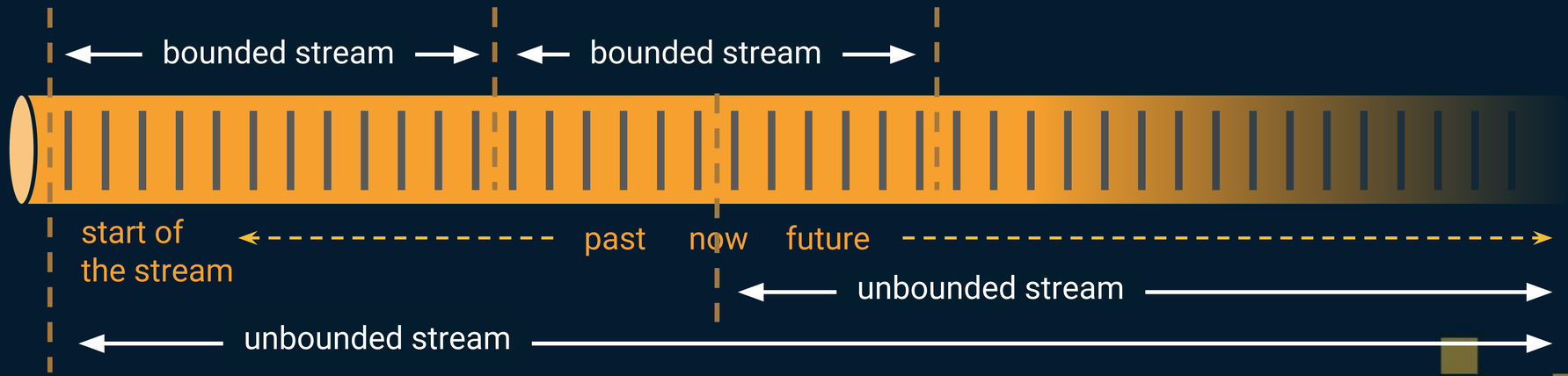
Stream Processing takes on Everything

Stephan Ewen

Flink Forward China 2018

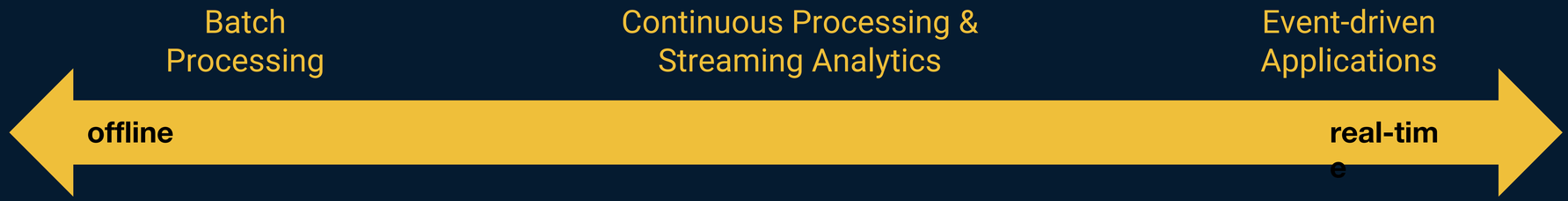
FLINK FORWARD

Streams as a Paradigm



FLINK FORWARD

Stream Processing takes on Everything



Some new Streaming Features in Flink 1.7

```
SELECT
    o.time AS time,
    o.price * r.rate AS price
FROM
    Orders AS o,
    LATERAL TABLE (Rates(o.time)) AS r
WHERE r.crcy = o.crcy
```

Orders

time	price	crcy
10:15	2	EUR
10:30	1	USD
10:32	50	YEN
10:52	3	EUR
11:04	5	USD

Rates

time	crcy	rate
09:00	USD	102
09:00	EUR	114
09:00	YEN	1
10:45	EUR	116
11:00	USD	105

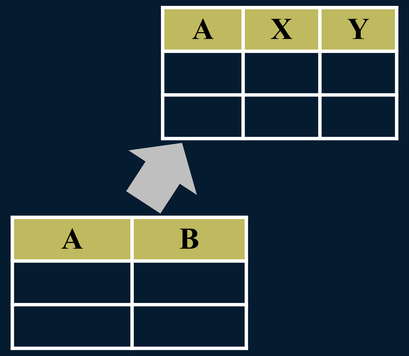
Result

time	price
10:15	228
10:30	102
10:32	50
10:52	348
11:04	525

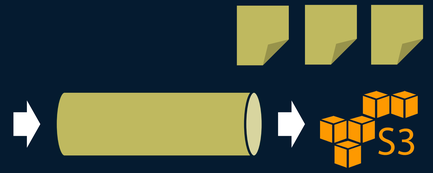
Time-versioned Joins

```
SELECT *
FROM TaxiRides
MATCH_RECOGNIZE (
   PARTITION BY driverId
   ORDER BY rideTime
   MEASURES
       S.rideId as sRideId
   AFTER MATCH SKIP PAST LAST ROW
   PATTERN (S M{2,} E)
   DEFINE
       S AS S.isStart = true,
       M AS M.rideId <> S.rideId,
       E AS E.isStart = false
           AND E.rideId = S.rideId)
```

MATCH_RECOGNIZE



Schema Upgrades

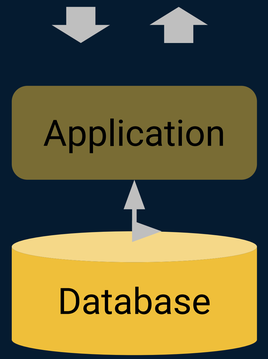


S3 exactly-once

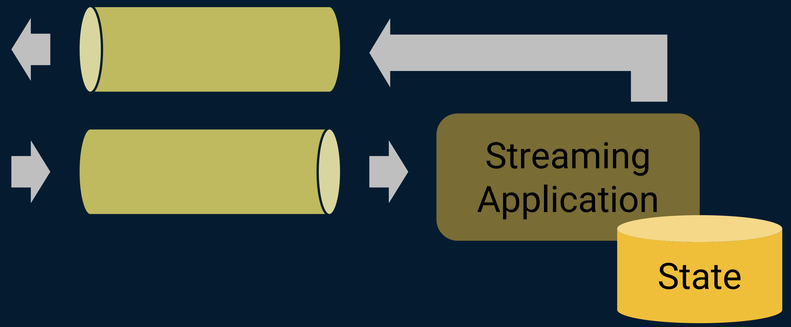
Applications and Services

FLINK FORWARD

Request/Response Applications



Event-sourced / Streaming Applications

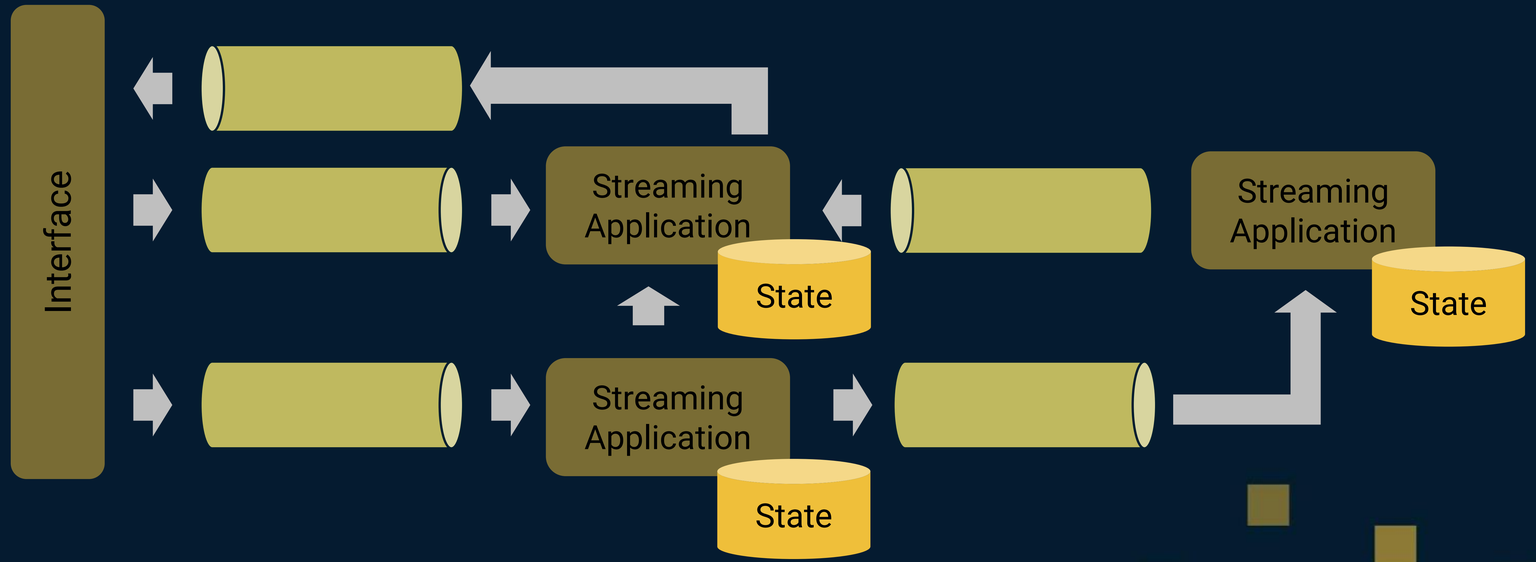


FLINK FORWARD

Streaming Applications

Event Sourcing

Command Query
Responsibility Segregation



What about Transactional Applications?

Example: Accounts and Transfers

A > 100?
A -= 50
B += 50

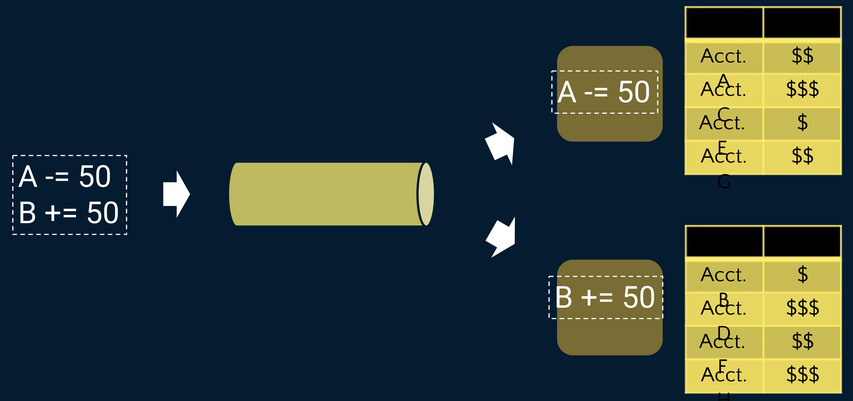
Acct. A	$$
Acct. C	$$$
Acct. E	$
Acct. G	$$

ACID

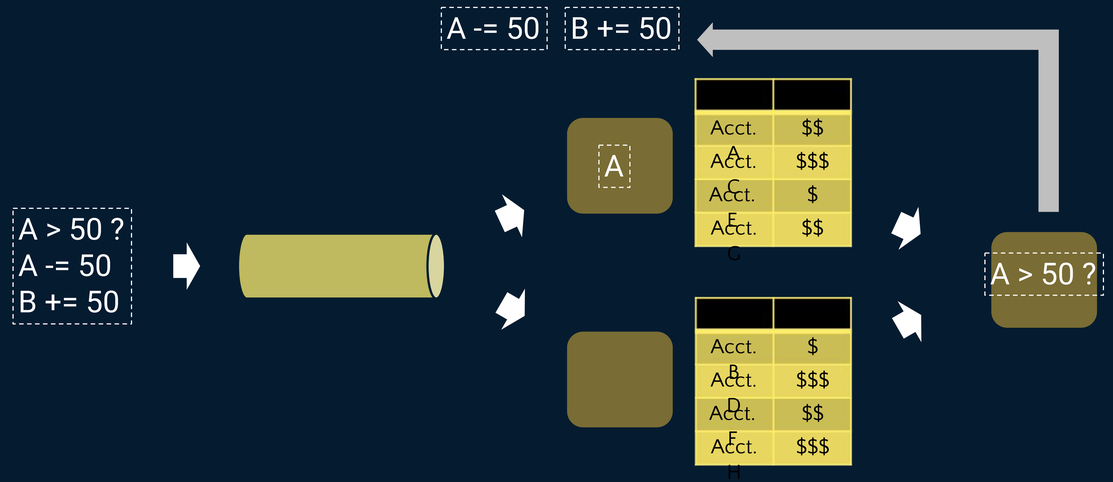
Atomicity **Consistency**

Isolation **Durability**

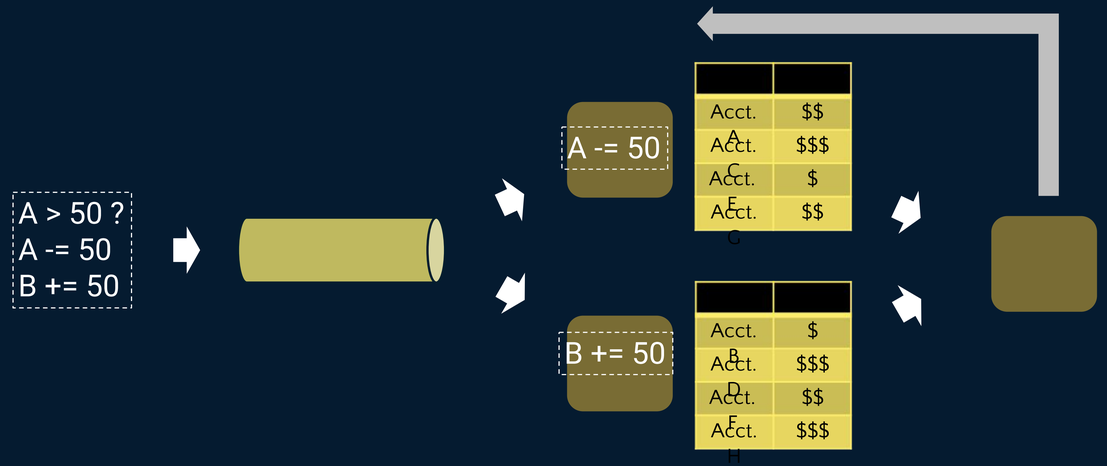
Streaming Transaction Processing



Streaming Transaction Processing

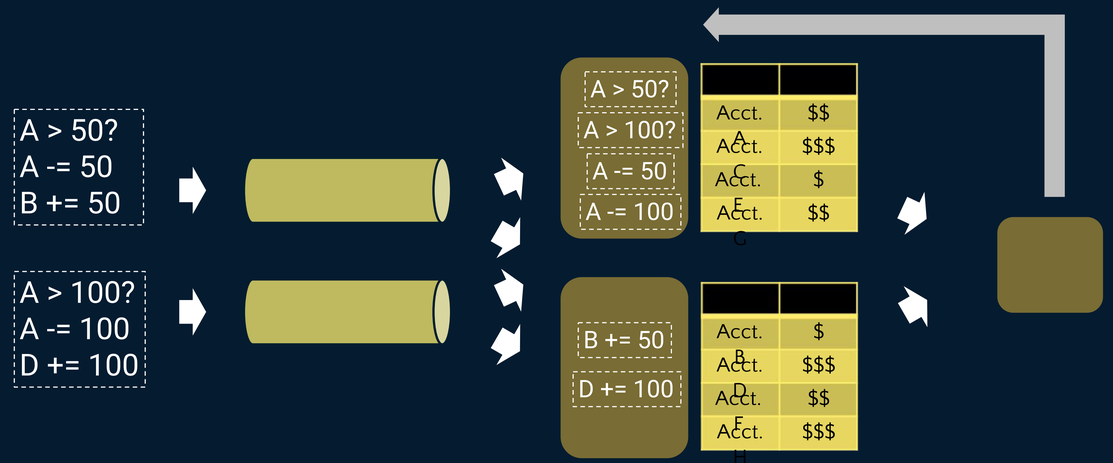


Streaming Transaction Processing



FLINK FORWARD

Streaming Transaction Processing



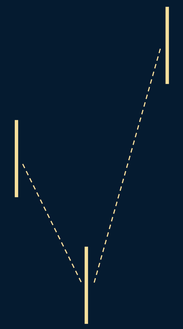
How to resolve this consistently?

Event-time to the rescue!

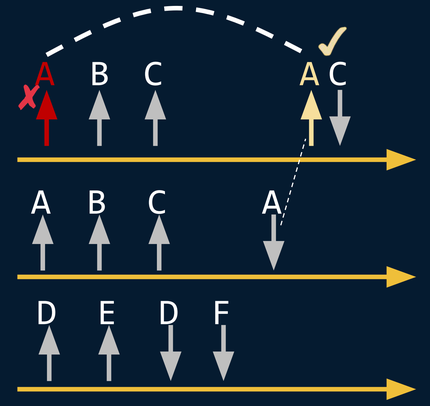
Serializable Isolation in Streaming Transactions

```
Txn: (A,B,C)
-> C

Txn: (A,B,C)
-> A

Txn: (D,E) ->
(D,F)
```

transaction events



define ordering for schedule



reorder events

Serializable Isolation in Streaming Transactions

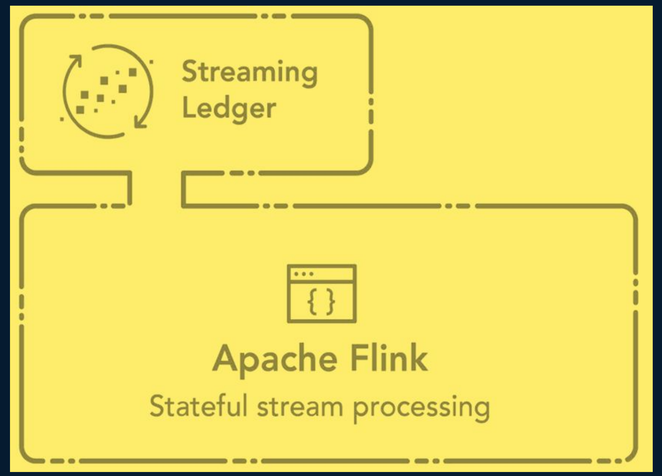
(0xf876ab78|0x0b7cc7a3)

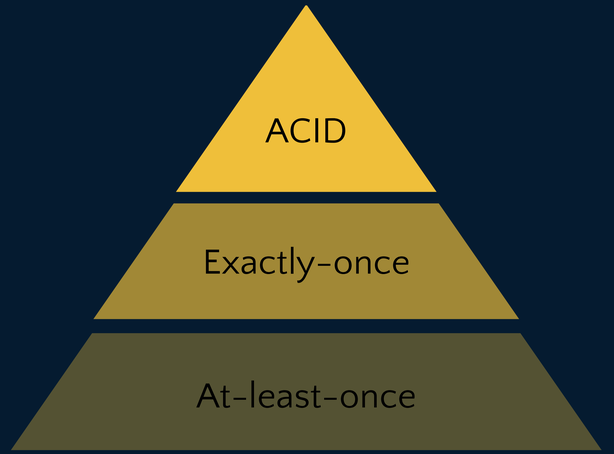
Logical Clocks
to define schedule

event re-ordering &
out-of-order processing

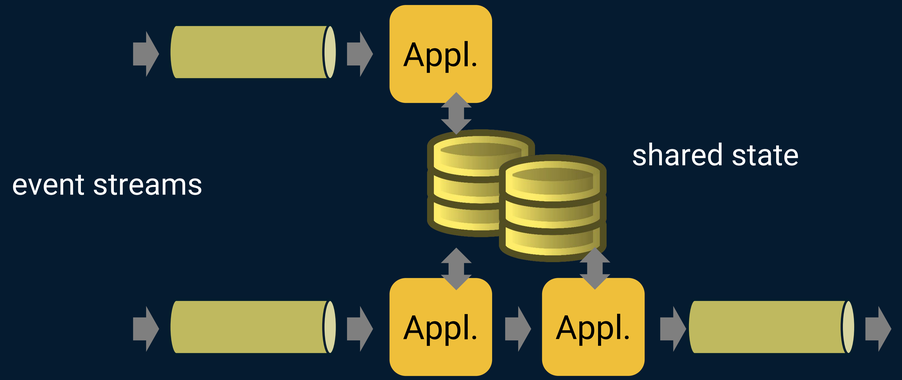
Iterative streaming
dataflows

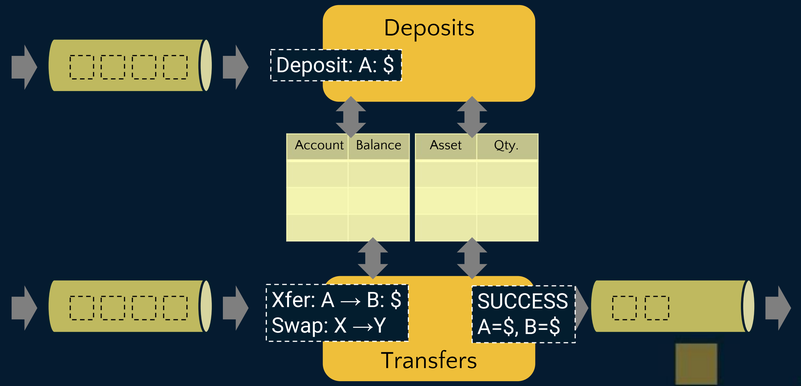
dA Streaming Ledger





Applications against Shared Consistent State

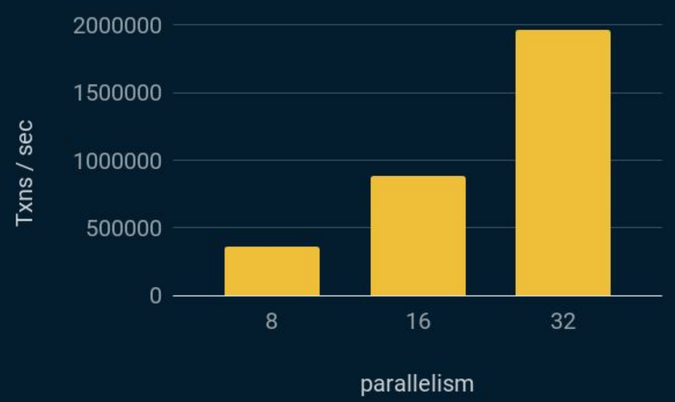




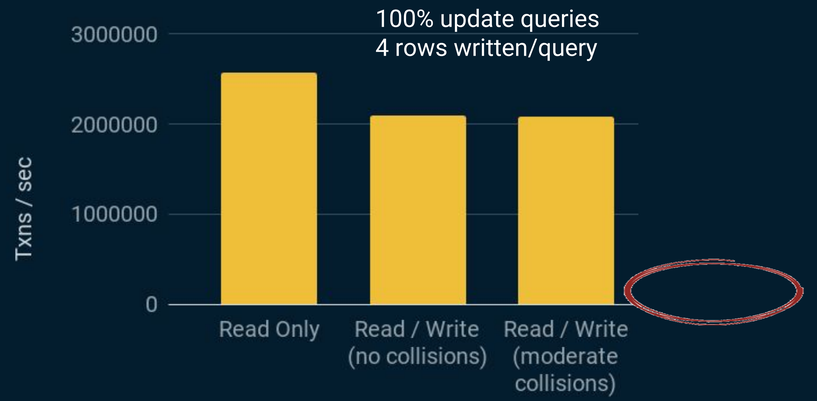
Performance

200 million rows
100% update queries
4 rows written/query

Scale Out



Hot Keys

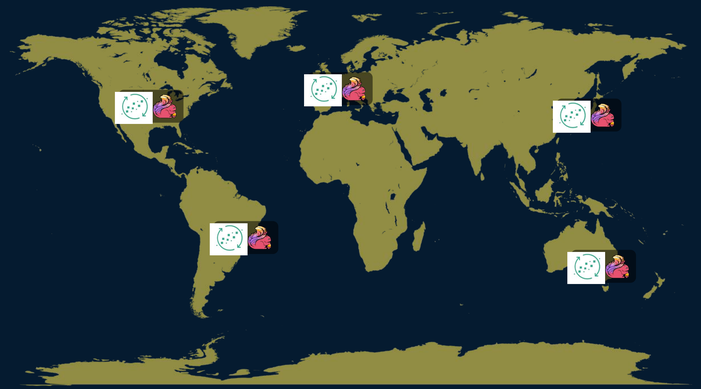


Extreme contention:

800,000 updates/sec on 1,000 keys

Slowdown, but stable

Performance in a Geo-distributed Setup



Apache Flink: The Powerful Foundation

This technology is possible, because Apache Flink offers such powerful building blocks

- Continuous processing
- Iterative flows
- Flexible state abstraction
- Asynchronous checkpoints
- Sophisticated event-time/watermarks

Stream Processing takes on Everything

FLINK FORWARD

Stream Processing takes on Everything

Batch Processing ✔

Continuous Processing & Streaming Analytics ✔

Event-driven Applications ✔

Transacional Processing ✔

analytical ⟷ **operational**

...with the right framework ;-)


Apache Flink
THANKS