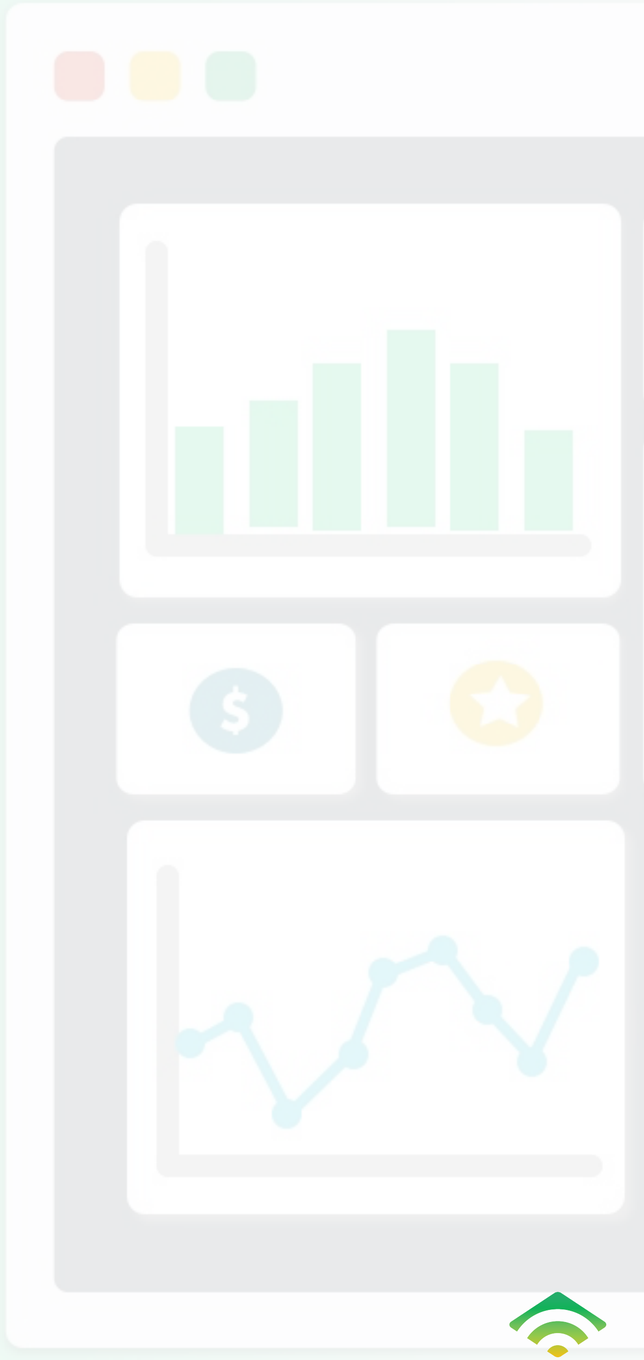
High Cardinality Data Stream Processing with Large States

Ning Shi, Klaviyo

Flink Forward San Francisco, 2019

Help ecommerce businesses grow

- Real-time analytics on consumer events
- Event-triggered actions based on analytics
- Best-in-class email marketing



Use Case

- Thousands of types of events
- Close to 100,000 events per second
- *“How many unique individuals opened emails from this campaign between 3:00am-4:00am?”*
- *“How many unique individuals purchased at least two black iPhone chargers after reading this campaign email yesterday?”*

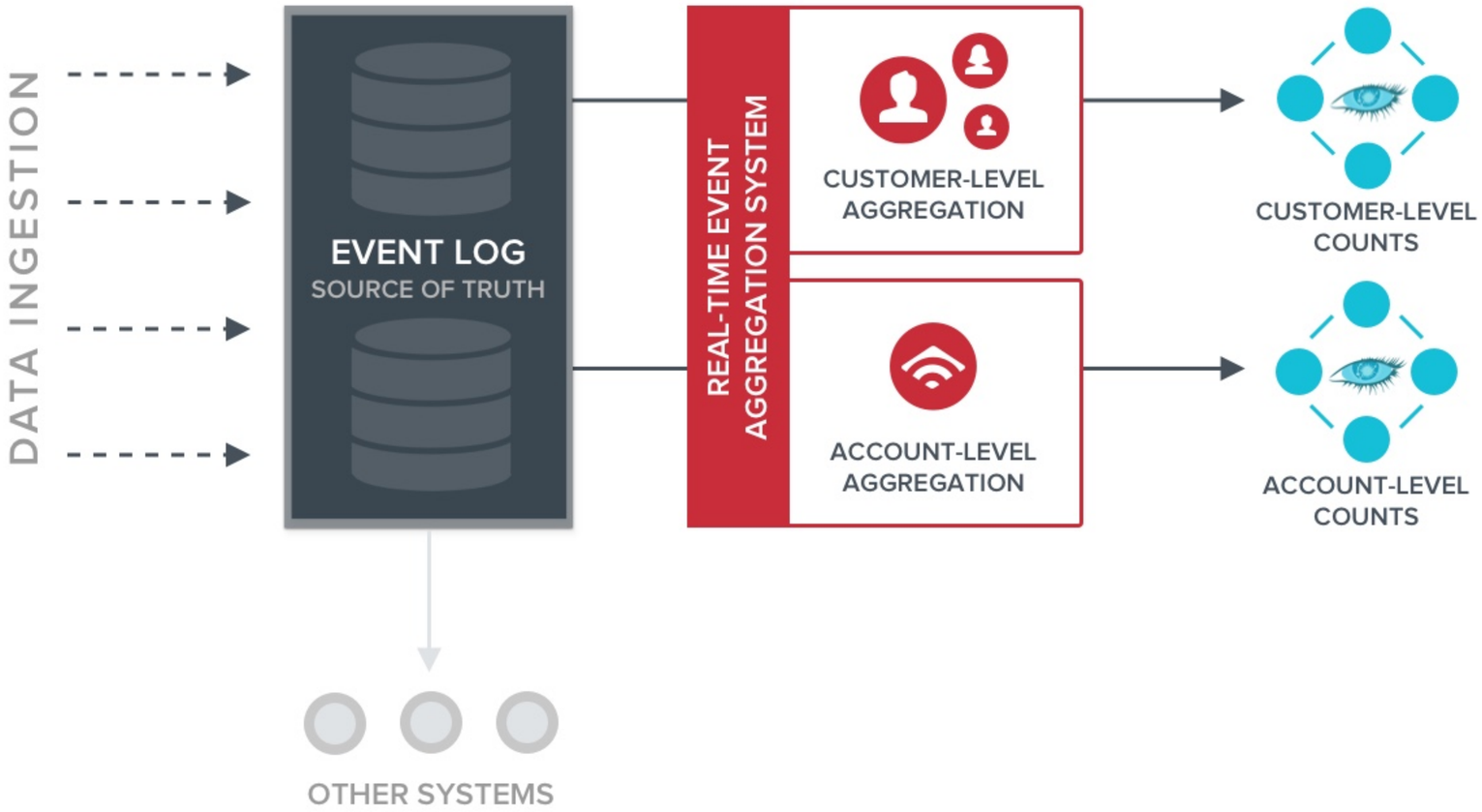
```
{
  “email”:       “john@example.com”,
  “message_id”: “ABCDE”,
  “timestamp”:  1544153562,
  “ip”:          “127.0.0.1”,
  “browser”:    “Safari 12.0.1”
}
```

Dimension Name	Dimension Value
provider	example.com
message_id	ABCDE
campaign	Holiday Sale!
.....	
browser	Safari 12.0.1

Technical Challenges

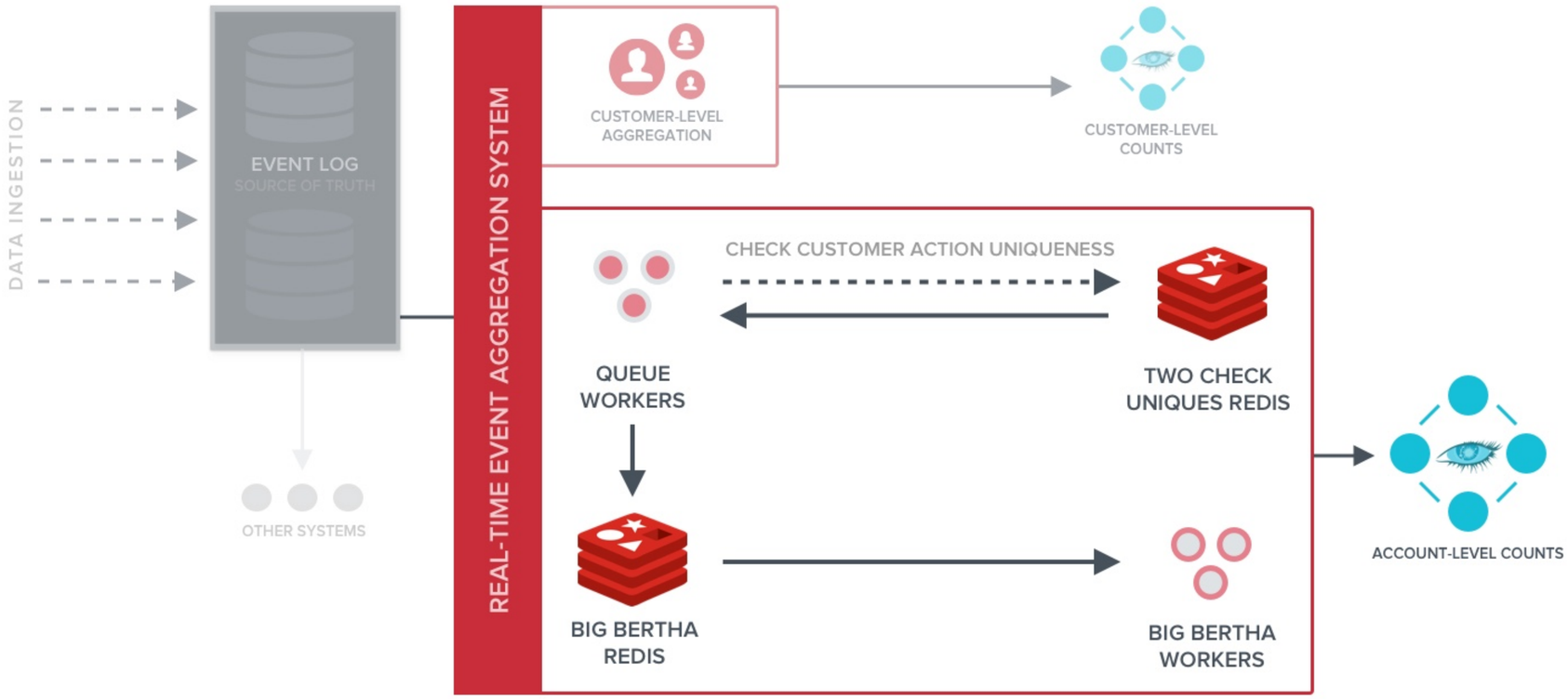
- Over 1 billion user profiles
- Large state (1.5TB compressed) for duplicate detection
- High fan-out ratio (one to hundreds)
- Millions of metrics to aggregate per second

Event Processing Pipeline



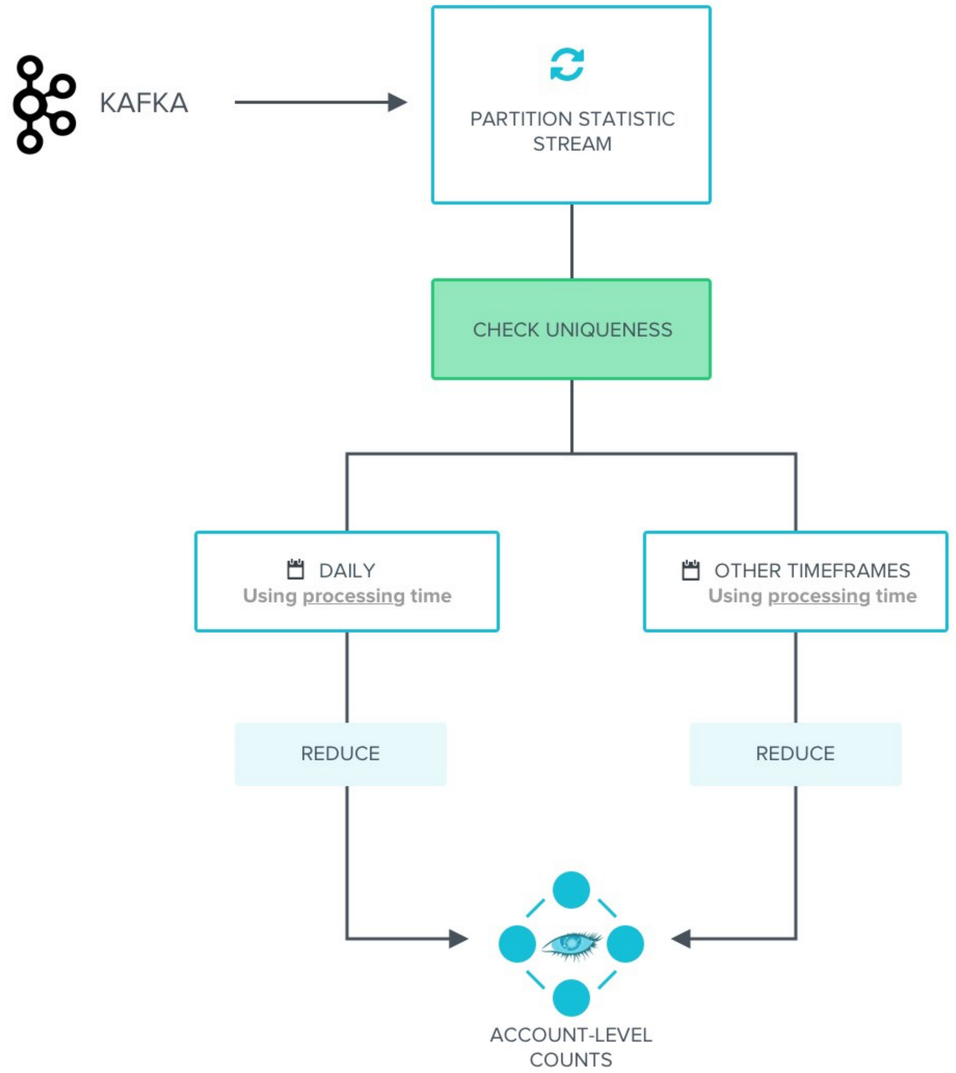
Initial Version

- Custom Python code
- Nondeterministic
- Too many moving parts



Abacus

- Apache Flink
- Exactly-once processing
- Easy to scale
- Significantly better performance (90% reduction of EC2 instances)



Lessons Learned

- Code changes
- Configuration

Code Changes

Time

Processing Time

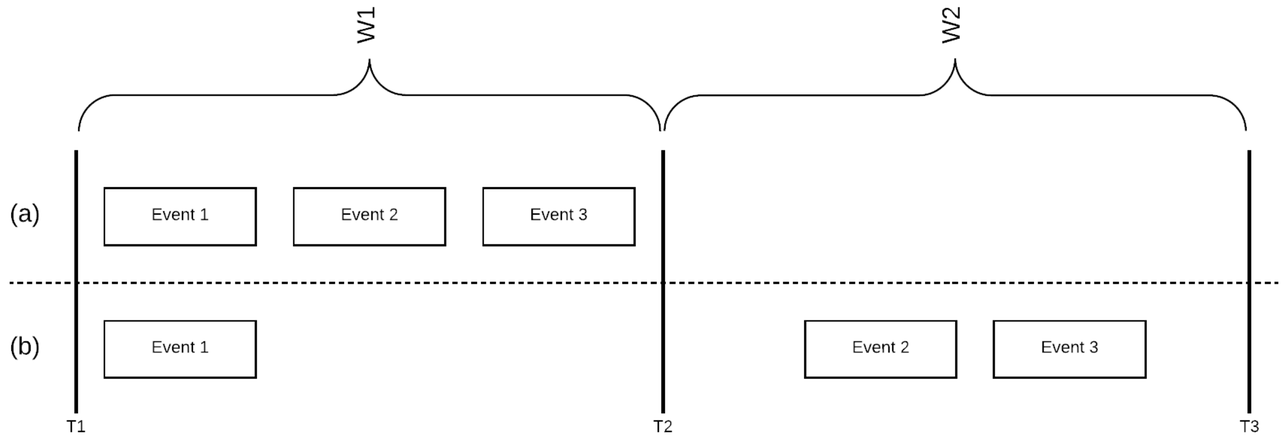
- Wall clock time on each TaskManager
- Nondeterministic
- Different on each TaskManager

Event Time

- Intrinsic to events
- Deterministic
- Synchronized across cluster

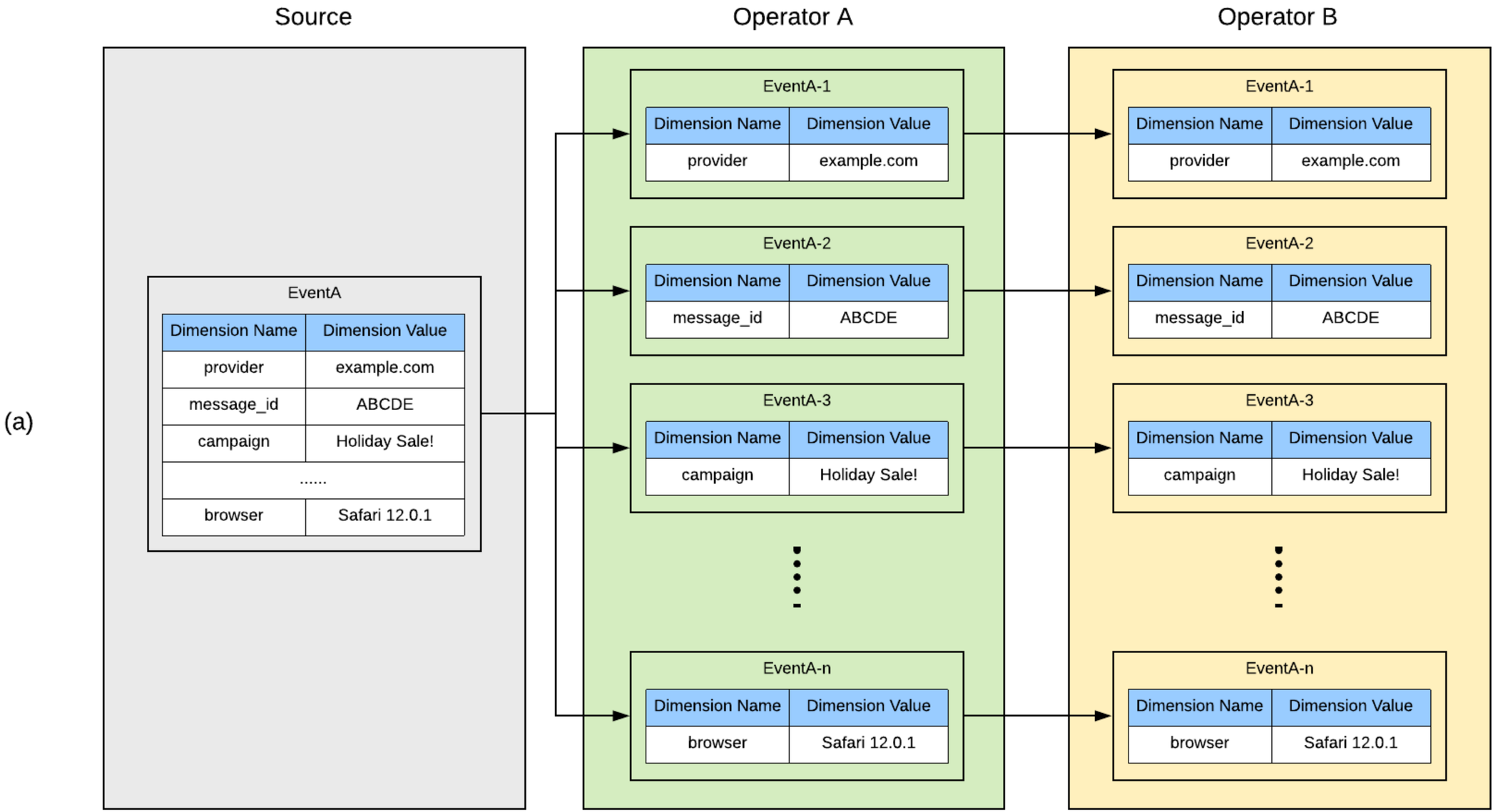
Processing Time

- Many unsynchronized external event sources (off by hours)
- Historical event synchronization (off by years)
- Handle all events in a single job with processing time



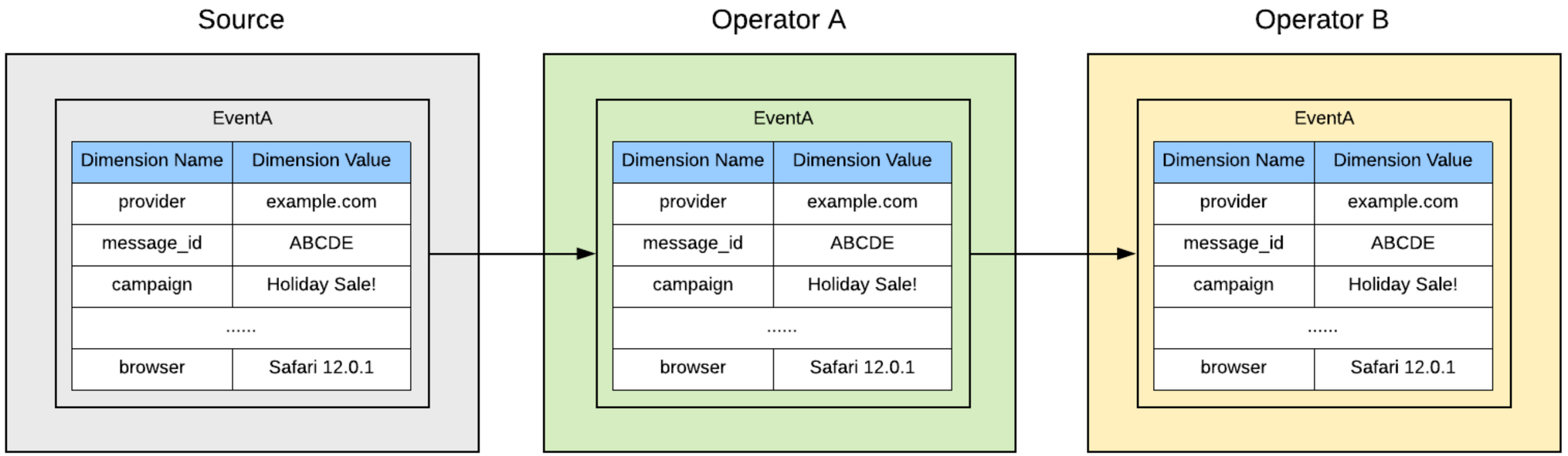
Flink Backward

Fan-out Approach (a)



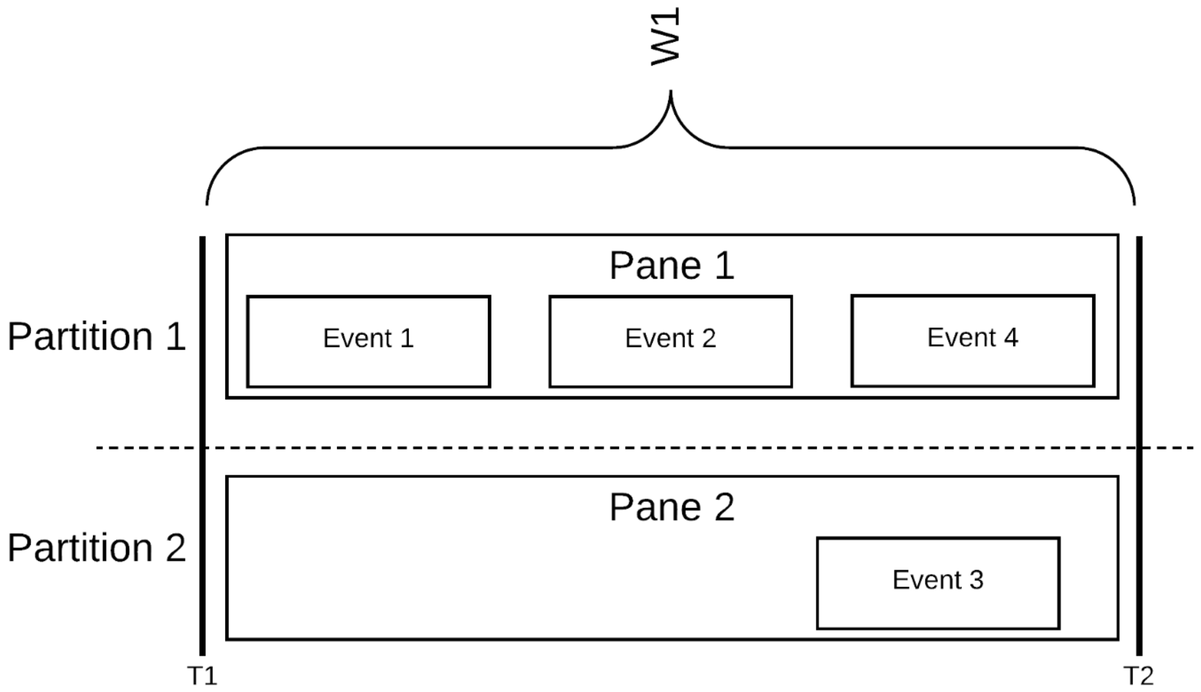
Fan-out Approach (b)

(b)



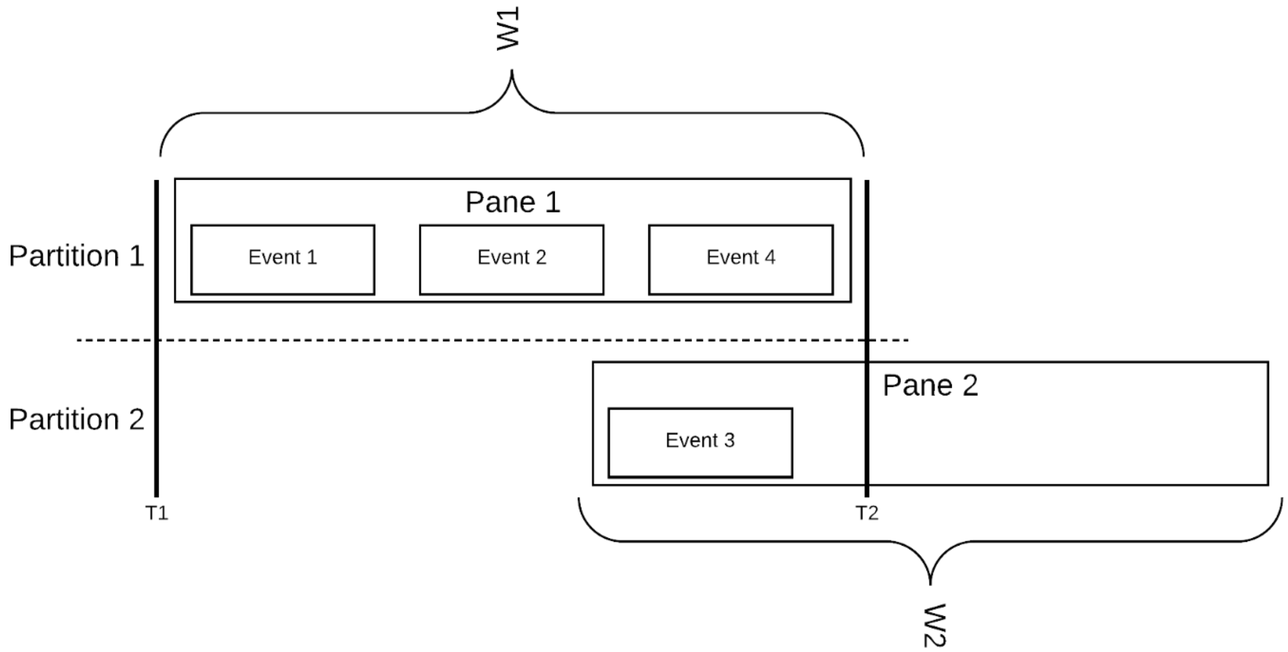
Windowing

- A window on a partition is called a pane
- Tumbling and sliding window panes fire simultaneously
- Triggers live on Java heap prior to Flink 1.6, or in RocksDB after Flink 1.6
- Millions of triggers have huge impact



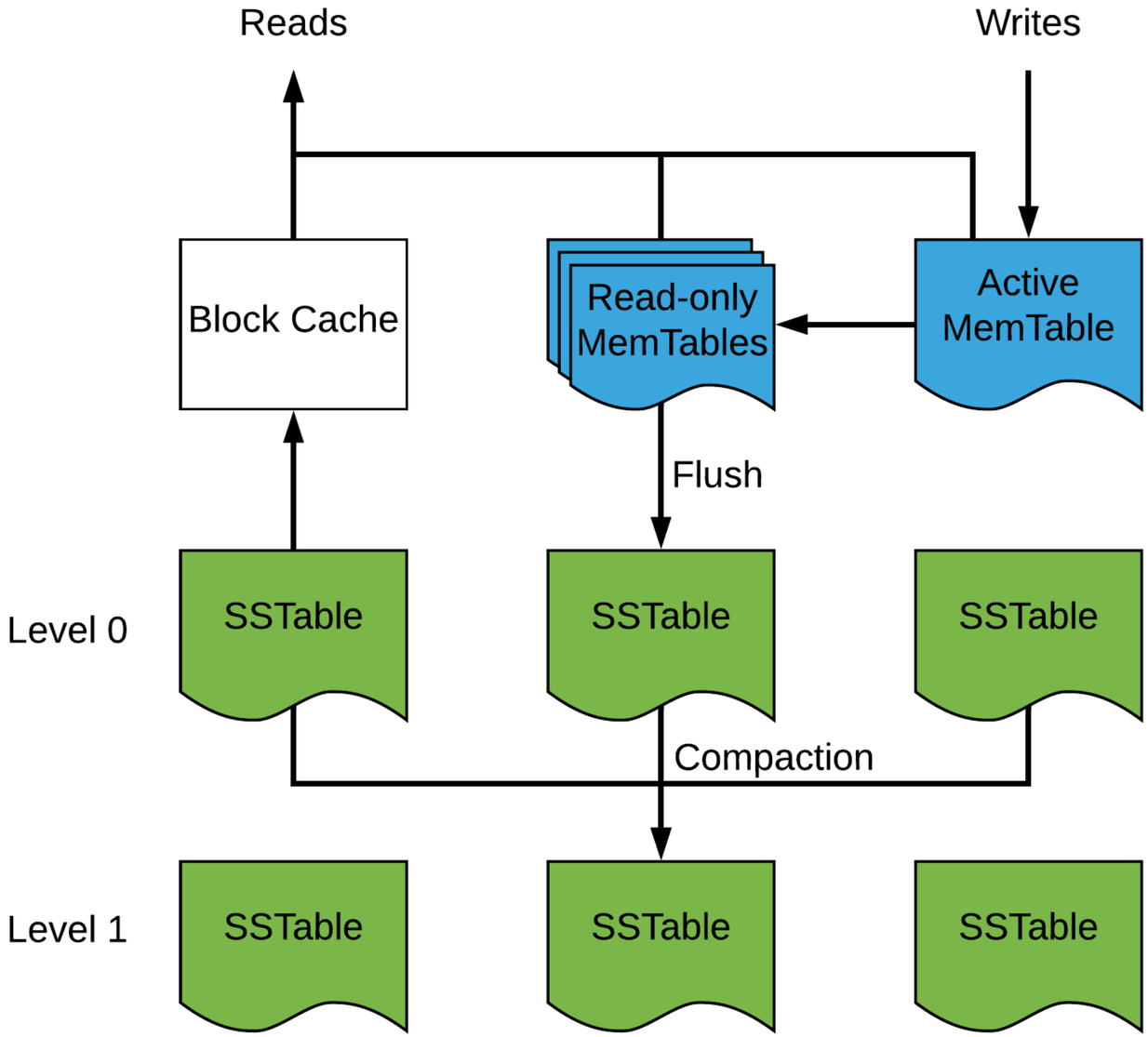
Windowing

- Pane opens only when there is event on partition
- Pane closes after fixed amount of time after earliest event in the pane
- Staggered trigger firings



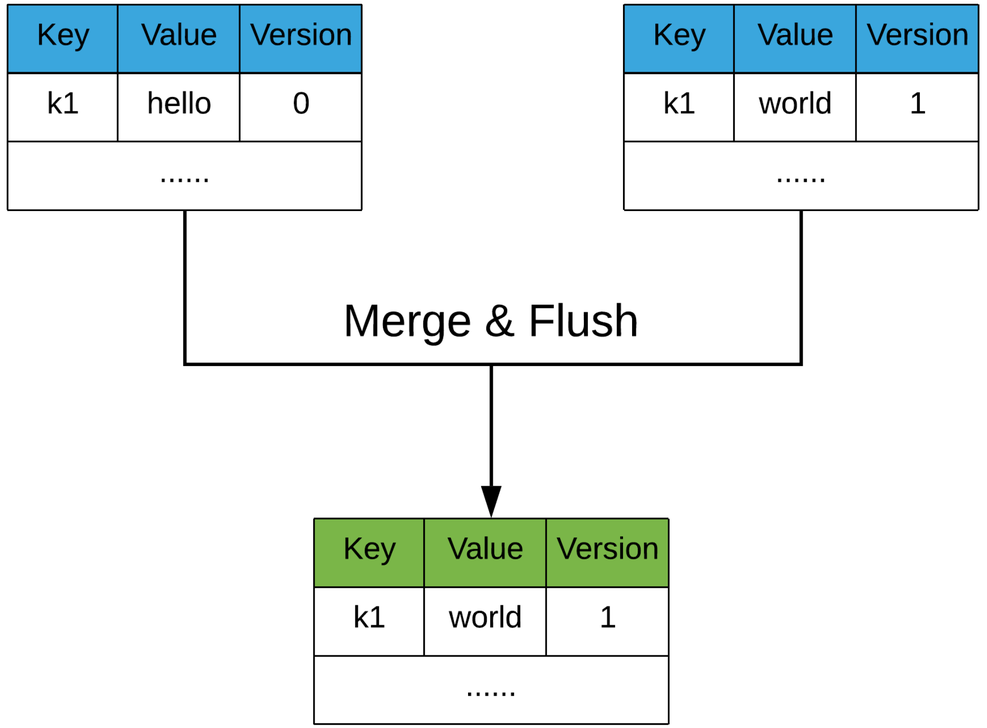
Configuration

RocksDB State Backend



RocksDB State Backend

- **Block cache size**: Cache of uncompressed blocks are read from SSTables for reads.
- **Write buffer size**: Size of MemTable.
- **Write buffer number**: Number of MemTables before flushing to disk.
- **Minimum write buffers to merge**: Number of MemTables to merge before flushing to SSTable.

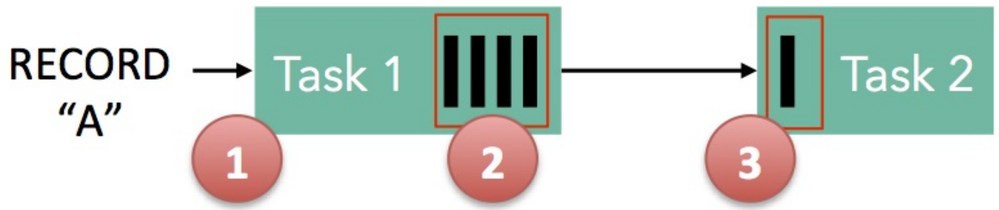


State TTL

- Feature introduced in Flink 1.6
- Can be managed by RocksDB
- Expired keys only removed on reads (or on savepoint restore)
- Flink 1.8 will add continuous cleanup and clean on compaction

Back Pressure

- Event consumption slower than event production
- Higher parallelism for expensive operator, at the cost of potentially shuffling events
- Rate limit source to avoid back pressure altogether
- Rate limiting other operators may increase checkpoint alignment time



Capacity Planning

- Job bounded by CPU, memory, or both?
- Use fewer slots than CPU cores
- Kryo serialization is expensive
- Slots per host also affects memory used by RocksDB

Summary

- “Time is of the essence”
- Reduce internal events
- Spiky workload is bad
- Understand RocksDB
- Avoid back pressure
- Do capacity planning

Questions

Real-time Analytics: klaviyo.tech/tagged/counting
Join Us: bit.ly/klaviyocareers

ning.shi@klaviyo.com
@ihsgnin