Trường: ĐH Bách Khoa Tp. Hồ Chí Minh
Khoa: Khoa học máy tính

# Streaming & Data Stream

GIẢNG VIÊN: PGS.TS THOẠI NAM

NHÓM:

VÕ NGUYÊN THÀNH 1570746

NGUYỄN LÊ QUANG NHẬT 1570741

BÙI THANH PHONG 1570743

# OUTLINE

- Introduction
- Data Stream Models
- Query processing systems

# Introduction

- We do not just have people who are entering information into a computer. Instead, we have computers entering data into each other.
- Nowadays, there are applications in which the data are better modeled not as persistent tables but rather as transient data streams.
- Examples of such applications include network monitoring, web mining, sensor networks, telecommunications data management, and financial applications.
- We are in presence of sources of data produced continuously at high-speed.

# An Illustrative Problem

- Finding the maximum value (MAX) or the minimum value (MIN) in a sliding window over a sequence of numbers.
- When we can store in memory all the elements of the sliding window, the problem is trivial and we can find the exact solution.
- When the size of the sliding window is greater than the available memory, there is no exact solution.
- Whatever the window size, the first element in the window is always the maximum. As the sliding window moves, the exact answer requires maintaining all the elements in memory.

# Data Stream Models

- Data streams can be seen as stochastic processes in which events occur continuously and independently from each another.
- Some relevant differences include:
  - The data elements in the stream arrive on-line.
  - The system has no control over the order in which data elements arrive, either within a data stream or across data streams.
  - Data streams are potentially unbound in size.

# Differences between DBMS vs DSMS

| Database management system (DBMS) | Data stream management system (DSMS) |
|---|---|
| Persistent data (relations) | Volatile data streams |
| Random access | Sequential access |
| One-time queries | Continuous queries |
| (theoretically) unlimited secondary storage | limited main memory |
| Only the current state is relevant | Consideration of the order of the input |
| relatively low update rate | potentially extremely high update rate |
| Little or no time requirements | Real-time requirements |
| Assumes exact data | Assumes outdated/inaccurate data |
| Plannable query processing | Variable data arrival and data characteristics |

# Data Stream Models

- Time-Series Model:
  - This naturally models time-series data streams. Note that this model poses a severe limitation on the update stream, essentially prohibiting updates from changing past (lower-index) entries.
  - Ex: The series of measurements from a temperature sensor or the volume of stock trades over time.
- Cash-Register Model:
  - We only allow increments to the entries but multiple updates can increment a given entry over the stream.
  - Ex: Streams monitoring the total packets exchanged between two IP addresses or the collection of IP addresses accessing a web server.
- Turnstile Model:
  - In this, most general, streaming model, no restriction is imposed, thus, we have a fully dynamic situation, where items can be continuously inserted and deleted from the stream.
  - Ex: Monitoring active IP network connections.

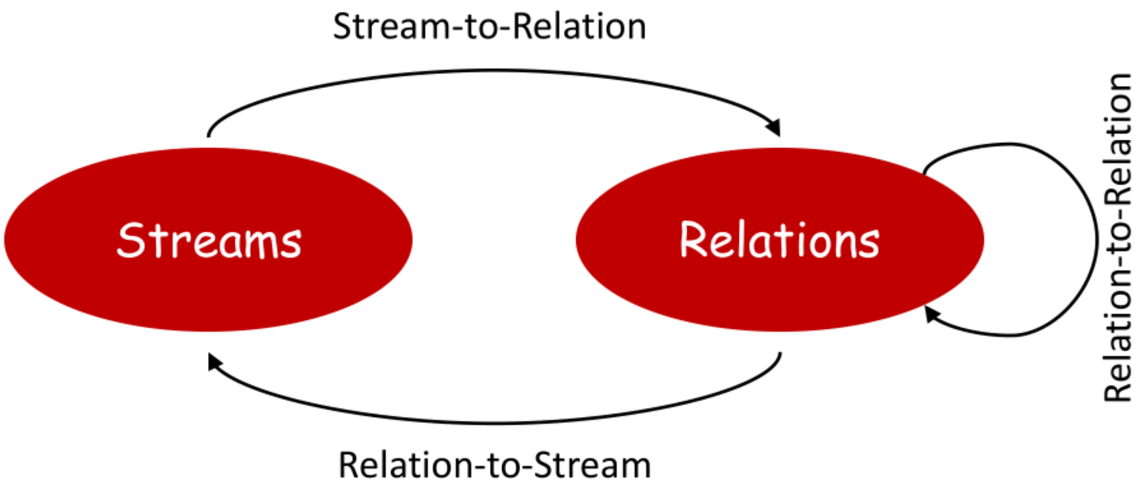
# Incorporating Recency: Time-Decayed and Windowed Streams

- In many applications, it is important to be able to downgrade the importance (or, weight) of older items in the streaming computation.
- For instance, in the statistical analysis of trends or patterns over financial data streams, data that is more than a few weeks old might naturally be considered "stale" and irrelevant.
- Various time-decay models have been proposed for streaming data, with the key differentiation lying in the relationship between an update's weight and its age.
- The sliding-window model: time-based (e.g., updates seen over the last W time units) or count-based (e.g., the last W updates). The key limiting factor in this streaming model is, naturally, the size of the window.
- The goal is to design query processing techniques that have space/time requirements significantly sublinear (typically, poly-logarithmic).

# STREAM CQL: Continuous Query Language

- SQL for Relation-to-Relation operations
- Additionally:
  - "Stream" as a new data type (in addition to "Relation")
  - Continuous instead of one-time query semantics
  - Stream-to-Relation operations:
    - Window specifications derived from SQL-99
  - Relation-to-Stream operations:
    - Three special operators: Istream, Dstream, Rstream
  - Simple sampling operations on streams
- No Stream-to-Stream operations

# CQL: Mappings between Streams and Relations




- Stream-to-Stream = Stream-to-Relation + Relation-to-Stream

# Aurora SQuAl: Stream Query Algebra

- Queries are represented with data-flow diagrams consisting of operators.
- Order-agnostic operators:
  - Filter, Map, Union
- Order-sensitive operators:
  - BSort, Aggregate, Join, Resample
- All operators are Stream-to-Stream

# Query processing system

- What is Database Management System (DBMS)?
  - What is database?
- What is Data Stream Management System (DSMS)?
  - What is data stream?
- Limit of data stream model
- Differences between  DBMS vs DSMS?
- Continuous Query Language - CQL

# Database Management System

- A Database is an organized collection of data
  - There are a lot of Database Models (Hierarchical, Relational, Semantic, XML, Object Oriented, NoSQL, …)
  - The most popular database systems since the 1980s have all supported the relational model as represented by the SQL language
- A Database Management System is a collection of programs that enables you to store, modify, and extract information from a database.

# Data Stream Management System

***What is Data stream?***

- Large data volume, likely structured, arriving at a very high rate.
- Definition (Golab and Ozsu, 2003):
    - *"A data stream is a real-time, continuous, ordered (implicitly by arrival time of explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor it is feasible to locally store a stream in its entirety"*

# Data Stream Management System

***What is Data stream?***

- Large data volume, likely structured, arriving at a very high rate.
- Definition (Golab and Ozsu, 2003):
  - *"A data stream is a real-time, continuous, ordered (implicitly by arrival time of explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor it is feasible to locally store a stream in its entirety"*

# Data Stream Management System

- A DSMS is a computer program that permits to manage continuous data streams (assumed infinite).
- Data received from a DSMS is moving at high pace.
- Queries are continuous (registered once, observed “forever”).
- Answer to queries in (nearly) real-time required.

# Limits of Data stream

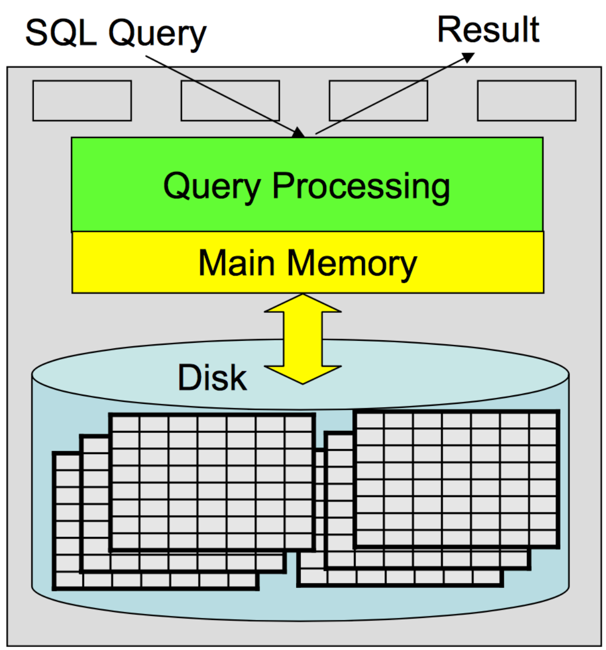
- Stream data is unbounded.
  - Memory is not unbounded, no way to store entire stream
- Query answer…
  - Is not exact, we can only approximate
- To compute query results.
  - Need to device algorithm with little memory consumption

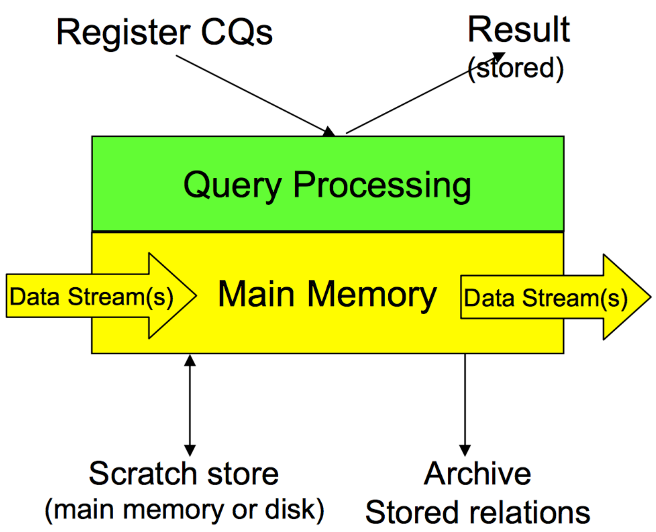
# Solutions

- *Sliding Window:* evaluate the query not over the entire past history of the data streams, but rather only over sliding windows of recent data from the streams.
- *Synopses:* maintain only a synopsis of the data selecting random data points called sampling to summarization using histograms, wavelets or sketching (both methods cannot reflect the data accurately).

# Differences between DBMS vs DSMS

**Traditional DBS** **DSMS**



# Continuous Query Language - CQL

# Data stream (Example)

- Monitoring of highway traffic:

**PosSpeedStr(vehicleId,speed,xPos,dir,hwy)**

# Continuous Queries

- In contrast to ad-hoc, single time queries in (relational) DBMS.

- Queries over Streams are considered continuous: registered once, run "forever".

- For instance:

– Compute average temperature.

  - *How to compute average values over an infinite stream? Block forever?*

– Select all orders of stock "Apple" with quantity larger than 100.

# Sliding Window Concept

- Focus attention to latest values of stream.
- Allows computation of aggregates

# Sliding Window Aggregates

- Output average for each window when it slides.
- Here:
  - 17.7°C.
  - 26.3°C
  - 19.1°C

18.3°C
13.5°C
27.0°C
11.6°C
29.6°C
39.7°C
24.2°C
11.5°C
12.7°C
27.9°C
....

# Overview DSMS

- STREAM (Stanford University), Aurora (Brandeis/ Brown/MIT), TelegraphCQ (UC Berkely), Cayuga (Cornell), PIPES (Uni Marburg), ...
- Large interest also from companies/startups: Oracle MicrosoU, IBM, Streambase.
- Lately open-source product for big data distributed streams: Yahoo! S4, Twi]er Storm (will see in detail later)

# STREAM

- Stanford Stream Data Manager
- “General purpose” DSMS for streams and stored data.
- Declarative query language to phrase continuous queries (SQL like).

# Continuous Query Language – CQL

SQL with:

- Streams
- Windows
- New semantics (stream)

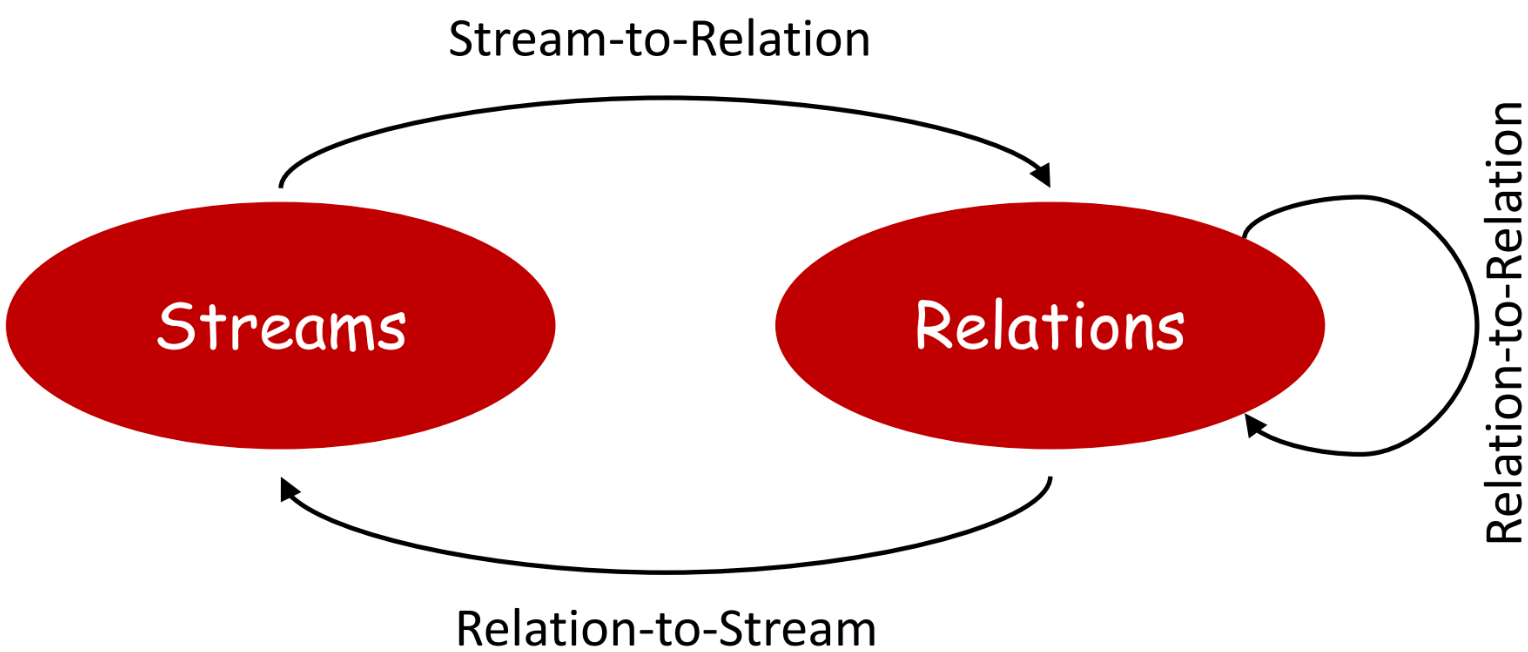
– Three relation-to-stream operators: *Istream*, *Dstream*, *Rstream*

- Sampling

# CQL: Relations and Streams

- T: discrete, ordered time domain
- A relation R is a mapping from time T to bag of tuples belonging to the schema of R.
- That is, R(t) varies over time
- A stream S is a set of (tuple, timestamp) elements

# CQL: Mappings between Streams and Relations



# Streams → Relations

Ways to construct these windows "[W]"

– Time-based

– Tuple-based

– Partitioned

# Time-Based Window

S [Range T]

– S [Now]

– S [Range Unbounded]

Examples:

- PosSpeedStr [**RANGE** 30 Seconds]
- PosSpeedStr [**NOW**]
- PosSpeedStr [**RANGE** Unbounded]

# Tuple-based Window

S [Rows N]

– [Rows Unbounded]

Examples:

- PosSpeedStr [**ROWS** 1]
- PosSpeedStr [**ROWS** 3]

# Partitioned Windows

S [Partition By A1,...,Ak Rows N]

Examples:

- PosSpeedStr [**PARTITION BY** vehicleId **ROWS** 1]

# Relation → Relation

- With previous window transform we get a relation, now we can apply.
- Any query expressed in SQL
– just that deal now with time-varying relations

Examples:

- **SELECT distinct** vehicleId **FROM** PosSpeedStr [**RANGE** 30 Seconds]

# Relation → Stream

- Istream(R) contains a stream element (r,t) whenever r in R(t) \ R(t-1) "**I**nsert stream"
- Dstream(R) contains a stream element (r,t) whenever r in R(t-1) \ R(t) "**D**elete stream"
- Rstream(R) contains a stream element (r,t) whenever r in R(t) "**R**elation stream"

# Istream, Dstream, and Rstream

- Istream(R): contains all tuples in R that are new within the last time period, i.e., insert stream
- Dstream(R): contains all tuples in R which where in the stream before the last period (and not anymore in now), i.e., delete stream
- Rstream(R): contains all tuples in R

# References


- “The CQL Continuous Query Language: Semantic Foundations and Query Execution”, A. Arasu, S. Babu, J. Widom, VLDB Journal, 15(2), 2006.
- “Stanford Data Stream Management System”, A. Arasu et al., book chapter, http://dbpubs.stanford.edu/pub/2004-20.
- https://docs.oracle.com/cd/E16764_01/doc.1111/e12048/operators.htm
- http://infolab.stanford.edu/~widom/cql-talk.pdf
- Minos Garofalakis, Johannes Gehrke, Rajeev Rastogi, Data-Centric Systems and Applications
- Joao Gama, Knowledge Discovery from Data Streams

# Thank you for listening!