Stream Data Processing and Querying

CPS 196.3
12/01/2003

Data Streams

- Traditional database systems have been designed for stable, static data
- Traditional systems cannot run queries over multiple, continuous, high volume, and time-varying data streams
  - i.e. sensor data

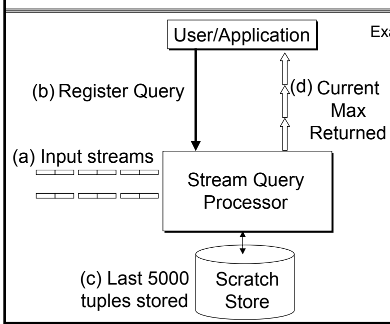
Four Major Issues

- Model Issues
- Language Issues
- System Issues
- Algorithm Issues

Model Issues

- DSMS is born (Data Stream Management System)
- DSMS needs to be capable of running multiple queries over multiple input streams
- DSMS also needs to be capable of running traditional queries over a traditional data set

STREAM Model



Example: Find the max price of last 5000 Simpson's dolls sold

(a) Sales records of a toy company
(b) User/Application queries max price out of last 5000 Simpson's dolls sold
(c) Processor stores prices of last 5000 Simpson's dolls in Scratch Store
(d) Processor returns the current max to user/Application

Language Issues

- A new query language is required to facilitate stream processing
- Case Study:
  - STREAM and their CQL (continuous query language)
    - Relational query language to start with
    - Window specification language
    - Relation-to-stream operators

Continuous Query Language

- Example of a "windowed" query
  - SELECT max(price)
    FROM
    (SELECT * FROM Sales
     WHERE description = 'Simpsons Doll')
    [Rows 5000];
  - SELECT max(price)
    FROM Sales [Range 10 minutes];
- What happens to blocking operators?
  - SELECT max(price)
    FROM Sales;
    - Only a running max needs to be maintained

System Issues

- Comprehensive DSMS query processor
- Broad suite of operators + synopses (state)
  - Join + input tuples seen so far in window
- Sophisticated "developer's workbench" interface
  - Submit queries in extended SQL or algebra
  - Submit or edit query plans in XML or GUI
  - Query plan execution visualizer
  - On-the-fly modification of memory allocation, scheduling policies, etc.

Algorithm Issues

Traditional DBMS

- Assumption that all data is already on disk
- Multiple passes can be run over current data
- Algorithms function to minimize I/O accesses to data

Continuous DSMS

- Data is NOT stored on disk
- Therefore, multiple passes cannot be run over current data
- Limited amount of memory for large quantities of data
- Algorithms maximize capture of data

References

- "STREAM: The Stanford Stream Data Manager", The STREAM Group
- "Stanford Data Streams Research Project", Professor Rajeev Motwani and Professor Jennifer Widom
- "Continuous Data Stream Processing", Roger Zimmermann, University of Southern California