

Apache Flink® Training

System Overview



Apache Flink® Training
dataArtisans

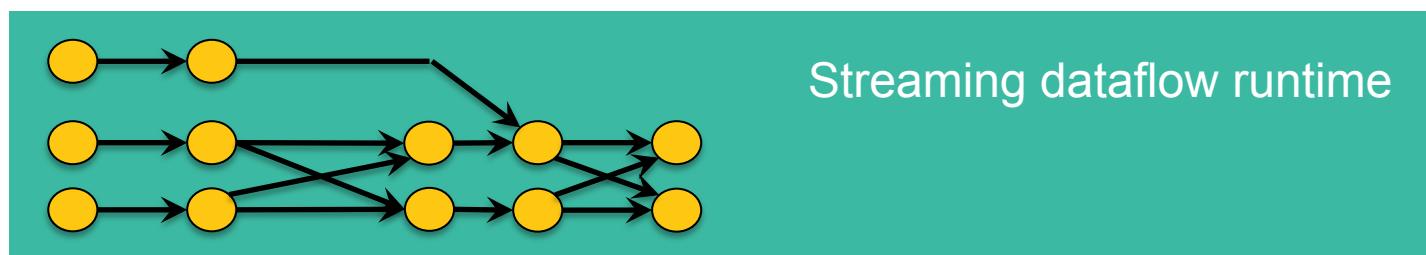
Flink v1.2 – 27.02.2017

What is Flink?



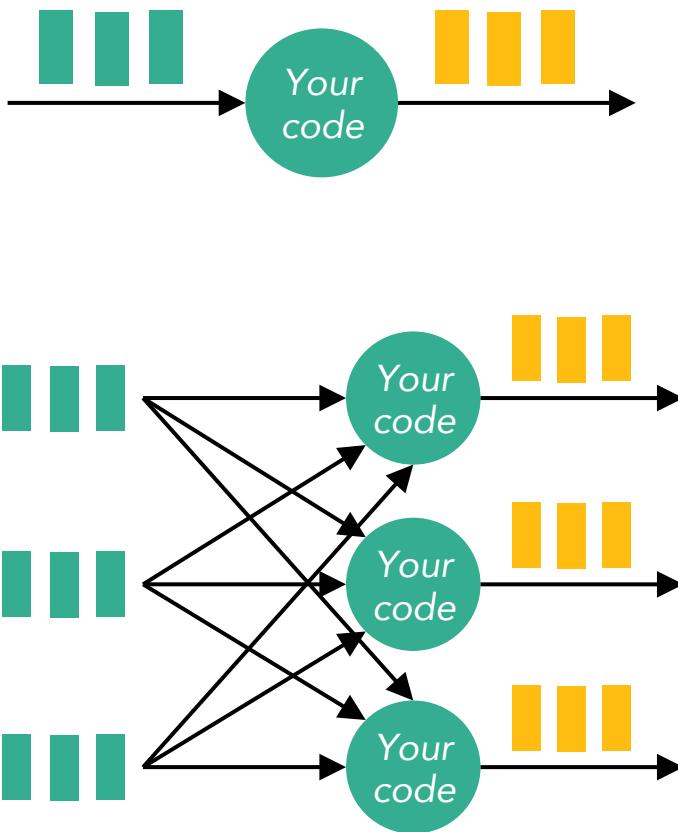
Apache Flink®

*A stream processor
with many applications*





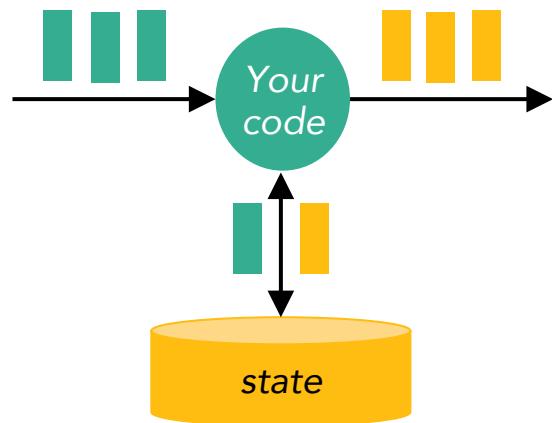
(Distributed) streaming



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



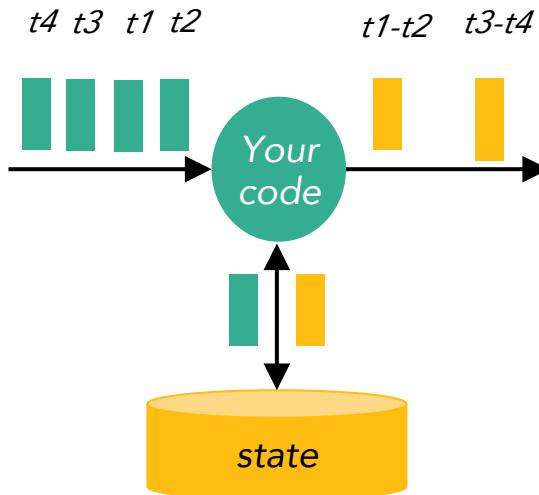
Stateful streaming



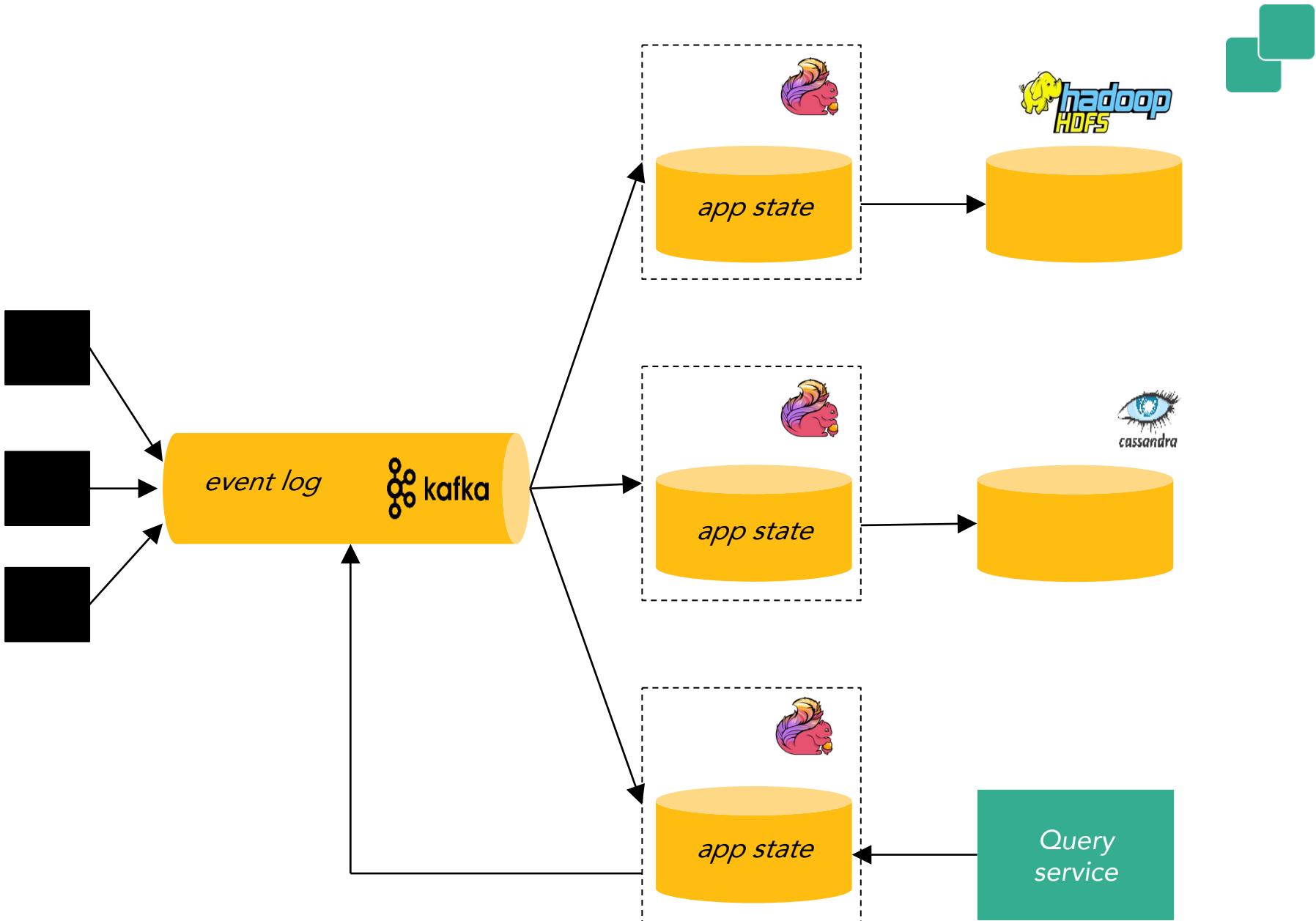
- Computation *and* state
 - E.g., counters, windows of past events, state machines, trained ML models
- Results depend on history of stream
- A stateful stream processor provides tools to manage state
 - Recover, roll back, version, upgrade, etc.



Event-time streaming

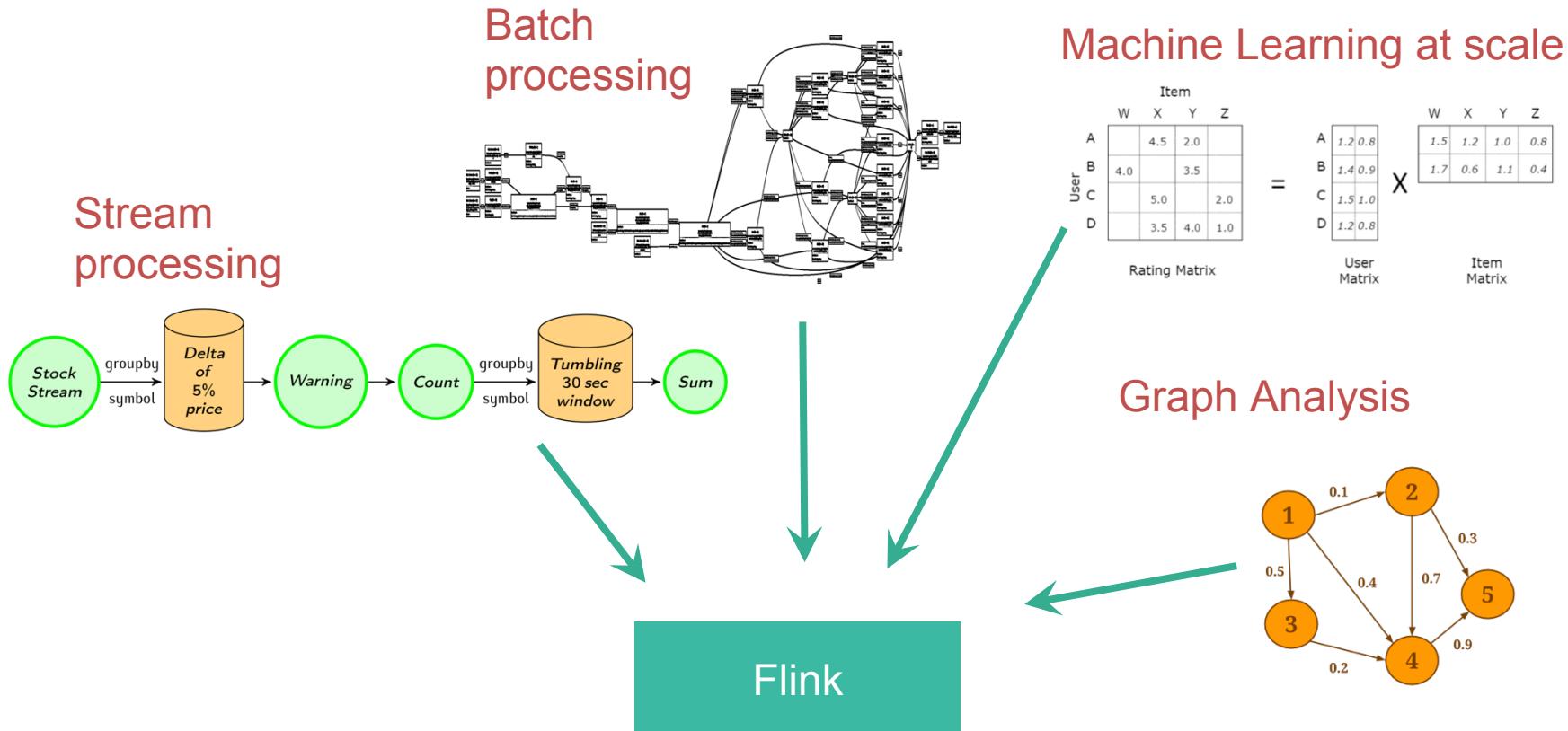


- Data records associated with timestamps (time series data)
- Processing depends on timestamps
- An event-time stream processor gives you the tools to reason about time
 - E.g., handle streams that are out of order





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

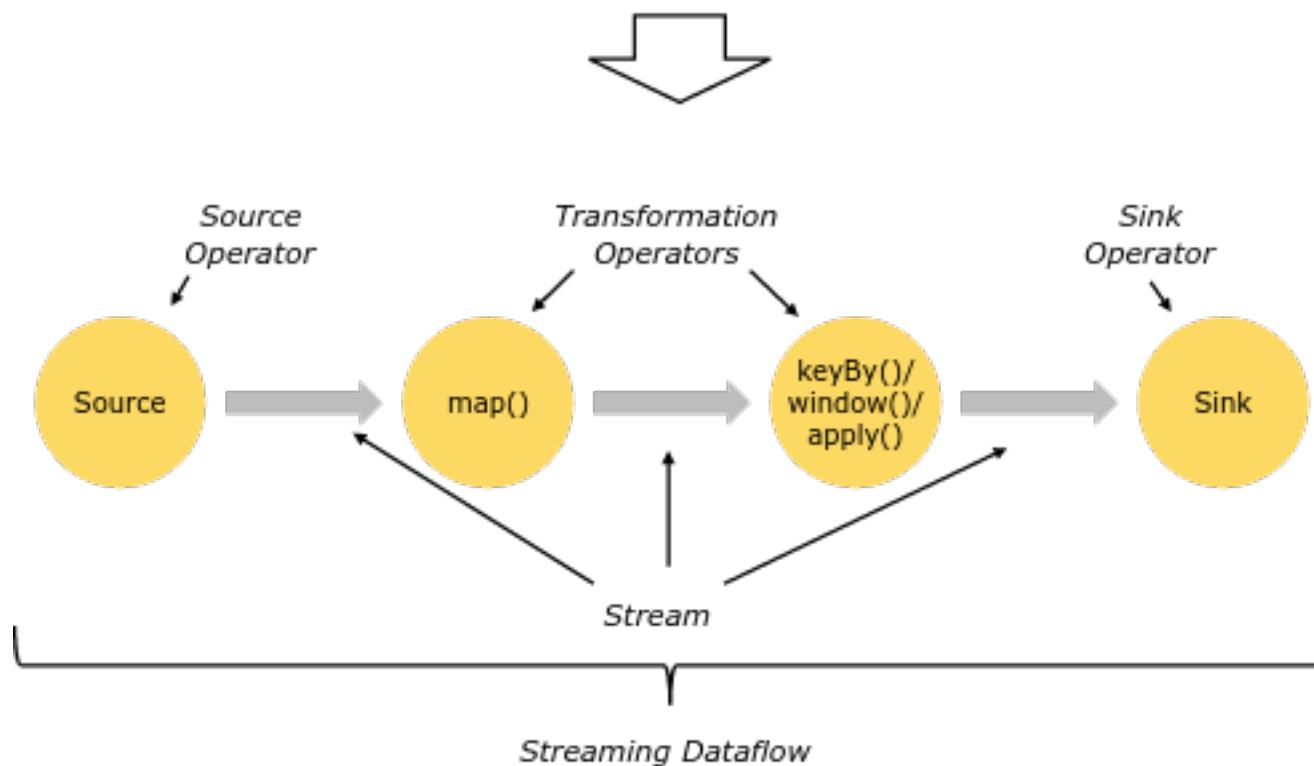


Accurate computation

- 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

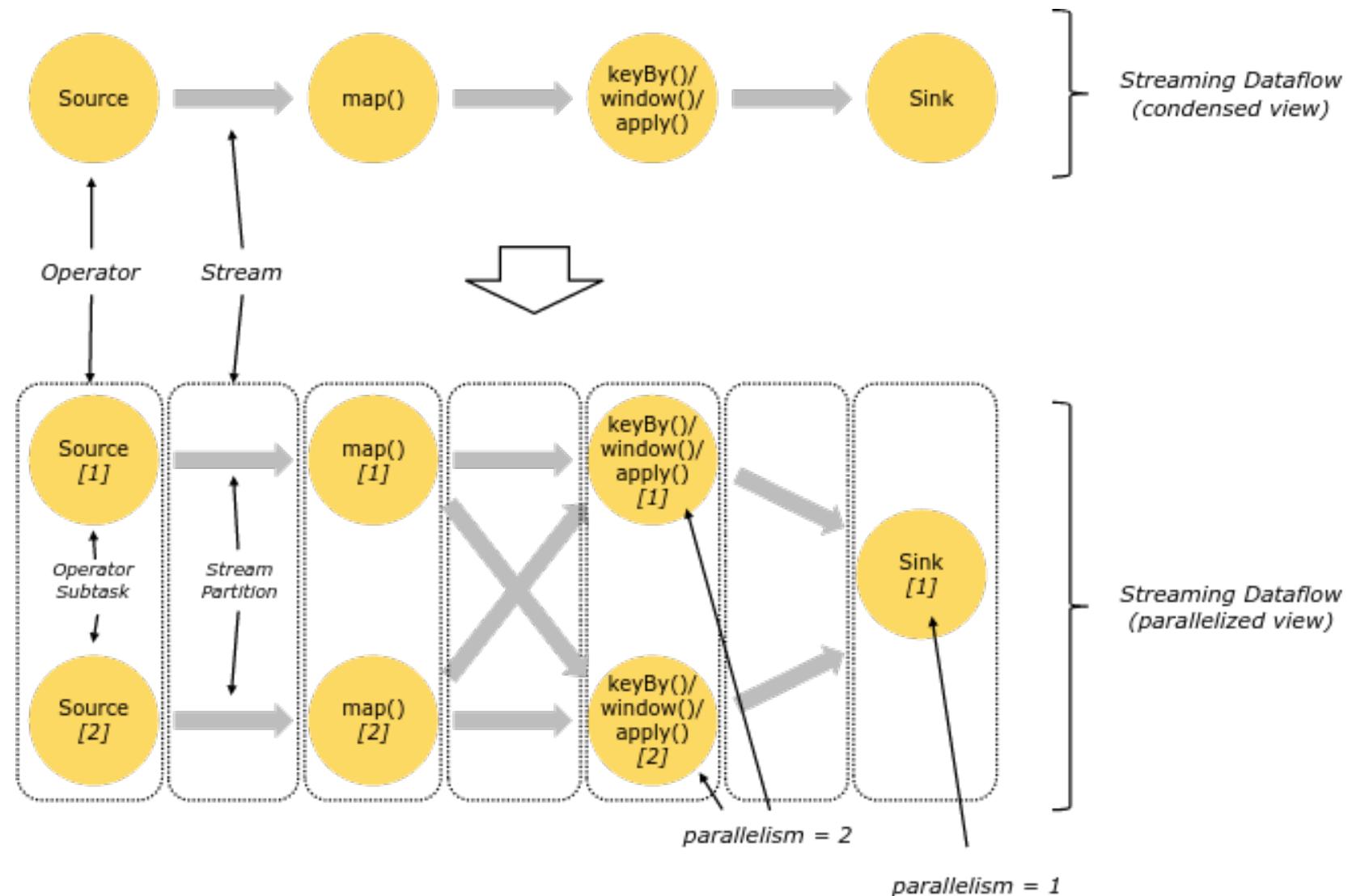
How does Flink execute my application?

```
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
```

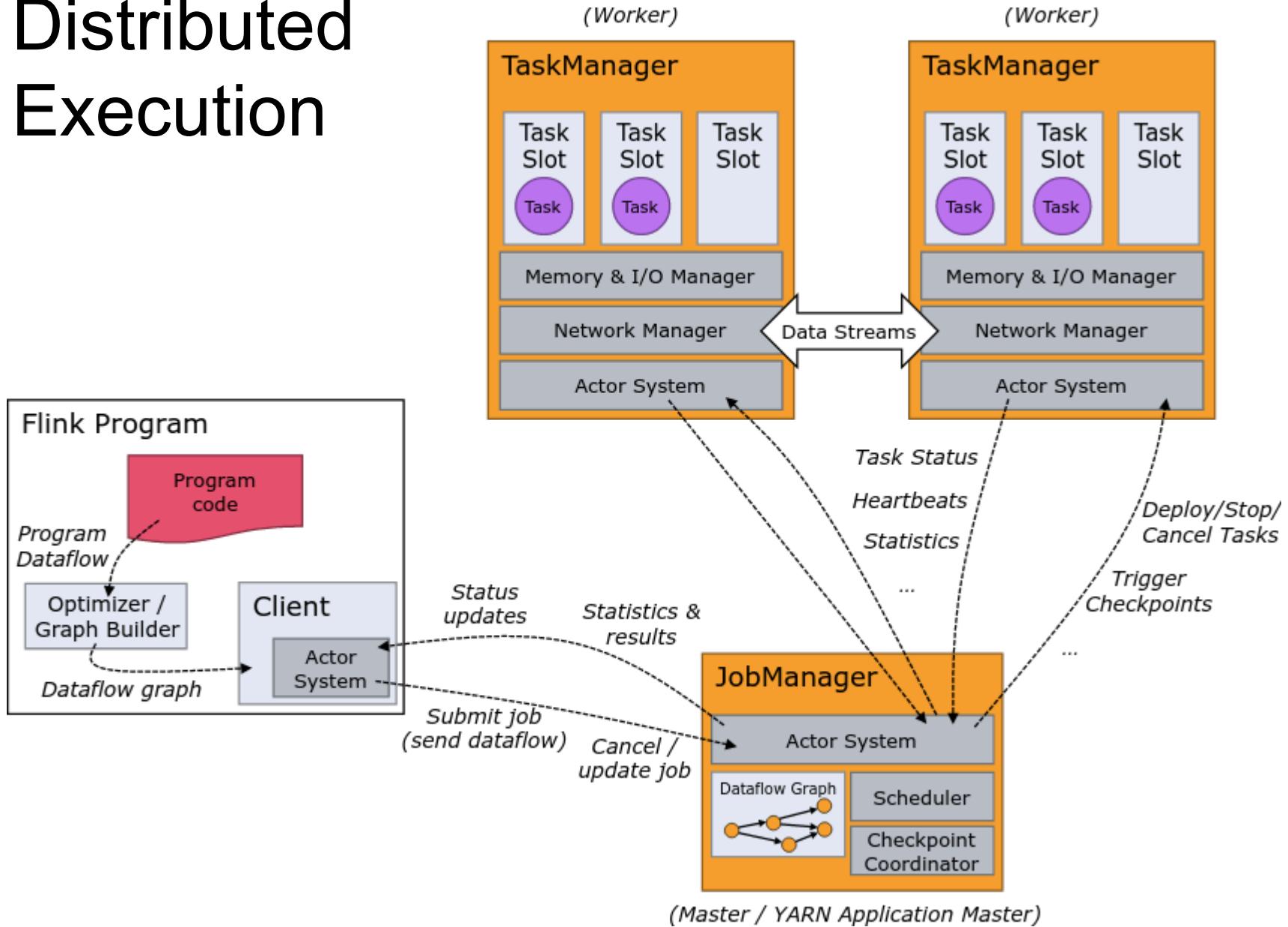




Parallelism



Distributed Execution



Deployment Options



Local Execution

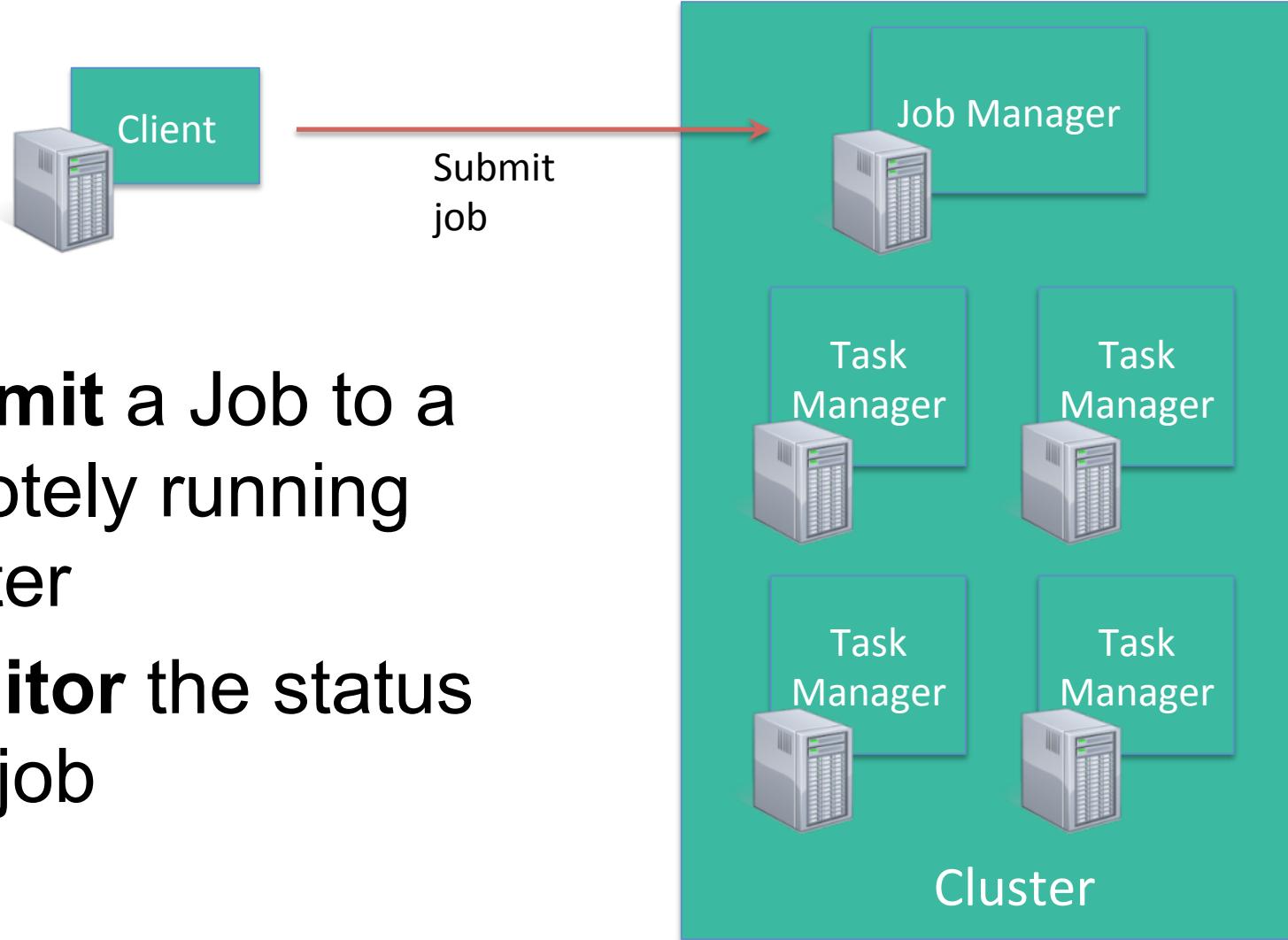
- Starts local Flink cluster
- All processes run in the same JVM
- Behaves just like a regular Cluster
- Local cluster can be started in your IDE!
- Very useful for developing and debugging



Remote Execution



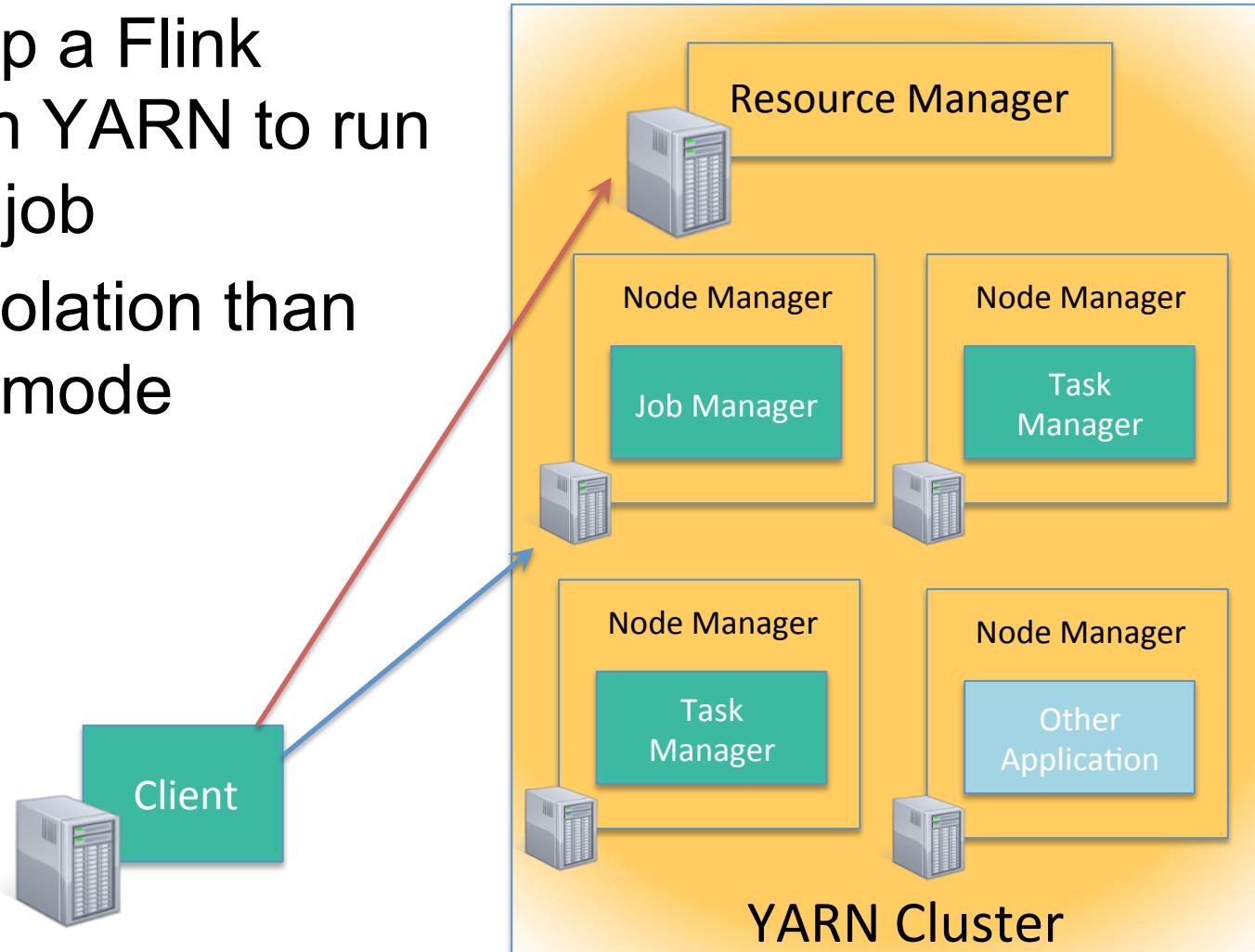
- **Submit** a Job to a remotely running cluster
- **Monitor** the status of a job





YARN Job Mode

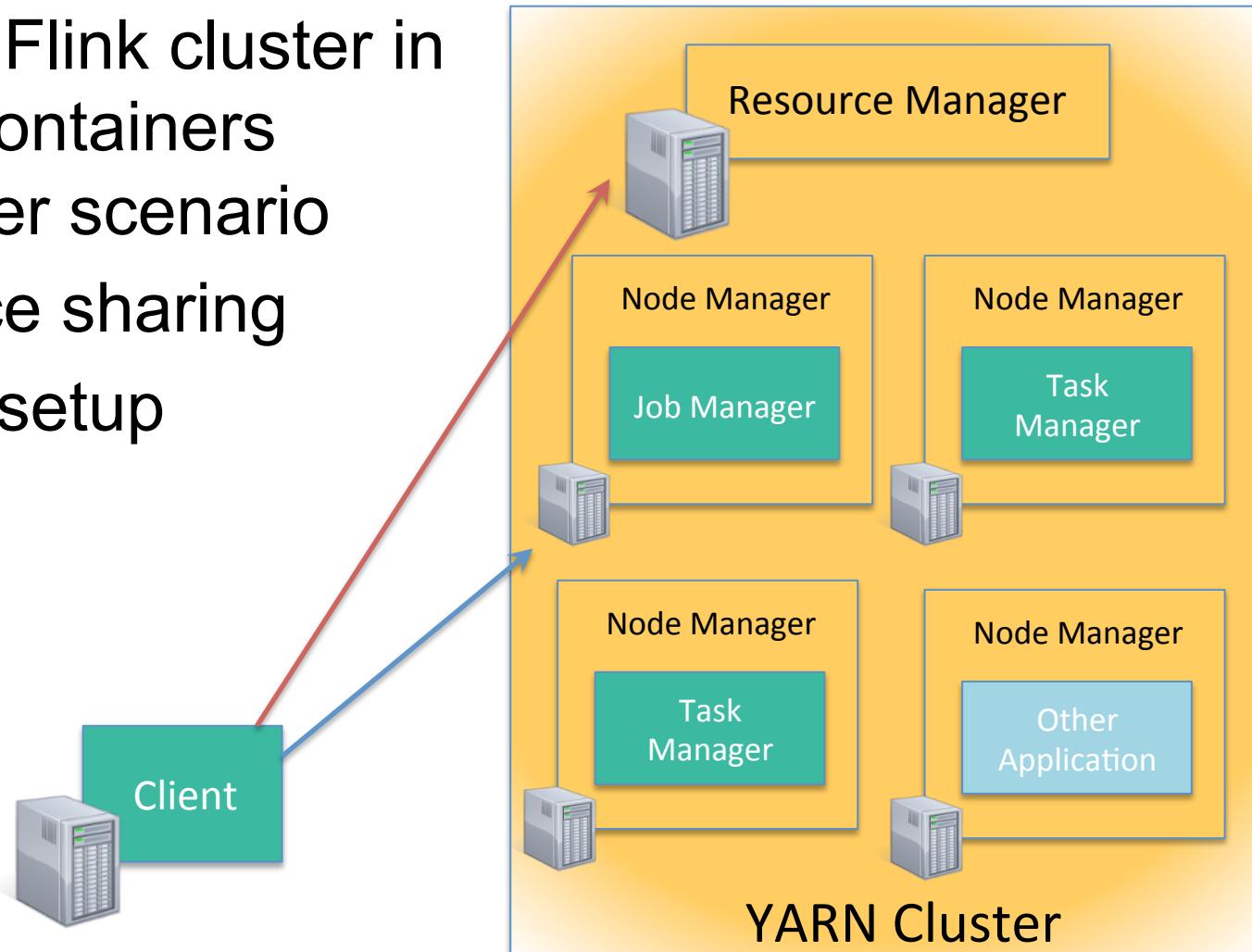
- Brings up a Flink cluster in YARN to run a single job
- Better isolation than session mode





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

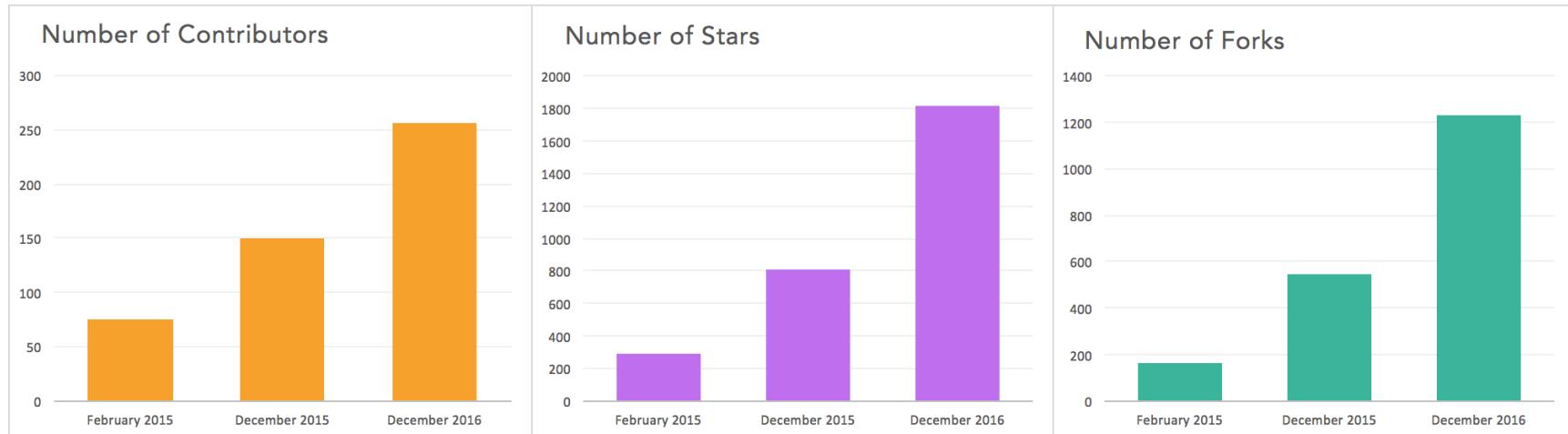


Flink in the real world



Flink community

Github



41 meetups
16,544 members





Flink Forward 2016

dataArtisans

EMC²

Google

Wing



otto group



ResearchGate



cloudera

MAPR

TNG TECHNOLOGY
CONSULTING

SICS



mgm

people pattern



ScaDS DRESDEN LEIPZIG



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 Group

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/powerdby.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