Persistent Data Sketching

Zhewei Wei
Renmin University of China

Ge Luo
The Hong Kong University of Science and Technology

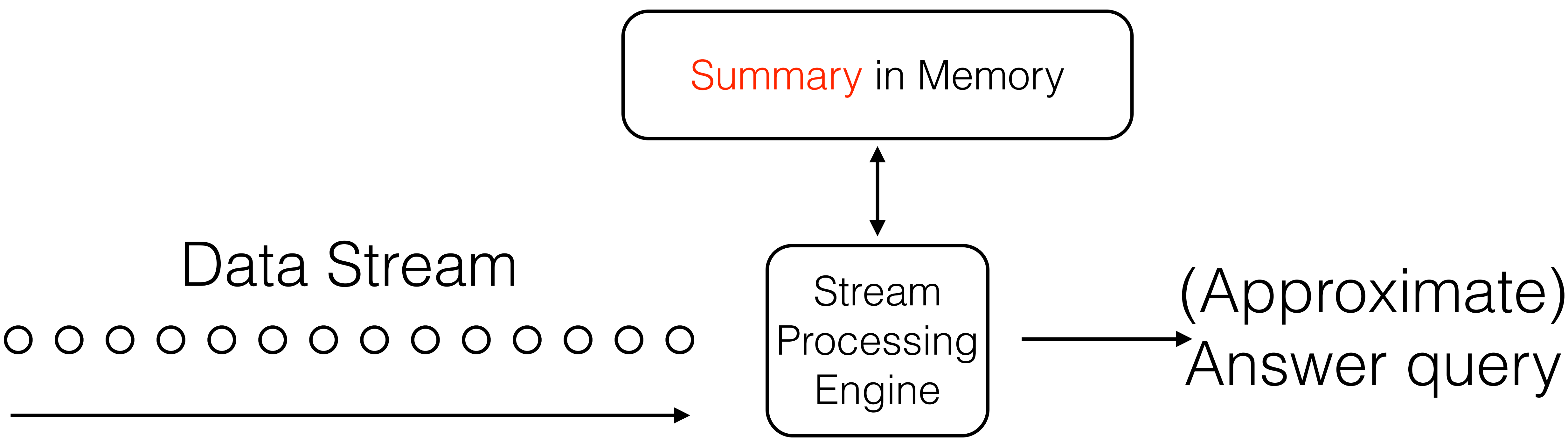
Ke Yi
The Hong Kong University of Science and Technology

Xiaoyong Du
Renmin University of China

Ji-Rong Wen
Renmin University of China

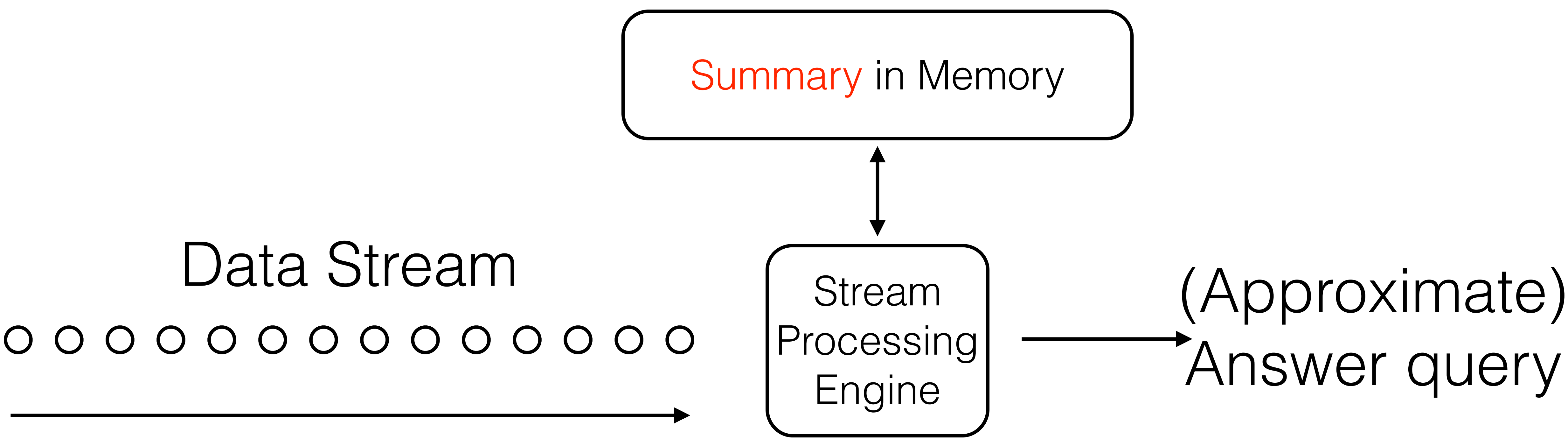
Streaming Algorithms

- A data stream is a (massive) sequence of data



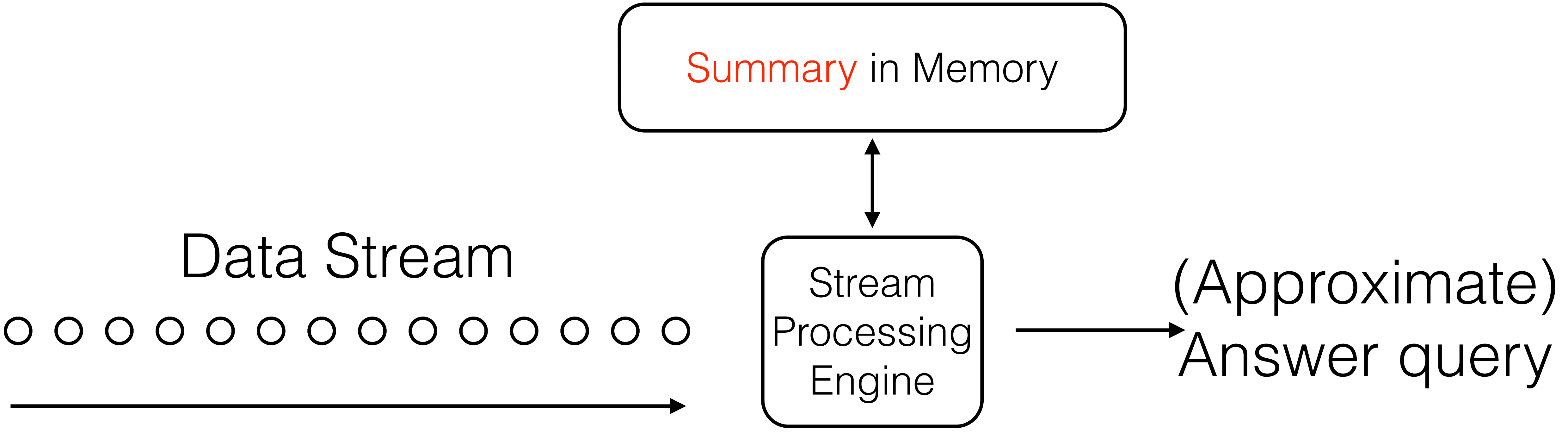
Streaming Algorithms

- A data stream is a (massive) sequence of data
  - Single Pass: Each record is examined at most once



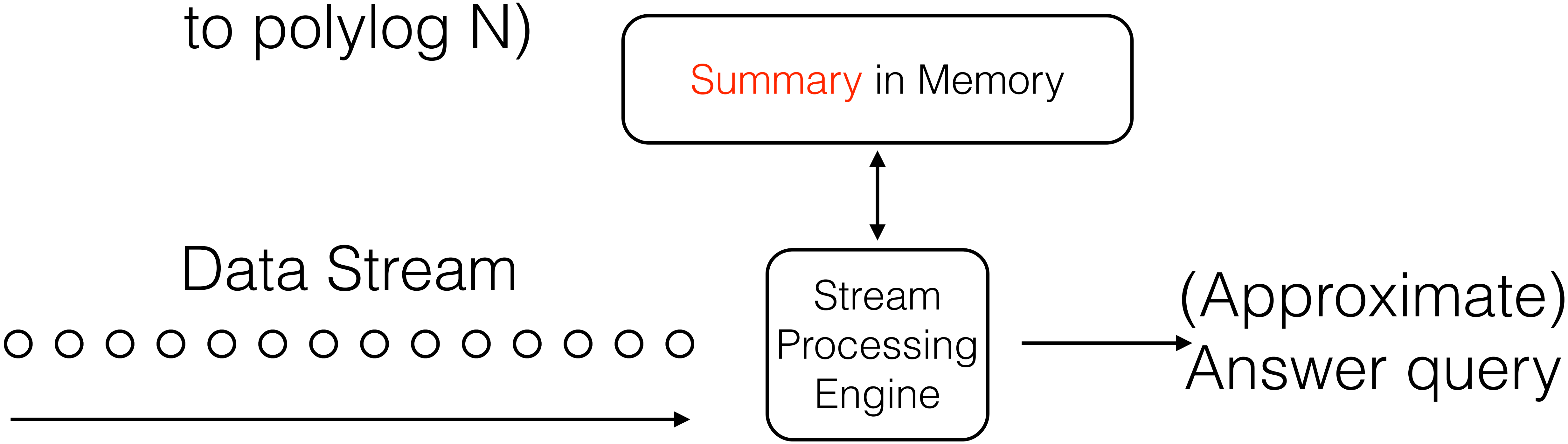
Streaming Algorithms

- A data stream is a (massive) sequence of data
  - Single Pass: Each record is examined at most once
  - Small Space: Log or polylog in data stream size



Streaming Algorithms

- A data stream is a (massive) sequence of data
  - Single Pass: Each record is examined at most once
  - Small Space: Log or polylog in data stream size
  - Small time: Low per-record processing time (O(1) to polylog N)



Sketches

- Sub-linear space
  - Fast update and query time

- Answer queries approximately

- Linear transformation of the data frequencies

Sketches

- Count-Min Sketch [Cormode and Muthukrishnan 2005]
  - Point queries, heavy hitters (frequent items)

- AMS Sketch [Alon et. al. 1999]
  - Frequency moments

- Count Sketch [Charikar et. al. 2002]
  - Join size queries, self join size queries [Rusu and Dobra 2007]

- …

Sketches

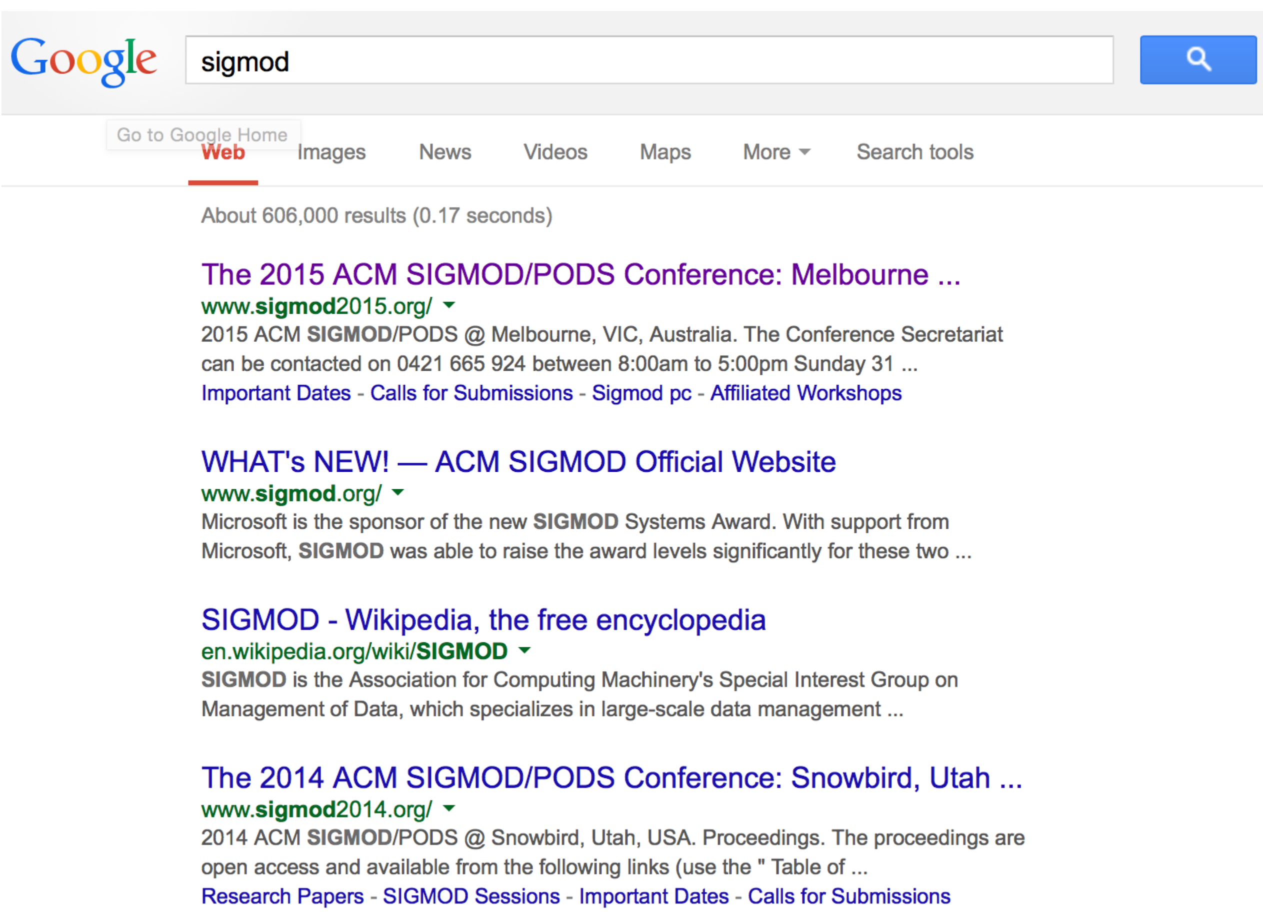
- Sub-linear space
  - Fast update and query time
- Answer queries approximately
- Linear transformation of the data frequencies

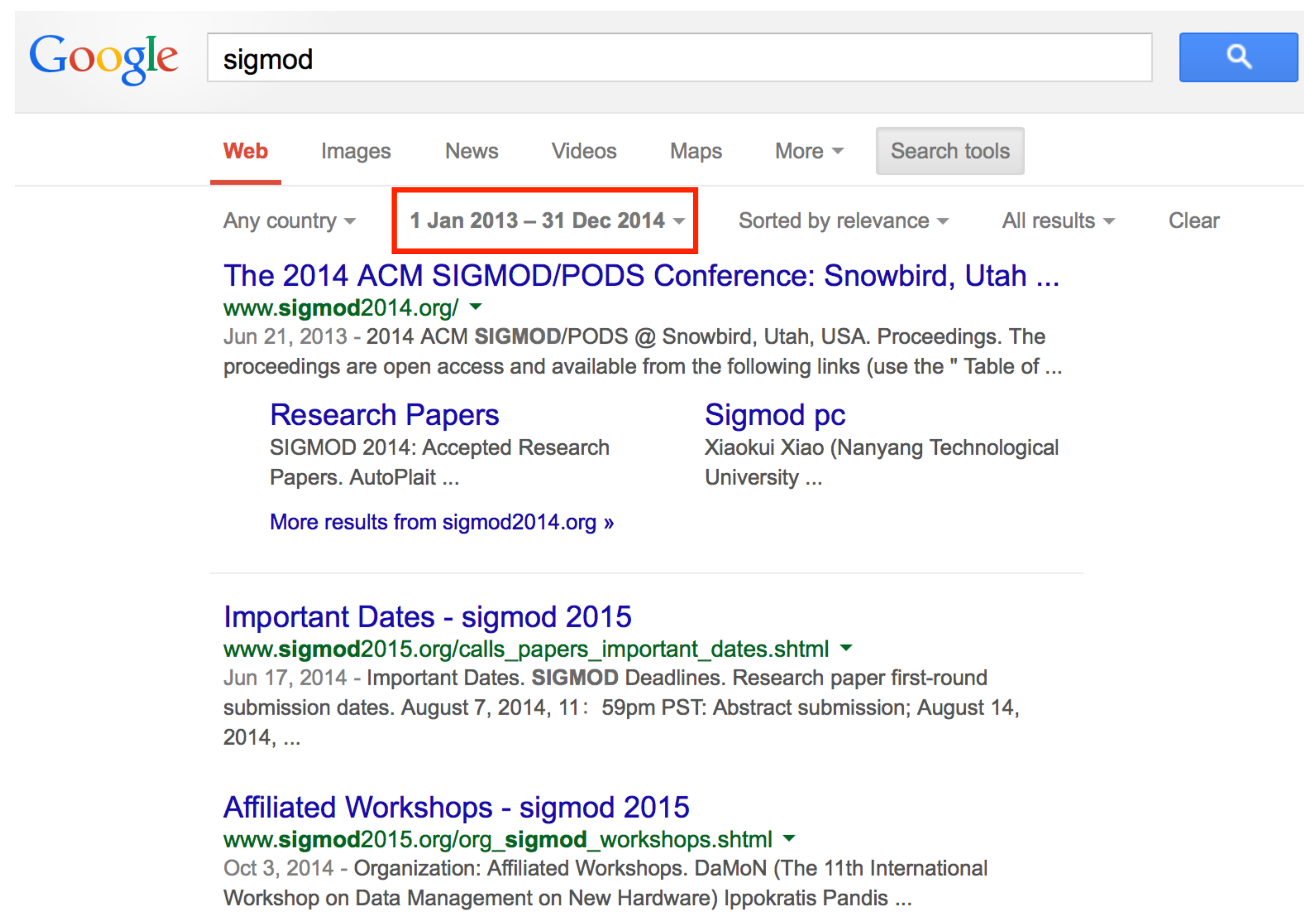
Sketches

- Sub-linear space
  - Fast update and query time
- Answer queries approximately
- Linear transformation of the data frequencies
- Ephemeral
  - Answer queries on current version of data stream

Query Back in Time

- The ability to query on historical data is necessary for analyzing trends&change pattern of data





Persistent Database/ Data Structure

- Answer queries on the past version of the database

Persistent Database/ Data Structure

- Answer queries on the past version of the database
- General technique to make data structure persistent [Driscoll et al. 1989], Multi-version B-tree [Becker et al. 1996, , Brodal et al. 2012], Time-Split B-tree [Lomet and Salzberg 1989]

Persistent Database/ Data Structure

- Answer queries on the past version of the database
- General technique to make data structure persistent [Driscoll et al. 1989], Multi-version B-tree [Becker et al. 1996, , Brodal et al. 2012], Time-Split B-tree [Lomet and Salzberg 1989]
- Microsoft Immortal DB [Lomet et. al. 2005], SNAP [Shrira and Xu 2005], Ganymed [Plattner et. al. 2006], Skippy [Shaull et. al. 2008] and LIVE[Sarma et. al. 2010]

Persistent Database/ Data Structure

- Answer queries on the past version of the database
- General technique to make data structure persistent [Driscoll et al. 1989], Multi-version B-tree [Becker et al. 1996, , Brodal et al. 2012], Time-Split B-tree [Lomet and Salzberg 1989]
- Microsoft Immortal DB [Lomet et. al. 2005], SNAP [Shrira and Xu 2005], Ganymed [Plattner et. al. 2006], Skippy [Shaull et. al. 2008] and LIVE[Sarma et. al. 2010]
- Space linear in # of updates
  - Large storage
  - Storage on disk (not in streaming setting)

Persistent Database

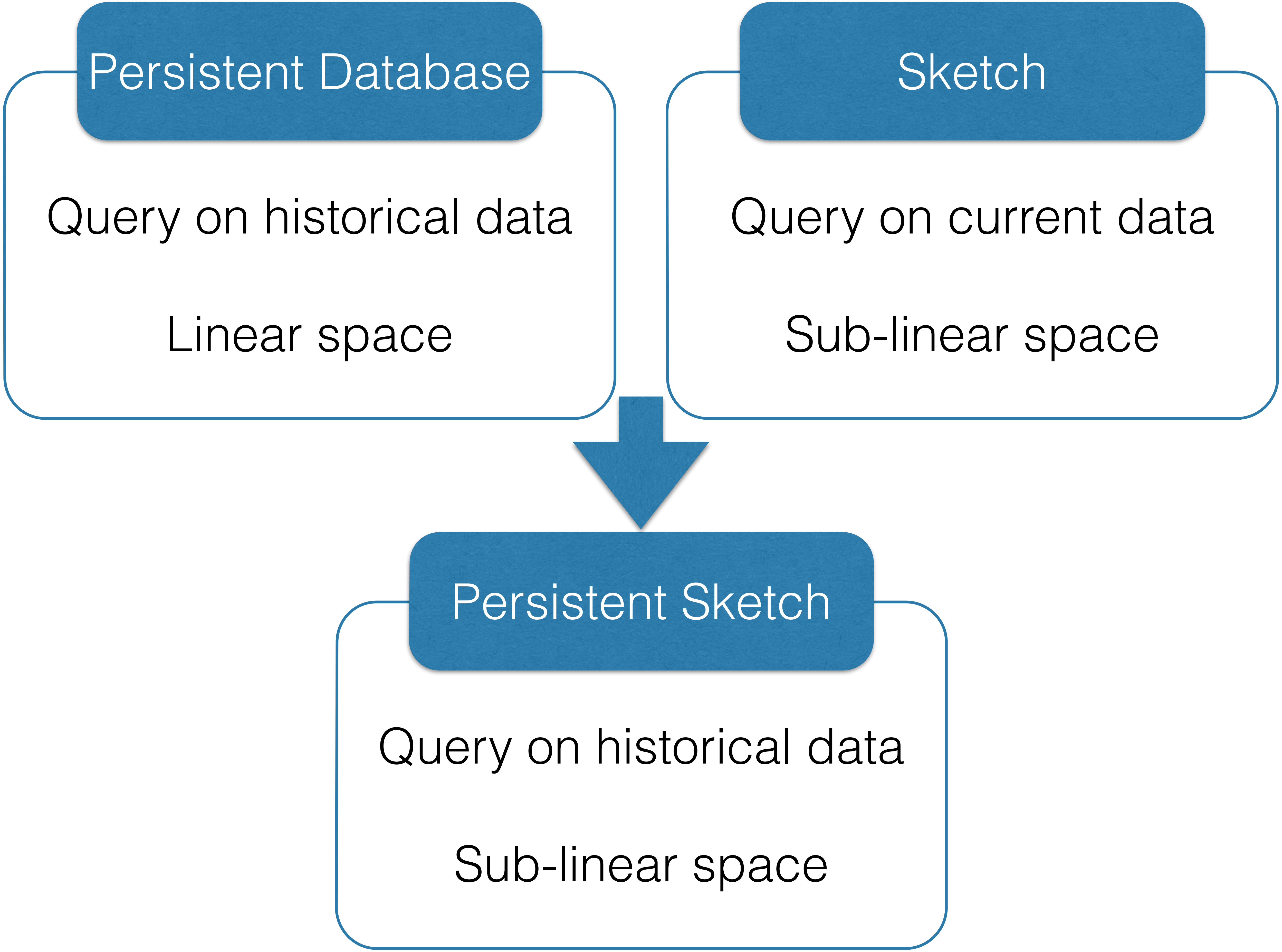
Query on historical data

Linear space

Sketch

Query on current data

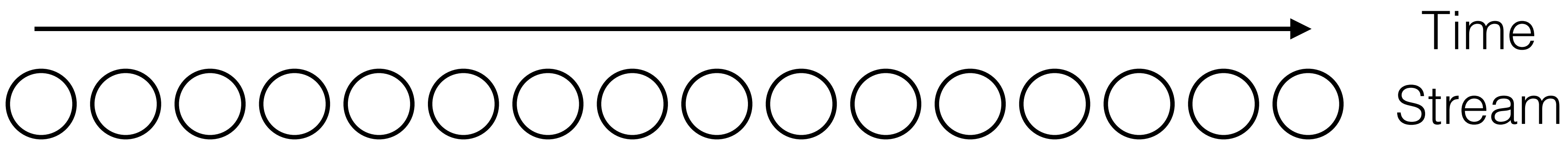
Sub-linear space



Persistent Database
Query on historical data
Linear space
Sketch
Query on current data
Sub-linear space
Persistent Sketch
Query on historical data
Sub-linear space

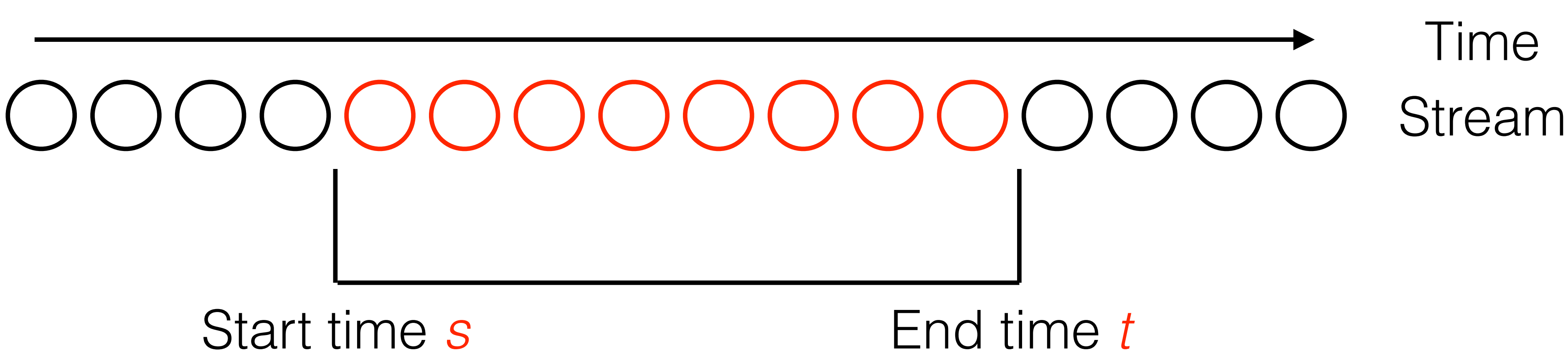
Persistent Sketch

- Historical window query



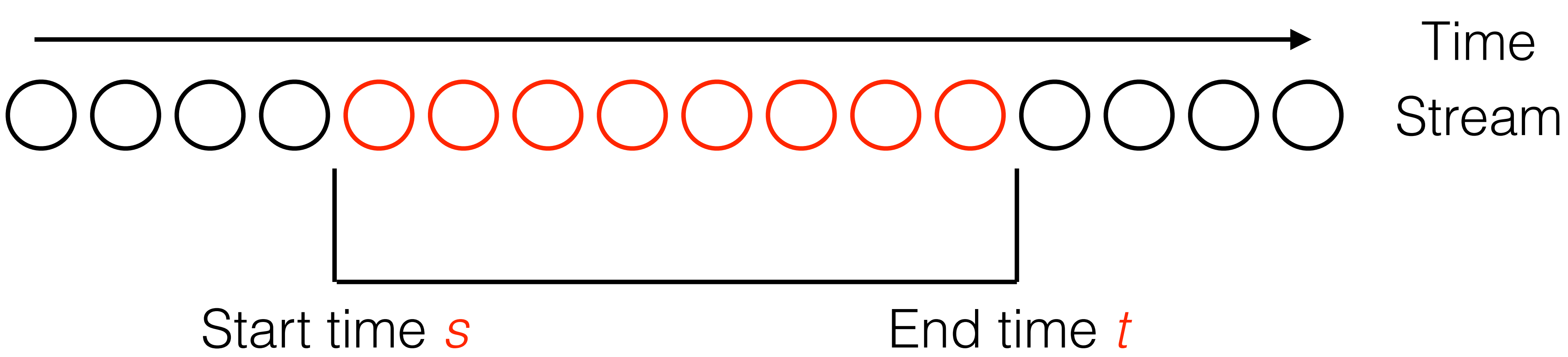
Persistent Sketch

- Historical window query



Persistent Sketch

- Historical window query



- Given a time interval (*s*, *t*], return a sketch for substream f(*s*, *t*)
- What is the top-k/frequency moment/join size of the stream between *s* and *t*?

High Level Ideas
&
Our Results

Count-Min Sketch

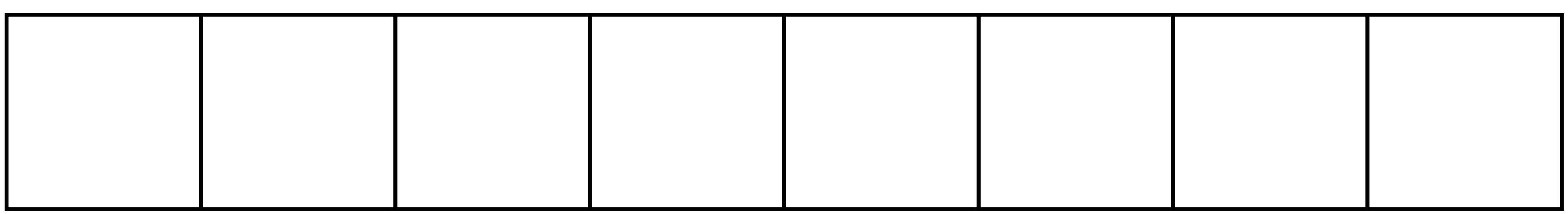
[Cormode and Muthukrishnan 2005]

- Given an error parameter ε
- Choose a hash function h: $[n] \to [2/\varepsilon]$ and build a hash table of size $2/\varepsilon$

Count-Min Sketch

[Cormode and Muthukrishnan 2005]

- Given an error parameter ε
- Choose a hash function h: $[n] \rightarrow [2/\varepsilon]$ and build a hash table of size $2/\varepsilon$

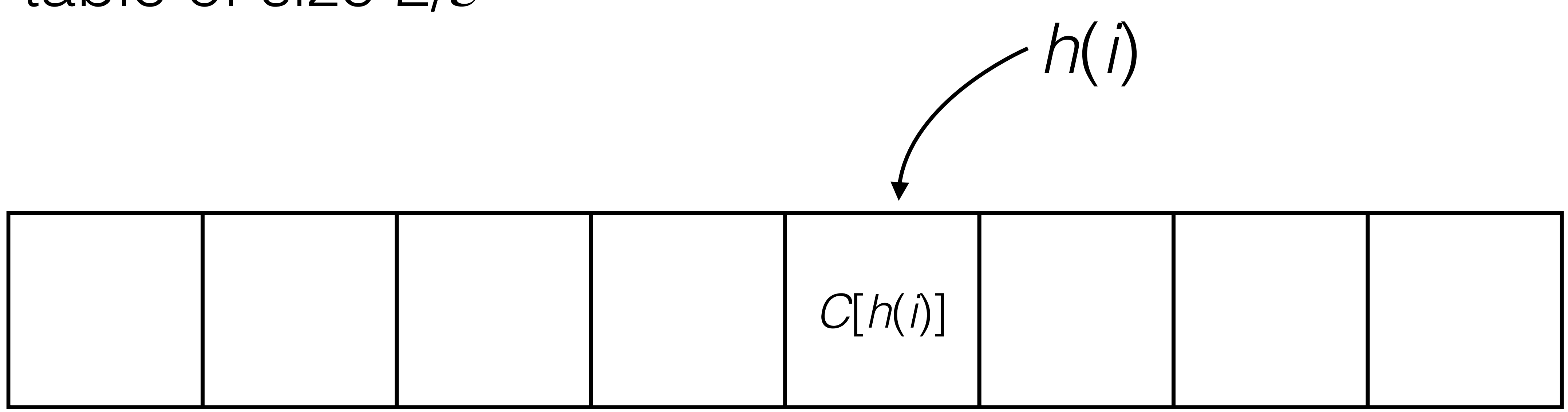


Count-Min Sketch

[Cormode and Muthukrishnan 2005]

- Given an error parameter ε

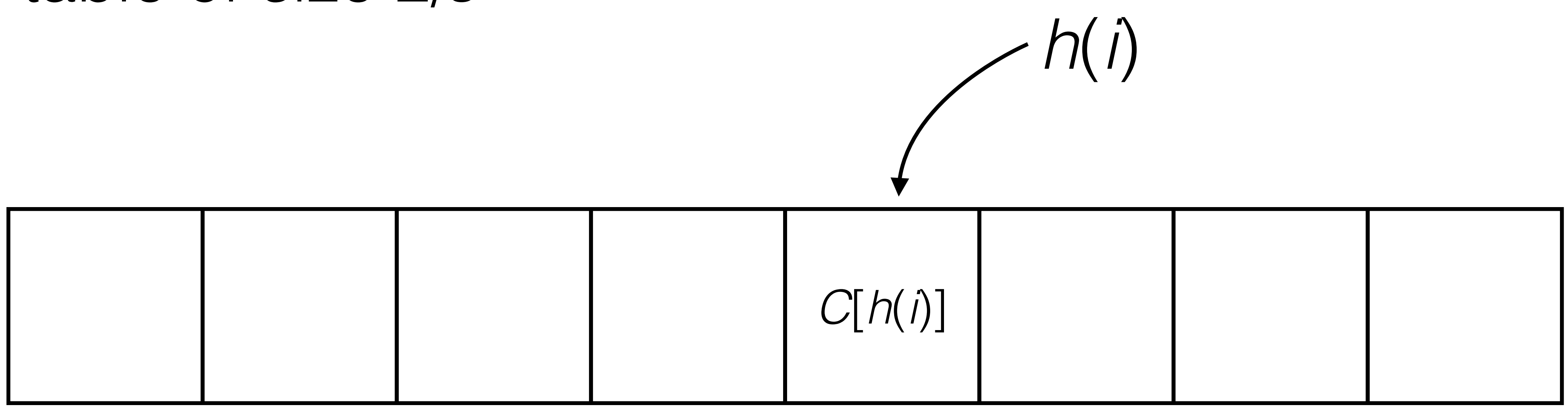
- Choose a hash function h: $[n] \rightarrow [2/\varepsilon]$ and build a hash table of size $2/\varepsilon$



Count-Min Sketch

[Cormode and Muthukrishnan 2005]

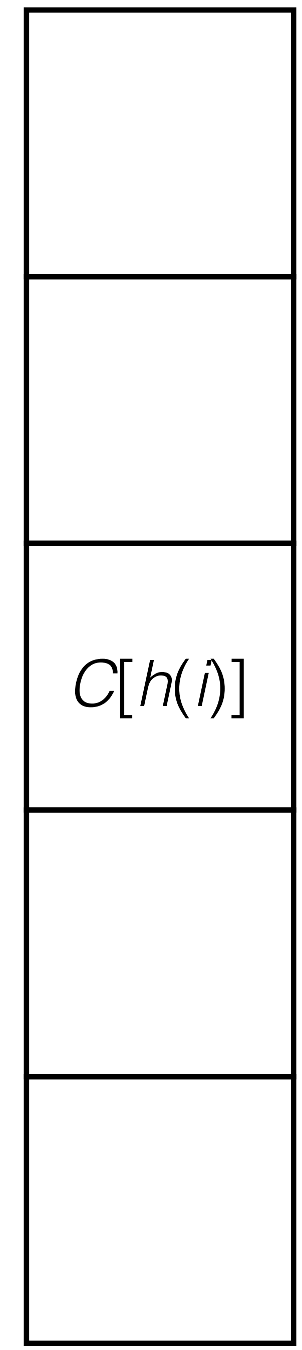
- Given an error parameter ε
- Choose a hash function h: $[n] \rightarrow [2/\varepsilon]$ and build a hash table of size $2/\varepsilon$



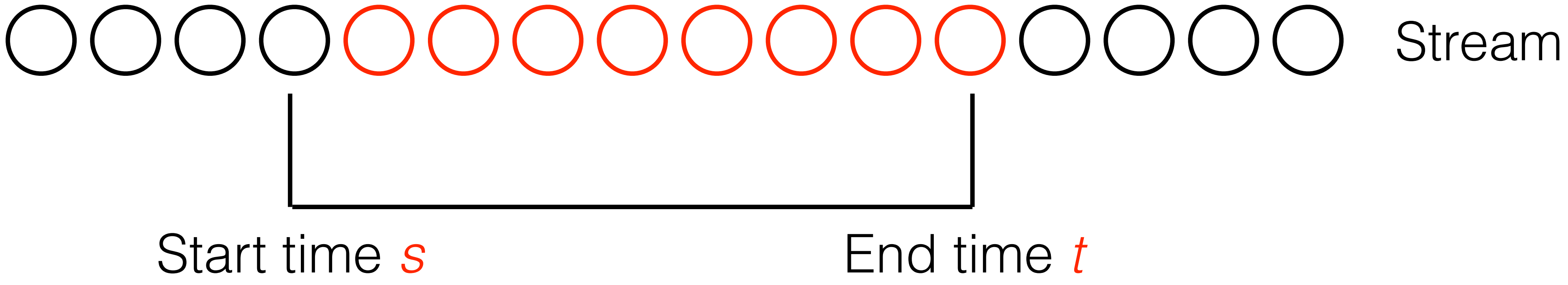
$$C[h(i)] = C[h(i)] + 1$$

Linear Transformation

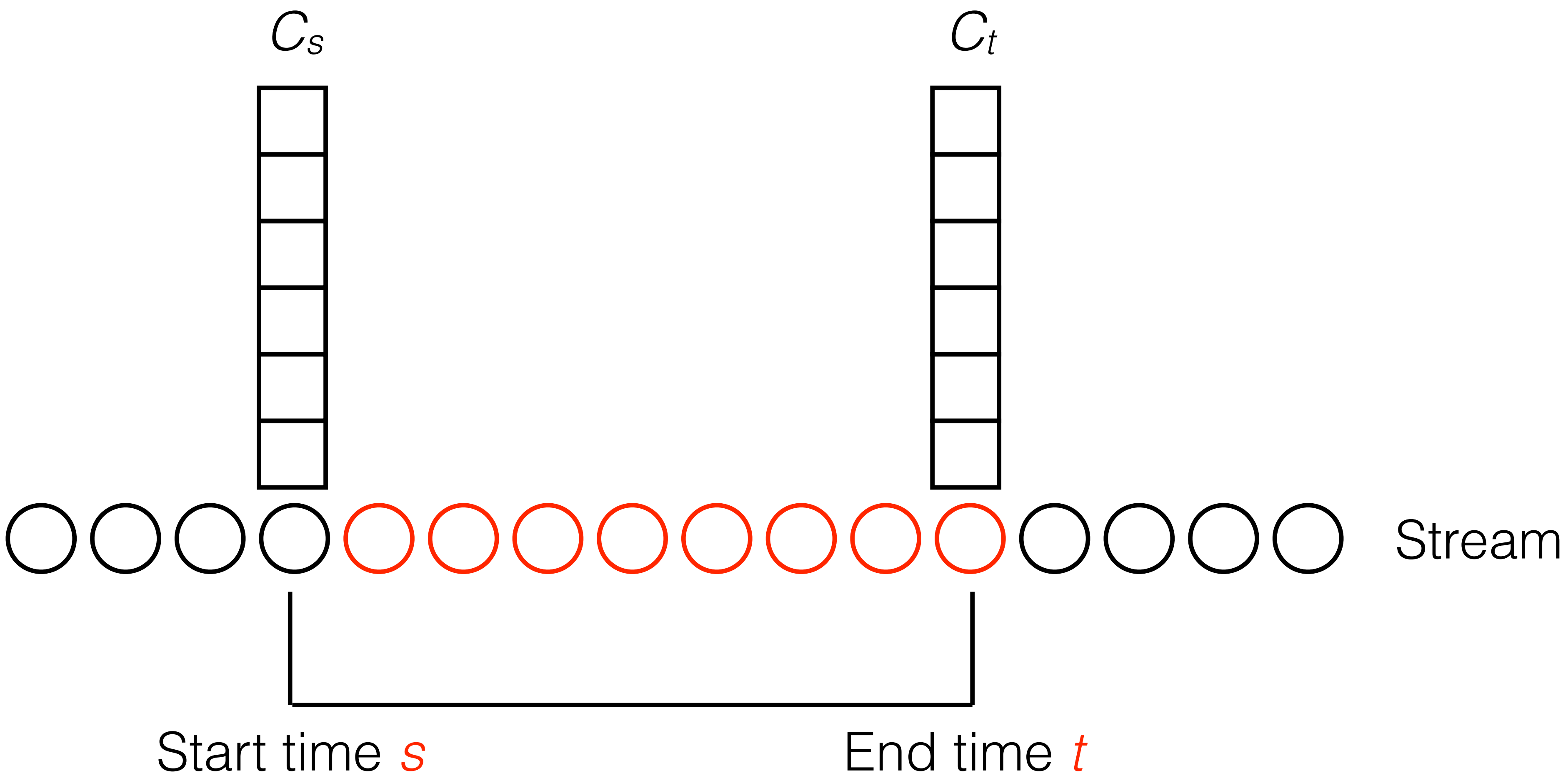
$$\begin{array}{cc} & \qquad\qquad\qquad i \\ h(i) & \begin{pmatrix} 0, & 1, & 0, \ldots, & 0, \ldots, & 0, & 0, \\ & & \ldots & & & \\ 0, & 0, & 0, \ldots, & 1, \ldots, & 0, & 0, \\ & & \ldots & & & \\ 0, & 0, & 0, \ldots, & 0, \ldots, & 1, & 0, \end{pmatrix} \end{array} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \ldots \\ f_i \\ \ldots \\ f_N \end{pmatrix} =$$



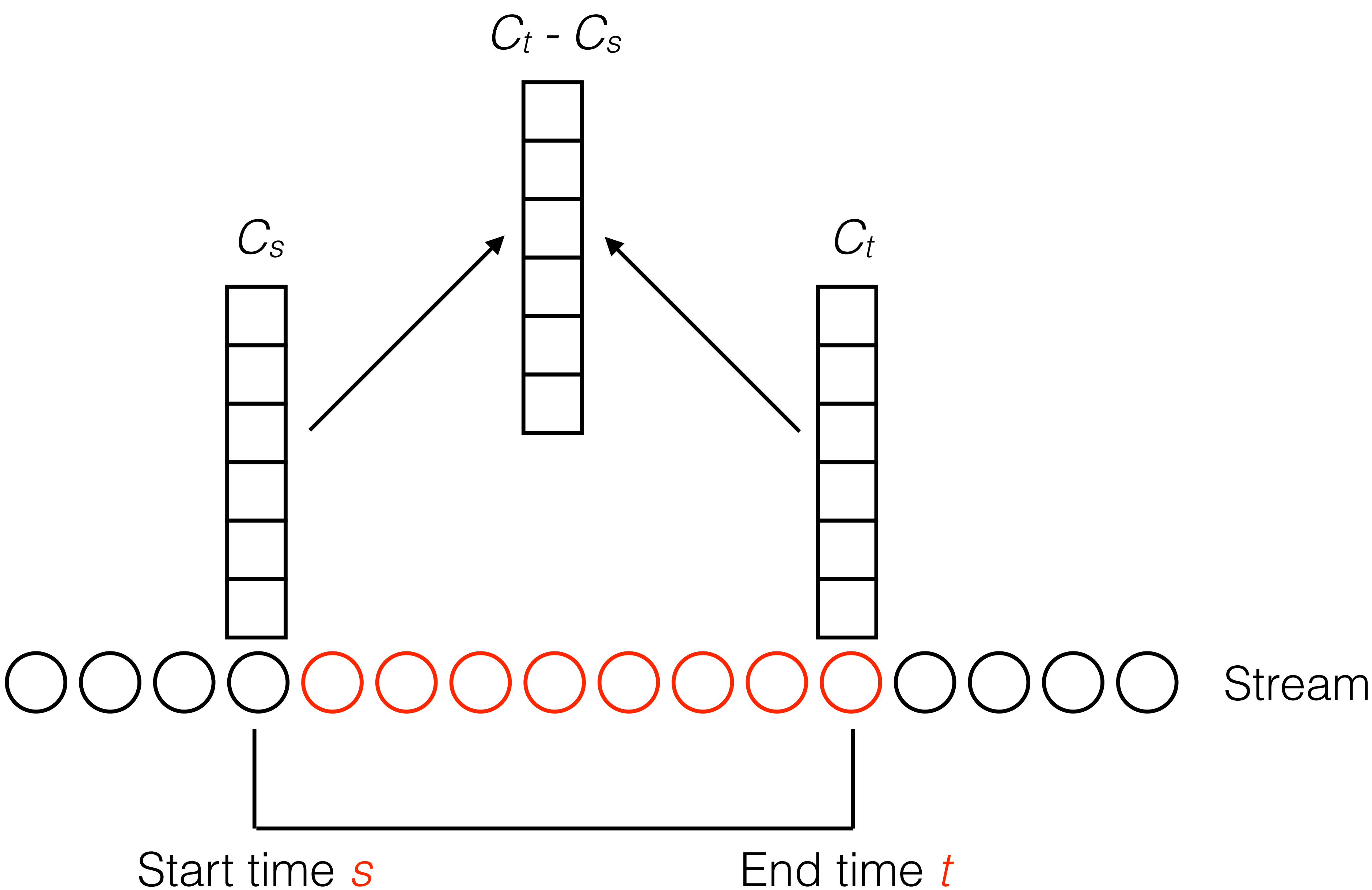
Linear Transformation



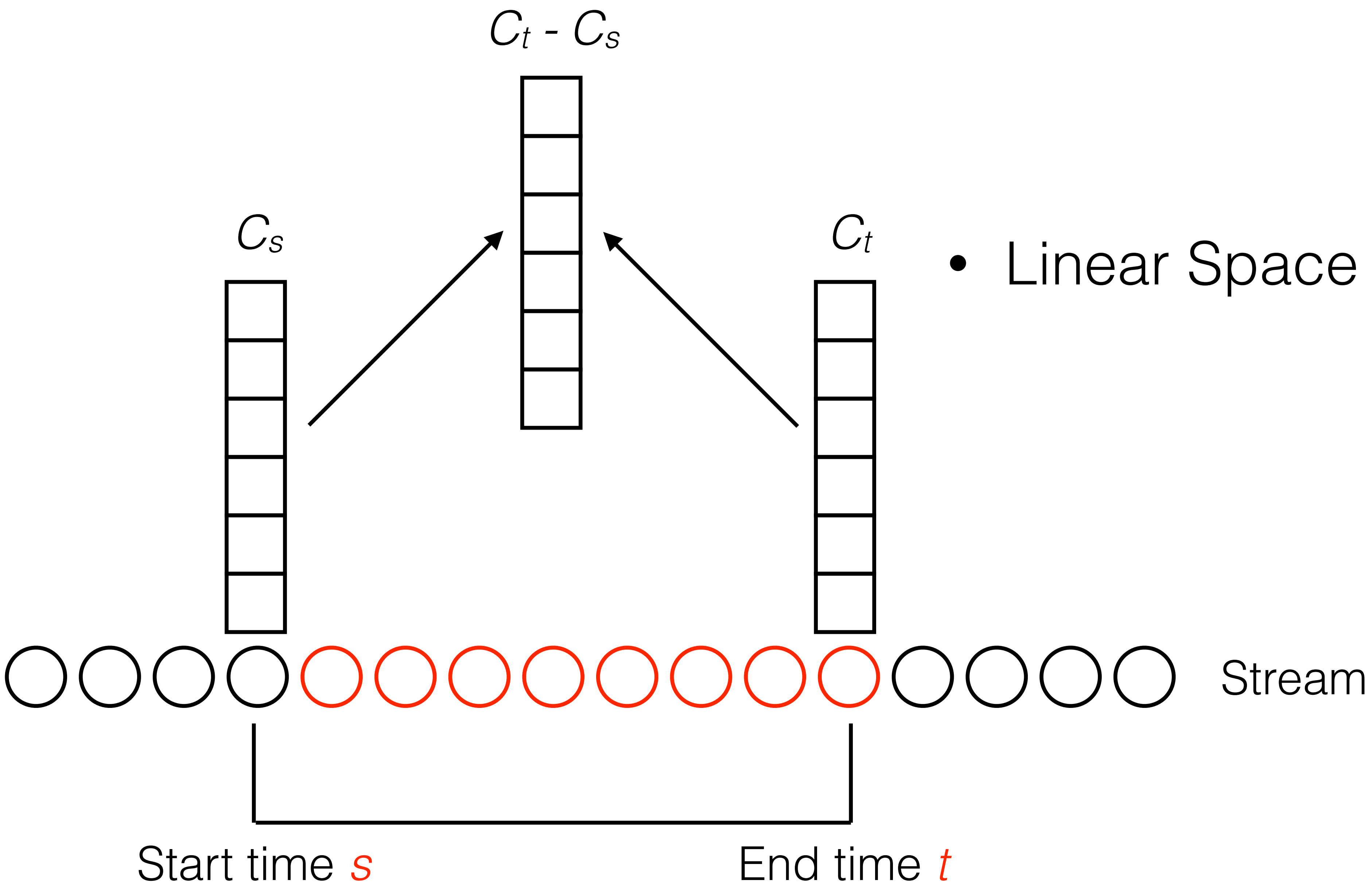
Linear Transformation



Linear Transformation

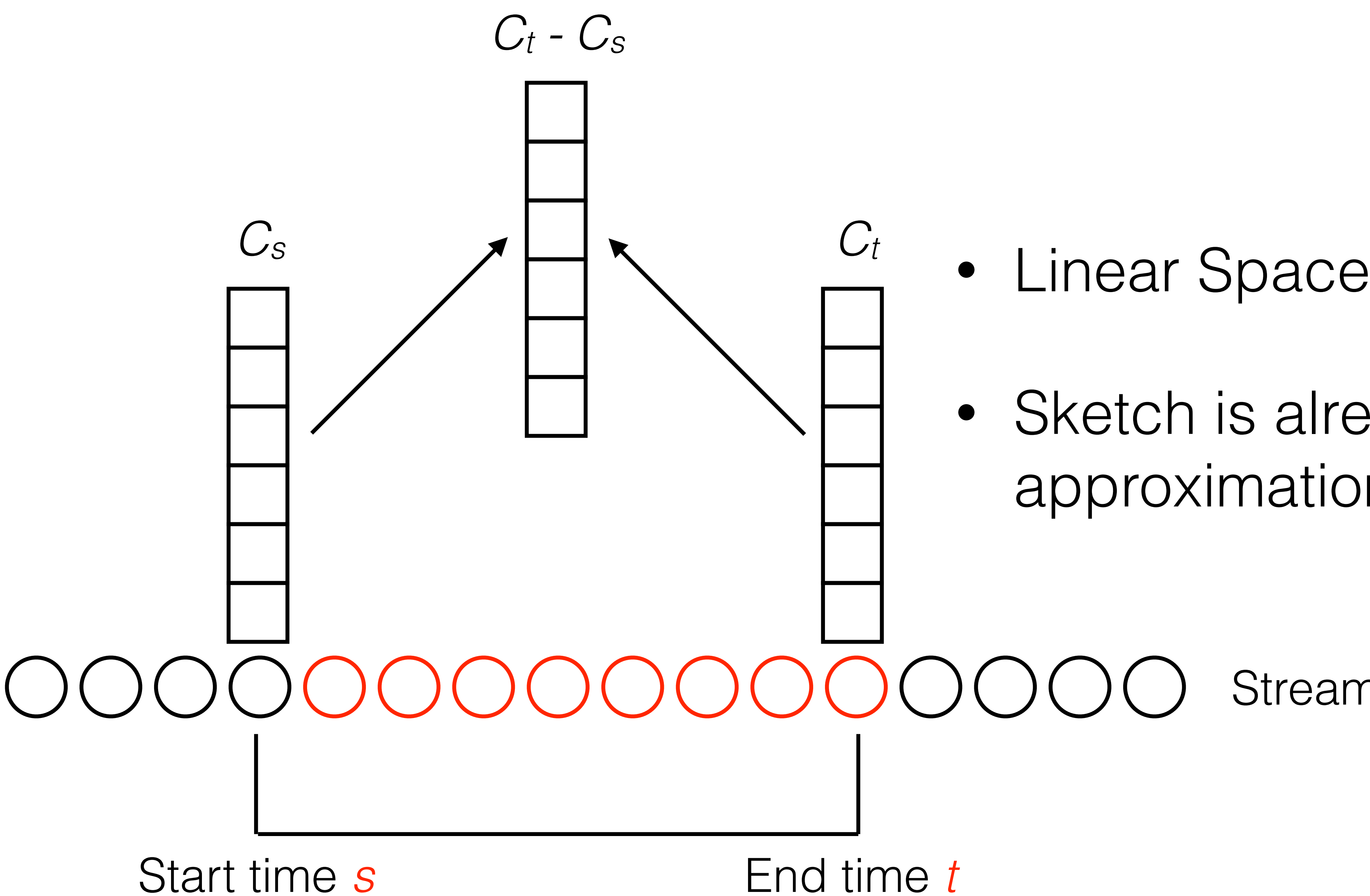


Linear Transformation



- Linear Space

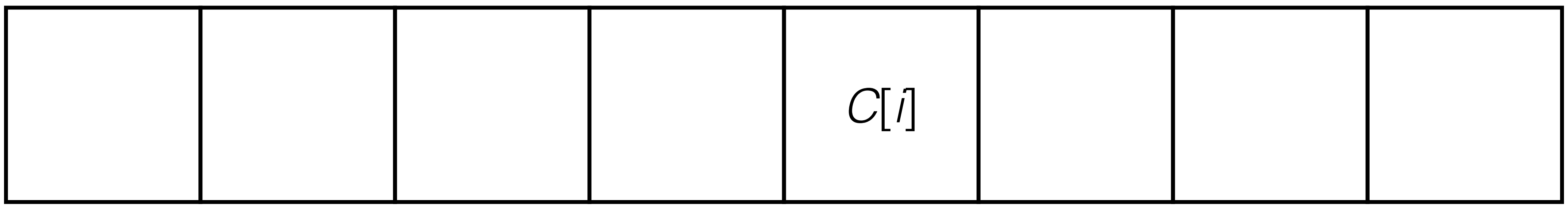
Linear Transformation



- Linear Space
- Sketch is already an approximation

Baseline Solution

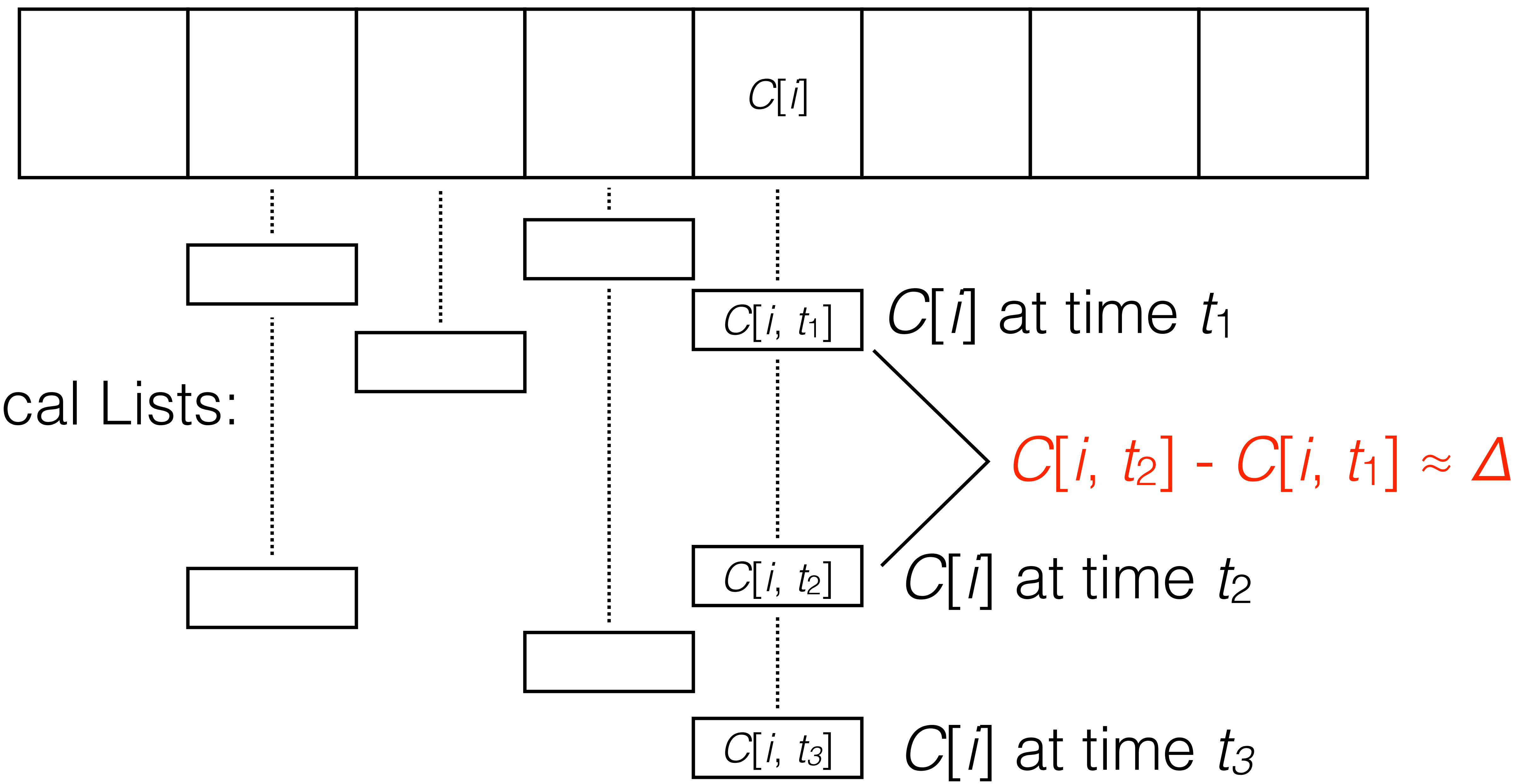
Ephemeral sketch:



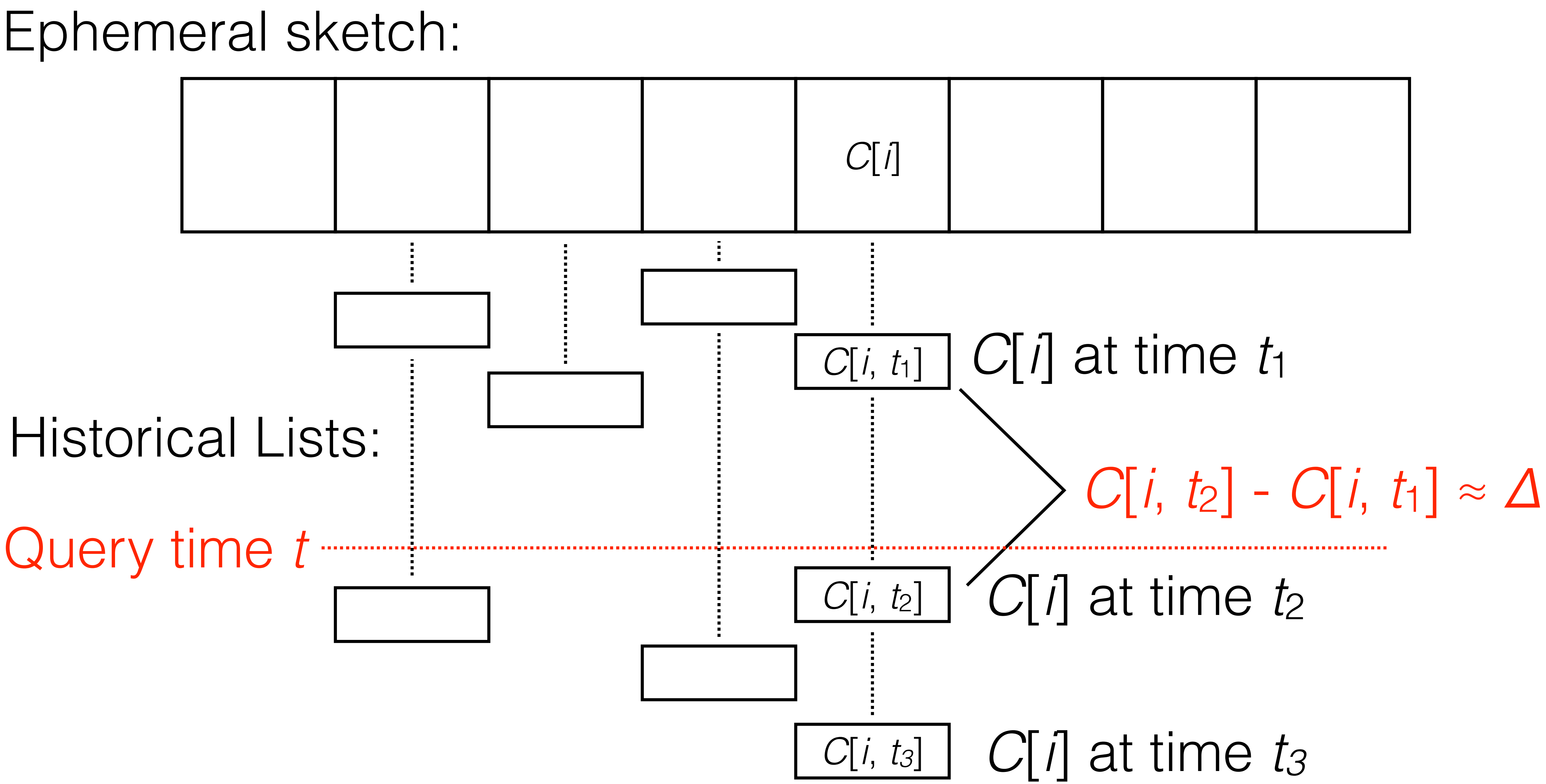
Baseline Solution

Ephemeral sketch:

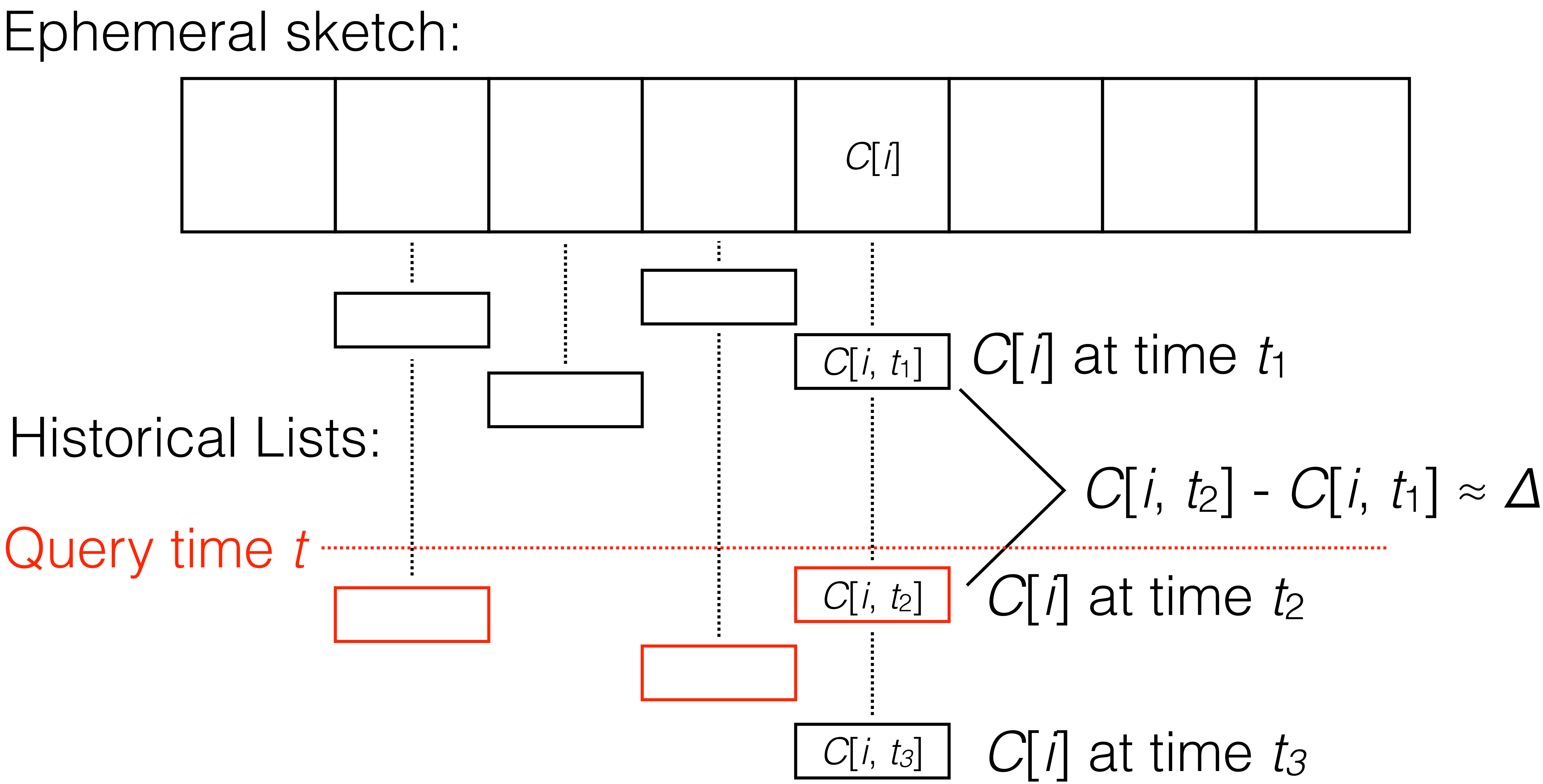
Historical Lists:



Baseline Solution



Baseline Solution



Baseline Solution

- Historical window point/heavy hitters query:
  - What is frequency of “/images/space.gif” between day 34 and day 37
  - What are the mostly requested URLs between day 34 and day 37”

Baseline Solution

- Historical window point/heavy hitters query:
  - What is frequency of “/images/space.gif” between day 34 and day 37
  - What are the mostly requested URLs between day 34 and day 37”
- Error: $\varepsilon||\mathbf{f}(s,t)||_1$ (ephemeral error) + Δ (persistent error)

Baseline Solution

- Historical window point/heavy hitters query:
  - What is frequency of "/images/space.gif" between day 34 and day 37
  - What are the [illegible] d URLs between day 34 and day 3[illegible]

Size of the stream between s and t

- Error: $\varepsilon||\mathbf{f}(s,t)||_1$ (ephemeral error) + Δ (persistent error)

Baseline Solution

- Historical window point/heavy hitters query:
  - What is frequency of "/images/space.gif" between day 34 and day 37
  - What are the m… …d URLs between day 34 and day 3…

Size of the stream between s and t

- Error: $\varepsilon\|\mathbf{f}(s,t)\|_1$ (ephemeral error) + Δ (persistent error)

- Space: proportional to $(1/\varepsilon + m/\Delta)$

Baseline Solution

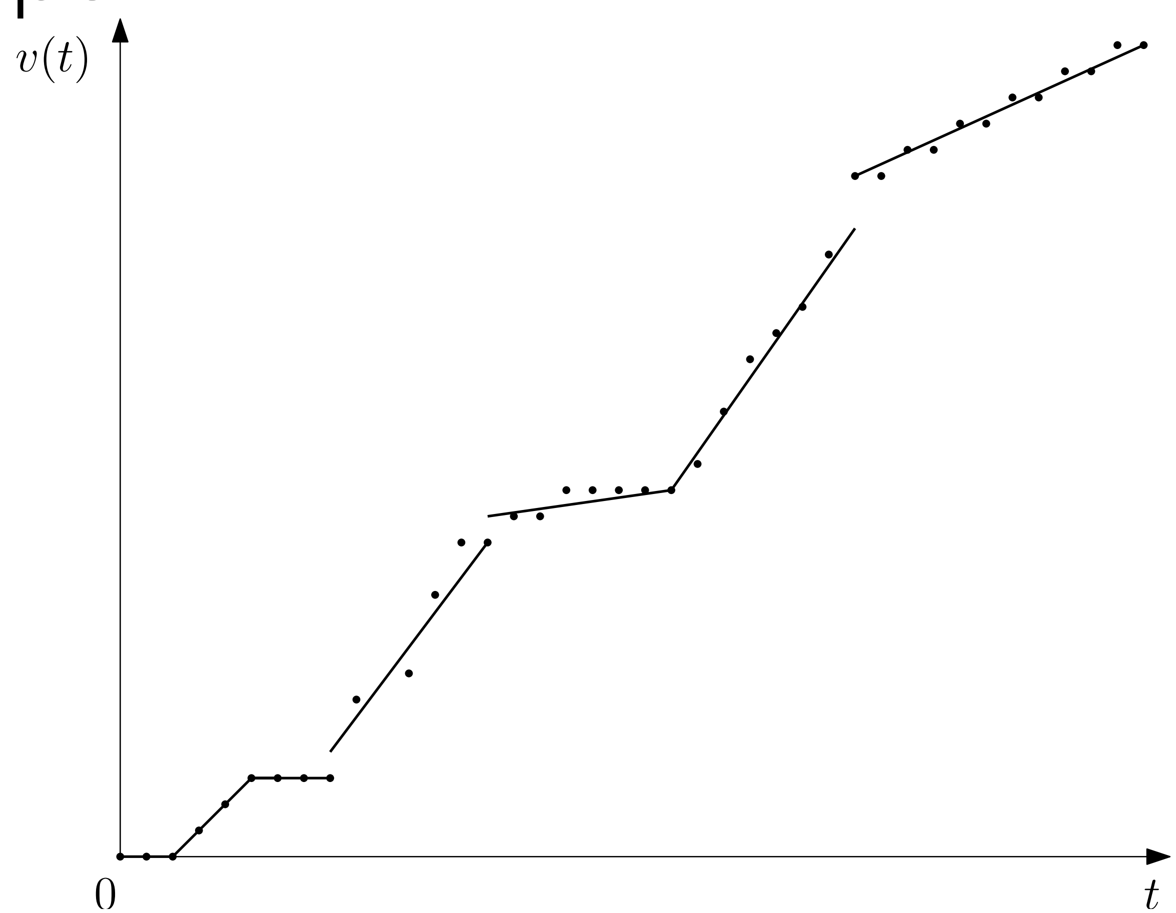
- Historical window point/heavy hitters query:
  - What is frequency of "/images/space.gif" between day 34 and day 37
  - What are the [illegible] URLs between day 34 and day 3[illegible]

Size of the stream between s and t

- Error: $\varepsilon\|\mathbf{f}(s,t)\|_1$ (ephemeral error) + Δ (persistent error)
- Space: proportional to $(1/\varepsilon + m/\Delta)$
- Cannot handle (self) join size queries

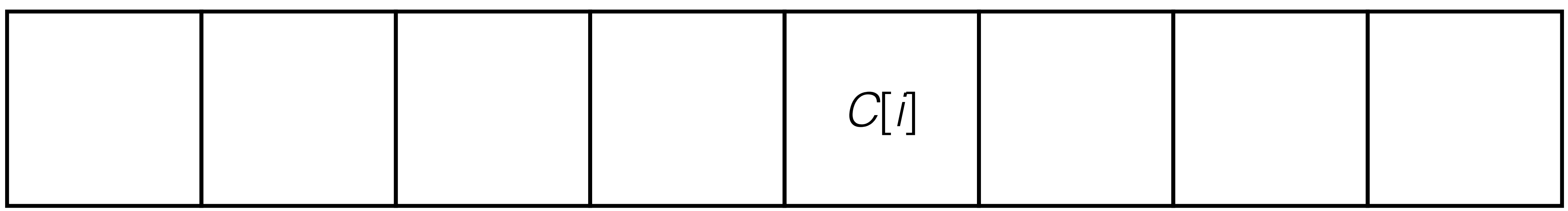
Piece-wise Linear Approximation

- Counter changes by at most 1 at each timestamp
- Each counter is a discrete function according to timestamps

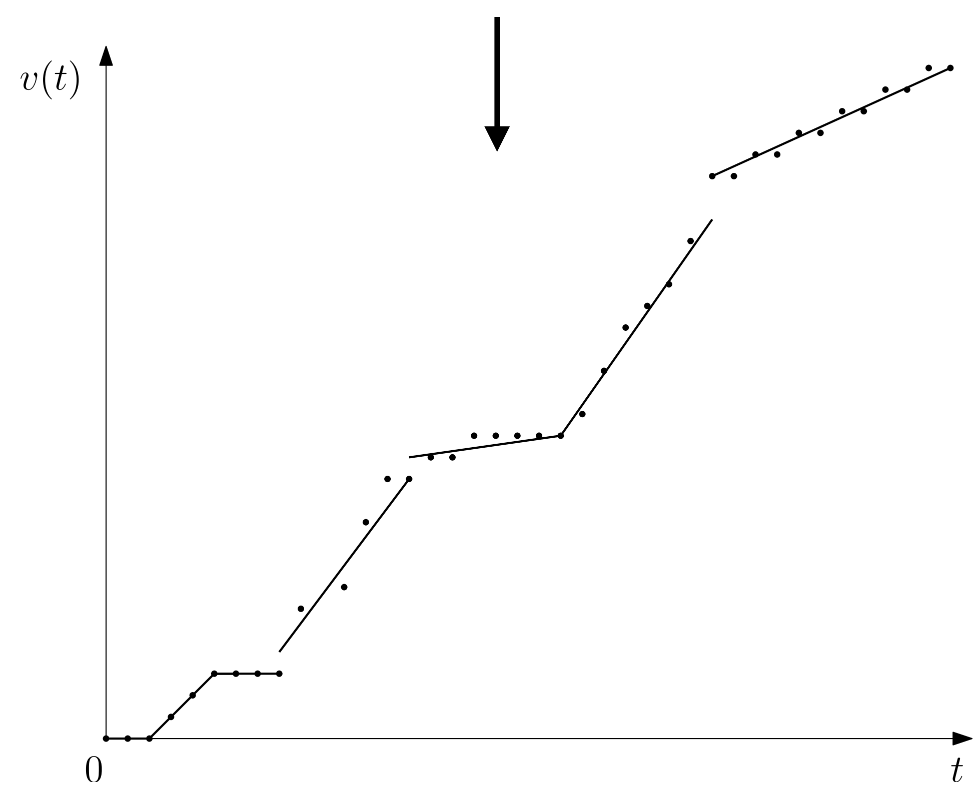


PLA-based Persistent Sketch

Ephemeral sketch:

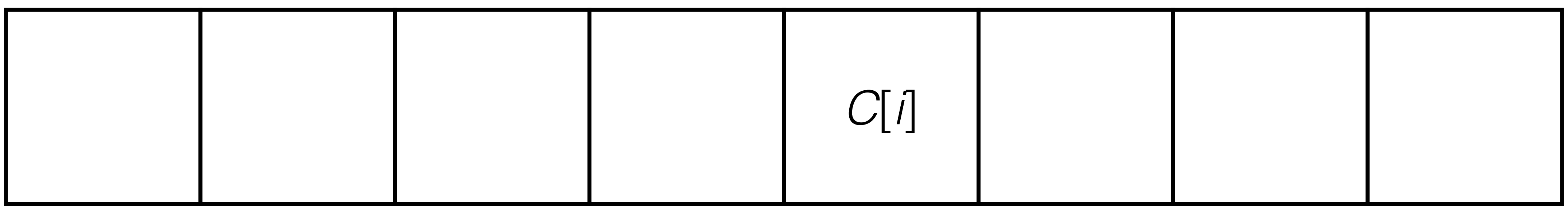


PLA generator:

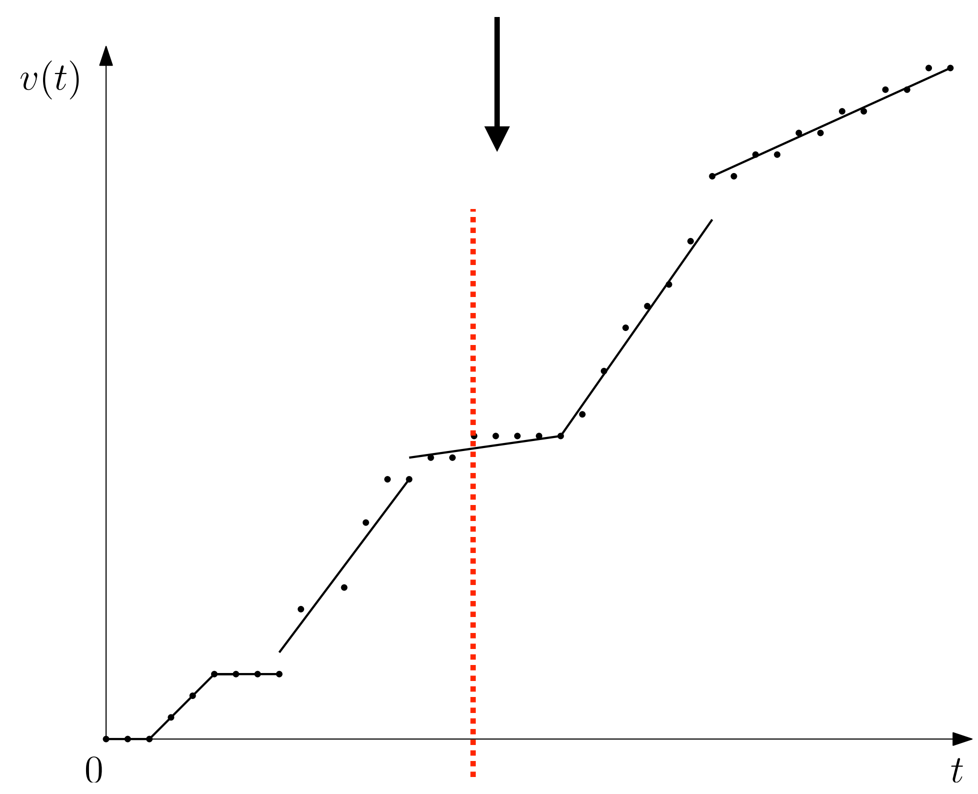


PLA-based Persistent Sketch

Ephemeral sketch:



PLA generator:



Query time t

PLA-based Persistent Sketch

- Historical window point/heavy hitters query:
  - What is frequency of “/images/space.gif” between day 34 and day 37
  - What are the mostly requested URLs between day 34 and day 37”

- Error: $\varepsilon||\mathbf{f}(s,t)||_1$ (ephemeral error) + Δ (persistent error)

PLA-based Persistent Sketch

- Historical window point/heavy hitters query:
  - What is frequency of “/images/space.gif” between day 34 and day 37
  - What are the mostly requested URLs between day 34 and day 37”
- Error: $\varepsilon\|\mathbf{f}(s,t)\|_1$ (ephemeral error) + Δ (persistent error)
- Space: proportional to $(1/\varepsilon + m/\Delta^2)$ in random stream model

Estimating Join Size

- Estimating (self) join size in an ephemeral sketch:

$$\Sigma_i\ C[i]^2$$

Estimating Join Size

- Estimating (self) join size in an ephemeral sketch:

$$\Sigma_i C[i]^2$$

- Estimating (self) join size in a persistent sketch:

Estimating Join Size

- Estimating (self) join size in an ephemeral sketch:

$$\Sigma_i \, C[i]^2$$

- Estimating (self) join size in a persistent sketch:

$$\Sigma_i \, (C[i] + \text{error of } \Delta)^2$$

Estimating Join Size

- Estimating (self) join size in an ephemeral sketch:

$$\Sigma_i \; C[i]^2$$

- Estimating (self) join size in a persistent sketch:

$$\Sigma_i \; (C[i] + \text{error of } \Delta)^2$$

- Bias will amplify error significantly

Estimating Join Size

- Estimating (self) join size in an ephemeral sketch:

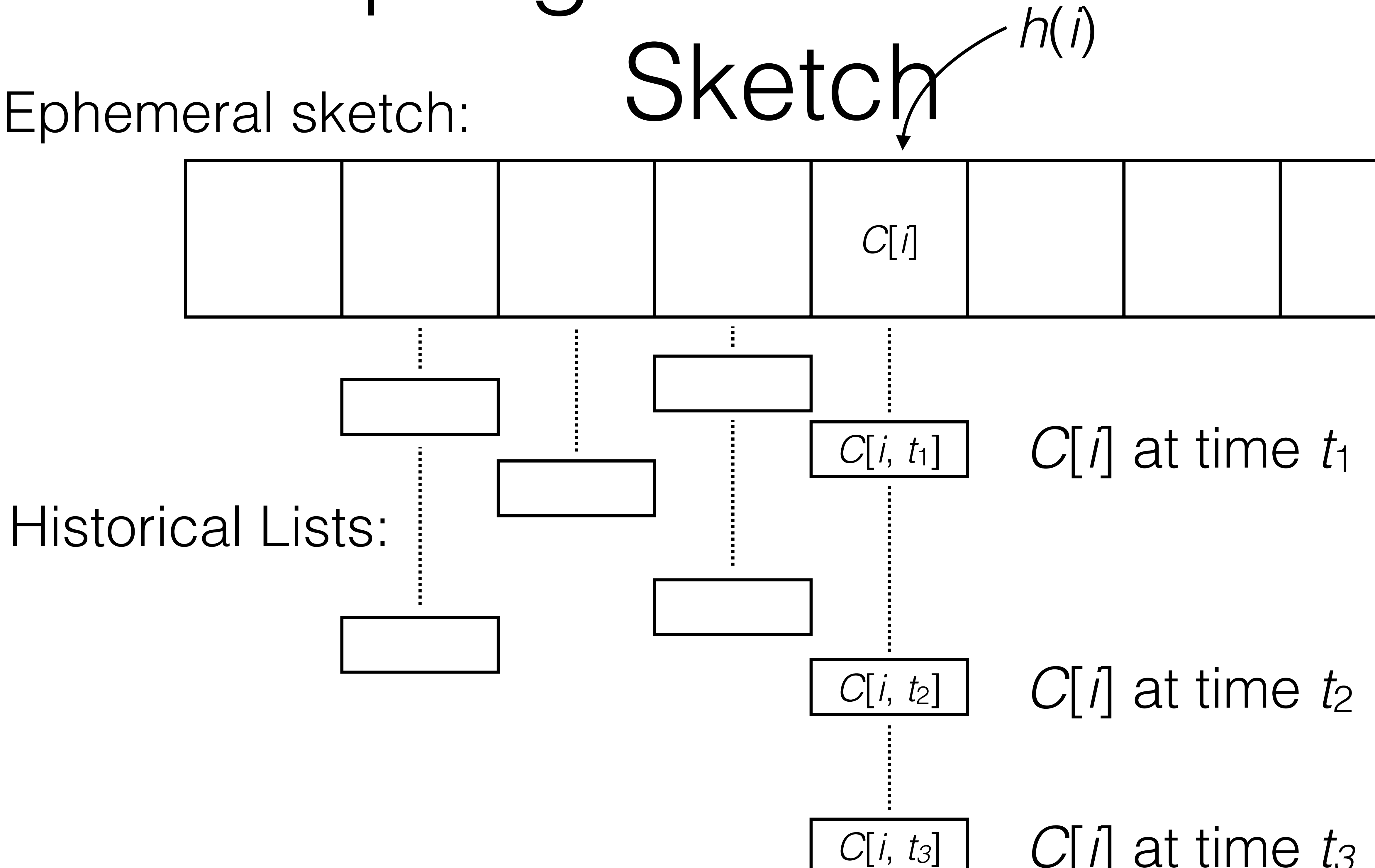
$$\Sigma_i \, C[i]^2$$

- Estimating (self) join size in a persistent sketch:

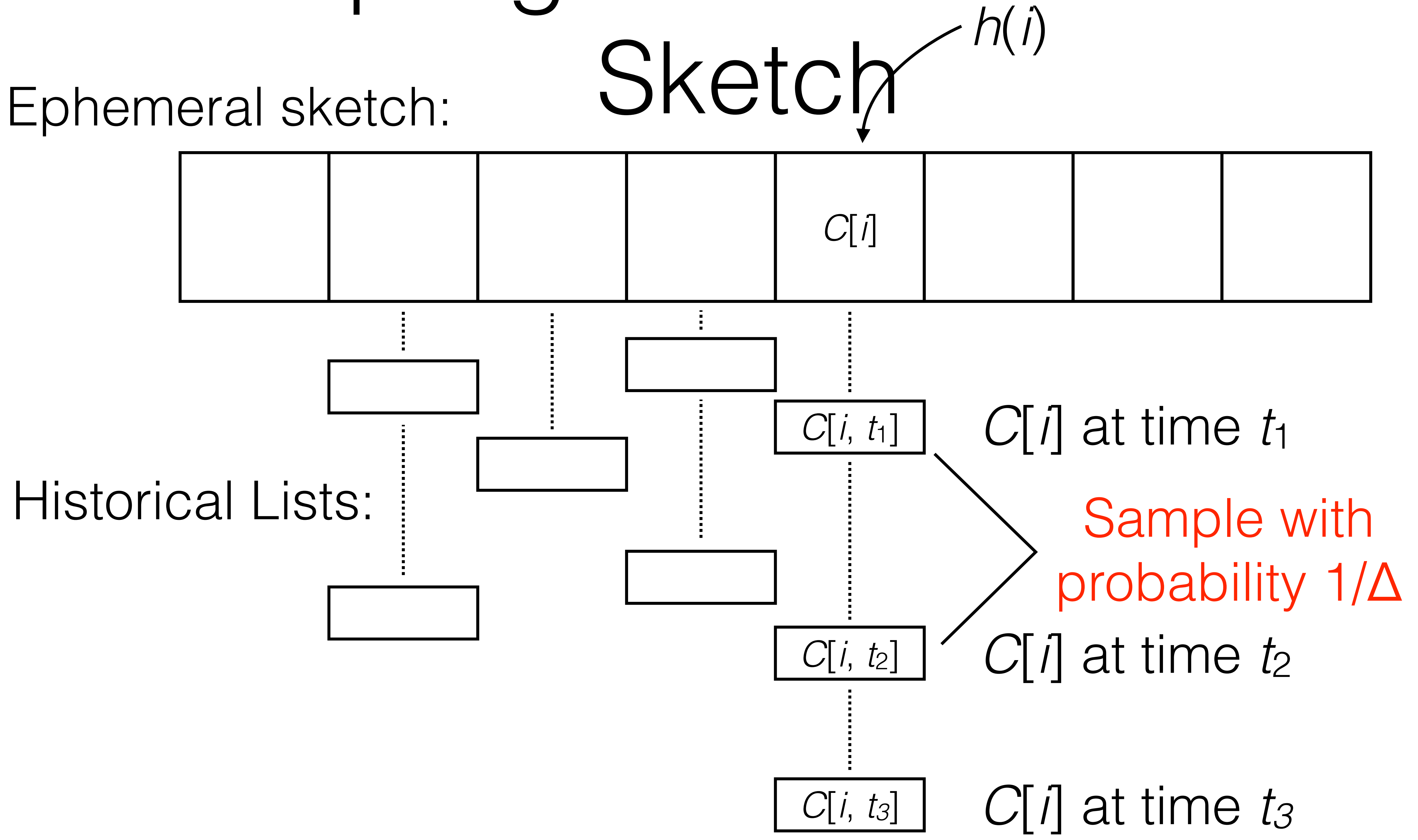
$$\Sigma_i \, (C[i] + \text{error of } \Delta)^2$$

- Bias will amplify error significantly
- Need unbiased estimator of the counter

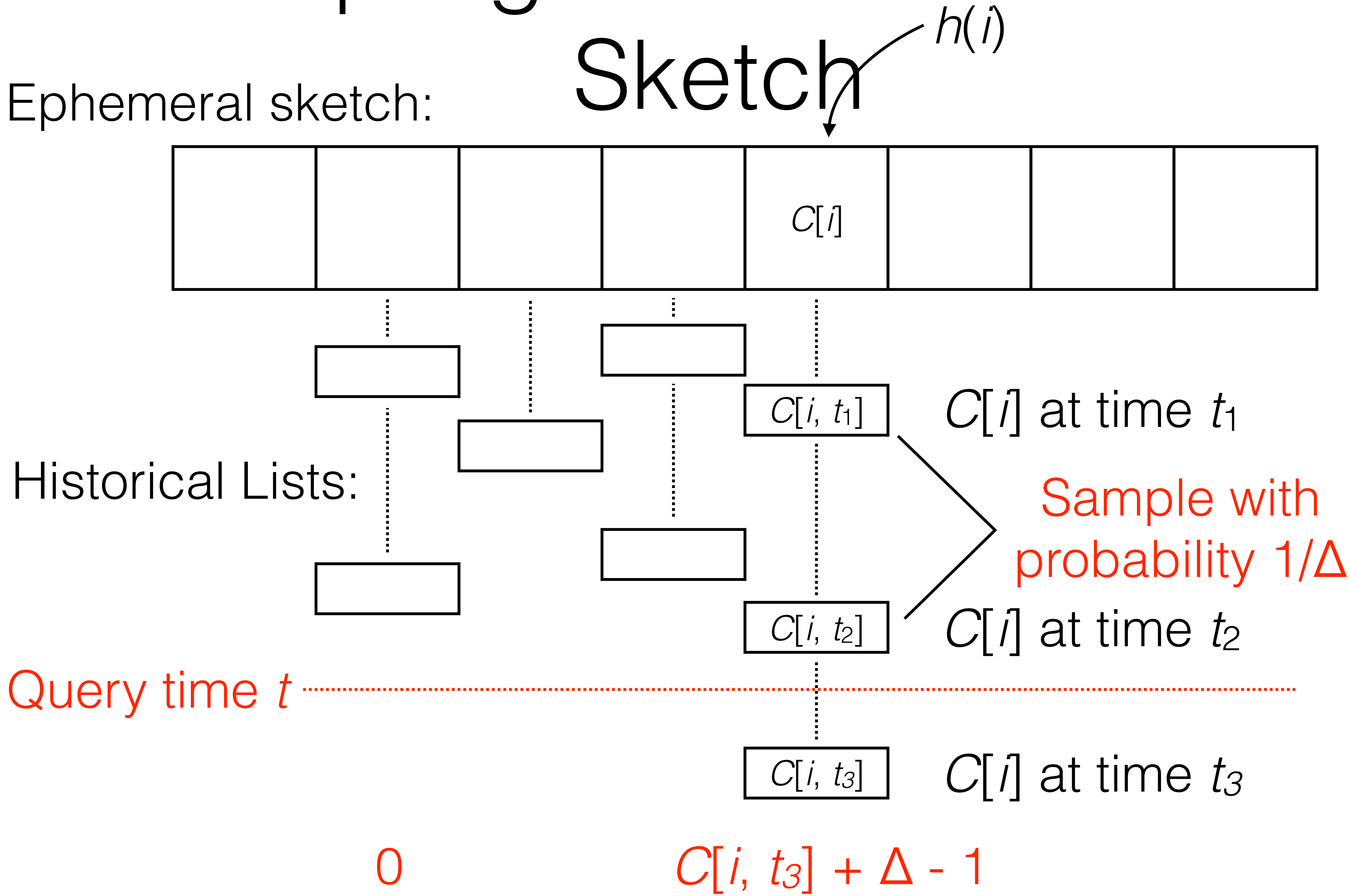
Sampling Based Persistent Sketch



Sampling Based Persistent Sketch



Sampling Based Persistent Sketch



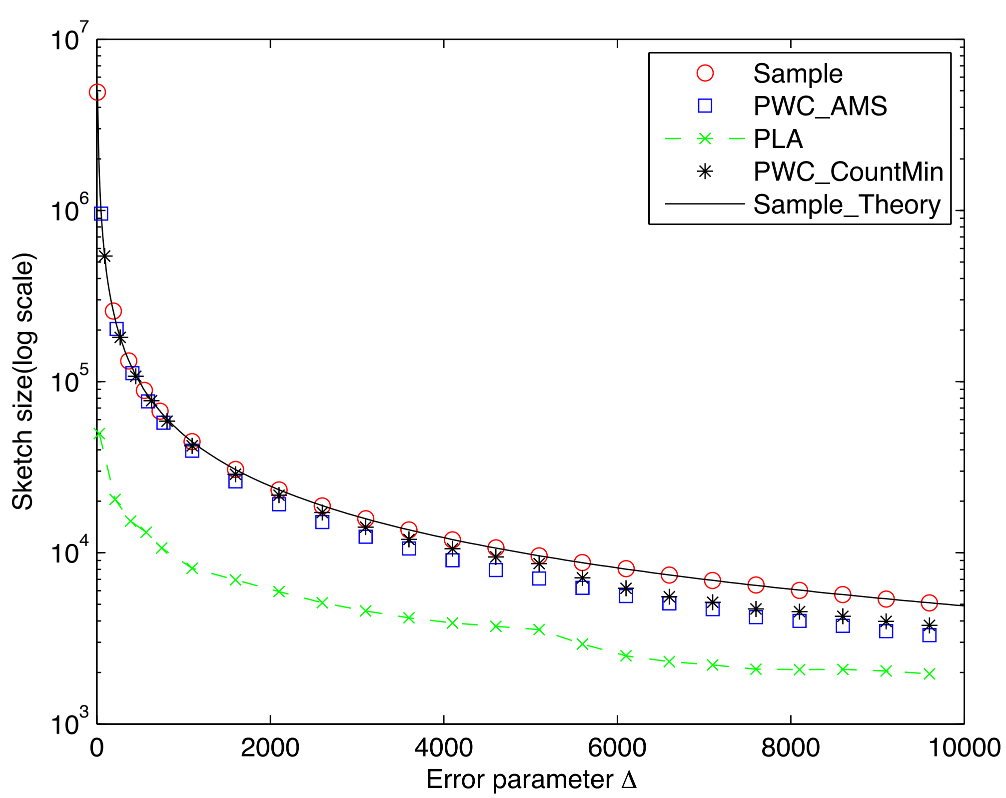
Sampling based AMS Sketch

- Unbiased Estimator
- Historical window join size query:
  - What is the join size of stream 1 and stream 2 between day 34 and day 37
- Error: $\varepsilon\sqrt{\left(\|\mathbf{f}_{s,t}\|_2^2 + (\frac{\Delta_{\mathbf{f}}}{\varepsilon})^2\right)\left(\|\mathbf{g}_{s,t}\|_2^2 + (\frac{\Delta_{\mathbf{g}}}{\varepsilon})^2\right)}$
- Space: proportional to $(1/\varepsilon + m/\Delta)$

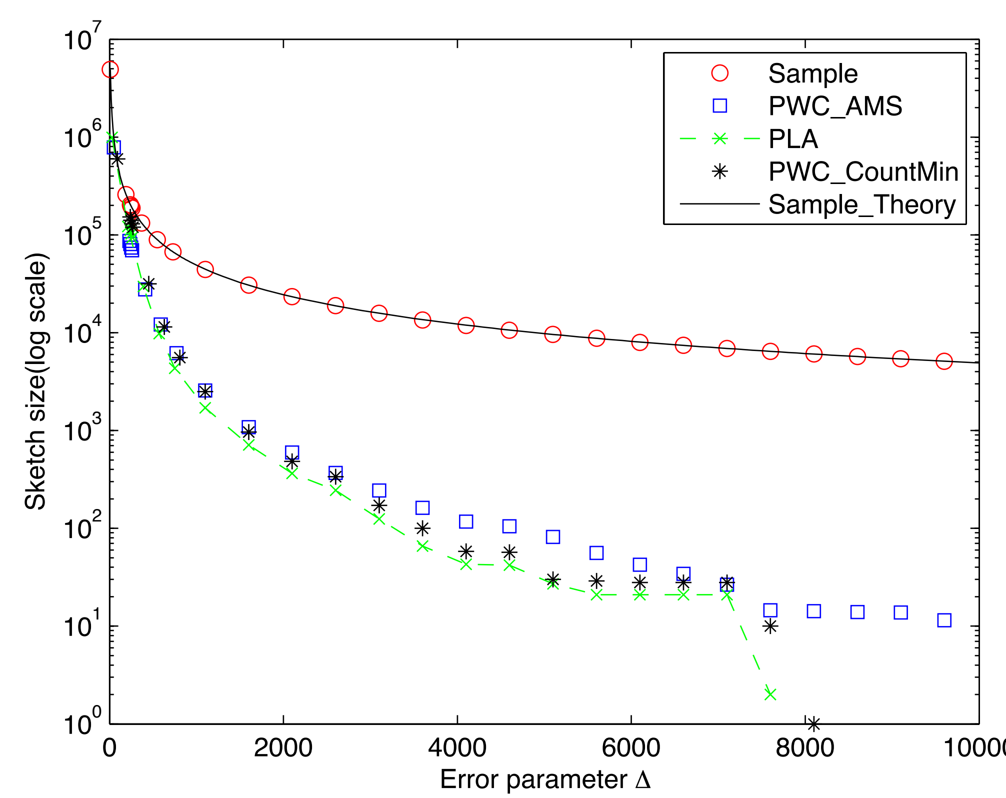
Experimental Study

- 7,000,000 requests from the 1998 World Cup web site access log
- Built sketches on two attributes

Requested URL



IP address of the request



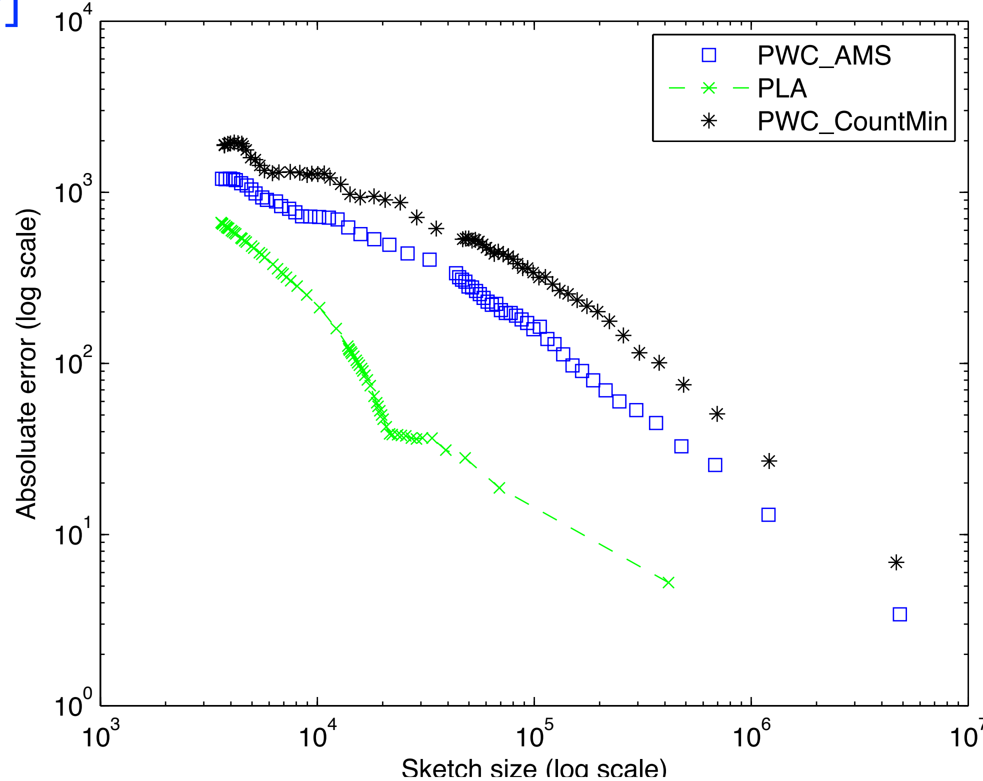
Experimental Study

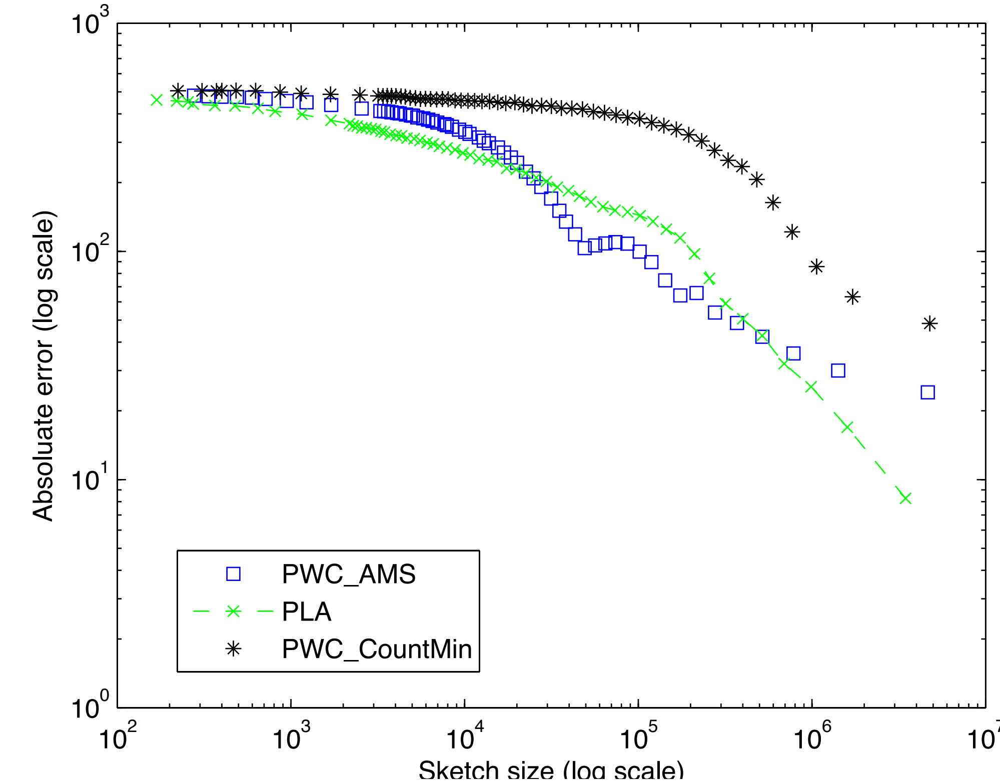
Query range
($0.2N$, $0.6N$]

Requested URL

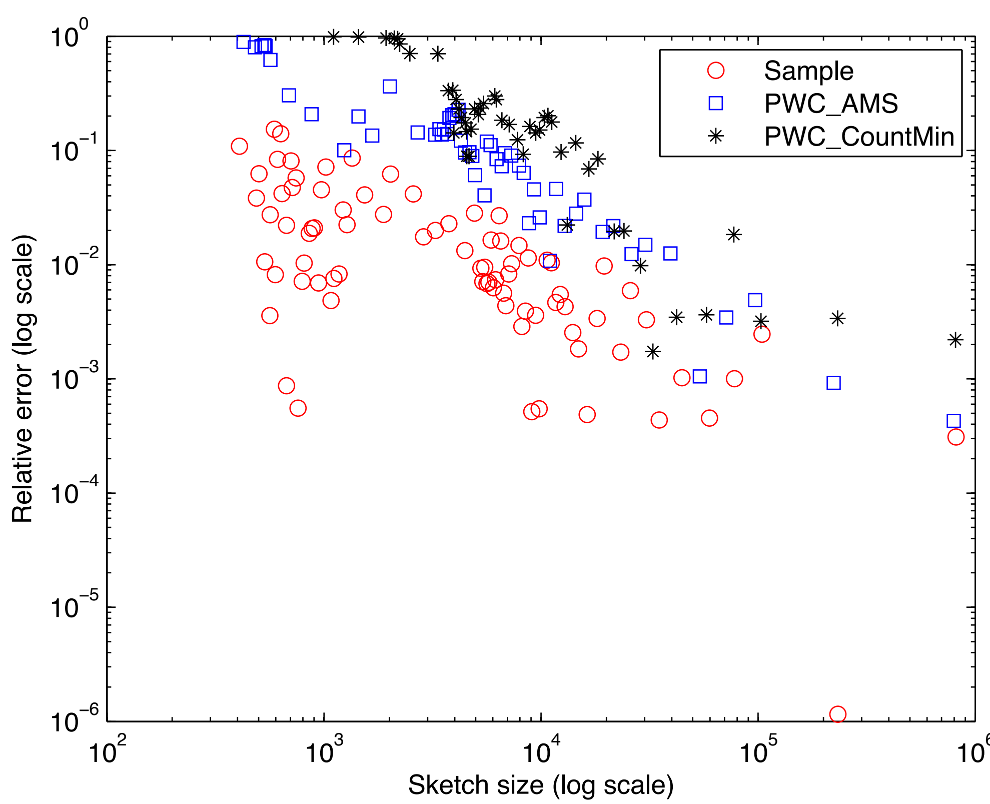
IP address of the request

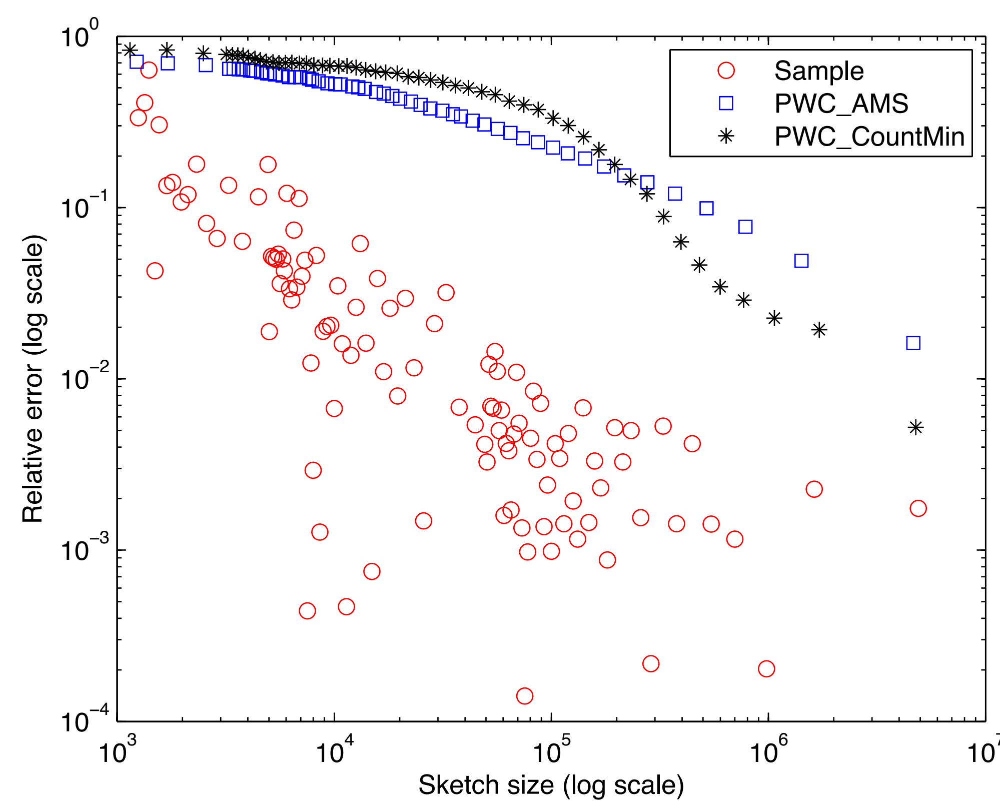
Point
Query





Self Join
Size
Query





Conclusion

- Persistent sketch
    - Query on historical data
    - Sub-linear space
- Support point/heavy hitters/join size queries
- Provable error and space bound
- Performs well in practice

Thanks!