Enabling Micro-Synchrophasor Data Analytics

Omid Ardakanian
University of California, Berkeley

NASPI March 2016
120 samples per second → **3.8 billion samples** per stream per year → **30 billion bytes** per stream per year
Introduction

• High-precision high-sample-rate data from distributed high fidelity sensors
  ∗ many sensors, a wide range of temporal scales, rare events

• Finding anomalies in these systems is the holy grail
  ∗ failing to identify and react to critical events in a timely manner may cost millions of dollars

• Energy data analytics (both real-time and historical) is critical yet computationally expensive
  ∗ the ability to detect, analyze, and control with a limited time budget
Goals

• Detect: identify rare events
  • using an efficient search algorithm that is logarithmic in the size of the data set and linear in the number of events that are found

• Analyze: run compute intensive tasks on smaller chunks of data

• Control: take corrective/preventive actions (in real-time applications)
System Architecture

- Sample events flow to Event Labeling, which provides ground truth.
- Event candidates feed into ML Algorithms for predictive model generation.
- Selected feature vectors are used by Event Detection.
- Data streams and statistical queries are analyzed by BTrDB Cluster.
- Distributed high fidelity sensors connect to Internet.
- Distillate Framework provides plotting services.
BTrDB Timeseries Database

- High throughput, fixed response-time timeseries store running on a four-node cluster
  - 53 million inserted values per second
  - 119 million queried values per second
- Provides nanosecond timestamp precision
- Supports out-of-order arrivals

References:
Abstraction for Timeseries Data

- Time-partitioning version-annotated copy-on-write K-ary tree

![Diagram showing a tree structure with nodes labeled Stats, Count, and [T,V] with version numbers and deleted ranges highlighted.](image)
Statistical Summaries

- statistical summaries (max, min, average, and count) are stored at different temporal resolutions
System Architecture

- Sample events → Event Labeling
- Event Library
- Event candidates → Event Detection
- Selected feature vectors
- Data streams, statistical queries

- ML Algorithms
  - Predictive model
  - Ground truth

- Internet
  - Distributed high fidelity sensors

- Distillate Framework

- BTrDB Cluster
- Plotting Service

multi-resolution search algorithm
Example Query

Find 5-second intervals that contain at least one value greater than a threshold
Example Query

Find 5-second intervals that contain a value greater than a threshold

• Query \texttt{max} at the given temporal resolution

• Dive down if $\text{max}_{\text{resolution}} > \text{threshold}$

• Repeat for the next temporal resolution until the desired resolution is reached
Multi-Resolution Search

• Start with a definition of the event (search criteria)

• Query statistical summaries of data at a given temporal resolution

• Compare a function of these statistical summaries against a threshold

• Dive down if the condition is satisfied

• Query raw data when the desired resolution is reached and run your algorithm on a small chunk of data
Interesting Events

- voltage sags
  - voltage magnitude stream
- tap changing events
  - angle difference stream
- reverse flows
  - real power or power factor stream
- switching events
- ...

14
Case Study: Voltage Sag Detection
Step 1: Querying Statistical Summaries at a Given Temporal Resolution

kernel: \((\text{mean}_{\text{res}} - \text{min}_{\text{res}})/\text{mean}_{\text{res}}\)

statistical records: min, mean
Step 2: Diving Down

kernel: \(\frac{\text{mean}_{\text{res}} - \text{min}_{\text{res}}}{\text{mean}_{\text{res}}}\)

statistical records: min, mean
Step 3: Querying Raw Data
Evaluation
Example Result

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>/clean/GP_BUS1/L1MAG</td>
<td>9</td>
<td>431.77</td>
<td>4</td>
<td>237.13</td>
<td>0</td>
<td>76.78</td>
<td>0</td>
<td>88.41</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/GP_BUS1/L2MAG</td>
<td>10</td>
<td>394.39</td>
<td>4</td>
<td>226.85</td>
<td>1</td>
<td>115.30</td>
<td>0</td>
<td>70.55</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/GP_BUS1/L3MAG</td>
<td>5</td>
<td>309.07</td>
<td>2</td>
<td>163.25</td>
<td>1</td>
<td>118.95</td>
<td>0</td>
<td>77.08</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/switch_a6/L1MAG</td>
<td>14</td>
<td>666.59</td>
<td>6</td>
<td>273.01</td>
<td>3</td>
<td>194.95</td>
<td>1</td>
<td>132.75</td>
<td>330</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/switch_a6/L2MAG</td>
<td>21</td>
<td>947.24</td>
<td>11</td>
<td>523.78</td>
<td>4</td>
<td>235.44</td>
<td>3</td>
<td>190.83</td>
<td>330</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/switch_a6/L3MAG</td>
<td>11</td>
<td>608.94</td>
<td>4</td>
<td>318.44</td>
<td>2</td>
<td>213.57</td>
<td>0</td>
<td>90.06</td>
<td>330</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/RPU/CE_CERT_Bld_1200/L1MAG</td>
<td>8</td>
<td>312.53</td>
<td>2</td>
<td>68.41</td>
<td>1</td>
<td>64.93</td>
<td>1</td>
<td>66.55</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/RPU/CE_CERT_Bld_1200/L2MAG</td>
<td>12</td>
<td>379.19</td>
<td>4</td>
<td>163.71</td>
<td>3</td>
<td>119.51</td>
<td>2</td>
<td>112.95</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/clean/RPU/CE_CERT_Bld_1200/L3MAG</td>
<td>12</td>
<td>627.72</td>
<td>4</td>
<td>228.18</td>
<td>2</td>
<td>111.41</td>
<td>2</td>
<td>133.00</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10% drop

logarithmic in the size of the data set and linear in the number of events that are found
Event Detection: A Data Driven Approach

multi-resolution search algorithm
Takeaways

• Complexity of the search algorithm is $O(n\text{Log}(L))$
  
  • Locating and analyzing rare events among billions of time-value pairs is possible in a fraction of a second

• Defining a kernel function can be quite challenging for some detectors

• Machine learning techniques can be used to develop sophisticated detectors