DBMS = Data Base Management System

DSMS = Data Stream Management System

- A **data stream** is a (potentially unbounded) sequence of tuples
- **Transactional** data streams: log interactions between entities
  - Credit card: purchases by consumers from merchants
  - Telecommunications: phone calls by callers to dialed parties
  - Web: accesses by clients of resources at servers
- **Measurement** data streams: monitor evolution of entity states
  - IP network: traffic at router interfaces
  - Sensor networks: physical phenomena, road traffic
  - Earth climate: temperature, moisture at weather stations

- Two recent developments: application- and technology-driven
  - Need for sophisticated near-real time queries/analyses
  - Massive data volumes of transactions and measurements
- Now need sophisticated near-real time queries/analyses
  - AT&T: fraud detection on call detail tuple streams
  - NOAA: tornado detection using weather radar data

- **Business Challenge 1:** AT&T wanted to track calling pattern of each of ~100M callers, and raise real-time fraud alerts

- Previous Approach: Handwritten, optimized C code, computing
evolving **signatures** for each customer, looking for variations
- Issues: Signature computation is I/O intensive, often modified
- Solution: Using Hancock domain-specific language
    - Abstract logical/physical streams and signatures
    - Express I/O and CPU efficient signature programs cleanly

- Lesson: Essential to consider I/O issues for data streams

- **Business Challenge 2:** AT&T IP customer wanted to accurately monitor peer-to-peer (P2P) traffic evolution within its network

- Previous Approach: Determine P2P traffic volumes using TCP port number found in Netflow data
- Issues: P2P traffic might not use known P2P port numbers
- Solution: Using Gigascope SQL-based DSMS
  - Search for P2P related keywords within each TCP datagram
  - Identified 3 times more traffic as P2P than using Netflow

- Lesson: Essential to query massive volume data streams

- **Business Challenge 3:** AT&T IP customer wanted to monitor latency observed by clients to find performance problems

- Previous Approach: Measure latency at "active clients" that
establish network connections with servers
- Issues: Use of "active clients" is not very representative
- Solution: Using Gigascope SQL-based DSMS
  - Track TCP synchronization and acknowledgement packets
  - Report round trip time statistics: latency

- Lesson: Essential to correlate multiple data streams

Data Stream Systems

- Resource (memory, per-tuple computation) limited
- Reasonably complex, near real time, query processing
- Useful to identify what data to populate in database

Database Systems

- Resource (memory, disk, per-tuple computation) rich
- Extremely sophisticated query processing, analyses
- Useful to audit query results of data stream system

Database Systems	**Data Stream Systems**
■ Model: persistent relations	■ Model: transient relations
■ Relation: tuple set/bag	■ Relation: tuple sequence
■ Data Update: modifications	■ Data Update: appends
■ Query: transient	■ Query: persistent
■ Query Answer: exact	■ Query Answer: approximate
■ Query Evaluation: arbitrary	■ Query Evaluation: one pass
■ Query Plan: fixed	■ Query Plan: adaptive

Stream Query Languages

- SQL-like proposals suitably extended for a stream environment:
  - Composable SQL operators
  - Queries reference/produce relations or streams
  - GSQL: SQL used by Gigascope
  - CQL: SQL used by STREAM
  - Extensions of PostgreSQL
- UDA-SQL: Monotonic sequence based queries

Windows

- Mechanism for extracting a finite relation from an infinite stream
- Various window proposals for restricting operator scope
  - Windows based on ordering attributes (e.g., time)
  - Windows based on tuple counts
  - Windows based on explicit markers (e.g., punctuations)

Prototype systems

- Aurora (Brandeis, Brown, MIT)
- Gigascope (AT&T)
- Hancock (AT&T)
- Nile (Purdue)
- STREAM (Stanford)
- TelegraphCQ (Berkeley)
- Esper
- StreamCruncher