SJTU SAIL (Software Architecture and Infrastructure Lab)

SHANGHAI JIAO TONG UNIVERSITY

Streaming

Yalun Lin Hsu & Yuchen Cheng

2020-10-12

MapReduce

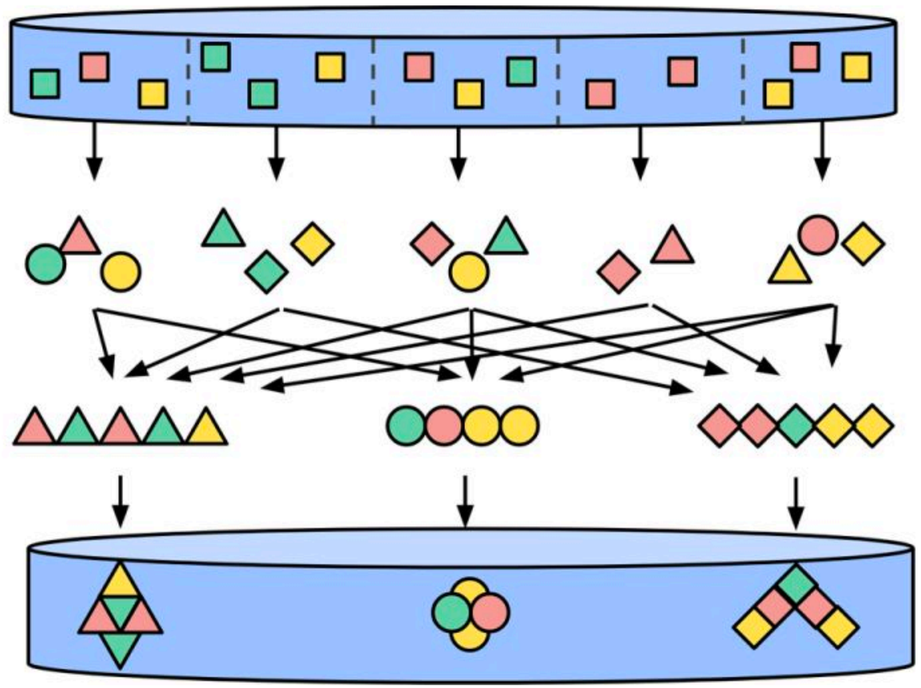
(Prepare)

Map

(Shuffle)

Reduce

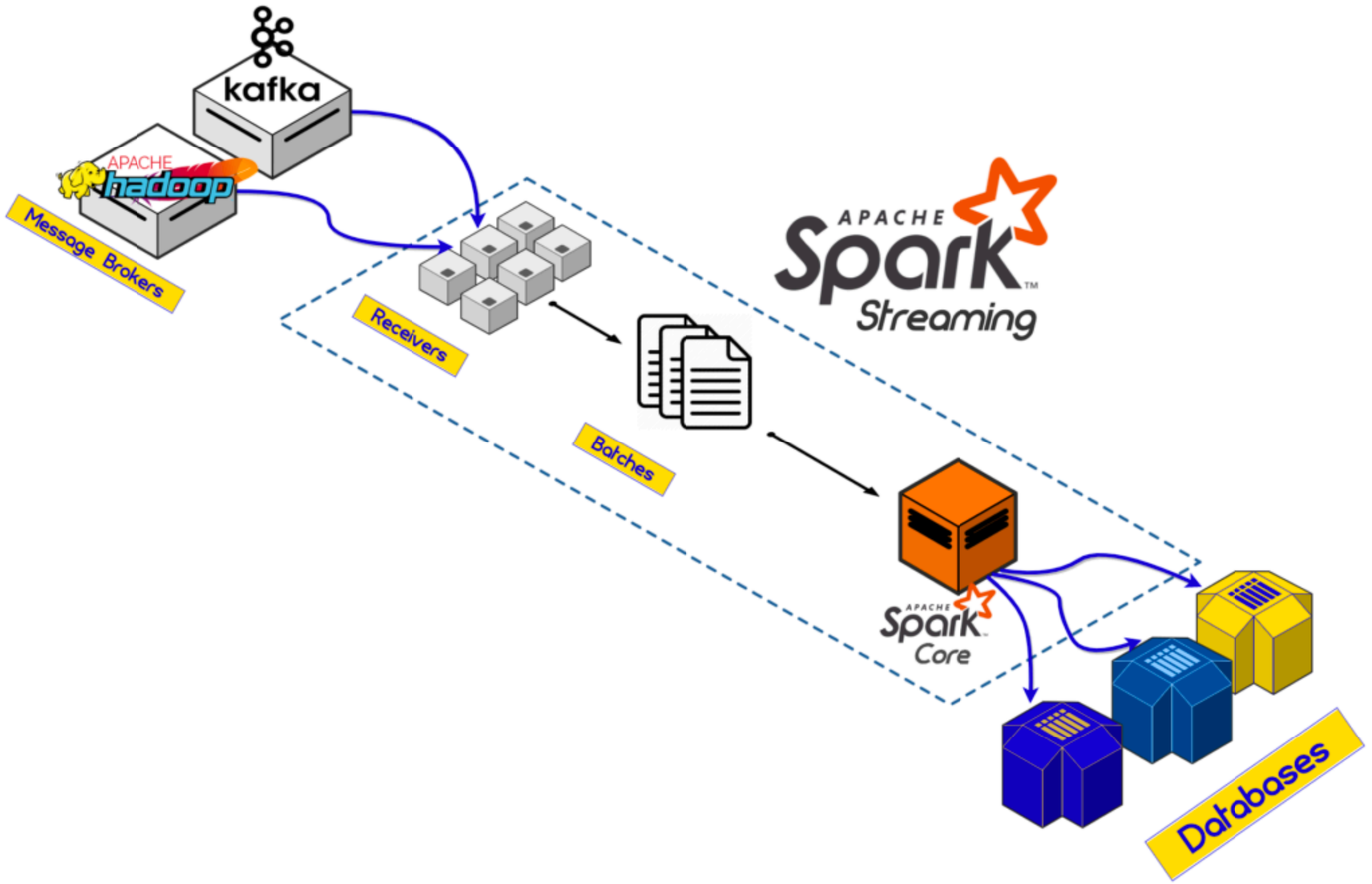
(Produce)



Hadoop & Spark

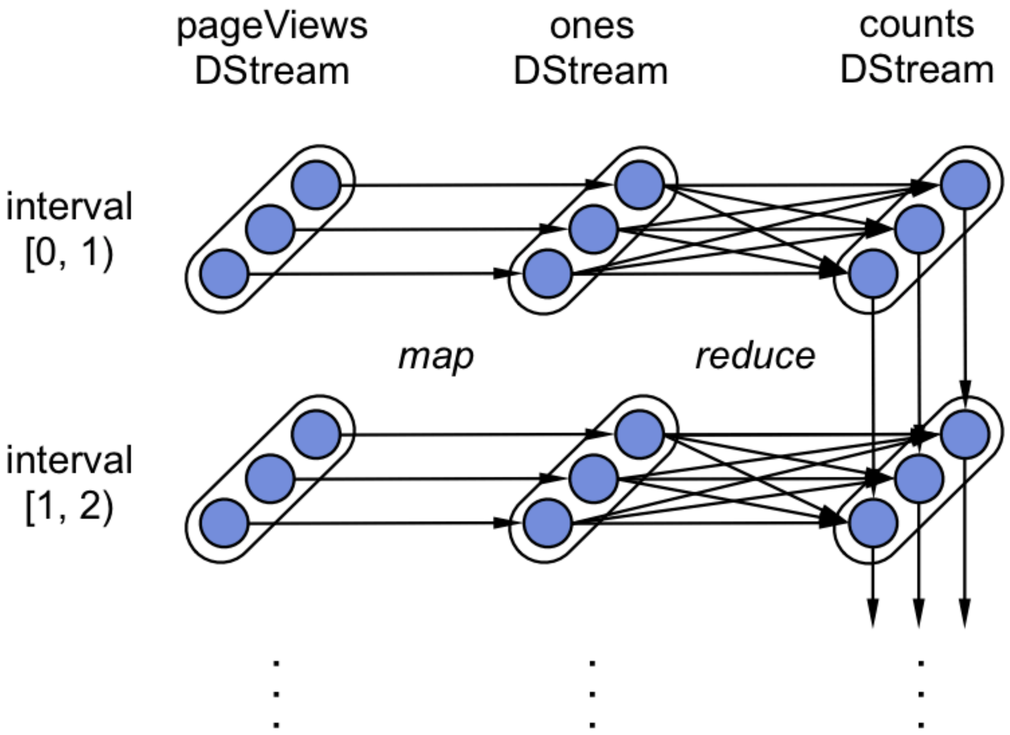
VS

Spark Streaming



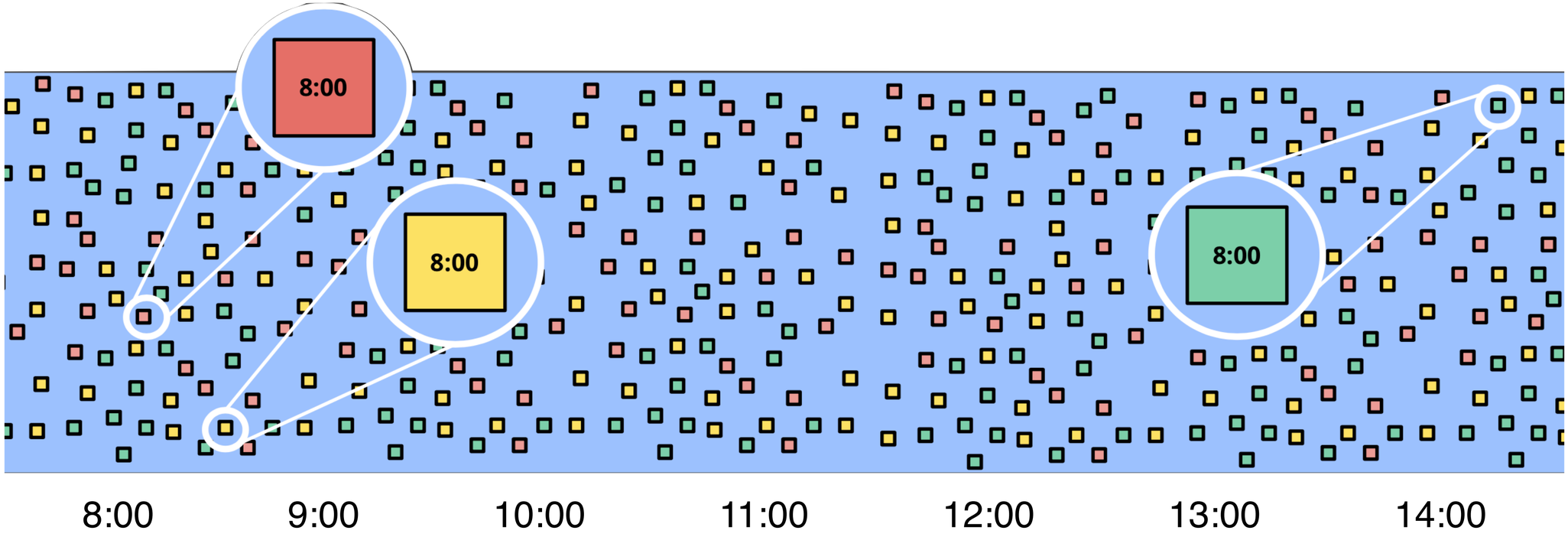
Discretized Streams

```
pageViews = readStream("http://...", "1s")
ones = pageViews.map(event ⇒ (event.url, 1))
counts = ones.runningReduce((a, b) ⇒ a + b)
```



https://dl.acm.org/doi/pdf/10.1145/2517349.2522737

What is Streaming?



Data can be infinitely big with unknown delays.

Back to Discretized Streams

Spark Streaming 1.x is not **TRUE** Streaming

Beat Batch, Build Stream Dynasty

- **Tools for reasoning about time**
  - The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-scale, Unbounded, Out-of-order Data Processing

- **Correctness**
  - Lightweight Asynchronous Snapshots for Distributed Dataflows

Beat Batch, Build Stream Dynasty

- **Tools for reasoning about time**
  - The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-scale, Unbounded, Out-of-order Data Processing

- **Correctness**
  - Lightweight Asynchronous Snapshots for Distributed Dataflows

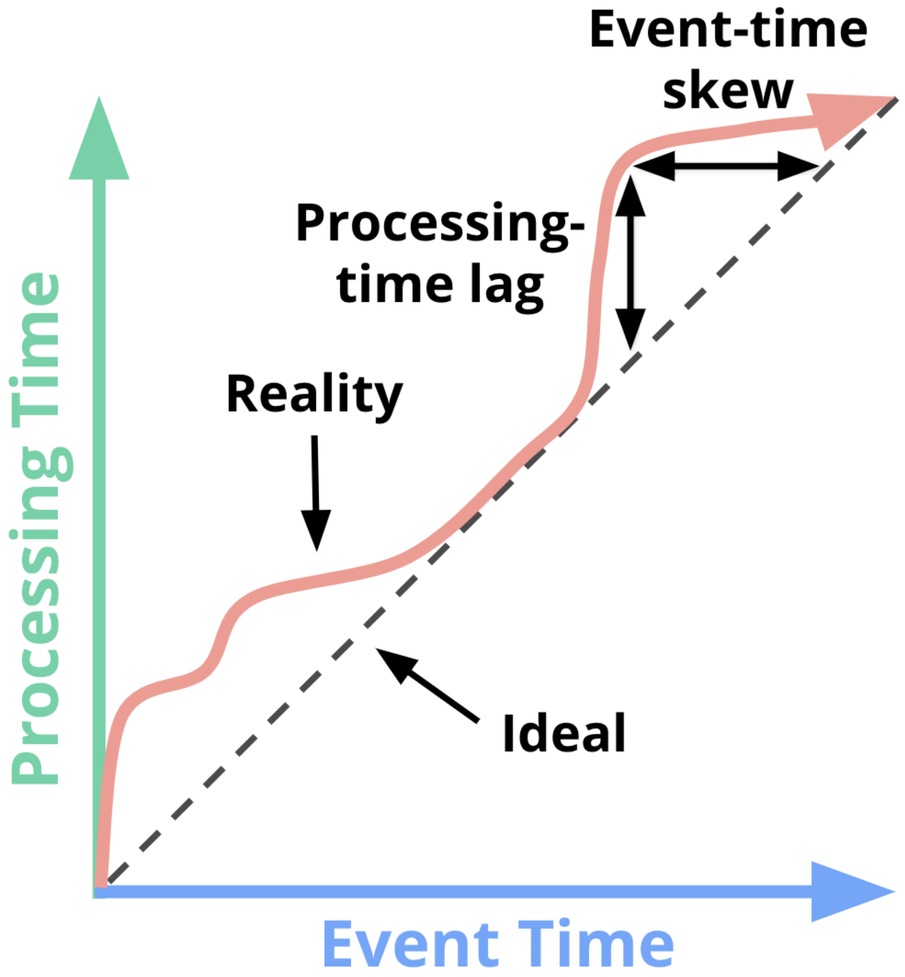
VLDB

The Dataflow Model

A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-scale, Unbounded, Out-of-order Data Processing

https://www.usenix.org/conference/atc18/presentation/oakes

Time-domain Mapping



The Dataflow Model

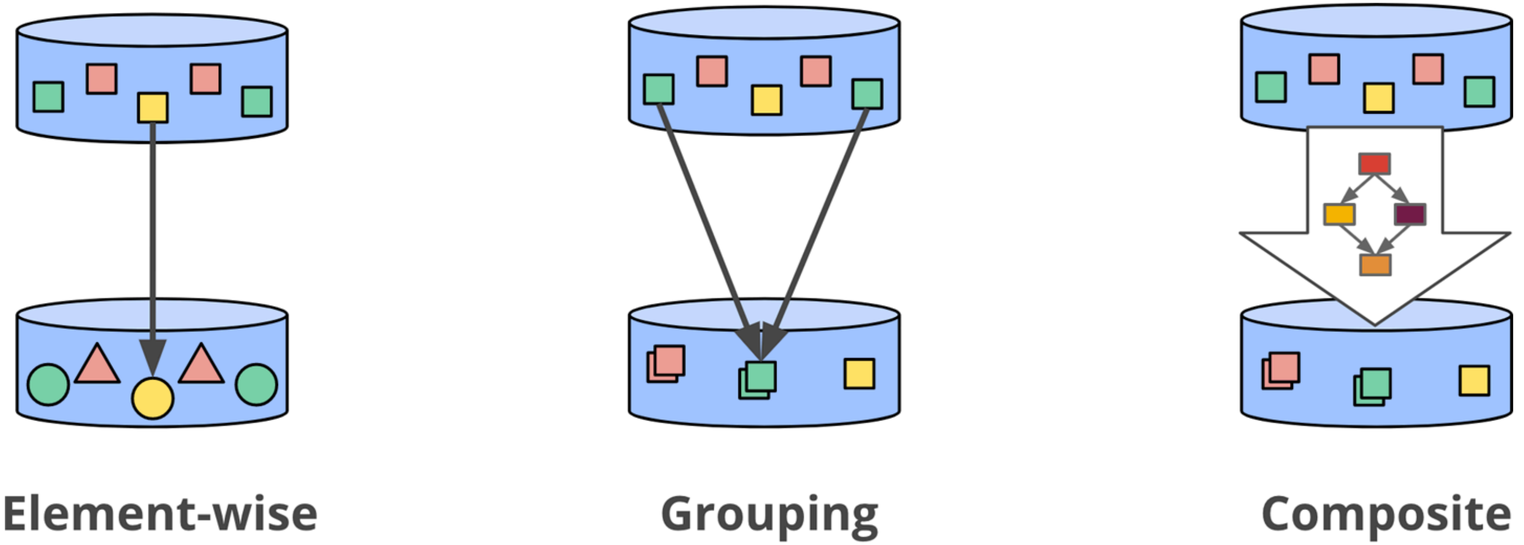
What are you computing?

Where in event time?

When in processing time?

How do refinements relate?

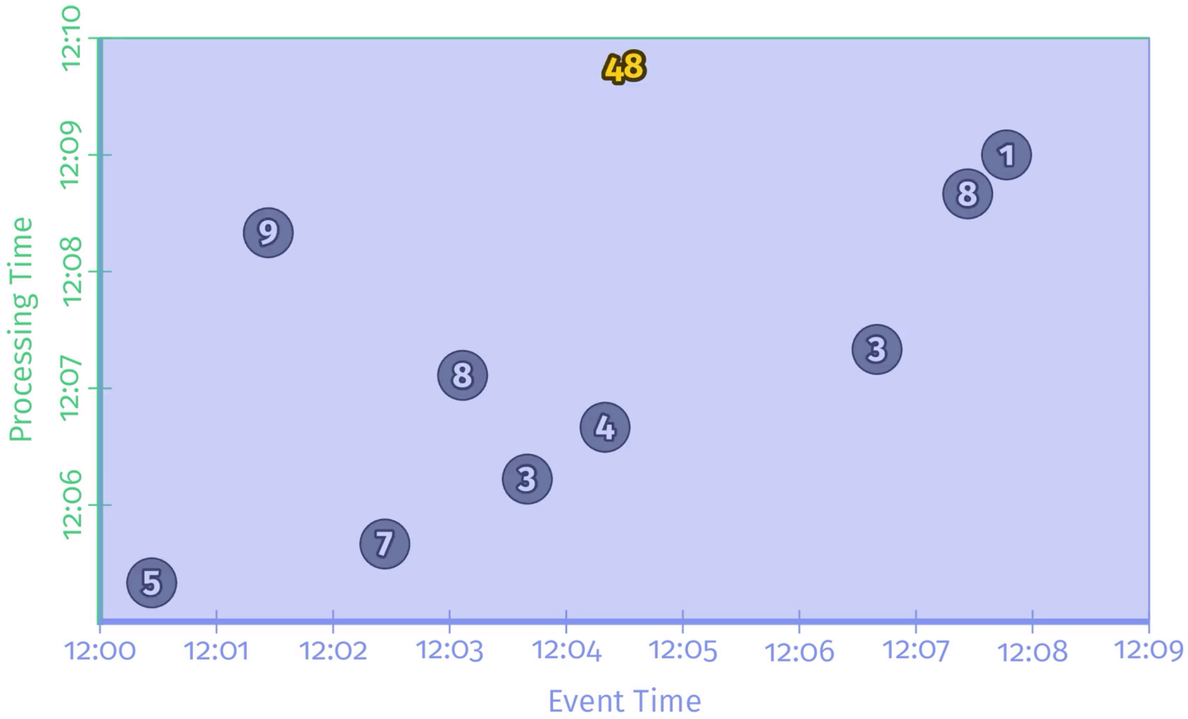
What are you computing?



Types of Transformations

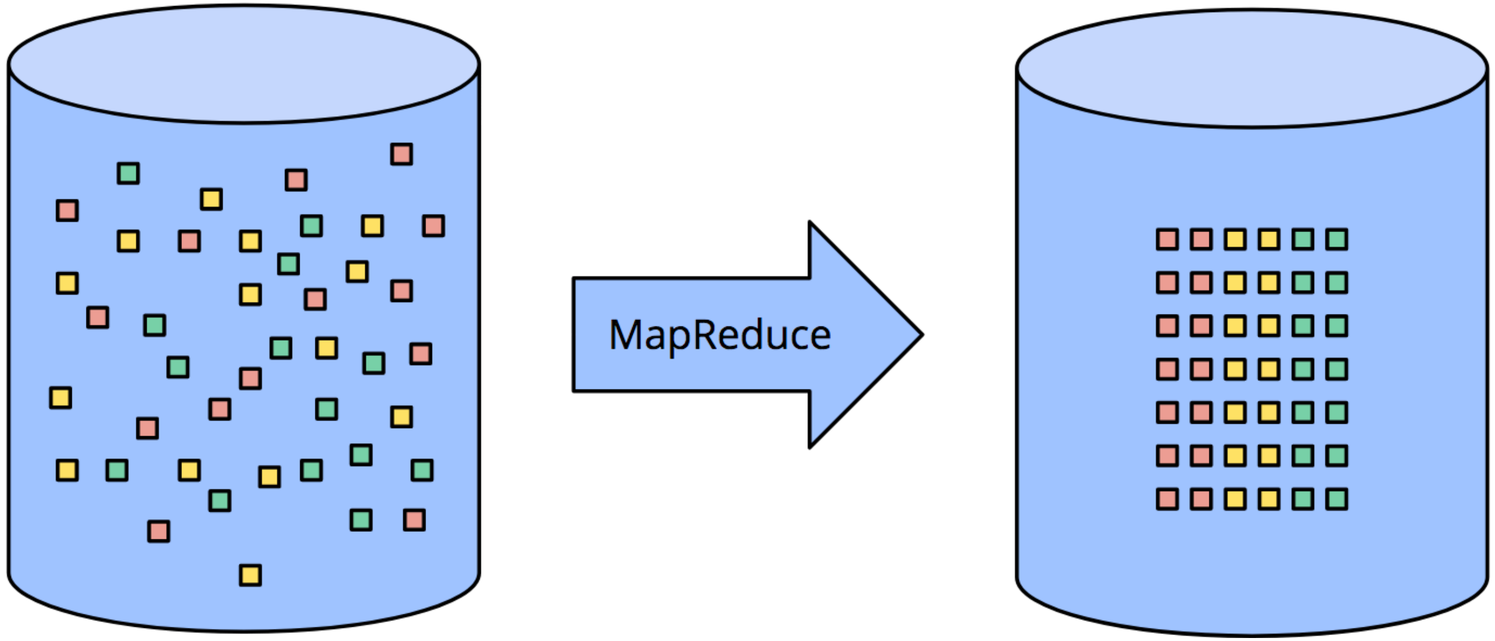
What are you computing?

```
PCollection<KV<String, Integer>> input = IO.read( ... );
PCollection<KV<String, Integer>> output = input
  .apply(Sum.integersPerKey());
```



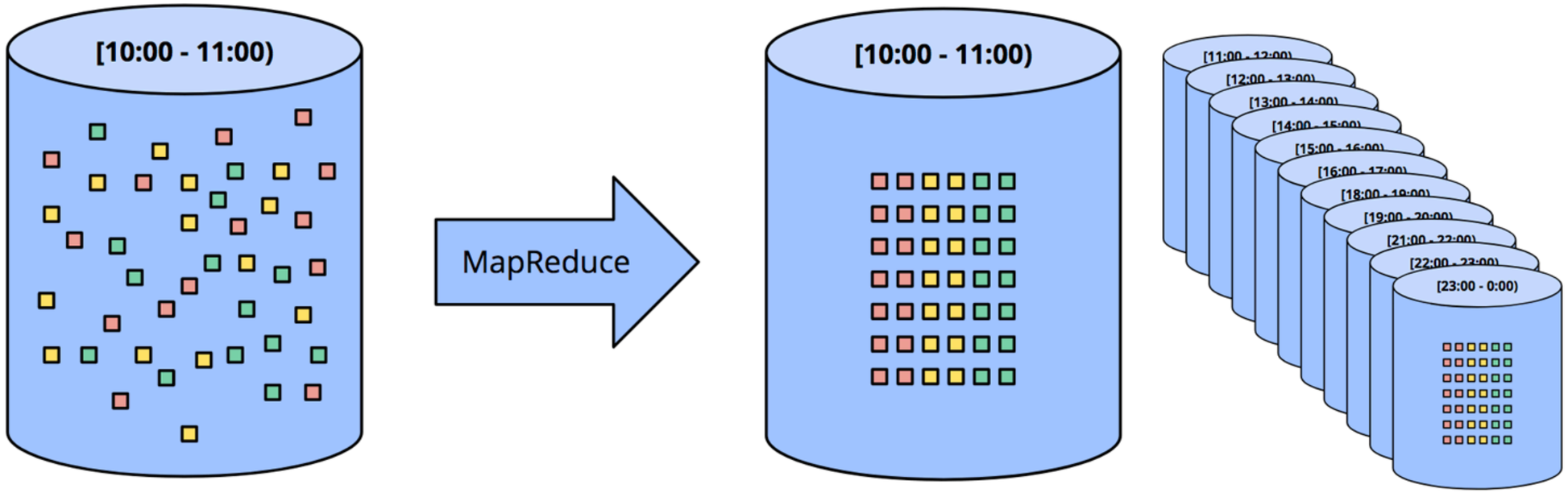
Where in event time?

Batch Example



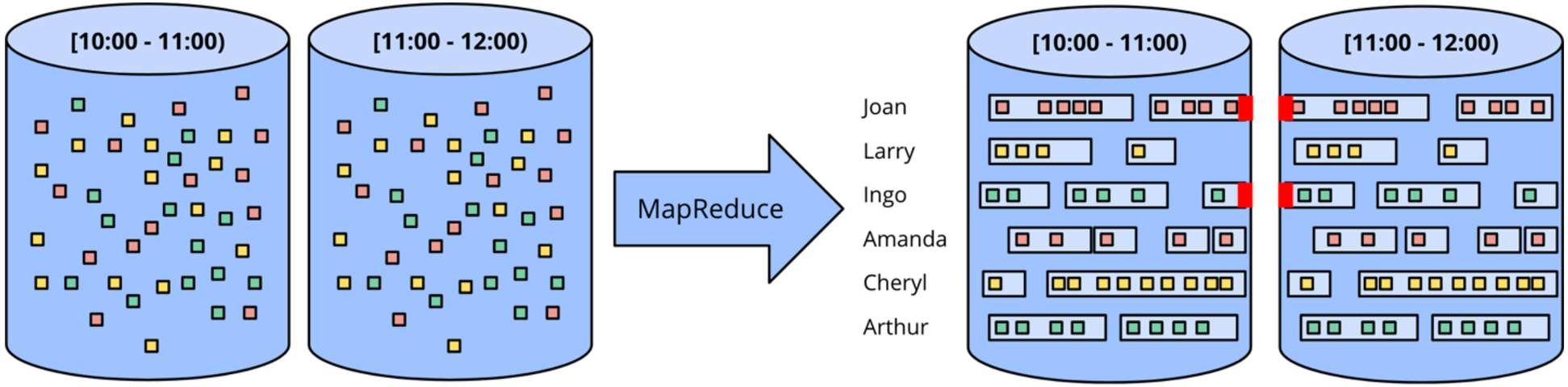
Where in event time?

Batch Example



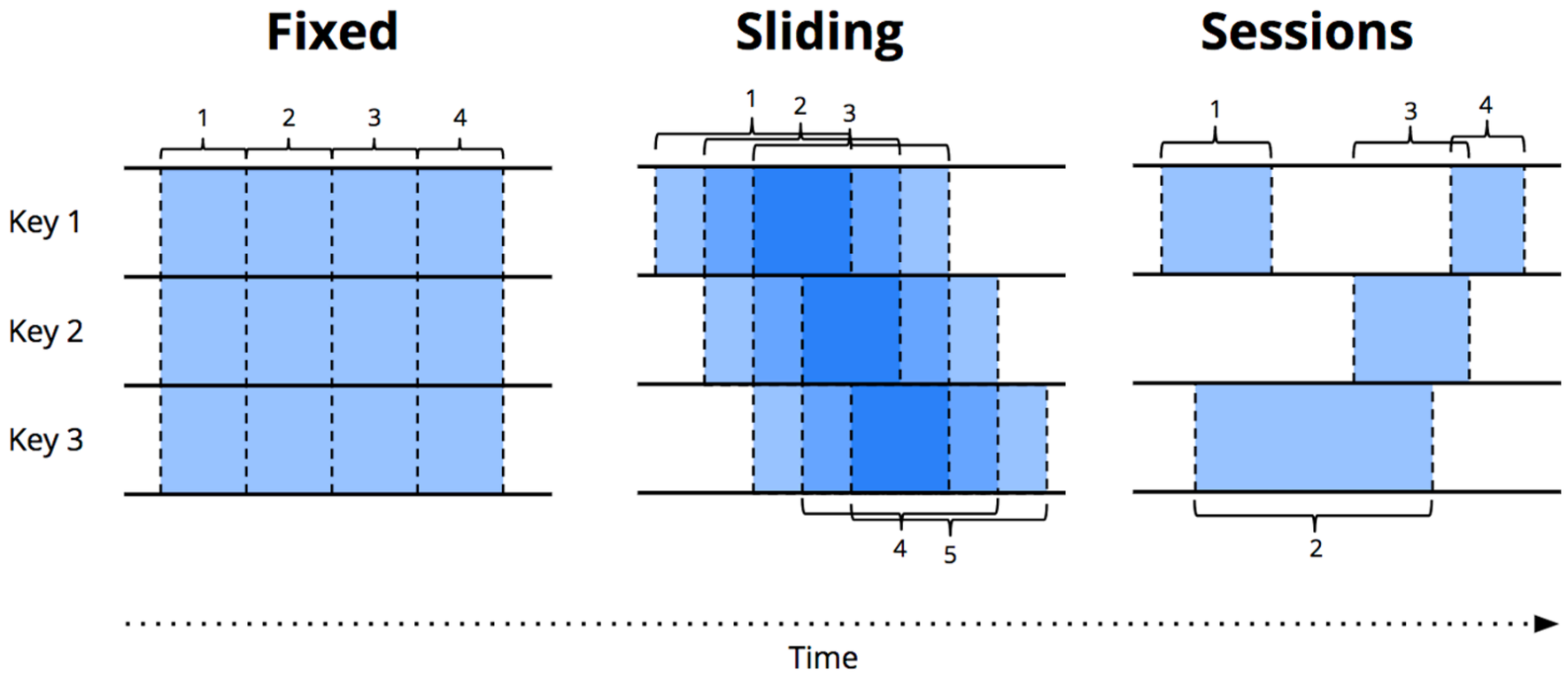
Where in event time?

Batch Example



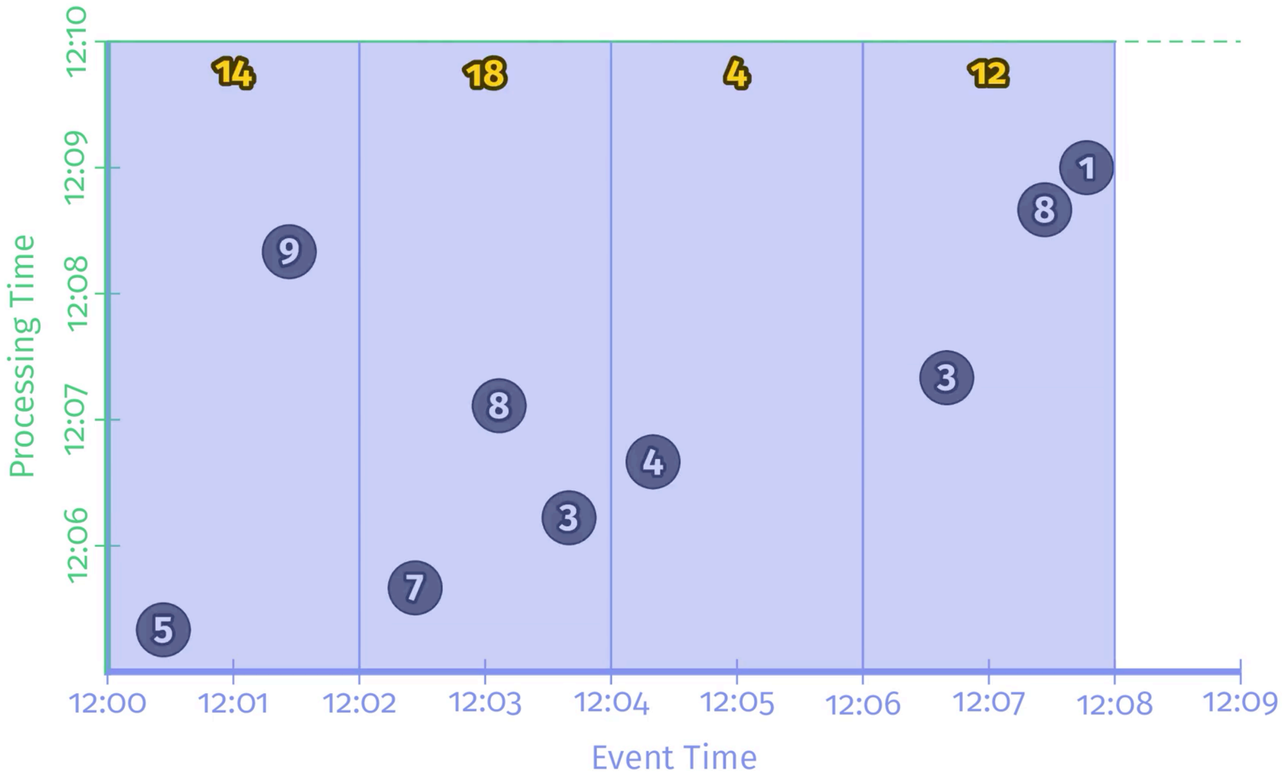
Where in event time?

Windowing



Where in event time?

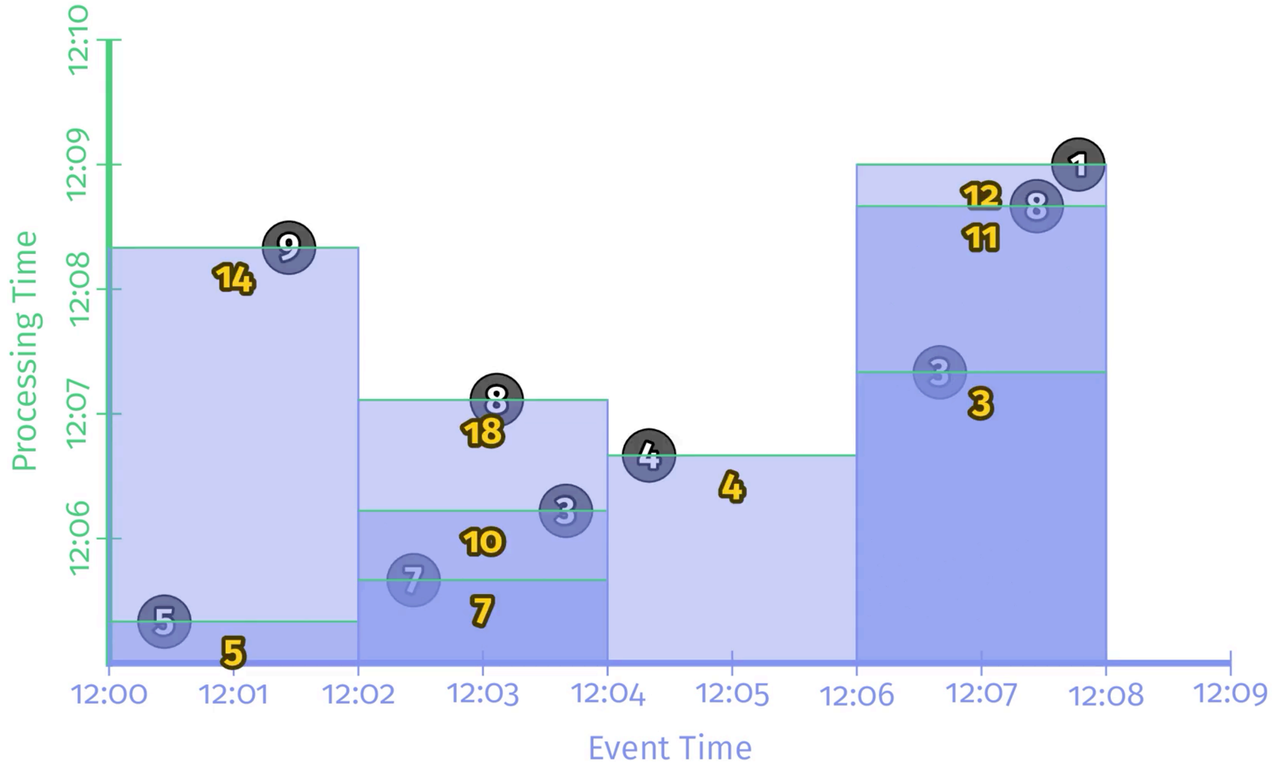
```
PCollection<KV<Team, Integer>> scores = input
  .apply(Window.into(FixedWindows.of(TWO_MINUTES)))
  .apply(Sum.integersPerKey());
```



When in processing time?

Repeated Update Triggers

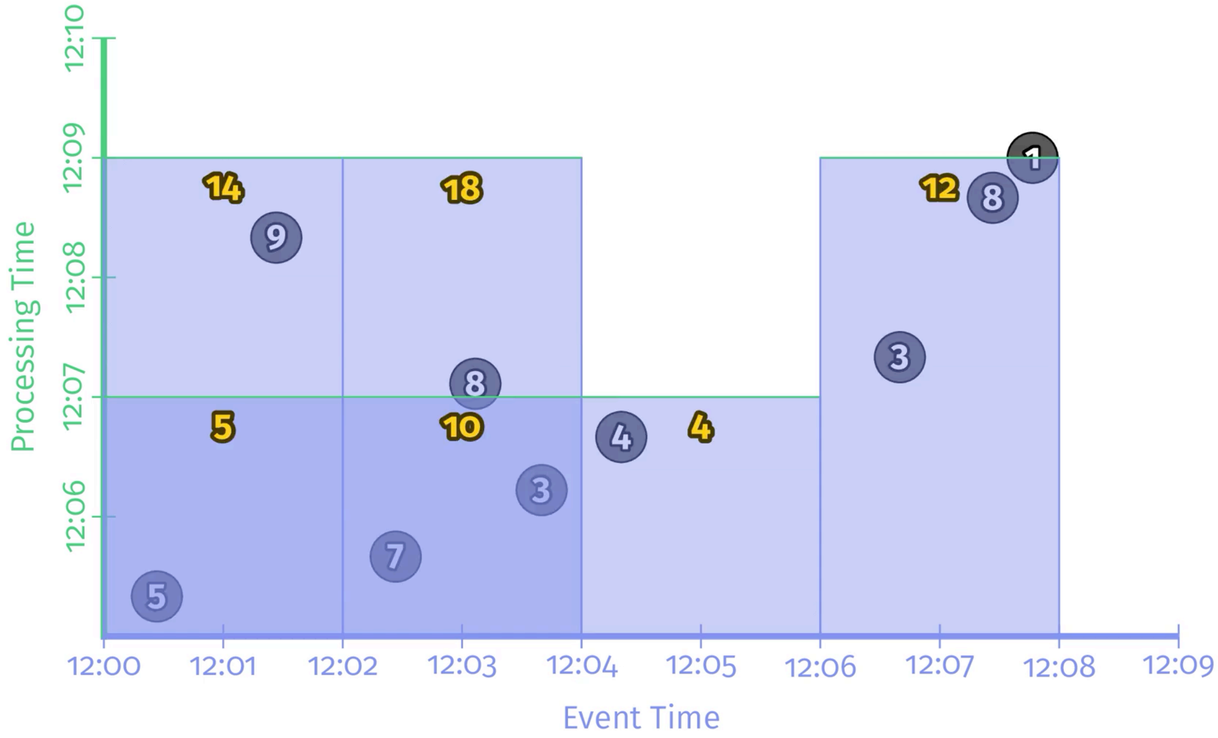
```
PCollection<KV<Team, Integer>> scores = input
 .apply(Window.into(FixedWindows.of(TWO_MINUTES))
                .triggering(Repeatedly(AfterCount(1))));
 .apply(Sum.integersPerKey());
```



When in processing time?

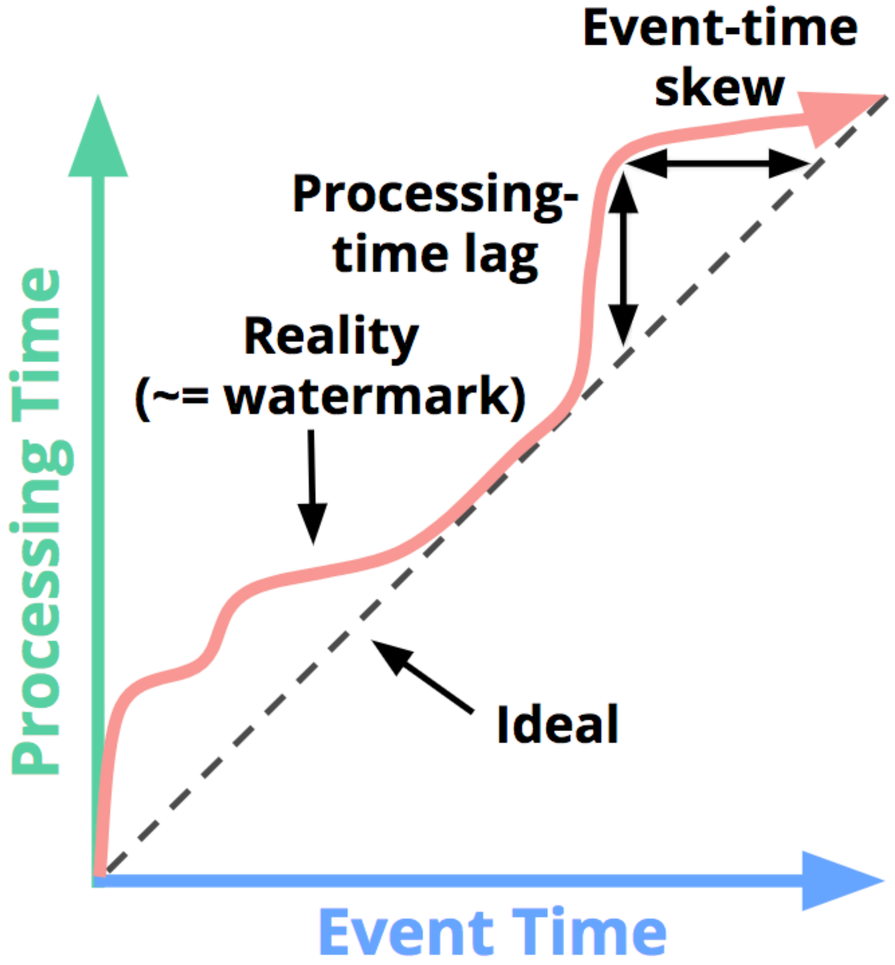
Repeated Update Triggers

```
PCollection<KV<Team, Integer>> scores = input
 .apply(Window.into(FixedWindows.of(TWO_MINUTES))
 |  |  |  |  |  |  |  .triggering(Repeatedly(AlignedDelay(TWO_MINUTES)))
 .apply(Sum.integersPerKey());
```



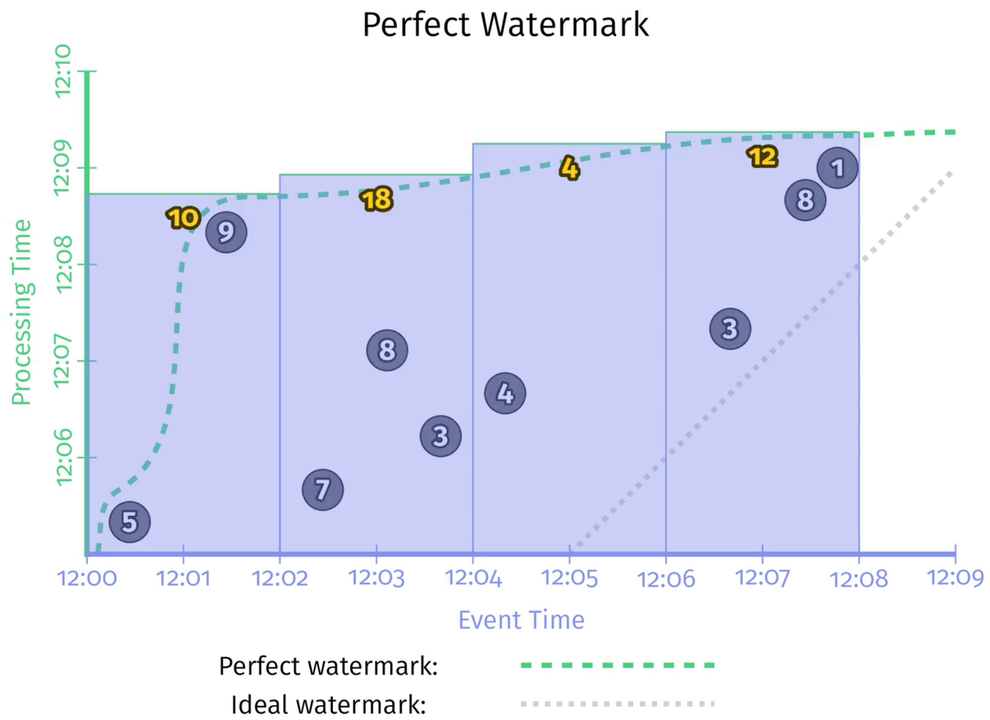
When in processing time?

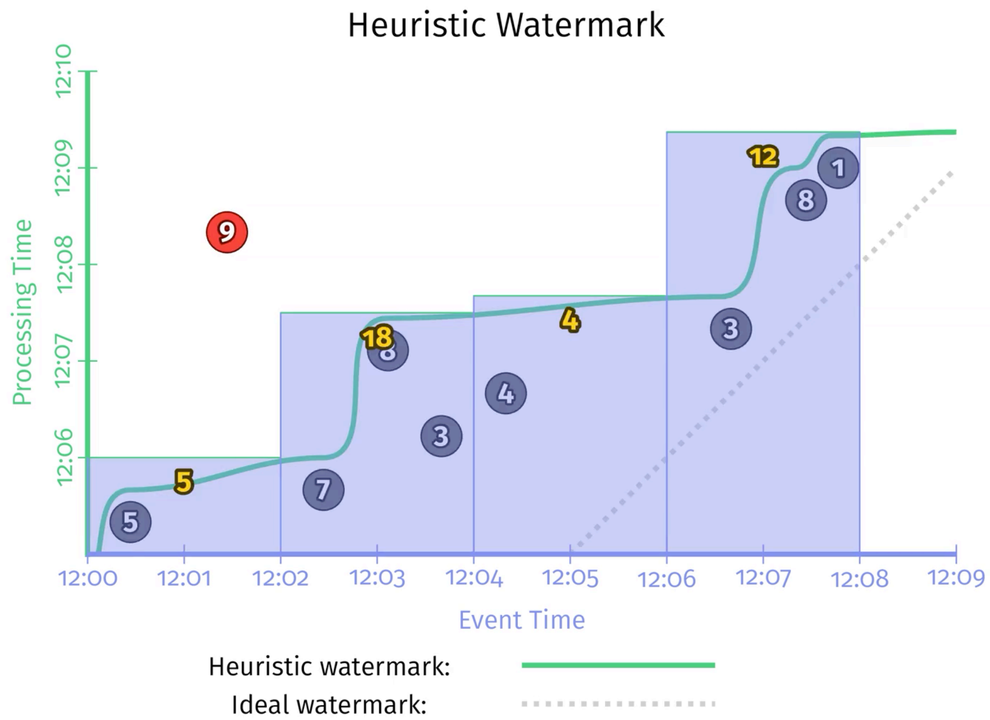
Watermarks — Completeness Triggers



When in processing time?

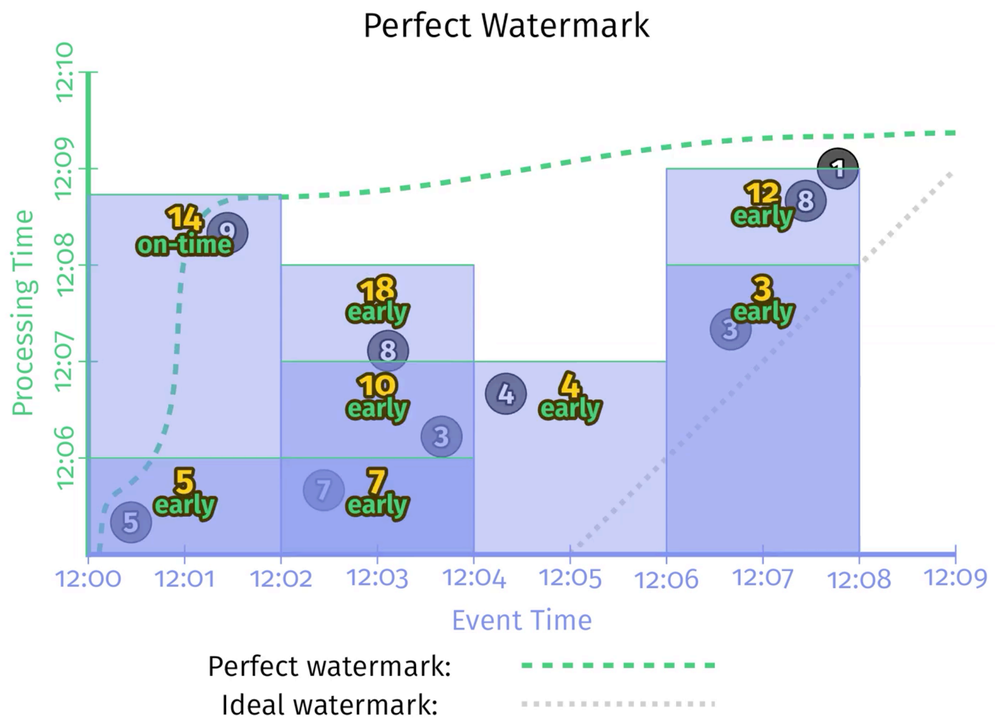
Watermarks — Completeness Triggers

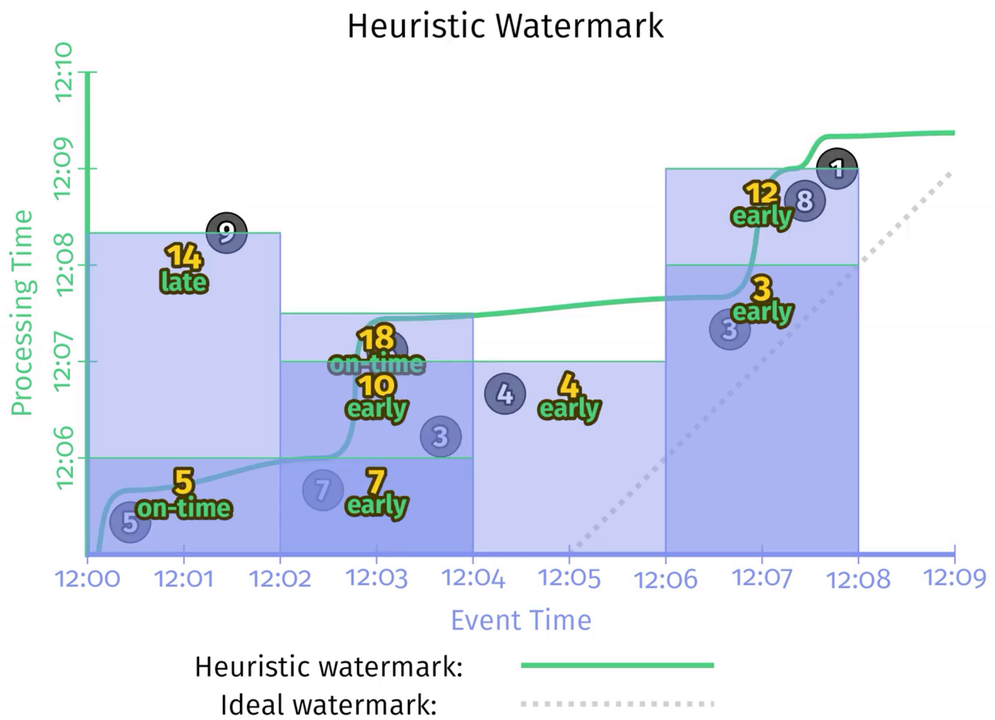




When in processing time?

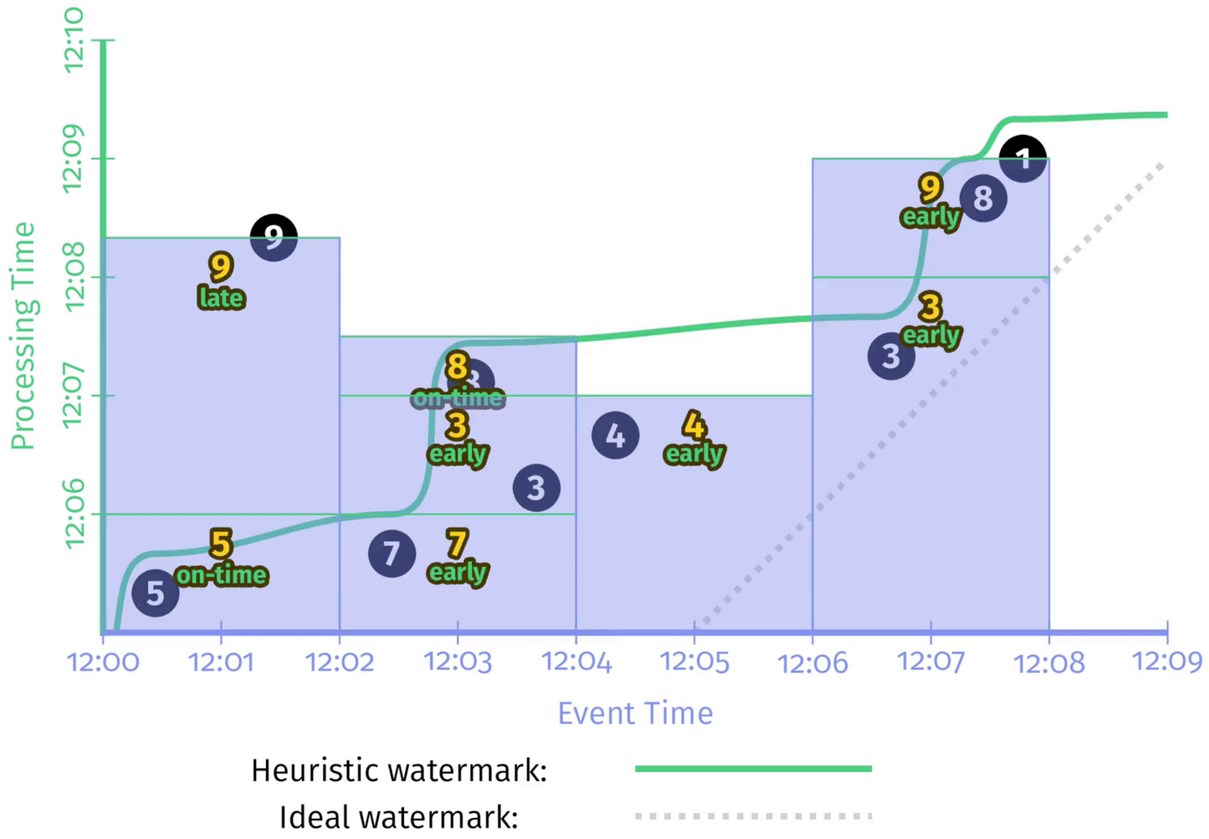
Early/On-time/Late Trigger





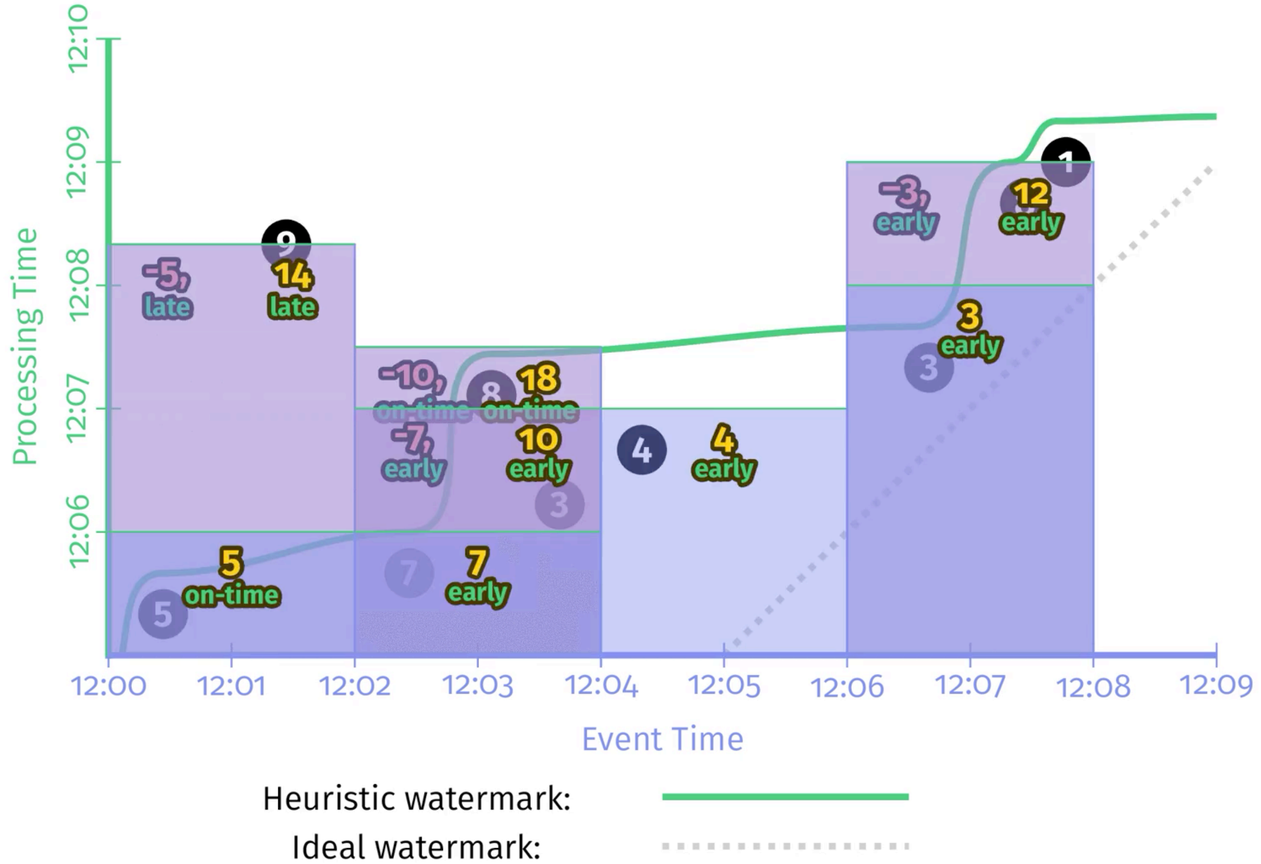
How do refinements relate?

Discarding



How do refinements relate?

Accumulating & Retracting



The Dataflow Model Summary

What are you computing?

- Pipeline Code

Where in event time?

- Windowing

When in processing time?

- Triggers & Watermark

How do refinements relate?

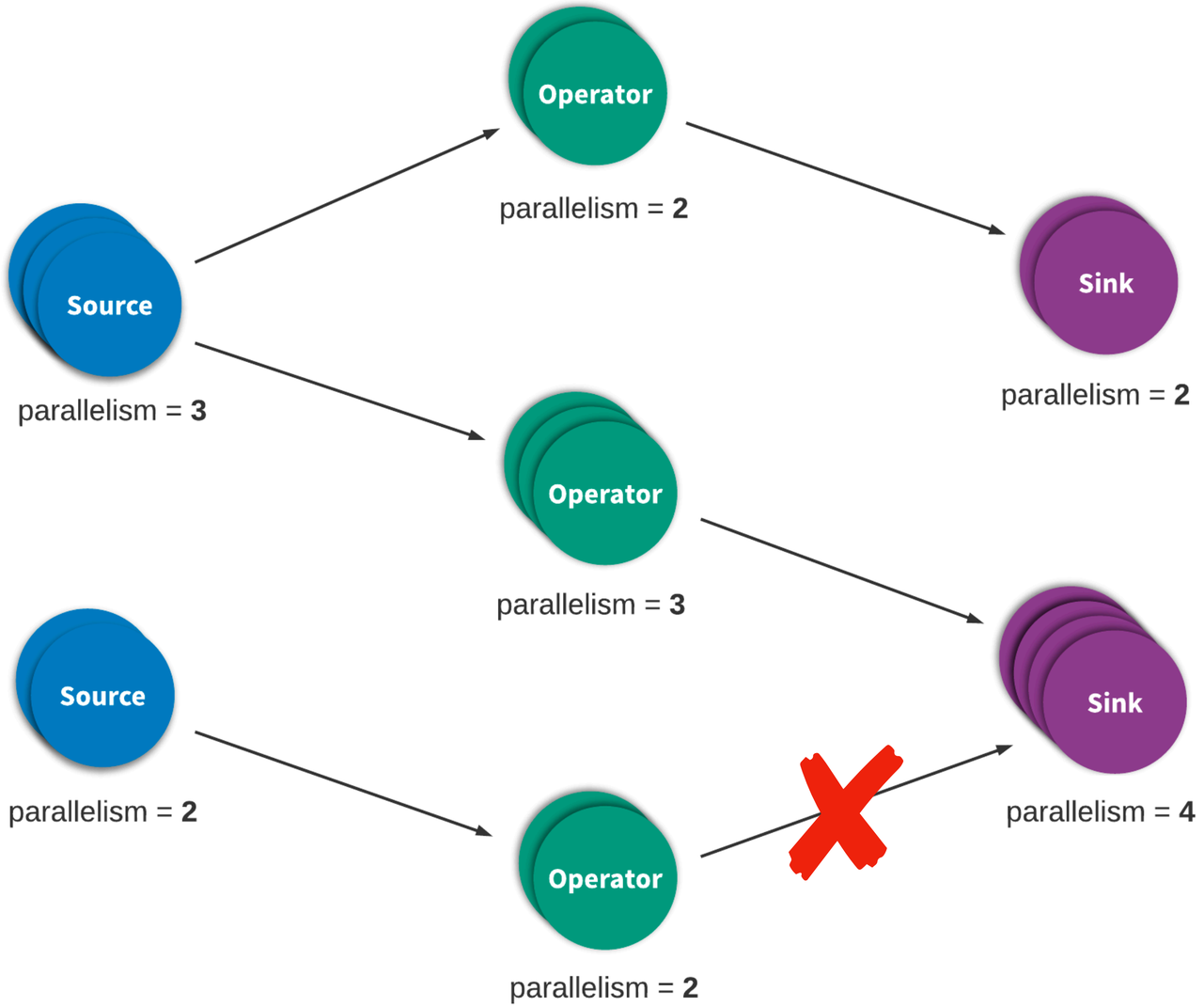
- Discarding, Accumulating and Accumulating&Retracting

Beat Batch, Build Stream Dynasty

- **Tools for reasoning about time**
  - The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-scale, Unbounded, Out-of-order Data Processing

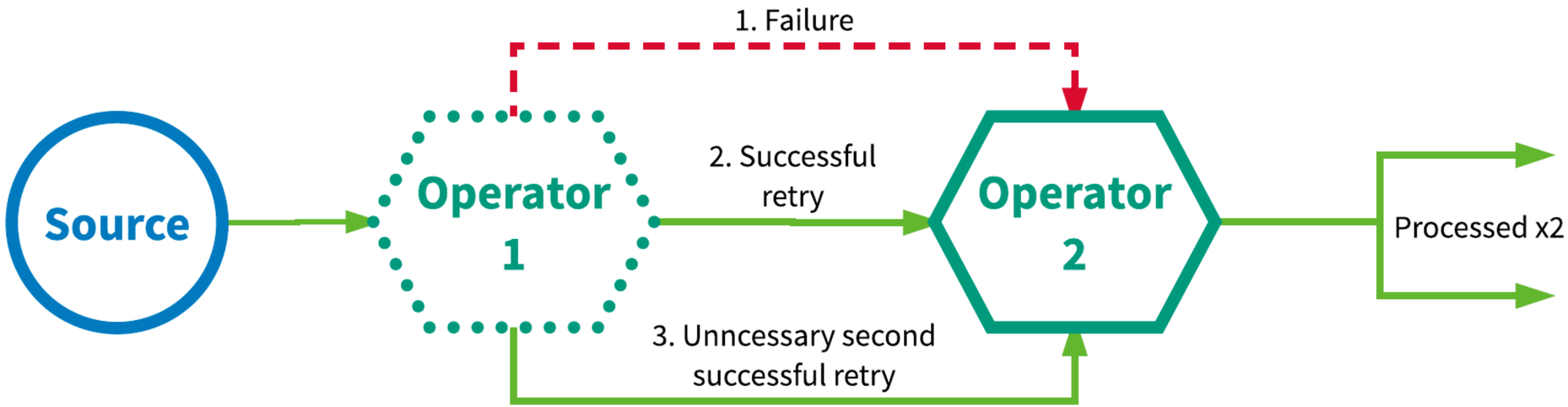
- **Correctness**
  - Lightweight Asynchronous Snapshots for Distributed Dataflows

Correctness

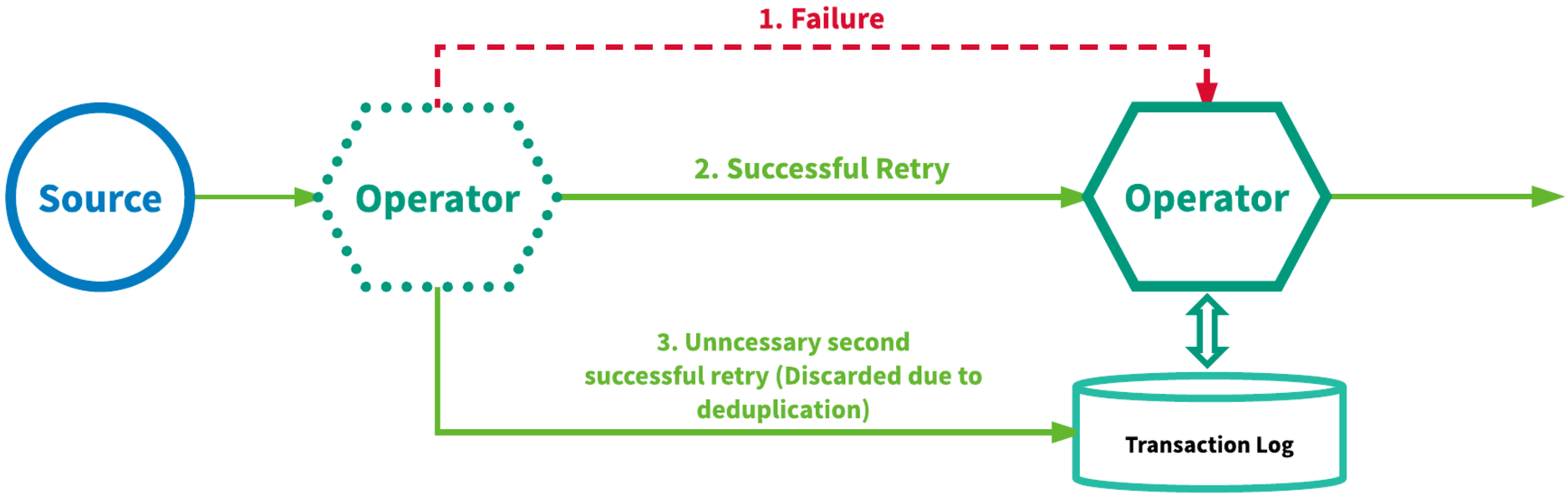


At-most-once

At-least-once



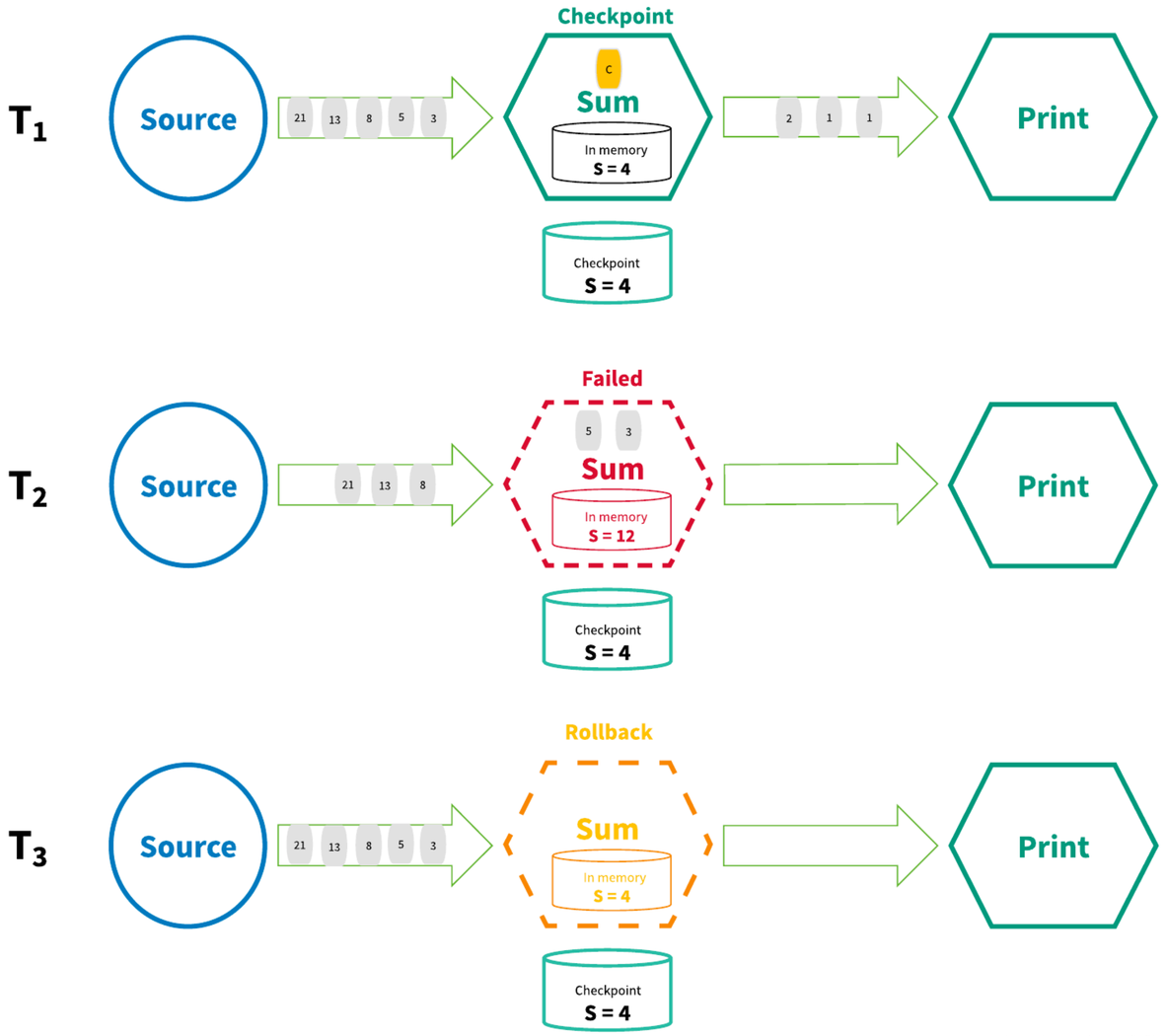
Exactly-once



MillWheel: fault-tolerant stream processing at internet scale

https://dl.acm.org/doi/10.14778/2536222.2536229

Exactly-once



ABS

Lightweight Asynchronous Snapshots for Distributed Dataflows

https://arxiv.org/abs/1506.08603

Flink System Architecture

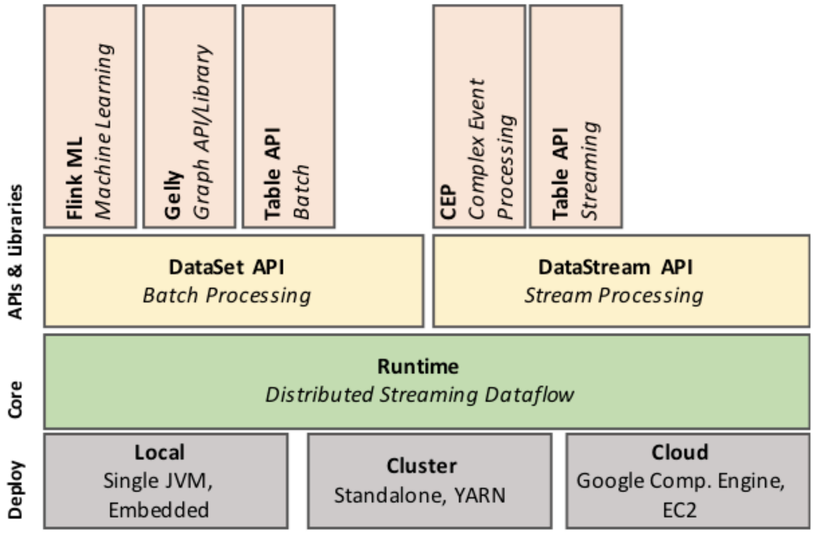


Figure 1: The Flink software stack.

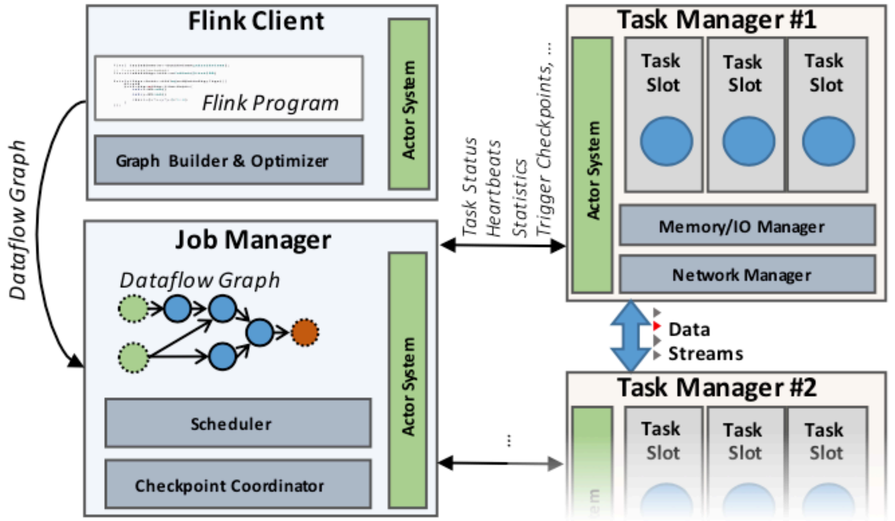
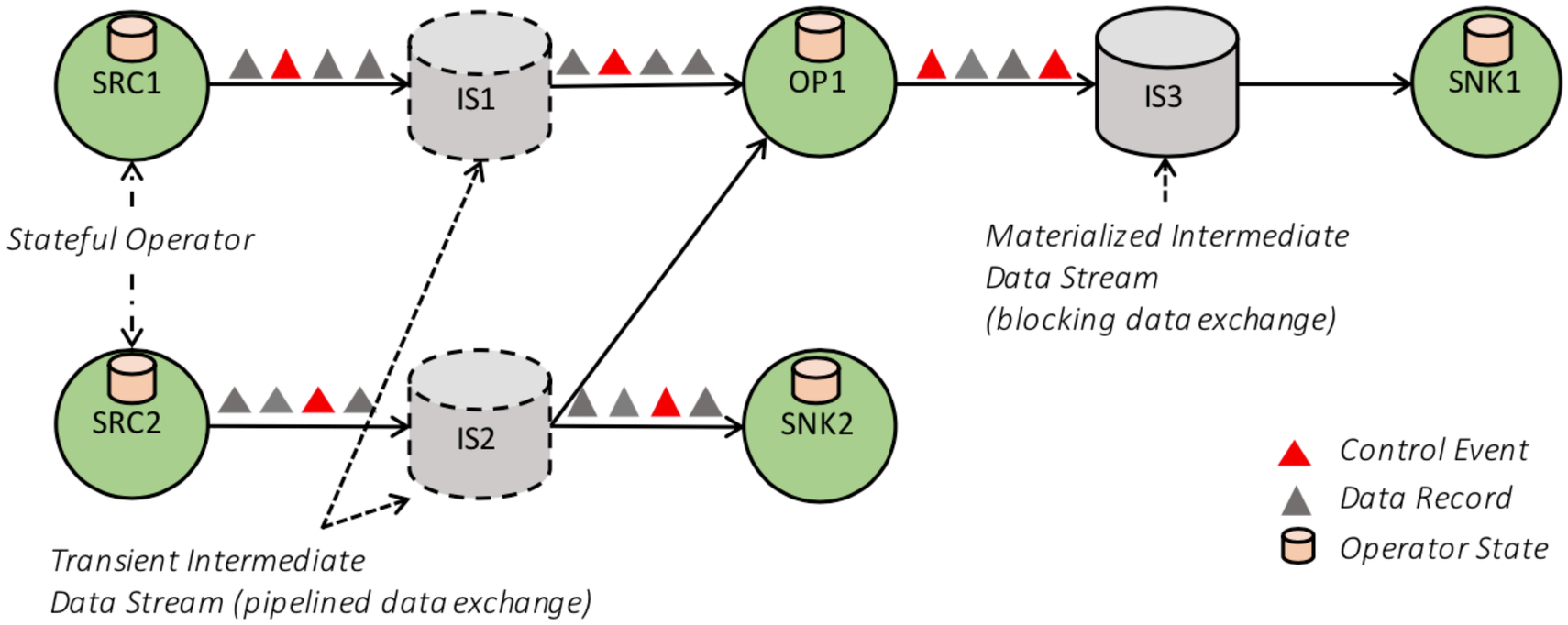


Figure 2: The Flink process model.

Apache Flink: Stream and Batch Processing in a Single Engine

http://sites.computer.org/debull/A15dec/p28.pdfs

Flink Dataflow Graph

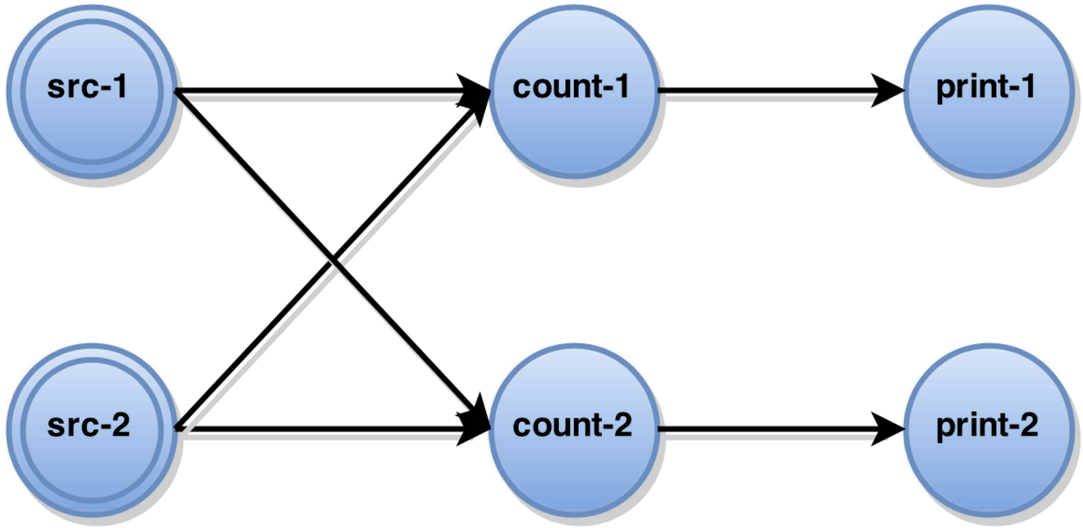


Apache Flink: Stream and Batch Processing in a Single Engine

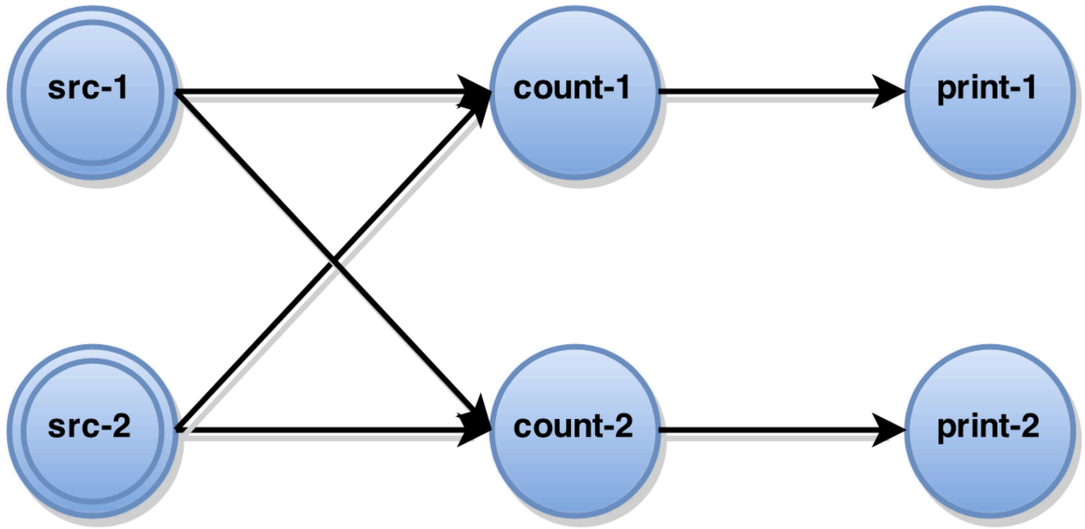
http://sites.computer.org/debull/A15dec/p28.pdfs

Word Count Example

```
1 val env : StreamExecutionEnvironment = ...
2 env.setParallelism(2)
3
4 val wordStream = env.readTextFile(path)
5 val countStream = wordStream.groupBy(_).count
6 countStream.print
```



Problem Definition



G=(T, E)

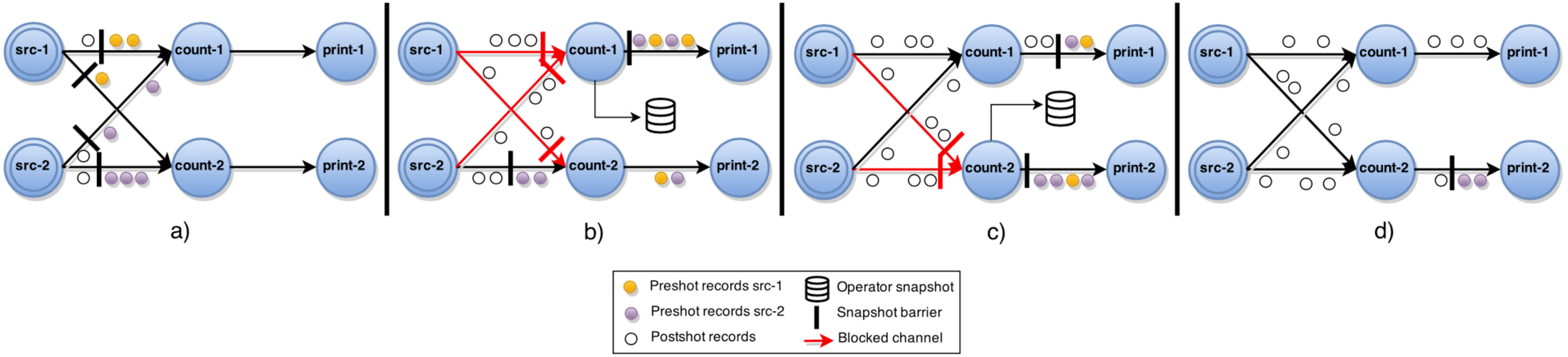
Vertex — Task

Edge — channel

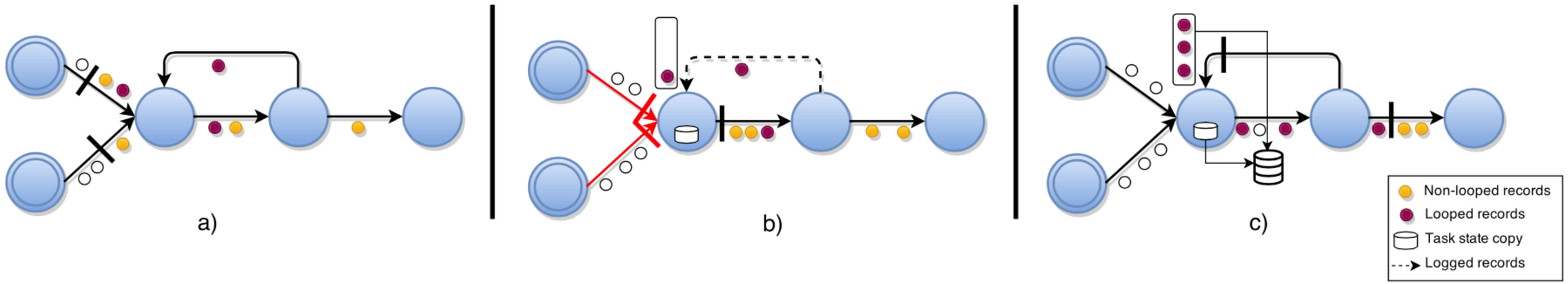
Asynchronous Barrier Snapshotting

- **ABS for Acyclic Dataflows**
- **ABS for Cyclic Dataflows**

ABS for Acyclic Dataflows



ABS for Cyclic Dataflows



Correctness Summary

- Exactly-Once
  - At-least-once Transmission + Deduplication
  - Distributed Snapshots
    - Asynchronous Barrier Snapshotting

Thanks