Intro to Stream Processing
Data Processing so far ...
Sensor Data Example

Motivation

Data Streams

Reservoir Sampling
Sensor Data Example

one million 4 byte reals every 100 ms

3.5 Tb per day
Sensor Data Example

Stream of large unbounded data

too large for memory
too high latency for disk

We need real time processing!
Sensor Data Example

Process data stream directly
Data Streams
What is a Data Stream?

**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.
What is a Data Stream?

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.

- continuous and sequential input
- typically unpredictable input rate
- can be large amounts of data
- not error free
Data Stream Applications

- Online, real time processing
- Event detection and reaction
- Aggregation
- Approximation
Data Stream Example

Stock monitoring
Data Stream Example

Stock monitoring
Website traffic monitoring
Data Stream Example

Stock monitoring
Website traffic monitoring
Network management
Data Stream Example

Stock monitoring
Website traffic monitoring
Network management
Highway traffic
Data Stream Characteristics

- All items have the same structure. For example a tuple or object: (sender, recipient, text body)
Data Stream Characteristics

- All items have the same structure. For example a tuple or object: (sender, recipient, text body)
- Timestamps: explicite vs. implicite, physical vs. logical
Data Streams

- Continuous sequences of data elements that are typically:
  - **Push-based** (like in publish/subscribe systems)
  - **Ordered** (e.g., by arrival time, or by explicit timestamps)
  - **Rapid** (e.g., ~ millions of messages/sec in market data)
  - **Potentially unbounded** (may have no (known) end)
  - **Time-sensitive** (real-time events, latency-critical)
  - **Time-varying** (in content and speed)
  - **Unpredictable** (autonomous data sources)
Example Applications

• Financial Services

Example:
- Trades(time, symbol, price, volume)

Typical Applications:
- Algorithmic Trading
- Foreign Exchange
- Fraud Detection
- Compliance Checking
Example Applications

• System and Network Monitoring

Example:
- Connections(time, srcIP, destIP, destPort, status)

Typical Applications:
- Server load monitoring
- Network traffic monitoring
- Detecting security attacks
  - Denial of Service
  - Intrusion
Example Applications

• Sensor-based Monitoring

Example:
- CarPositions(time, id, speed, position)

Typical Applications:
- Monitoring congested roads
- Route planning
- Rule violations
- Tolling
A Paradigm Shift in Data Processing Model

Traditional Data Management

Data Base

Query → DBMS → Answer

Data Stream Management

Query Base

Data → DSMS → Answer

Query
DBMS vs. DSMS

- Persistent relations
- Read-intensive
- One-time queries
- Random access
- Access plan determined by query processor and physical DB design

- Transient streams
- Update-intensive (mostly append-only)
- Continuous queries (a.k.a., long-running, standing, or persistent queries)
- Sequential access
- Unpredictable data characteristics and arrival patterns