

# DataSketches

A Required Toolkit for the  
Analysis of Big Data

Lee Rhodes, Kevin Lang  
Yahoo, Inc. | 13 Oct 2016

# The Challenge

## Web Site Logs

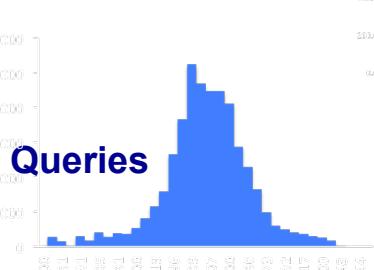
| Time     | User ID | Site  | Time Spent Sec | Items Viewed |
|----------|---------|-------|----------------|--------------|
| 9:00 AM  | U1      | Apps  | 59             | 5            |
| 9:30 AM  | U2      | Apps  | 179            | 15           |
| 10:00 AM | U3      | Music | 29             | 3            |
| 1:00 PM  | U1      | Music | 89             | 10           |

Billions of rows ...

## Unique User Queries



Quantile Queries

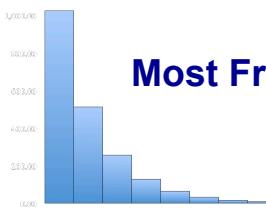


## Financial Transactions System Log

| Time                 | User ID | Site  | Purchased   | Revenue |
|----------------------|---------|-------|-------------|---------|
| 9:00 AM              | U1      | Apps  | FaceTune    | \$3.99  |
| 9:30 AM              | U2      | Apps  | Minecraft   | \$6.99  |
| 10:00 AM             | U3      | Music | Purple Rain | \$1.29  |
| Billions of rows ... |         |       |             |         |

Billions of rows ...

## Most Frequent Queries



# Unique Count Queries



Apps

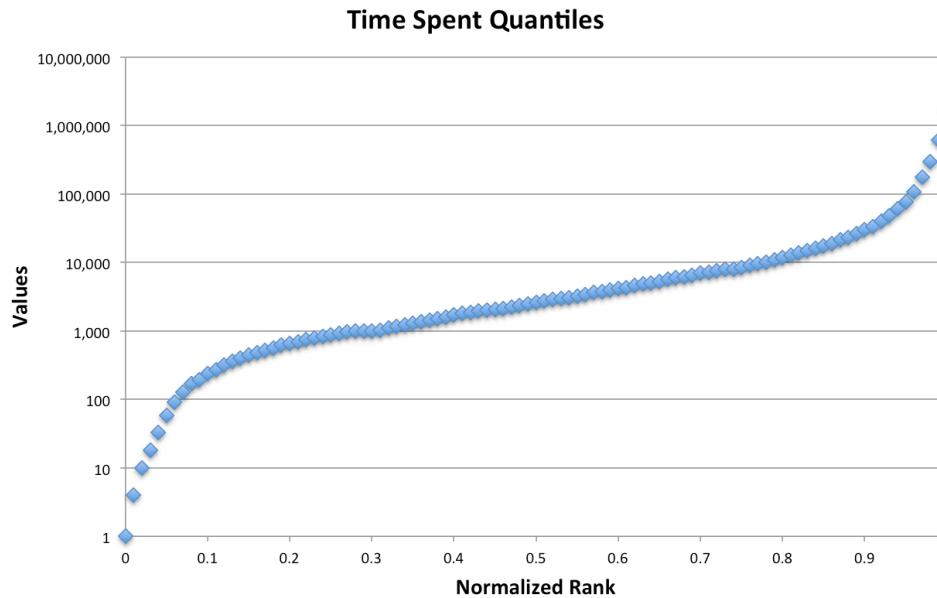


Music

Number of Unique Users that visited ...

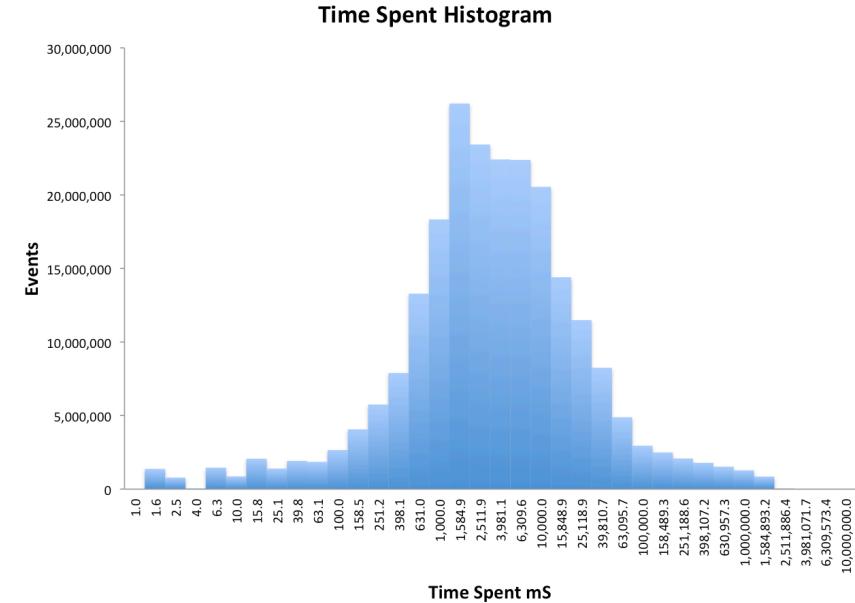
- Apps?
- Music?
- Apps OR Music?
- Apps AND Music?
- Apps AND NOT Music?
- Etc.

# Quantiles Queries



#### Time-Spent Quantiles:

- 90<sup>th</sup> %ile:  $Q(.9) \approx 30$  sec
- Median:  $Q(.5) \approx 3$  sec
- 10<sup>th</sup> %ile:  $Q(.1) \approx 0.3$  sec



#### Histogram (Frequency Distribution):

- 1 sec < # Events < 3 sec

#### PMF (Probability Mass Function):

- 1 sec < % Events < 3 sec

# Frequent Items Queries

| Song Titles            | Frequency |
|------------------------|-----------|
| One Dance              | 1,234,567 |
| Can't Stop The Feeling | 987,654   |
| Cheap Thrills          | 321,234   |
| ...                    | ...       |

## Song Titles:

- What songs are the most frequent?
- What is the frequency of “One Dance”

# Why are these operations so difficult with Big Data?

## Computational Theory for Exact Computation of Single Streams

| Problem<br>(Exact) | Lower Space<br>Bound | Means:  |
|--------------------|----------------------|---|
| Distinct Count     | $\Omega(u)$          | Must keep at least one copy of every unique value |
| Frequent Items     |                      |   |
| Quantiles          |                      |   |

$n$  = Total Items

$u$  = Unique Items,  $u < n$

$\Omega(\bullet)$  = Lower Bound, hiding constant factors

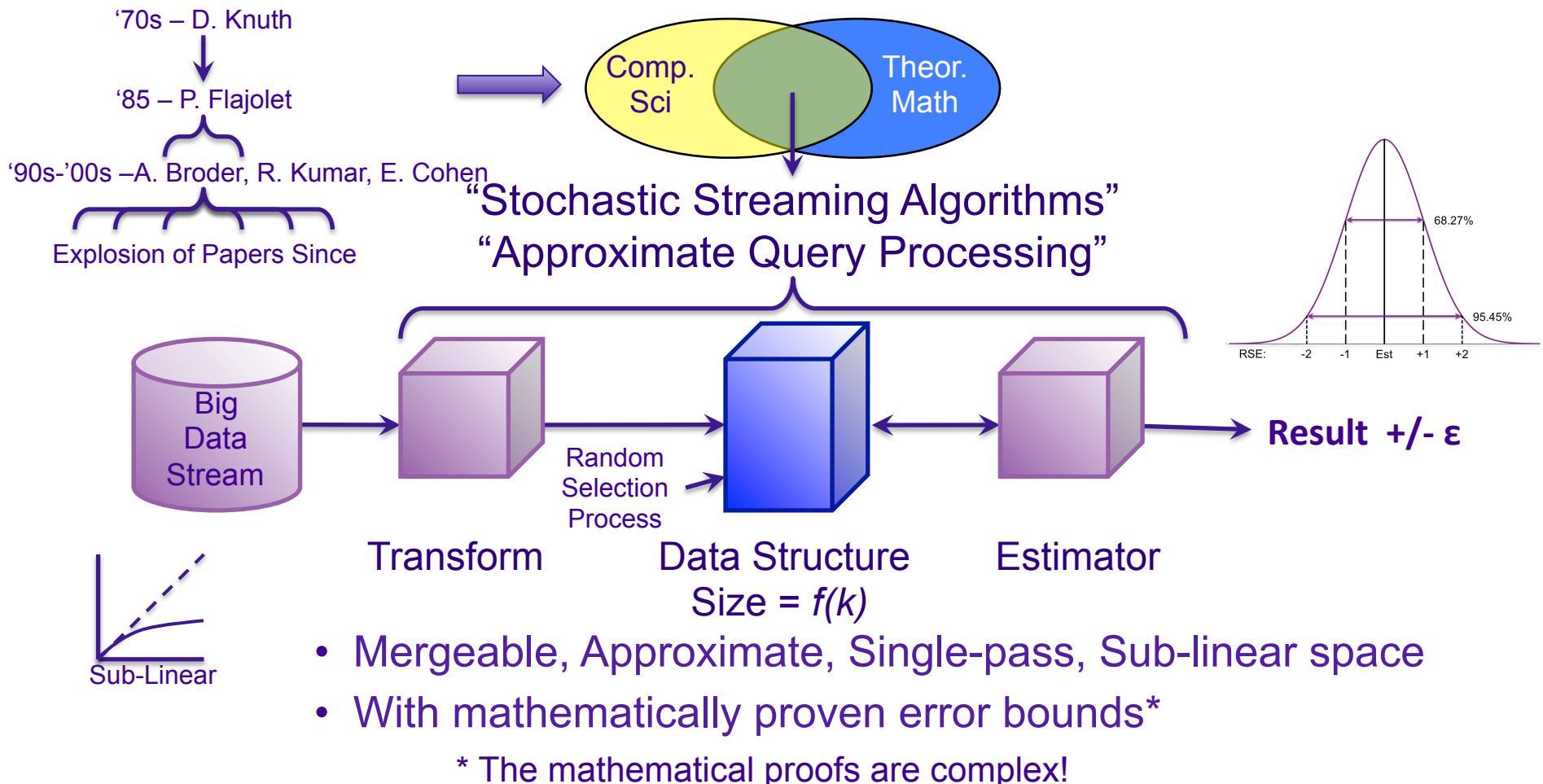
If An Approximate Answer Is Acceptable ...

And You Are Concerned About:

- Speed
- Resource Utilization
- Flexibility
- Well defined accuracy

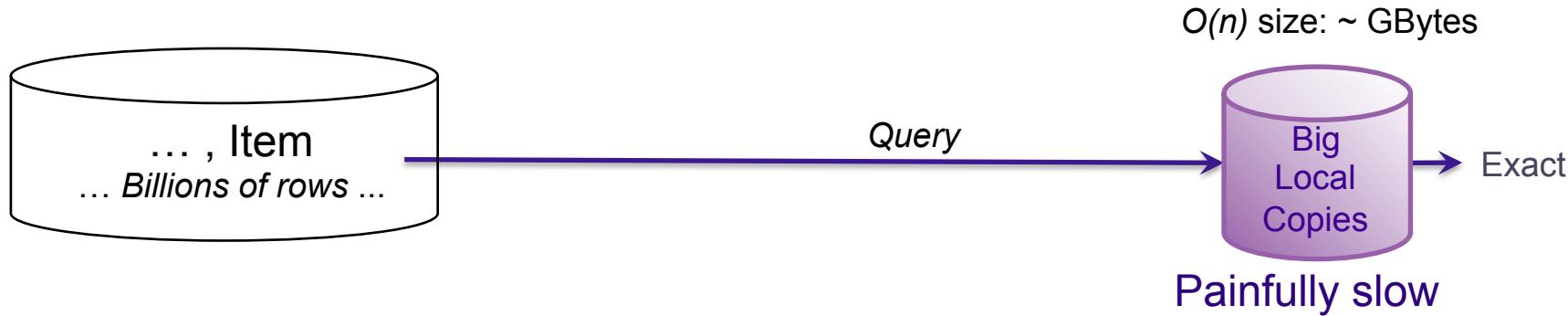
Sketches Offer an Excellent Solution!  
( ... and may be the only solution! )

# “Sketch” Applies to a Broad Range of Algorithms

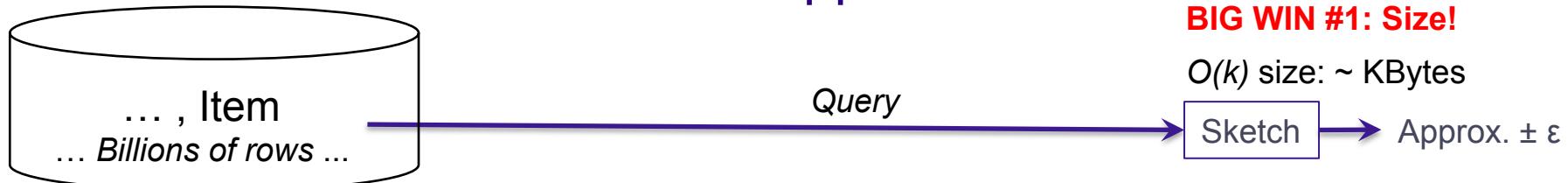


# Architecture: Processing Big Data

## Traditional (Exact) Approach



## Sketch Approach



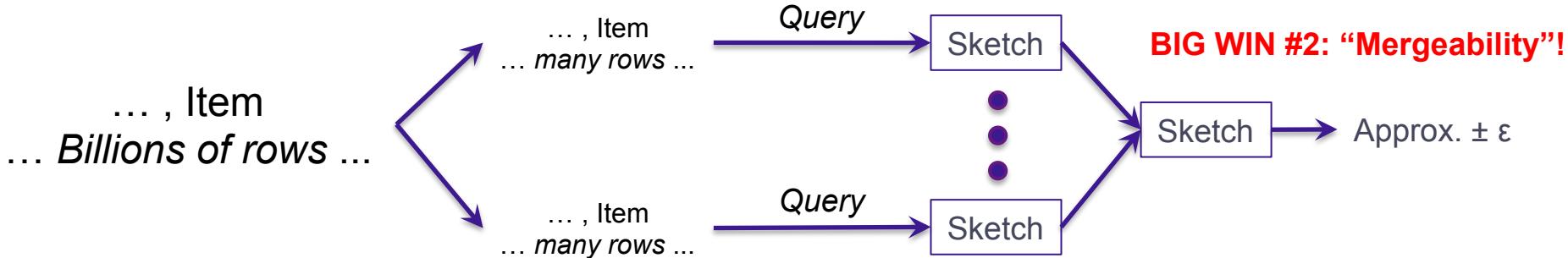
Still difficult to process all the rows fast

# Architecture: Processing Big Data

Difficult to process all the rows fast



Parallel, Multiple Stream: Partitioning, data skew can be a challenge

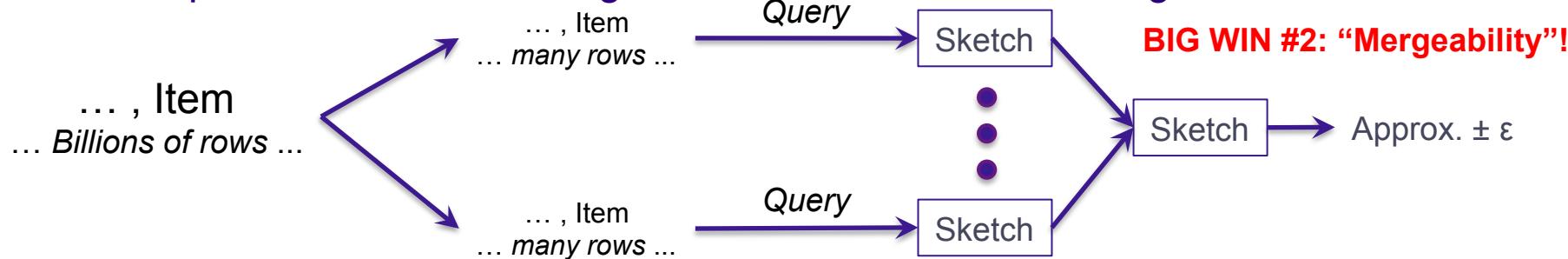


# Architecture: Processing Big Data

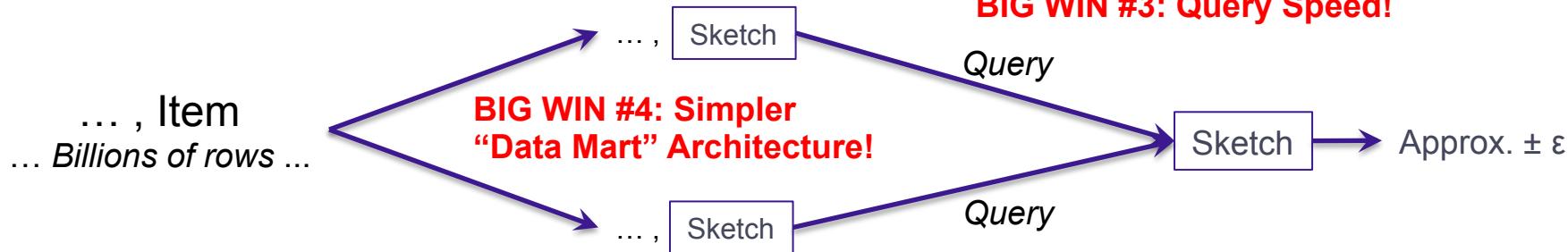
Difficult to process all the rows fast



Parallel, Multiple Stream: Partitioning, data skew can be a challenge



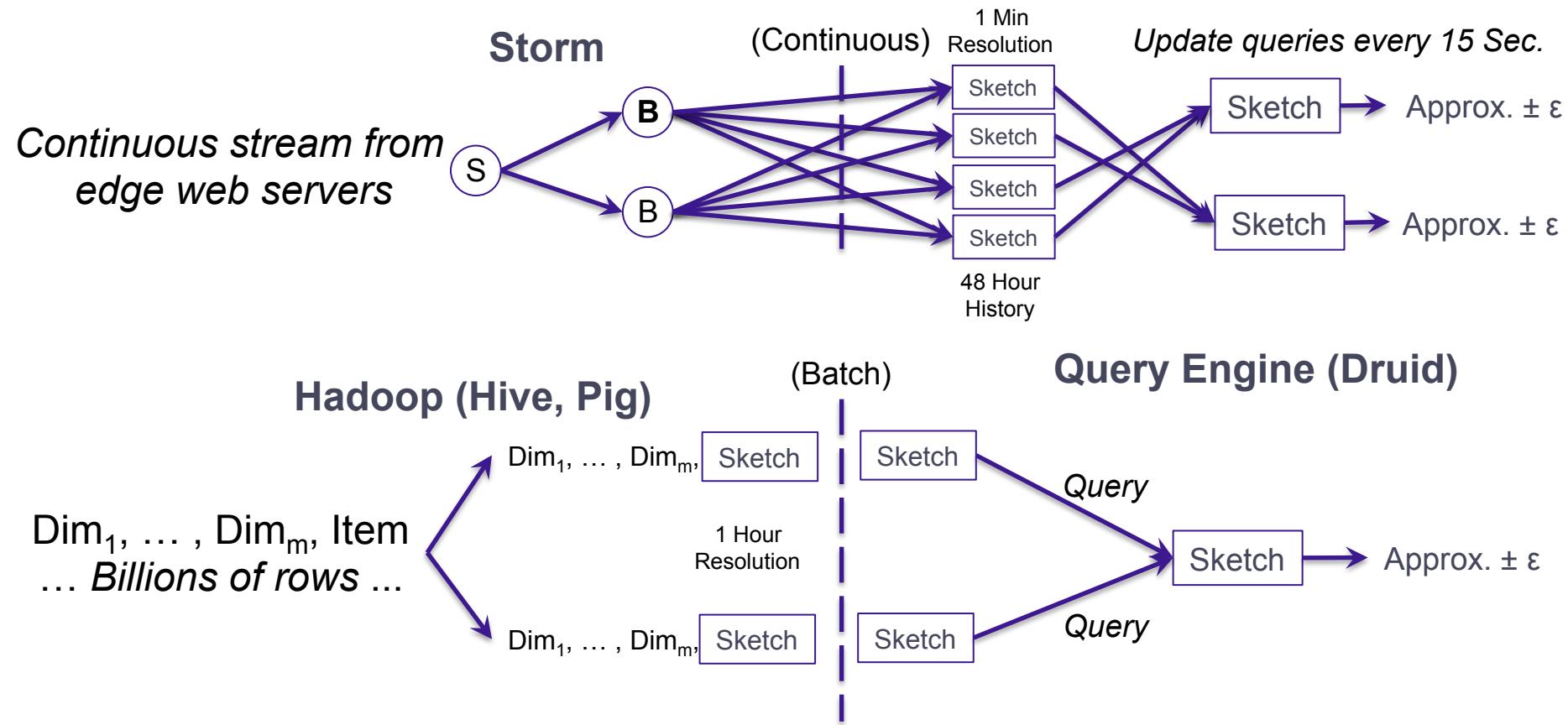
Parallel, Multiple Stream, Intermediate sketch staging:



# Case Study: Flurry Sketch Flow Architecture

BIG WIN #5: Real Time!

BIG WIN #6: Late Data Updates!



## Real Time

All Apps (337) ▾

Android Yahoo Mail++ - Production XiPhone Yahoo Mail++ - Production X

+

+

+

## Usage

## Retention

## Active Devices

## Active Users

## Audience

## Events Summary

## Events Details

## Real Time Dashboard

ACTIVE DEVICES LAST 24 HOURS

**15,365,820**

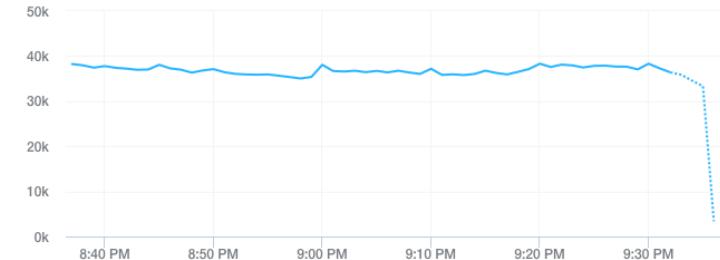
SESSIONS TODAY

**72,541,482**

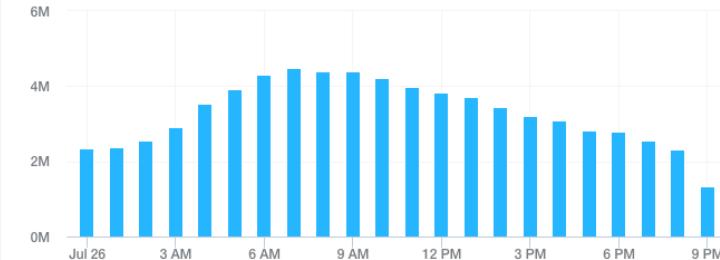
SESSIONS LAST 24 HOURS

**77,724,619**

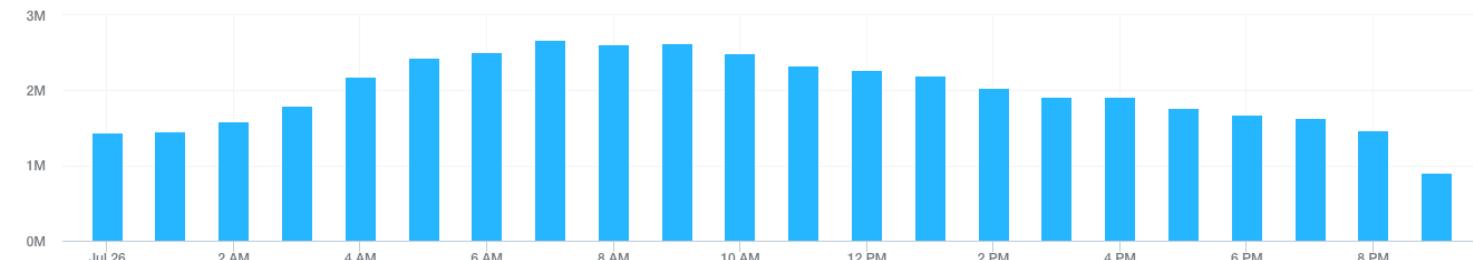
## SESSIONS - MINUTE



## SESSIONS - HOUR



## ACTIVE DEVICES - HOUR



Real Time

Usage

**Retention**

Active Devices

Active Users

Audience

Events Summary

Events Details

All Apps (337) ▾

Yahoo Mail++ - Production



7News - Dogfood



Jun 27, 2016 - Jul 26, 2016 ▾

## Retention Dashboard

AVG DAY 7 RETURN RATE

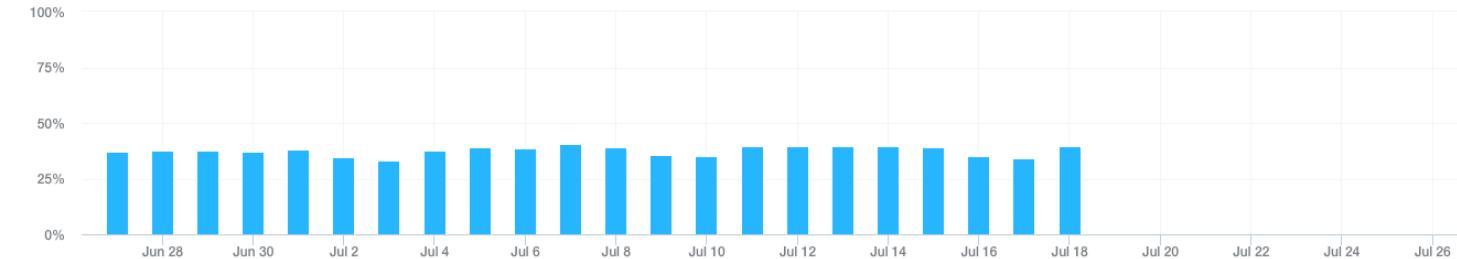
**37.63%**

AVG DAY 7 ROLLING RATE

**64.22%**

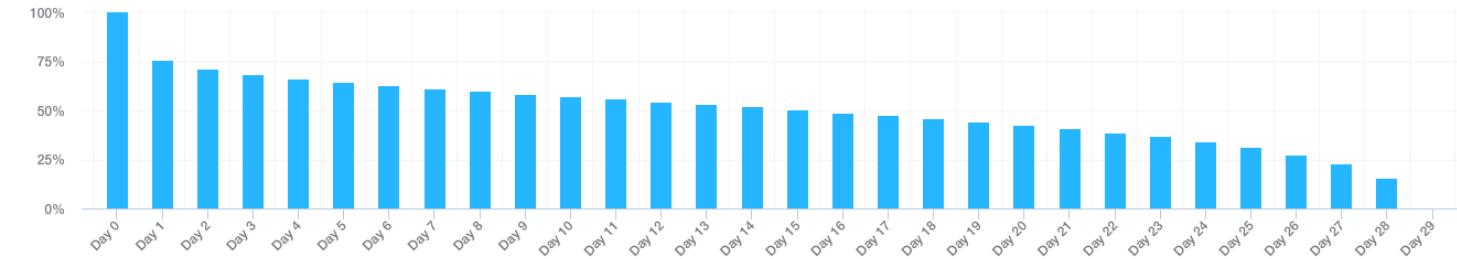
PERCENT RETURNING ON DAY 7 ▾ AFTER INSTALL

DAY ▾



ROLLING RETENTION

DAY ▾



# Case Study: Flurry, Before and After

- Customers: >250K Mobile App Developers
- Data: 40-50 TB per day
- Platform: 2 clusters X 80 Nodes / cluster; each Node is 24 CPUs/48 hyper-threads, 250GB RAM

|                               | Before Sketches   | After Sketches  |
|-------------------------------|---|---|
| System environment            | <b>Hadoop:</b> complete brute-force, pre-computation of uniques<br><b>HBase:</b> Query Engine | <b>Hadoop:</b> w/ Sketches<br><b>Storm:</b> Real-Time w/ Sketches<br><b>Druid:</b> Query Engine w/ Sketches |
| Cost (Virtual Core-Sec) (VCS) | <b>Mo Total: ~80B vcs</b>   | <b>Mo. Total: ~20B vcs</b><br><b>BIG WIN #7: Eco Friendly! Lower \$</b>                                     |
| Result Freshness              | <b>Daily: 2 to 8 hours</b><br><b>Weekly: ~3 days</b><br><b>Real-time Not Feasible</b>         | <b>15 seconds!</b>  |
| Flexibility                   | Uniques not additive,<br>Must always compute from raw data,<br>Not flexible.                  | Sketches are additive across time + other dim.<br>Allow set expressions,<br>Architecturally simpler         |

# DataSketches.GitHub.io Open Source Library

- Dedicated to **production quality** Sketch implementations.
- Common Attributes
  - Single pass
  - Mergeable
  - Sub-Linear
  - In-sketch error bounds estimation
  - Designed for large-scale computing environments:
    - Easy integration
    - Hadoop/Pig, Hadoop/Hive, Druid Adaptors
    - Standard library promotes sharing across platforms and organizations
  - Maven deployable, registered
  - Comprehensive unit tests
  - Extensive documentation
  - All algorithms are backed by published mathematical theory

# Current Sketch Implementations

## Theta Sketch\* and HLL Sketch Families

- Estimating Cardinality
- Theta Sketch Set Expressions (e.g., Union, Intersection, and Difference)

## Quantiles Sketches\*

- Normal or Inverse PMF's, CDF's of streams of comparable values, using after-the-fact queries.

## Frequent Items Sketches\*

- Identify the Heavy Hitters of arbitrary objects from a stream of objects
- Estimate the frequency of any item from the stream

## Tuple Sketches: Count Unique Sketches with Attributes\*

- Shares similar properties as the Theta Sketches

\* Requires code extensions of Java generic base classes for non-numerics

# Thank You!

More material available on

*DataSketches.GitHub.io*