AURORA: A Data Stream Management System


A Brandeis, Brown, MIT Production (http://www.cs.brown.edu/research/aurora)

Stream-based Monitoring Applications

- Monitoring applications:
  - A new class of apps that require timely processing of large volumes of continuous data streams.
  - E.g., tracking/monitoring services, financial analysis, sensor networks.
  - Traditional DB models are inherently ill suited for these apps.
    - Pull vs push-based architecture
    - Real-time response requirements
    - Time-series data
    - Approximate answers
- Aurora is a data-stream processing system that is being designed and implemented to support stream-based monitoring applications.

Quality-of-Service

- Per-app QoS specs describe the utility of “imperfect” query results:
  - Delay-based (specify utility of “late” results)
  - Delivery-based (specify utility of “partial” results)
- QoS drives all resource data management decisions.
- CPU scheduling, storage management, and load shedding, ...

Operator Scheduling

- Goal: “Minimize per-tuple processing overhead”
- Default Operation: 

Run-Time Architecture

- Applications provide:
  - Queries over input data streams
  - Quality-of-Service (QoS) specifications
    - (specifies the utility of partial or late results)

Load Shedding

- Drop access load (i.e., tuples) when the system gets overloaded.
- Insert drop operators ( ) such that excess load is shed with minimum drop in the perceived QoS.
- Two types of drop operators:
  - Randomized Drop
  - Semantic Drop

Graphical User Interface

- Boxes → operators
- arcs → tuple queues
- queries can access historical data through connection points.

Distributed Processing: Aurora

- Data Stream Sources
- Aurora
- Monitoring Apps