Online and Scalable Data Validation in Advanced Metering Infrastructures

Vincenzo Gulisano, Magnus Almgren and Marina Papatriantafilou
Agenda

1. Problem statement
2. Preliminaries – Data Streaming
3. Streaming-based Data Validation
4. Conclusions
Agenda

1. Problem statement
2. Preliminaries – Data Streaming
3. Streaming-based Data Validation
4. Conclusions
Online and Scalable Data Validation in Advanced Metering Infrastructures

Noisy and Lossy data: bad-calibrated / faulty devices, lossy communication, malicious users, ...

How can we validate data given that...
• There is a large volume of continuous data demanding for distributed and parallel analysis
• Validation rules depend on installation-specific features such as brands, devices, protocols, ...
  • System experts should define installation-specific validation rules?
Agenda

1. Problem statement
2. Preliminaries – Data Streaming
3. Streaming-based Data Validation
4. Conclusions
Data Streaming - Motivation

- Financial applications, sensor networks monitoring, ...
  require
  - Continuous processing of data streams
  - Real Time fashion

- Store and process is not feasible
  - Financial markets: process millions of messages/second,
    take fast decisions (< 100 microseconds)

- Data Streaming:
  - In memory
  - Bounded resources
  - Efficient one-pass analysis
Data Streaming - System Model

- Data Stream: unbounded sequence of tuples
- Example: consumption readings

<table>
<thead>
<tr>
<th>Field</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>House</td>
<td>text</td>
</tr>
<tr>
<td>Device</td>
<td>text</td>
</tr>
<tr>
<td>Time (secs)</td>
<td>int</td>
</tr>
<tr>
<td>Consumption (kWh)</td>
<td>double</td>
</tr>
</tbody>
</table>

A B 8:00 1
C D 8:20 2
A E 8:35 4
Data Streaming - System Model

• Operators:

  - **Stateless**
    - 1 input tuple
    - 1 output tuple

  - **Stateful**
    - 1+ input tuple(s)
    - 1 output tuple

Online and Scalable Data Validation in Advanced Metering Infrastructures.
Vincenzo Gulisano, Magnus Almgren and Marina Papatriantafilou
Data Streaming - System Model

• Infinite sequence of tuples / bounded memory
  → windows

• Example: 1 hour windows

- [8:00,9:00)
- [8:20,9:20)
- [8:40,9:40)
Data Streaming - System Model

• Infinite sequence of tuples / bounded memory
  → windows
• Example: 1 hour windows

Counter: 1  Counter: 3
Counter: 2  Counter: 4

<table>
<thead>
<tr>
<th>Time</th>
<th>Counter</th>
<th>Time</th>
<th>Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:05</td>
<td></td>
<td>8:45</td>
<td></td>
</tr>
<tr>
<td>8:15</td>
<td></td>
<td>9:05</td>
<td></td>
</tr>
<tr>
<td>8:22</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Output: 4

[8:00,9:00)
Data Streaming - System Model

• Infinite sequence of tuples / bounded memory
  → windows

• Example: 1 hour windows

Counter: 3

[8:20, 9:20)
Agenda

1. Problem statement
2. Preliminaries – Data Streaming
3. Streaming-based Data Validation
4. Conclusions
Why Streaming-based data Validation?

- Expressive
- Online
- Parallel & Distributed
Sample Streaming-based Data Validation: Interpolate missing consumption values

- **Aggregate**
  - Match consecutive readings from each meter

- **Filter**
  - Forward if time distance exceeds a certain threshold

- **Map**
  - Interpolate missing values

<table>
<thead>
<tr>
<th>Field</th>
<th>Device</th>
<th>Time</th>
<th>Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Time_1</td>
<td>Time_2</td>
<td>Consumption_1</td>
</tr>
<tr>
<td>Consumption</td>
<td>Consumption_2</td>
<td>Consumption_2</td>
<td></td>
</tr>
</tbody>
</table>
Sample Streaming-based Data Validation:
Interpolate missing consumption values
Evaluation - Setup

• From a real-world AMI, data extracted from 50 meters
• Data covers 13 months (May 2012 – June 2013)

• Implemented on top of Storm, a widely-used SPE (e.g., used in Twitter)

• Evaluated in terms of throughput (tuples/second) and processing latency (milliseconds)
Evaluation
Agenda

1. Problem statement
2. Preliminaries – Data Streaming
3. Streaming-based Data Validation
4. Conclusions
Conclusions

• Streaming-based Data Validation
  • Expressive / Distributed-Parallel / Online
  • Implemented on top of the Storm SPE
  • Evaluated with real-world AMI data