Solving Manufacturing Equipment Monitoring Through Efficient Complex Event Processing

Tilmann Rabl, Kaiwen Zhang, Mohammad Sadoghi, Navneet Kumar Pandey, Aakash Nigam, Chen Wang, Hans-Arno Jacobsen

Middleware Systems Research Group, University of Toronto
Agenda

- Complex Event Processing Scenarios
- Architecture
- Evaluation
- Demo
Motivation

- Seminal course “Large-Scale Data Management”
  - Data storage
  - Event processing

- Complex event processing scenarios
  - Application Performance Management (APM)
  - Smart traffic monitoring
  - Energy monitoring

- Team project
  - DEBS Grand Challenge
Application Performance Management

- Monitoring of enterprise systems
- Find bottlenecks, problems
- Trace transactions, measure utilization

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Value</th>
<th>Min. Value</th>
<th>Max. Value</th>
<th>Data Points</th>
<th>Start Time (mllls)</th>
<th>Stop Time (mllls)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frontends:ApplicationX:AverageResponseTime (ms)</td>
<td>3</td>
<td>2.0</td>
<td>4.0</td>
<td>2</td>
<td>131412145</td>
<td>131412145</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5000</td>
<td>5000</td>
</tr>
</tbody>
</table>
Smart Traffic Monitoring

- Traffic data from
  - cars, mobile devices, street sensors
- Event aggregation, filtering, correlation
Energy Monitoring

- Green computing
- Application-level energy monitoring
- API-based energy consumption estimation
Common Denominators

- High data rates
  - 1000 – 1000000+ events / sec
- Small data points
  - < 1 KB
- Complex queries
  - Filtering, aggregation, correlation
- Distributed setup

DEBS Grand Challenge
High-Level Architecture

- Java-based implementation
- Monitoring Service
  - Input data stream, marshalling
- Event Dissemination Substrate
  - (Optional) pub/sub layer, queues
- Continuous Query Evaluation
  - Consumes input, computes results
- Storage Manager
  - Stores data, enables querying
- Client
  - Visualize query results
Storage Architecture

- Data is stored in tables
  - Key-value pairs

- Table Index
  - Fast lookup

- Compressor
  - Efficient data storage
  - Run length encoding

- Grand Challenge data
  - Run-length: 20000
  - Compression: 99.99%
Google Web Toolkit-based
Client Java-code compiled to JavaScript
Displays all Grand Challenge results
  Plots, results, alarms
Evaluation

- Distributed setup
  - Separated servers for data generator and monitoring tool
- Configuration:
  - 2 servers
  - 2 x dual core Xeon processor
  - 4 GB RAM
  - Gigabit Ethernet
- Data set: 5 min + 27 days + synthetic
- Metrics: latency & throughput
- Synthetic data
  - Many conflicts
  - Maximum throughput
Processing Overhead
Real Workload

- 0 – 200 queries (Q1, Q2 repeated)
- Linear in the number of queries
- Stable with increasing generator speedup
Throughput 
Real Workload

- Throughput controlled by data generator
- Data generator does not saturate system
- Peak 9000 events/sec ~ 0.11 ms arrival rate
- Well below maximum throughput
Latency Synthetic Workload

- Maximum throughput
  - 0.019 ms (10 queries)
- Minimum latency
  - 0.019 ms (10 queries)
- Maximum latency
  - 0.63 ms (1400 queries)
Throughput
Synthetic Workload

- **Maximum throughput**
  - 40000 events / sec (20 queries)

- **Minimum throughput**
  - 1500 events / sec (1400 queries)
Demo

- Live!
Conclusion

- Complex event processing scenarios
  - Application performance management
  - Smart traffic monitoring
  - Energy monitoring
  - DEBS Grand Challenge

- Efficient implementation of the Grand Challenge
  - Java-based
  - Google Web Toolkit GUI
  - Synthetic data generator
    - Up to 40000 events per second
    - Up to 1400 queries
Questions

- Thank you!