Monitoring Streams
A New Class of Data Management Applications
Carney et. al.
Presentation: Kenji Okuma
Discussion: Navjot Singh

Outline
• Introduction
• Aurora System Model
• Aurora Optimization
• Run-Time Operation
• Conclusion

Introduction: Traditional DBMSs
1. Passive repository: Human-Active, DBMS-Passive (HADP) model
2. The current of state of the data is important: Previous data needs to be extracted from the log
3. Triggers and alerts as second-class citizens
4. Perfect synchronization of data elements and exact query answers
5. No real-time services from applications

Introduction: Monitoring apps
Monitoring applications are applications that monitor continuous streams of data
Target Applications: military financial analysis tracking other real-time applications…

Introduction: Monitoring apps
1. Active repository: DBMS-Active, Human-Passive (DAHP) model
2. History of the data is important: Not only the current state but also the previous history
3. Triggers and alerts as the first-class citizens
4. Missing or imprecise data, and approximate query answers
5. Real-time services required by applications

Car Navigation System
1. Data (e.g., the location of the car) comes from external sources
2. History of the data is required (e.g., display a trajectory of your car in the past 20 minutes)
3. Trigger and alert oriented: an alert for the driver when the car is approaching to an intersection
4. The location of the car is not always perfectly transmitted due to interferences etc..
5. Real-time services (e.g., the current location of the car)
Outline

• Introduction
• Aurora System Model
• Aurora Optimization
• Run-Time Operation
• Conclusion

Aurora System Model

Aurora System: operators

Windowed operators operate on sets of consecutive tuples from a stream, (“windows”)

• Operators for a single tuple
  Slide
  Tumble
  Latch
  Resample
• Operators for multiple tuples
  Filter
  Map
  Groupby
  Join

Aurora System: query model

Outline

Aurora Optimization

The objective is to obtain an online optimization with the minimum number of operator boxes and a continuous update of Aurora network over time

• Dynamic Continuous Query Optimization
  • Inserting Projections
  • Combining Boxes
  • Reordering Boxes
  • Ad-Hoc Query Optimization
Outline

• Introduction
• Aurora System Model
• Aurora Optimization
• Run-Time Operation
• Conclusion

Run-Time Operation

• QoS (Quality of Service) in Aurora
• Aurora Storage Management
• Real-Time Scheduling
• Introspection
• Load Shedding

Aurora run-time architecture

Assumptions:
• All QoS graphs are normalized
• The value \( \delta \) is chosen so that Aurora network operates with all outputs in the good zone
• All QoS graphs are convex except for value-based ones for scheduling and load shedding

Aurora Storage Management

The objective is to manage storage for the tuples that are active in the network and maintain extra storage space for the tuples required at the connection points

• Queue Management
  • Priority block elimination
  • Contiguous data block
  • Interacting with scheduler
• Connection Point Management
  • Managing historical requirement of tuples
  • B-tree built on timestamps

Scheduler ↔ Storage Manager

The objective is to have an efficient run-time operation by avoiding the hysteresis effect
### Real-Time Scheduling

The objective is to not only maximize overall QoS but also reduce overall tuple execution costs.

<table>
<thead>
<tr>
<th>1. Priority Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-based approach</td>
</tr>
<tr>
<td>Feedback-based approach</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Train Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-box non-linearity</td>
</tr>
<tr>
<td>Intra-box non-linearity</td>
</tr>
</tbody>
</table>

### Introspection

- **Static Analysis**
  Making sure an Aurora network has sufficient computational resources (hardware requirement)

- **Dynamic Analysis**
  Detecting an unpredictable, long-duration spikes in input rates that causes unexpected performance degradation

### Load Shedding

- **Load shedding by dropping tuples**
  Minimizing the overall performance degradation as a result of static analysis

- **Semantic load shedding by filtering tuples**
  Semantic load shedding based on value-based QoS information if available

### Conclusion

- **Aurora is a new rising star in DBMS**
- **More demand for monitoring applications**
- **Future direction 1:**
  Aurora* for distributed processing
- **Future direction 2:**
  More efficient data handling algorithm for missing and/or imprecise data that is common in sensor network
- **Aurora* and beyond….to Borealis**

---

**Thank you for your attention!**