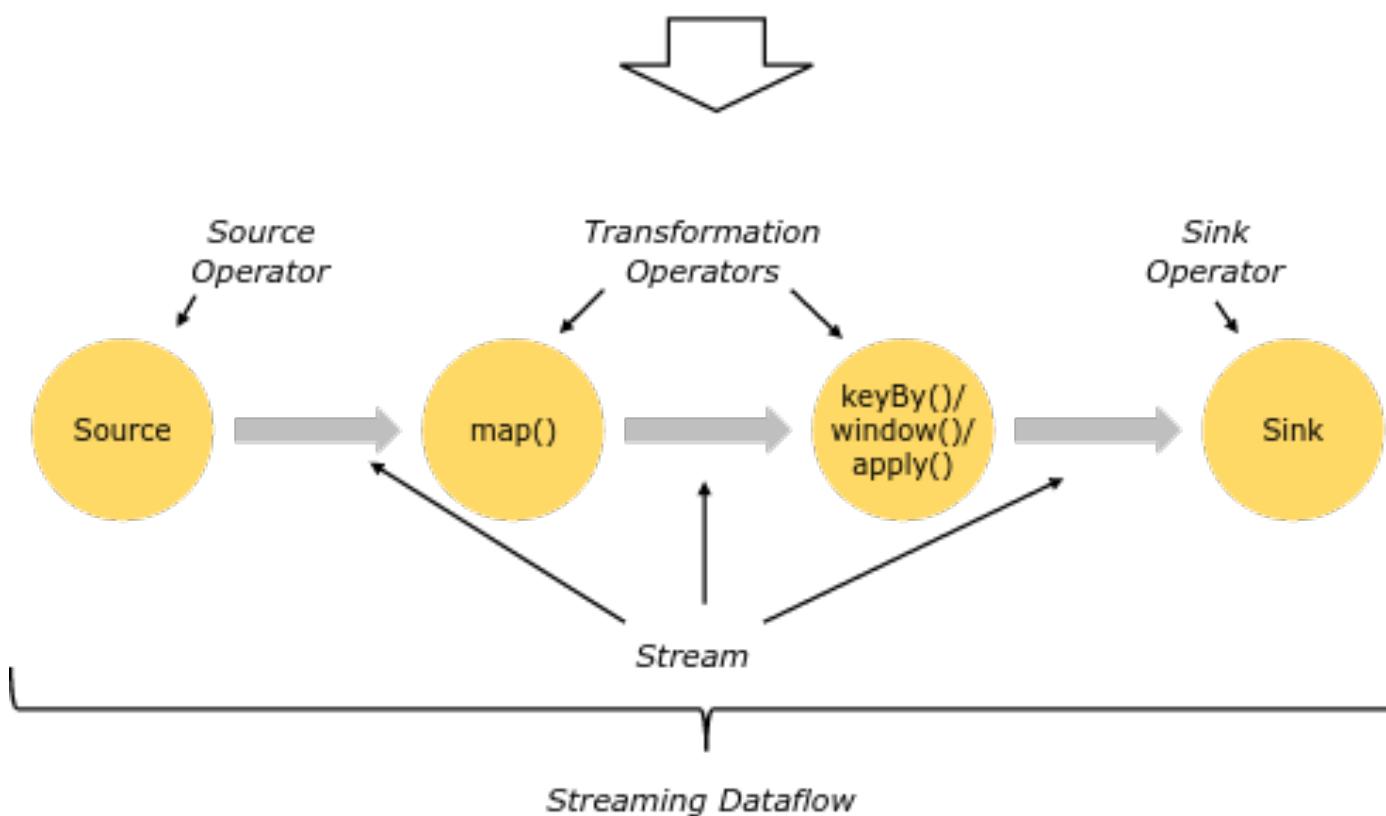


Accurate computation

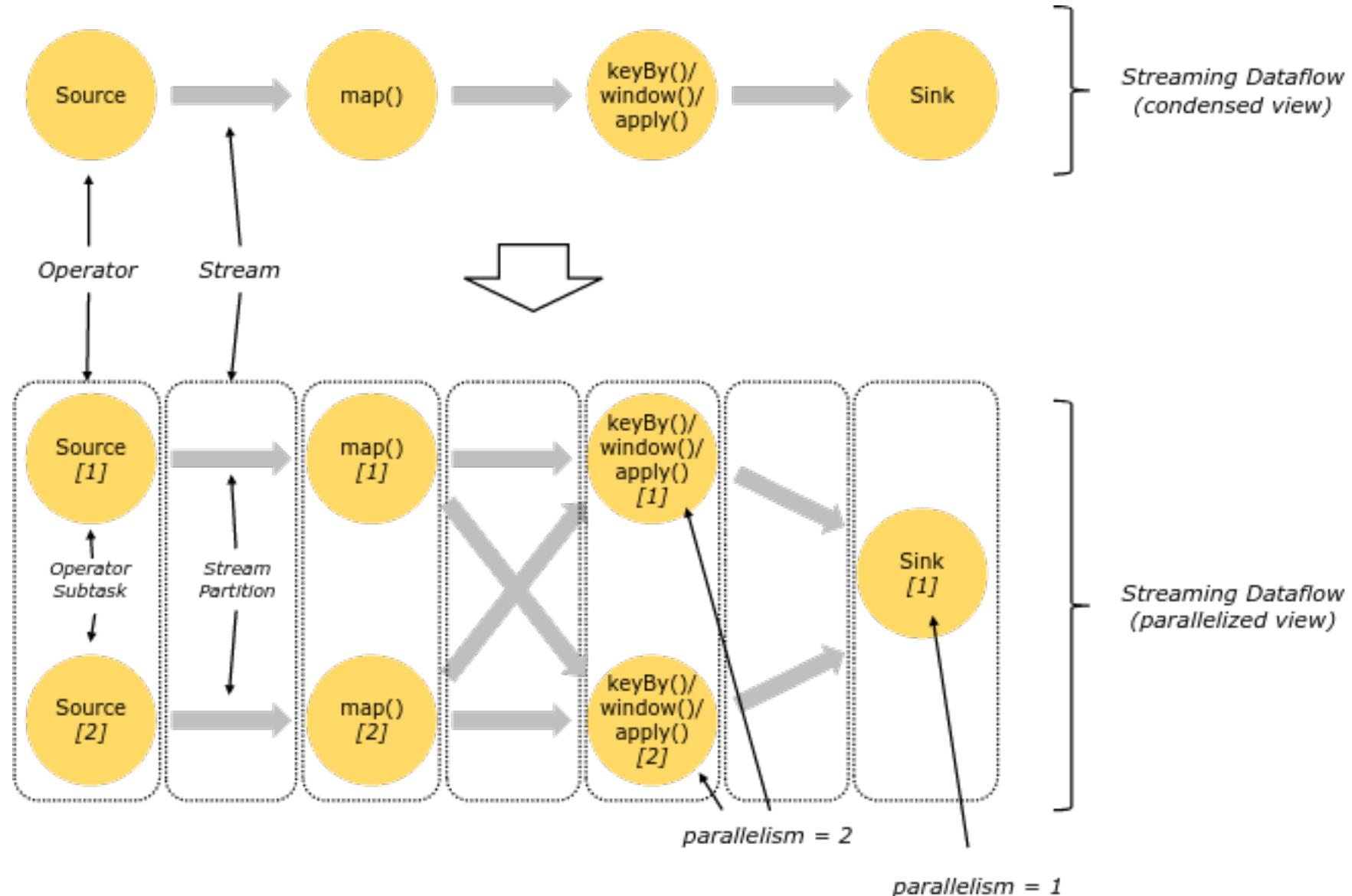
- Batch processing is not an accurate computation model for continuous data
 - Misses the right concepts and primitives
 - Time handling, state across batch boundaries
- Stateful stream processing a better model
 - Can achieve high throughput and low latency while robustly delivering accurate results
 - Real-time/low-latency is the **icing on the cake**

How does Flink execute my application?

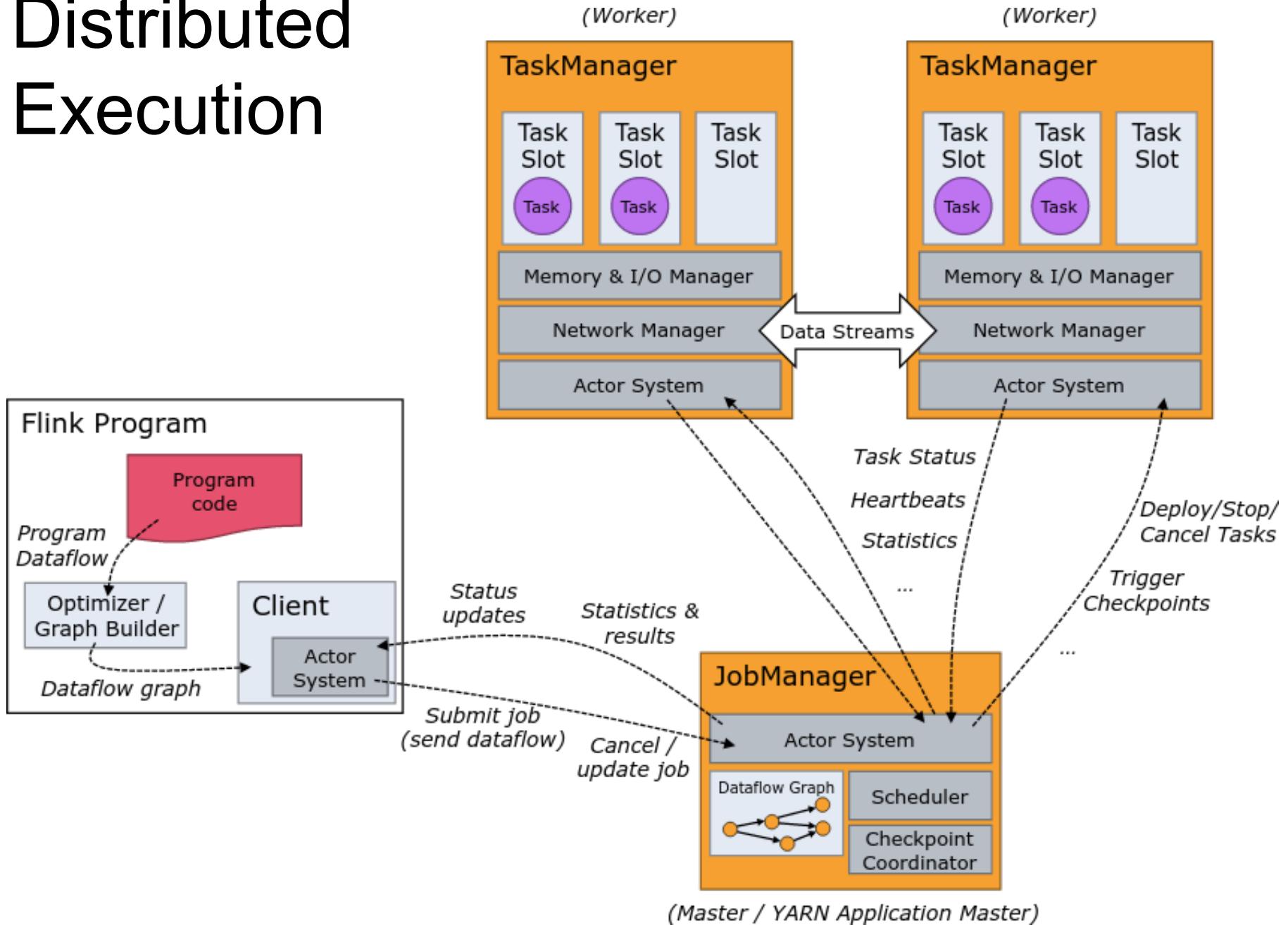
```
DataStream<String> lines = env.addSource(  
    new FlinkKafkaConsumer<>(...)); } Source  
  
DataStream<Event> events = lines.map((line) -> parse(line)); } Transformation  
  
DataStream<Statistics> stats = events  
    .keyBy("id")  
    .timeWindow(Time.seconds(10))  
    .apply(new MyWindowAggregationFunction()); } Transformation  
  
stats.addSink(new RollingSink(path)); } Sink
```



Parallelism



Distributed Execution

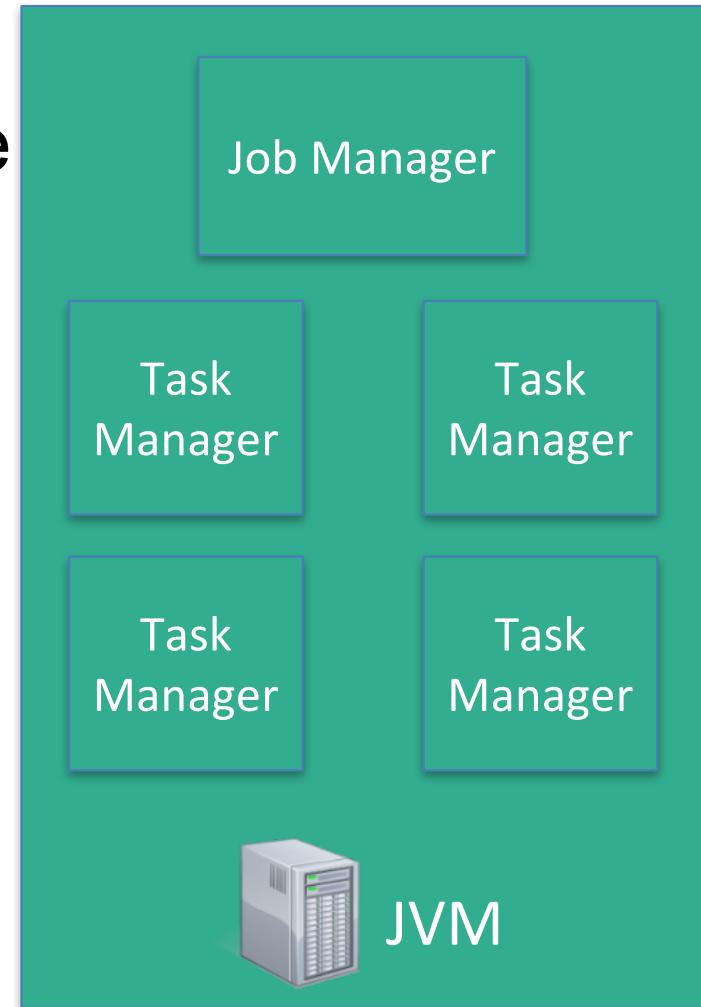


Deployment Options

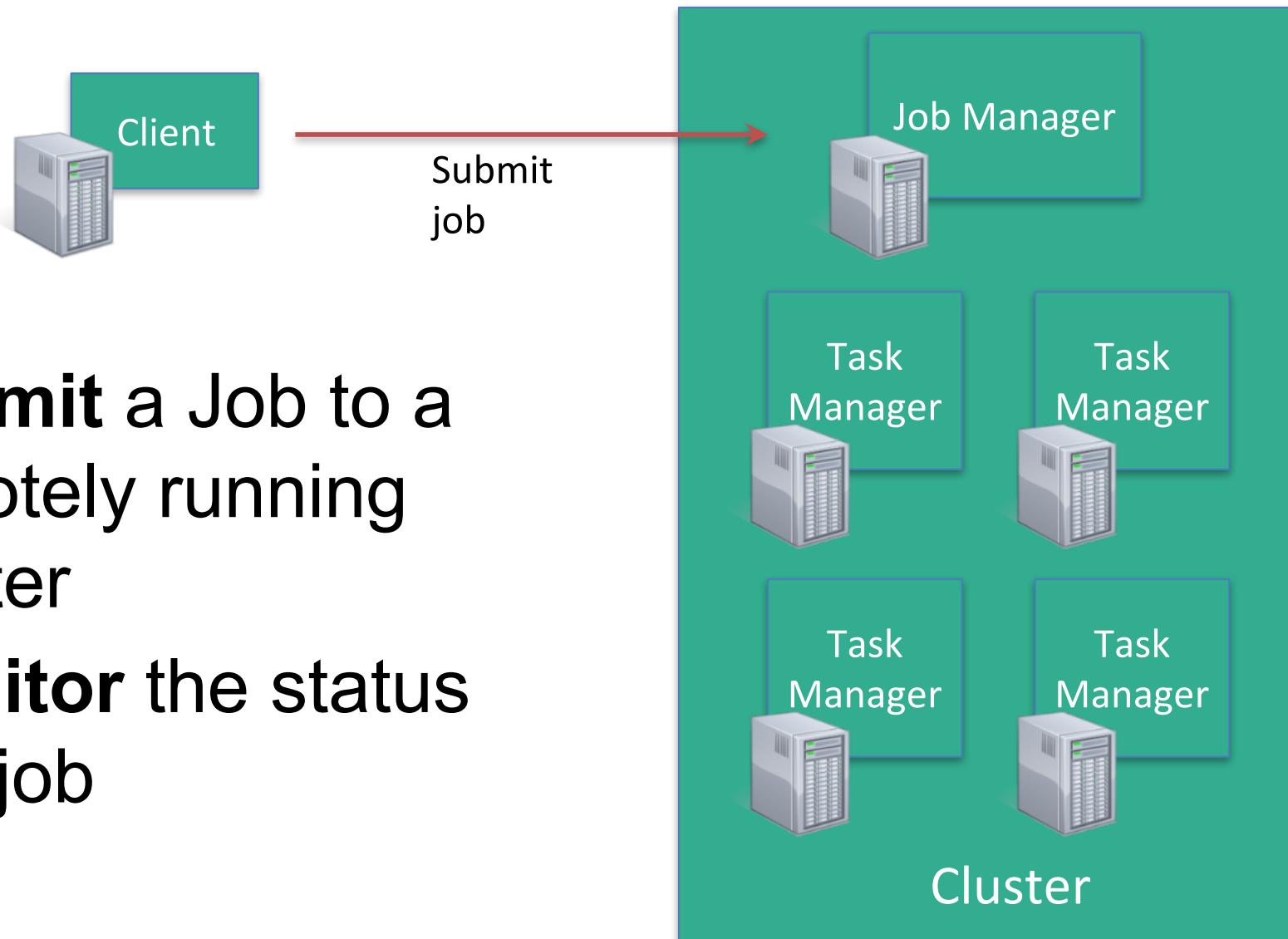
Local Execution



- Starts local Flink cluster
- All processes run in the same JVM
- Behaves just like a regular Cluster
- Local cluster can be started in your IDE!
- Very useful for developing and debugging



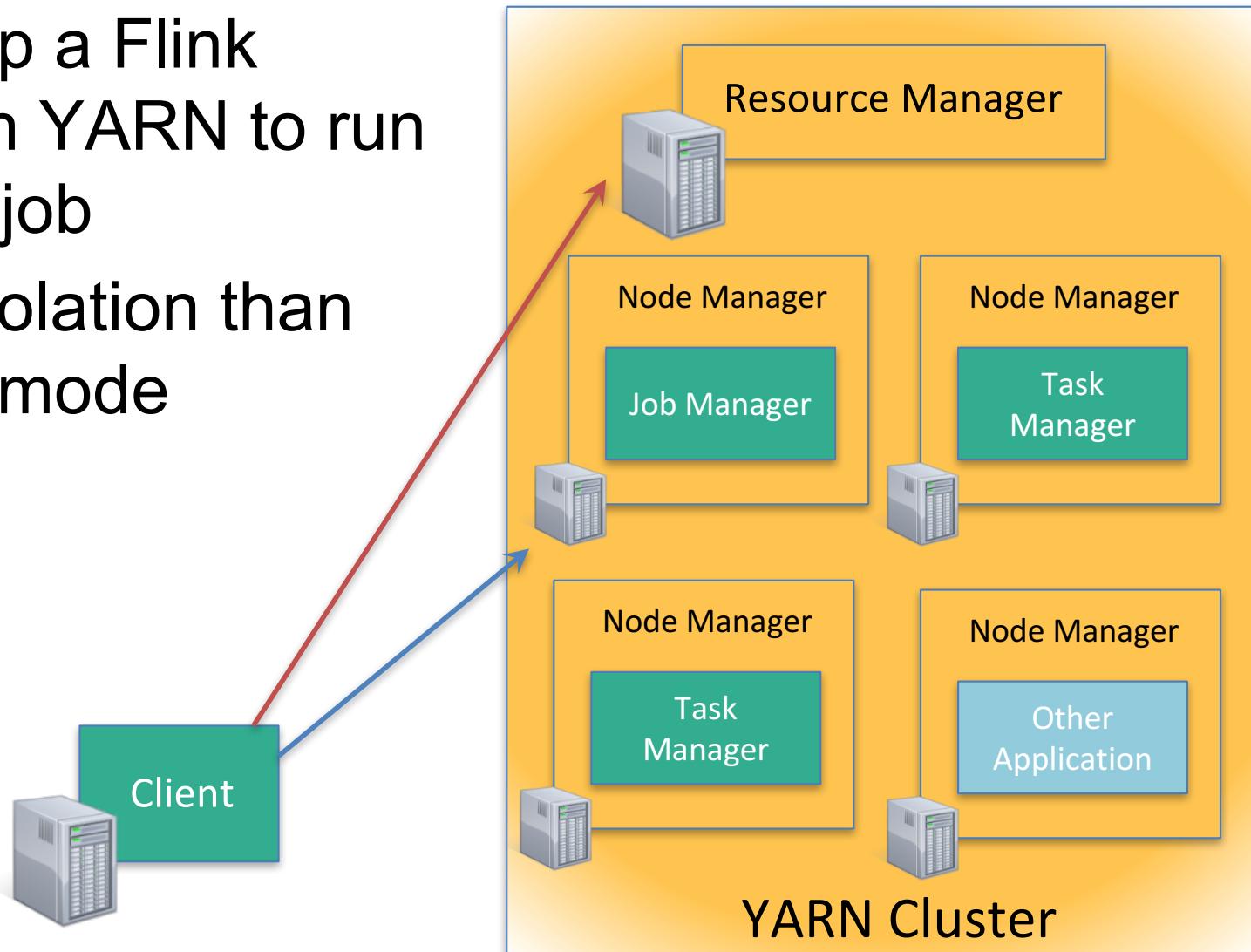
Remote Execution



YARN Job Mode



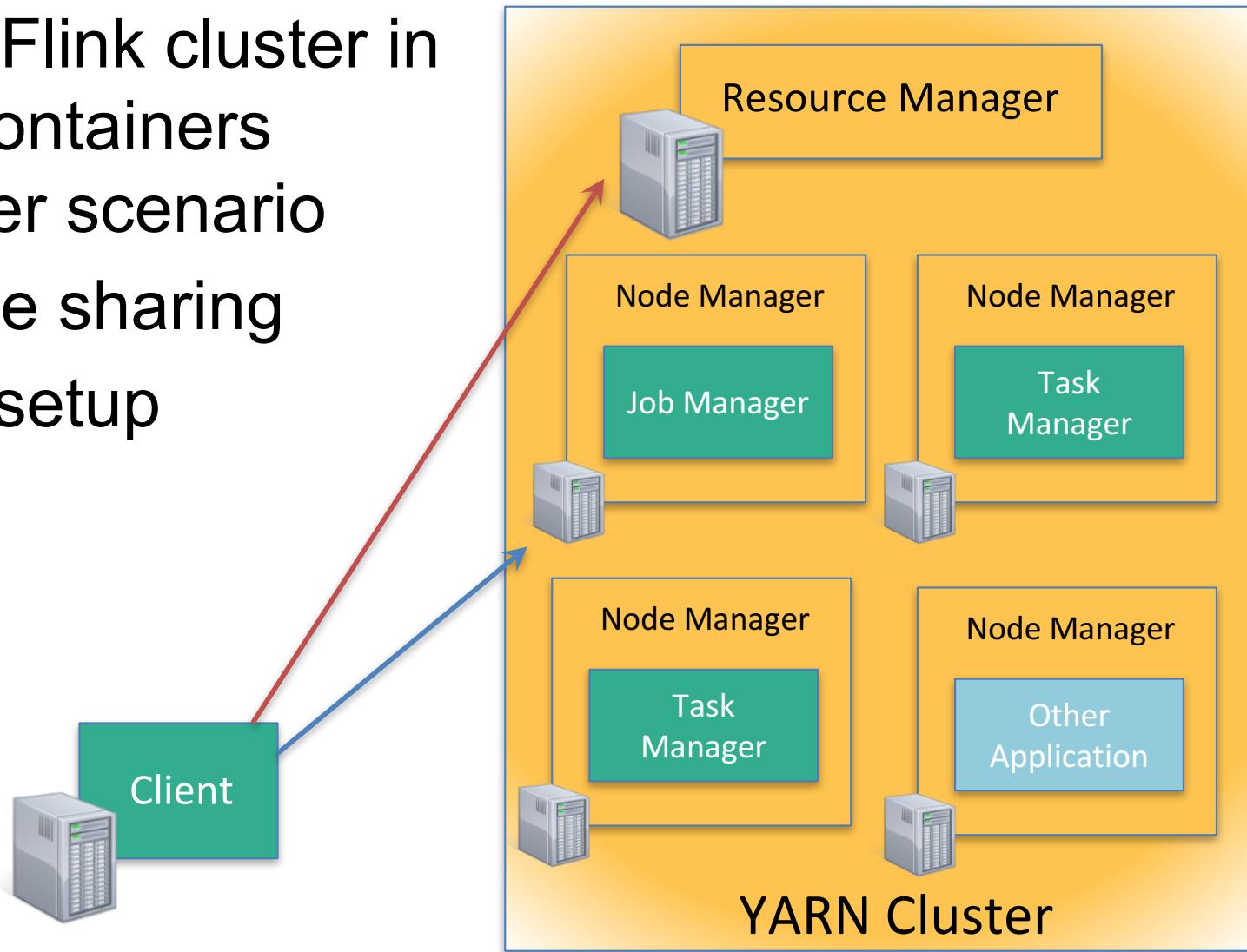
- Brings up a Flink cluster in YARN to run a single job
- Better isolation than session mode



YARN Session Mode



- Starts a Flink cluster in YARN containers
- Multi-user scenario
- Resource sharing
- Easy to setup





Other Deployment Options

- Apache Mesos
 - Either with or without DC/OS
- Amazon Elastic MapReduce
 - Available in EMR 5.1.0
- Google Compute Engine
 - Available via bdutil

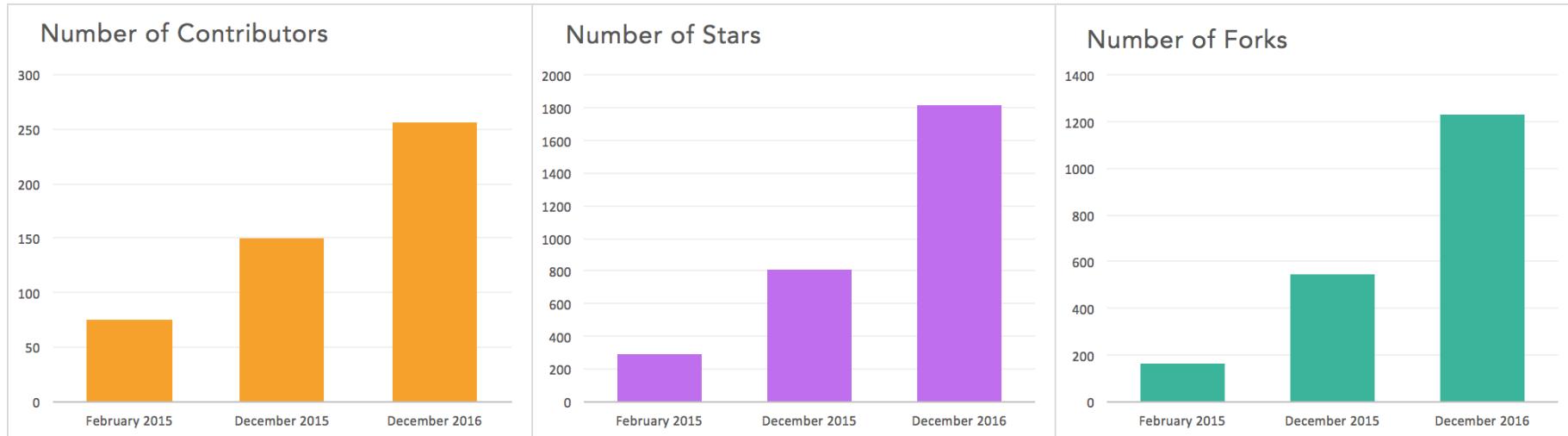


Flink in the real world

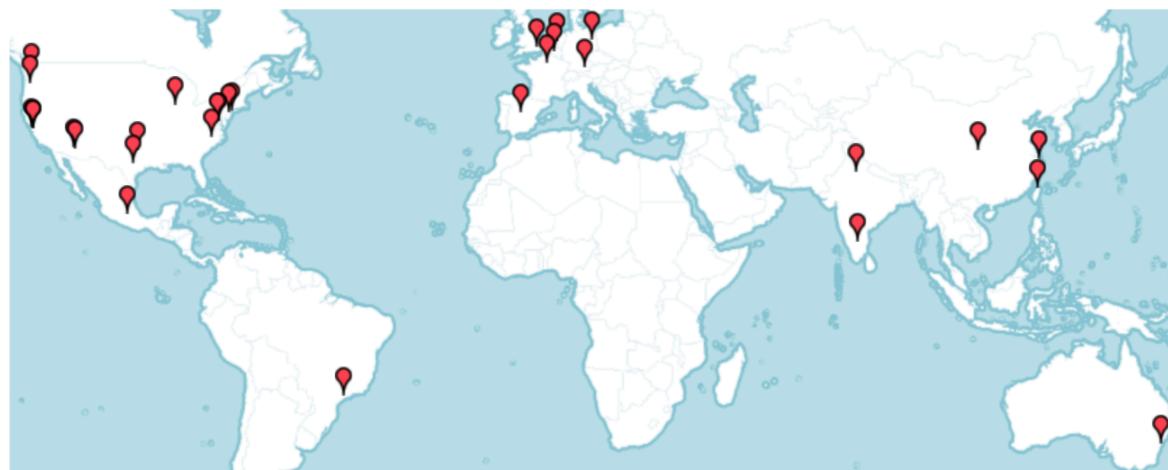


Flink community

Github



41 meetups
16,544 members



Powered by Flink



Zalando, one of the largest ecommerce companies in Europe, uses Flink for real-time business process monitoring.



King, the creators of Candy Crush Saga, uses Flink to provide data science teams with real-time analytics.



Alibaba, the world's largest retailer, built a Flink-based system (Blink) to optimize search rankings in real time.



Bouygues Telecom uses Flink for real-time event processing over billions of Kafka messages per day.

See more at flink.apache.org/poweredsby.html





Largest job has > 20 operators, runs on > 5000 vCores in 1000-node cluster, processes millions of events per second



Complex jobs of > 30 operators running 24/7, processing 30 billion events daily, maintaining state of 100s of GB with exactly-once guarantees



30 Flink applications in production for more than one year. 10 billion events (2TB) processed daily



Flink Forward 2016

dataArtisans

EMC²

Google

King



zalando

otto group



redhat.



ResearchGate



NETFLIX



CapitalOne

cloudera

MAPR

TNG TECHNOLOGY
CONSULTING

SICS

MTA SZTAKI



mgm

**people
pattern**

CRS4
IDEAS BECOME LIFE

ScaDS
DRESDEN LEIPZIG

Flink Forward 2017



San Francisco

- 10-11 April 2017
- The first Flink Forward event outside of Berlin
- Talks are online at sf.flink-forward.org/

Berlin

- 11-13 September 2017
- Over 350 attendees at the last event
- Registration opening soon!



<http://dataartisans.github.io/flink-training/>