

بسم الله الرحمن الرحيم

تکنولوژی کامپیوتر

جلسه‌ی بیست و سوم
پردازش جریان

جلسه گذشته

جلسه‌ی جدید

Polling for changes...

مفاهيم

Topic

- a “stream” refers to data that is incrementally made available over time.
- In a stream processing context, a record is more commonly known as an event, but it is essentially the same thing: a small, selfcontained, immutable object containing the details of something that happened at some point in time.
- Each group of related events is called a topic

Producer

- Publisher or sender

Consumer

- Subscribers or recipients

انواع جریان‌های ایونت

Messaging Systems

- A common approach for notifying consumers about new events is to use a messaging system.

Two question for design choice

- What happens if the producers send messages faster than the consumers can process them?
 - *Drop*
 - *Buffer*
 - *Backpressure*

Two question for design choice

- What happens if nodes crash or temporarily go offline—are any messages lost?

Direct messaging from producers to consumers

- UDP/TCP multicast
- UDP messages
- Direct HTTP or RPC / Webhooks

Message brokers (Message queue)

Handling Multiple consumers

- Load balancing
- Fan-out

Handling Multiple consumers

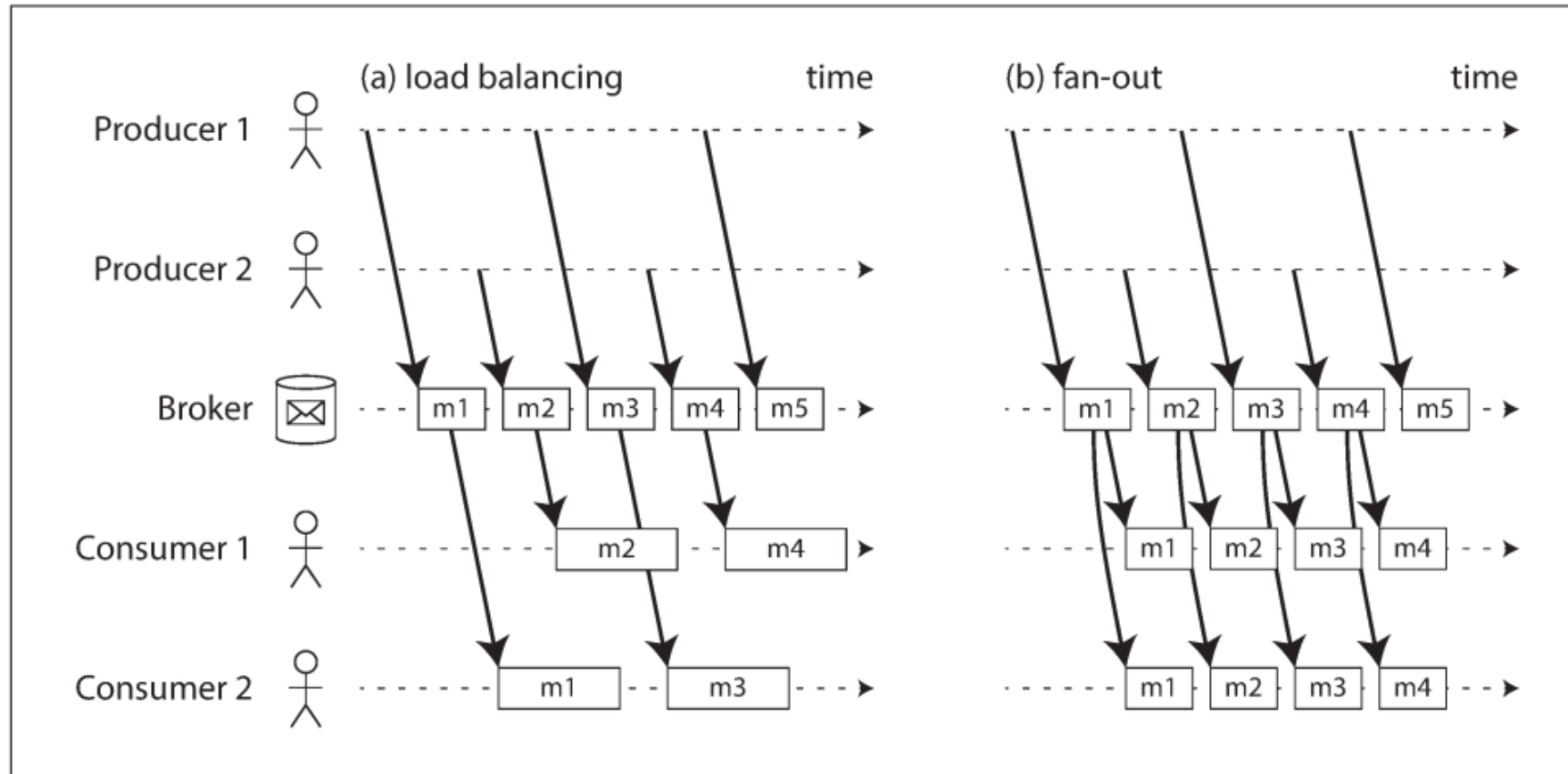


Figure 11-1. (a) Load balancing: sharing the work of consuming a topic among consumers; (b) fan-out: delivering each message to multiple consumers.

Acknowledgments and redelivery

Acknowledgments and redelivery

■ Ordering Is

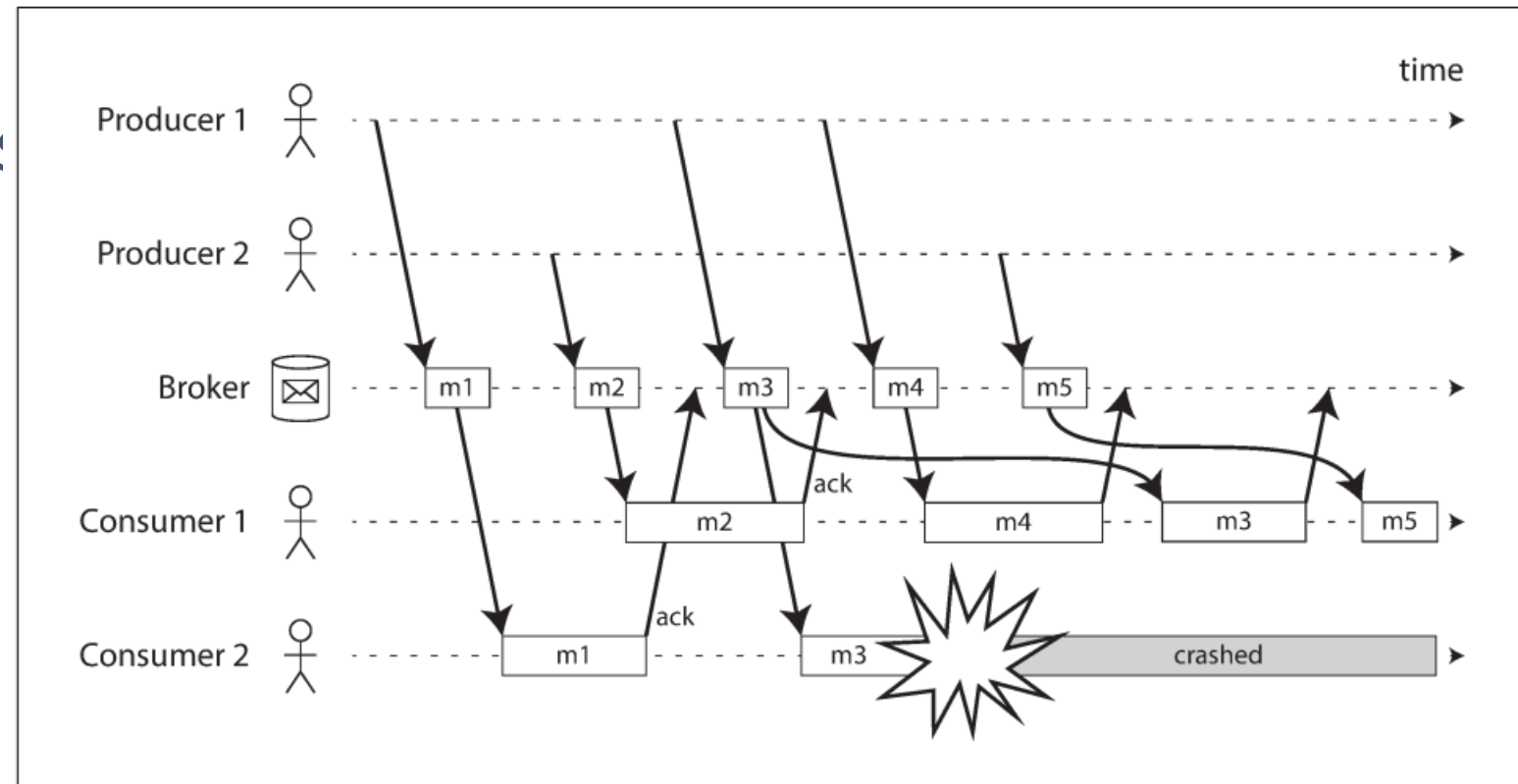


Figure 11-2. Consumer 2 crashes while processing m3, so it is redelivered to consumer 1 at a later time.

Partitioned Logs

- Append new event to file
- Offset
- Partition logs

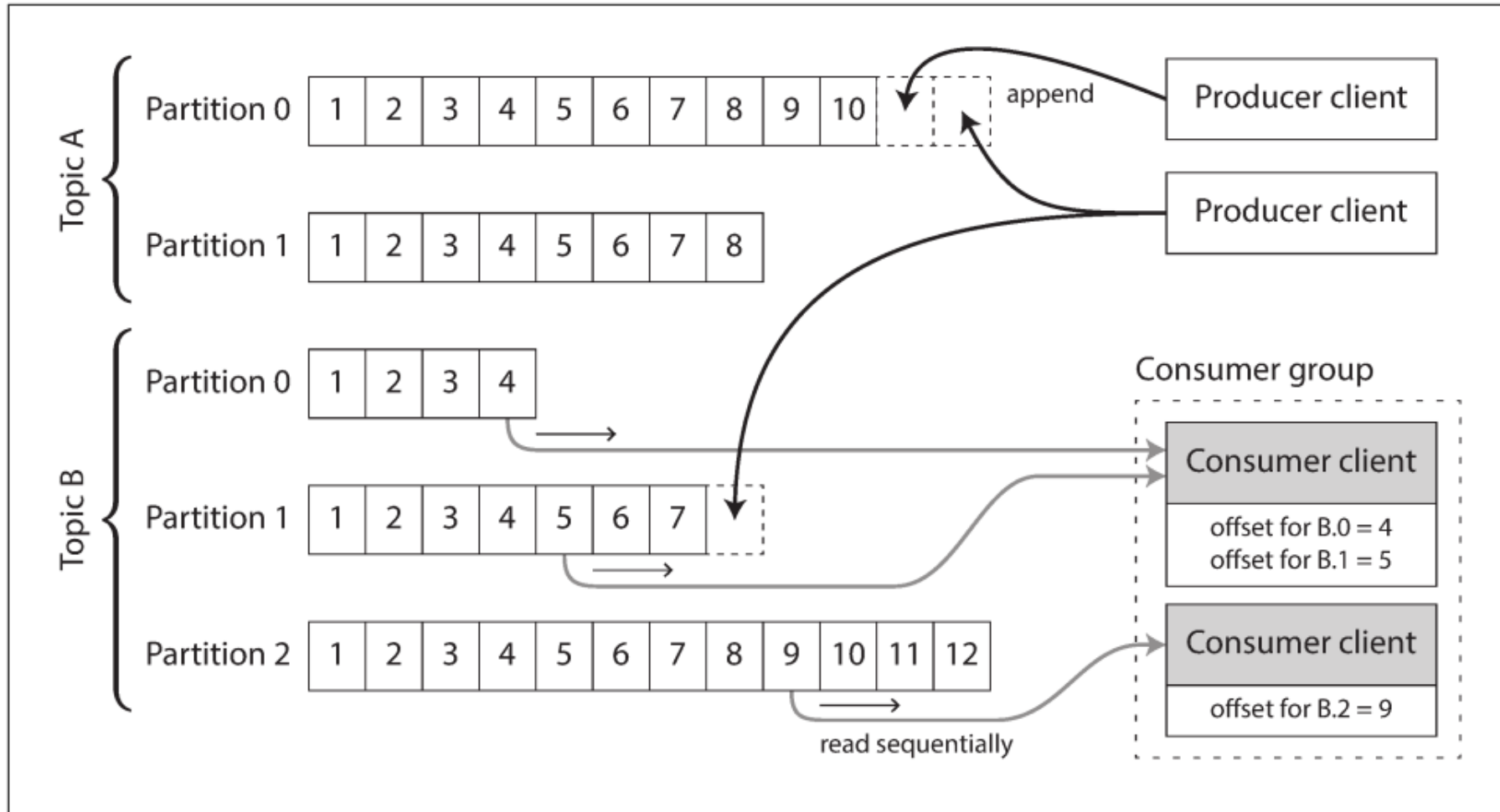


Figure 11-3. Producers send messages by appending them to a topic-partition file, and consumers read these files sequentially.

Logs compared to traditional messaging

Consumer offsets

Disk space usage

Handling disk usage

- Circular buffer
- Delete old segment
- Compaction

پردازش جریان و پایگاه داده‌ها

Keeping Systems in Sync

- Assume you have a OLTP database + Search Index (two databases)
 - *Dual writes*

Keeping Systems in Sync

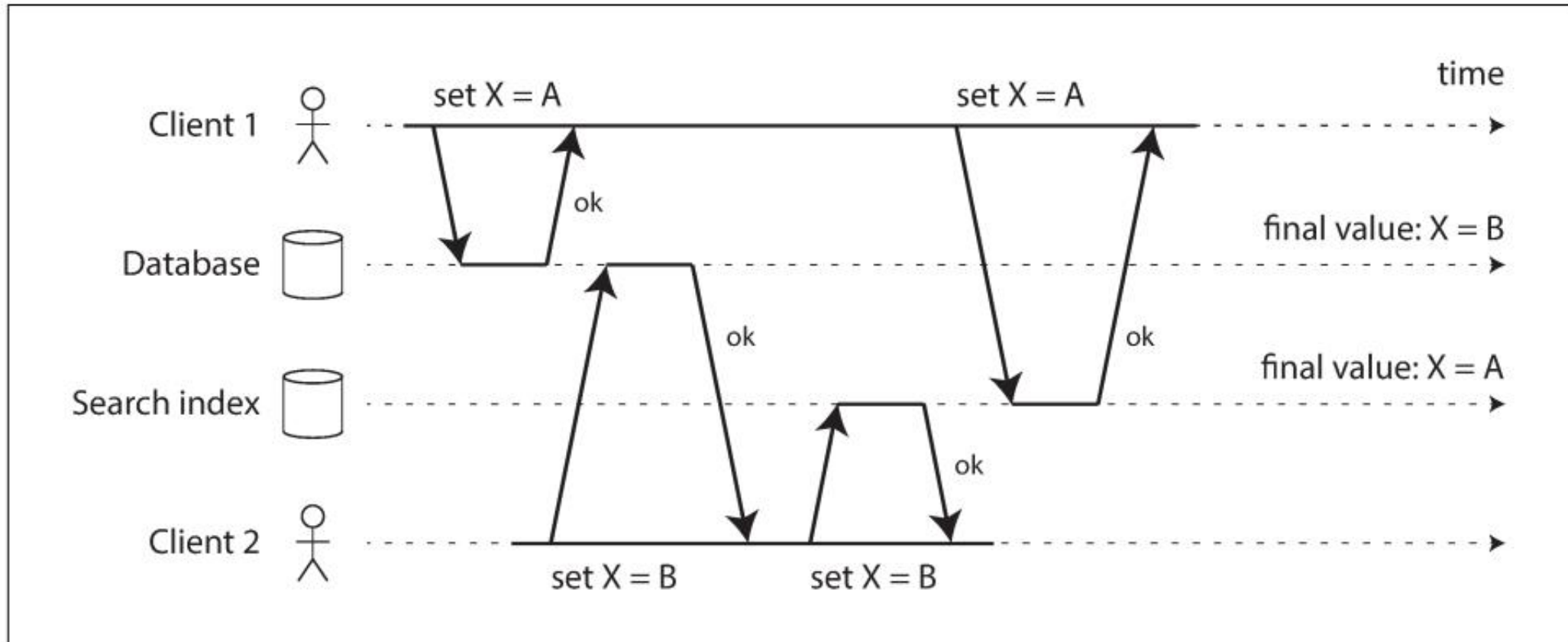


Figure 11-4. In the database, X is first set to A and then to B, while at the search index the writes arrive in the opposite order.

Change Data Capture

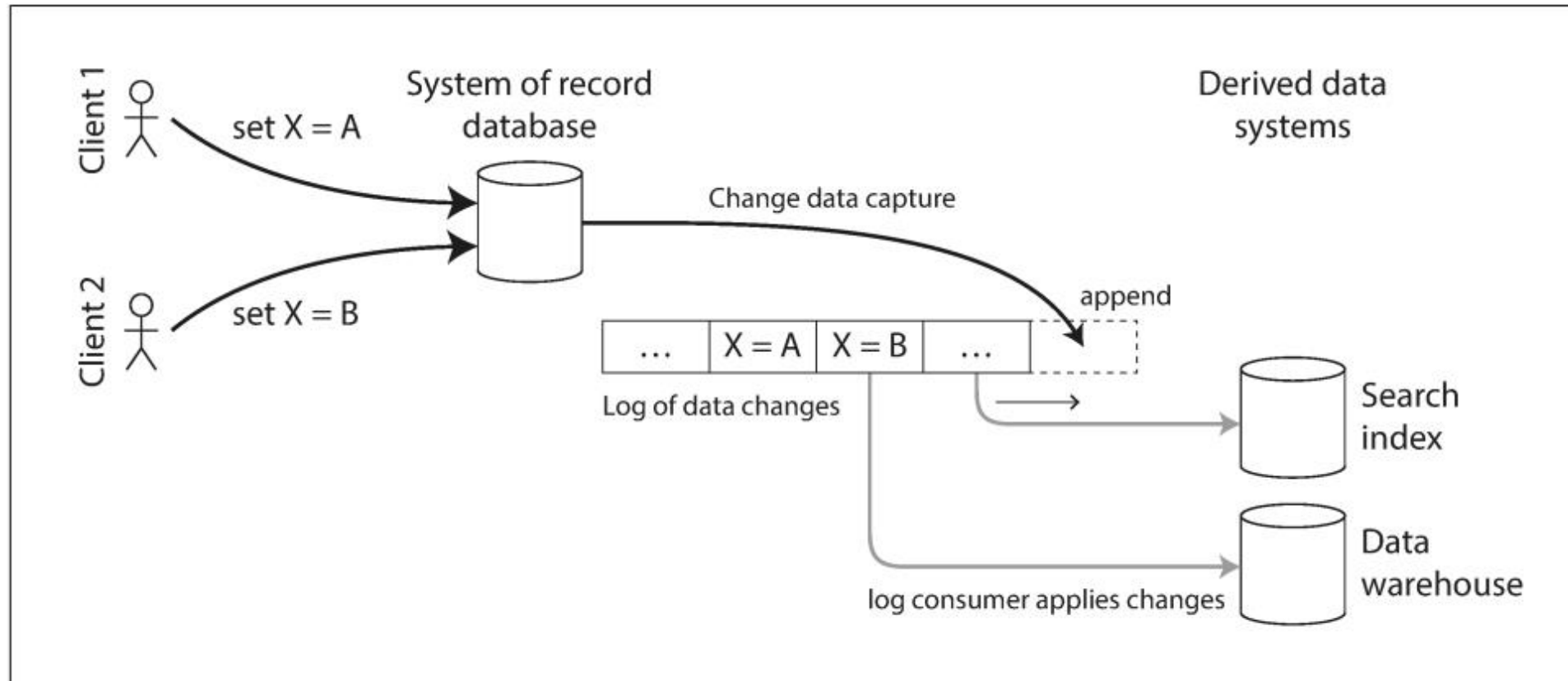


Figure 11-5. Taking data in the order it was written to one database, and applying the changes to other systems in the same order.

Implementing change data capture

- Like one leader
- Capture changes
 - *SQL Trigger*
 - *Read logical replication log*

Reconstruct the state from logs

- Initial snapshot
- Log compaction

API Support for change streams

- Some databases like RethinkDB
- Kafka Connect

Event sourcing

- A technique in Domain Driven Design community
- Event Sourcing Vs. Change data capture
- Event as source of truth
 - *Then transform it into application state*
- Event vs Commands

PROCESSING STREAMS

Options in Stream Processing

- You can take the data in the events and write it to a database, cache, search index, or similar storage system, from where it can then be queried by other clients.
- You can push the events to users in some way, for example by sending email alerts or push notifications
- You can process one or more input streams to produce one or more output streams. Streams may go through a pipeline consisting of several such processing stages before they eventually end up at an output (option 1 or 2).

Stream analytics

Reasoning About Time

- the average over the last five minutes.
- the last five minutes
- Event time versus processing time

Reasoning About Time

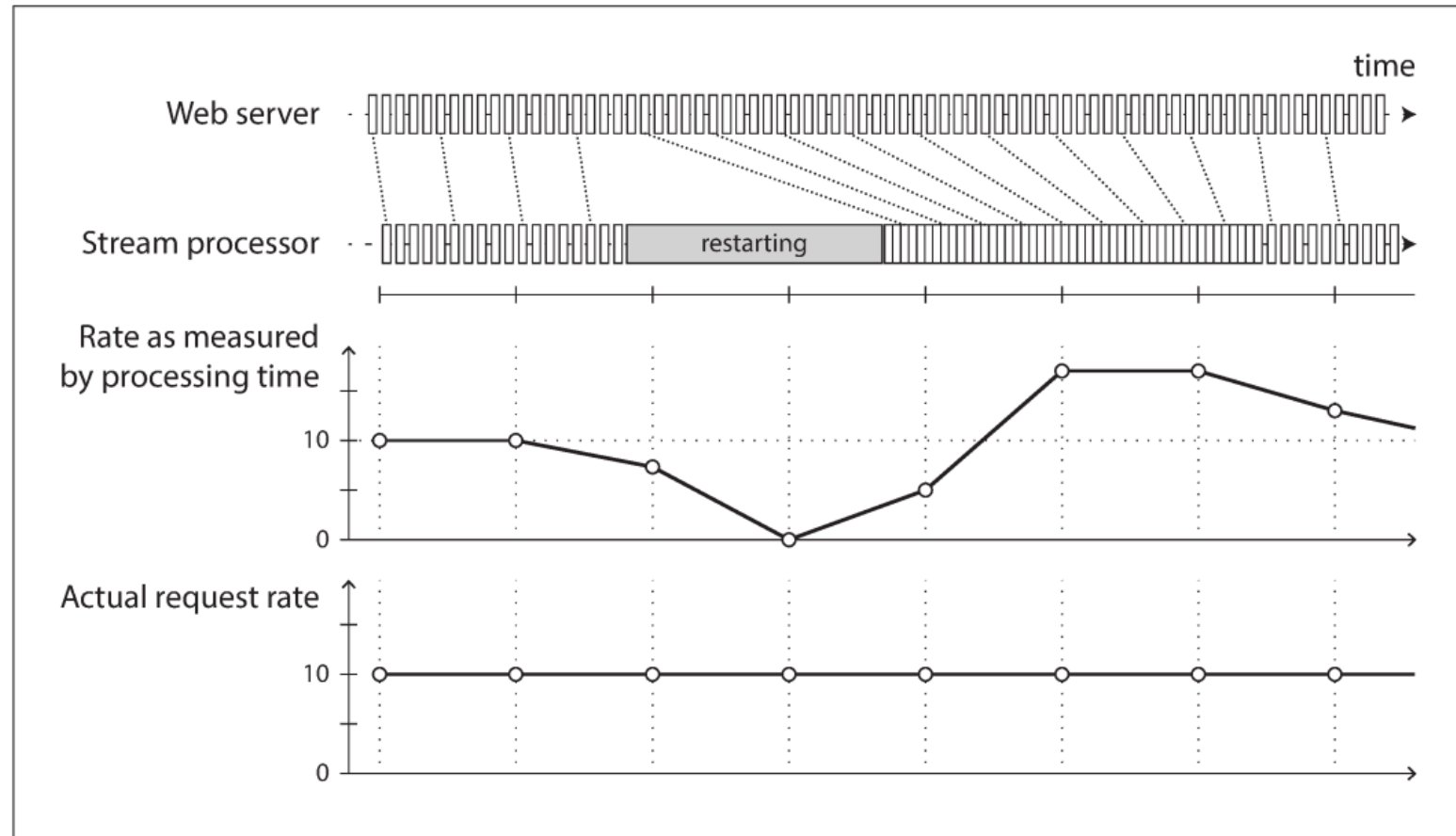


Figure 11-7. Windowing by processing time introduces artifacts due to variations in processing rate.

Calculating events on a given window?

- you can never be sure when you have received all of the events for a particular window, or whether there are some events still to come.

Calculating events on a given window?

■ Options:

- *Ignore the straggler events, as they are probably a small percentage of events in normal circumstances. You can track the number of dropped events as a metric, and alert if you start dropping a significant amount of data.*
- *Publish a correction, an updated value for the window with stragglers included. You may also need to retract the previous output.*

Whose clock should be used?

- To adjust for incorrect device clocks, one approach is to log three timestamps:
 - *The time at which the event occurred, according to the device clock*
 - *The time at which the event was sent to the server, according to the device clock*
 - *The time at which the event was received by the server, according to the server clock*

Types of window

- Tumbling window
- Hopping window
- Sliding window
- Session window

Stream Joins

- Stream-stream join (window join)
- Stream-table join (stream enrichment)
- Table-table join (materialized view maintenance)

Problem: Time-dependence of joins

Fault Tolerance

- What should we do when streaming job failed?

Fault Tolerance

- Microbatching and checkpointing

Fault Tolerance - External side effects

- Atomic commit and distributed transaction

Fault Tolerance - External side effects

- Idempotence

Fault Tolerance

- Problem: Rebuilding state after a failure