Apache® DataSketches™

A Production Quality Sketching Library for the Analysis of Big Data

Lee Rhodes
Distinguished Architect
Yahoo, Inc.

Outline

Problematic Queries of Big Data

Where traditional analysis methods don’t work well

Approximate Analysis Using Sketches

How using stochastic processes and probabilistic analysis
wins in a systems architecture context

The Open Source Apache DataSketches Library

A quick overview of this unique library dedicated to production systems
that process big data.

The Data Analysis Challenge ...

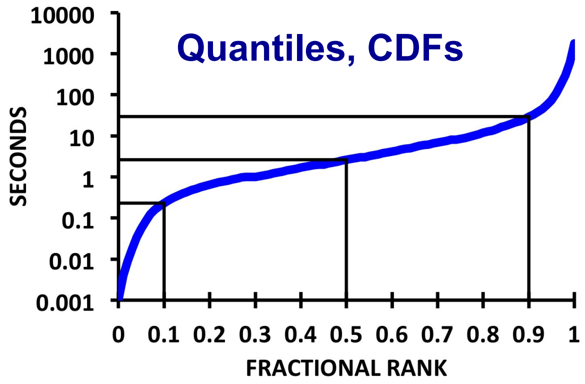
Example: Web Site Logs

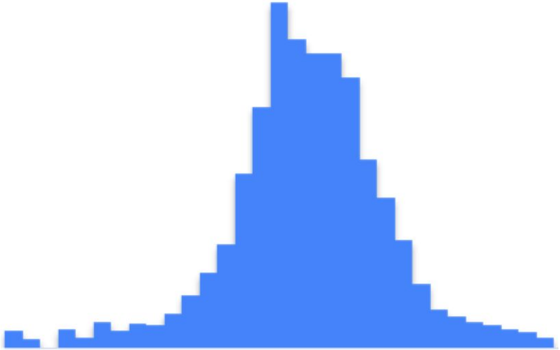
Time Stamp	User ID	Device ID	Site	Time Spent Sec	Items Viewed
9:00 AM	U1	D1	Apps	59	5
9:30 AM	U2	D2	Apps	179	15
10:00 AM	U3	D3	Music	29	3
1:00 PM	U1	D4	Music	89	10
Billions of Rows or *K,V* Pairs …					

… Analyze This Data In Near-Real Time.

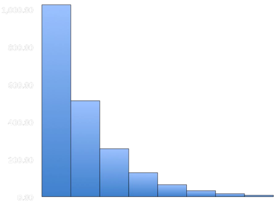
Some Very Common, but Problematic, Queries …

Unique Identifiers with Set Expressions: (A∪B)∩(C∪D) \ E





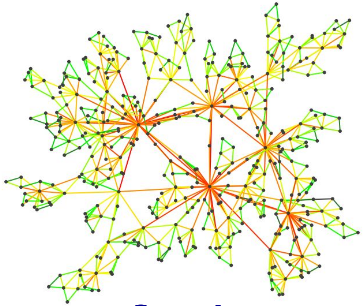
Histograms, PMFs



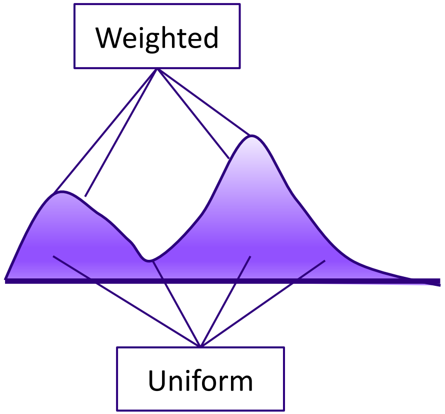
Frequent Items / Heavy Hitters

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

Vector & Matrix Operations: SVD, etc.



Graph Analysis



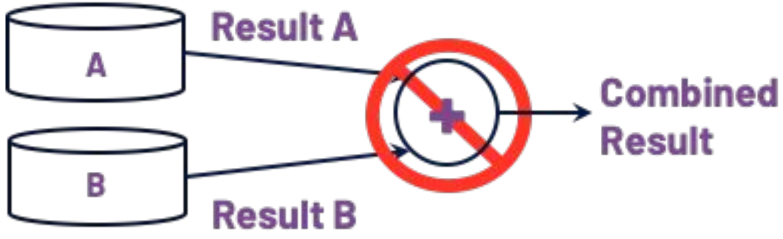
Reservoir Sampling

All Non-Additive

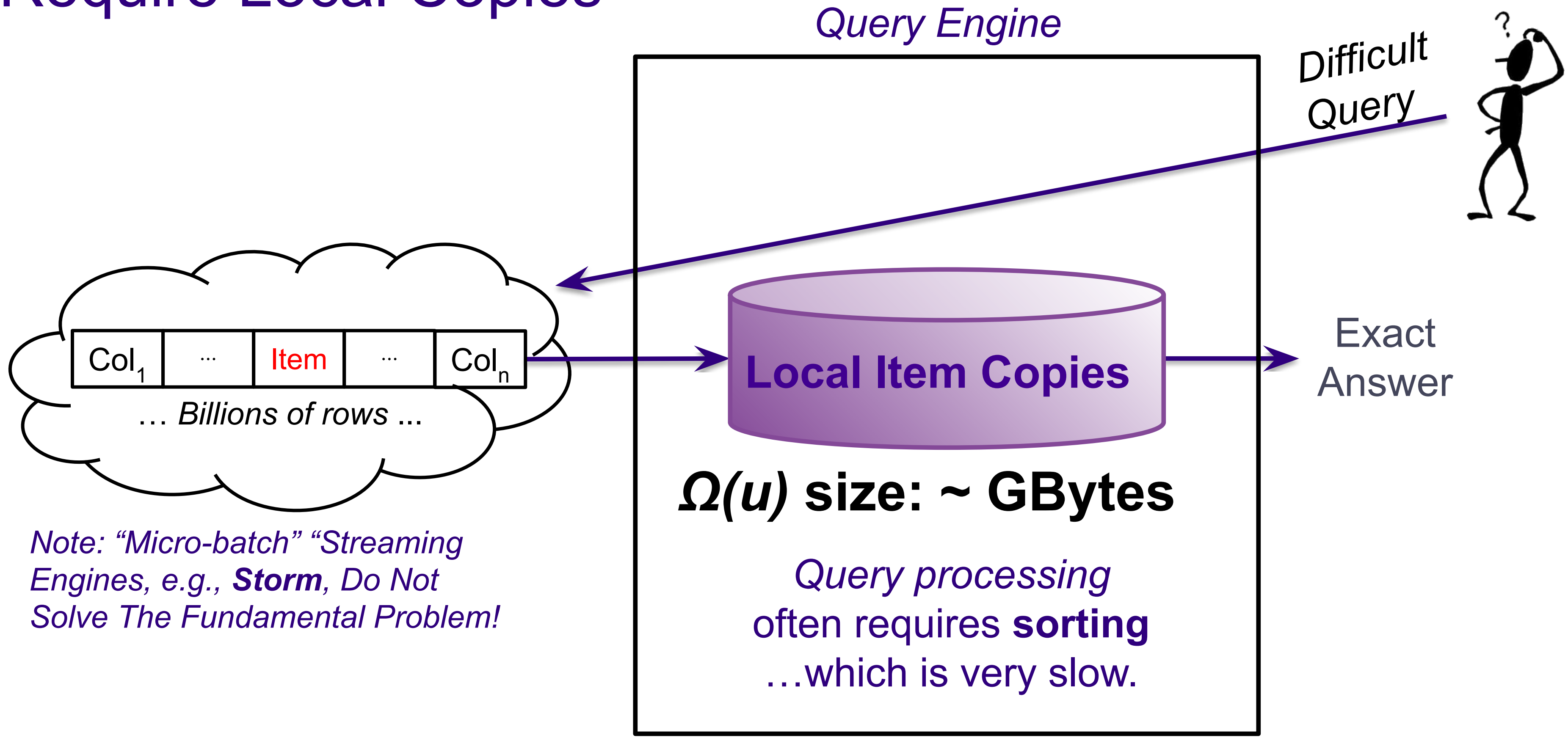
When The Data Gets Large Or Resources are Limited,

All Of These Queries Become Problematic

Because the Aggregations are Non-Additive or Non-Linear



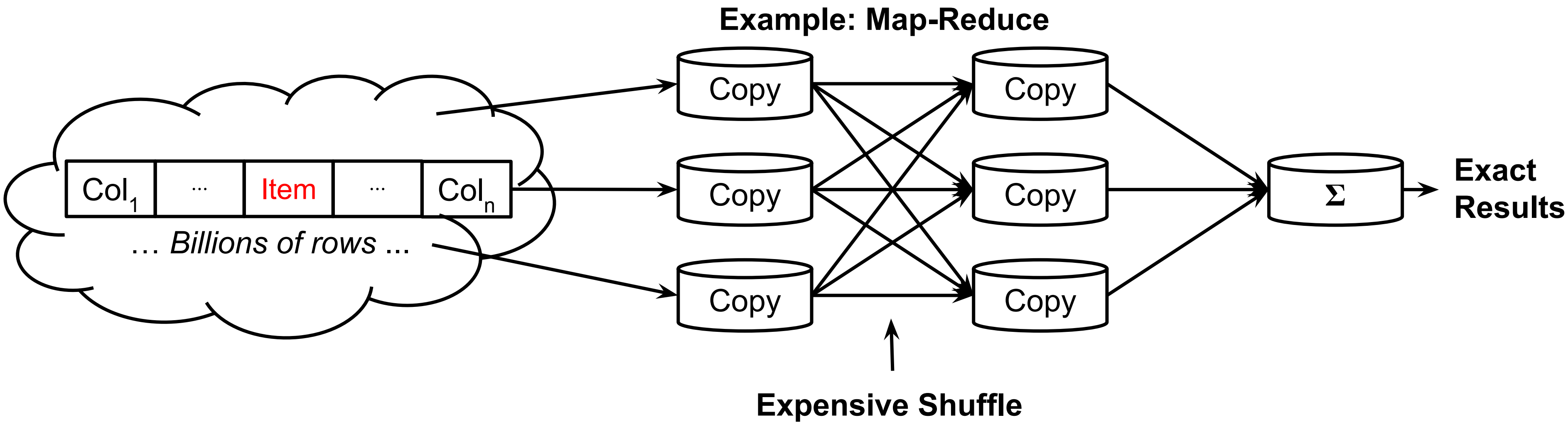
Traditional Exact Analysis Methods Require Local Copies



Note: "Micro-batch" "Streaming Engines, e.g., ***Storm****, Do Not Solve The Fundamental Problem!*

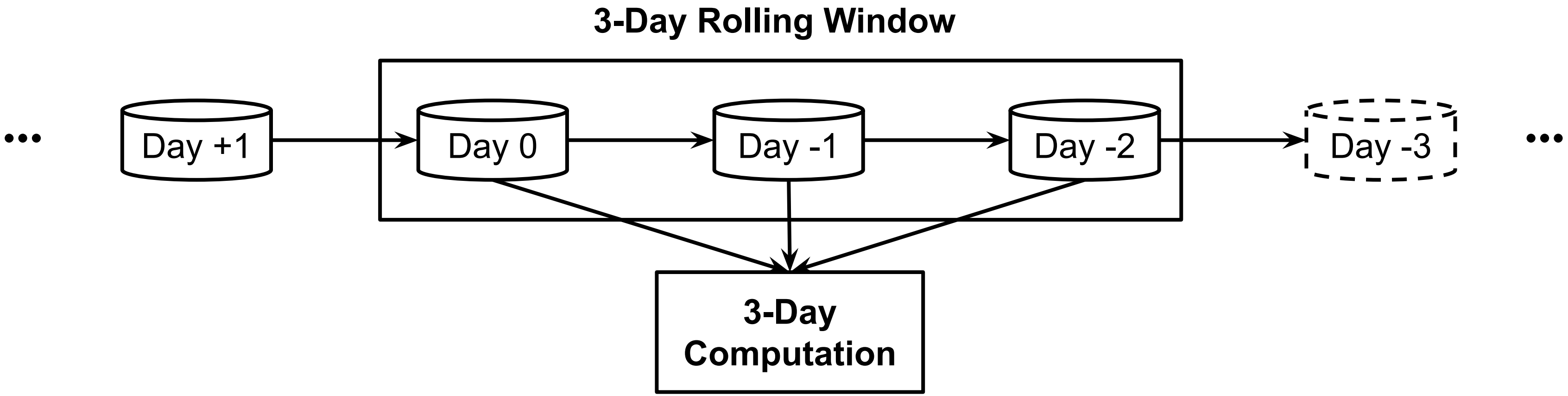
Parallelization Does Not Help Much

Because of Non-Additivity.
You have to keep the copies somewhere!
And with some operations, you may need ALL of the data always available.



Traditional Time Windowing

Requires Multiple Touches of Every Item in Every Daily Dataset



Every daily dataset is processed N times for a rolling N-day window!

Let’s challenge a fundamental premise:
… that our results must be exact!

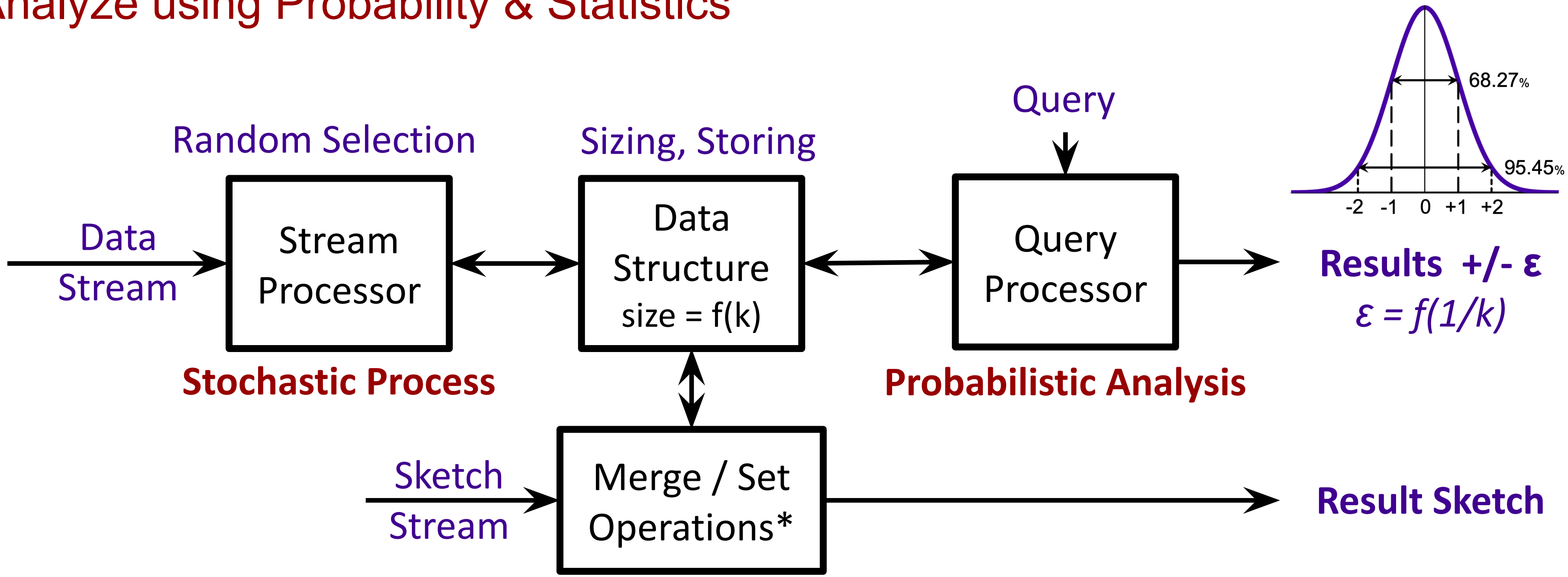
If we can allow for approximation,
along with some accuracy guarantees,

we can achieve orders-of-magnitude improvement in
- speed and
- reduction of resources.

Introducing the ***Sketch*** (a.k.a, Stochastic Streaming Algorithm)

Model the Problem as a Stochastic Process with a Dynamic Data Structure.

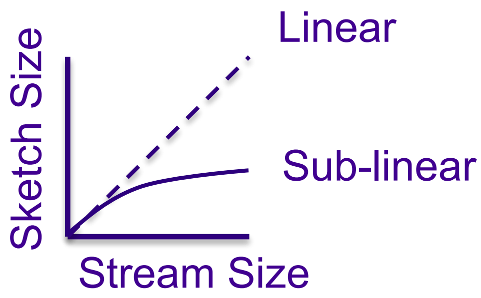
Analyze using Probability & Statistics



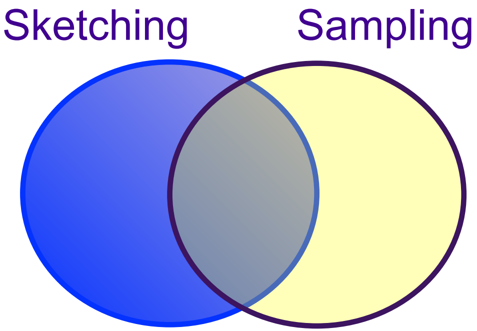
A Single Sketch Contains Many Algorithms

Key Sketch Properties Important to Us

- Small Stored Size
- Sub-linear in Space
- Single-pass, “One-Touch”
- Data Insensitive
- Mergeable
- **Mathematically Proven Error Bounds**



Sketches Overlap with Sampling



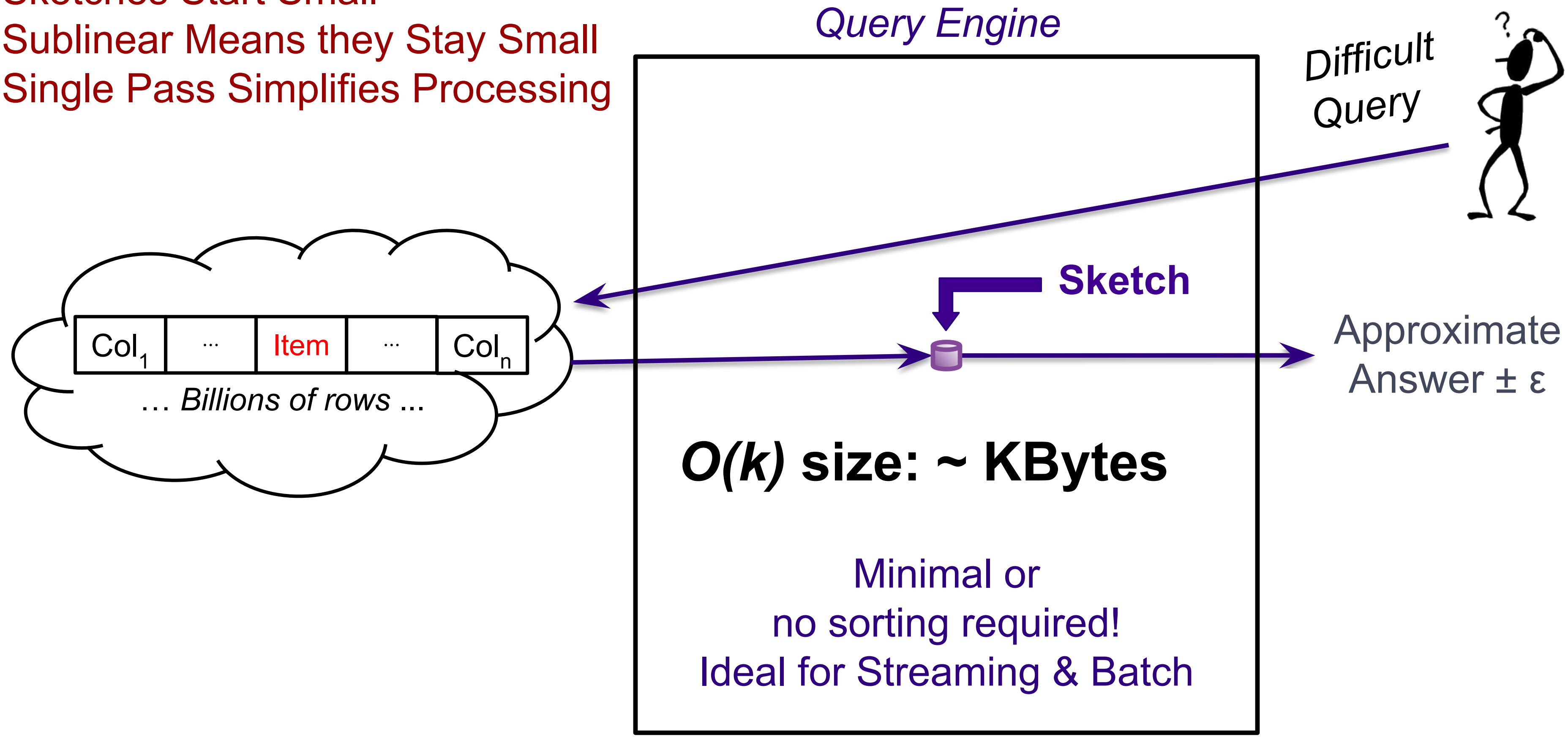
Based on the Specific Sketch

Why & How Sketches Achieve Superior Performance

For Systems Processing Massive Data

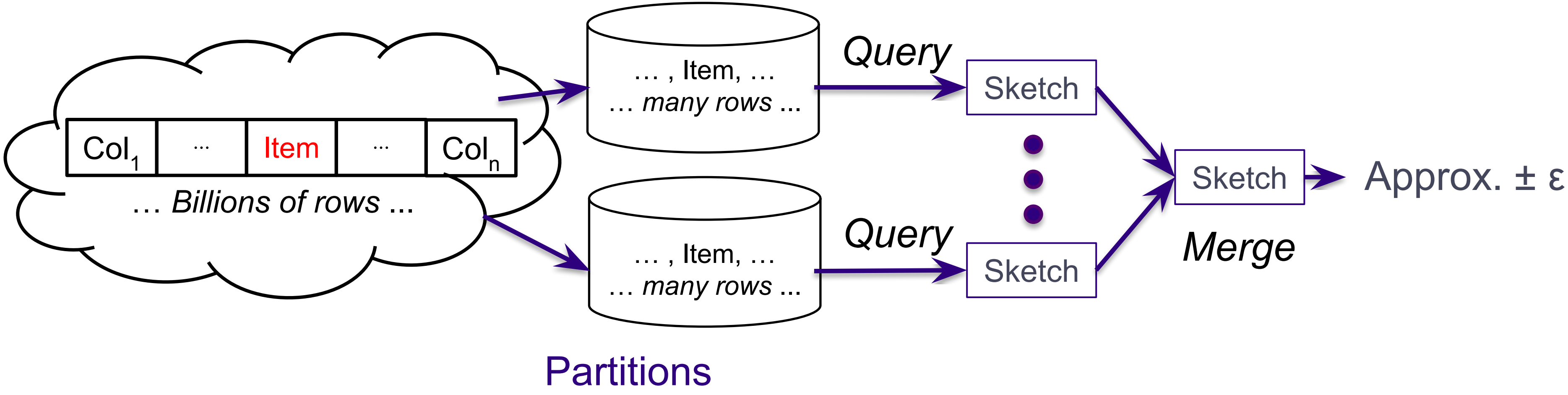
Win #1: Small Query Space

Sketches Start Small
Sublinear Means they Stay Small
Single Pass Simplifies Processing



Win #2: Mergeability

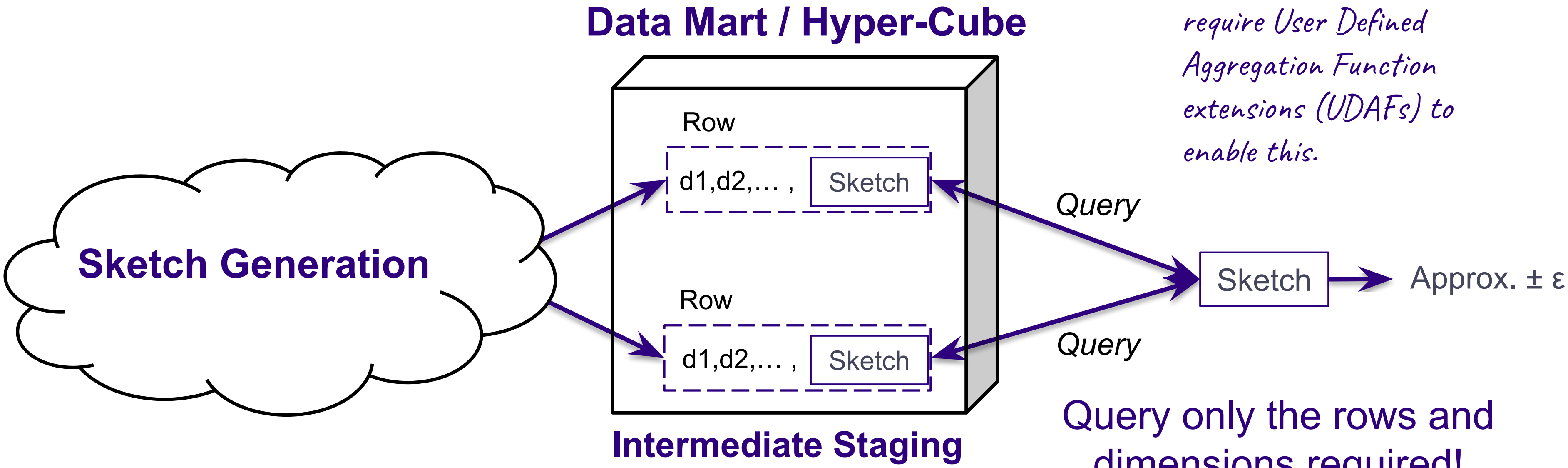
Mergeability Enables Parallelism … With No Additional Loss of Accuracy!
Sketches transform **Non-Additive** metrics into **Additive** Objects
The Result of a Sketch Merge is Another Sketch
… Enabling Set Expressions for Selected Sketches



Wins #3, 4: Speed, Simpler Architecture

Intermediate Hyper-Cube Staging Enables Query Speed
Additivity Enables Simpler Architecture
Sketches are small enough to store in the Hyper-Cube of other data!

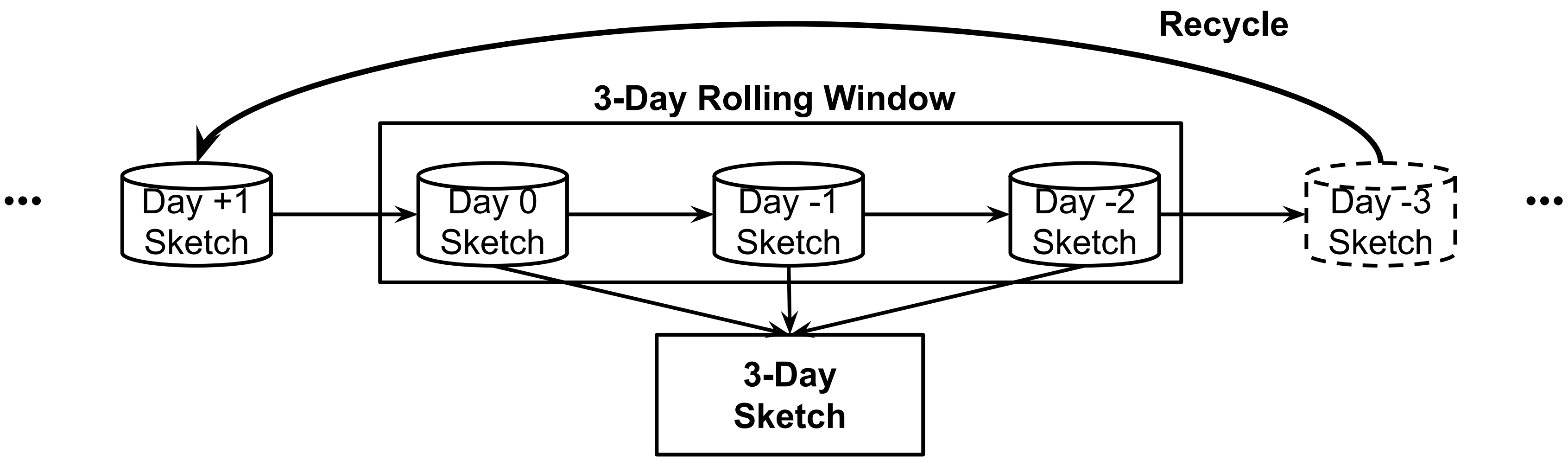
Proprietary DBs will often require User Defined Aggregation Function extensions (UDAFs) to enable this.



Query only the rows and dimensions required!

Stored Sketches Can Be Merged
By Any Dimensions, Including Time!

Win #5: Simplified Time Windowing & Late Data Processing



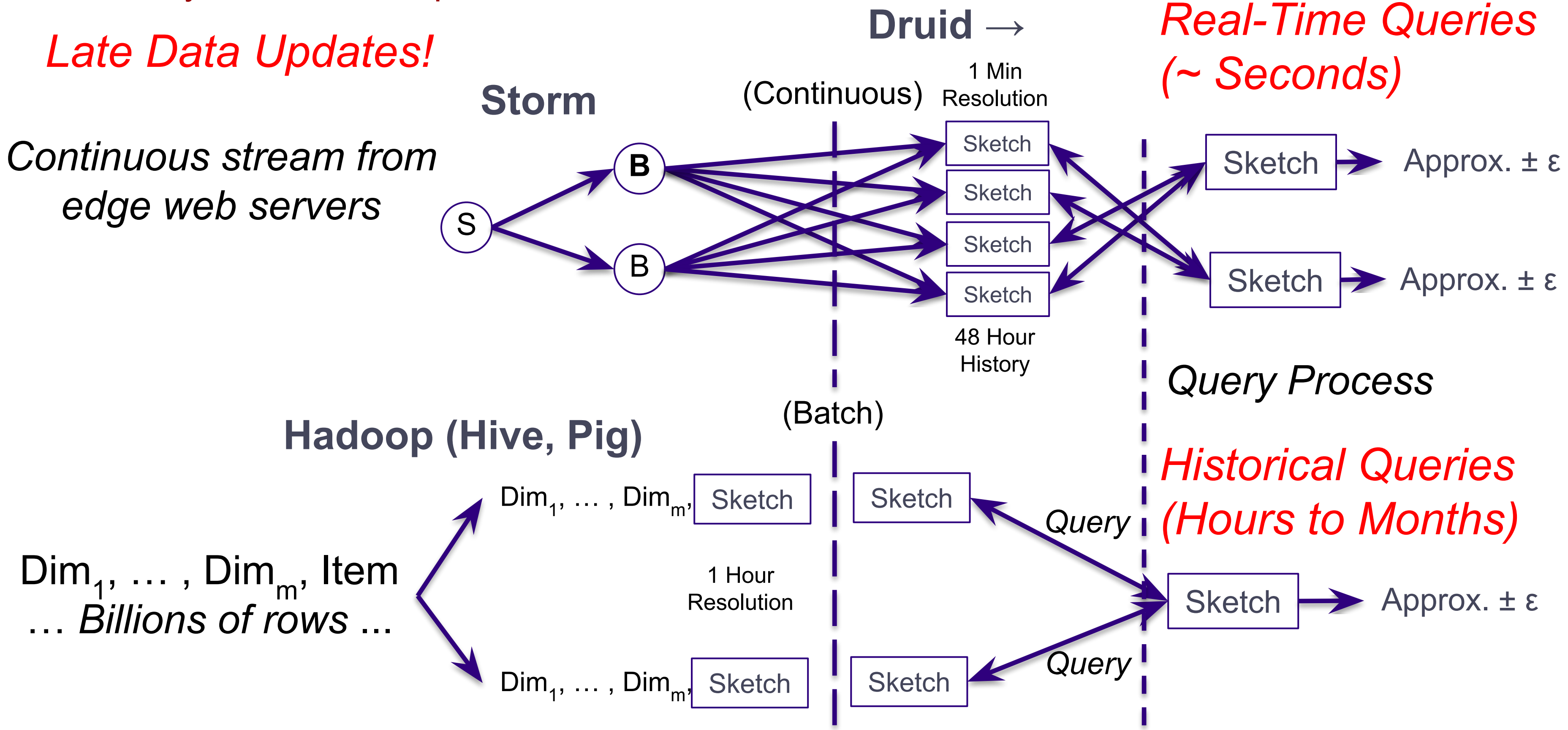
Every daily dataset is processed only **once** for a rolling N-day window!

Late-data processing is now possible.

Sketches can be recycled.

Near-Real Time Results, with History

Combine Off-Line, On-Line for Real-Time + History
Case Study: Storm/Hadoop/Druid Sketch Flow Architecture



Win #6: Lower System Cost ($)
Case Study: Real-time Flurry, Before & After

- Customers: >250K Mobile App Developers
- Data: 40-50 TB per day
- Platform: 2 clusters X 80 Nodes = 160 Nodes
  - Node: 24 CPUs, 250GB RAM

Big Wins!
Near-Real Time
Lower System $

	Before Sketches	After Sketches
VCS* / Mo.	~80B	~20B
Result Freshness	Daily: 2 to 8 hours; Weekly: ~3 days Real-time Results Not Feasible!	15 seconds!

* VCS: Virtual Core Seconds

Introducing

Apache® DataSketches™

https://datasketches.apache.org

Our Mission…

Combine Deep Science with Exceptional Engineering

To Develop **Production Quality** Sketches

That Address These Difficult Queries

The Sketch Design Process

1. ## The Art:

   Model a problem as a stochastic process and a data structure …

2. ## The Science:

   Analyze the data structure using probability, statistics to
   extract the desired result with well understood error properties.
   Prove that it works! Publish to Scientific Venues.
   https://datasketches.apache.org/docs/Community/Research.html

3. ## The Engineering:

   Transform the Art and the Science Theory into a Product!

   - Create useful APIs for use in production systems
   - Document with code examples for system engineers
   - Exhaustively test & characterize to ensure robustness
   - Publish to Open Source

The **Apache DataSketches** Library

Cardinality, 4 Families

- **HLL (on/off Heap)** A very high performance implementation of this well-known sketch
- **CPC** The best accuracy per space
- **Theta Sketches**: Set Expressions (e.g., Union, Intersection, Difference), on/off Heap
- **Tuple Sketches**: Generic, Associative Theta Sketches, multiple derived sketches:

Quantiles Sketches, 3 Families

- **Quantiles**, Histograms, PMF's and CDF's of streams of comparable objects, on/off Heap.
- **KLL**, highly optimized for accuracy-space.
- **Relative Error Quantiles,** Extremely accurate at the ends of the rank domain

Frequent Items (Heavy-Hitters) Sketches, 3 Families

- **Frequent Items**: Weighted or Unweighted
- **Frequent Directions** (Approximate SVD) (a Vector Sketch)
- **Frequent Distinct Tuples**: Multi-dimensional Frequency & Distinct Analysis

Reservoir and VarOpt (Edith Cohen) Sketches, 2 Families

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

Specialty Sketches

- **Customer Engagement,, Maps**, etc.

Languages Supported:

- Java, C++, Python
- Binary Compatibility across languages

Bright Future for Sketching Technology & Solutions ...

Items (words, IDs, events, clicks, …)

- Count Distinct
- Frequent Items, Heavy-Hitters, etc
- Quantiles, Ranks, PMFs, CDFs, Histograms
- Set Operations
- Sampling
- Mobile (IoT)
- Moment and Entropy Estimation

Graphs (Social Networks, Communications, …)

- Connectivity
- Cut Sparsification
- Weighted Matching
- …

Vectors (text docs, images, features, …) And Matrices (text corpora, recommendations, …)

- Dimensionality Reduction (SVD)
- Ridge Regression
- Covariance Estimation
- Low Rank Approximation
- Sparsification
- Clustering (k-means, k-median, …)
- Linear Regression
- Machine Learning (in some areas)
- Density Estimation

Areas where we have sketch implementations

Areas of research (World-wide)

THANK YOU!

Open Invitation for Collaboration

Learn more about Apache DataSketches

Come and Visit Us!

https://datasketches.apache.org