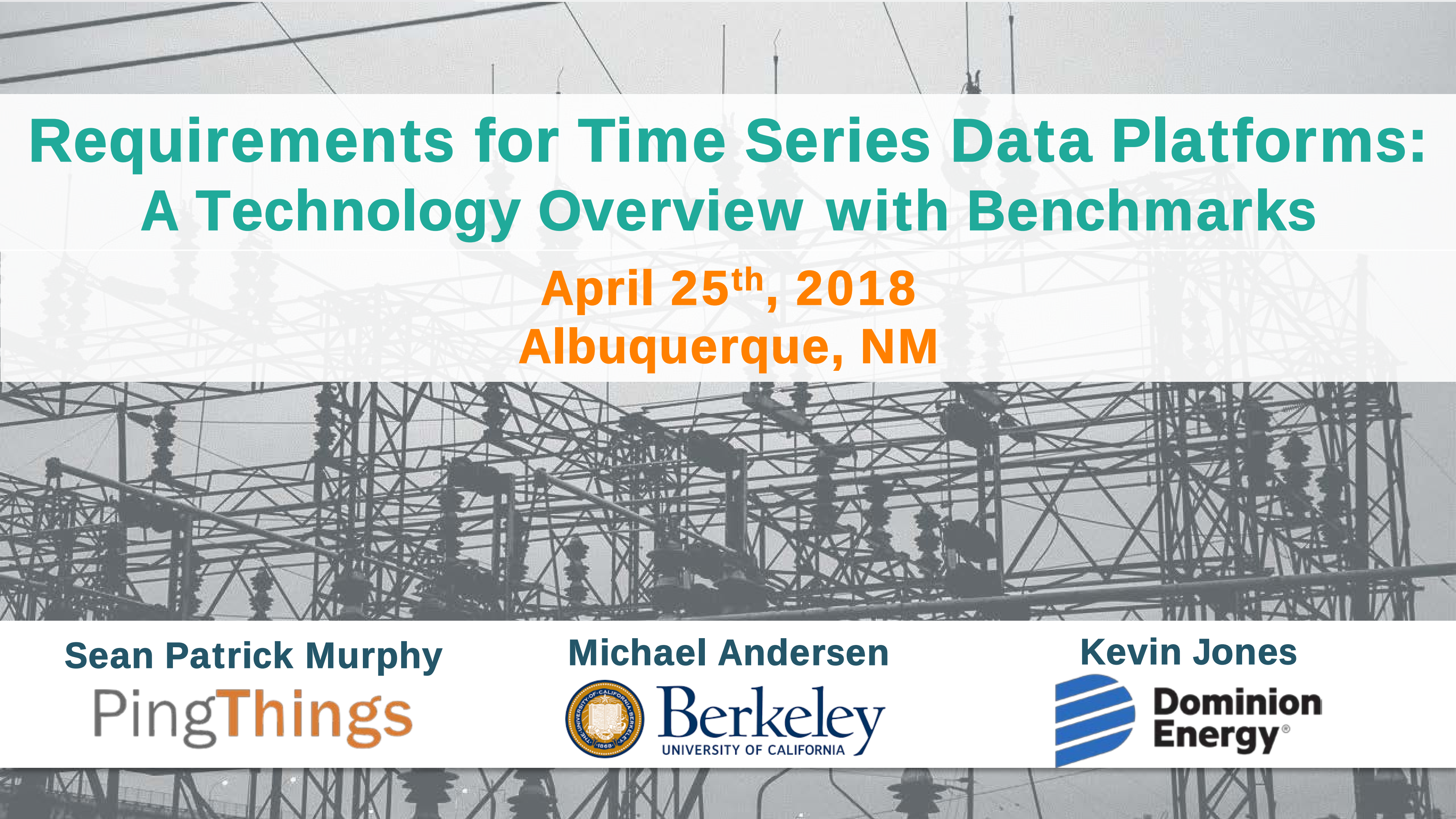


Requirements for Time Series Data Platforms: A Technology Overview with Benchmarks

April 25th, 2018
Albuquerque, NM

Sean Patrick Murphy
PingThings

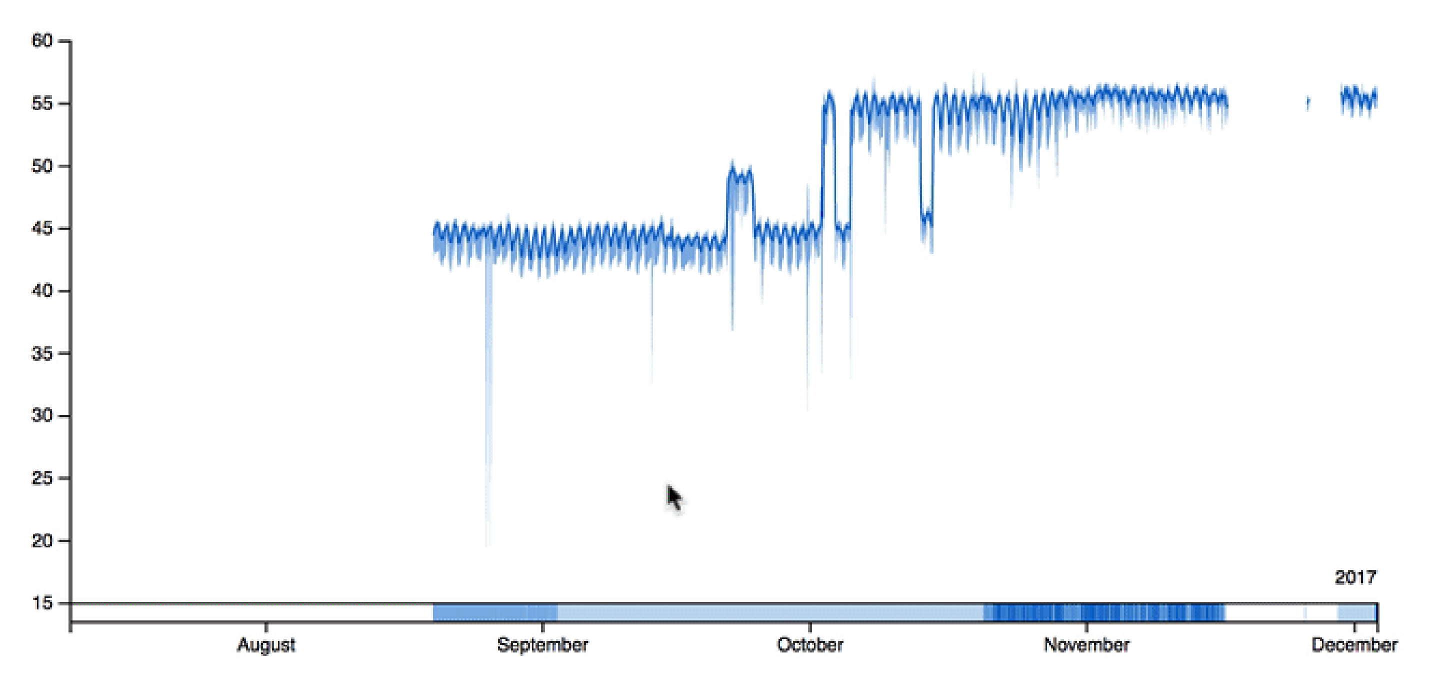
Michael Andersen
Berkeley
UNIVERSITY OF CALIFORNIA

Kevin Jones
Dominion Energy

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)
```

PingThings

`(timestamp, value)`

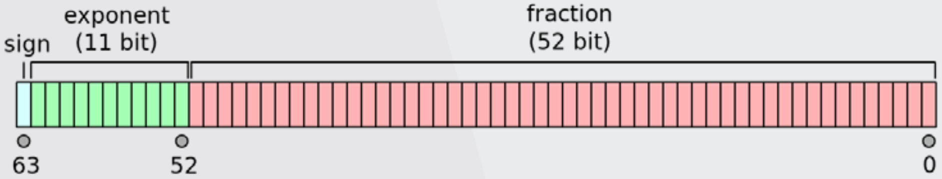
64-bit integer

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

Or

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

64-bit float



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

(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)
```

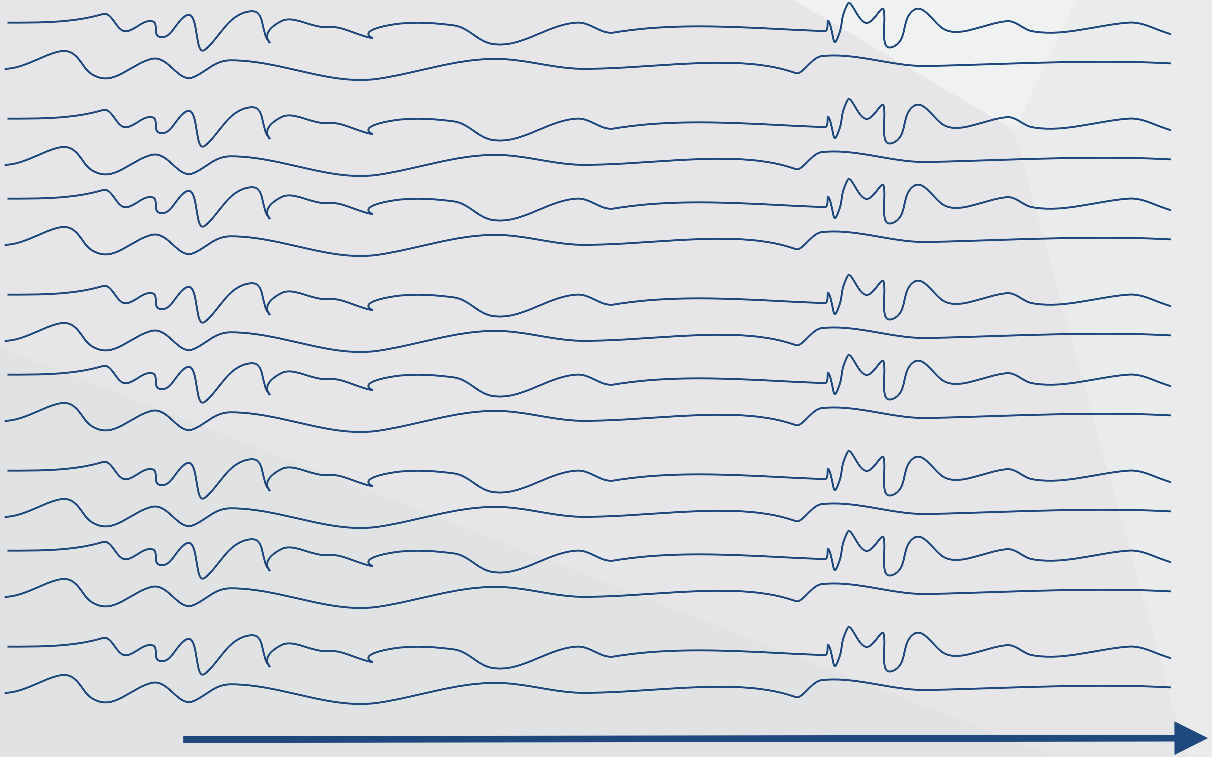
Intrinsic Redundancy

- 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
- We achieve ~3:1 lossless compression

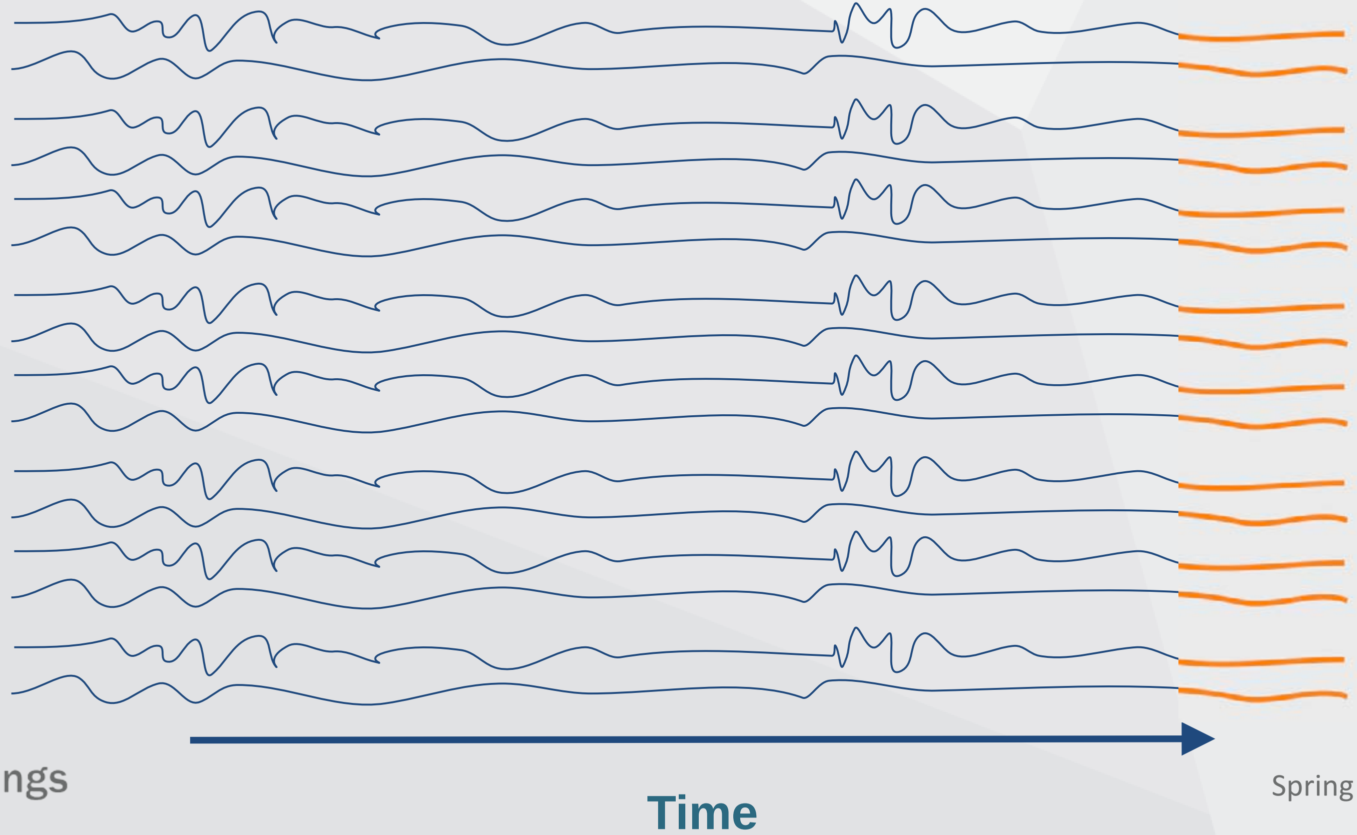
Write Patterns



Time

PingThings

Write Patterns



PingThings

Read Patterns

1. Human interaction and exploration of the data
2. Analytics

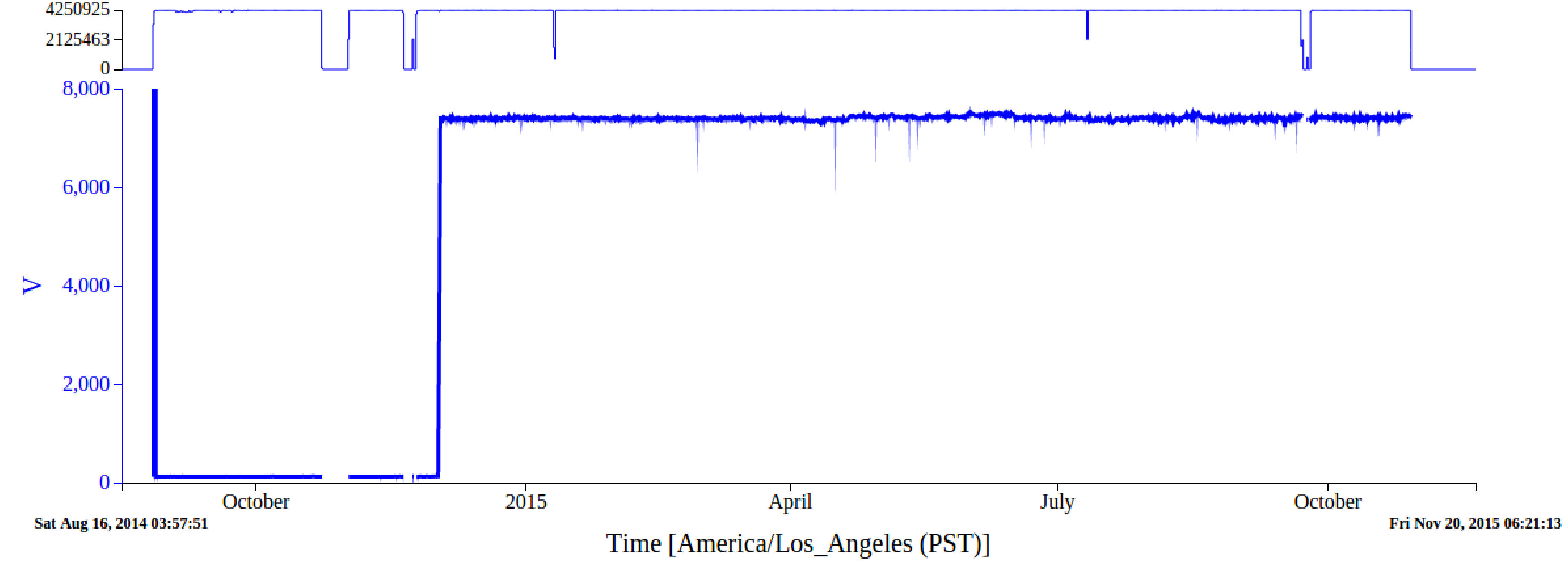
Read Patterns

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

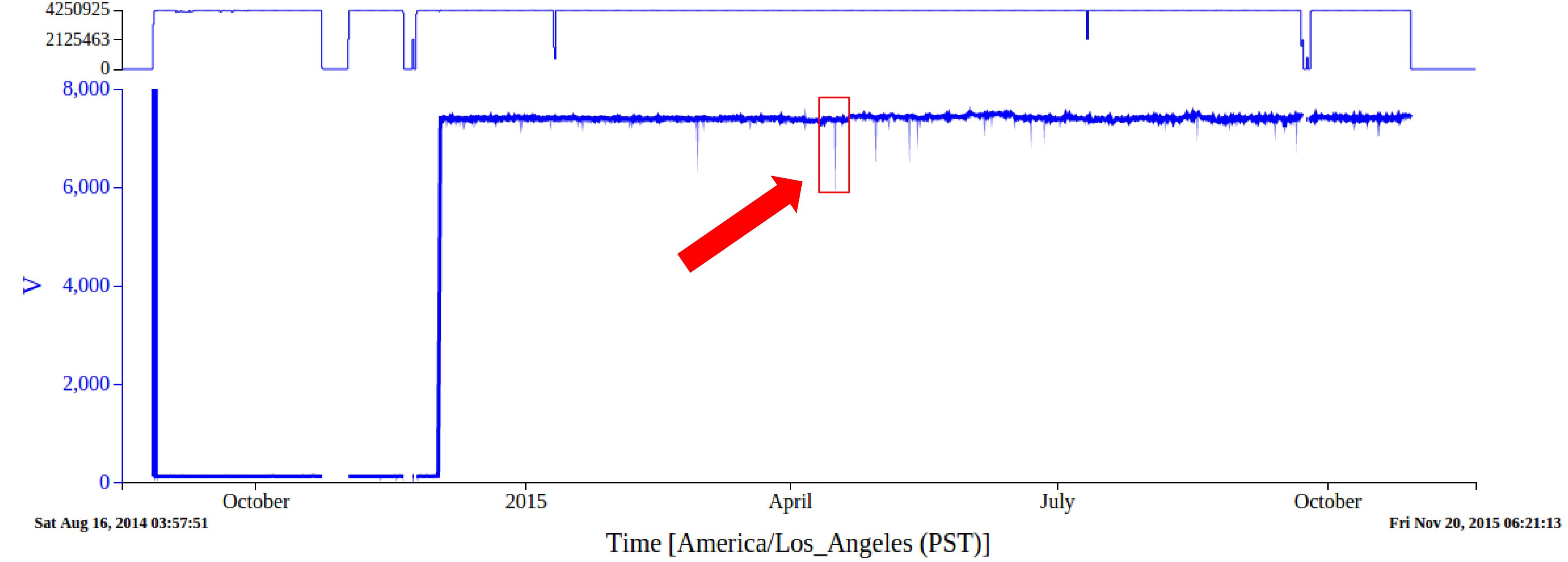
The Visual Information-Seeking Mantra [Shneiderman, 1996] summarizes many visual design guidelines and provides an excellent framework for designing information visualization applications.

Random, Multi-Resolution Read Patterns

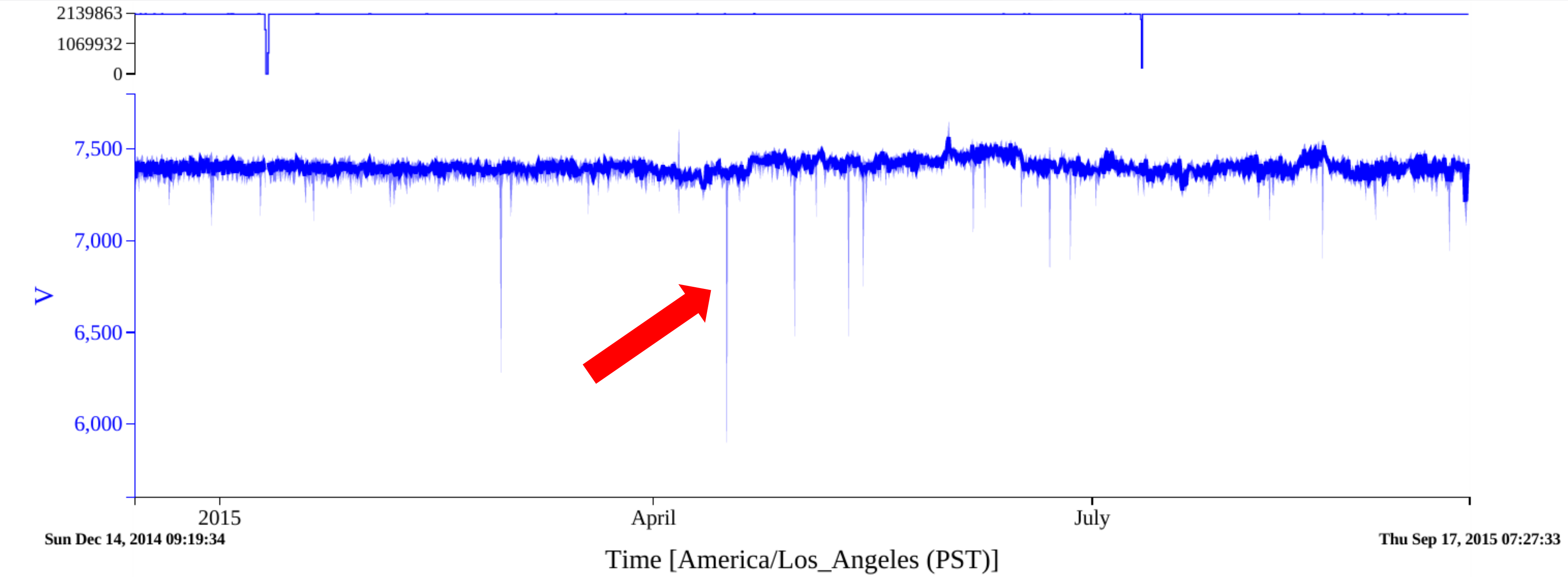
About 4 billion data points



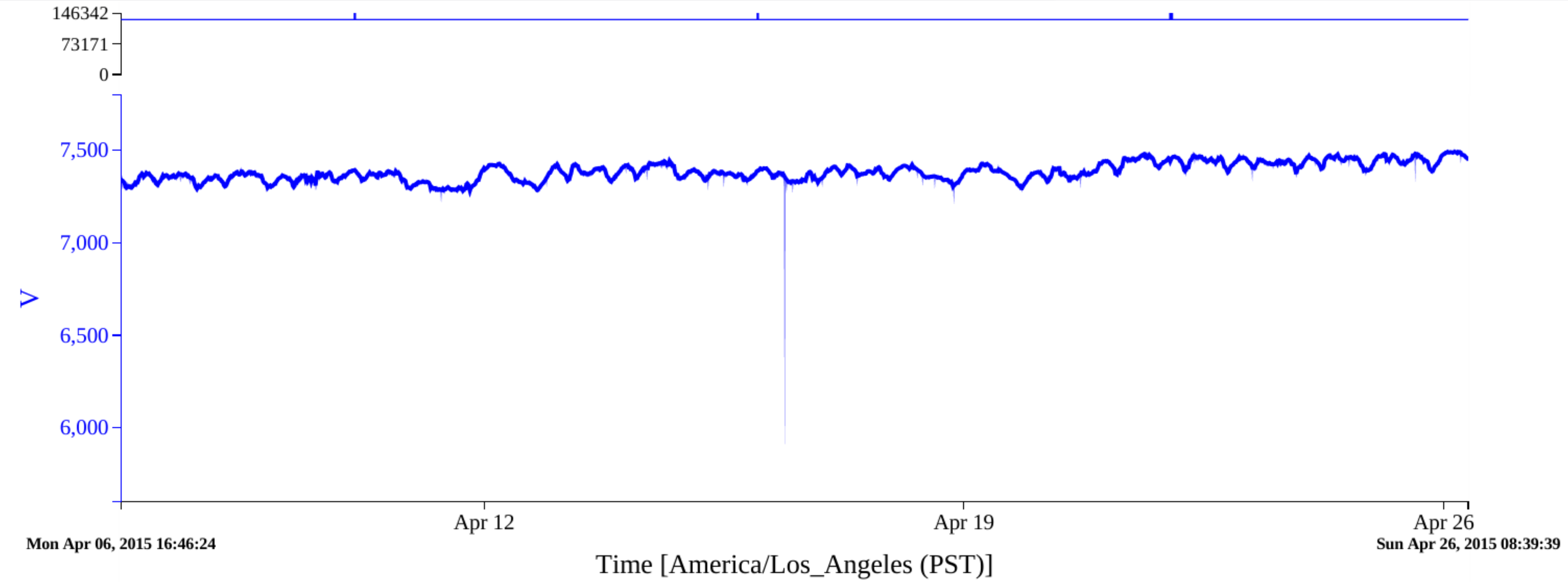
Random, Multi-Resolution Read Patterns



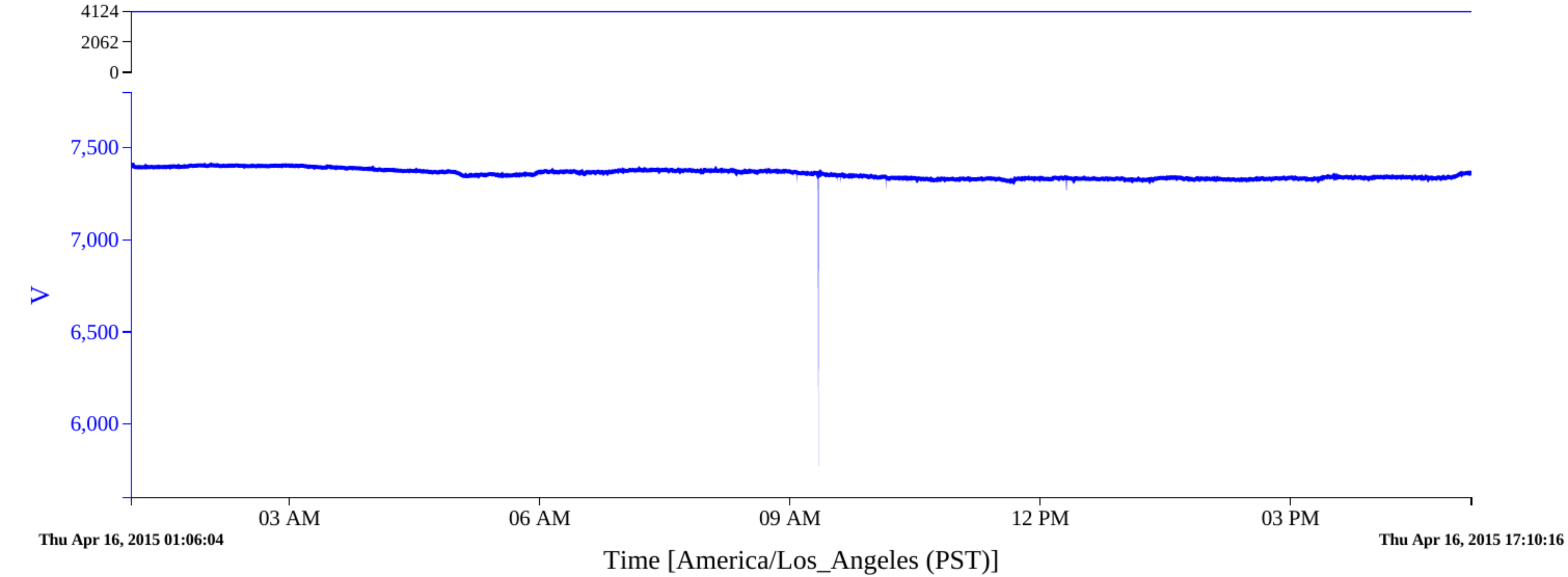
Random, Multi-Resolution Read Patterns



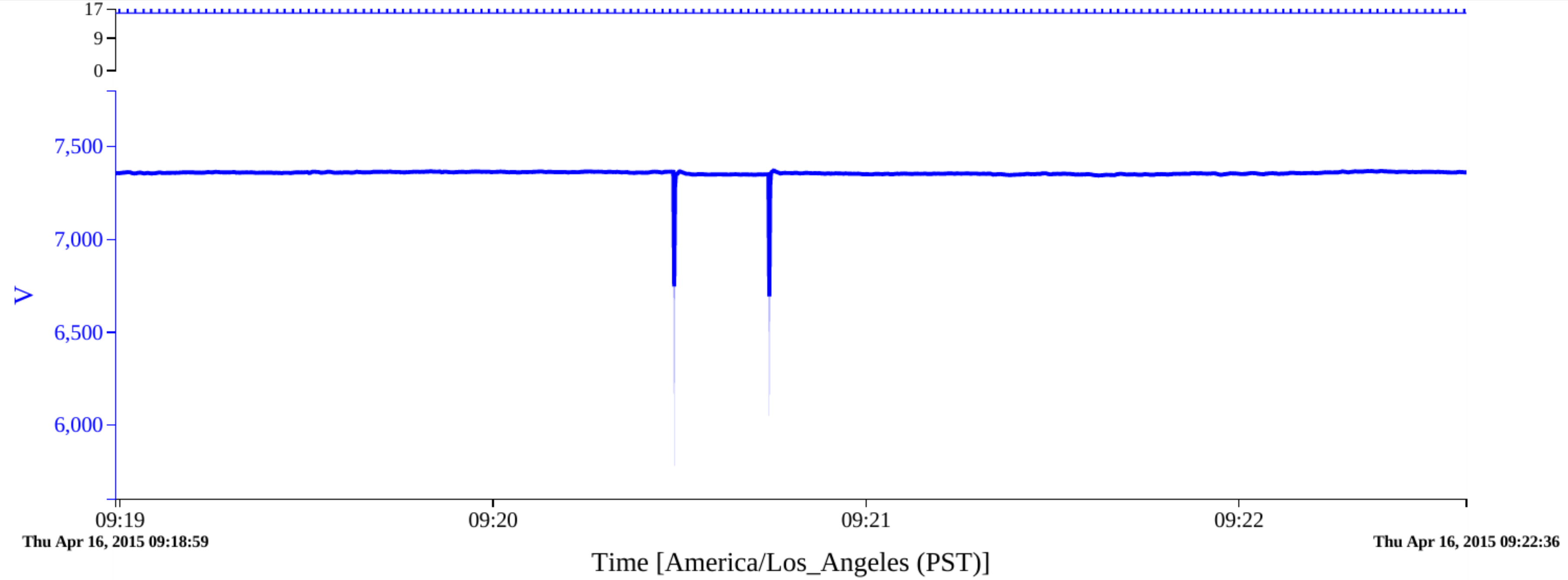
Random, Multi-Resolution Read Patterns



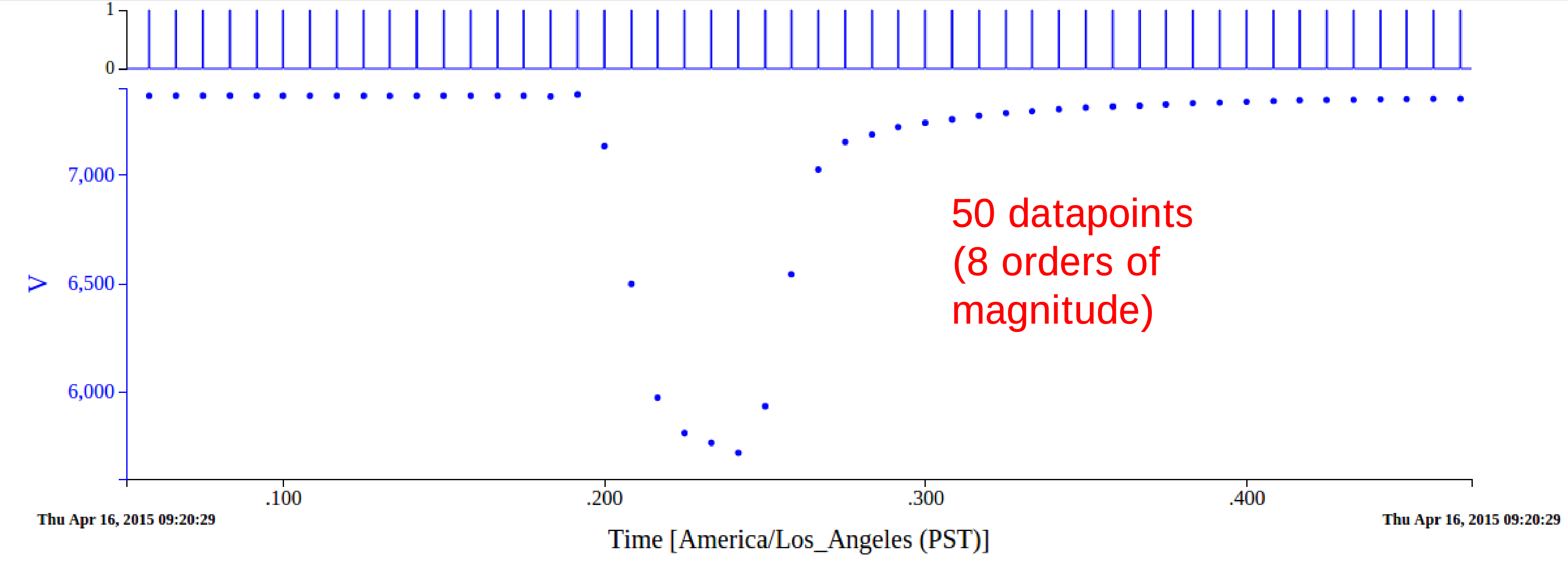
Random, Multi-Resolution Read Patterns



Random, Multi-Resolution Read Patterns

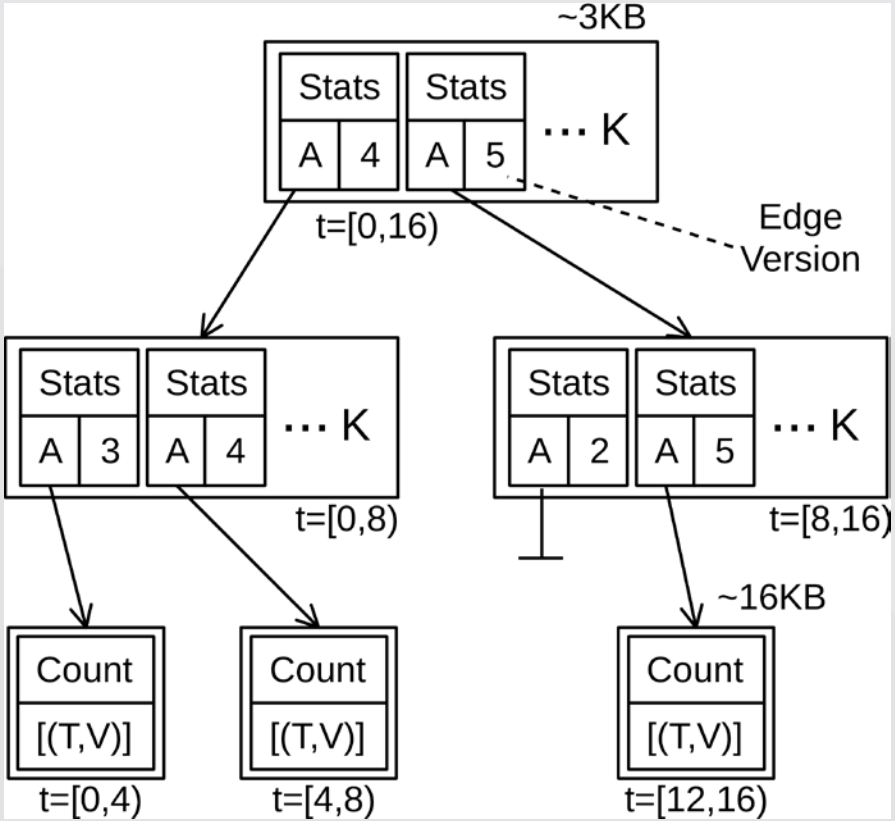


Random, Multi-Resolution Read Patterns



50 datapoints (8 orders of magnitude)

Berkeley tree Data Structure



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

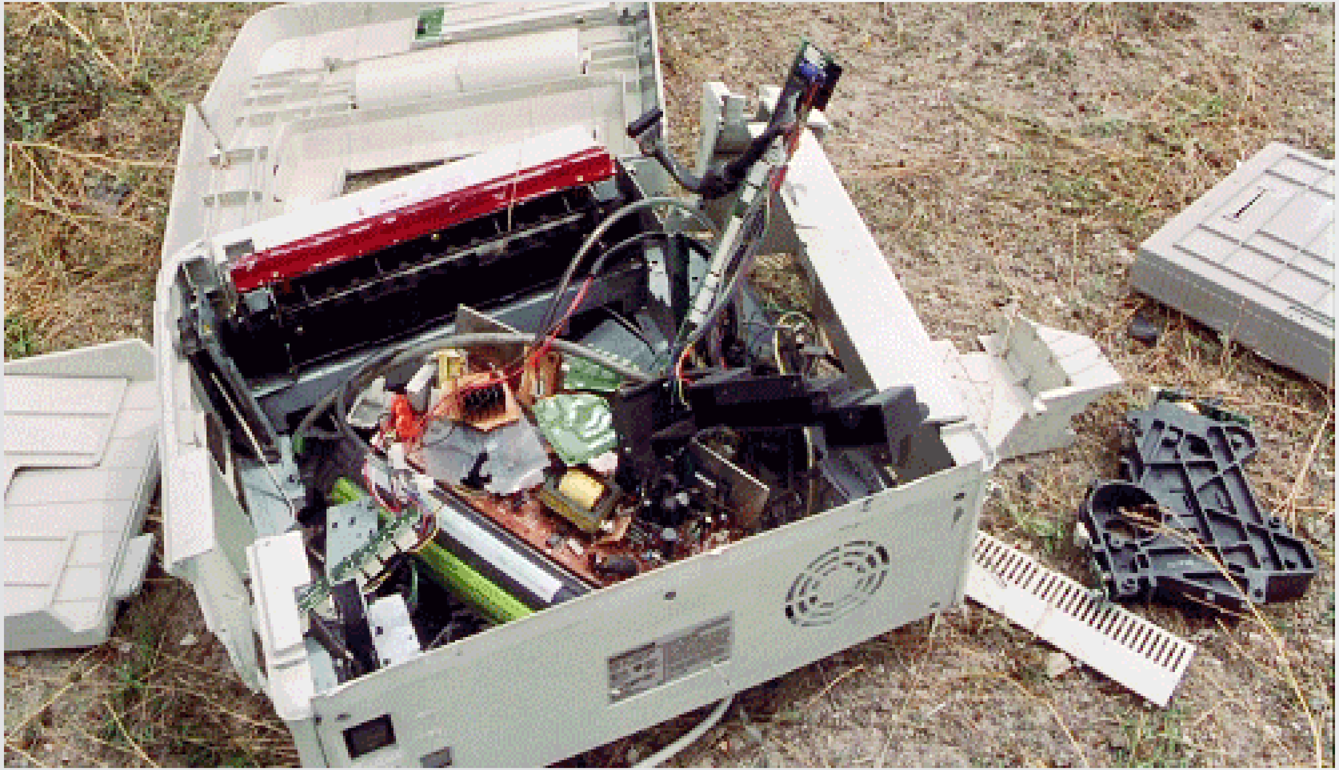
<u>Leaf nodes</u>

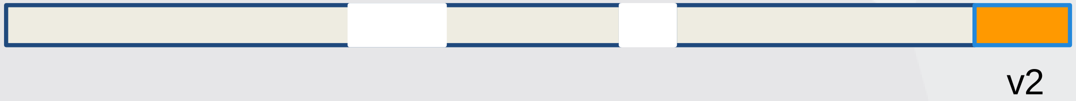
- Time, value pairs + length

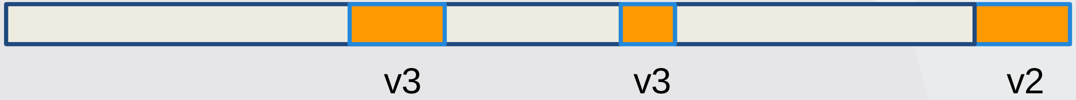
<u>Internal nodes</u>

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

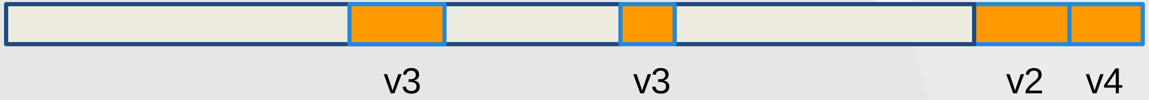
The Real World is Messy



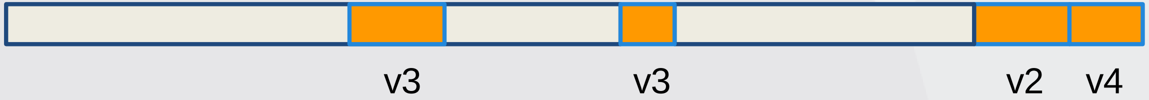

v2



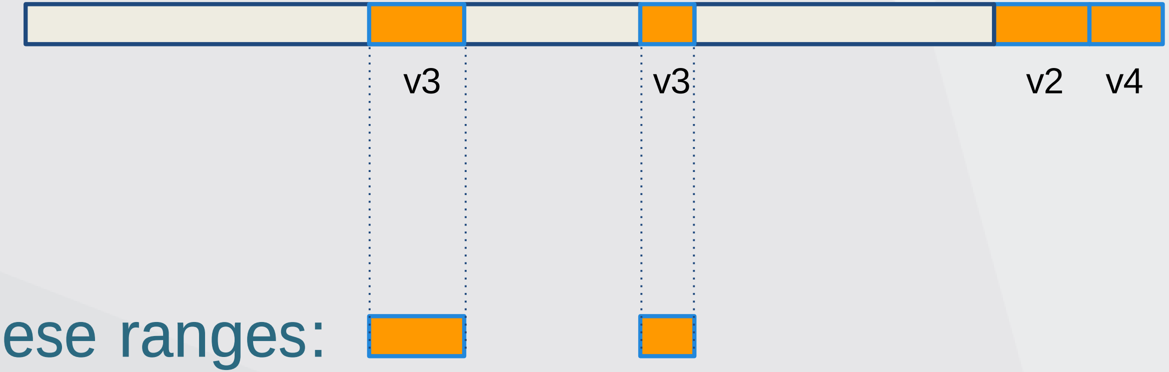
v3
v3
v2


v3
v3
v2
v4

Q: What changed between v2 and v3?

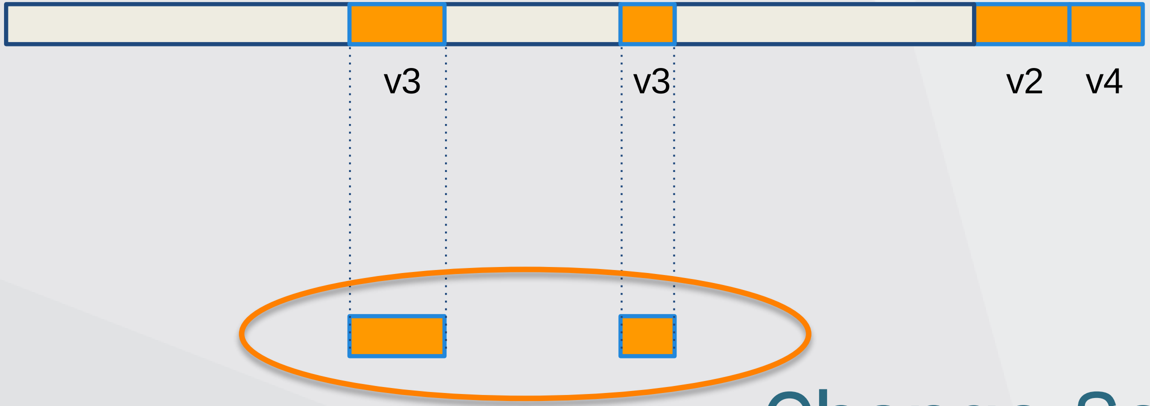


Q: What changed between v2 and v3?



A: These ranges:

Q: What changed between v2 and v3?



Change Set

Who Cares?

Who Cares?

- Efficiently update calculations on out of order data

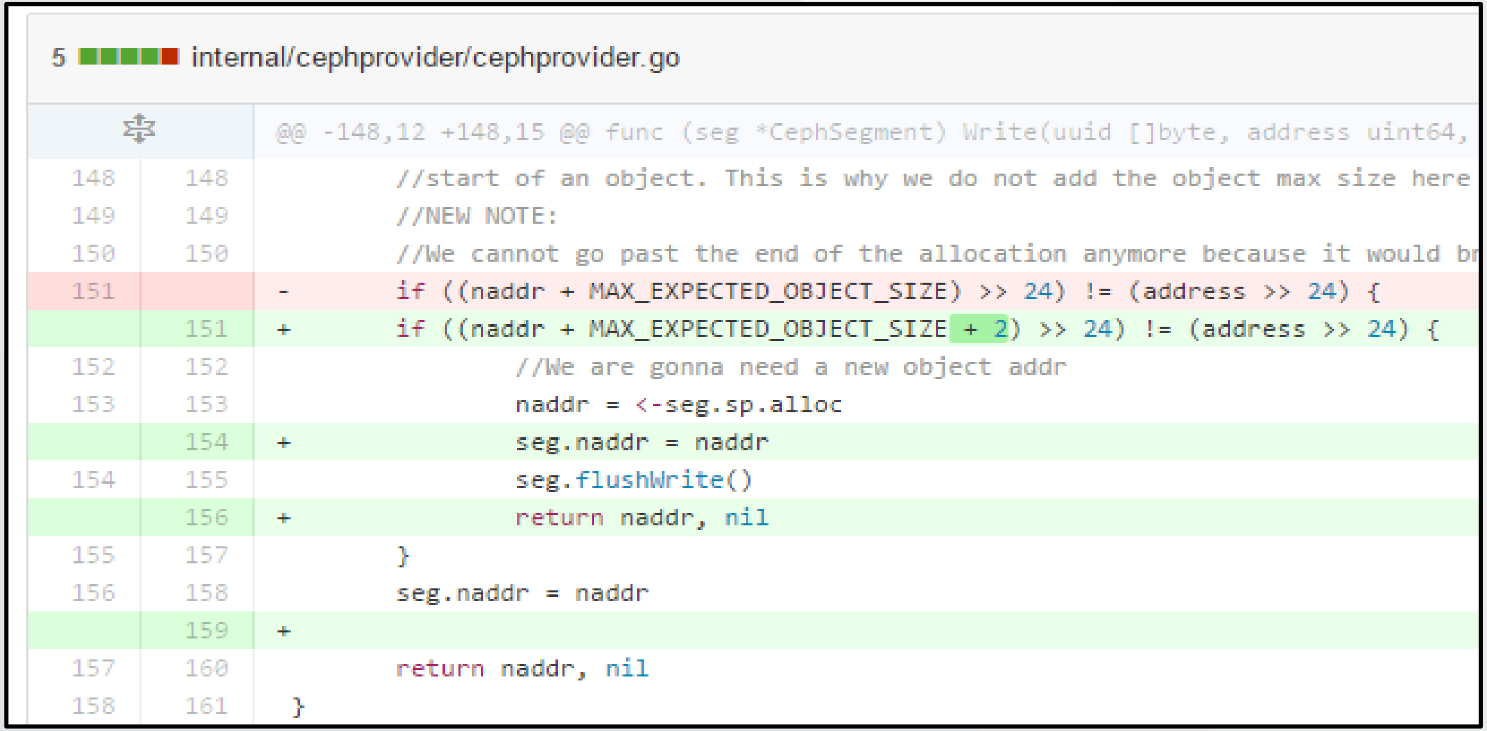
Who Cares?

- 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



5 🟩🟩🟩🟩🟥 internal/cephprovider/cephprovider.go

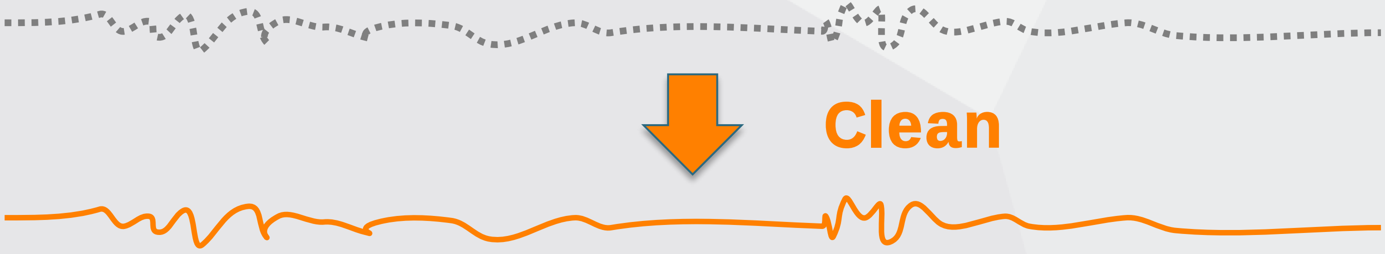
```diff
@@ -148,12 +148,15 @@ func (seg *CephSegment) Write(uuid []byte, address uint64,
 		//start of an object. This is why we do not add the object max size here
 		//NEW NOTE:
 		//We cannot go past the end of the allocation anymore because it would br
-		if ((naddr + MAX_EXPECTED_OBJECT_SIZE) >>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

Analytics as First Class Citizens

Common Analytics Patterns

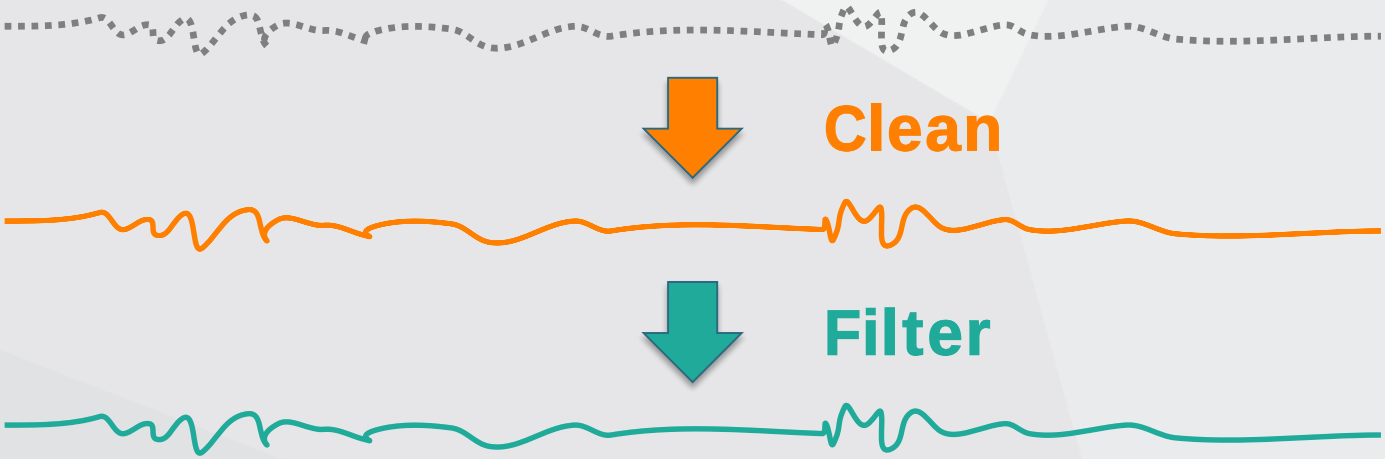
PingThings

Common Analytics Patterns

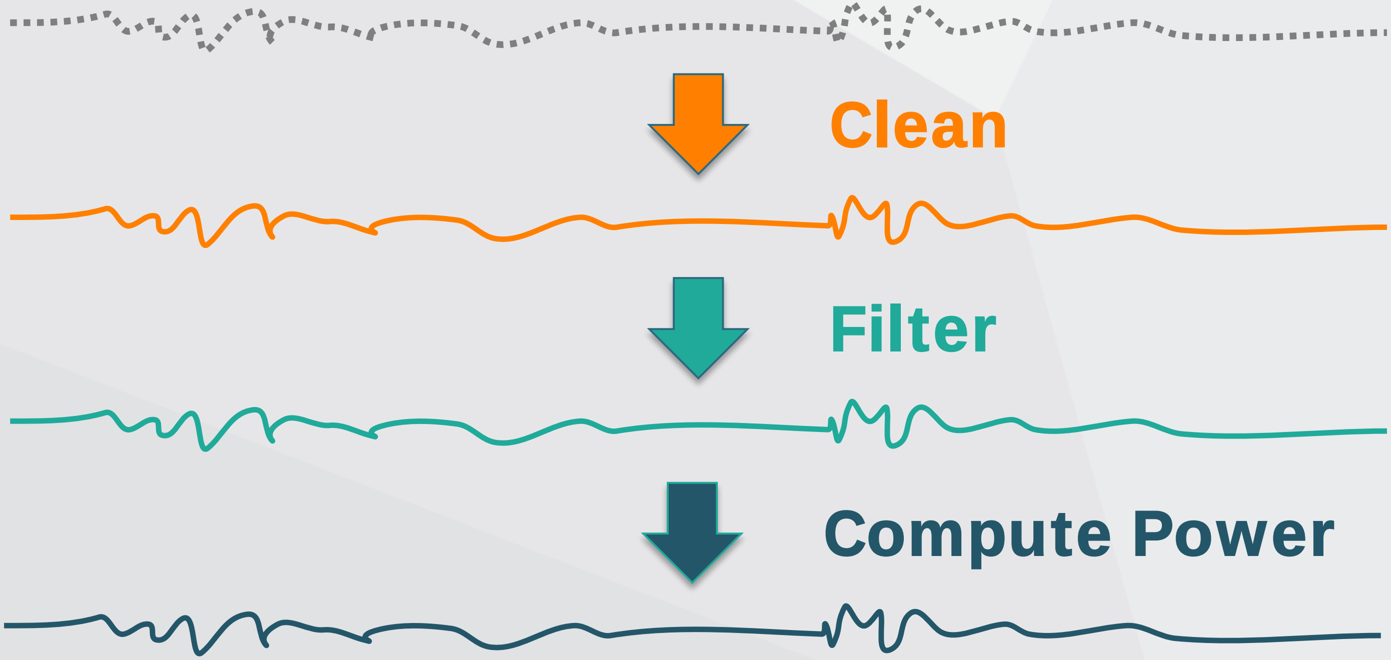


PingThings

Common Analytics Patterns



Common Analytics Patterns

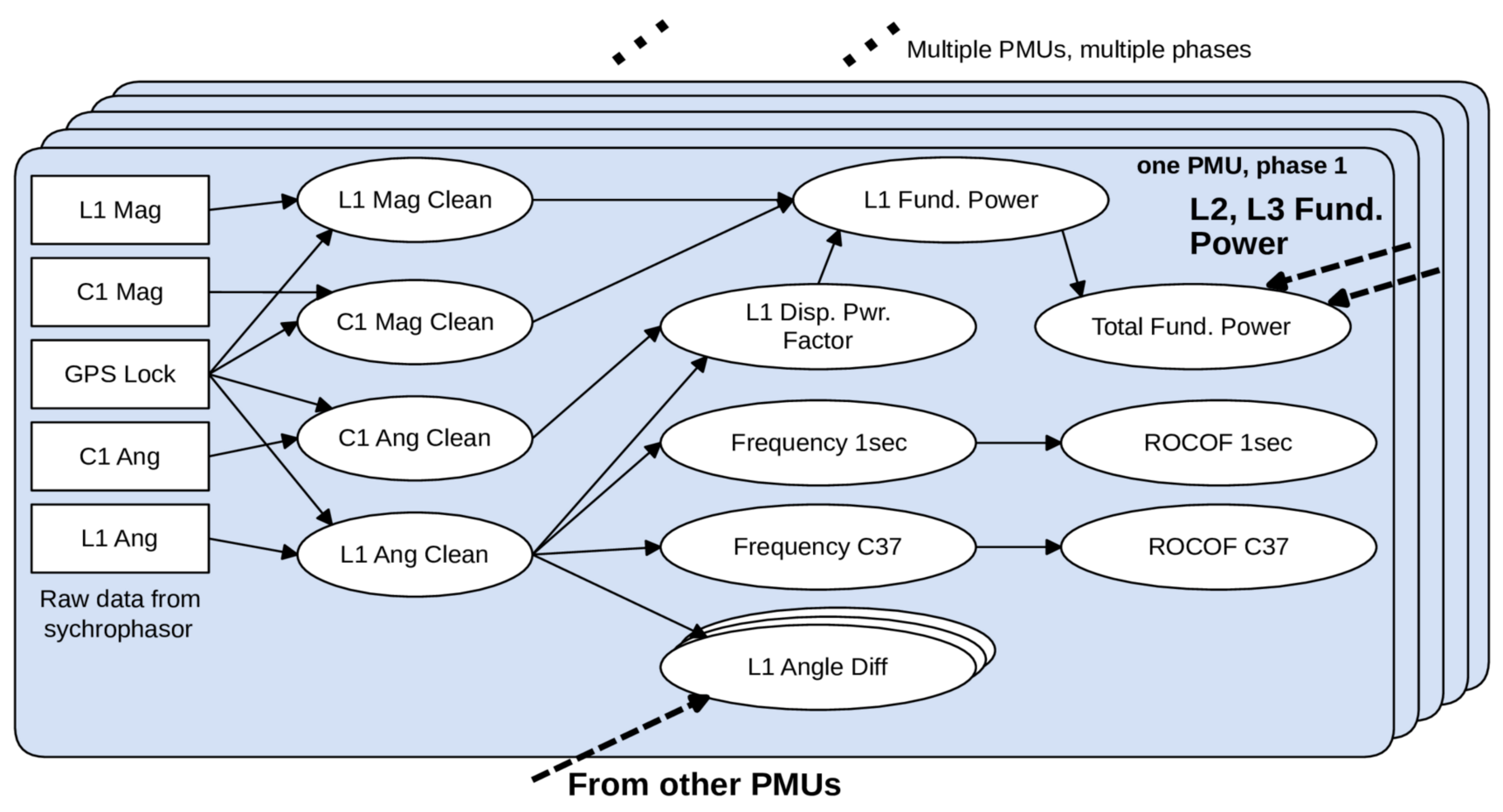


Clean

Filter

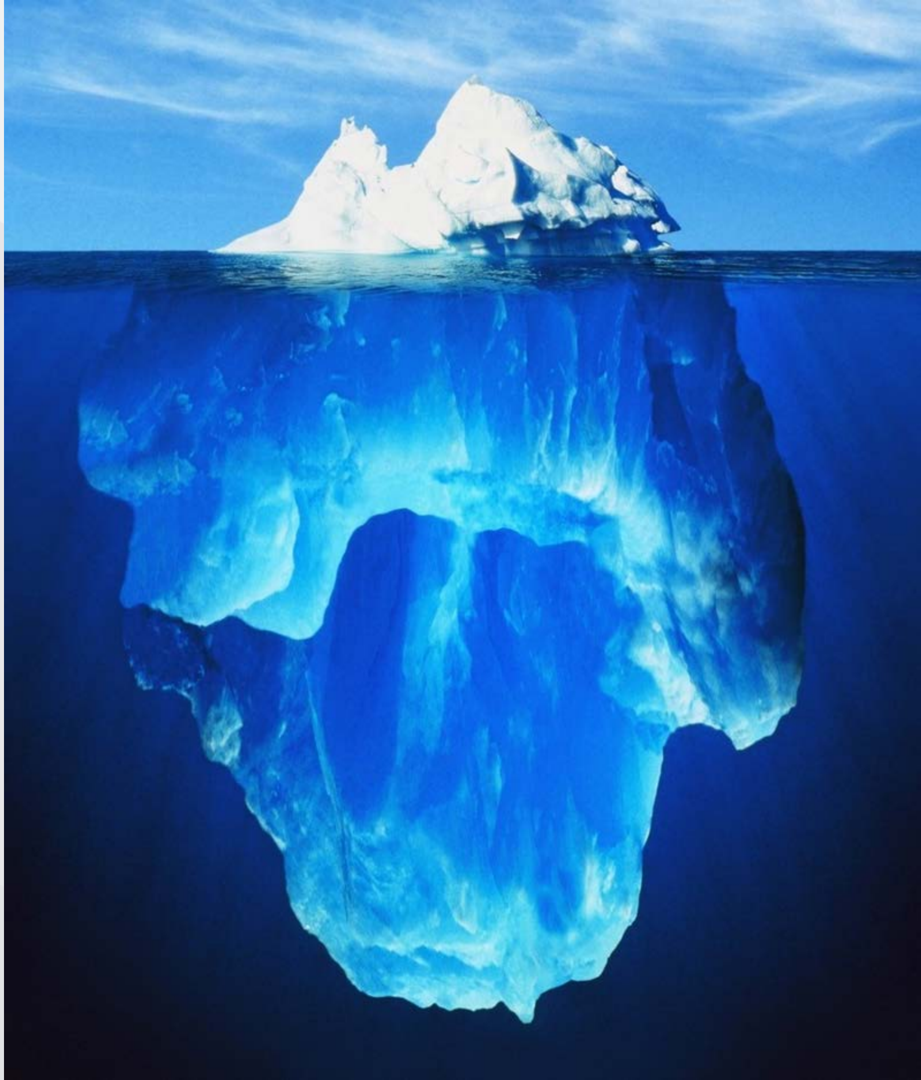
Compute Power

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

Deep Blue beats Gary Kasparov 1997

Watson beats champions 2011

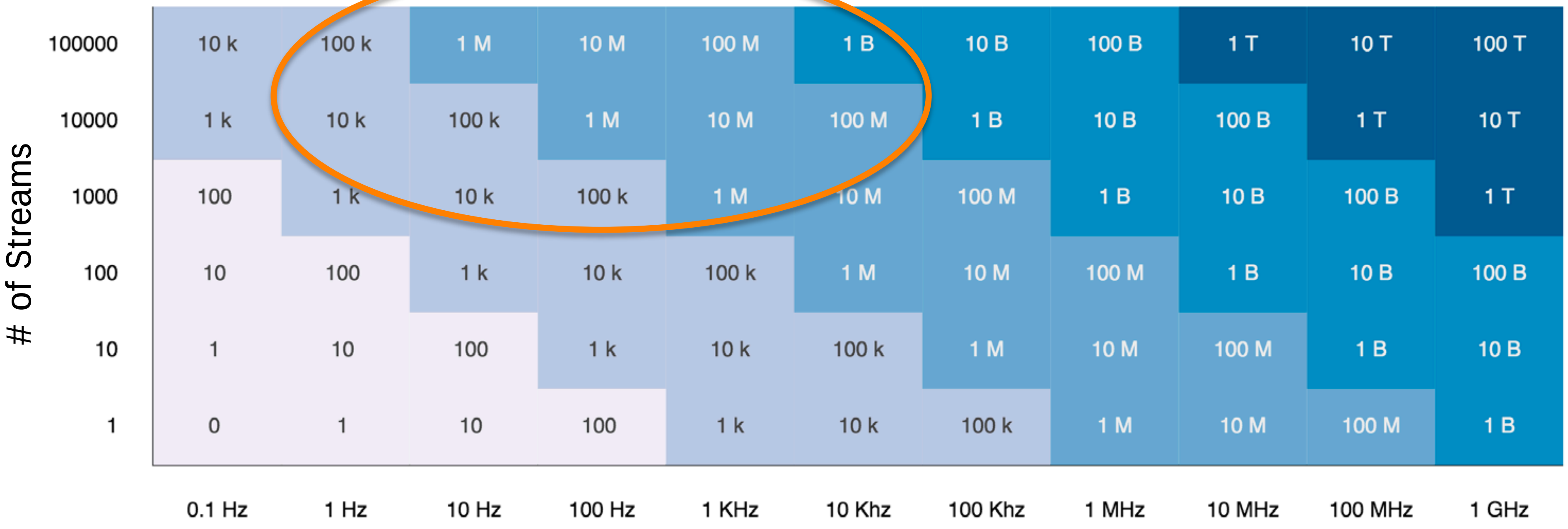
AlphaGo beats Lee Sedol 2016

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

Measurements Per Second



# of Streams

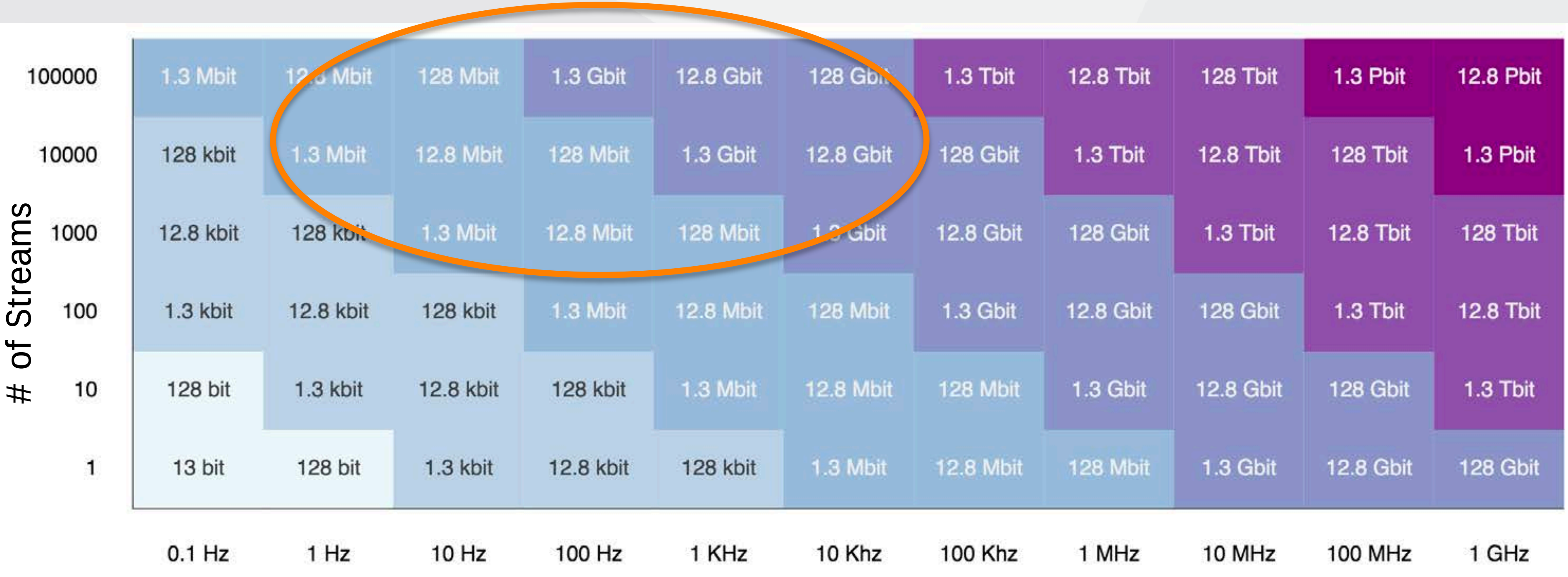
100000	10 k	100 k	1 M	10 M	100 M	1 B	10 B	100 B	1 T	10 T	100 T
10000	1 k	10 k	100 k	1 M	10 M	100 M	1 B	10 B	100 B	1 T	10 T
1000	100	1 k	10 k	100 k	1 M	10 M	100 M	1 B	10 B	100 B	1 T
100	10	100	1 k	10 k	100 k	1 M	10 M	100 M	1 B	10 B	100 B
10	1	10	100	1 k	10 k	100 k	1 M	10 M	100 M	1 B	10 B
1	0	1	10	100	1 k	10 k	100 k	1 M	10 M	100 M	1 B
	0.1 Hz	1 Hz	10 Hz	100 Hz	1 KHz	10 Khz	100 Khz	1 MHz	10 MHz	100 MHz	1 GHz

SCADA PMU DFR

Sampling Rates

Ones	Thousands	Millions	Billions	Trillions

Required Network Bandwidth (Lower Bound)

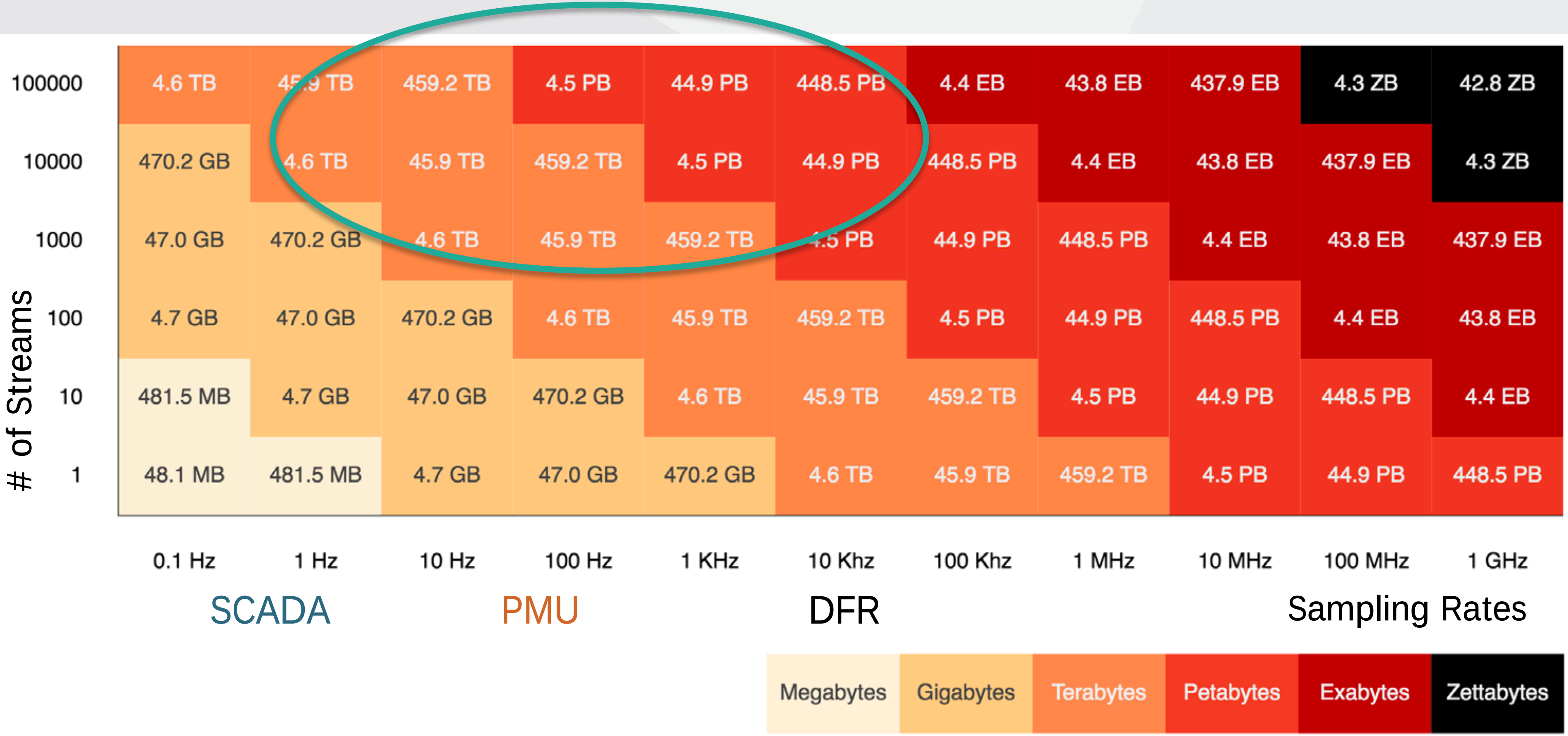


# of Streams	0.1 Hz	1 Hz	10 Hz	100 Hz	1 KHz	10 Khz	100 Khz	1 MHz	10 MHz	100 MHz	1 GHz
100000	1.3 Mbit	12.8 Mbit	128 Mbit	1.3 Gbit	12.8 Gbit	128 Gbit	1.3 Tbit	12.8 Tbit	128 Tbit	1.3 Pbit	12.8 Pbit
10000	128 kbit	1.3 Mbit	12.8 Mbit	128 Mbit	1.3 Gbit	12.8 Gbit	128 Gbit	1.3 Tbit	12.8 Tbit	128 Tbit	1.3 Pbit
1000	12.8 kbit	128 kbit	1.3 Mbit	12.8 Mbit	128 Mbit	1.3 Gbit	12.8 Gbit	128 Gbit	1.3 Tbit	12.8 Tbit	128 Tbit
100	1.3 kbit	12.8 kbit	128 kbit	1.3 Mbit	12.8 Mbit	128 Mbit	1.3 Gbit	12.8 Gbit	128 Gbit	1.3 Tbit	12.8 Tbit
10	128 bit	1.3 kbit	12.8 kbit	128 kbit	1.3 Mbit	12.8 Mbit	128 Mbit	1.3 Gbit	12.8 Gbit	128 Gbit	1.3 Tbit
1	13 bit	128 bit	1.3 kbit	12.8 kbit	128 kbit	1.3 Mbit	12.8 Mbit	128 Mbit	1.3 Gbit	12.8 Gbit	128 Gbit

SCADA PMU DFR Sampling Rates

Bits	Kilobits	Megabits	Gigabits	Terabits	Petabits

Annual Data Volumes (Lower Bound)



# of Streams	0.1 Hz	1 Hz	10 Hz	100 Hz	1 KHz	10 Khz	100 Khz	1 MHz	10 MHz	100 MHz	1 GHz
100000	4.6 TB	45.9 TB	459.2 TB	4.5 PB	44.9 PB	448.5 PB	4.4 EB	43.8 EB	437.9 EB	4.3 ZB	42.8 ZB
10000	470.2 GB	4.6 TB	45.9 TB	459.2 TB	4.5 PB	44.9 PB	448.5 PB	4.4 EB	43.8 EB	437.9 EB	4.3 ZB
1000	47.0 GB	470.2 GB	4.6 TB	45.9 TB	459.2 TB	4.5 PB	44.9 PB	448.5 PB	4.4 EB	43.8 EB	437.9 EB
100	4.7 GB	47.0 GB	470.2 GB	4.6 TB	45.9 TB	459.2 TB	4.5 PB	44.9 PB	448.5 PB	4.4 EB	43.8 EB
10	481.5 MB	4.7 GB	47.0 GB	470.2 GB	4.6 TB	45.9 TB	459.2 TB	4.5 PB	44.9 PB	448.5 PB	4.4 EB
1	48.1 MB	481.5 MB	4.7 GB	47.0 GB	470.2 GB	4.6 TB	45.9 TB	459.2 TB	4.5 PB	44.9 PB	448.5 PB

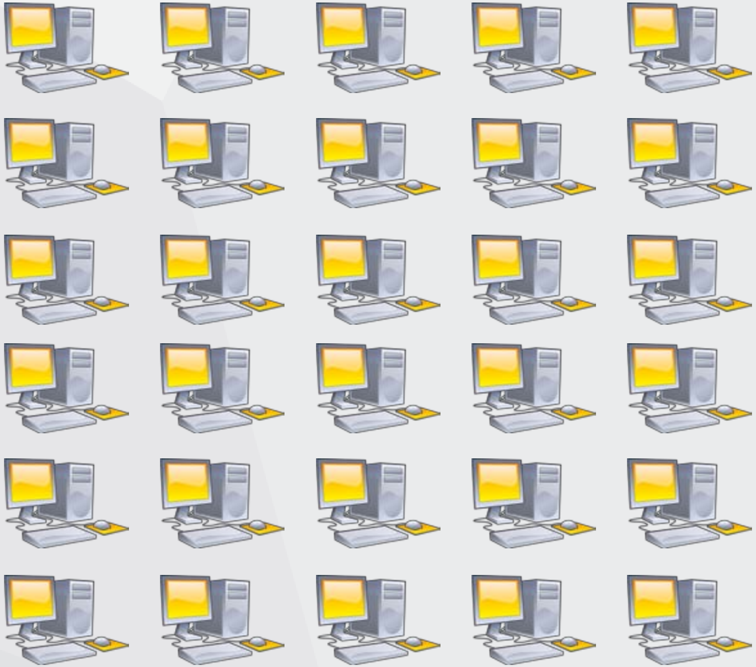
SCADA PMU DFR Sampling Rates

Megabytes	Gigabytes	Terabytes	Petabytes	Exabytes	Zettabytes

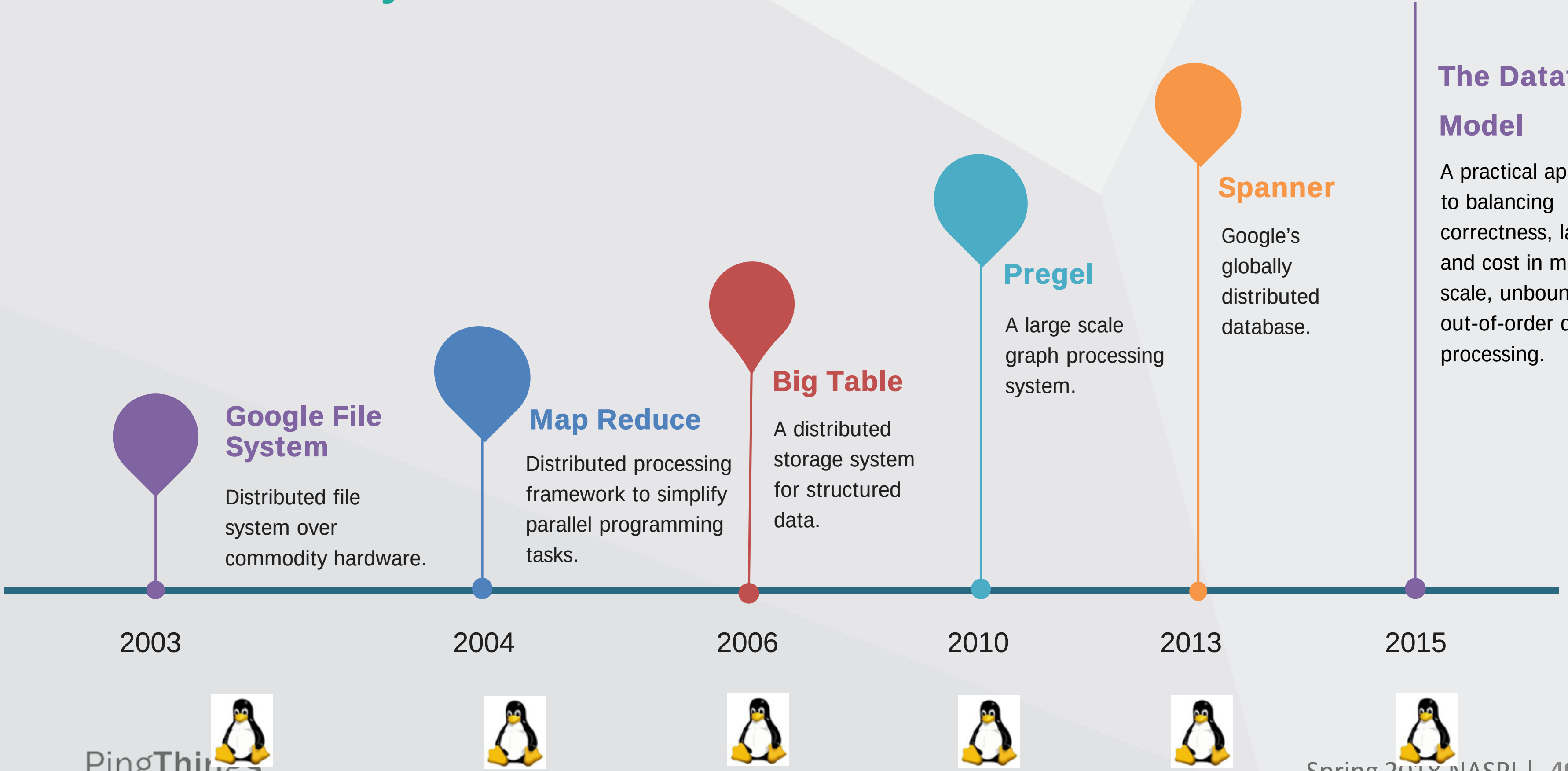
Big Data Conclusions

- 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

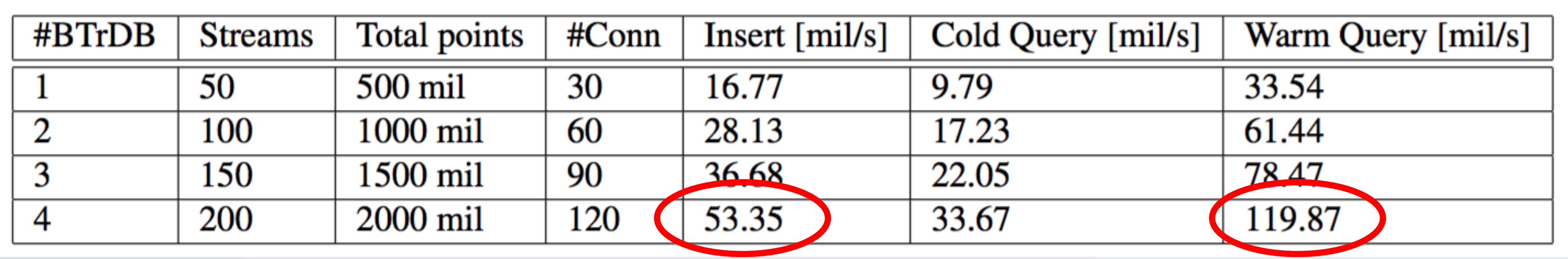
A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing.

Who Cares?

- No single point of failure/resilient
- Much more cost effective
- Buy more capacity when you need it
- Flexible and adaptable

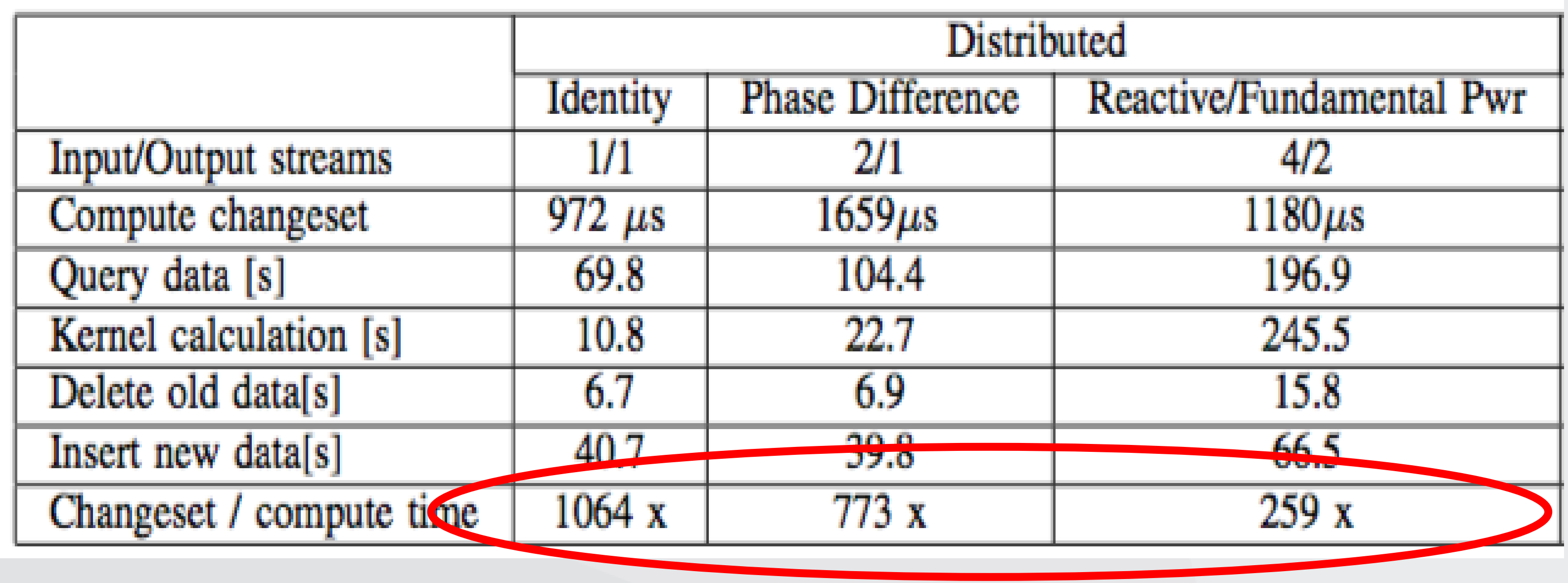
Benchmark Considerations

Core Benchmarks – Reading and Writing Data



#BTrDB	Streams	Total points	#Conn	Insert [mil/s]	Cold Query [mil/s]	Warm Query [mil/s]
1	50	500 mil	30	16.77	9.79	33.54
2	100	1000 mil	60	28.13	17.23	61.44
3	150	1500 mil	90	36.68	22.05	78.47
4	200	2000 mil	120	53.35	33.67	119.87

Analytics Benchmarks

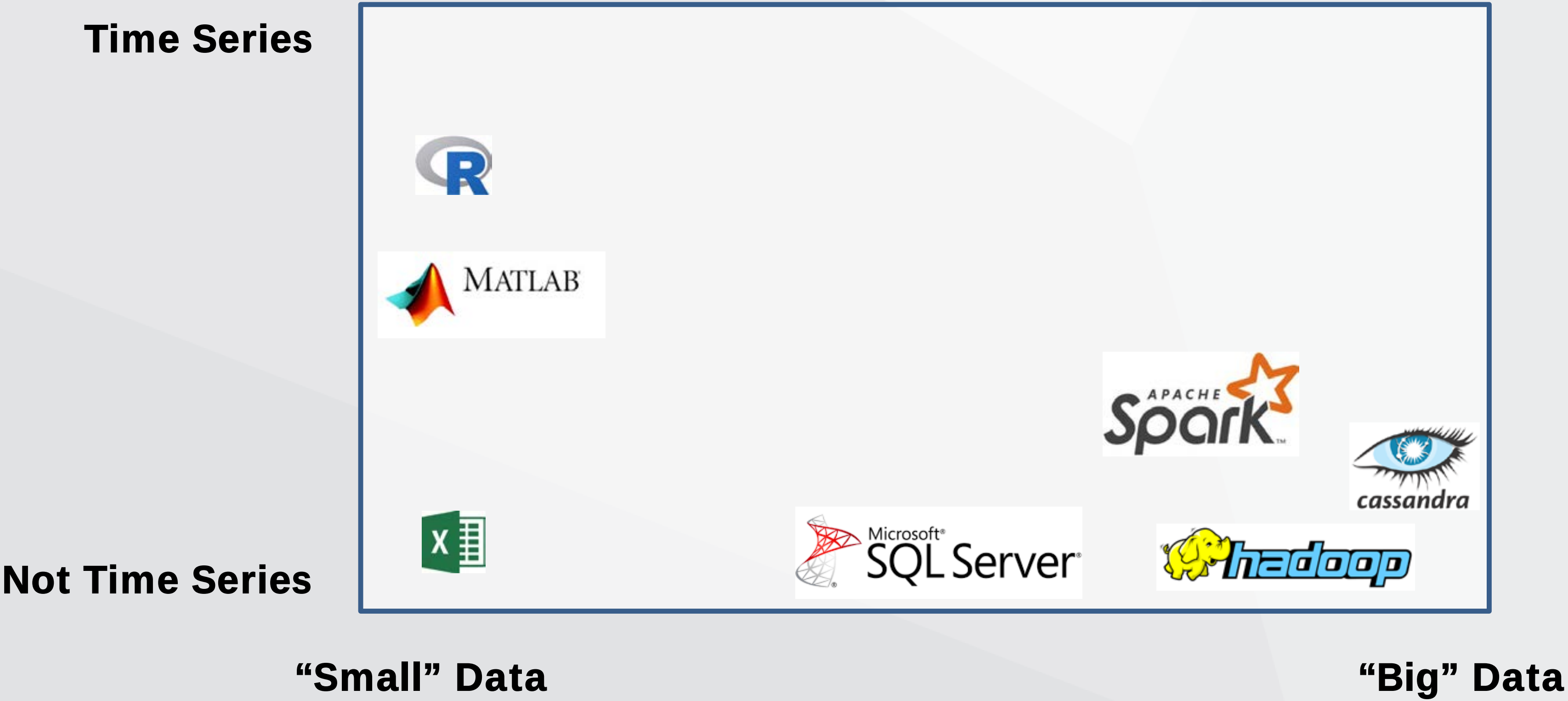


	Distributed		
	Identity	Phase Difference	Reactive/Fundamental Pwr
Input/Output streams	1/1	2/1	4/2
Compute changeset	972 μs	1659μs	1180μs
Query data [s]	69.8	104.4	196.9
Kernel calculation [s]	10.8	22.7	245.5
Delete old data[s]	6.7	6.9	15.8
Insert new data[s]	40.7	39.8	66.5
Changeset / compute time	1064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

PingThings

What is a Platform?

What isn’t a Time Series Platform?



Time Series

Not Time Series

“Small” Data

“Big” Data

What Does a Time Series Data Platform Do?

Ingest/Egest

Clean/Condition/Fix

Store

Visualize

Access/Use/Build With

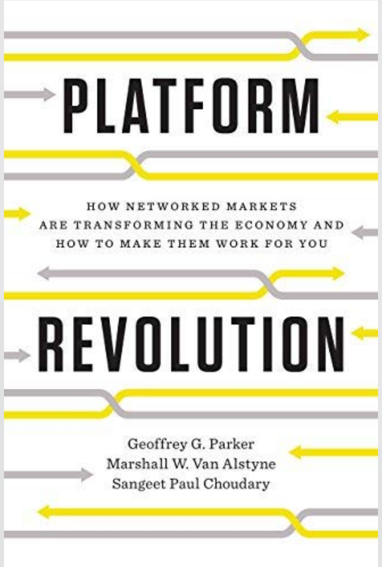
Analyze

Learn From

The Goal of the Platform

**Allow
utility subject matter experts
to create value
from (sensor) data.**

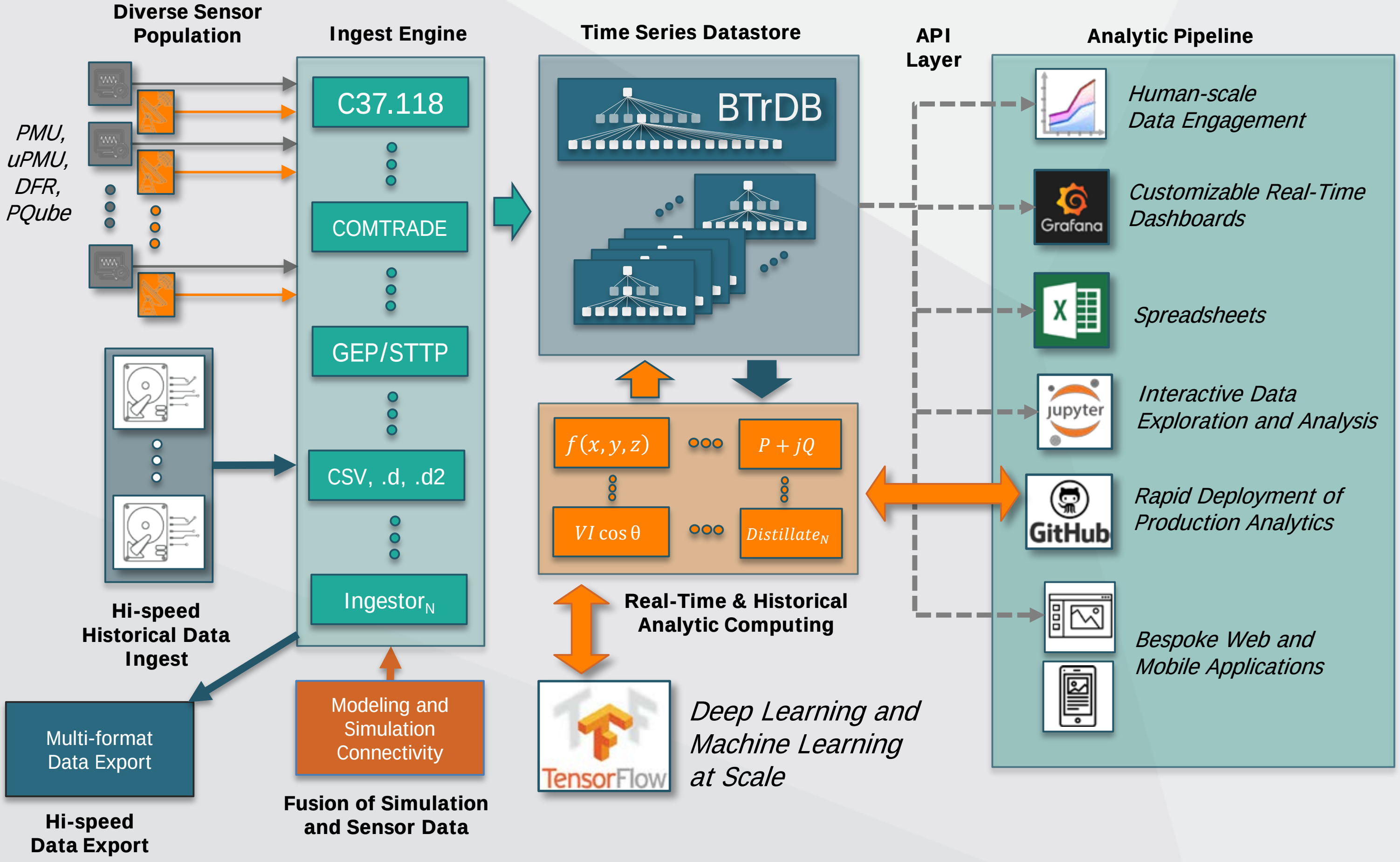
Platform?


PLATFORM
HOW NETWORKED MARKETS ARE TRANSFORMING THE ECONOMY AND HOW TO MAKE THEM WORK FOR YOU
REVOLUTION
Geoffrey G. Parker
Marshall W. Van Alstyne
Sangeet Paul Choudary

“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.”

PingThings PredictiveGrid™



Diverse Sensor Population
PMU, uPMU, DFR, PQube
Ingest Engine
C37.118
COMTRADE
GEP/STTP
CSV, .d, .d2
Ingestor$_N$
Hi-speed Historical Data Ingest
Multi-format Data Export
Hi-speed Data Export
Modeling and Simulation Connectivity
Fusion of Simulation and Sensor Data
Time Series Datastore
BTrDB
$f(x,y,z)$
$P + jQ$
$VI \cos\theta$
Distillate$_N$
Real-Time & Historical Analytic Computing
TensorFlow
Deep Learning and Machine Learning at Scale
API Layer
Analytic Pipeline
Human-scale Data Engagement
Grafana
Customizable Real-Time Dashboards
Spreadsheets
jupyter
Interactive Data Exploration and Analysis
GitHub
Rapid Deployment of Production Analytics
Bespoke Web and Mobile Applications

sean@pingthings.ai