Stream-based Databases

DATA 604

Leanne Wu

lewu@ucalgary.ca

Department of Computer Science

UNIVERSITY OF CALGARY

What happens when your data doesn’t end?

Streaming databases

Ways to look at temporal data arriving continuously:

- Time series
  - Traditional relational data
  - Unstructured data
- Streaming databases
  - Can manage uncertainty and inaccuracy in data
  - Can process large amounts of data in real-timeUsed

Used for applications where real-time monitoring is desirable (stock markets, patient care, network and computer security, sports analytics, power grid monitoring, marketing and advertising ...)

Existing implementations

Early research prototypes:

- Aurora, Borealis
- NiagaraCQ, TelegraphCQ
- PSoup
- NILE

Implementations used in practice:

- Apache Spark Streaming
- Apache Kafka, Apache Storm
- PipelineDB (Confluent)
- Amazon Kinesis

What is stream-based processing good for?

- Simple data monitoring
- Incomplete data (individual records are continuing to arrive)
- Rolling averages (or other aggregates)

Continuous Queries

Snapshot query:

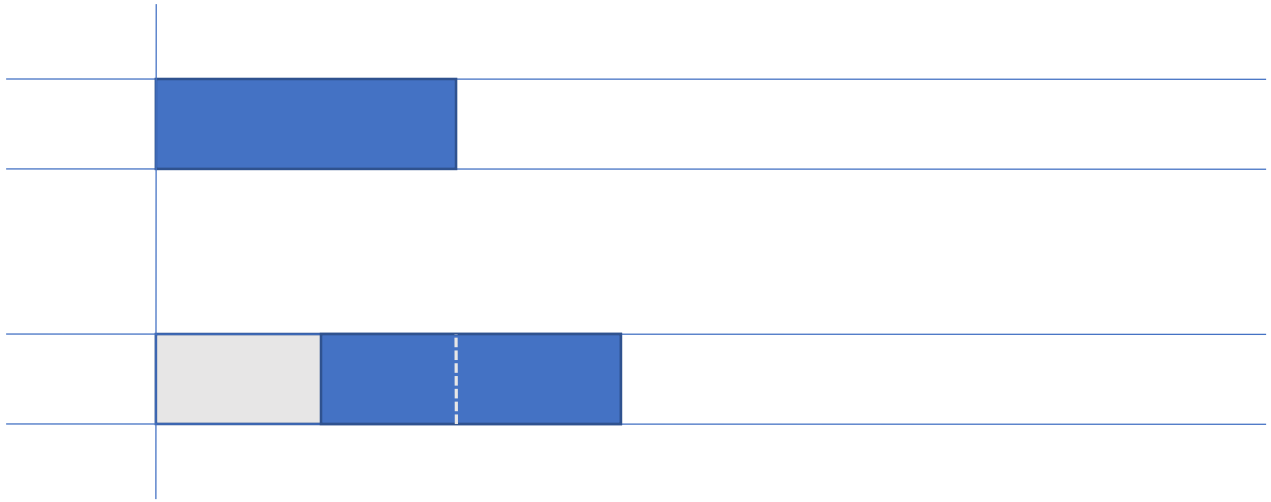
- “Give me the highest price of stock X over the trading day”
- “Find me the rain gauge which measured the most rain in June 2013”

Continuous query:

- “Give me a list of IPs which originate more than X requests in the last five minutes”
- “Alert Emergency Services if we receive more than 10 cm of rain in half an hour”

Sliding Window Queries

Continuous queries are called sliding window queries if the timespan over which the query exists shifts as time progresses



Query processing for streams

- Maintaining a sliding window can be complex, especially for aggregation operations
- Consider a sum operation:
  - $s_1 = a_1 + a_2 + a_3 + a_4$
- A sliding window might maintain a window for a set number of operations, or a set period of time
  - $s_2 = a_2 + a_3 + a_4 + a_5$
  - $s_3 = a_3 + a_4$
- The question is how you calculate s_2 or s_3 from s_1…