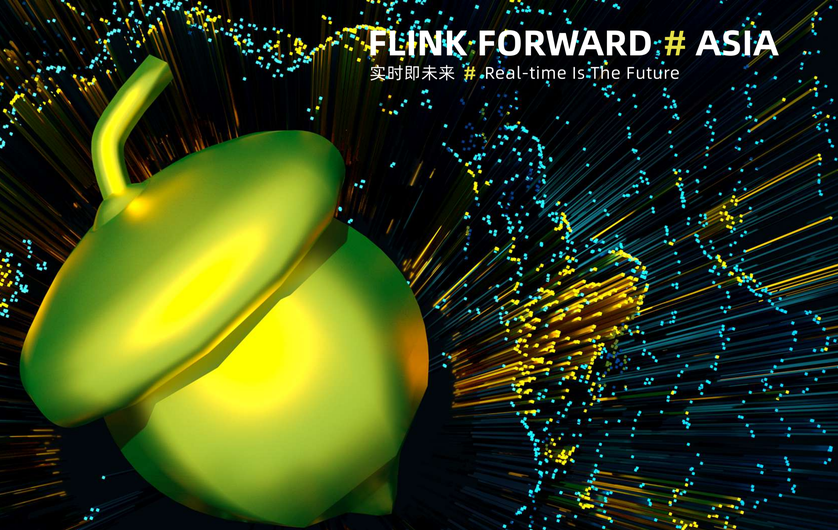
Stateful Functions: Building general-purpose Applications and Services on Apache Flink



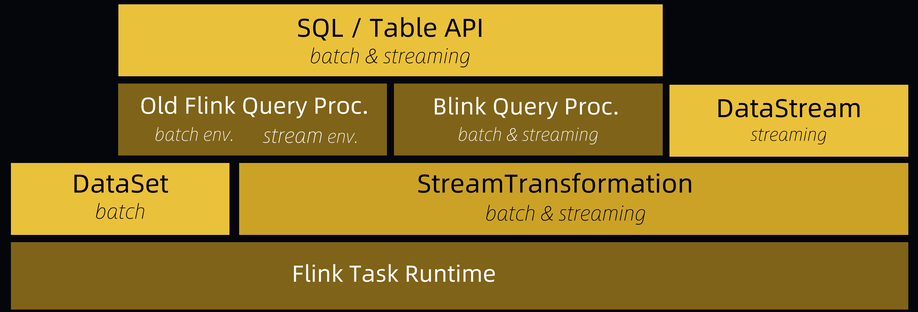
Stephan Ewen

Apache Flink PMC,Ververica Co-founder, CTO

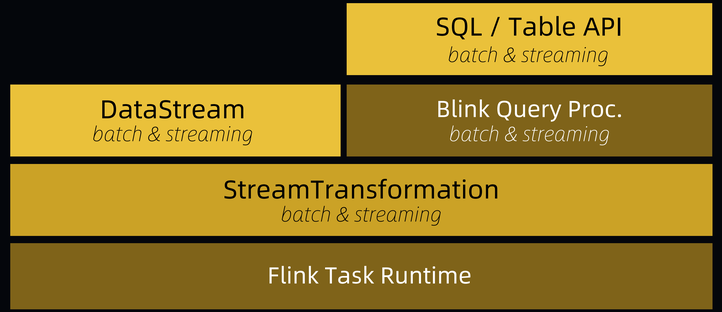
Some Features of 2019

- Catalogs
- New Table APIStructure
- Resource Profile Support
- Unaligned Checkpoints
- Blink Query Engine
- Full Hive compatibility
- Machine Learning Pipelines
- Better Memory Management for Streaming State Backends
- Unified Operator Runtime
- Python Table API
- Fine-grained batch fault tolerance

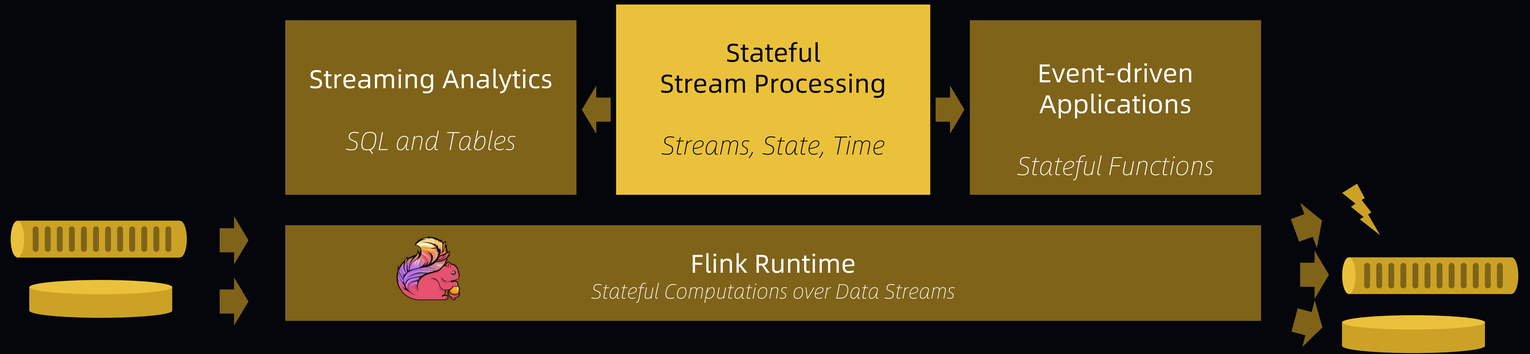
API Stack in Apache Flink 1.9



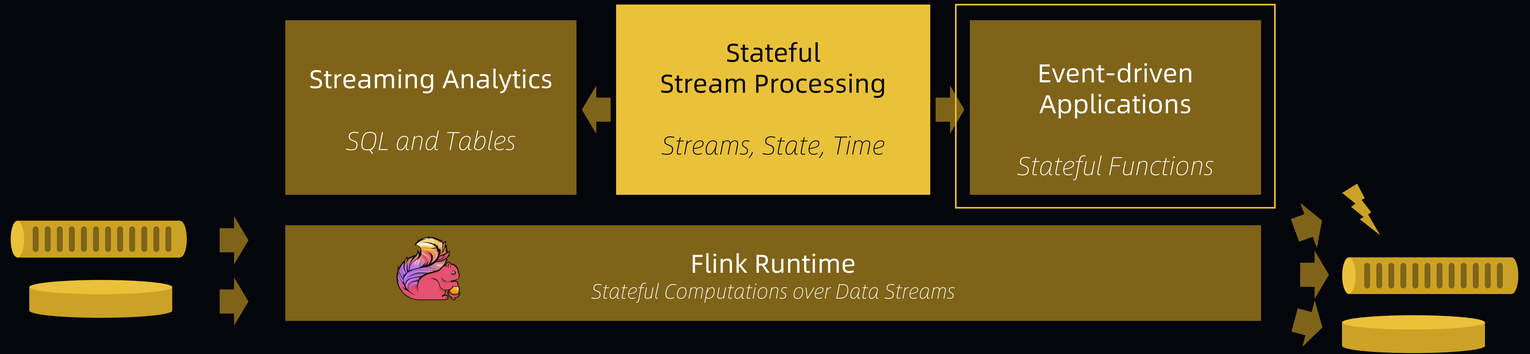
Future API Stack

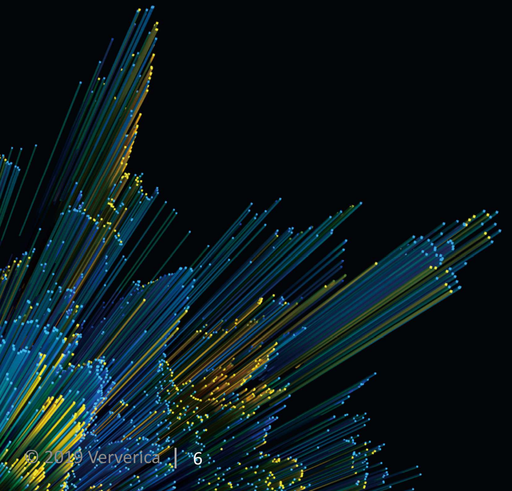


Apache Flink: Analytics and Applications on Streaming Data



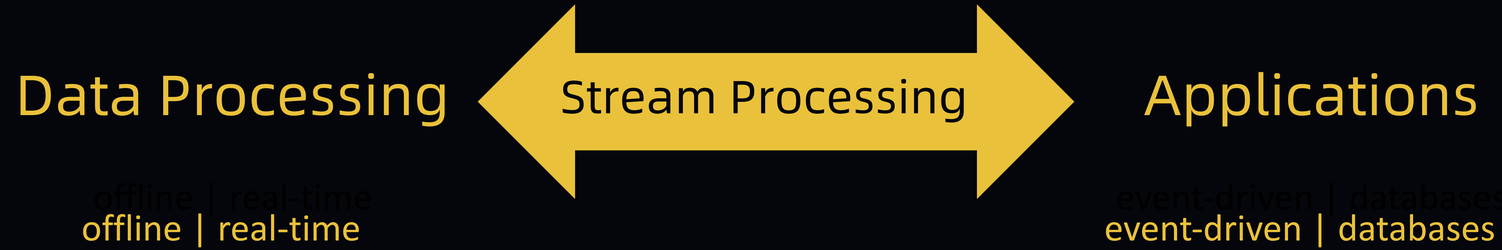
Apache Flink: Analytics and Applications on Streaming Data





Let’s look at building Applications

Stream Processing is at the Intersection of Data Processing and Applications



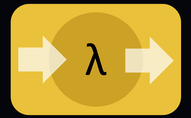
Building an Application Today

Building an Application Today

The big trend:
Serverless

Function as a Service

an event-driven function

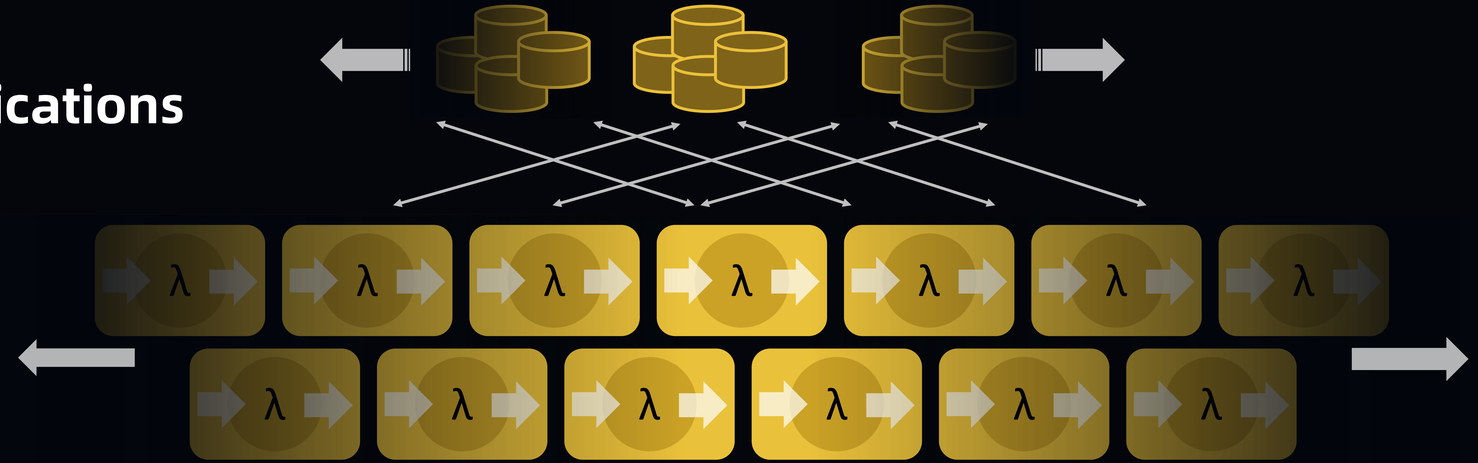


Function as a Service

Function as a Service Handing State in Applications

often bottlenecked by state access & I/O

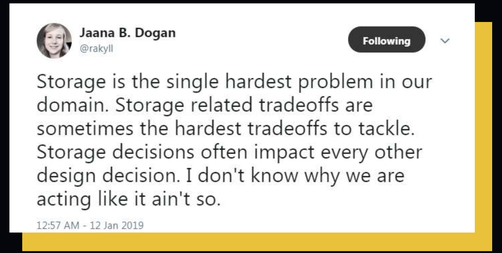
state consistency?

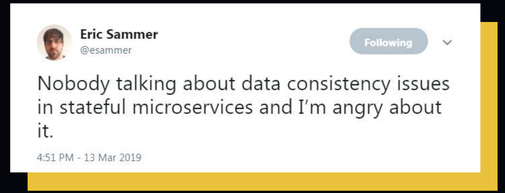


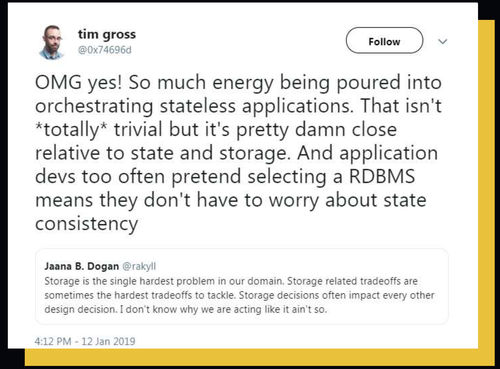
connections, request rates, ...

scaling the database?

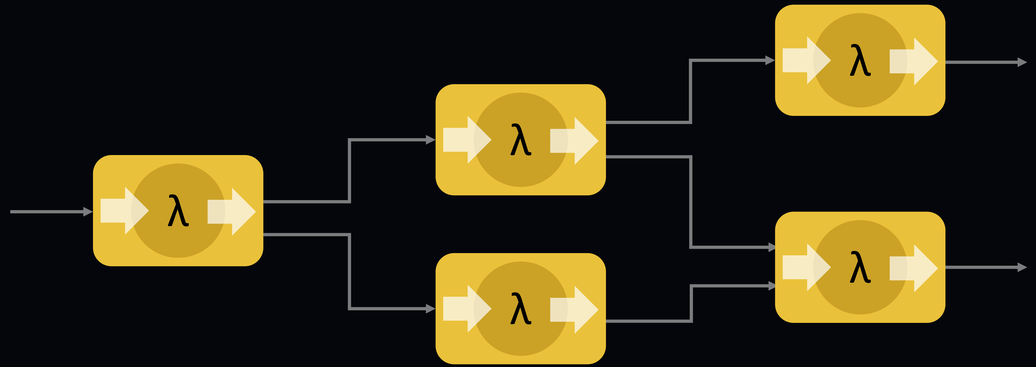
Handling state is a challenge for applications. That remains true in the serverless world.







Composition of Functions



Difficult to build complex applications

Lack of messaging and support for composition

event-driven & composable & state management

...that sounds like...

Stream Processing

Stream Processing

composability

state management

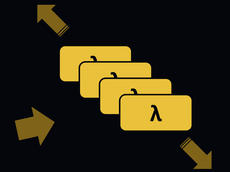
performance

event-driven

Can we combine some of these properties

?

F-a-a-S



simplicity / generality

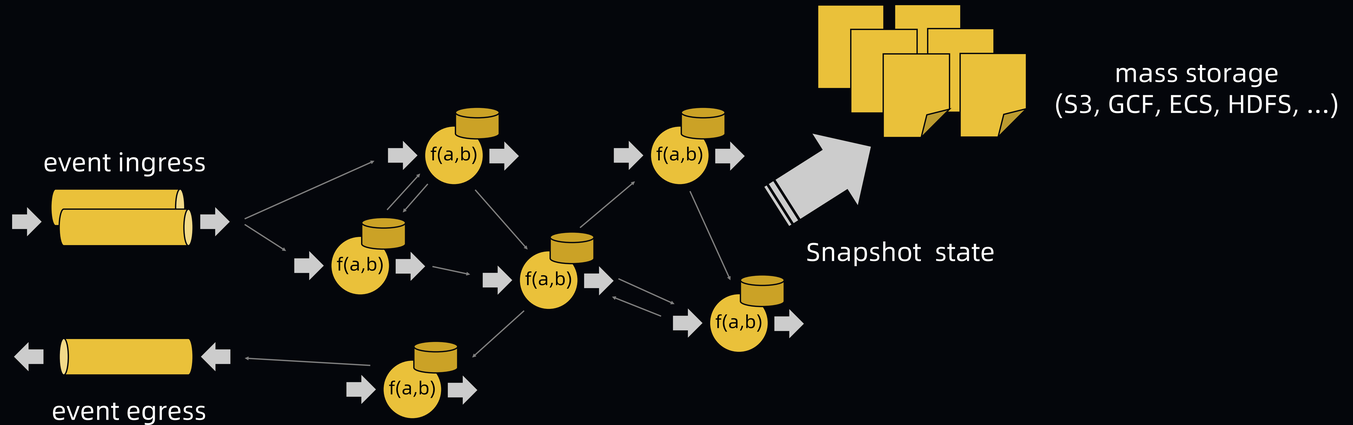
lightweight resources

Stateful Functions

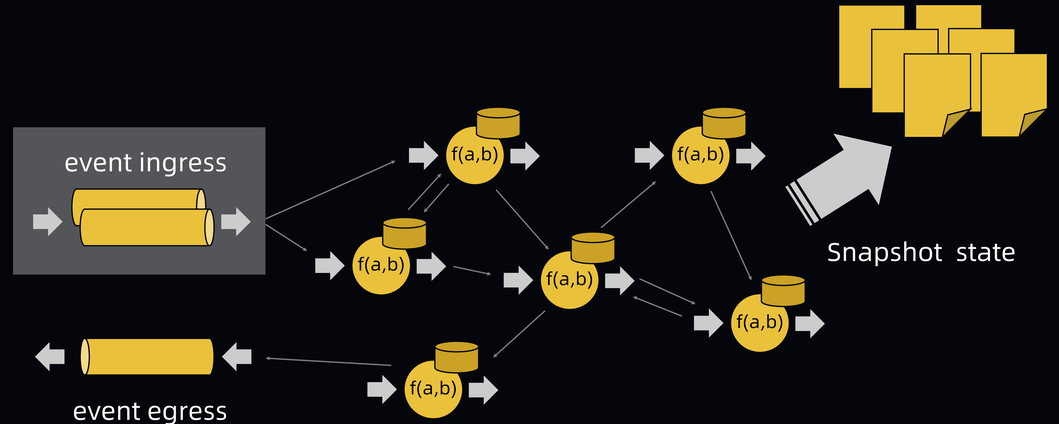
Bringing together ideas from *Stateful Stream Processing* and *FaaS* to create a new way of building Stateful Applications

https://statefun.io/

Stateful Functions

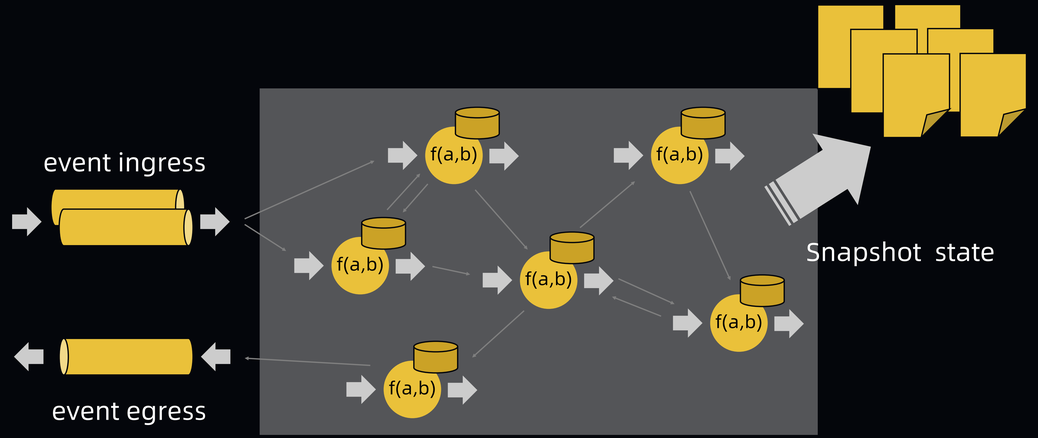


Stateful Functions



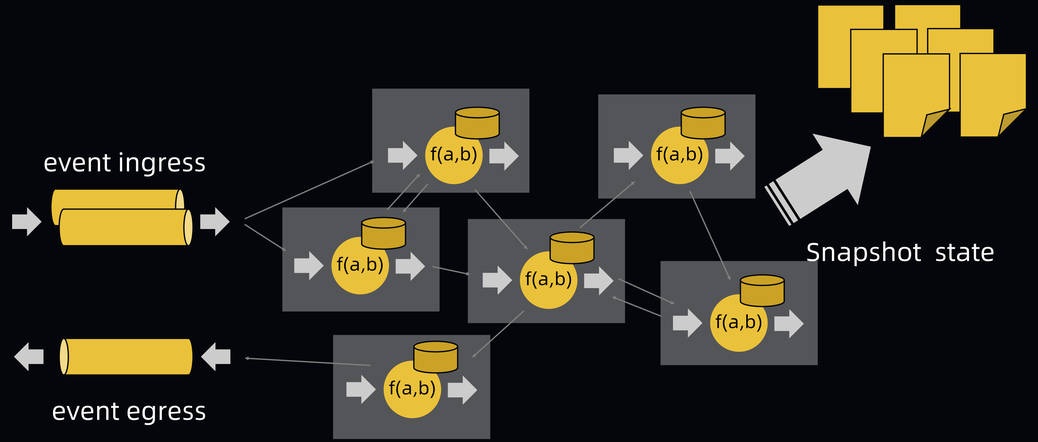
Event ingresses supply events that trigger functions

Stateful Functions



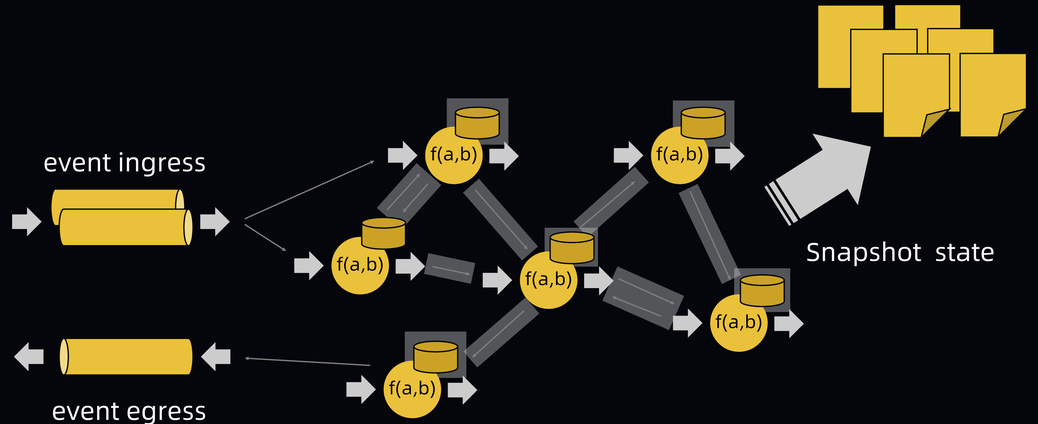
Multiple functions send event to each other
Arbitrary addressing, no restriction to DAG

Stateful Functions



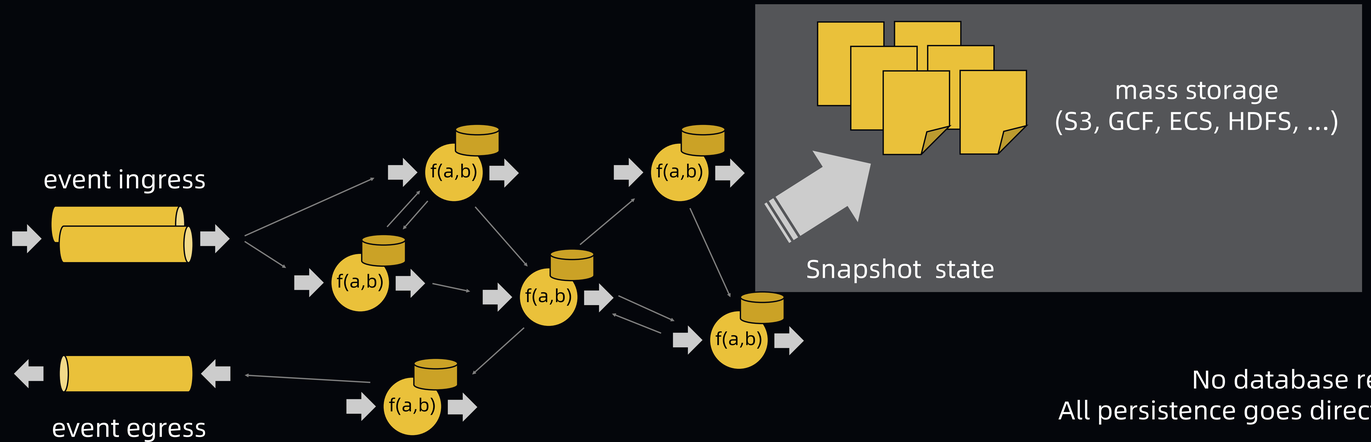
Functions have locally embedded state

Stateful Functions



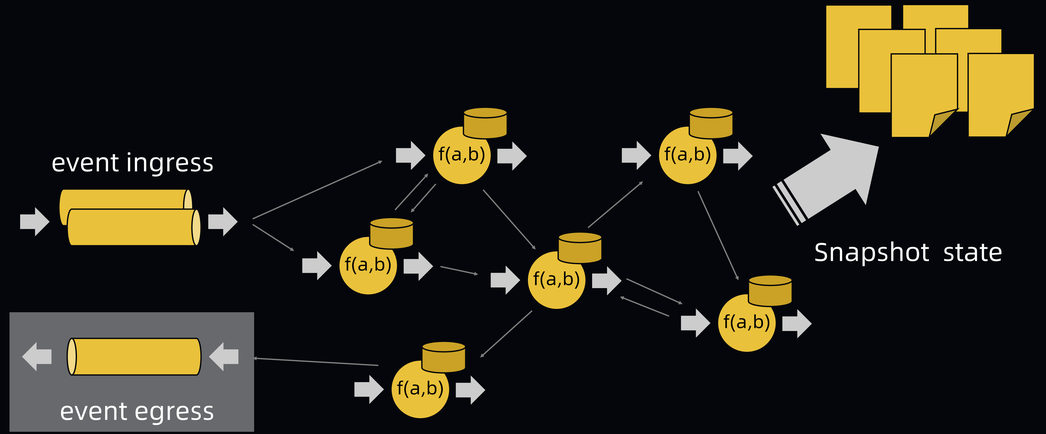
State and messaging are consistent
with exactly-once semantics

Stateful Functions



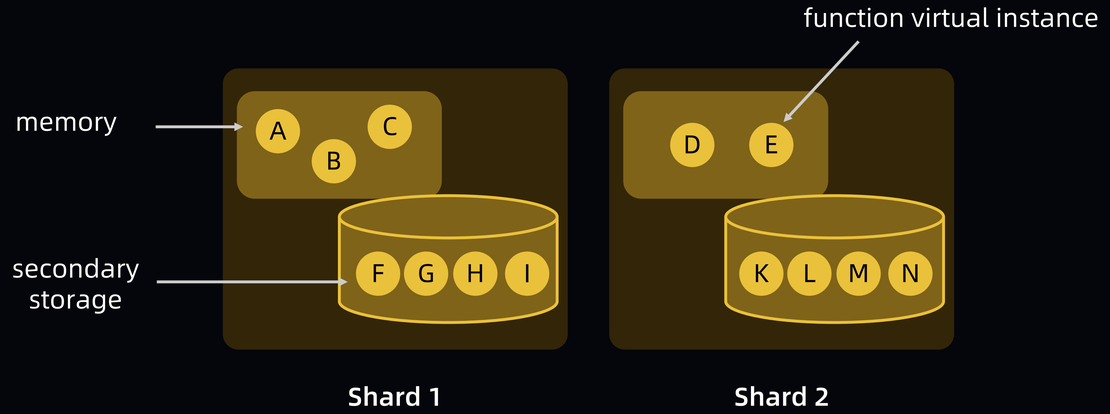
No database required
All persistence goes directly to blob storage

Stateful Functions

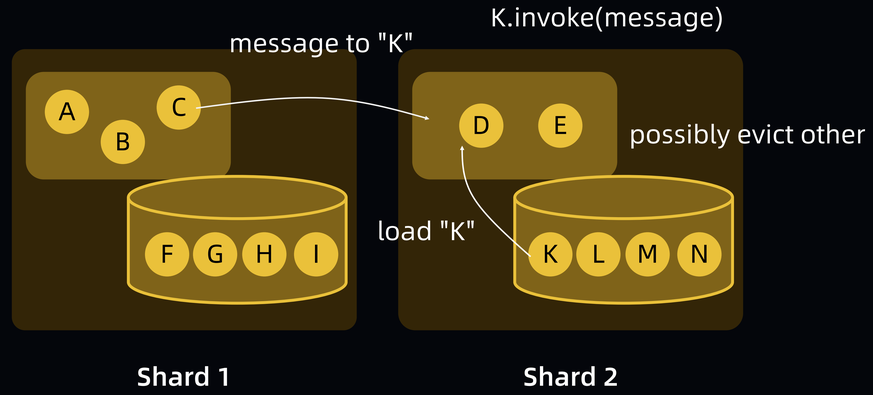


Event egresses to respond via event streams

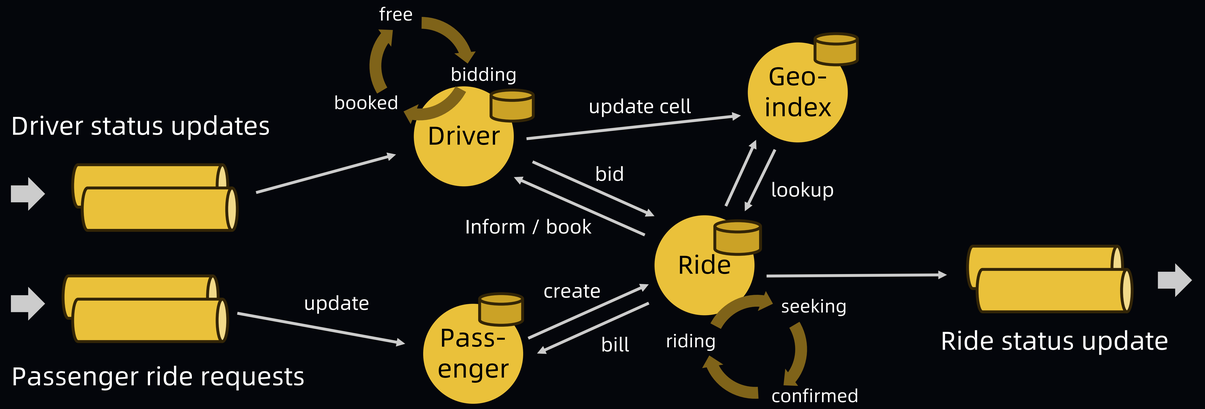
Logical/Virtual Instances



Logical/Virtual Instances

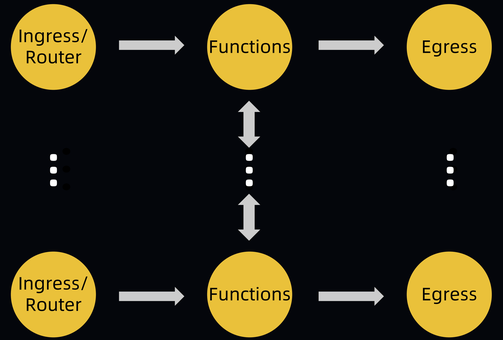


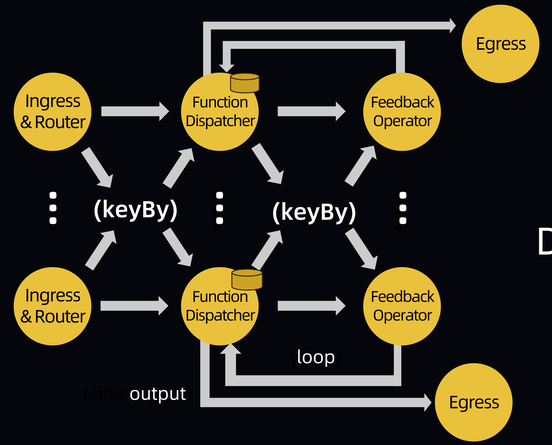
Example: Ride Sharing App



Apache Flink is the State and Event Streaming Fabric

Conceptual Dataflow

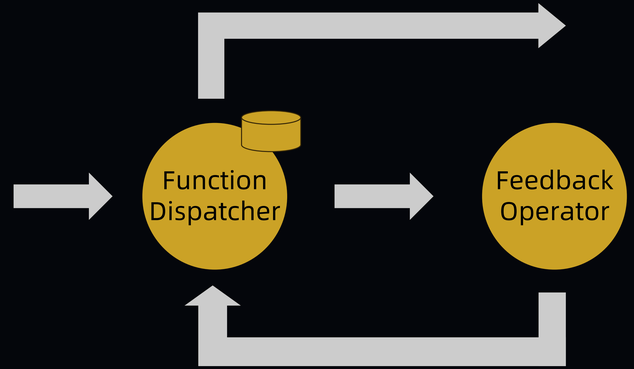




Apache Flink Dataflow Graph

Running Stateful Functions on Apache Flink

Exactly-once checkpointing for streaming loops



Stream Processing
Streaming SQL

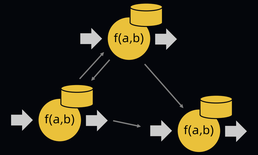
event/stream-centric

data preparation,
combining knowledge/information

filtering, enriching,
aggregating, joining events

Stateful Functions

state-centric

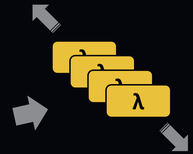


coordination,
(interacting) state machines

complex event/state
interactions

F-a-a-S

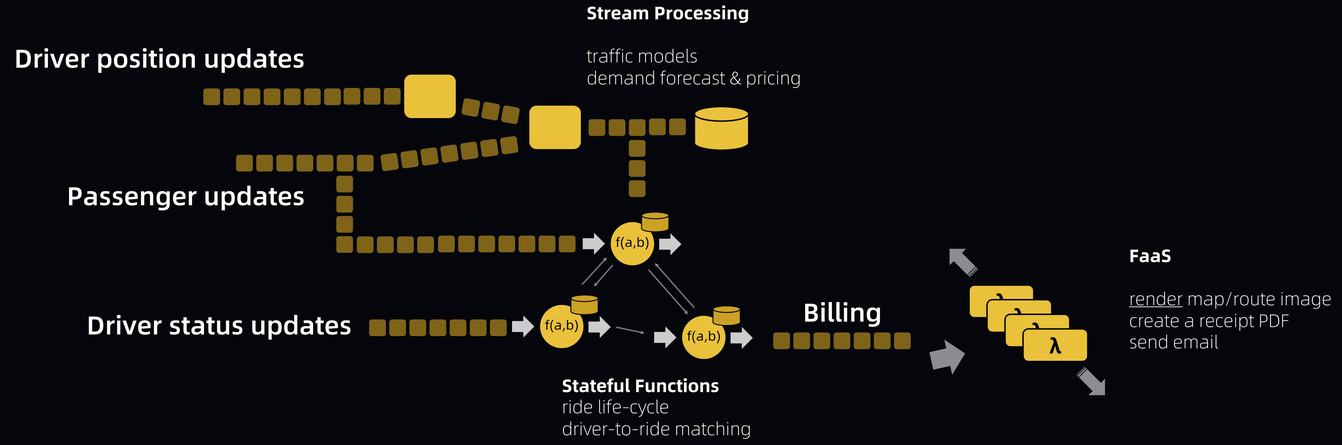
stateless / compute-centric



“occasional” actions or spiky loads

compute-intensive
or blocking

Putting it all together: Ridesharing again



Currently being contributed to Apache Flink

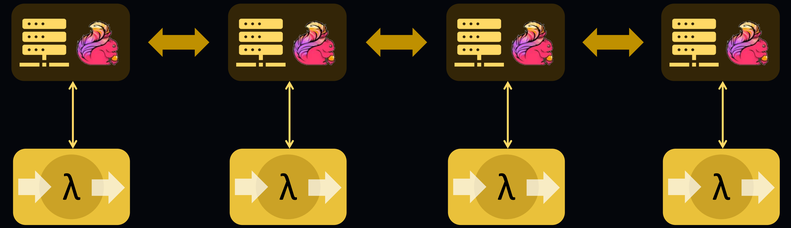
Fully Open Source
on Ververica's GitHub
under ASL 2

Propose contribution
for Apache Flink

Community discussion
about project proposal

Upon acceptance,
handover to the
Flink project

The Next Steps



Stateful Multi-language Functions

Join the project!

Try it out, let us know how it works, file a bug, open a PR…

Code

https://github.com/ververica/stateful-functions

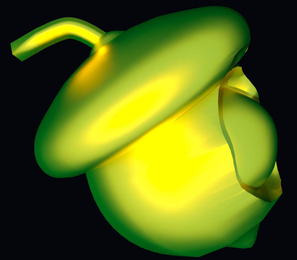
Website

https://statefun.io/

Twitter

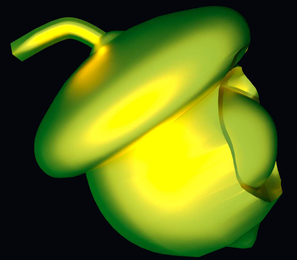
@statefun_io @ApacheFlink

@StephanEwen

Flink Forward Asia  2019

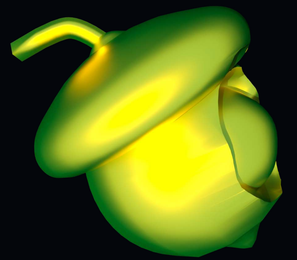
Thanks!

Thanks!

Flink Forward Asia  2019

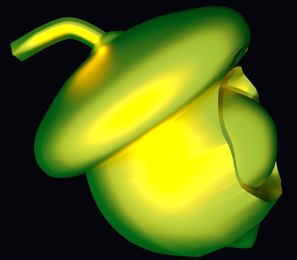
Thanks!

Thanks!

Flink Forward Asia  2019

Thanks!

Thanks!

Flink Forward Asia  2019

Thanks!

Thanks!