Data Integrity and History

Magdalena Balazinska
joint work with
Gaetano Borriello, Nodira Khoussainova,
YongChul Kwon, Dan Suciu, and Evan Welbourne

University of Washington

August 2006
Two Challenges

● Problem 1
  – Errors are frequent; cannot always be masked
  – Errors affect application logic
  – How should we handle errors?

● Problem 2
  – Two types of data: current and historical
  – Want to enable queries over both types of data
  – Can we complement current info with history?
Example: RFID-Based Tracking

Show my tag reads over time:

Tagged object: all_objects
Start time: 01 Jun, 2006
Stop time: 01 Jul, 2006
Submit Query

Show a person's tag reads over time:

Person: suciu, gaetano, nodira
Start time: 01 Jun, 2006
Stop time: 01 Jul, 2006
Submit Query
Sample Queries

Where is my object:

Where is this object:  purse  Submit Query

How much time have I spent in the Allen Center:

Start time: 01 Jun, 2006  Stop time: 01 Jul, 2006  Submit Query

Where is this person:

Where is this person:  gaetano  Submit Query
Sample Useful Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>magda</td>
<td>purse</td>
<td>7</td>
<td>1</td>
<td>Wed Jul 26 09:22:12 2006</td>
</tr>
<tr>
<td>magda</td>
<td>purse</td>
<td>7</td>
<td>1</td>
<td>Wed Jul 26 09:22:13 2006</td>
</tr>
</tbody>
</table>

Magda is in the office ...

<table>
<thead>
<tr>
<th>magda</th>
<th>person_public</th>
<th>7</th>
<th>1</th>
<th>Wed Jul 26 19:28:50 2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>magda</td>
<td>purse</td>
<td>7</td>
<td>1</td>
<td>Wed Jul 26 19:28:51 2006</td>
</tr>
<tr>
<td>magda</td>
<td>person_public</td>
<td>7</td>
<td>1</td>
<td>Wed Jul 26 19:28:52 2006</td>
</tr>
</tbody>
</table>

Magda is home ...
Example of Erroneous Input

<table>
<thead>
<tr>
<th>CSE Net ID</th>
<th>Object</th>
<th>Reader Number</th>
<th>Antenna Number</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>magda</td>
<td>person_public</td>
<td>7</td>
<td>1</td>
<td>Thu Jul 27 09:21:51 2006</td>
</tr>
<tr>
<td>magda</td>
<td>person_public</td>
<td>7</td>
<td>2</td>
<td>Thu Jul 27 09:21:52 2006</td>
</tr>
<tr>
<td>magda</td>
<td>person_public</td>
<td>7</td>
<td>2</td>
<td>Thu Jul 27 09:21:52 2006</td>
</tr>
<tr>
<td>magda</td>
<td>person_public</td>
<td>7</td>
<td>2</td>
<td>Thu Jul 27 09:21:53 2006</td>
</tr>
<tr>
<td>magda</td>
<td>person_public</td>
<td>7</td>
<td>1</td>
<td>Thu Jul 27 09:21:54 2006</td>
</tr>
<tr>
<td>magda</td>
<td>laptop_power_cord</td>
<td>7</td>
<td>1</td>
<td>Thu Jul 27 09:21:54 2006</td>
</tr>
<tr>
<td>magda</td>
<td>laptop_power_cord</td>
<td>7</td>
<td>1</td>
<td>Thu Jul 27 09:21:55 2006</td>
</tr>
</tbody>
</table>

What happened to the laptop?
No purse this morning?
Another Example with Error

<table>
<thead>
<tr>
<th>magda</th>
<th>person_public</th>
<th>Z</th>
<th>1</th>
<th>Thu Jul 27 12:44:31 2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>magda</td>
<td>person_public</td>
<td>Z</td>
<td>2</td>
<td>Thu Jul 27 12:44:32 2006</td>
</tr>
<tr>
<td>magda</td>
<td>purse</td>
<td>Z</td>
<td>1</td>
<td>Thu Jul 27 12:44:32 2006</td>
</tr>
<tr>
<td>magda</td>
<td>purse</td>
<td>Z</td>
<td>2</td>
<td>Thu Jul 27 12:44:33 2006</td>
</tr>
<tr>
<td>magda</td>
<td>purse</td>
<td>Z</td>
<td>2</td>
<td>Thu Jul 27 13:23:54 2006</td>
</tr>
<tr>
<td>magda</td>
<td>purse</td>
<td>Z</td>
<td>1</td>
<td>Thu Jul 27 13:23:54 2006</td>
</tr>
</tbody>
</table>

The purse came back from lunch by itself...
Problem 1: Data Integrity

- **Data produced by sensors contains errors**
  - Missing input data (missed readings)
  - Erroneous input data (duplicate readings)
  - Sensor or system failures

- **Errors affect applications**
  - Errors compromise data integrity
  - Applications produce wrong results
  - Or they need to include code to handle errors
Error Handling

- Increase fault-tolerance
  - Replicate system components
  - Clean input data at various levels
- BUT
  - Some errors only visible at application level
    - ex: Person returns to office but never left it
  - Cannot always clean the data with certainty
    - ex: Missed purse reading vs forgotten purse
  - On a large scale, impossible to mask all errors
Possible Approach

- Show applications data quality information
- Enable apps to specify integrity constraints
  - Constraints defined over the data
  - Constraints can involve complex temporal events
- Use constraints to
  - Clean the data
  - Take app-defined action upon violation
- Leverage presence of multiple applications
  - Clean data incrementally
Using Constraints to Clean Data

FORALL INPUT1 AS I1, INPUT2 AS I2, ..., WHERE EXPR1

CHECK EXPR2

CONFIDENCE c

- Approach
  - Use constraints to identify missing data and conflicting data; generate missing data
  - Transform constraints into equations
  - Solve using maximum entropy
  - Produces probabilistic input data
Challenges

- **Complexity**
  - Integrity constraints can be complex

- **Efficiency**
  - Need to verify constraints in near-real-time
  - Need to clean data in near-real-time

- **Incremental cleaning, continuous processing**
  - Often can clean older data based on new info
  - How should this affect continuous processing?
Problem 2: Data History

- Two types of sensor data
  - Current (live) data: streams continuously
  - Historical data: stored on disk

- Typically, want to enable
  - Continuous queries over live data
  - Ad-hoc queries over data archives

- BUT
  - How to integrate history into continuous queries?
  - Data archive too large to query in near-real-time
Challenges

- Complement near-real-time data with history
  - “Alert me if there is a parking space available with a low history of theft”
  - “Find K events most similar to the current event”
- Hard problem when
  - Many different types of queries
  - Cannot index all data attributes
- Approach
  - Selectively examine relevant subsets of history
Conclusion

● Many data management problems
● We have emphasized two of them
  – Manage data integrity
  – Exploit data history
● Already challenging at a small scale
● Web-scale makes these problems harder