MOIRAE: EXPLOITING HISTORY IN MONITORING SYSTEMS

Magdalena Balazinska, YongChul Kwon, Kristin Lee, Guiping Xu

University of Washington
Monitoring Application

- Abundant monitoring applications
  - RFID Ecosystem
  - Web service monitoring
  - Network monitoring
  - Car traffic monitoring
  - ...
- Is *event detection* the only thing we can do?
An end user is experiencing poor service quality. What causes this anomaly?

- Detect
- Correlate
- Analyze
- Report

User profiles
Network measurements
Service logs
Server stats

5 most probable causes & reasons
Monitoring & Analysis

Either
Two separate systems
Or
Tight integration

Detect
Correlate
Analyze
Report

Data Archive

Mostly offline
Huge data size
Long latency
Moirae Project

♦ Goal
  • Support integrated queries over live and archived data

♦ Challenges
  • What types of queries should we support?
  • What query language is suitable?
  • How to efficiently support their execution?
Query Type 1: Standard Hybrid Query

- Request specific historical data to complement an event
  - Did any of the users already experienced this failure in last 7 days?

![Diagram]

- Input Streams
- Stream Proc. Operators
- Recall
- Other Stream Proc. Ops.

- Continuous stream processing (event query)
- Look up historical information (historical query)
Query Type 2: Contextual Hybrid Query

- Similar past events
  - Similarity in correlated information, Context

- Example
  - Find similar past poor service experiences
  - Take top-5 cases in similarity
  - Join them with customer service report
  - Report top-5 excuses explanations for poor service!
## Event & Context

### Event

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>User</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6:02 p.m</td>
<td>Alice</td>
<td>svr7</td>
</tr>
</tbody>
</table>

### Context

**Aspect 1: Concurrent users from same region**

<table>
<thead>
<tr>
<th>User</th>
<th>Loc</th>
<th>ISP</th>
<th>Serve</th>
<th>Avg. latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>WA</td>
<td>C</td>
<td>svr7</td>
<td>30 sec</td>
</tr>
<tr>
<td>Bob</td>
<td>WA</td>
<td>V</td>
<td>svr1</td>
<td>2 sec</td>
</tr>
</tbody>
</table>

**Aspect 2: Server resource usage stats**

<table>
<thead>
<tr>
<th>Server</th>
<th>CPU</th>
<th>Mem</th>
<th>Rx</th>
<th>Tx</th>
</tr>
</thead>
<tbody>
<tr>
<td>svr1</td>
<td>40%</td>
<td>90%</td>
<td>5Mbps</td>
<td>38Mbps</td>
</tr>
<tr>
<td>svr7</td>
<td>30%</td>
<td>85%</td>
<td>4.5Mbps</td>
<td>40Mbps</td>
</tr>
</tbody>
</table>

How do we compare event contexts?

Other aspects…
Context Distance Measure

- Key intuitions
  - Abnormal values are more interesting
  - Abnormal values are observed infrequently

- Based on Match-And-Compare measure
- Filter normal entities when abnormal ones exist
- Use histogram to compute distances
- Weigh more heavily abnormal values and tuples
Filtering Normal Values

- Does eliminating normal tuples improve results?
- Lower = Better

Base measure of CDM

Traditional measures
Distance in Distribution

- Lower = Better

Distance w.r.t manually ranked list

- Euclidean
- Histogram

NMAC  |  MAC  |  EMD  |  Hausdorff distance
0.3   |  0.4  |  0.5  |  0.4
Abnormality Weights

Distance to manually ranked list

Distance at 90%

Median
Current Status & Future Work

- Initial system design
  - Examining emerging technology
- Measure of Context Similarity
  - Integration with automatic clustering
  - Evaluation in more application domains
- Query model & Language
  - Clean and easy-to-write query language
Summary

- Monitoring applications accumulate history
- History can improve the quality of monitoring
- We are bridging the gap between present and past 😊

- For more information
Database/Ubicomp Poster & Demo Session

- Where
  - Database Lab (CSE605)
- When
  - From 1:30pm to 4:00pm TODAY!