Time Series Data at Scale

IEEE PES General Meeting
August 7th, 2018 – Portland, OR

Sean Patrick Murphy
PingThings

Michael Andersen
Berkeley
UNIVERSITY OF CALIFORNIA

Kevin Jones
Dominion Energy®

(timestamp, value)

(timestamp, value)

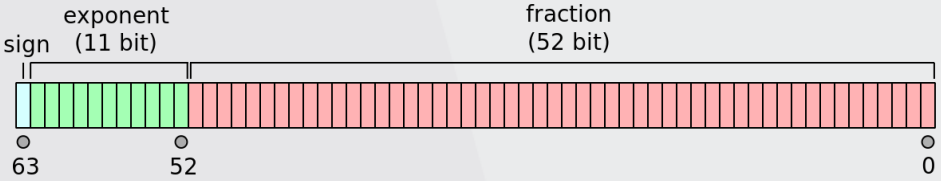
64-bit integer

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

Or

[-9,223,372,036,854,775,808,
9,223,372,036,854,775,807]

64-bit float



$$(-1)^{\text{sign}}(1.b_{51}b_{50}\ldots b_0)_2 \times 2^{e-1023}$$

Types of Time Series Data

Regular

- Measurements by software or hardware sensors at regular intervals of time.
- Can represent an "analog" physical process.
- Disciplines:
  - Financial
  - Server metrics/devops
  - Science/Engineering/Industrial
  - IoT

Irregular

- Time stamped "event."
- Generated either by:
  - Users (ex: arrival of tweet)
  - External events (ex: threshold reached, etc)

The Evolution of Time Series Data

Past

Store the last measurement/state (sensor, bank account, etc)

Present

Store the last month of measurements/state, maybe downsample (sensor, bank transactions, etc)

Future

Store every measurement/state for all time at full resolution (sensor, bank interactions, etc)

What Does this Mean?

500 PMUs

- 40 Streams/Channels
- 30Hz

What Does this Mean?

500 PMUs

- 40 Streams/Channels
- 30Hz

20,000 streams
600,000 points/second
302 TB/year
18,934,560,000,000 points/year

What Does this Mean?

500 PMUs

- 40 Streams/Channels
- 30Hz

20,000 streams
600,000 points/second
302 TB/year
18,934,560,000,000 points/year

- Copying this data over Gigabit Ethernet takes ~1 month
- Ignoring memory bandwidth constraints, Intel's i7-8700K (32 Gflops) Hexacore Processor would take 10 minutes to add a single number to each data point

What Does this Mean?

500 PMUs

- 40 Streams/Channels
- 30Hz

20,000 streams
600,000 points/second
302 TB/year
18,934,560,000,000 points/year

- Copying this data over Gigabit Ethernet takes ~1 month
- Ignoring memory bandwidth constraints, Intel's i7-8700K (32 Gflops) Hexacore Processor would take 10 minutes to add a single number to each data point

1. **Don't move the data, move the calculations.**

What Does this Mean?

500 PMUs

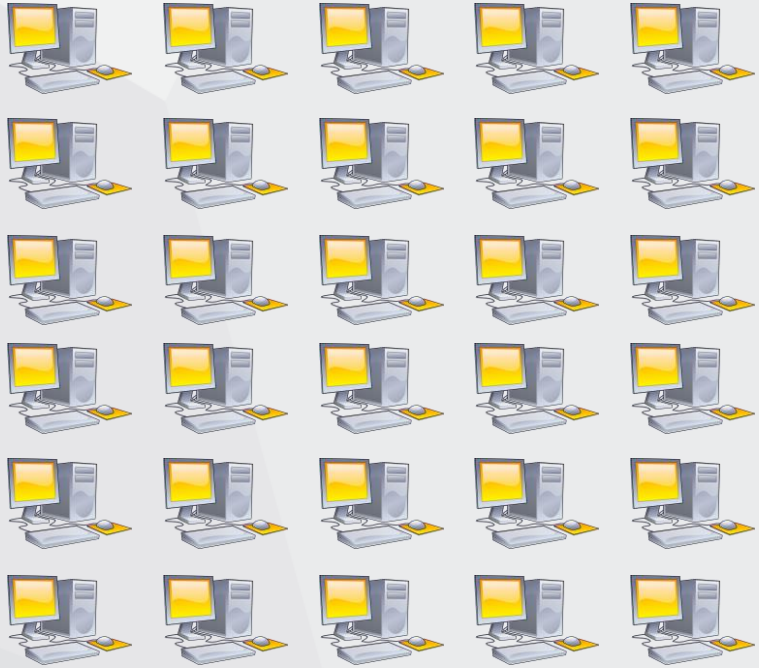
- 40 Streams/Channels
- 30Hz

20,000 streams
600,000 points/second
302 TB/year
18,934,560,000,000 points/year

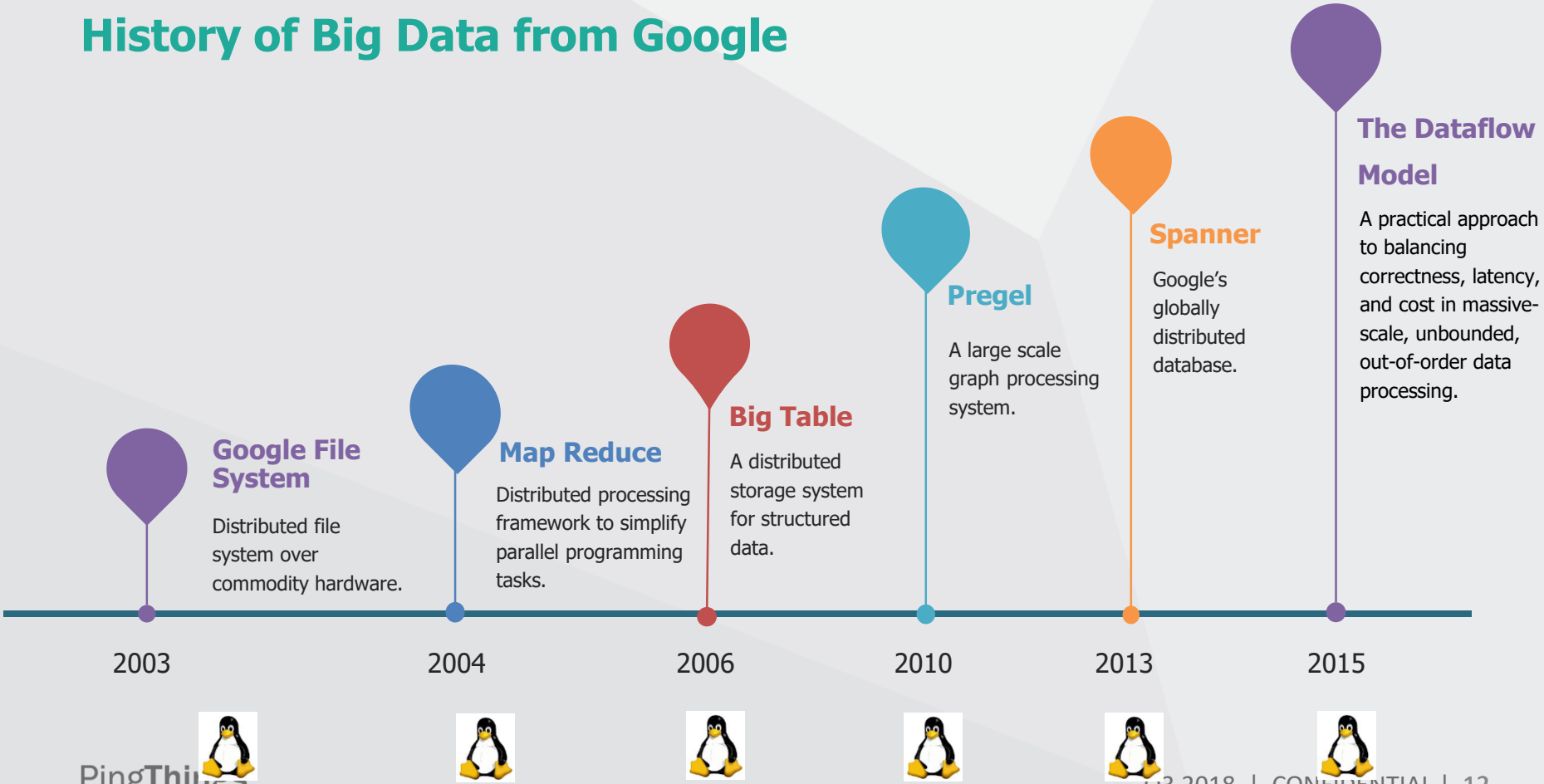
- Copying this data over Gigabit Ethernet takes ~1 month
- Ignoring memory bandwidth constraints, Intel's i7-8700K (32 Gflops) Hexacore Processor would take 10 minutes to add a single number to each data point

1. **Don't move the data, move the calculations.**
2. **We are going to need a bigger [machine(s)].**

Two Options

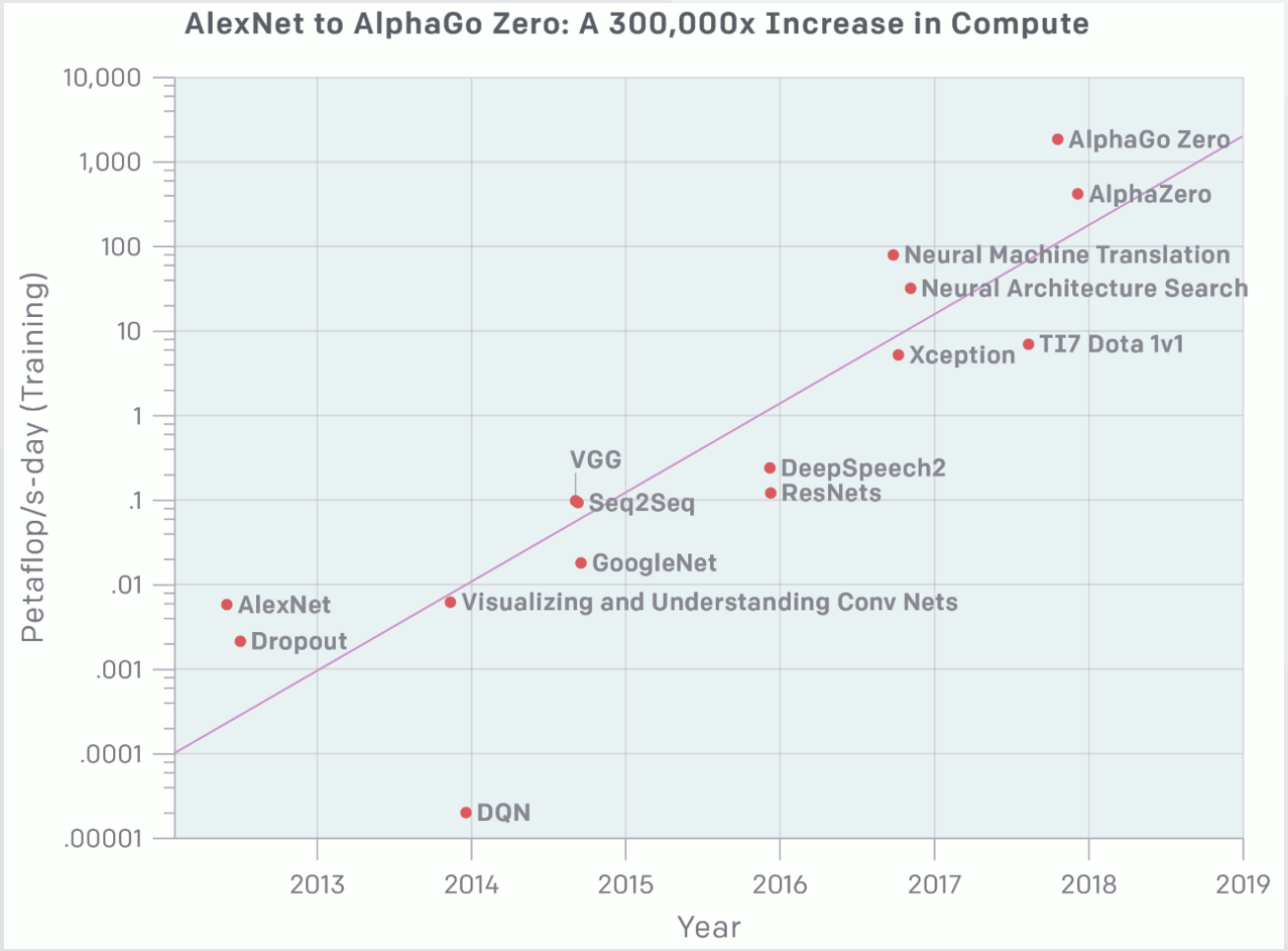


History of Big Data from Google



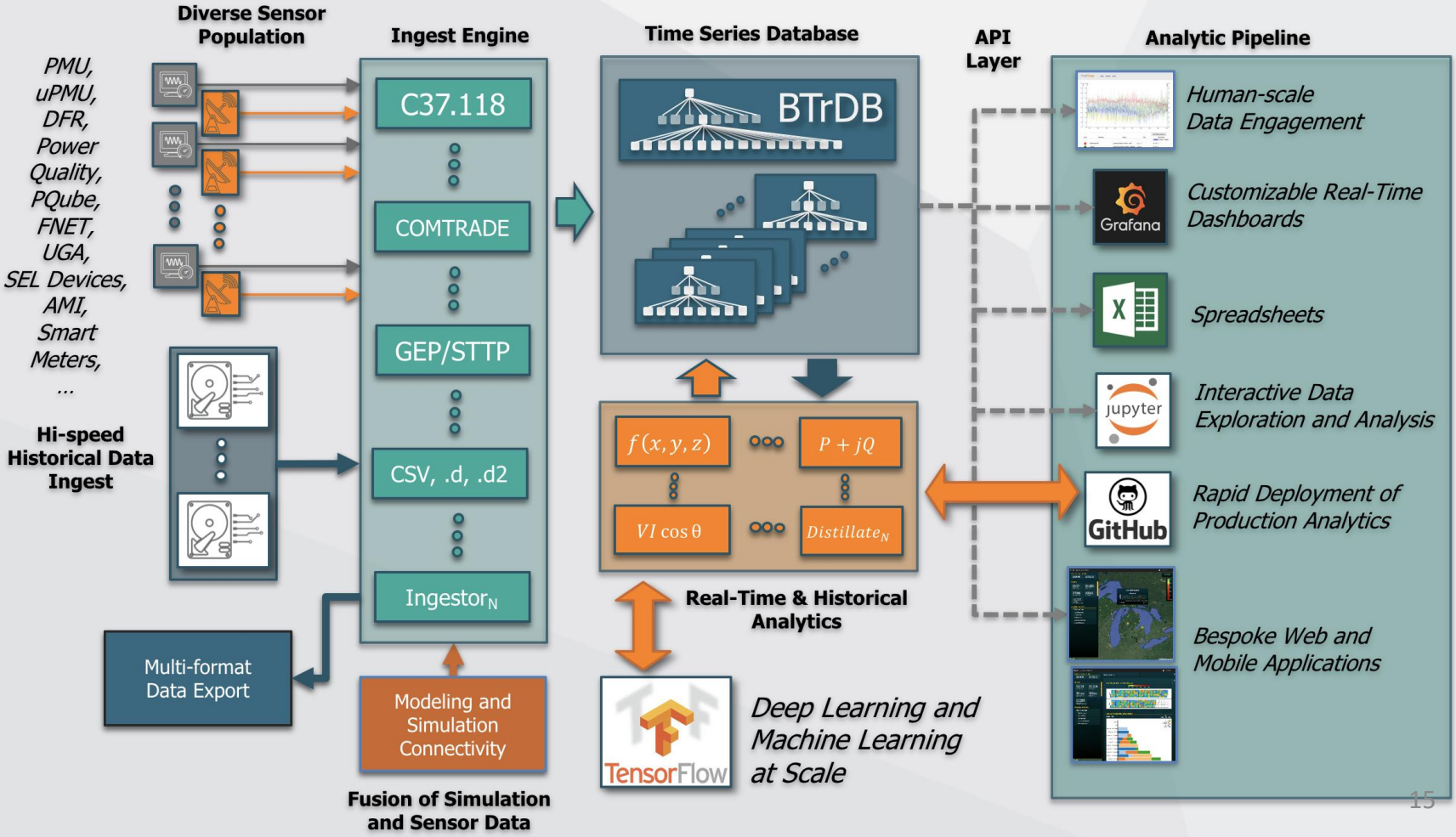
The Three Generations of Big Data Systems

	(1) Hadoop	(2) Spark	(3) Custom
Workload	Batch processing	Iterative processing	Continuous processing
Dominant Analytics Paradigm	Classic Business Analytics	Machine Learning	ML and Deep Learning
Features	Map reduce Disk oriented Generic	In memory Better tooling	Data type specific Industry and application focused
Limiting Reagent	Disk bandwidth	Memory capacity	Compute



"We're releasing an analysis showing that since 2012, the amount of compute used in the largest AI training runs has been increasing exponentially with a 3.5 month-doubling time (by comparison, Moore's Law had an 18-month doubling period). Since 2012, this metric has grown by more than 300,000x (an 18-month doubling period would yield only a 12x increase). Improvements in compute have been a key component of AI progress, so as long as this trend continues, it's worth preparing for the implications of systems far outside today's capabilities."

Universal Sensor Analytics Platform



A True 3rd Generation Platform

- For heterogeneous, multi-scale, multi-resolution sensor data
- Open source, open source tools, open data formats
- Tested beyond 100,000 PMUs
- Each signal can be up to 1 Gigahertz
- Designed to make the humans better
- Horizontally scalable and distributed
- Analytics as a first class citizen

(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)
```

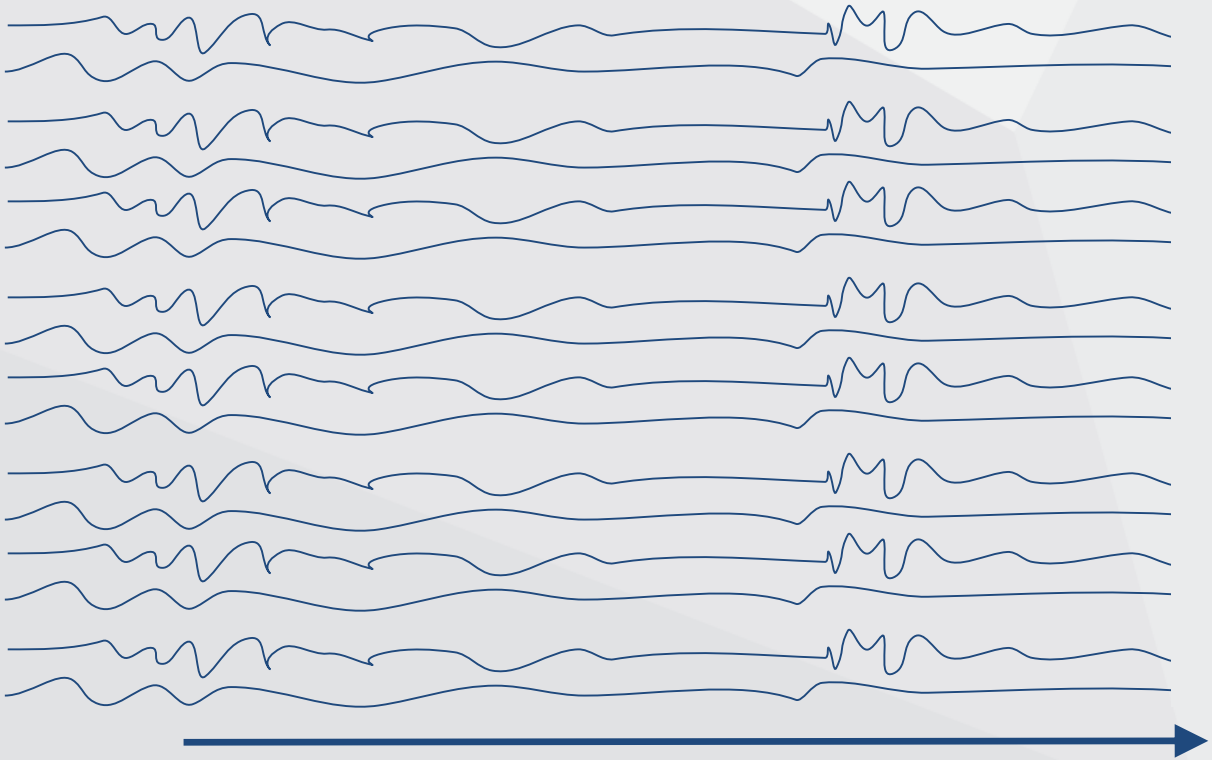
Intrinsic Redundancy

- Can compress timestamps and values
  - Lossy and Lossless
  - Intra-stream and Inter-stream
- Many approaches available

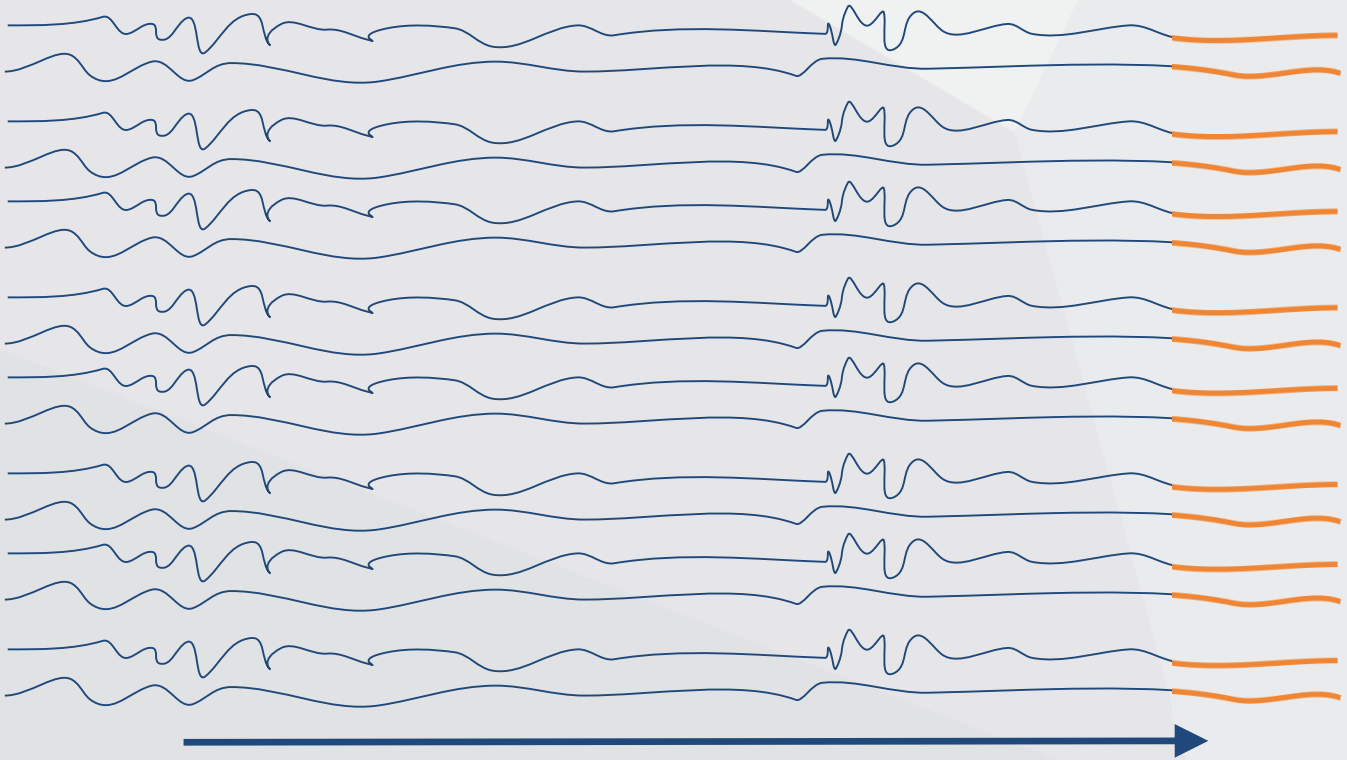
P. Lindstrom and M. Isenburg. Fast and Efficient Compression of Floating-Point Data. Visualization and Computer Graphics, IEEE Transactions on, 12(5):1245–1250, 2006.
P. Ratanaworabhan, J. Ke, and M. Burtscher. Fast Lossless Compression of Scientific Floating-Point Data. In DCC, pages 133–142. IEEE Computer Society, 2006.

- Classic space/time tradeoff
- 3:1 lossless compression – 10:1 is possible

Write Patterns



Write Patterns



Read Patterns

1. Human interaction and exploration of the data
2. Analytics
3. Training ML/DL Patterns

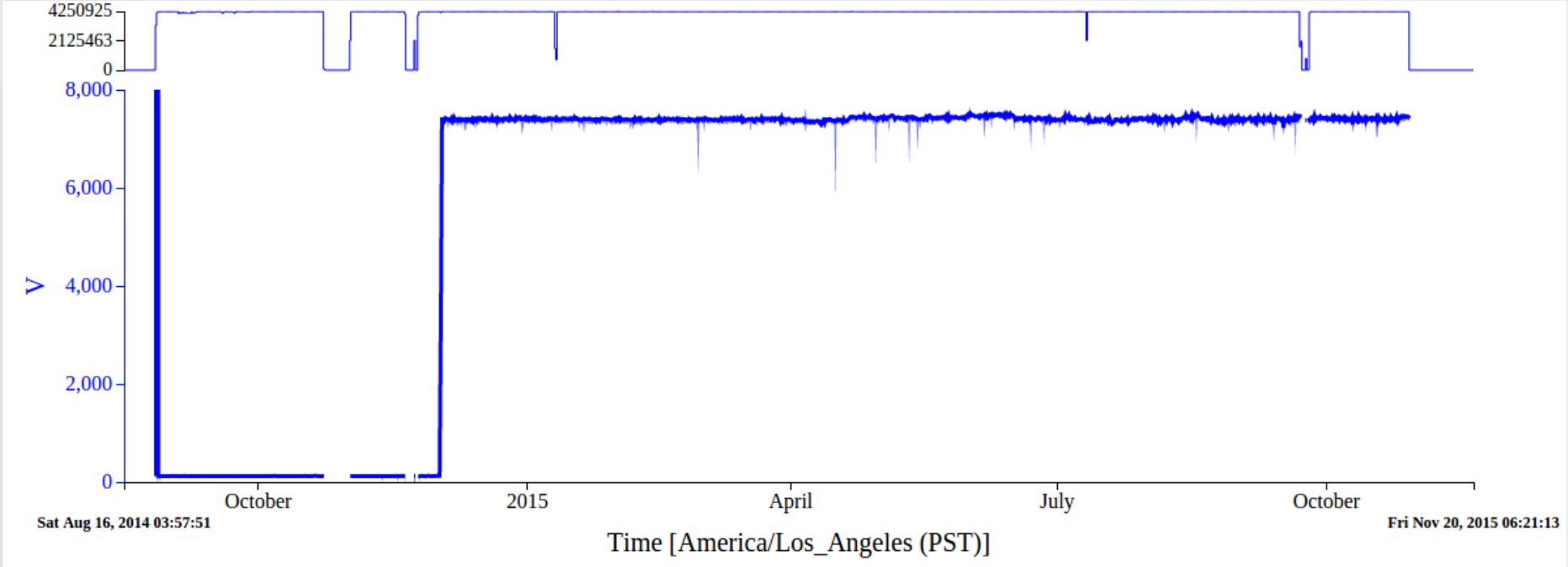
Read Patterns

“Overview first, zoom and filter, then details-on-demand.”

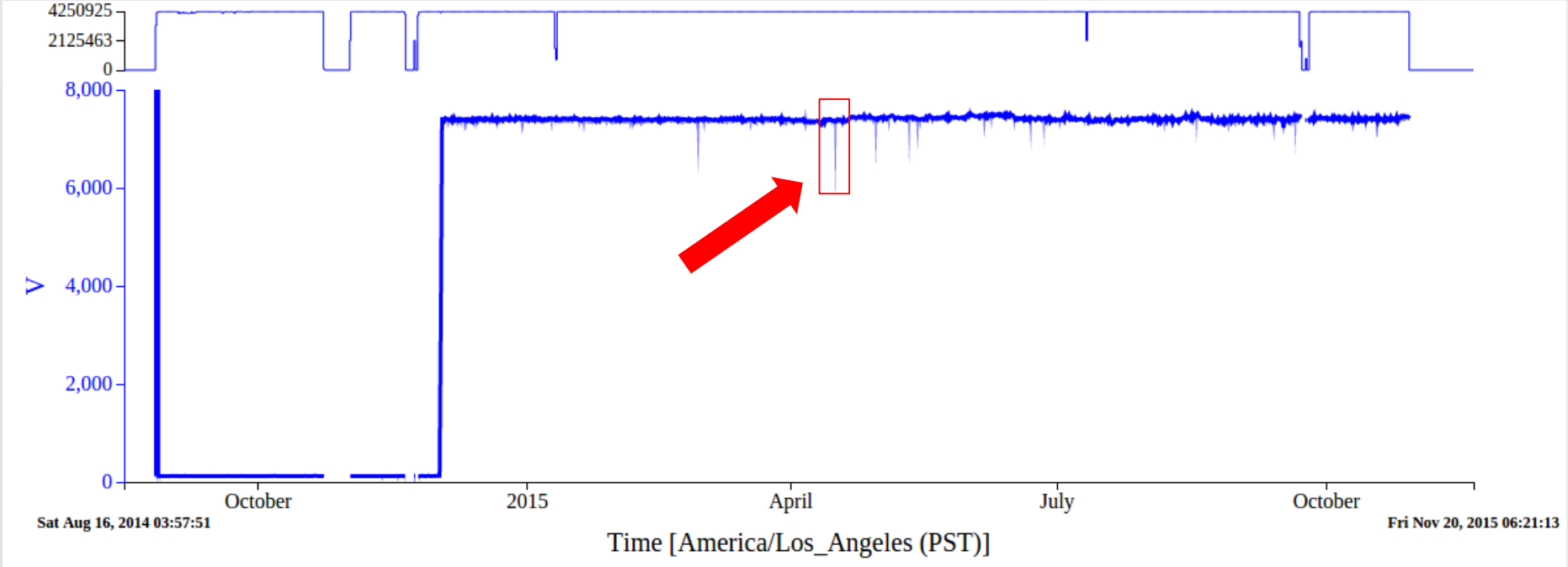
The Visual Information-Seeking Mantra [Shneiderman, 1996] summarizes many visual design guidelines and provides an excellent framework for designing information visualization applications.

Random, Multi-Resolution Read Patterns

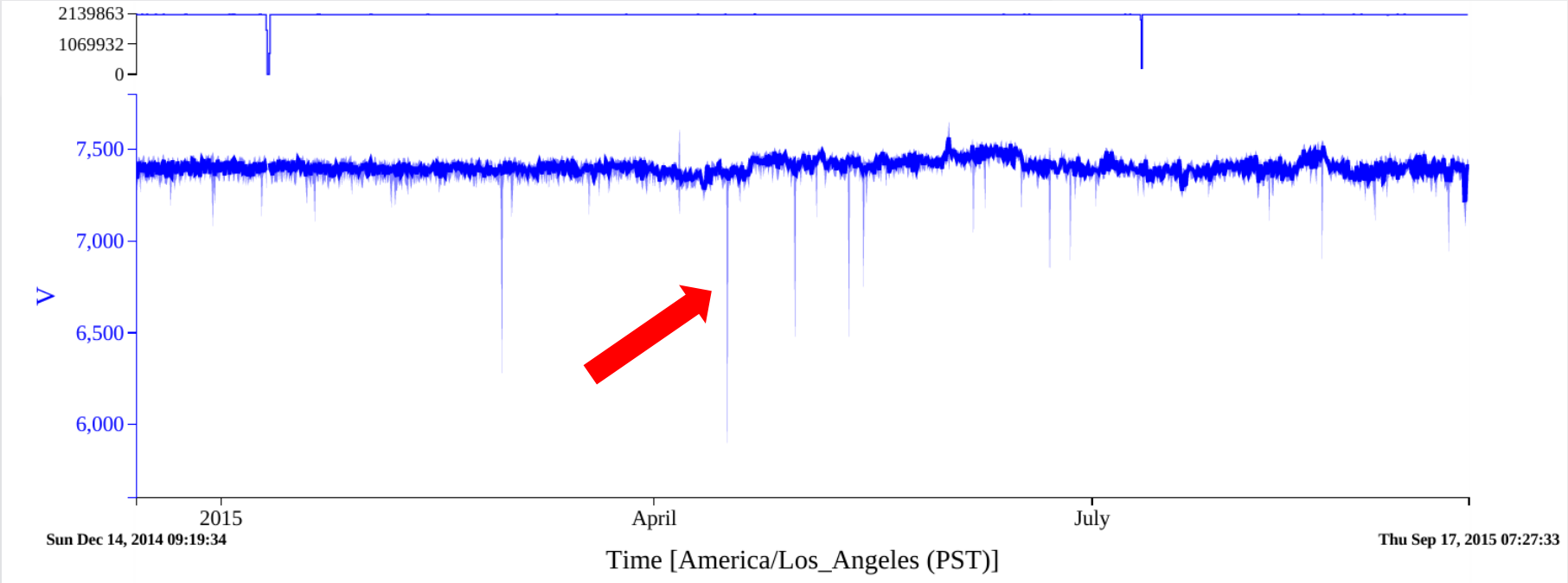
About 4 billion data points



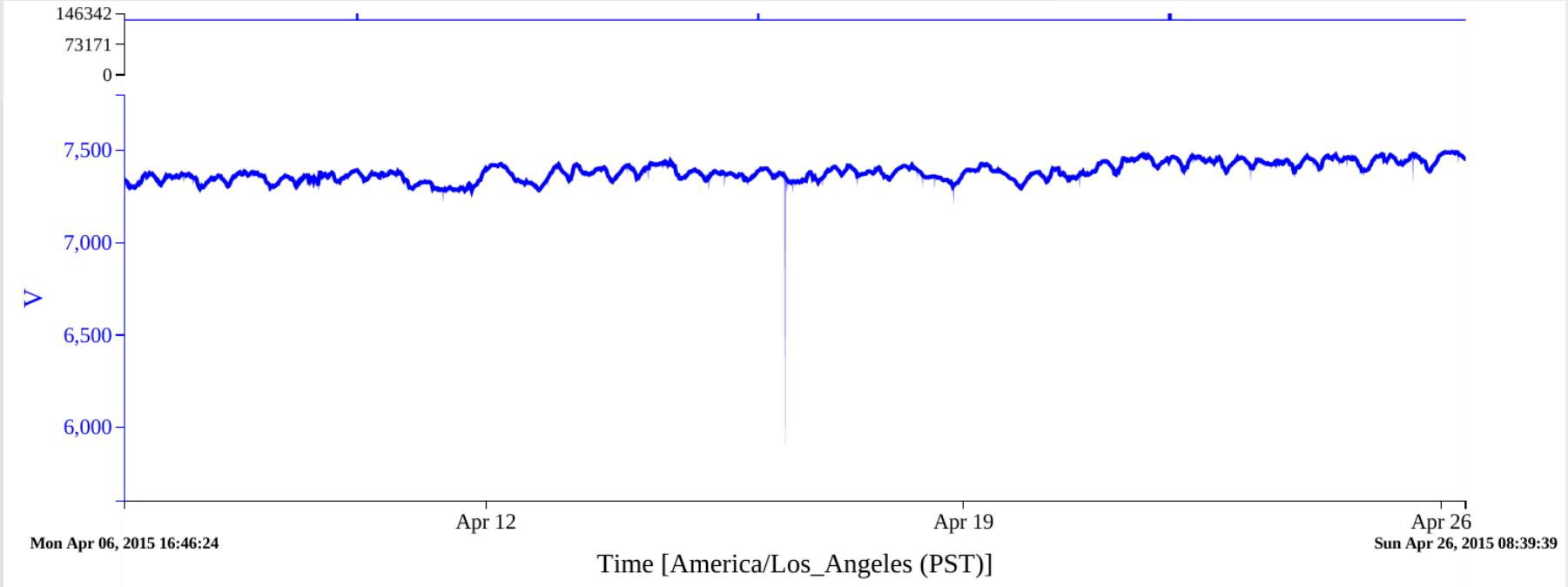
Random, Multi-Resolution Read Patterns



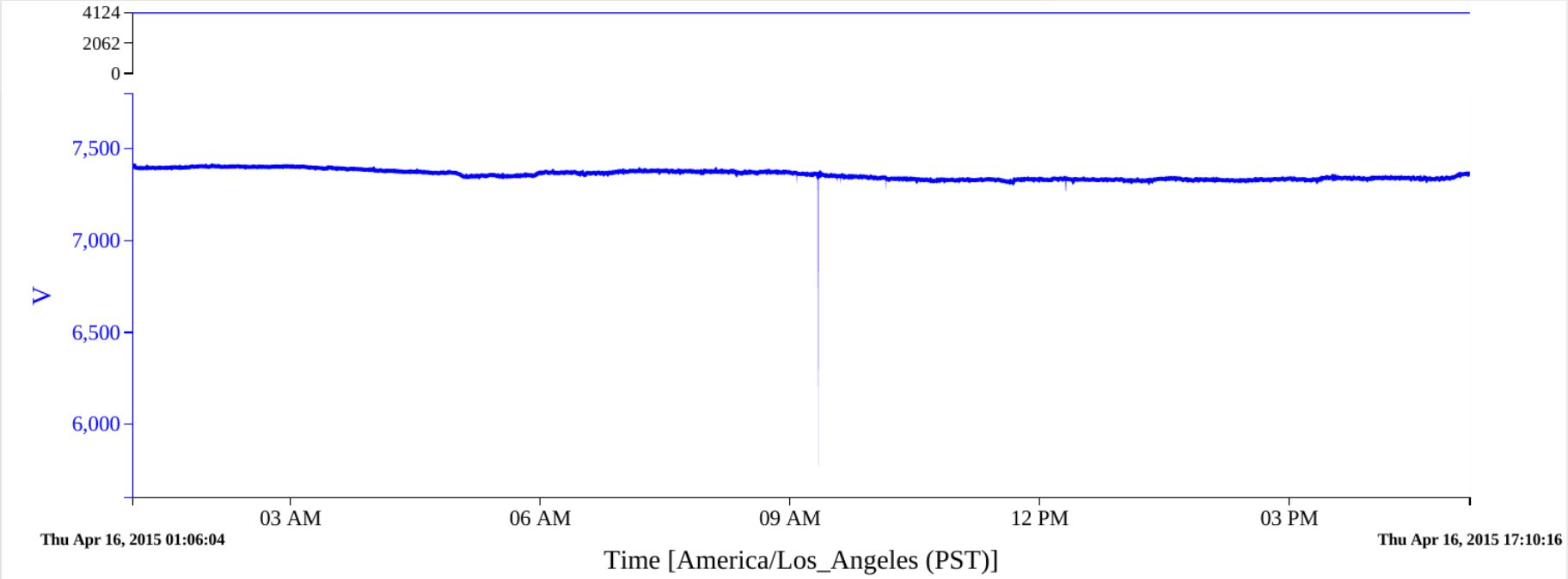
Random, Multi-Resolution Read Patterns



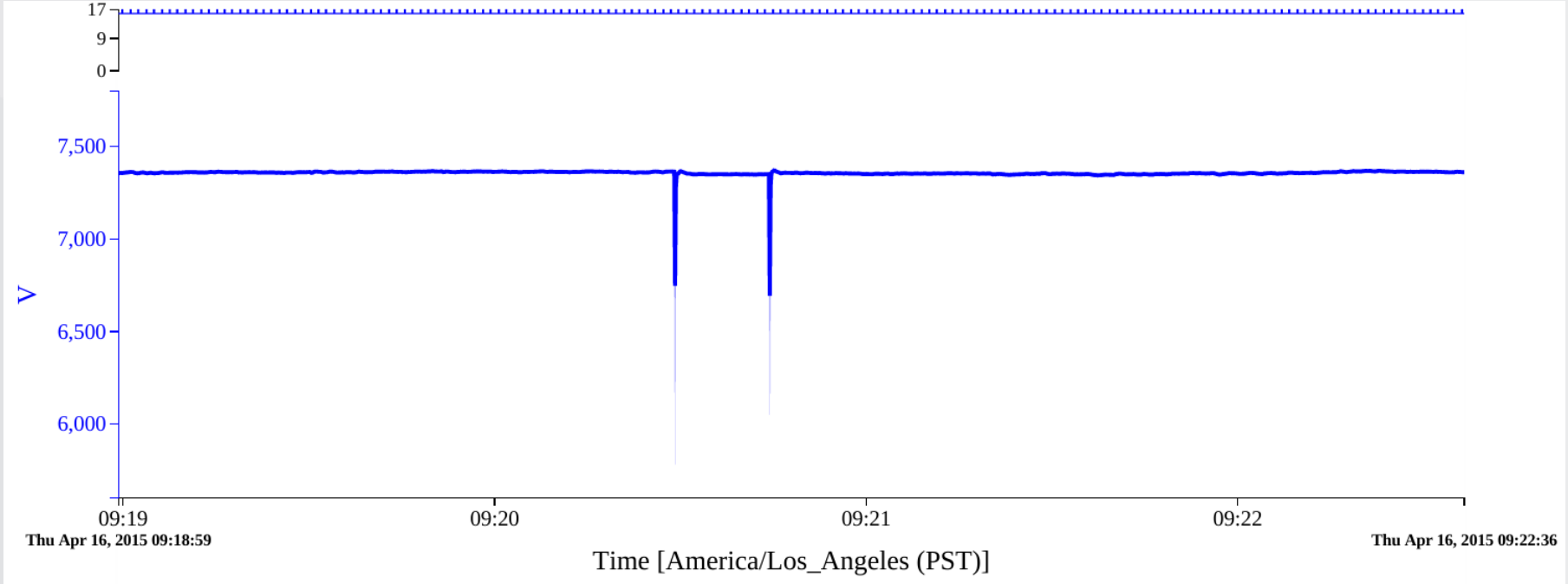
Random, Multi-Resolution Read Patterns



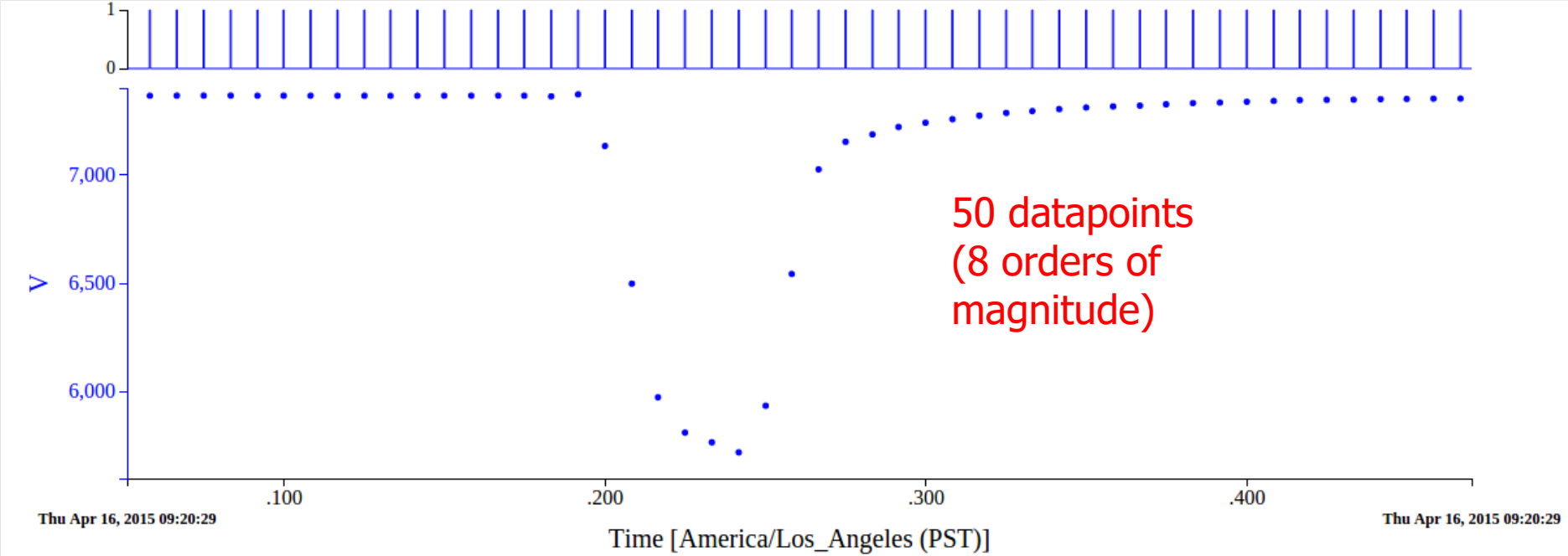
Random, Multi-Resolution Read Patterns



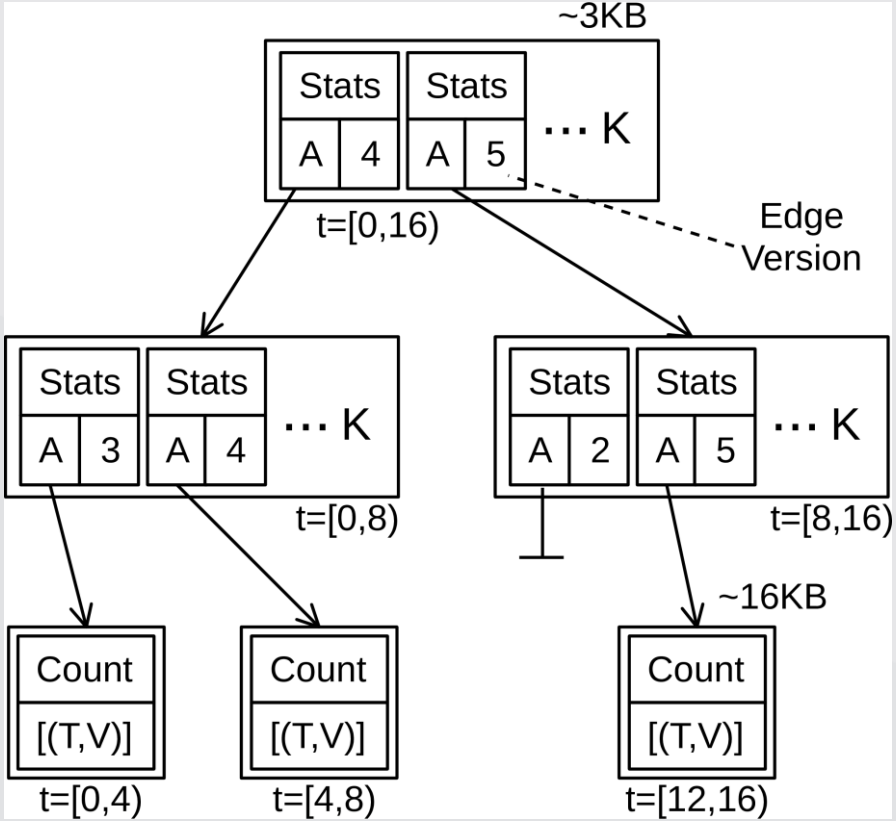
Random, Multi-Resolution Read Patterns



Random, Multi-Resolution Read Patterns



Berkeley Tree Data Structure



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

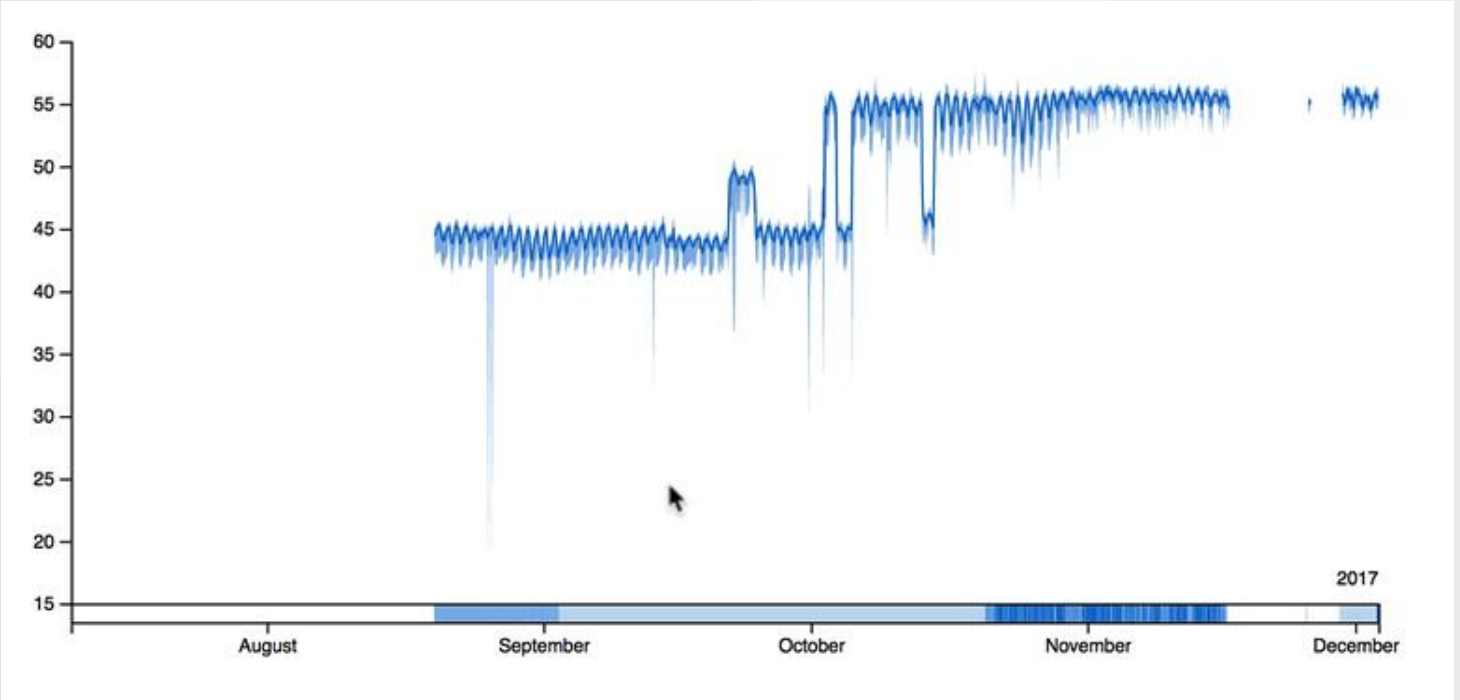
Leaf nodes

- Time, value pairs + length

Internal nodes

- Pointers to children
- **Version annotations** for children
- **Aggregates** for children
  - Min, Mean, Max, Count
  - Any associative operator

Did it Work?


60
55
50
45
40
35
30
25
20
15
2017
August
September
October
November
December

Who Cares?

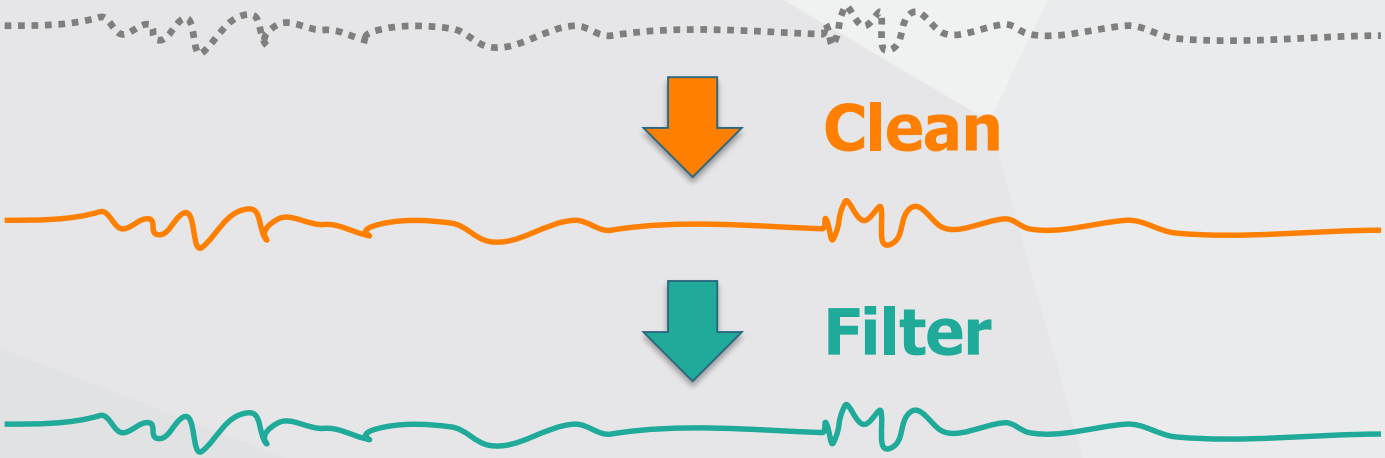
- Supports heterogeneous, multi-scale, multi-resolution sensor data
- Supports up to 1GHz sensors
- Nanosecond time resolution
- Write 53M points/second
- Read 120M points/second
- 10x Faster than anything else available
- Tested beyond 100,000 PMUs

Analytics as First Class Citizens

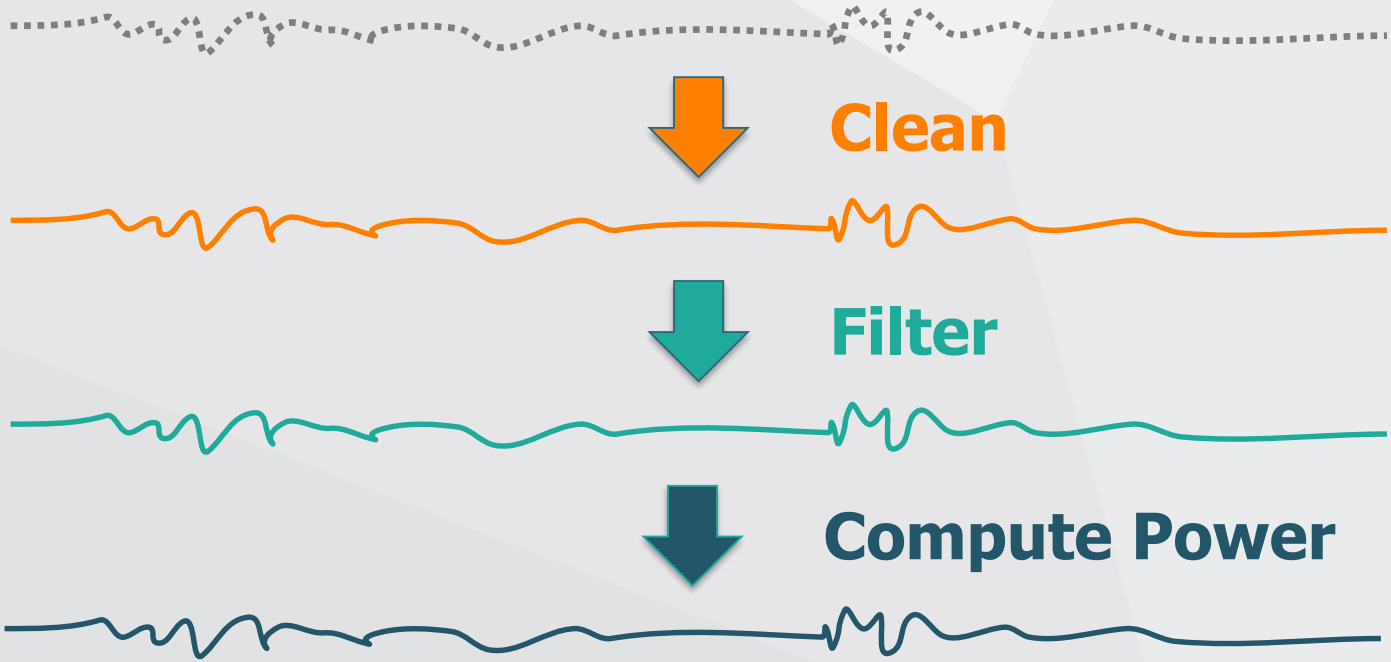
Common Analytics Patterns

Common Analytics Patterns

Common Analytics Patterns

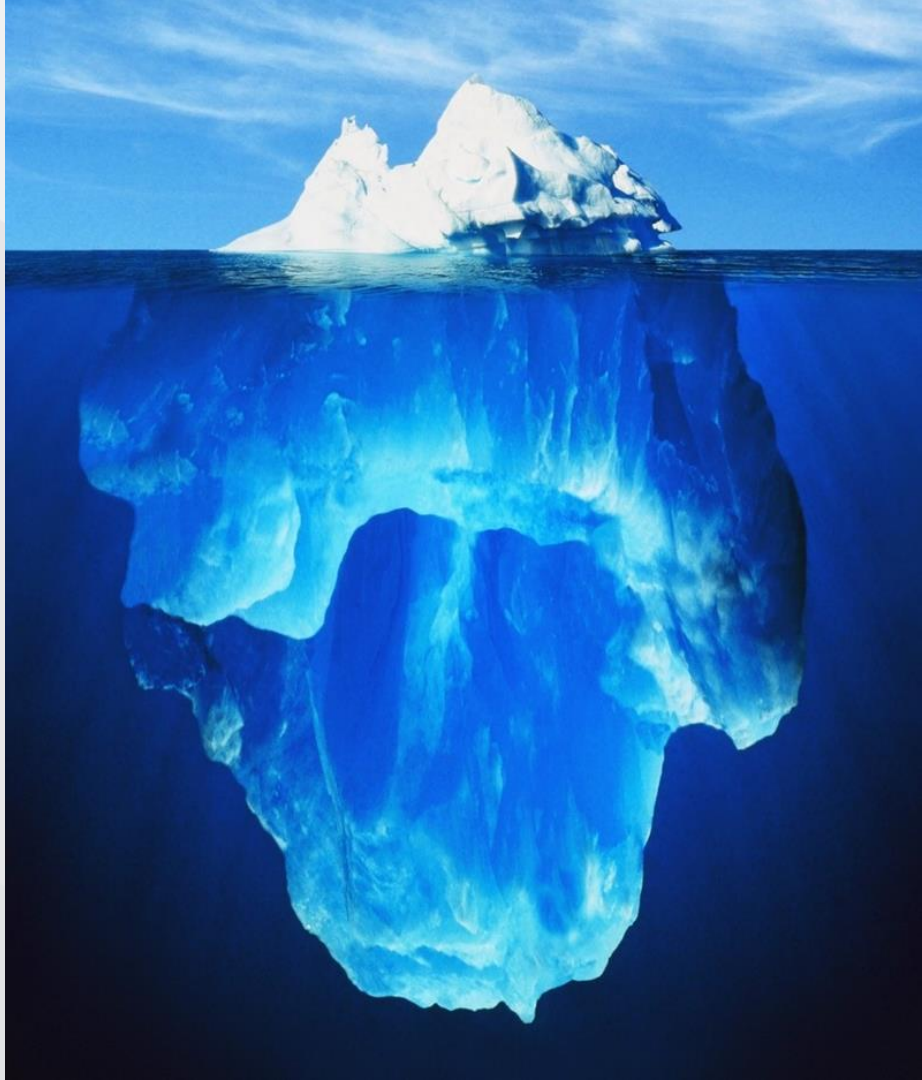


Common Analytics Patterns

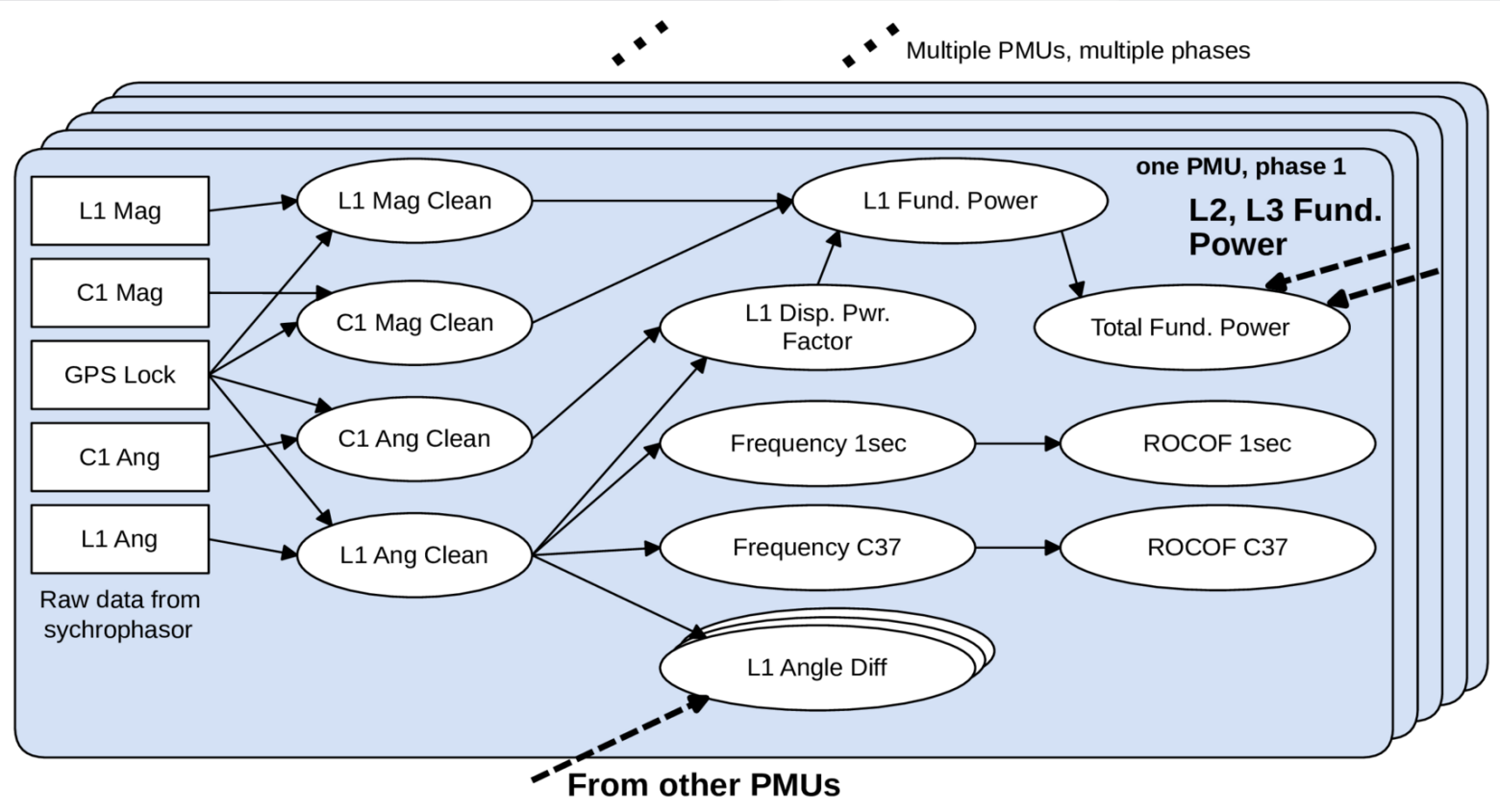


But Wait, There's More

- Windowing operations
- Spatial/Frequency Transforms
  - Wavelet
  - Fourier
  - Stockwell
- Indexing
- Clustering
- Classification
- Categorization
- Anomaly/Event/Novelty Detection
- Motif Discovery



The DISTIL Framework



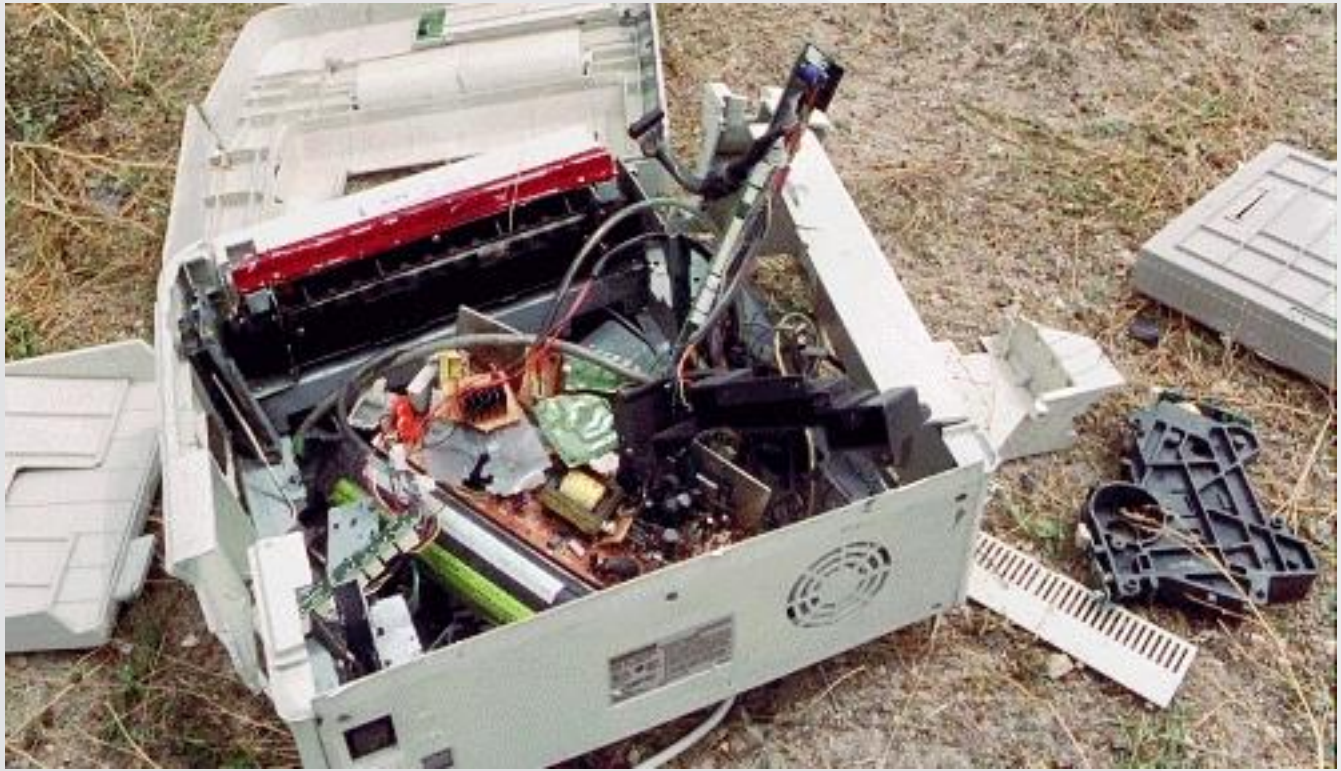
Analytics Benchmarks

	Distributed		
	Identity	Phase Difference	Reactive/Fundamental Pwr
Input/Output streams	1/1	2/1	4/2
Compute changeset	972 μs	1659μs	1180μs
Query data [s]	69.8	104.4	196.9
Kernel calculation [s]	10.8	22.7	245.5
Delete old data[s]	6.7	6.9	15.8
Insert new data[s]	40.7	39.8	66.5
Changeset / compute time	1064 x	773 x	259 x

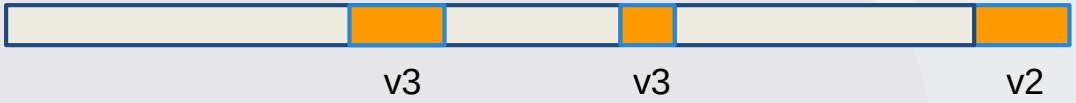
Who Cares?

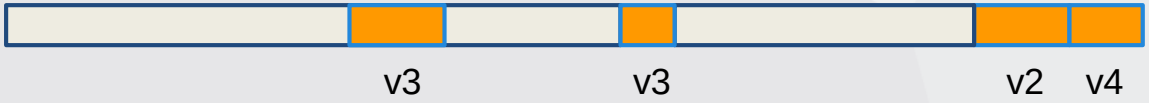
- The platform must be built from the ground up to support relevant analytic use cases
- Why capture sensor data if you aren’t going to use it?

The Real World is Messy

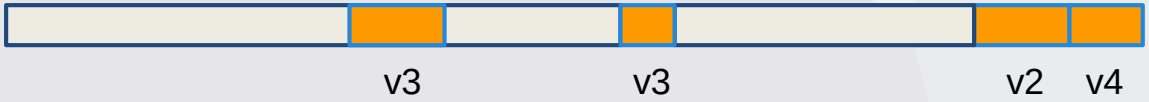



v2

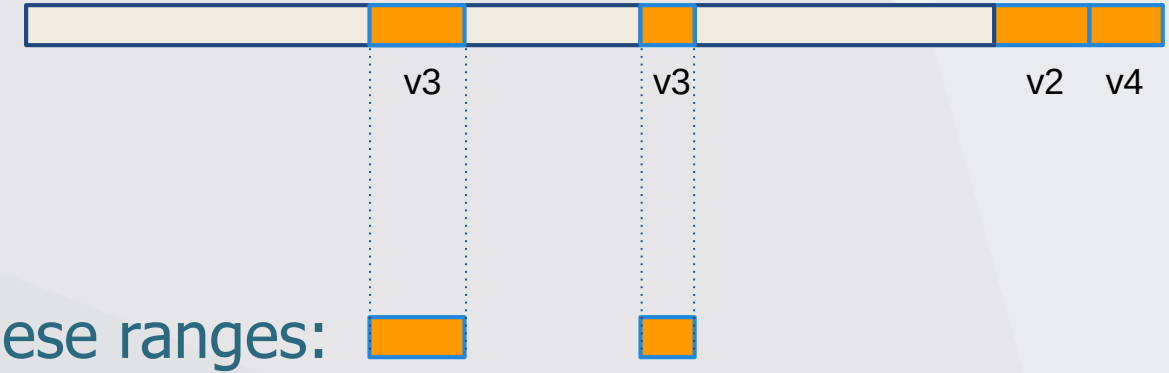

v3
v3
v2


v3
v3
v2
v4

Q: What changed between v2 and v3?

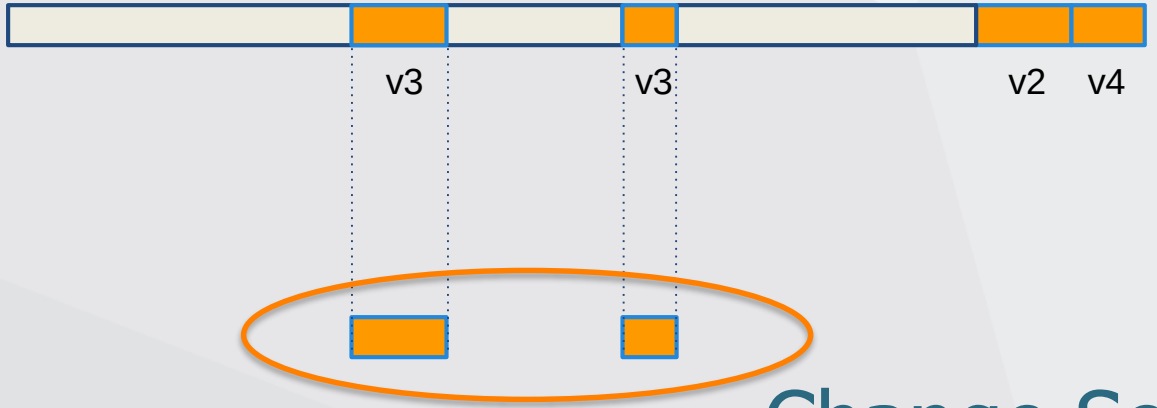


Q: What changed between v2 and v3?



A: These ranges:

Q: What changed between v2 and v3?



Change Set

Versioning Source Code

5 internal/cephprovider/cephprovider.go

```
@@ -148,12 +148,15 @@ func (seg *CephSegment) Write(uuid []byte, address uint64,
148 148         //start of an object. This is why we do not add the object max size here
149 149         //NEW NOTE:
150 150         //We cannot go past the end of the allocation anymore because it would br
151     -       if ((naddr + MAX_EXPECTED_OBJECT_SIZE) >> 24) != (address >> 24) {
    151 +       if ((naddr + MAX_EXPECTED_OBJECT_SIZE + 2) >> 24) != (address >> 24) {
152 152                 //We are gonna need a new object addr
153 153                 naddr = <-seg.sp.alloc
    154 +               seg.naddr = naddr
154 155                 seg.flushWrite()
    156 +               return naddr, nil
155 157         }
156 158         seg.naddr = naddr
    159 +
157 160         return naddr, nil
158 161 }
```

Who Cares?

- Efficiently update calculations on out of order data
- Idempotent calculations
- Rewinding data arrival to understand and diagnose problems

A Universal Sensor Analytics Platform for Utilities

- For heterogeneous, multi-scale, multi-resolution sensor data
- Open source, open source tools, open data formats
- Tested beyond 100,000 PMUs
- Each signal can be up to 1 Gigahertz
- Horizontally scalable and distributed
- Analytics as a first class citizen
- **Designed to make the humans better**

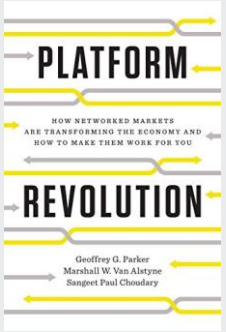
Jevons Paradox

In 1865, a twenty-nine-year-old Englishman named William Stanley Jevons published a book, "The Coal Question," in which he argued that the bonanza couldn't last. Britain's affluence, he wrote, depended on its endowment of coal, which the country was rapidly depleting. He added that such an outcome could not be delayed through increased "economy" in the use of coal—what we refer to today as energy efficiency. He concluded, in italics, "*It is wholly a confusion of ideas to suppose that the economical use of fuel is equivalent to a diminished consumption. The very contrary is the truth.*"*

> "occurs when technological progress increases the efficiency with which a resource is used (reducing the amount necessary for any one use), but the rate of consumption of that resource rises because of increasing demand"

https://www.newyorker.com/magazine/2010/12/20/the-efficiency-dilemma

Building a True Platform



"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."

The Punchline

- Free private beta for universities for the upcoming Academic Year
- Platform is pre-populated with TB of real synchrophasor data
- New sensor data is streaming into the platform

Sean Patrick Murphy
sean@pingthings.ai