Data Streams

Erica Lee & Emily Yeh

Databases, Spring 2019

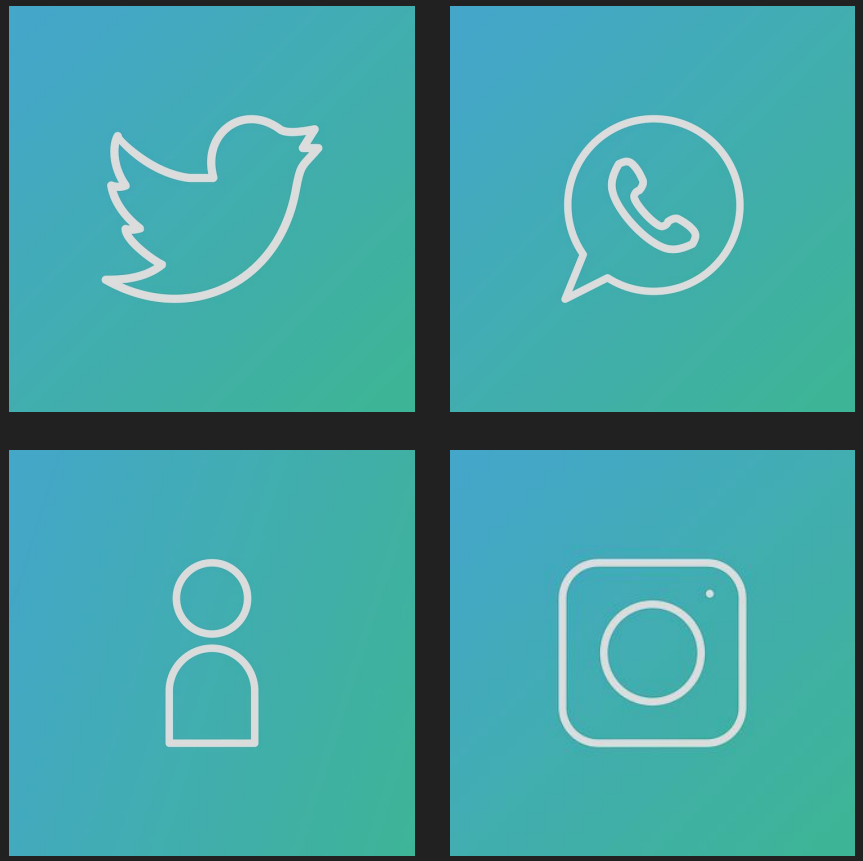
What are data streams?

First, a quick recap of the history of data...

Data used to be **static**.

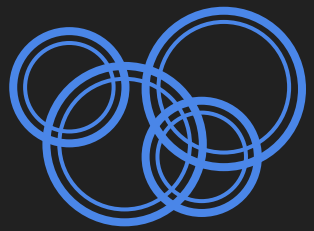
Old-school data analysis involved static sets of data and executing single queries on this data.

Then, people invented social media...

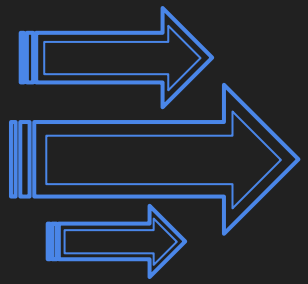


With the invention of social media, the way we used data changed drastically.

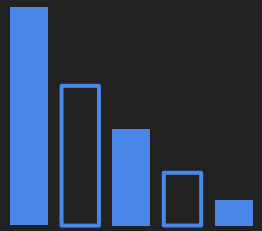
This introduced some interesting new challenges:



Data generated continuously by multiple sources

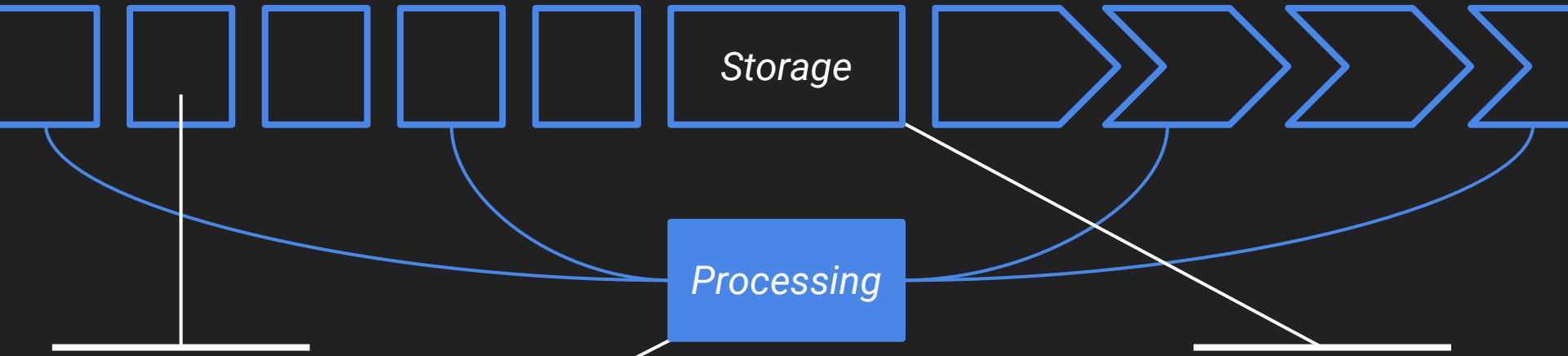


Highly frequent writes and fast reads



Methods for analyzing data quickly and meaningfully

Data streams generally have two layers:



Typically uses time-based indexing

Continuously updates results while ingesting new data

Needs to be strongly consistent

How are data streams managed?

With a **data stream management system (DSMS).**

DBMS and DSMS are similar in that they both are systems for managing data.

That's pretty much where the similarities end, though.

DBMS Data	DSMS Data
Persistent data	Volatile data
Generally low update rate	Often very high update rate
Assumes exact data	Assumes outdated and inaccurate data
Time doesn’t matter that much	Real-time requirements and constraints

DBMS: Random Access

Can read or write anywhere in a file

Data are spread apart

(Theoretically) infinite storage space

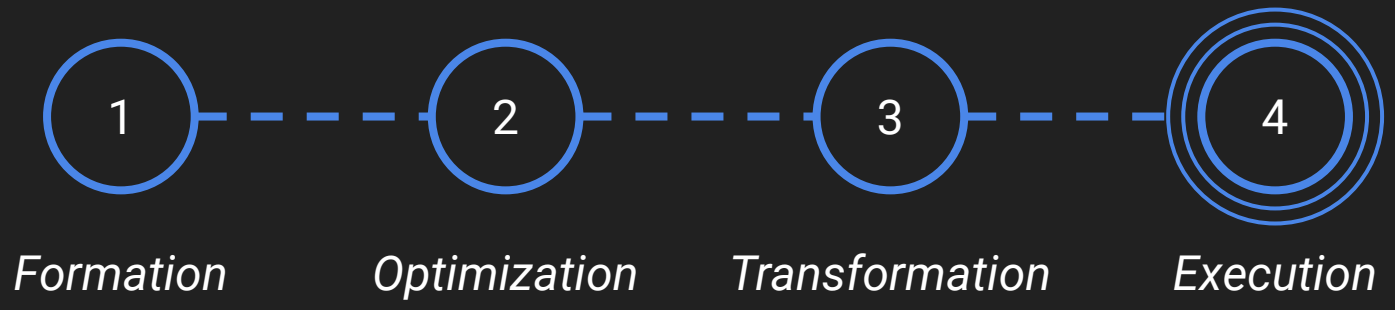
DSMS: Sequential Access

Reads or writes sequentially

Data are grouped together

Storage space must be limited

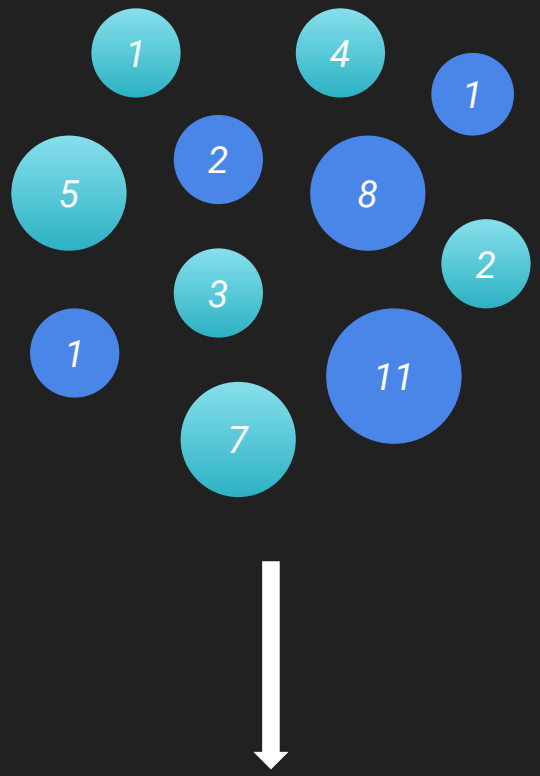
A DSMS query is **continuous.**



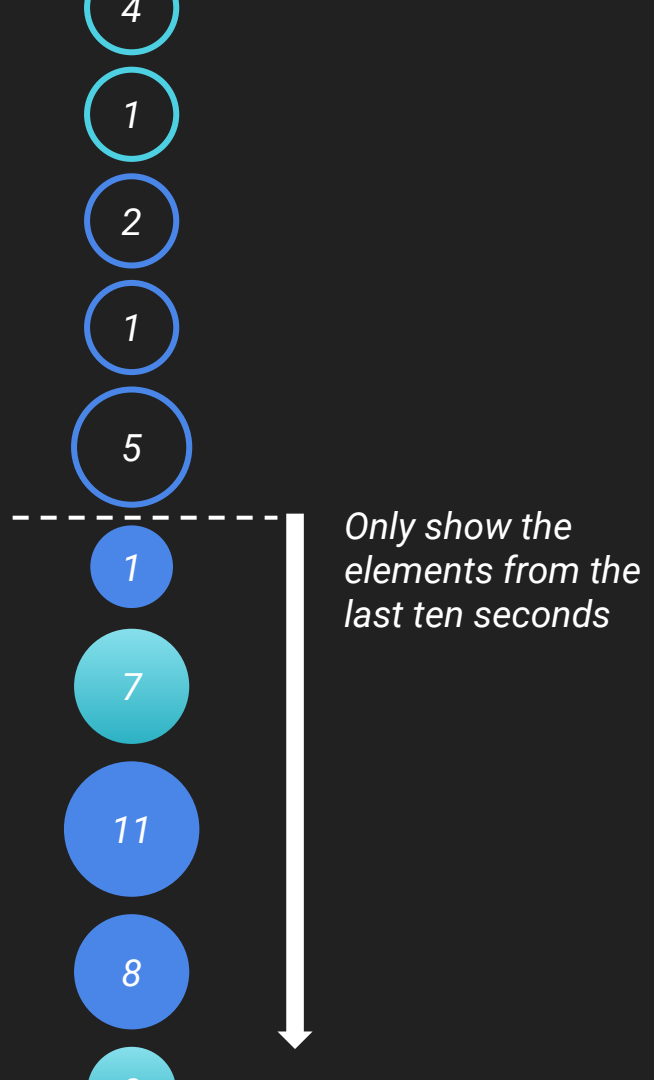
How do you process a data stream?

Synopses:

Maintain only a synopsis of the data (as opposed to all of the raw data), thereby drastically reducing the amount of data that needs to be stored.



Number of elements: 11
Average: 4.09



Windows:

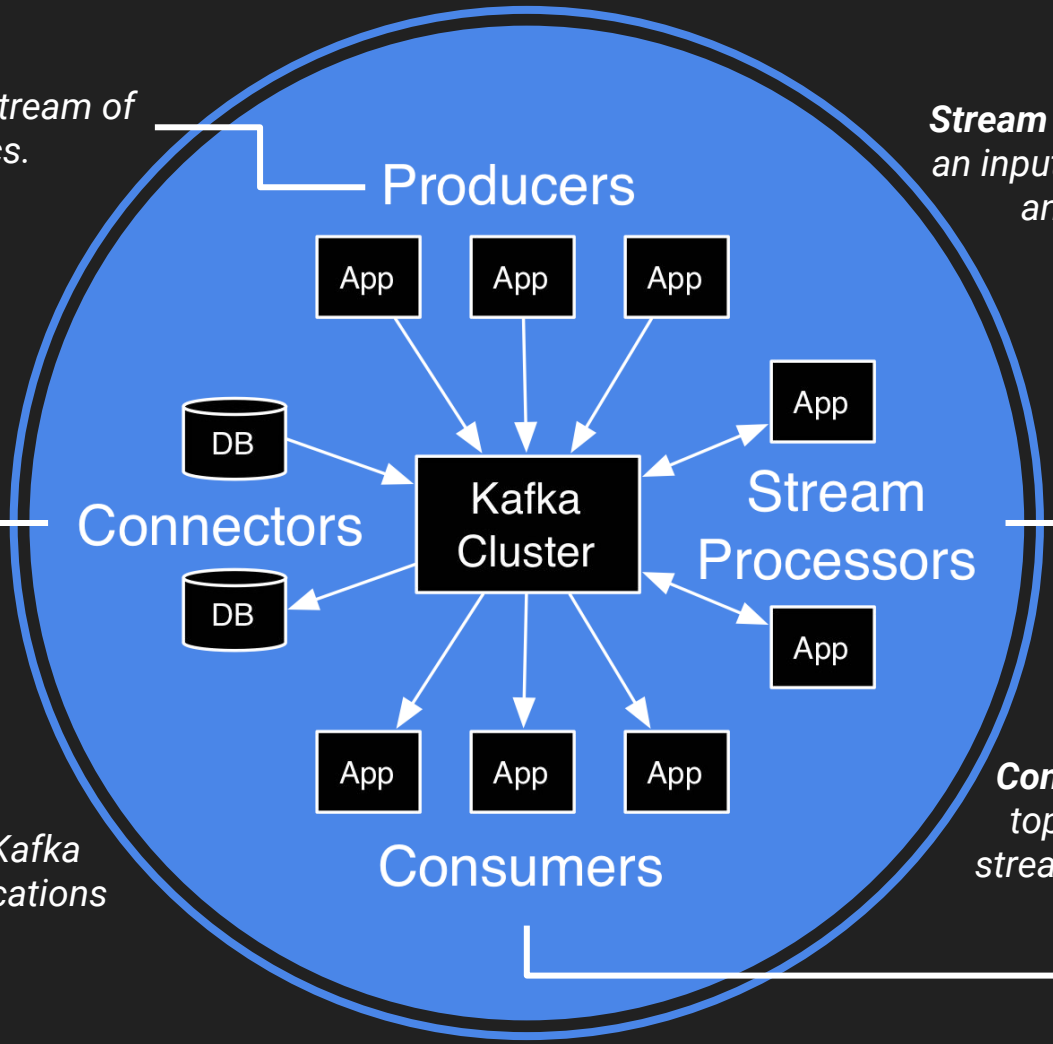
Under the assumption that only recent data are relevant, show only a part of the data, e.g. the last ten data stream elements or the data from the last ten seconds.

Let’s see an example of real data stream processing software.

Apache Kafka is an *open-source stream-processing* software platform.

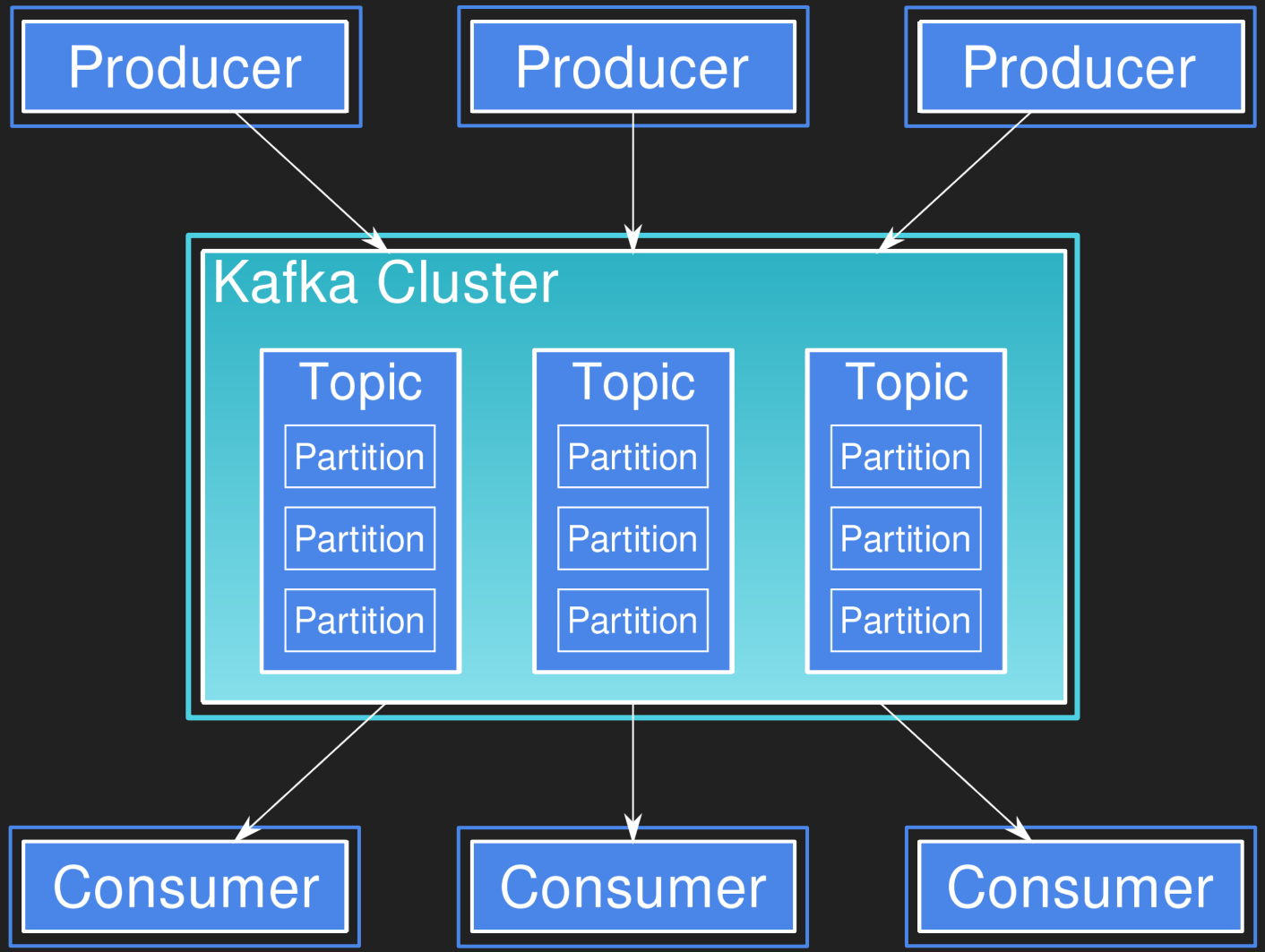
Producers *publish a stream of records to topics.*

Stream processors *consume an input stream and produce an output stream.*



Connectors *connect Kafka topics to existing applications or data systems.*

Consumers *subscribe to topics and process the streams produced to them.*


Producer
Producer
Producer
Kafka Cluster
Topic
Partition
Partition
Partition
Topic
Partition
Partition
Partition
Topic
Partition
Partition
Partition
Consumer
Consumer
Consumer

Demo time?

Thanks for listening!