Outline

• Platform Requirements - 4 Key Drivers
  1. The Nature of Time Series
  2. The Real World is Messy (or Sensors)
  3. Analytics as First Class Citizens
  4. Size Matters

• Benchmarking Implications

• What is a Platform?
The Nature of Time Series
(timestamp, value)
(timestamp, value)

64-bit integer

$[-(2^{63}), 2^{63} - 1]$

Or

(timestamp, value)

(1515112200000000000, 50.6285209655761)
(1515112200008333333, 50.6273155212402)
(1515112200016666666, 50.6269416809082)
(1515112200024999999, 50.6258087158203)
(1515112200033333332, 50.6216735839843)
(1515112200041666665, 50.6205940246582)
(1515112200049999998, 50.6227645874023)
(1515112200058333331, 50.6207199096679)
(1515112200066666664, 50.6192970275878)
(1515112200074999997, 50.6227836608886)
(1515112200083333330, 50.6249427795410)
Intrinsic Redundancy

• Can compress timestamps and values
  – Lossy and Lossless
  – Intra-stream and Inter-stream

• Many approaches available


• Classic space/time tradeoff

• We achieve ~3:1 lossless compression
Write Patterns
Write Patterns
1. Human interaction and exploration of the data
2. Analytics
“Overview first, zoom and filter, then details-on-demand.”

The Visual Information-Seeking Mantra [Shneiderman, 1996] summarizes many visual design guidelines and provides an excellent framework for designing information visualization applications.
Random, Multi-Resolution Read Patterns

About 4 billion data points
Random, Multi-Resolution Read Patterns
Random, Multi-Resolution Read Patterns
Random, Multi-Resolution Read Patterns
Random, Multi-Resolution Read Patterns
Random, Multi-Resolution Read Patterns

50 datapoints (8 orders of magnitude)
Berkeley tree Data Structure

Copy on write K-ary Tree
Partitioning static time (1933 to 2079)

Leaf nodes
- Time, value pairs + length

Internal nodes
- Pointers to children
- Version annotations for children
- Aggregates for children
  - Min, Mean, Max, Count
  - Any associative operator
The Real World is Messy
Q: What changed between v2 and v3?
Q: What changed between v2 and v3?

A: These ranges:
Q: What changed between v2 and v3?
Who Cares?
Who Cares?

• Efficiently update calculations on out of order data
Who Cares?

- Efficiently update calculations on out of order data
- Idempotent calculations???
Who Cares?

- Efficiently update calculations on out of order data
- Idempotent calculations????
- Rewinding data arrival to understand and diagnose problems
Versioning Source Code
Analytics as First Class Citizens
Common Analytics Patterns
Common Analytics Patterns

Clean
Common Analytics Patterns

Clean

Filter
Common Analytics Patterns

1. Clean
2. Filter
3. Compute Power
Directed Acyclic Graph of Calculations
But Wait, There’s More

- Windowing operations
- Spatial/Frequency Transforms
  - Wavelet
  - Fourier
  - Stockwell
- Indexing
- Clustering
- Classification
- Categorization
- Anomaly/Event/Novelty Detection
- Motif Discovery
But Wait, There’s Even More!

Old Paradigm - Software Engineering
- Humans write the code
- Limited by ability to describe exactly what must be done without error

New Paradigm - Machine Learning
- Data teaches algorithms to perform function or task
- Limited by the amount of data and algorithms
- Algorithms need **ALL** available data
- Capable of tackling high dimensional problems
Who Cares?

- Analytics must be first class citizens of the platform
- The platform must be built from the ground up to support relevant analytic use cases
Size Matters
# of Streams

<table>
<thead>
<tr>
<th>100000</th>
<th>10000</th>
<th>1000</th>
<th>100</th>
<th>10</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>1.3 Mbit</td>
<td>128 Mbit</td>
<td>1.3 Gbit</td>
<td>128 Gbit</td>
<td>1.3 Tbit</td>
</tr>
<tr>
<td>10000</td>
<td>1.3 Mbit</td>
<td>128 Mbit</td>
<td>1.3 Gbit</td>
<td>128 Gbit</td>
<td>1.3 Tbit</td>
</tr>
<tr>
<td>1000</td>
<td>1.3 Mbit</td>
<td>128 Mbit</td>
<td>1.3 Gbit</td>
<td>128 Gbit</td>
<td>1.3 Tbit</td>
</tr>
<tr>
<td>100</td>
<td>1.3 Mbit</td>
<td>128 Mbit</td>
<td>1.3 Gbit</td>
<td>128 Gbit</td>
<td>1.3 Tbit</td>
</tr>
<tr>
<td>10</td>
<td>1.3 Mbit</td>
<td>128 Mbit</td>
<td>1.3 Gbit</td>
<td>128 Gbit</td>
<td>1.3 Tbit</td>
</tr>
<tr>
<td>1</td>
<td>1.3 Mbit</td>
<td>128 Mbit</td>
<td>1.3 Gbit</td>
<td>128 Gbit</td>
<td>1.3 Tbit</td>
</tr>
</tbody>
</table>

## Sampling Rates

- **SCADA**
- **PMU**
- **DFR**
- **Sampling Rates**
Annual Data Volumes (Lower Bound)

<table>
<thead>
<tr>
<th># of Streams</th>
<th>0.1 Hz</th>
<th>1 Hz</th>
<th>10 Hz</th>
<th>100 Hz</th>
<th>1 KHz</th>
<th>10 KHz</th>
<th>100 KHz</th>
<th>1 MHz</th>
<th>10 MHz</th>
<th>100 MHz</th>
<th>1 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>4.6 TB</td>
<td>45.9 TB</td>
<td>459.2 TB</td>
<td>4.5 PB</td>
<td>44.9 PB</td>
<td>448.5 PB</td>
<td>4.4 EB</td>
<td>43.8 EB</td>
<td>437.9 EB</td>
<td>4.3 ZB</td>
<td>42.8 ZB</td>
</tr>
<tr>
<td>1000</td>
<td>470.2 GB</td>
<td>4.6 TB</td>
<td>45.9 TB</td>
<td>459.2 TB</td>
<td>4.5 PB</td>
<td>44.9 PB</td>
<td>448.5 PB</td>
<td>4.4 EB</td>
<td>43.8 EB</td>
<td>437.9 EB</td>
<td>4.3 ZB</td>
</tr>
<tr>
<td>100</td>
<td>47.0 GB</td>
<td>470.2 GB</td>
<td>4.6 TB</td>
<td>45.9 TB</td>
<td>459.2 TB</td>
<td>4.5 PB</td>
<td>44.9 PB</td>
<td>448.5 PB</td>
<td>4.4 EB</td>
<td>43.8 EB</td>
<td>437.9 EB</td>
</tr>
<tr>
<td>10</td>
<td>481.5 MB</td>
<td>4.7 GB</td>
<td>470.2 GB</td>
<td>4.6 TB</td>
<td>45.9 TB</td>
<td>459.2 TB</td>
<td>4.5 PB</td>
<td>44.9 PB</td>
<td>448.5 PB</td>
<td>4.4 EB</td>
<td>43.8 EB</td>
</tr>
<tr>
<td>1</td>
<td>48.1 MB</td>
<td>481.5 MB</td>
<td>4.7 GB</td>
<td>470.2 GB</td>
<td>4.6 TB</td>
<td>45.9 TB</td>
<td>459.2 TB</td>
<td>4.5 PB</td>
<td>44.9 PB</td>
<td>448.5 PB</td>
<td>4.4 EB</td>
</tr>
</tbody>
</table>

Sampling Rates

SCADA | PMU | DFR | Zettabytes | Exabytes | Petabytes | Terabytes | Gigabytes | Megabytes

Spring 2018 NASPI | 46
Big Data Conclusions

• Don’t move the data, move the calculations.  
  – (Why analytics are first class citizens)
• We are going to need a bigger [machine(s)].
Two Options
How this Played Out

2003
Google File System
Distributed file system over commodity hardware.

2004
Map Reduce
Distributed processing framework to simplify parallel programming tasks.

2006
Big Table
A distributed storage system for structured data.

2010
Pregel
A large scale graph processing system.

2013
Spanner
Google’s globally distributed database.

2015
The Dataflow Model
A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing.
Who Cares?

- No single point of failure/resilient
- Much more cost effective
- Buy more capacity when you need it
- Flexible and adaptable
Benchmark Considerations
Core Benchmarks - Reading and Writing Data

<table>
<thead>
<tr>
<th>#BTrDB</th>
<th>Streams</th>
<th>Total points</th>
<th>#Conn</th>
<th>Insert [mil/s]</th>
<th>Cold Query [mil/s]</th>
<th>Warm Query [mil/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>500 mil</td>
<td>30</td>
<td>16.77</td>
<td>9.79</td>
<td>33.54</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>1000 mil</td>
<td>60</td>
<td>28.13</td>
<td>17.23</td>
<td>61.44</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>1500 mil</td>
<td>90</td>
<td>36.68</td>
<td>22.05</td>
<td>78.47</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>2000 mil</td>
<td>120</td>
<td>53.35</td>
<td>33.67</td>
<td><strong>119.87</strong></td>
</tr>
</tbody>
</table>
## Analytics Benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Distributed</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identity</td>
<td>Phase Difference</td>
<td>Reactive/Fundamental Pwr</td>
</tr>
<tr>
<td>Input/Output streams</td>
<td>1/1</td>
<td>2/1</td>
<td>4/2</td>
</tr>
<tr>
<td>Compute changeset</td>
<td>972 μs</td>
<td>1659 μs</td>
<td>1180 μs</td>
</tr>
<tr>
<td>Query data [s]</td>
<td>69.8</td>
<td>104.4</td>
<td>196.9</td>
</tr>
<tr>
<td>Kernel calculation [s]</td>
<td>10.8</td>
<td>22.7</td>
<td>245.5</td>
</tr>
<tr>
<td>Delete old data [s]</td>
<td>6.7</td>
<td>6.9</td>
<td>15.8</td>
</tr>
<tr>
<td>Insert new data [s]</td>
<td>40.7</td>
<td>39.8</td>
<td>66.5</td>
</tr>
<tr>
<td>Changeset / compute time</td>
<td>1064 x</td>
<td>773 x</td>
<td>259 x</td>
</tr>
</tbody>
</table>
What is a Platform?
What isn’t a Time Series Platform?

Time Series

Not Time Series

“Small” Data

“Big” Data

PingThings
What Does a Time Series Data Platform Do?

Ingest/ Egest
Clean/ Condition/ Fix
Store
Visualize
Access/ Use/ Build With
Analyze
Learn From
The Goal of the Platform

Allow utility subject matter experts to create value from (sensor) data.
“A platform is a business based on enabling value-creating interactions between external producers and consumers. The platform provides an open, participative infrastructure for these interactions and sets governance conditions for them. The platform’s overarching purpose: to consummate matches among users and facilitate the exchange of goods, services, or social currency, thereby enabling value creation for all participants.

Strategy has moved from controlling unique internal resources and erecting competitive barriers to orchestrating external resources and engaging vibrant communities. And innovation is no longer the province of in-house experts and research and development labs, but is produced through crowdsourcing and the contribution of ideas by independent participants in the platform. External resources don’t completely replace internal resources—more often they serve as a complement. But platform firms emphasize ecosystem governance more than product optimization, and persuasion of outside partners more than control of internal employees.”
Hi-speed Historical Data Ingest

Diverse Sensor Population

Ingest Engine

Time Series Datastore

API Layer

Analytic Pipeline

Diverse Sensor Population

PMU, uPMU, DFR, PQube

Hi-speed Historical Data Ingest

Multi-format Data Export

Fusion of Simulation and Sensor Data

Multi-format Data Export

Hi-speed Data Export

Modeling and Simulation Connectivity

Fusion of Simulation and Sensor Data

Time Series Datastore

BTrDB

Deep Learning and Machine Learning at Scale

Human-scale Data Engagement

Customizable Real-Time Dashboards

Spreadsheets

Interactive Data Exploration and Analysis

Rapid Deployment of Production Analytics

Bespoke Web and Mobile Applications

Hi-speed Data Export

PredictiveGrid™

Deep Learning and Machine Learning

Ingest Engine

C37.118

COMTRADE

GEP/STTP

CSV, .d, .d2

Ingestor_N

Modeling and Simulation Connectivity

Hi-speed Data Export

TensorFlow

API Layer

Analytic Pipeline

GitHub

f(x,y,z)

P + jQ

Distillate_N

Real-Time & Historical Analytic Computing

Deep Learning and Machine Learning

sean@pingthings.ai