DataSketches

A Required Toolkit for the Analysis of Big Data

Lee Rhodes
Verizon/Oath/Yahoo, Inc.
Alan Turing Institute 1 Nov 2017

Our Challenge

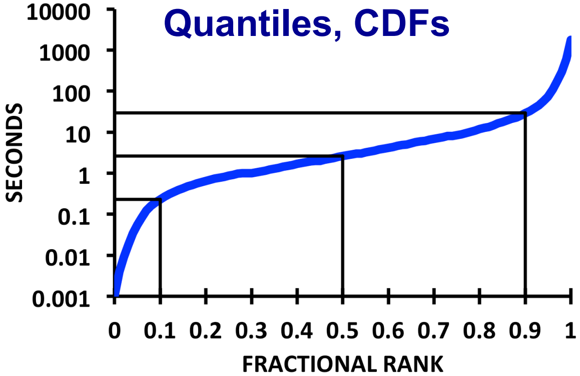
Example: Web Site Logs

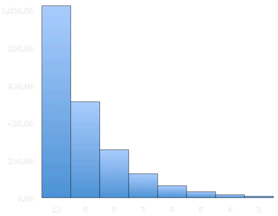
Time Stamp	User ID	Device ID	Site	Time Spent Sec	Items Viewed
9:00 AM	U1	D1	Apps	59	5
9:30 AM	U2	D2	Apps	179	15
10:00 AM	U3	D3	Music	29	3
1:00 PM	U1	D4	Music	89	10
Billions of *K,V* Pairs …					

Analyze This Data In Near-Real Time.

Some Very Common Queries …

Quantiles, CDFs



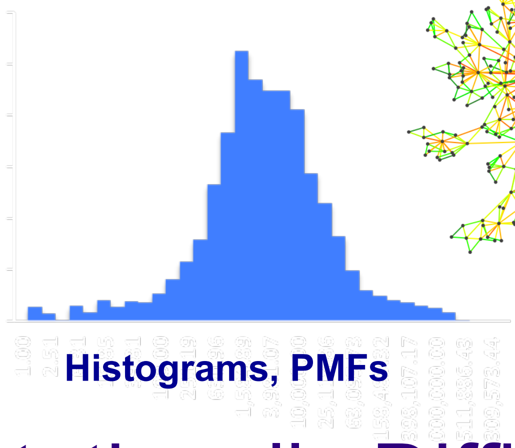


Frequent Items

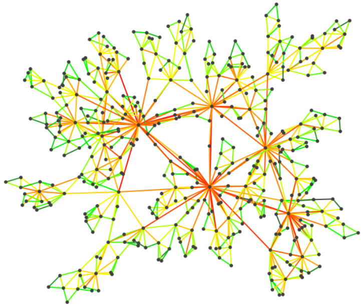
Vector & Matrix Operations: SVD, PCA, *k*-means, etc.

$$\begin{pmatrix} 5 & \cdots & 2 \\ \vdots & \ddots & \vdots \\ 4 & \cdots & 3 \end{pmatrix}$$

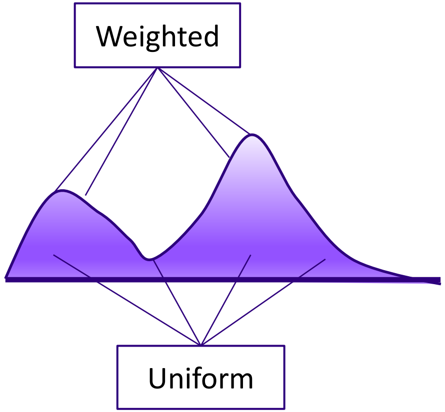
Unique Identifiers



Histograms, PMFs



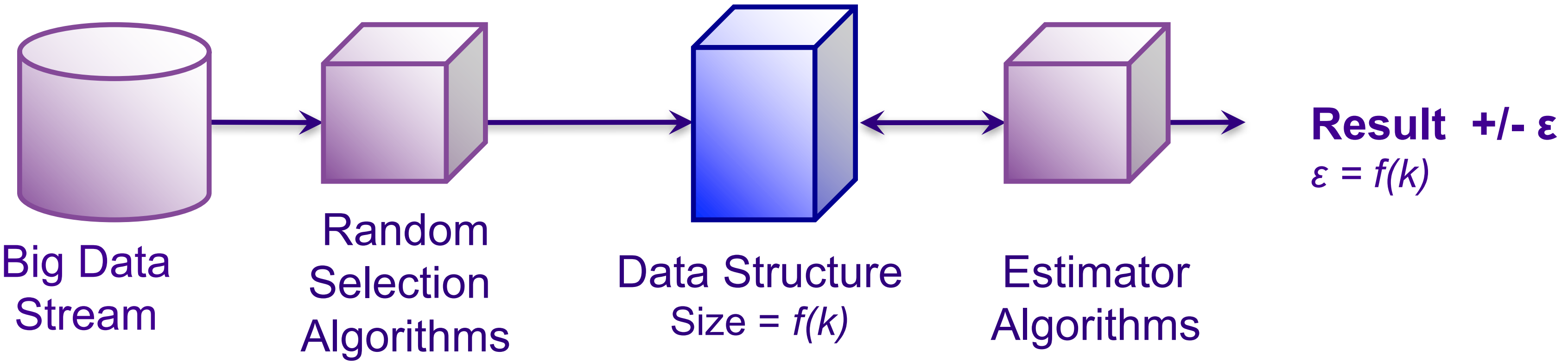
Graphs



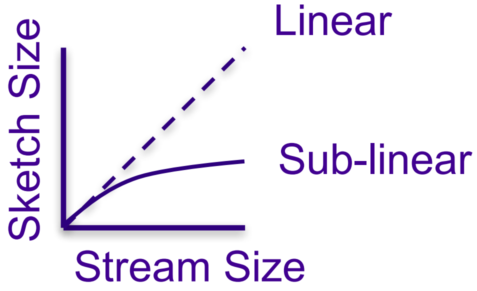
Reservoir Sampling

Are All Computationally Difficult

Our Mission: Develop Production Quality Streaming Algorithms (Sketches) to Address these Difficult Queries



- Small Size, Sub-linear in Space
- Single-pass
- Mergeable
- Mathematically proven error bounds



What Does “Production Quality” Mean?

- Mergeable with different size-accuracy parameters: (e.g. *k*)
- Unit-tests with > 90% code coverage
- Comprehensive Accuracy and Speed Characterization Studies
- High-Speed Performance
- Excellent Space Utilization: (Across Millions of Sketches)
- Minimal External Dependencies
- Operations on Stored Sketches Must Be Backward Compatible
- Design for Off-Heap operation, where possible
- Minimize State Exceptions

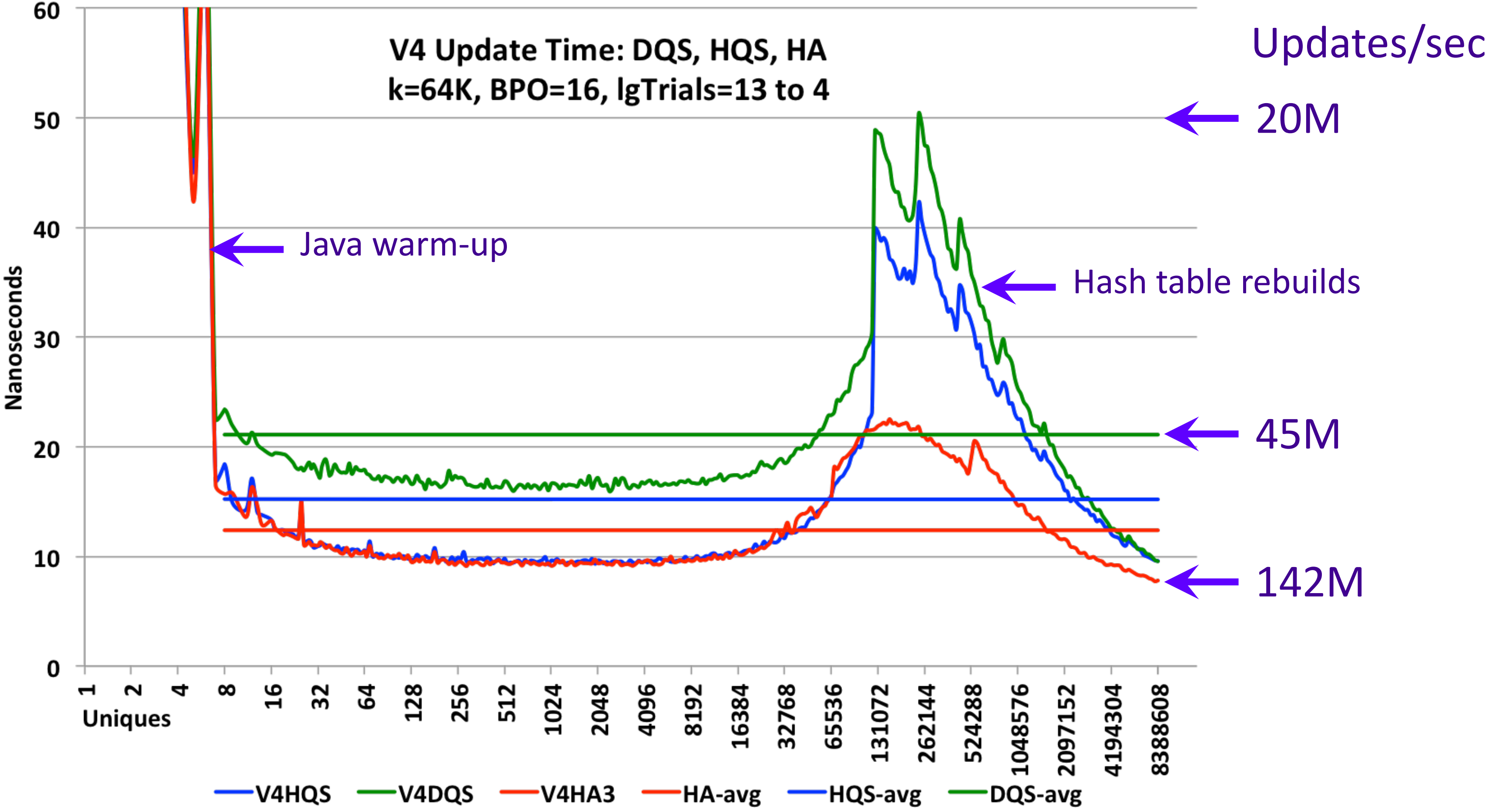
Innovations for Unique Counting Sketches

Theta Sketch Framework (TSF):

A. Dasgupta, K. Lang, L. Rhodes, J. Thaler, A Framework for Estimating Stream Expression Cardinalities, *ACM ICDT 2016*

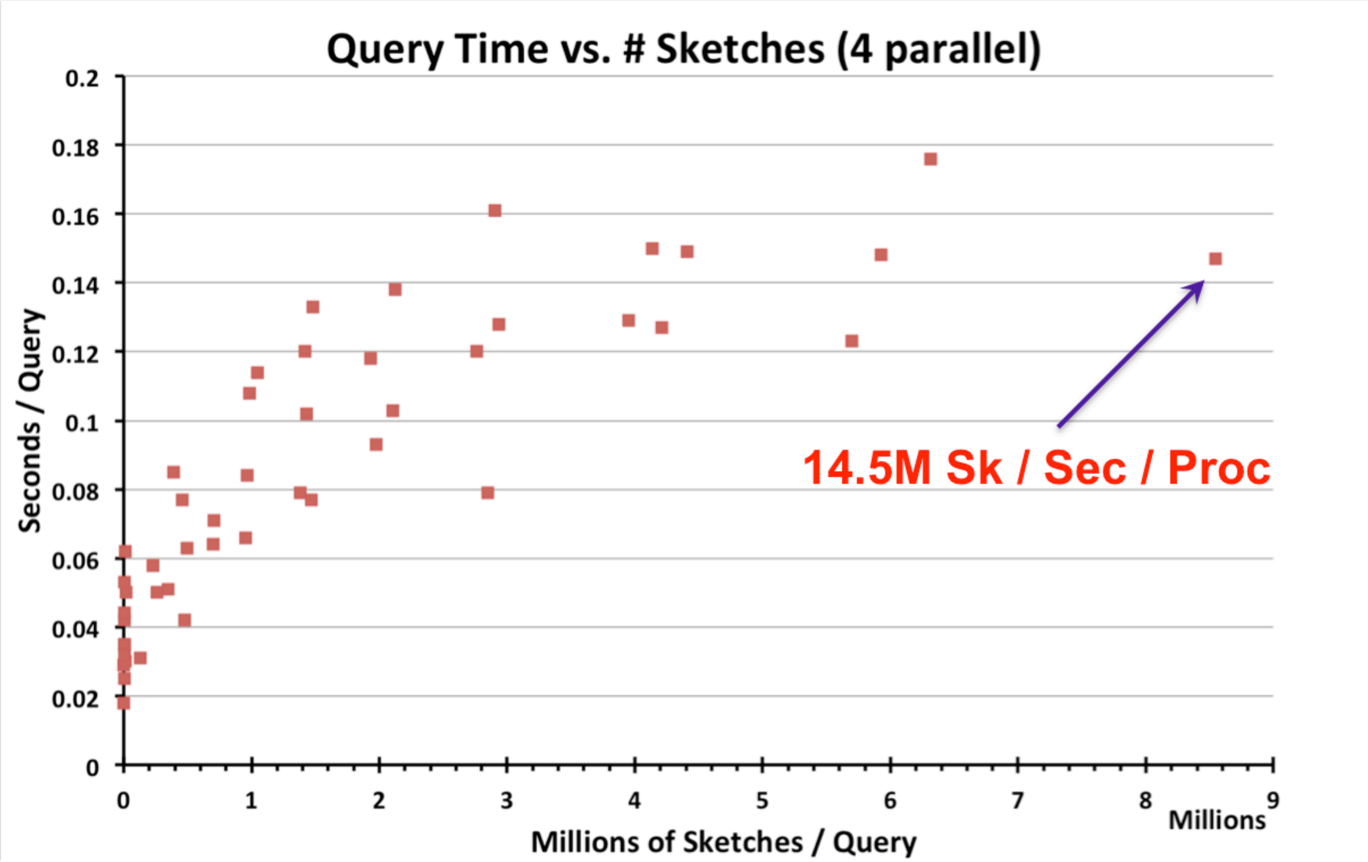
- Builds on Bar-Yossef, et al, 2002 "Counting Distinct Elements…"
- TSF applies to a whole family of sketches
- Enables simple methods for enabling set expressions and multiple-k merging
- Enables trivial up-front, (*p*KMV) sampling for tighter space usage in large systems
- Library Theta Sketch Framework:
  - Sketches: UpdateSketch, CompactSketch, AlphaSketch
  - Set Expressions: Union, Intersection, AnotB: $(A \cup B) \cap (C \cup D) \setminus E$
  - Tuple Sketch (Update Sketch with User-defined attributes)

TSF: Theta Sketch Update Speed, 64K

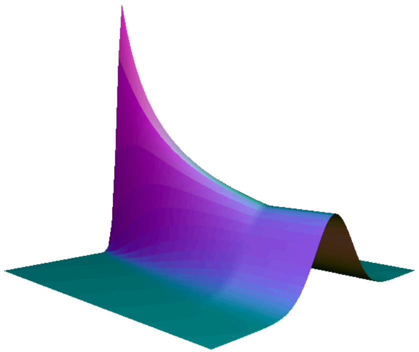


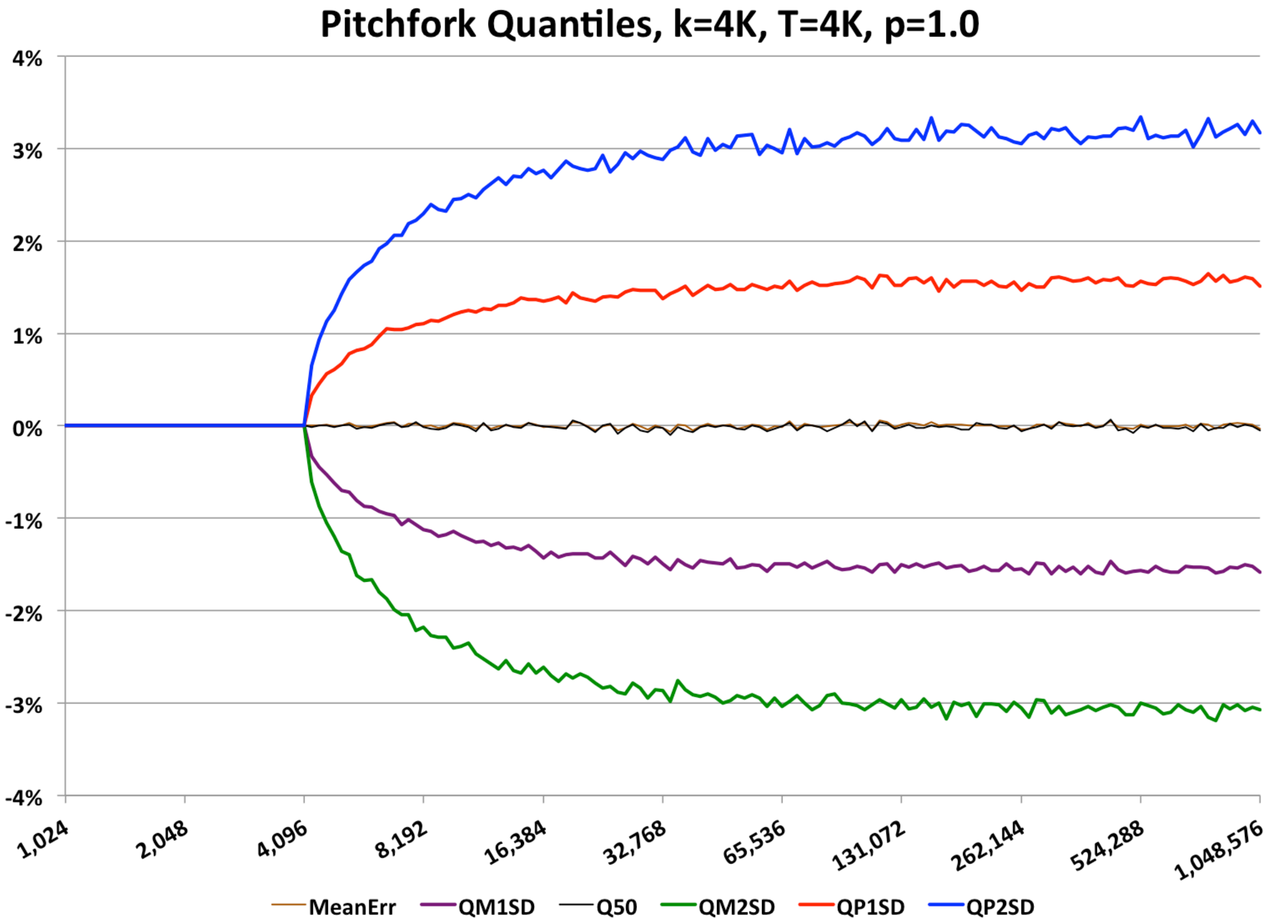
Theta Sketch Framework

Sketch Merge Time / Query



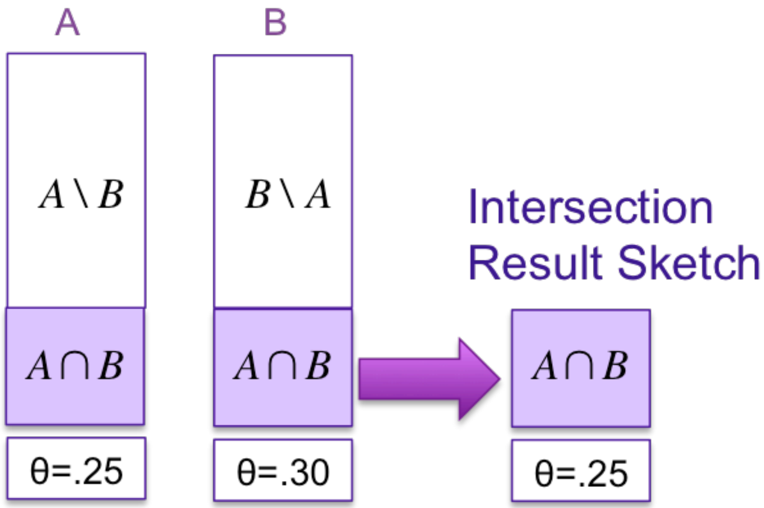
Theta Sketch Framework: Theta Sketch Accuracy





Theta Sketch Framework

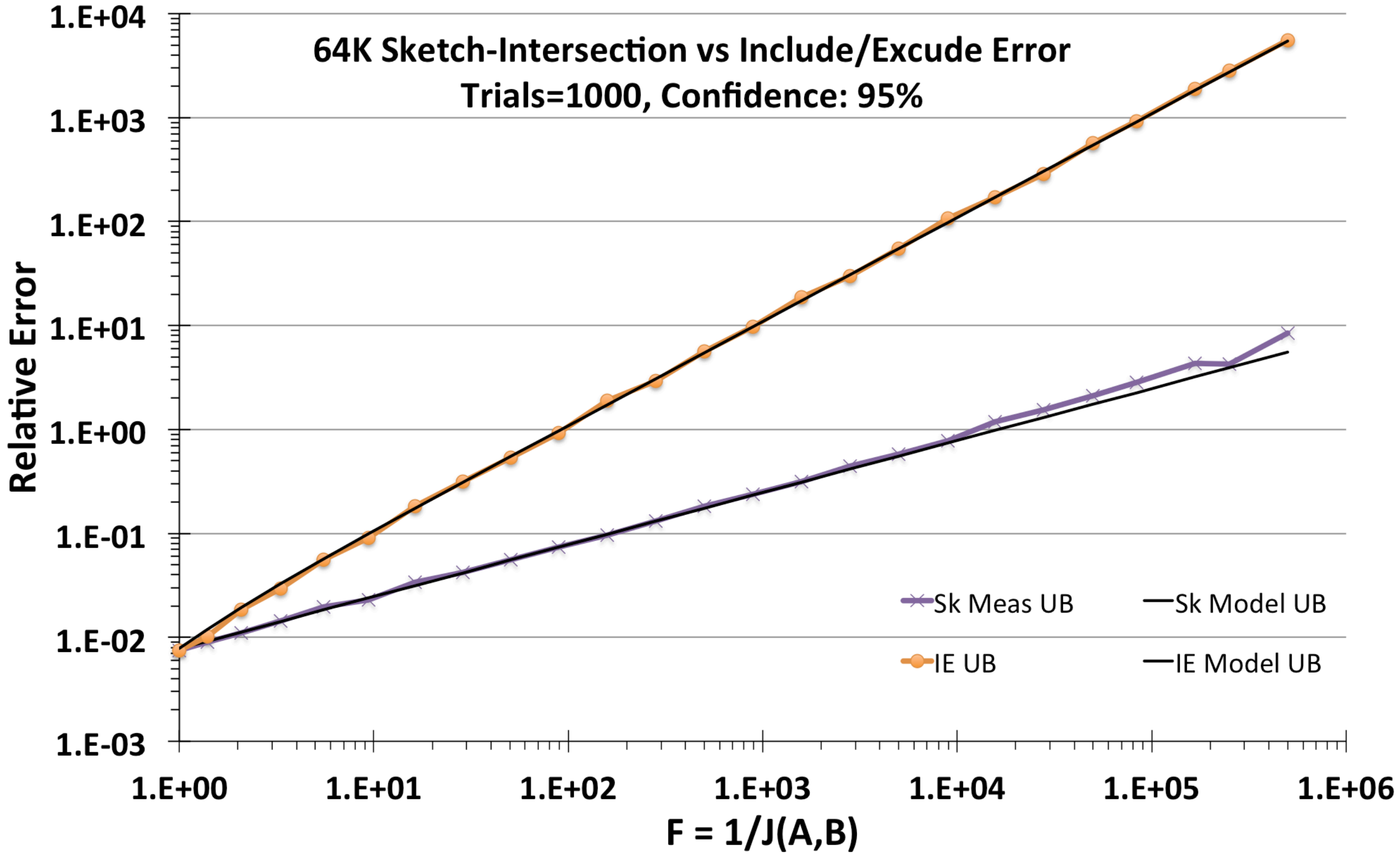
Set Expressions



$$\Delta = \{\cup, \cap, \setminus\}; \qquad \theta_{A\Delta B} = \min(\theta_A, \theta_B); \qquad S_{A\Delta B} = \{x < \theta_{A\Delta B}; x \in (S_A \Delta S_B)\}$$

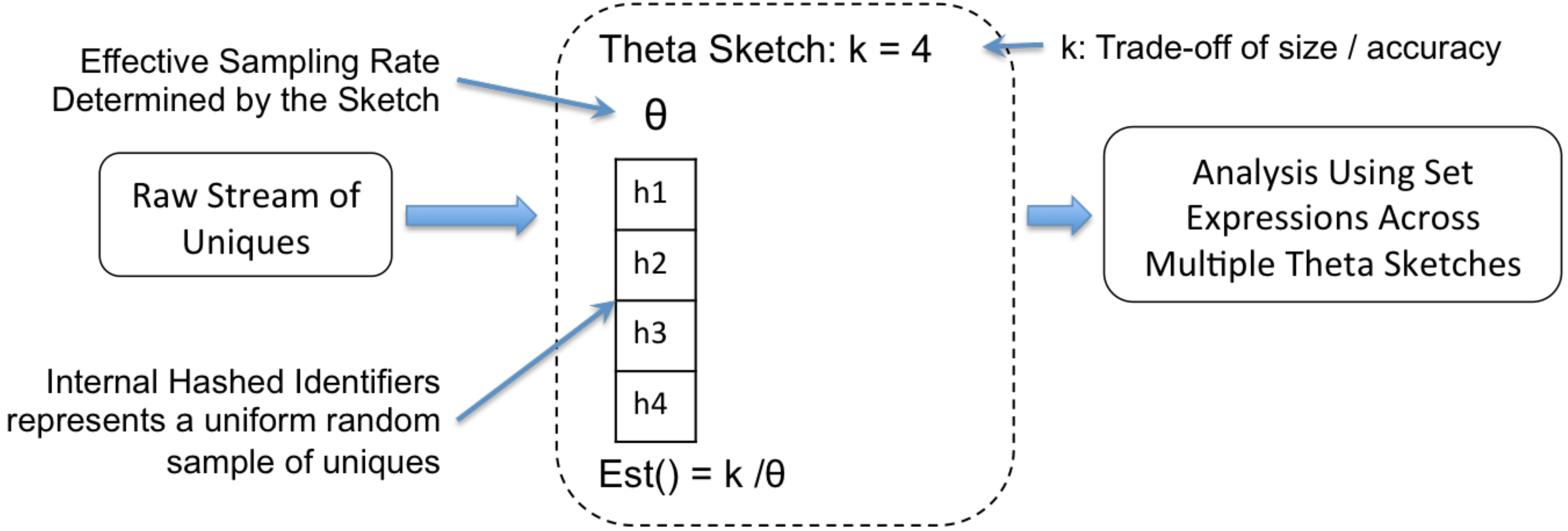
$$\mathrm{est}(|A\Delta B|) = \frac{|S_{A\cup B}|}{\min(\theta_A, \theta_B)} \frac{|S_{A\Delta B}|}{|S_{A\cup B}|} = \frac{|S_{A\Delta B}|}{\min(\theta_A, \theta_B)}, \quad \text{Using "Broder Rule"}$$

Theta Sketch Framework: Intersection Accuracy



Theta Sketch Framework

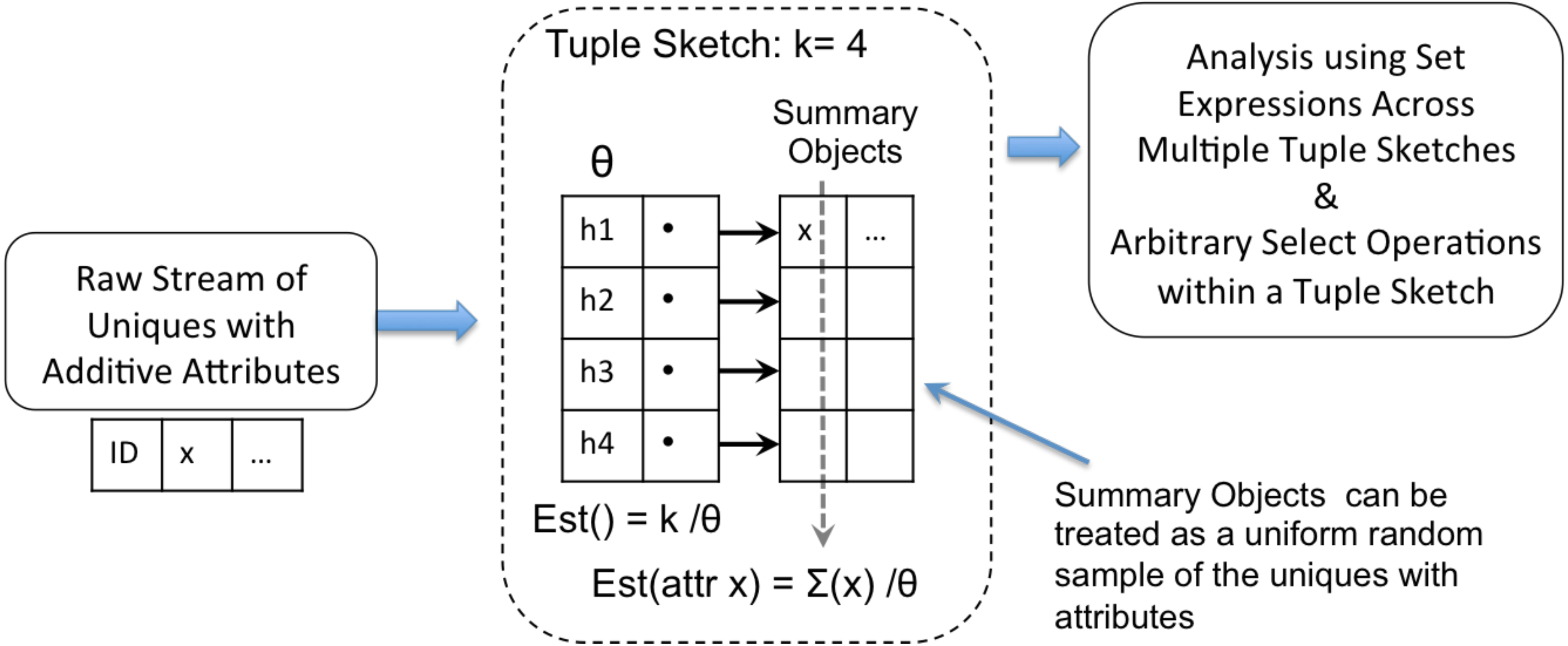
Simple Theta Sketch



Theta Sketch Framework: Tuple Sketch (cont.)

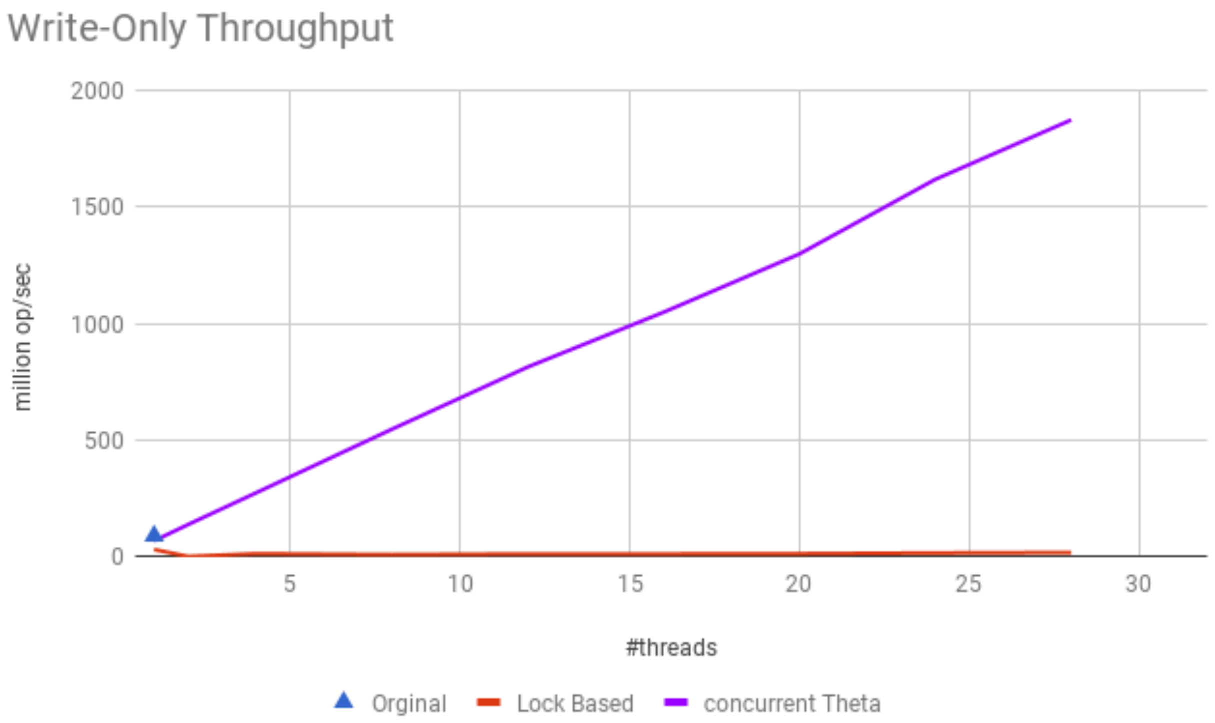
Tuple Sketch: Adding Attributes to the Theta Sketch

Add User-defined Attributes



Innovations for Unique Counting Sketches (cont.)

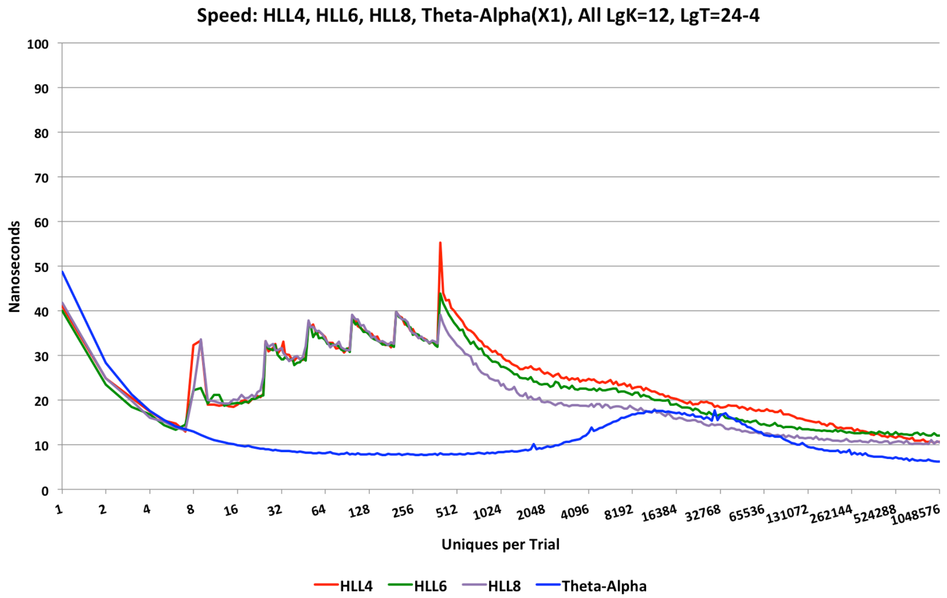
Breaking Up The Sketch for Concurrency (early research)

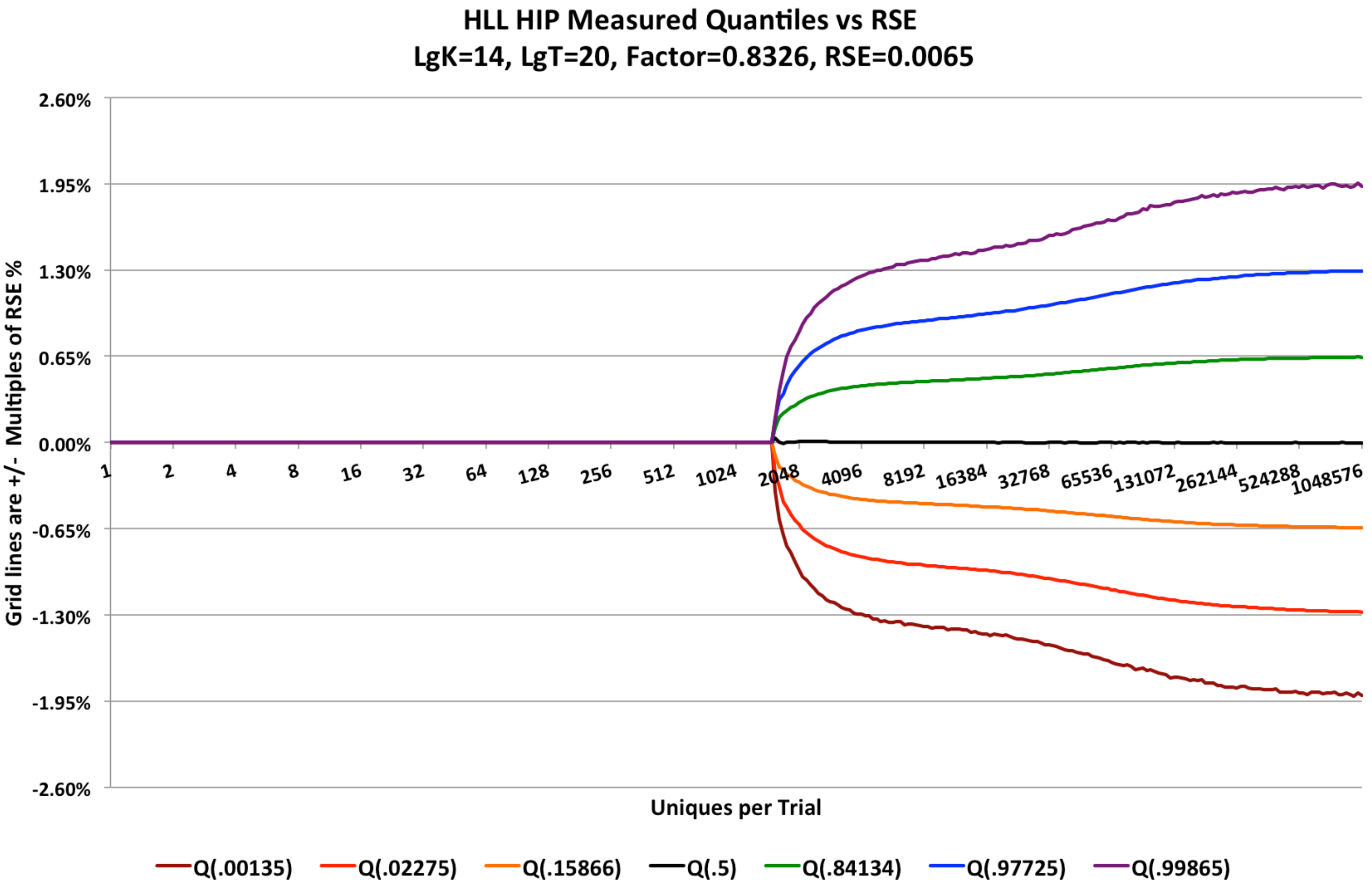


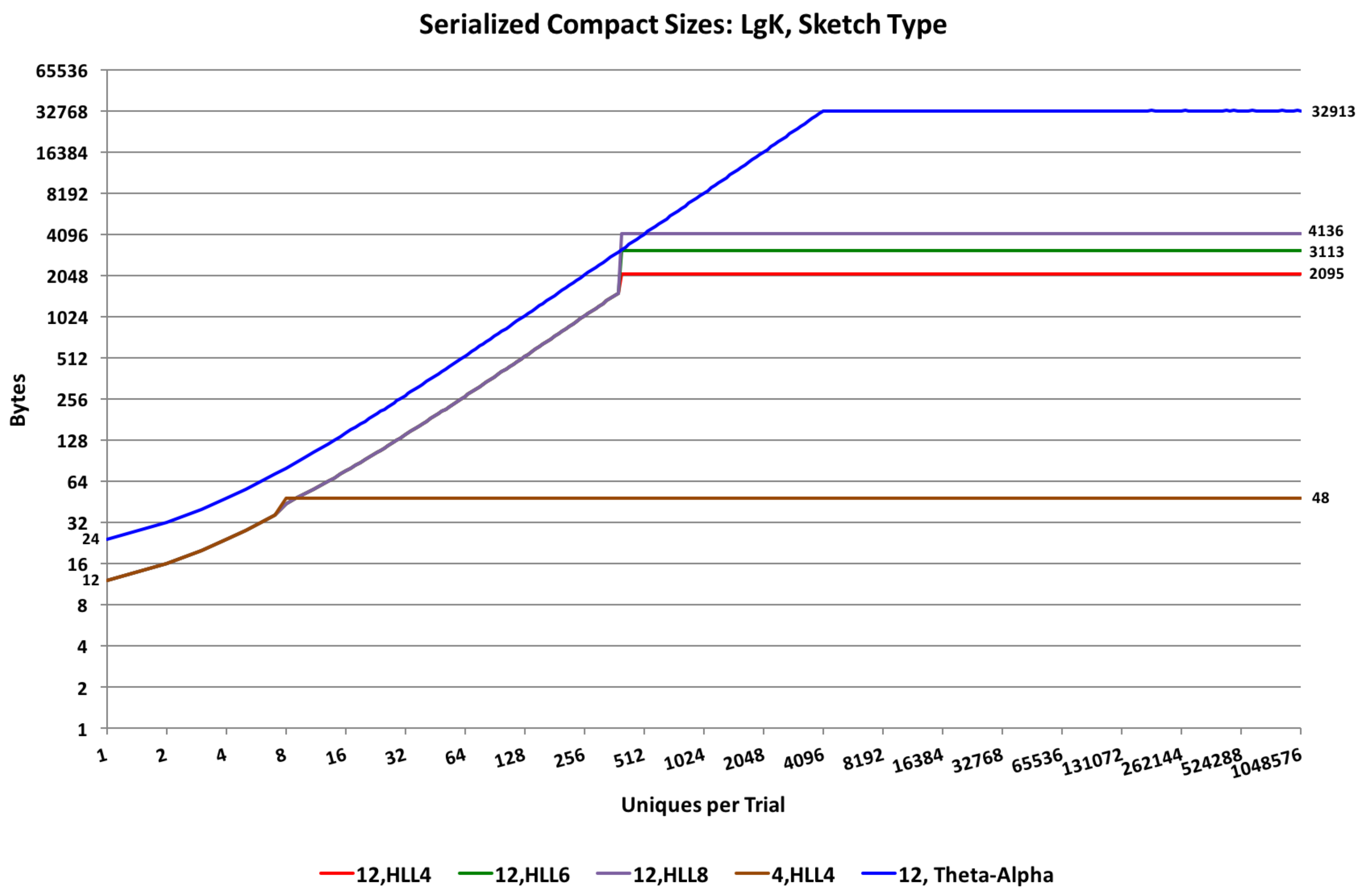
Innovations for Hyper Log Log Sketches (cont.)

HllSketch, The Fastest, Most Accurate HLL Sketch Out There

- Highly tuned for speed
- Simple-to-use API
- Operates either On-Heap or Off-Heap
- Leverages low-range estimators from the FM85 paper (mentioned below)




HLL HIP Measured Quantiles vs RSE
LgK=14, LgT=20, Factor=0.8326, RSE=0.0065
Grid lines are +/- Multiples of RSE %
2.60%
1.95%
1.30%
0.65%
0.00%
-0.65%
-1.30%
-1.95%
-2.60%
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
Uniques per Trial
Q(.00135)
Q(.02275)
Q(.15866)
Q(.5)
Q(.84134)
Q(.97725)
Q(.99865)

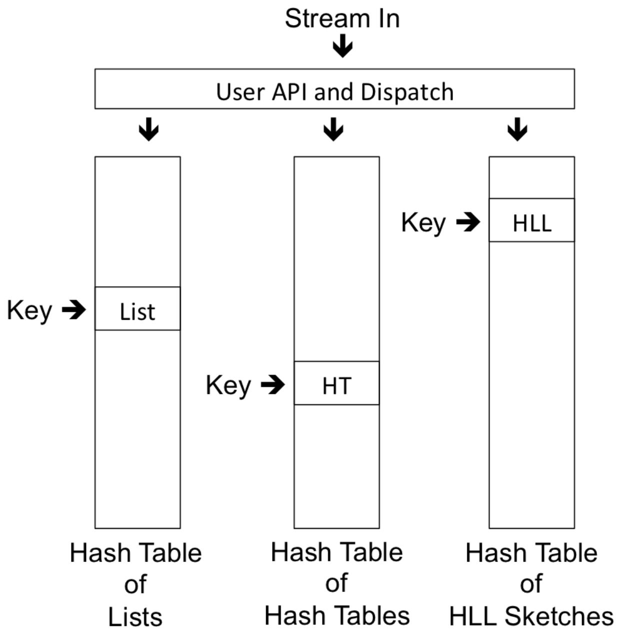


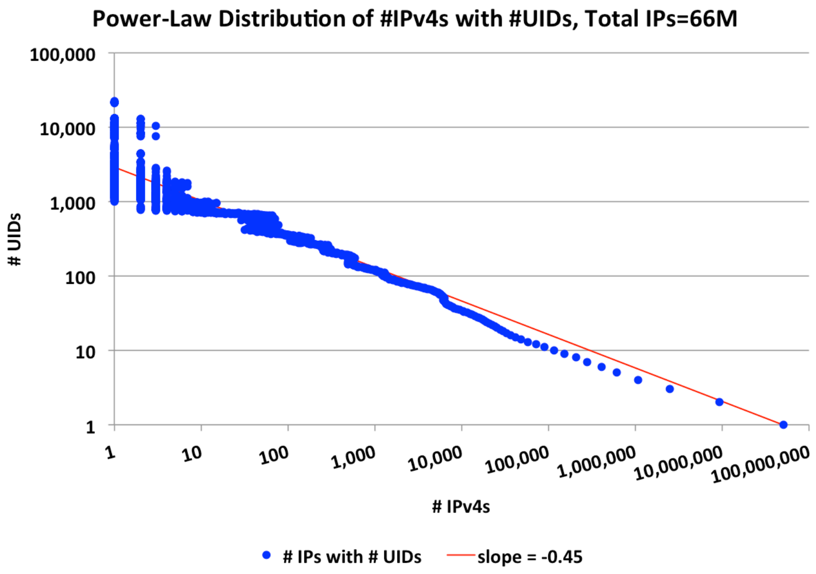
Serialized Compact Sizes: LgK, Sketch Type
Bytes
65536
32768
16384
8192
4096
2048
1024
512
256
128
64
32
24
16
12
8
4
2
1
32913
4136
3113
2095
48
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
Uniques per Trial
12,HLL4
12,HLL6
12,HLL8
4,HLL4
12, Theta-Alpha

Innovations for Hyper Log Log Sketches (cont.)

UniqueCountMap (streams of millions of *K,V* pairs)

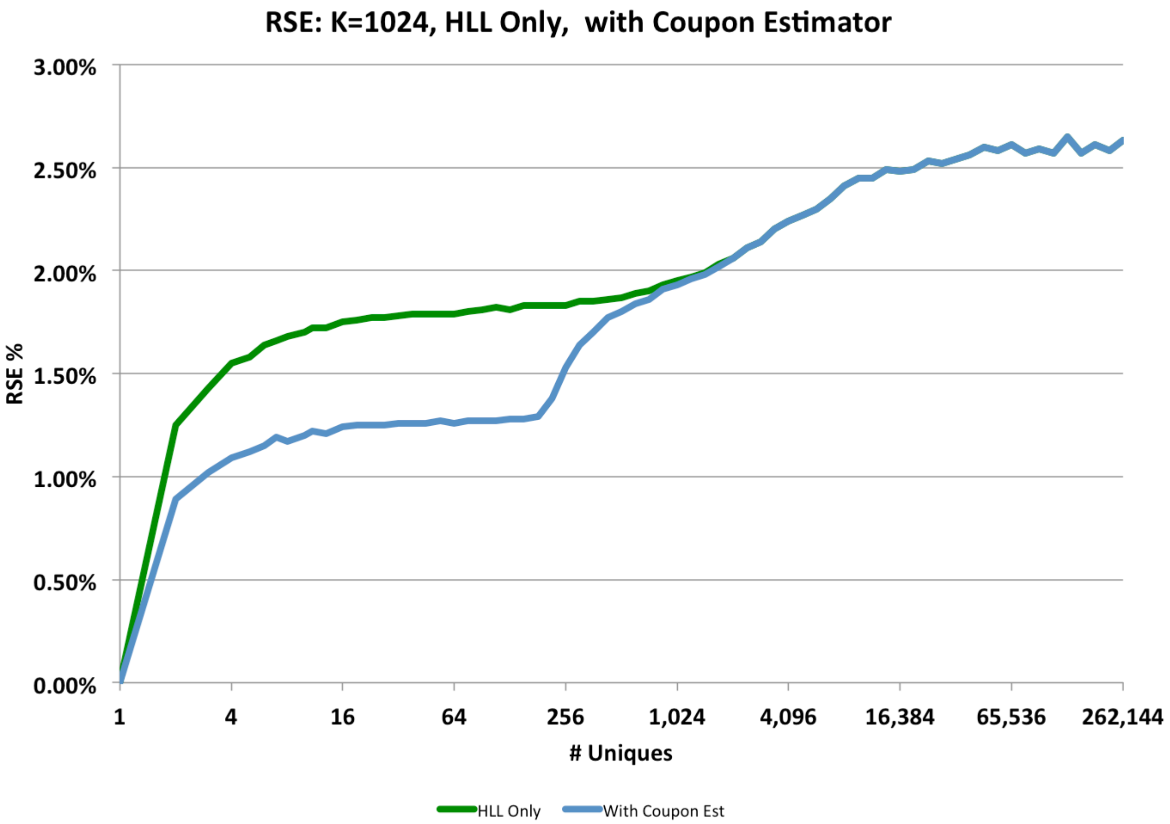
- Real-time cardinality estimates of *V per Key*
- Highly space-efficient (100M 4-byte keys require ~1.3GB: ~9 bytes / K for card.
- Separate data structures manage different phases of sketching process
- Simple-to-use API



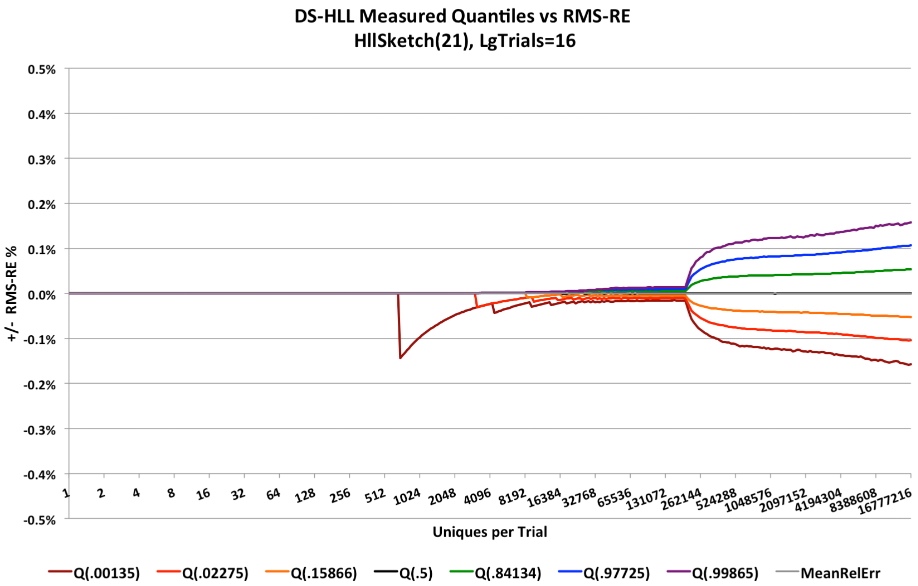


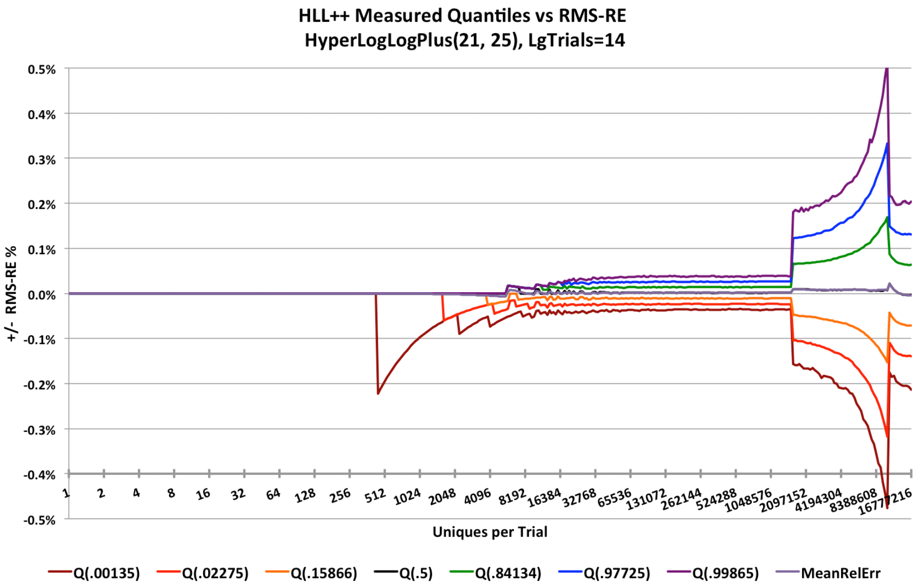
Innovations for Hyper Log Log Sketches (cont.)

UniqueCountMap Accuracy, K = 1024

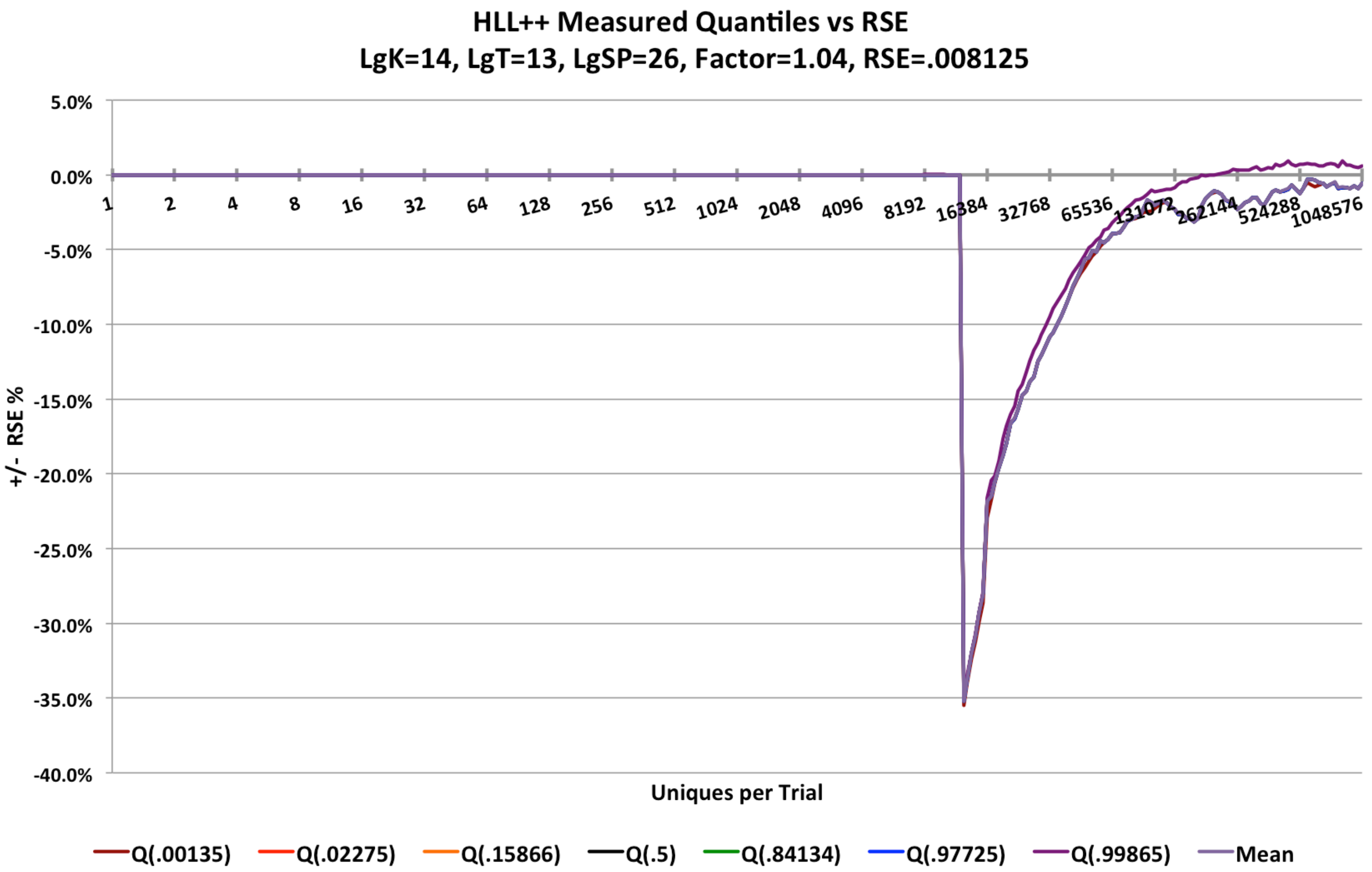


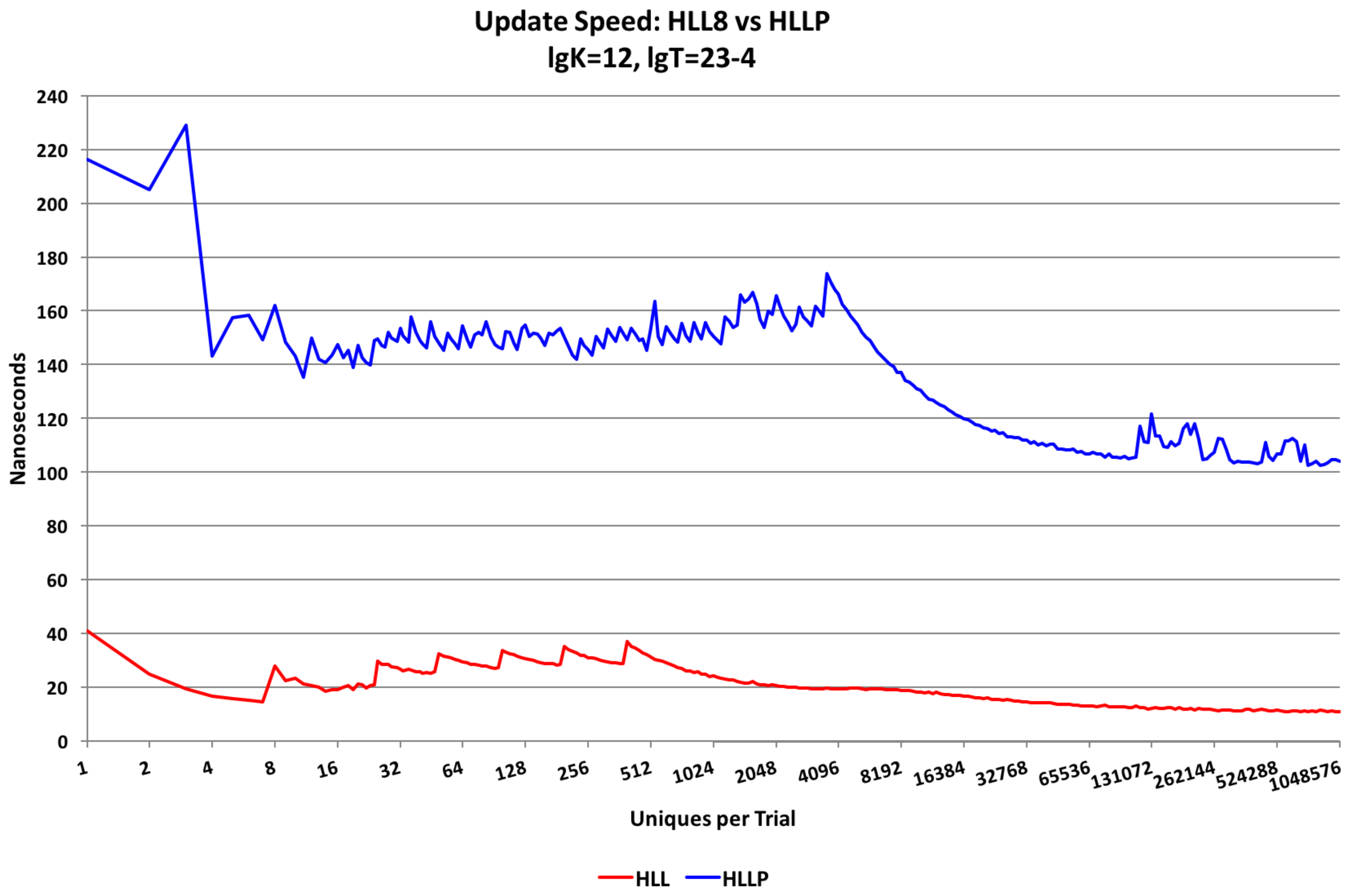
DataSketches HllSketch vs Clearspring HyperLogLogPlus (Google HLL++)

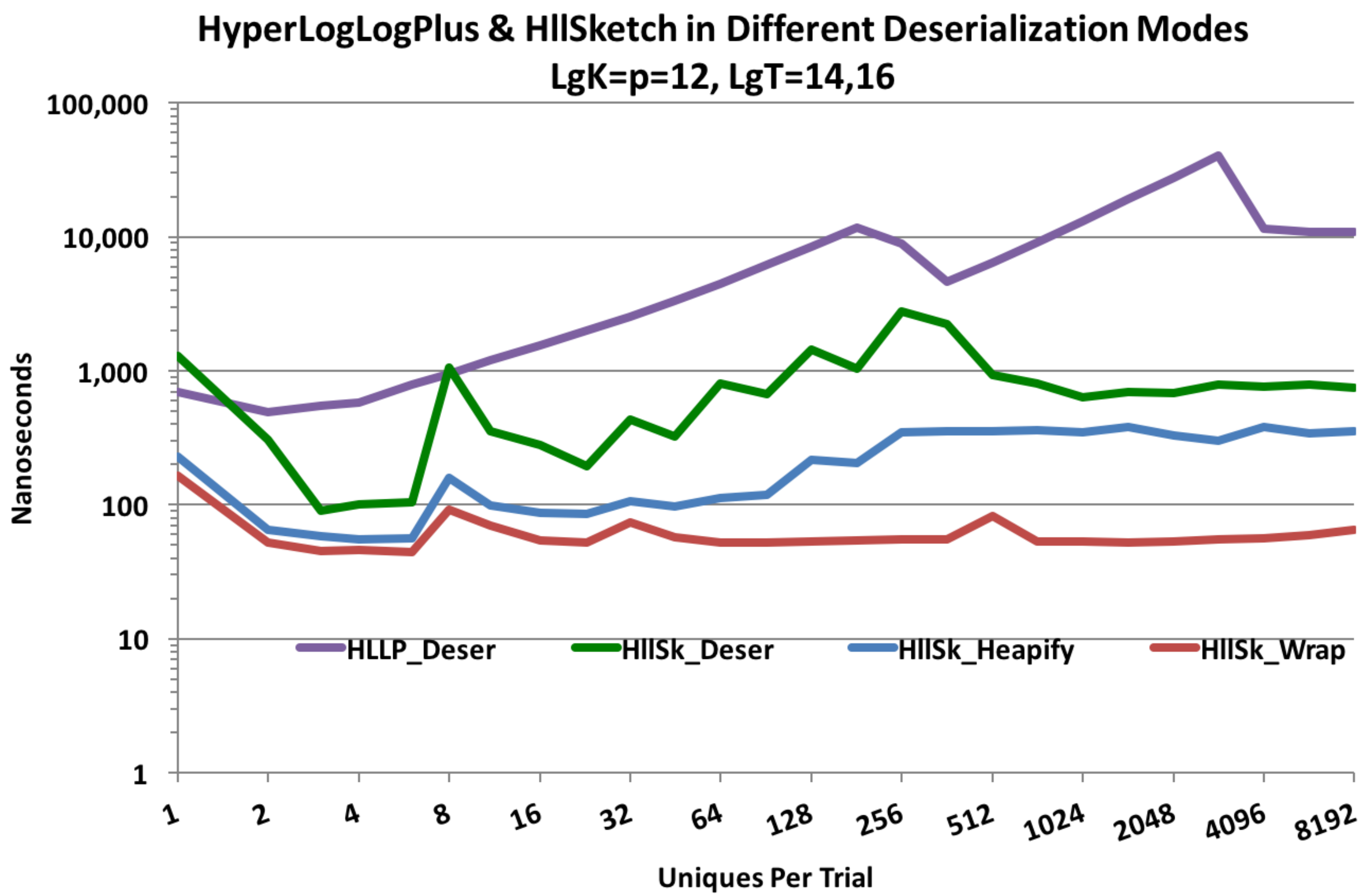




Catastrophic Failure of CS Google HLL++




Update Speed: HLL8 vs HLLP
lgK=12, lgT=23-4
Nanoseconds
240
220
200
180
160
140
120
100
80
60
40
20
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
Uniques per Trial
HLL
HLLP


HyperLogLogPlus & HllSketch in Different Deserialization Modes
LgK=p=12, LgT=14,16
100,000
10,000
1,000
100
10
1
Nanoseconds
HLLP_Deser
HllSk_Deser
HllSk_Heapify
HllSk_Wrap
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
Uniques Per Trial

Innovations for Unique Counting Sketches (cont.)

FM85 / ICON, The Next Generation: Better than HyperLogLog

K. Lang, Back to the Future: an Even More Nearly Optimal Cardinality Estimation Algorithm, arxiv.org/abs/1708.06839 (preparing for publication)

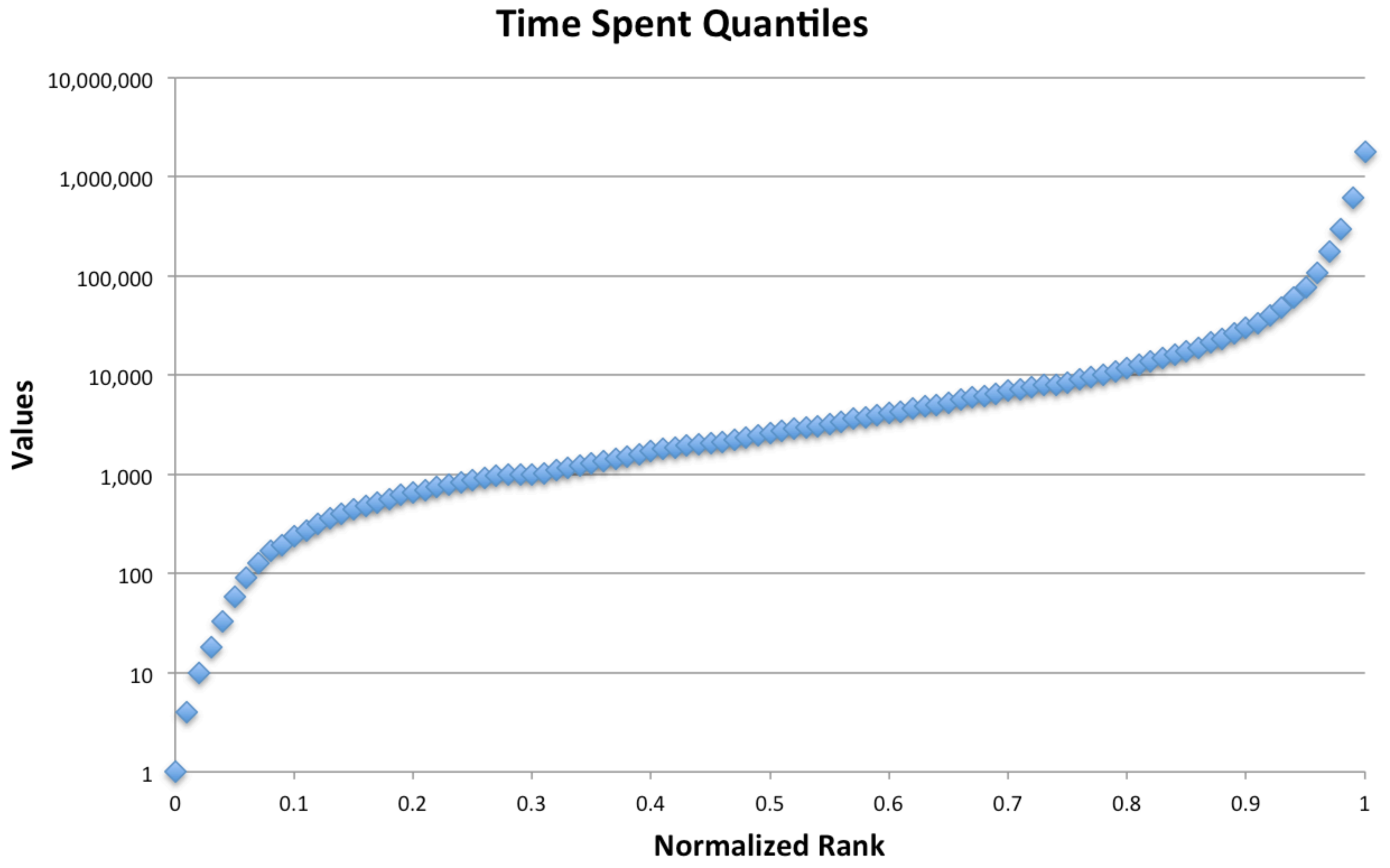
- Builds on Flajolet-Martin 1985 “Probabilistic Counting Algorithms For Data Base Applications”
- Three new estimators: all more accurate than original paper estimators
- More accurate per bit-of-entropy than Flajolet’s 2008 HLL sketches
- Simultaneously wins on all three dimensions of the time/space/accuracy tradeoff.
- Practical implementation is possible
- Already partially implemented in DataSketches HLL sketches

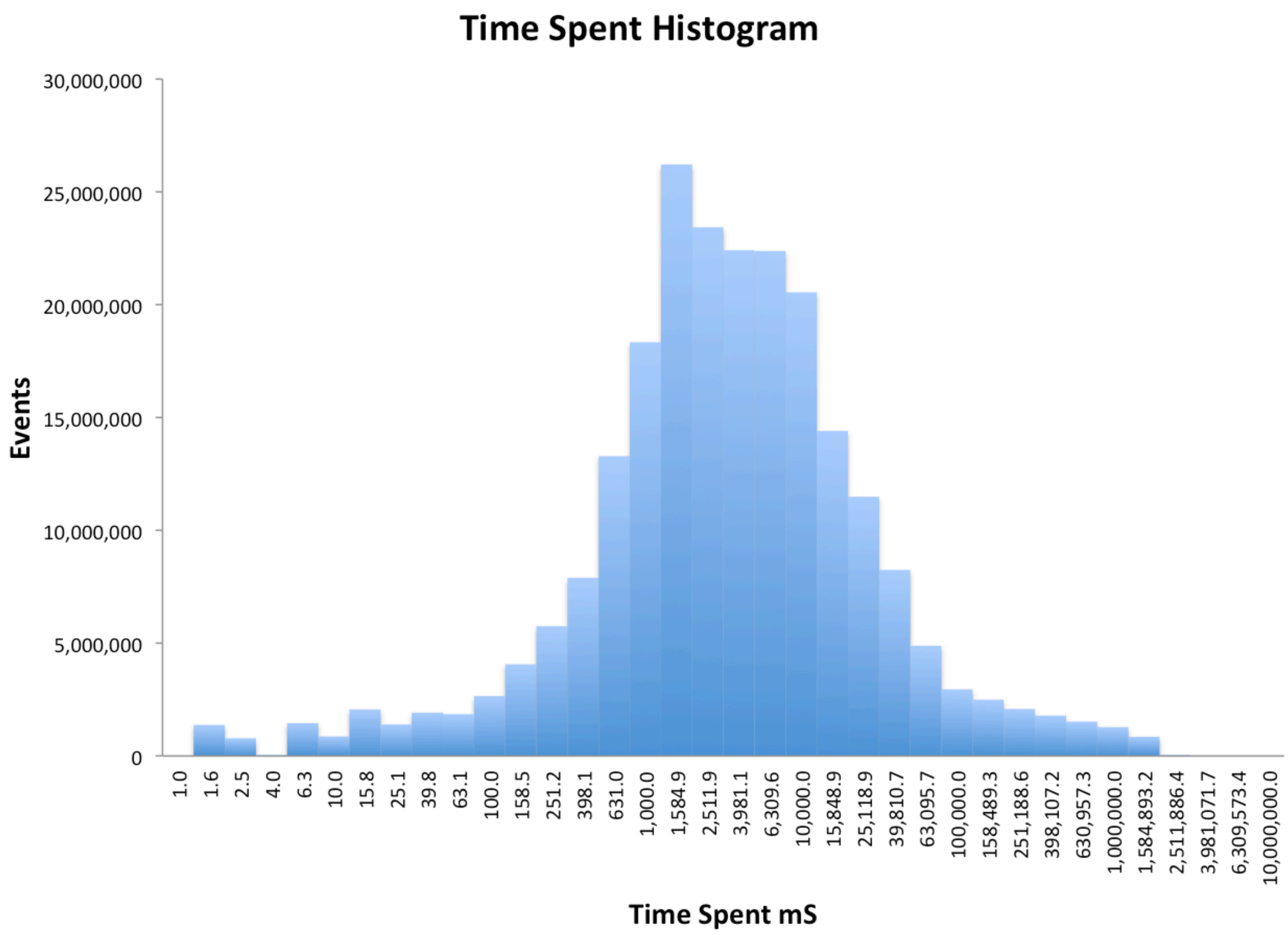
Innovations for Quantiles / Histogram Sketches

Quantiles, PMF's and CDF's of streams of comparable objects

Z. Karnin, K. Lang, E. Liberty: Optimal Quantile Approximation in Streams, *IEEE FOCS*, 2016

- Resolves one of the longest standing basic problems in the streaming computational model: The optimal construction of quantile sketches.
- The library implementation is simplified for speed performance
- Operates both On-Heap and Off-Heap


Time Spent Quantiles
Values
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Rank



Time Spent Histogram
Events
30,000,000
25,000,000
20,000,000
15,000,000
10,000,000
5,000,000
0
1.0
1.6
2.5
4.0
6.3
10.0
15.8
25.1
39.8
63.1
100.0
158.5
251.2
398.1
631.0
1,000.0
1,584.9
2,511.9
3,981.1
6,309.6
10,000.0
15,848.9
25,118.9
39,810.7
63,095.7
100,000.0
158,489.3
251,188.6
398,107.2
630,957.3
1,000,000.0
1,584,893.2
2,511,886.4
3,981,071.7
6,309,573.4
10,000,000.0
Time Spent mS

Innovations for Frequent Items Sketches

Frequent Items summaries for numerics and generic objects

E. Liberty, M. Mitzenmacher, J. Thaler, J. Ullman: Space Lower Bounds for Itemset Frequency Sketches, *ACM PODS*, 2016

D. Anderson, P. Bevan, K. Lang, E. Liberty, L. Rhodes, J. Thaler:
A High Performance Algorithm for Identifying Frequent Items in Data Streams.
ACM IMC 2017

- Handles weighted updates in amortized constant time
- Uses simple and fast method for merging sketches that improves on prior work.
- Currently implemented in our Library

Innovations for Weighted Sampling Sketches

An extension of Edith Cohen's VarOpt Paper

E. Cohen, N. Duffield, H. Kaplan, C. Lund, M. Thorup: Stream sampling for variance-optimal estimation of subset sums. *ACM-SIAM Symposium on Discrete Algorithms,* 2009*.*

- Created an innovative and efficient implementation
- Extended concepts in the paper to achieve merging with multiple size parameters.
- Currently implemented in our Library

Innovations for Vector / Matrix Sketches

Frequent Directions is Latest Family of Sketches

Mina Ghashami, E. Liberty, J. Phillips: Efficient Frequent Directions Algorithm for Sparse Matrices, *ACM KDD* 2016

- Approximate SVD for very large matrices.
- Created an innovative and efficient implementation
- Currently implemented in our Library

Major Sketch Families in DataSketches Library

Cardinality: Theta Sketch and HLL Sketch Families

- Theta: Cardinality & Set Expressions (e.g., Union, Intersection, Difference)
- HLL: Highly compact; HLL Map

Quantiles Sketches

- Quantiles, PMF's and CDF's of streams of comparable objects

Frequent Items Sketches

- Heavy Hitters from a stream of weighted objects

Tuple Sketches

- Theta Sketches with associated attributes

Reservoir and VarOpt Sketches

- Uniform and weighted sampling to fixed-*k* sized buckets

Vector & Matrix Sketches

- Frequent Directions (Approximate SVD)

Invitation for Collaboration

Thank You!

More material available on

DataSketches.GitHub.io