

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

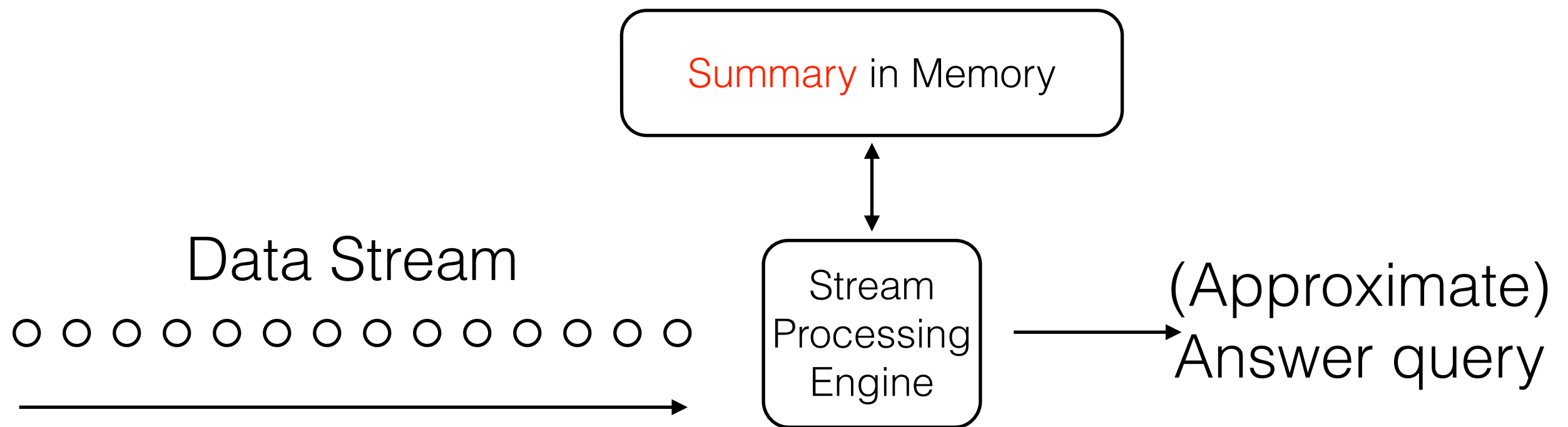
Renmin University of China

Ji-Rong Wen

Renmin University of China

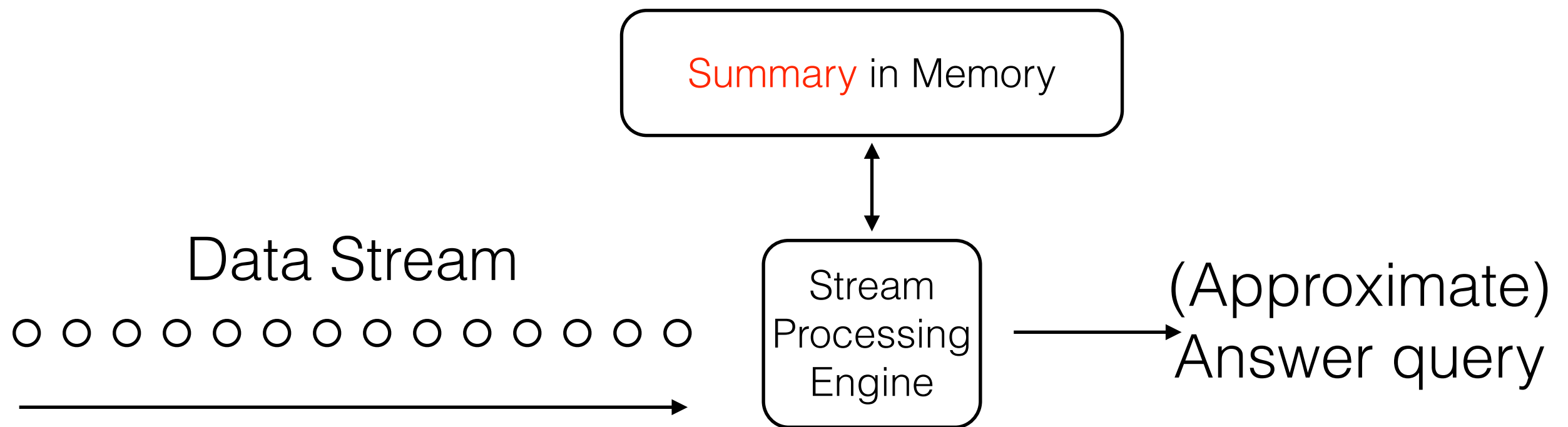
Streaming Algorithms

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



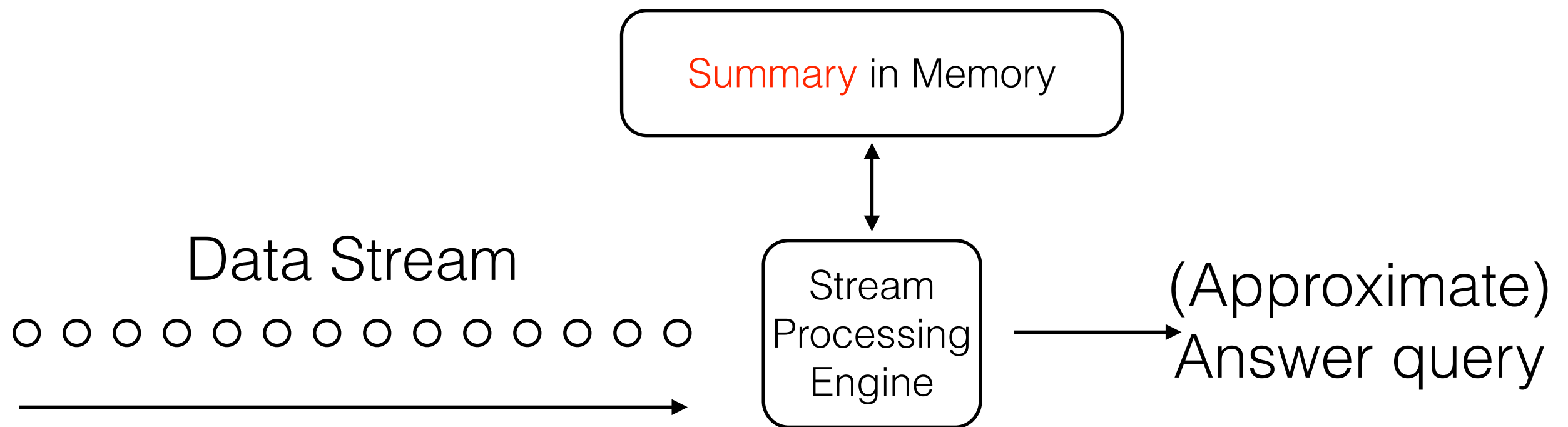
Streaming Algorithms

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



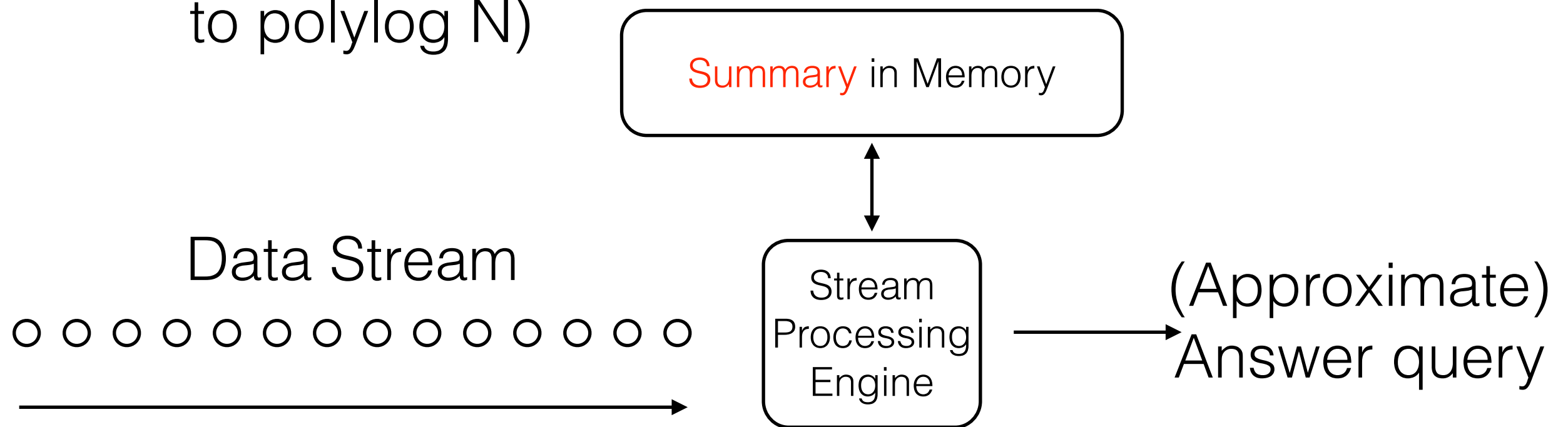
Streaming Algorithms

- 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

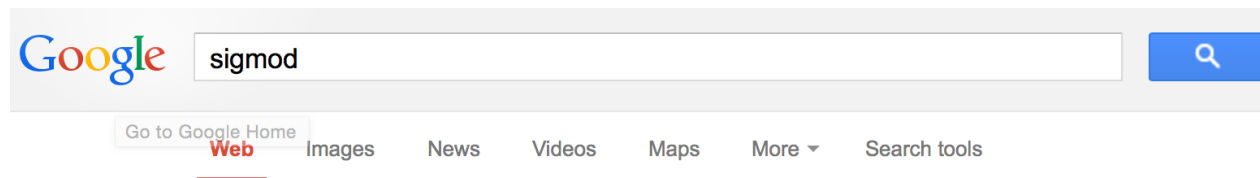
- Sub-linear space
 - Fast update and query time
- Answer queries approximately
- Linear transformation of the data frequencies

Sketches

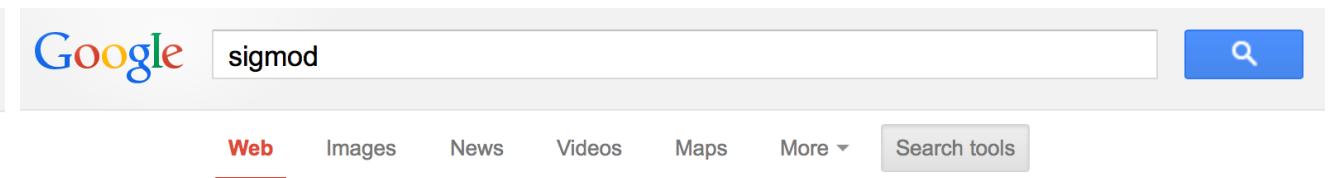
- Sub-linear space
 - Fast update and query time
- Answer queries approximately
- Linear transformation of the data frequencies
- Ephemeral
 - Answer queries on current version of data stream

Query Back in Time

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



Google search for 'sigmod'. The search bar contains 'sigmod' and a magnifying glass icon. Below the search bar are navigation links: 'Go to Google Home', 'Web', 'Images', 'News', 'Videos', 'Maps', 'More', and 'Search tools'. The search results show 'About 606,000 results (0.17 seconds)'.



Google search for 'sigmod'. The search bar contains 'sigmod' and a magnifying glass icon. Below the search bar are navigation links: 'Web', 'Images', 'News', 'Videos', 'Maps', 'More', and 'Search tools'. A date range filter '1 Jan 2013 – 31 Dec 2014' is applied, highlighted with a red box. Other filters include 'Any country', 'Sorted by relevance', 'All results', and 'Clear'.

About 606,000 results (0.17 seconds)

The 2015 ACM SIGMOD/PODS Conference: Melbourne ...

www.sigmod2015.org/

2015 ACM SIGMOD/PODS @ Melbourne, VIC, Australia. The Conference Secretariat can be contacted on 0421 665 924 between 8:00am to 5:00pm Sunday 31 ...

[Important Dates](#) - [Calls for Submissions](#) - [Sigmod pc](#) - [Affiliated Workshops](#)

WHAT'S NEW! — ACM SIGMOD Official Website

www.sigmod.org/

Microsoft is the sponsor of the new SIGMOD Systems Award. With support from Microsoft, SIGMOD was able to raise the award levels significantly for these two ...

SIGMOD - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/SIGMOD

SIGMOD is the Association for Computing Machinery's Special Interest Group on Management of Data, which specializes in large-scale data management ...

The 2014 ACM SIGMOD/PODS Conference: Snowbird, Utah ...

www.sigmod2014.org/

2014 ACM SIGMOD/PODS @ Snowbird, Utah, USA. Proceedings. The proceedings are open access and available from the following links (use the " Table of ...

[Research Papers](#) - [SIGMOD Sessions](#) - [Important Dates](#) - [Calls for Submissions](#)

Any country ▾ 1 Jan 2013 – 31 Dec 2014 ▾ Sorted by relevance ▾ All results ▾ Clear

The 2014 ACM SIGMOD/PODS Conference: Snowbird, Utah ...

www.sigmod2014.org/

Jun 21, 2013 - 2014 ACM SIGMOD/PODS @ Snowbird, Utah, USA. Proceedings. The proceedings are open access and available from the following links (use the " Table of ...

Research Papers

SIGMOD 2014: Accepted Research Papers. AutoPlait ...

Sigmod pc

Xiaokui Xiao (Nanyang Technological University ...

[More results from sigmod2014.org »](#)

Important Dates - sigmod 2015

www.sigmod2015.org/calls_papers_important_dates.shtml

Jun 17, 2014 - Important Dates. SIGMOD Deadlines. Research paper first-round submission dates. August 7, 2014, 11: 59pm PST: Abstract submission; August 14, 2014, ...

Affiliated Workshops - sigmod 2015

www.sigmod2015.org/org_sigmod_workshops.shtml

Oct 3, 2014 - Organization: Affiliated Workshops. DaMoN (The 11th International Workshop on Data Management on New Hardware) Ippokratis Pandis ...

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



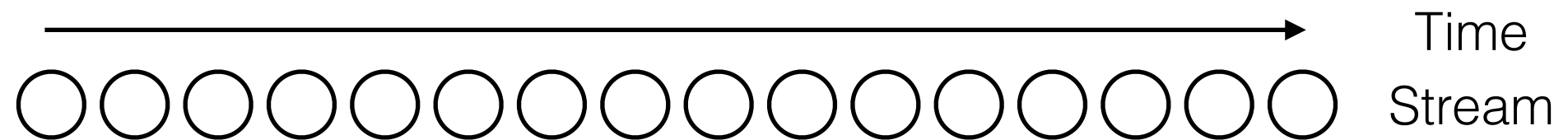
Persistent Sketch

Query on historical data

Sub-linear space

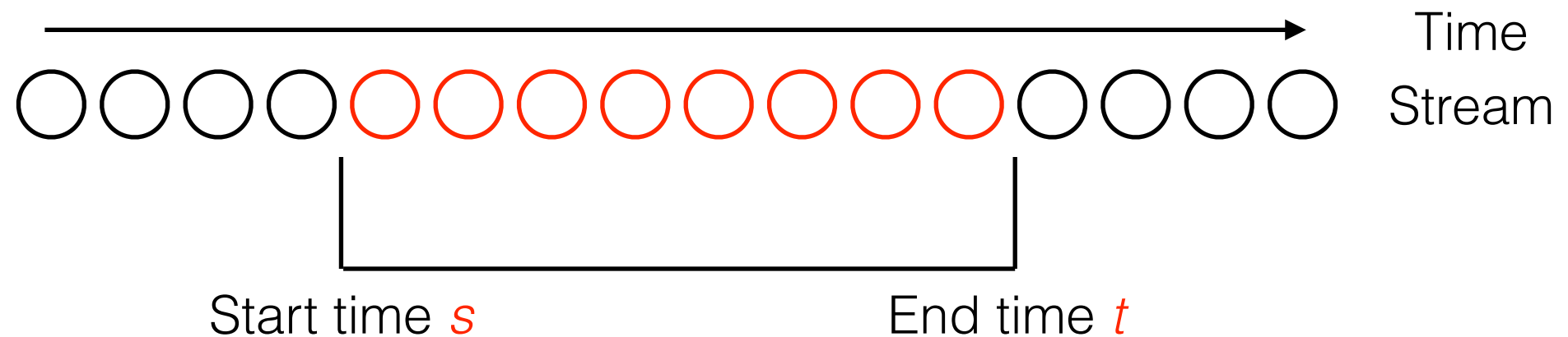
Persistent Sketch

- Historical window query



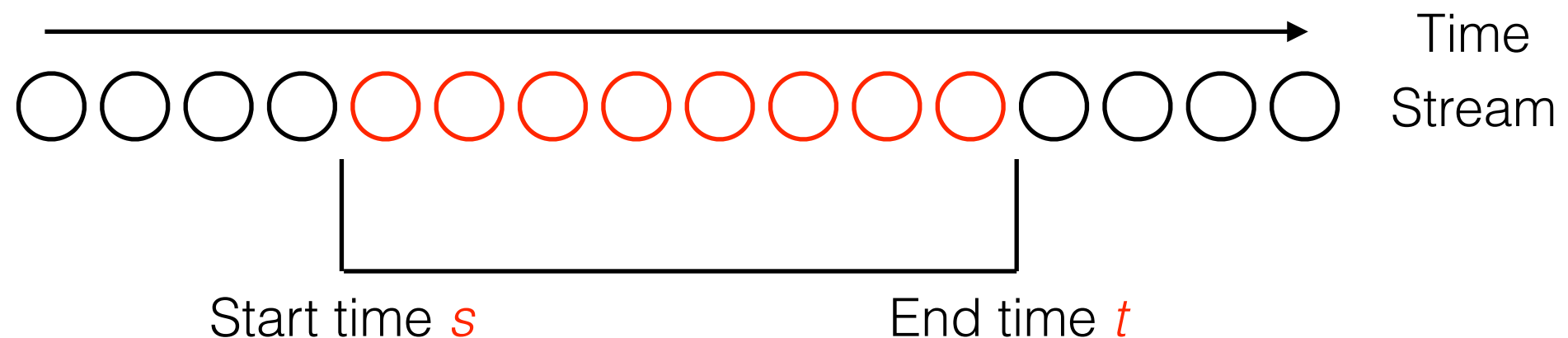
Persistent Sketch

- 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

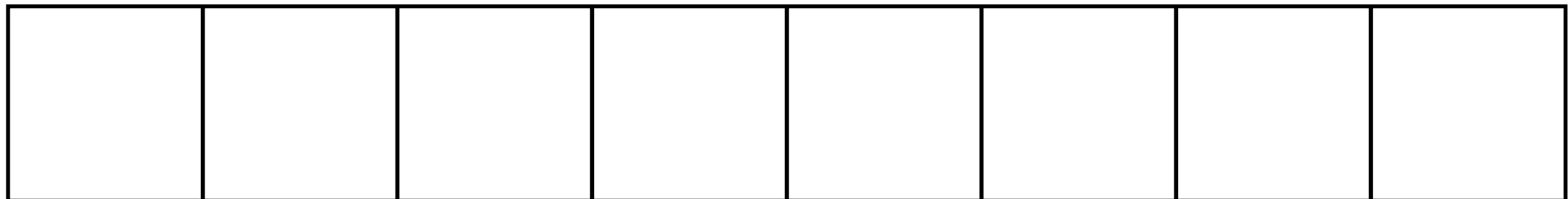
[Cormode and Muthukrishnan 2005]

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

Count-Min Sketch

[Cormode and Muthukrishnan 2005]

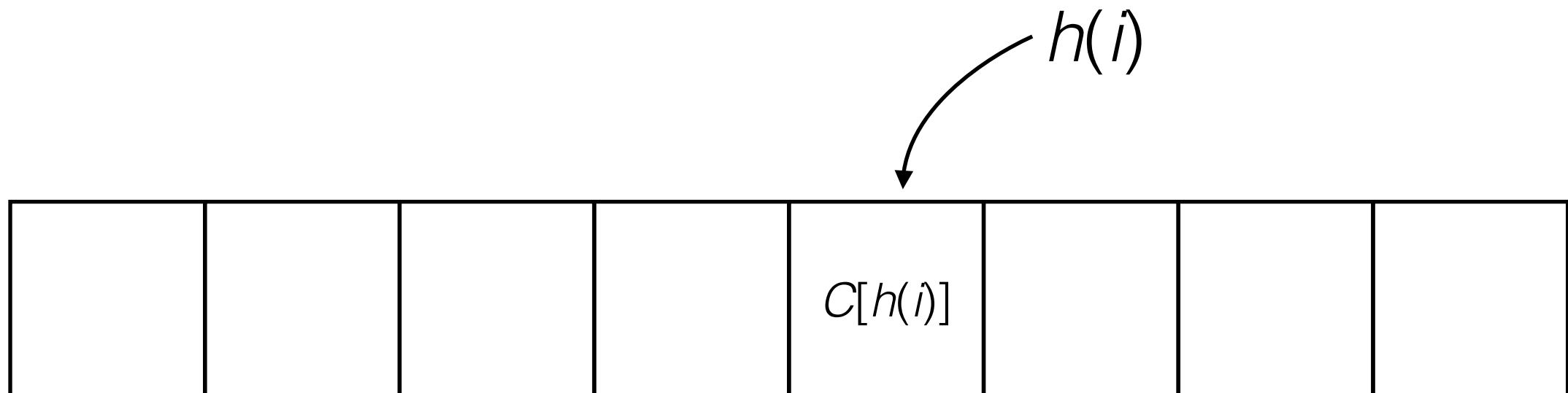
- Given an error parameter ϵ
- Choose a hash function $h: [n] \rightarrow [2/\epsilon]$ and build a hash table of size $2/\epsilon$



Count-Min Sketch

[Cormode and Muthukrishnan 2005]

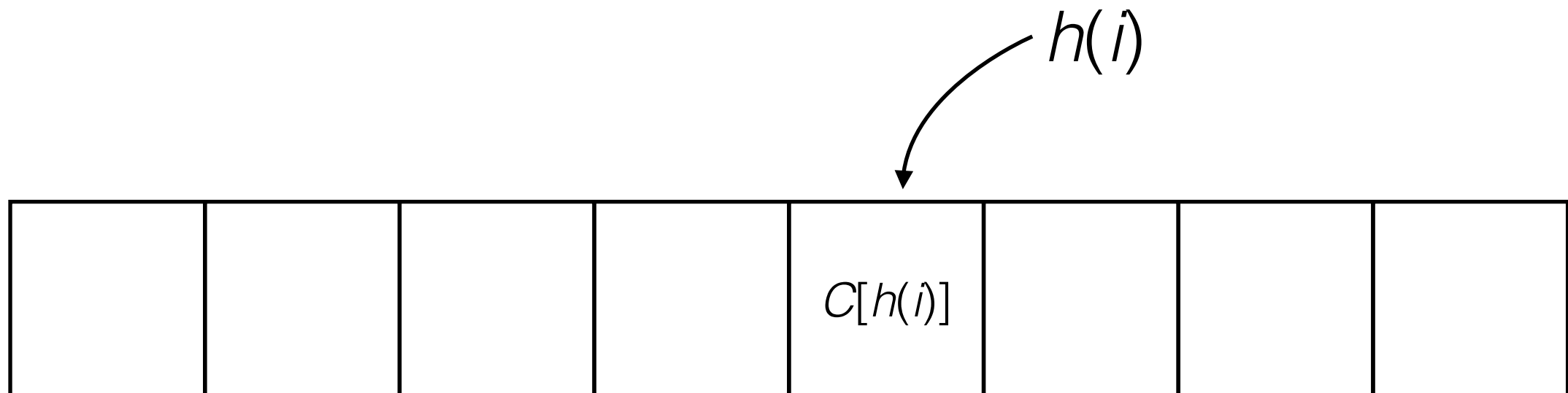
- Given an error parameter ϵ
- Choose a hash function $h: [n] \rightarrow [2/\epsilon]$ and build a hash table of size $2/\epsilon$



Count-Min Sketch

[Cormode and Muthukrishnan 2005]

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

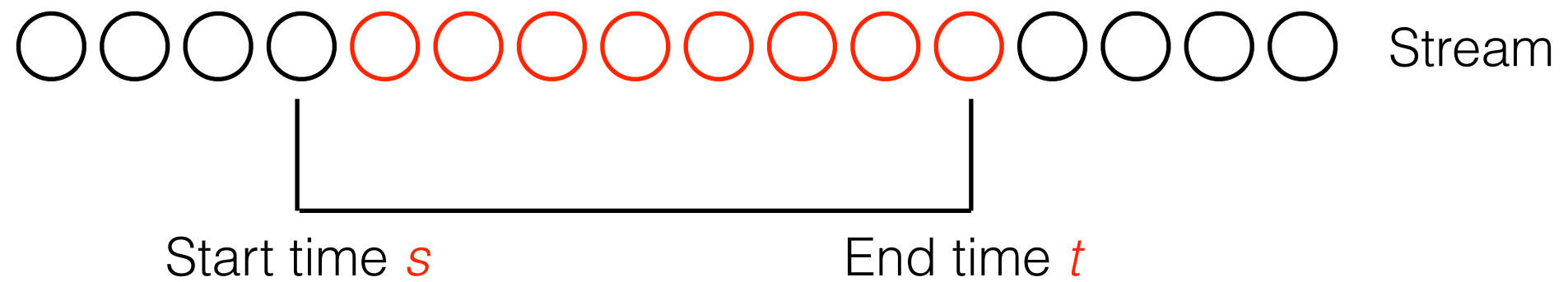


$$C[h(i)] = C[h(i)] + 1$$

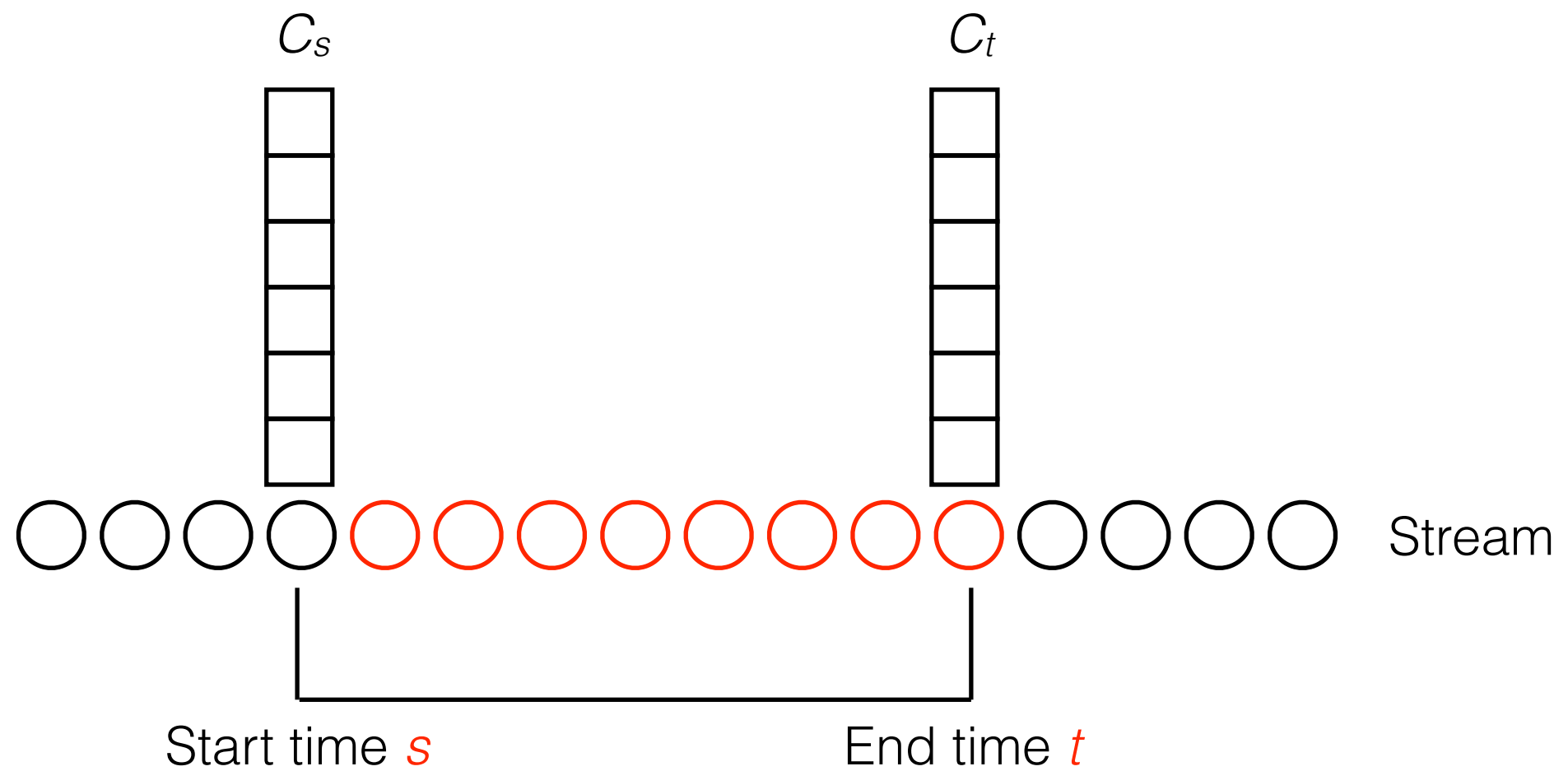
Linear Transformation

$$h(i) \begin{pmatrix} 0, 1, 0, \dots, 0, \dots, 0, 0, \\ \dots \\ 0, 0, 0, \dots, 1, \dots, 0, 0, \\ \dots \\ 0, 0, 0, \dots, 0, \dots, 1, 0, \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \dots \\ f_i \\ \dots \\ f_N \end{pmatrix} = \begin{array}{|c|} \hline \\ \hline \\ \hline C[h(i)] \\ \hline \\ \hline \\ \hline \end{array}$$

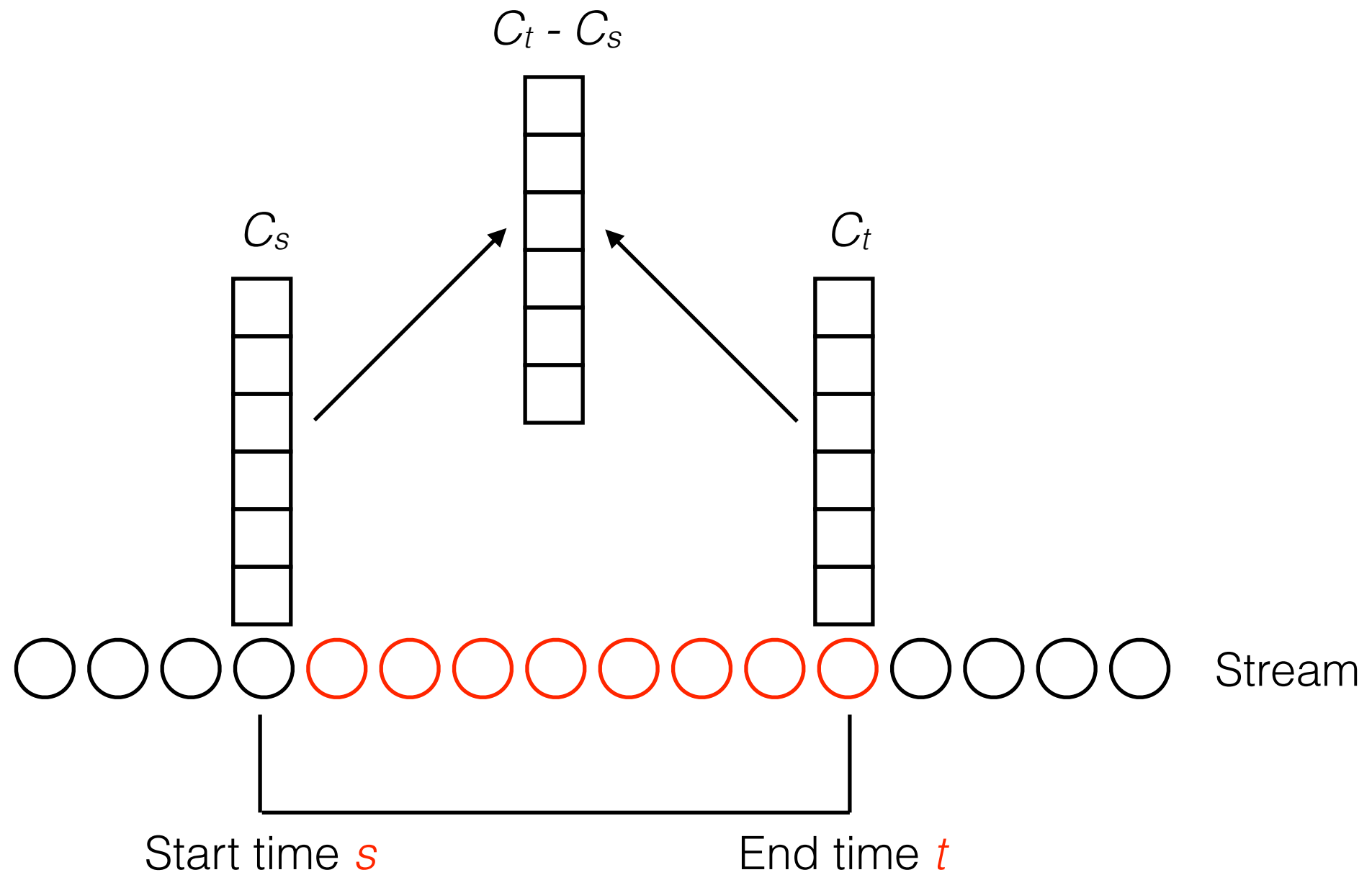
Linear Transformation



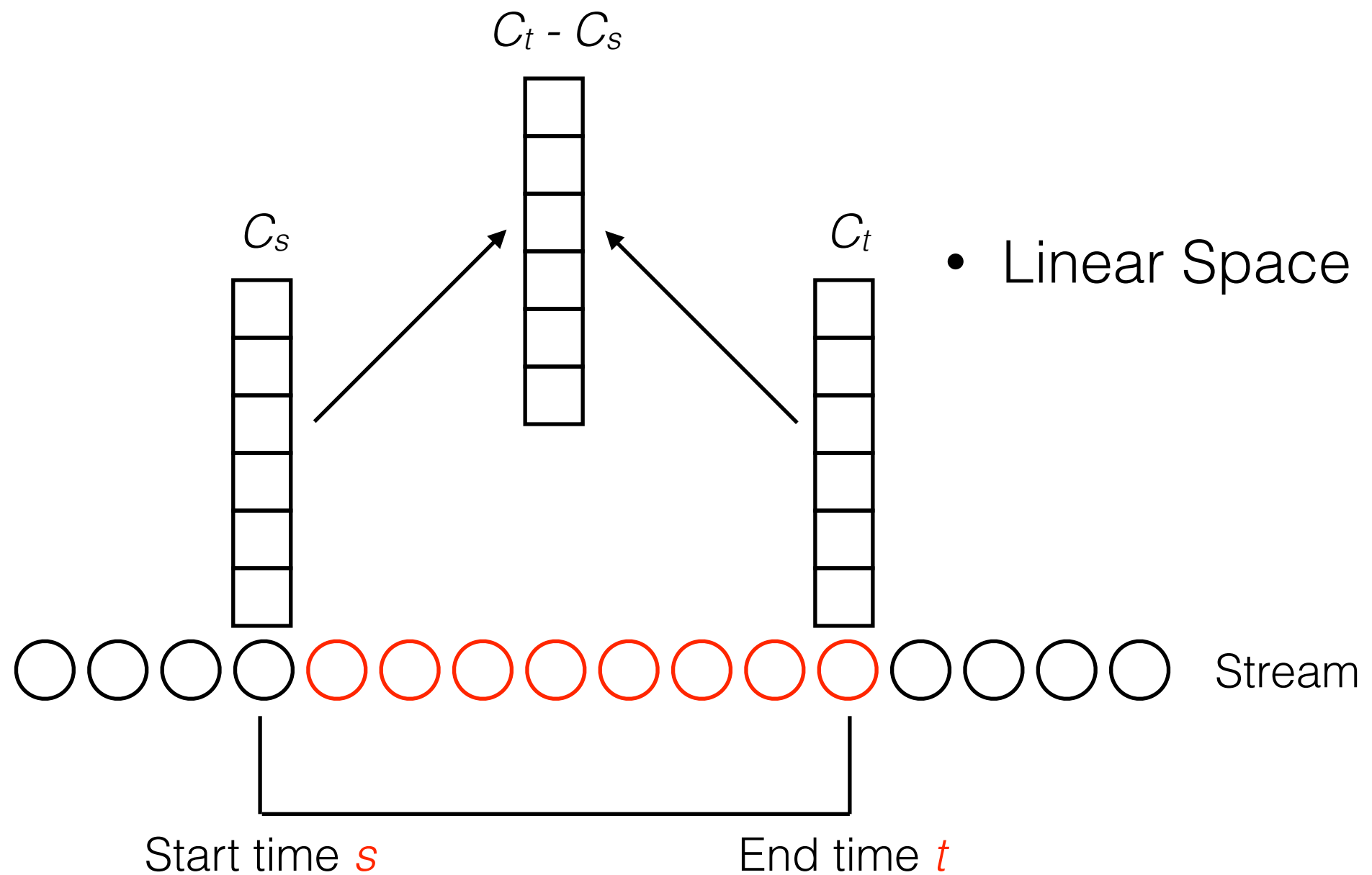
Linear Transformation



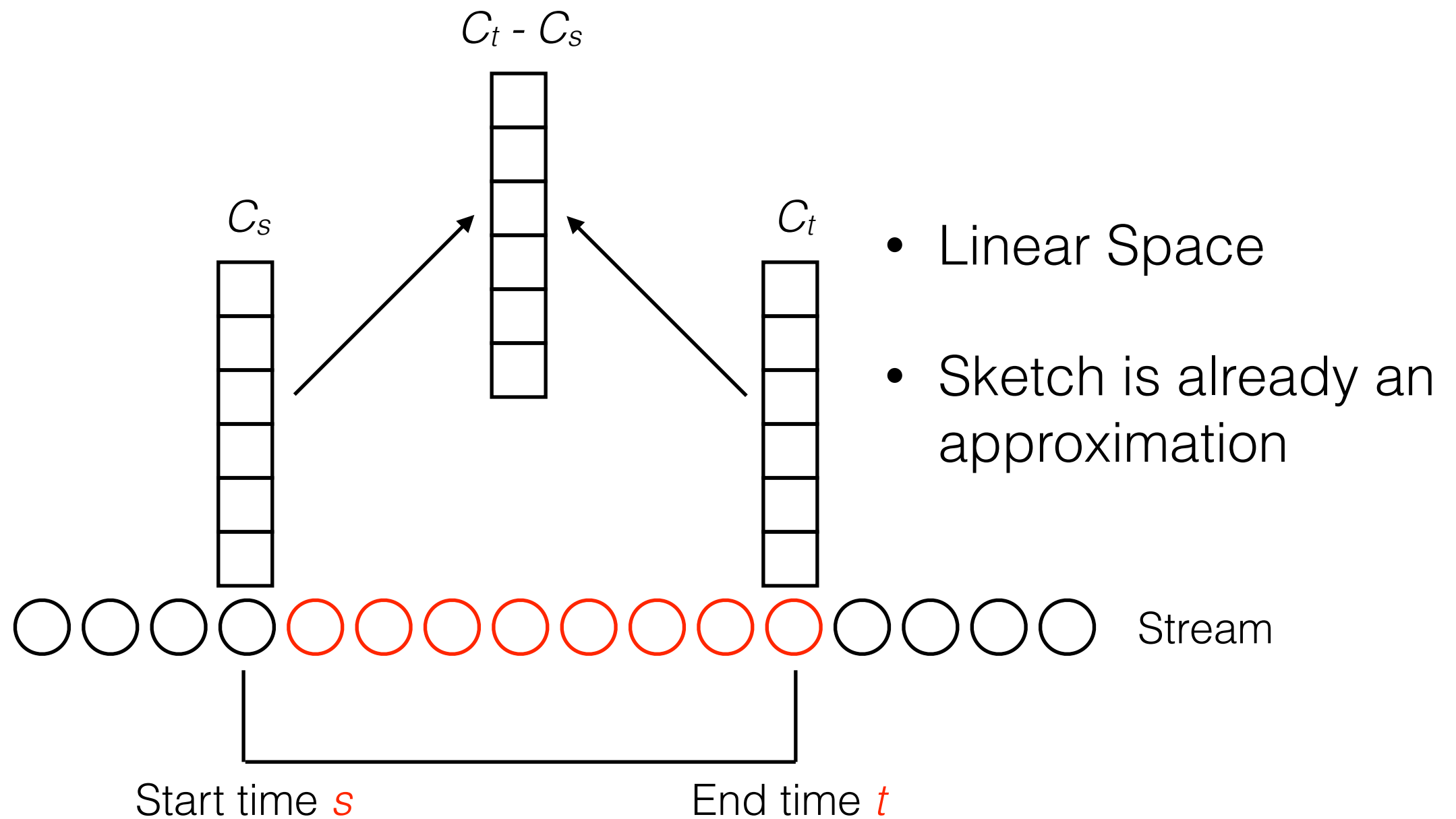
Linear Transformation



Linear Transformation

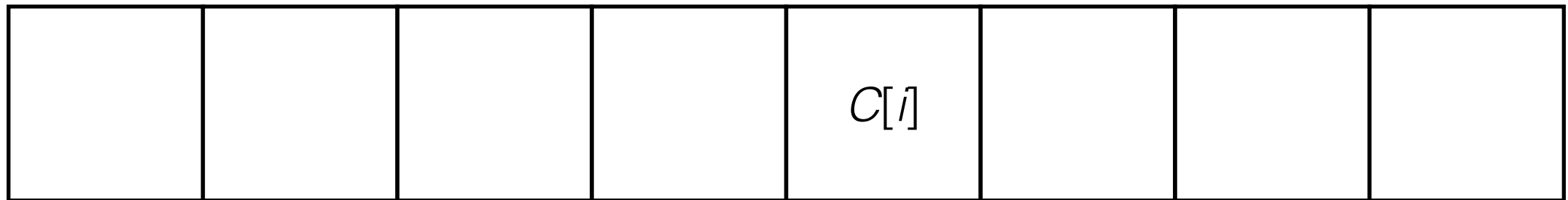


Linear Transformation



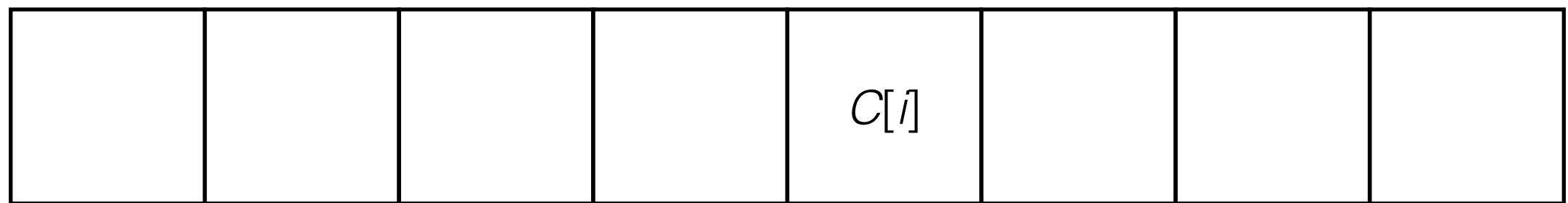
Baseline Solution

Ephemeral sketch:

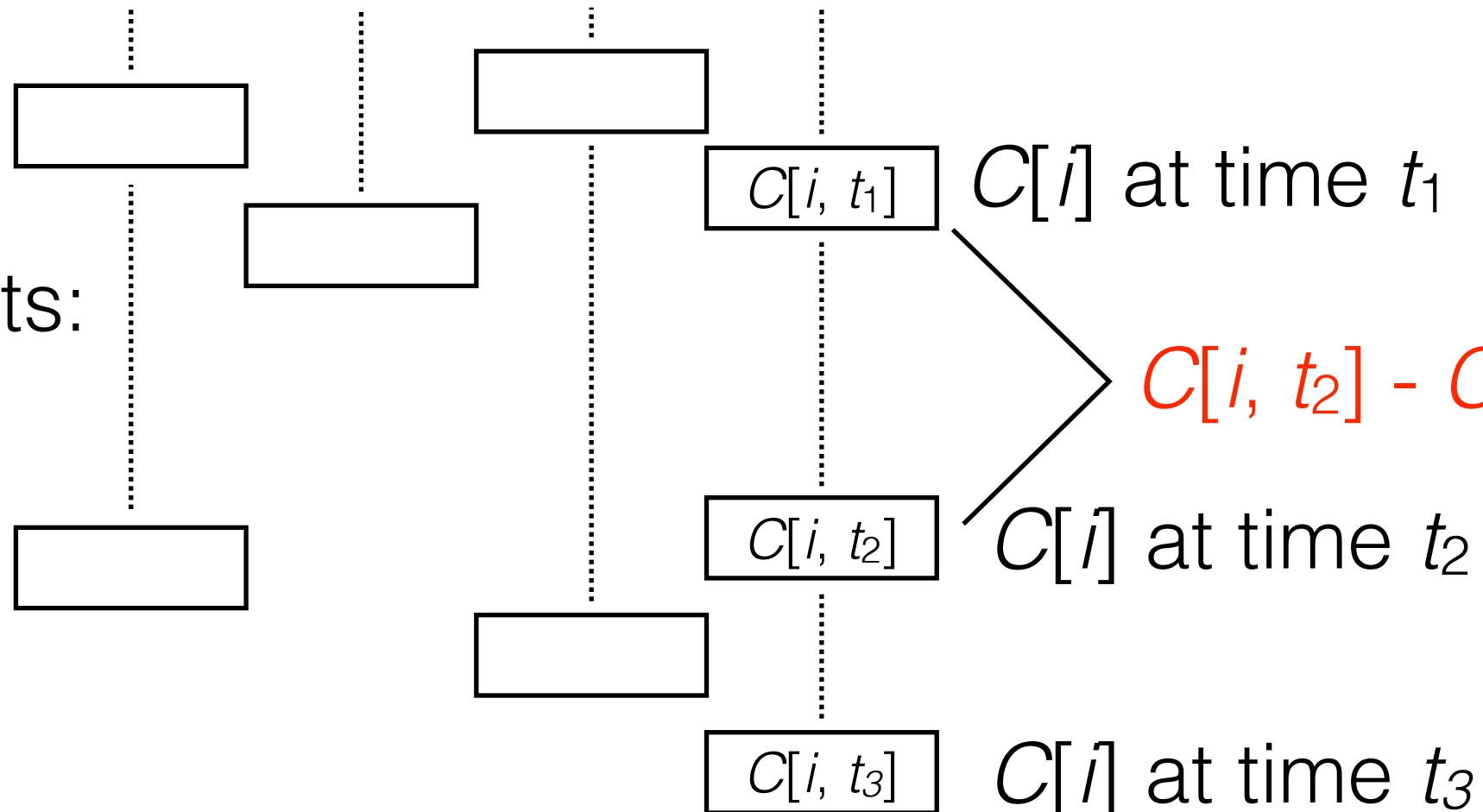


Baseline Solution

Ephemeral sketch:

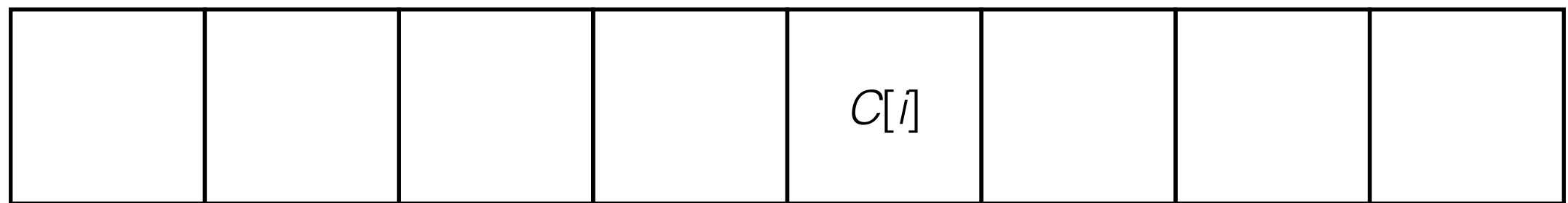


Historical Lists:



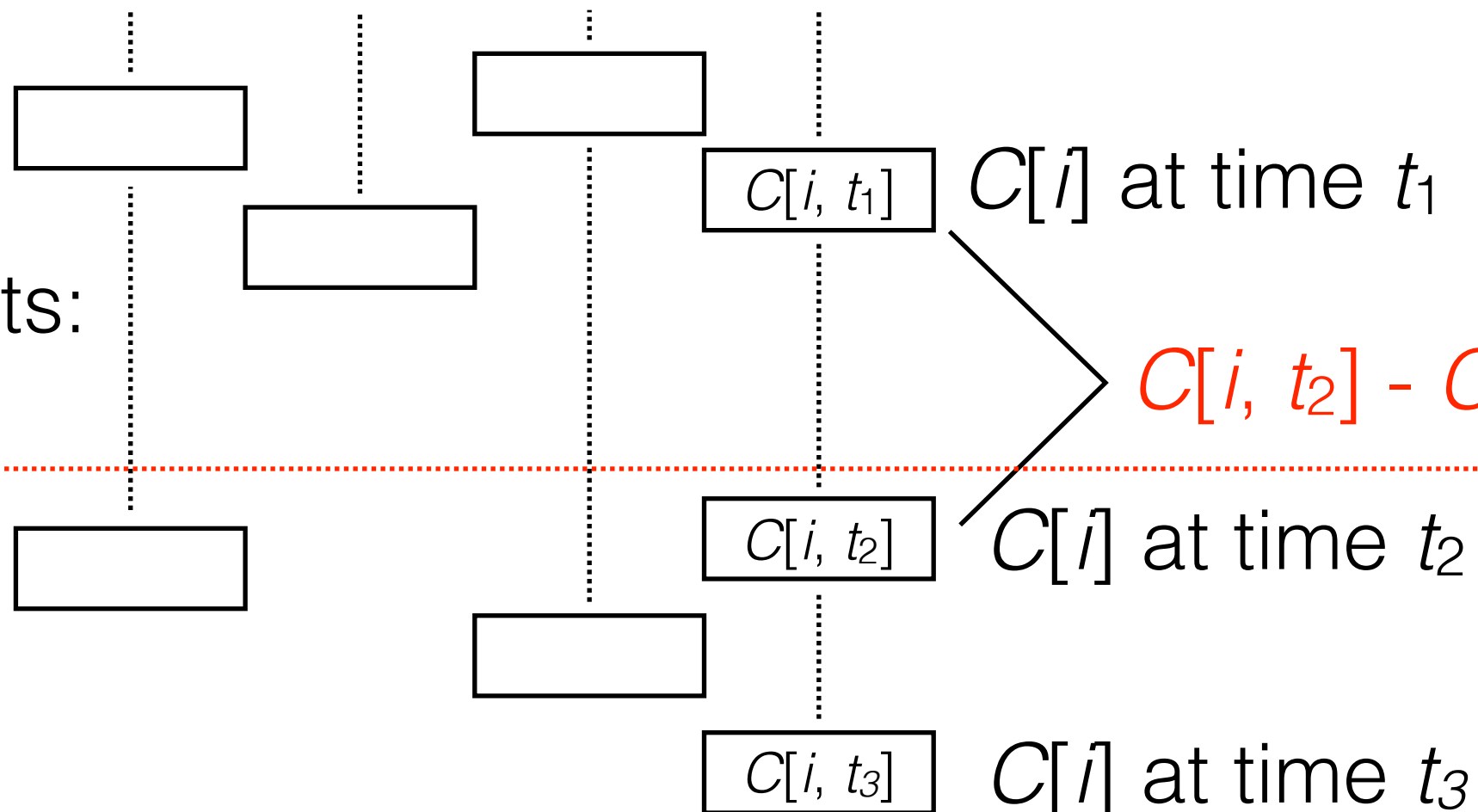
Baseline Solution

Ephemeral sketch:



Historical Lists:

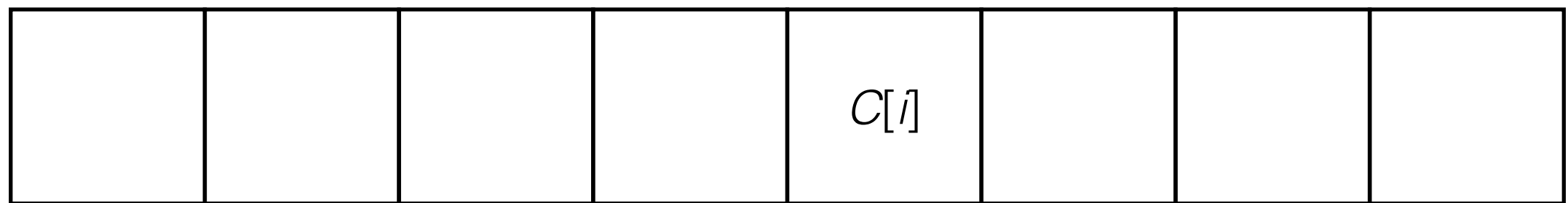
Query time t



$$C[i, t_2] - C[i, t_1] \approx \Delta$$

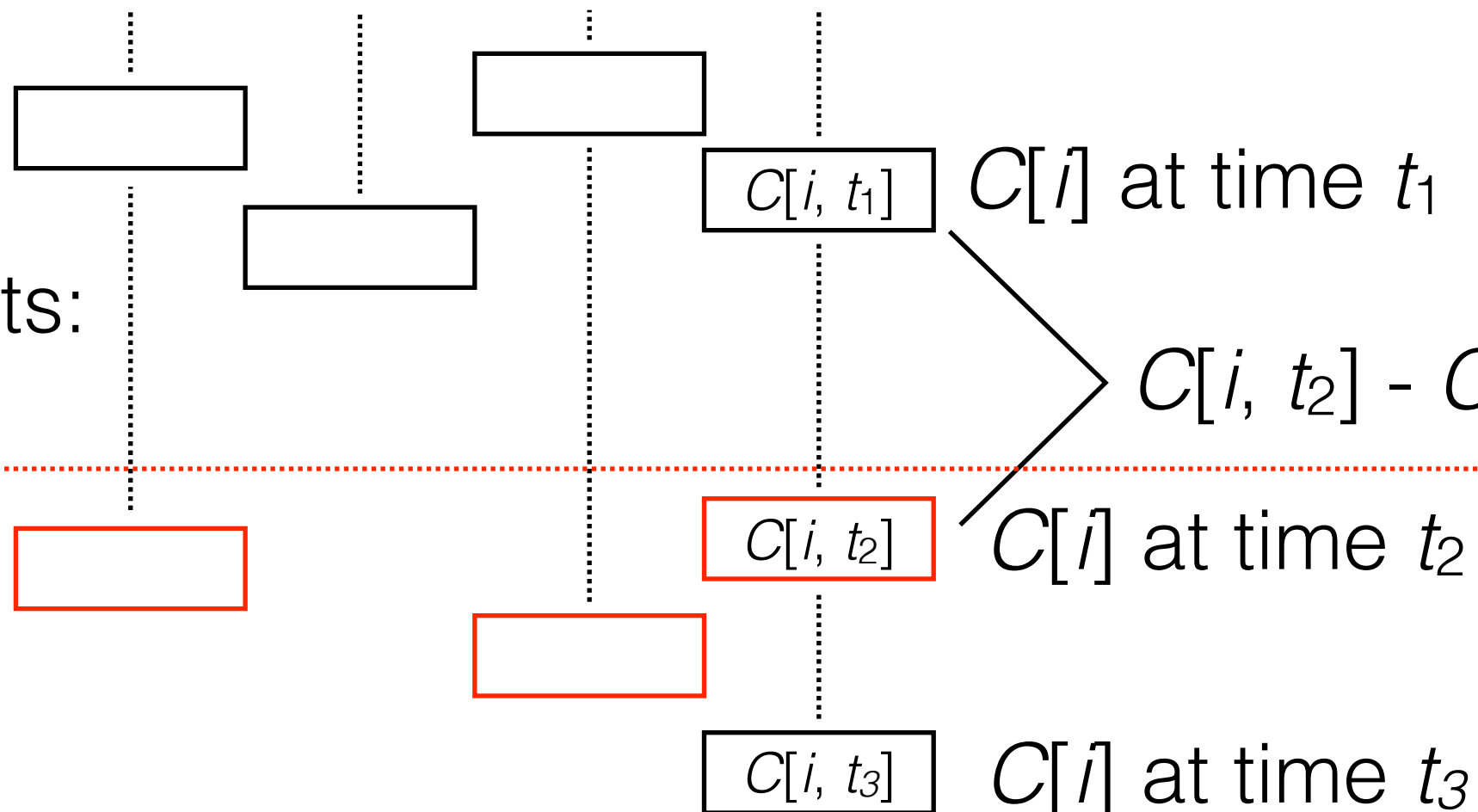
Baseline Solution

Ephemeral sketch:



Historical Lists:

Query time t



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: $\epsilon \|\mathbf{f}(s, t)\|_1$ (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 most frequent URLs between day 34 and day 37
- Error: $\epsilon \|\mathbf{f}(s, t)\|_1$ (ephemeral error) + Δ (persistent error)

Size of the stream
between s and t

Baseline Solution

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

Size of the stream
between s and t

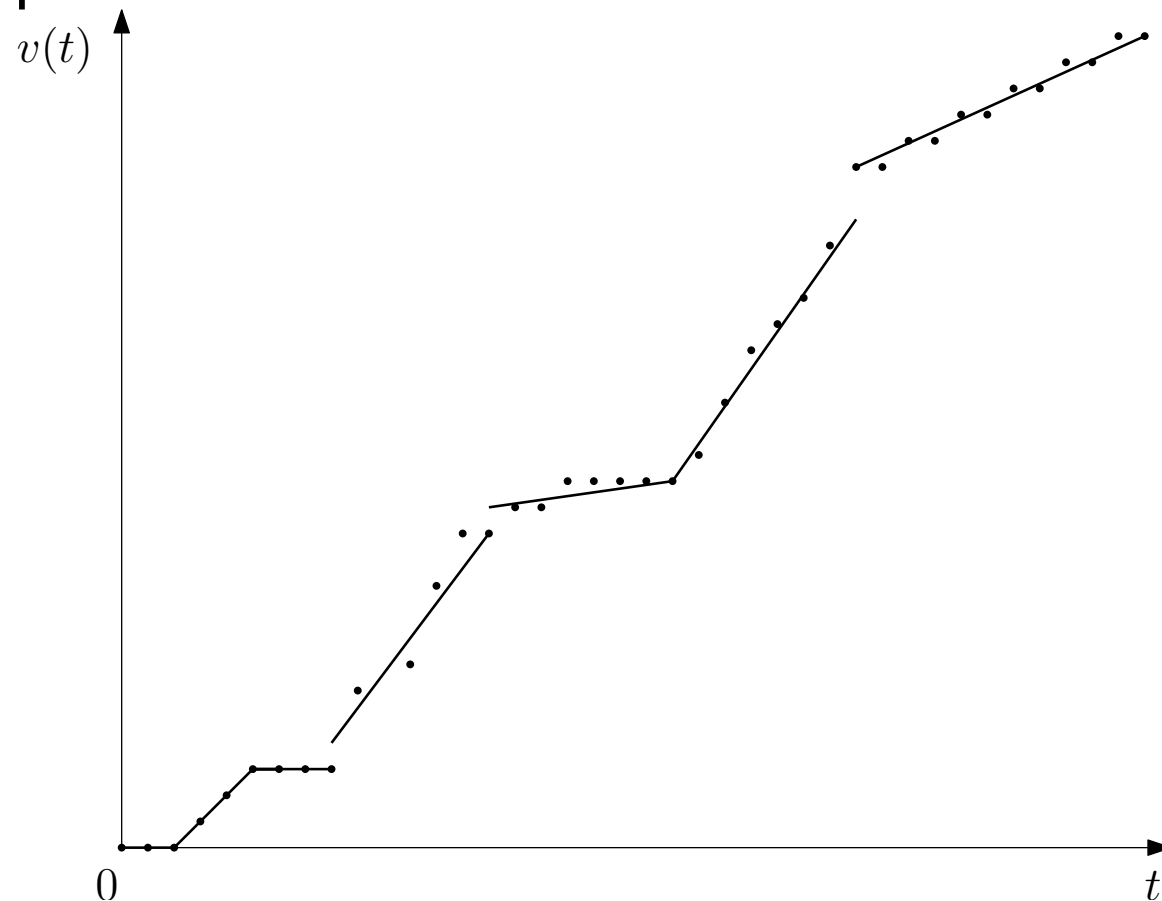
Baseline Solution

- Historical window **point/heavy hitters** query:
 - What is frequency of “/images/space.gif” between day 34 and day 37
 - What are the most frequent URLs between day 34 and day 37
- Error: $\epsilon \|\mathbf{f}(s, t)\|_1$ (ephemeral error) + Δ (persistent error)
- Space: proportional to $(1/\epsilon + m/\Delta)$
- Cannot handle (self) join size queries

Size of the stream
between s and t

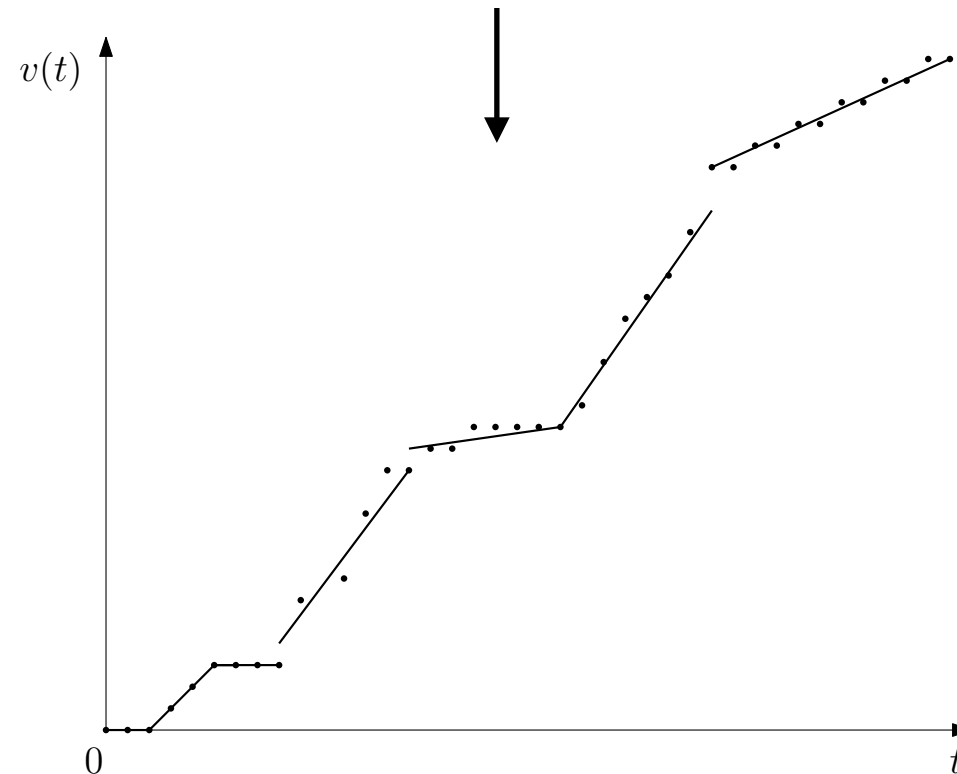
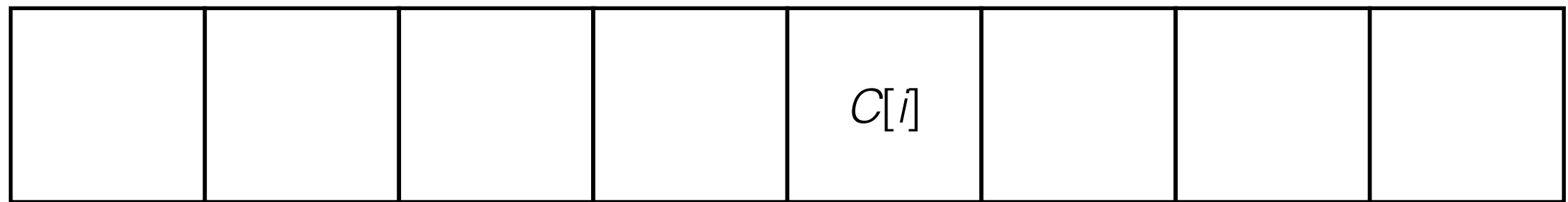
Piece-wise Linear Approximation

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



PLA-based Persistent Sketch

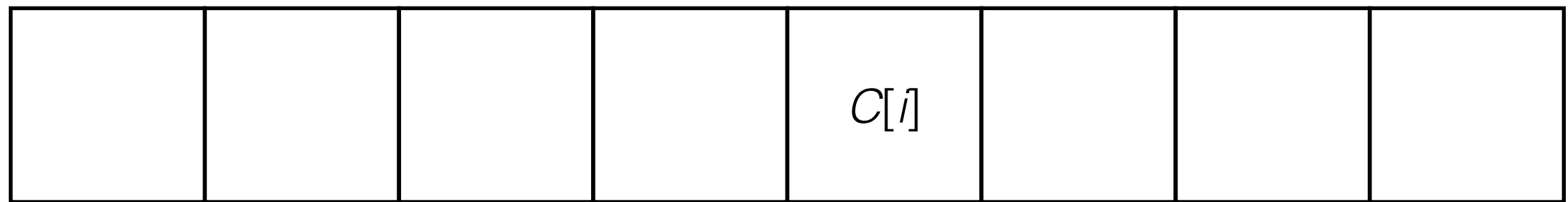
Ephemeral sketch:



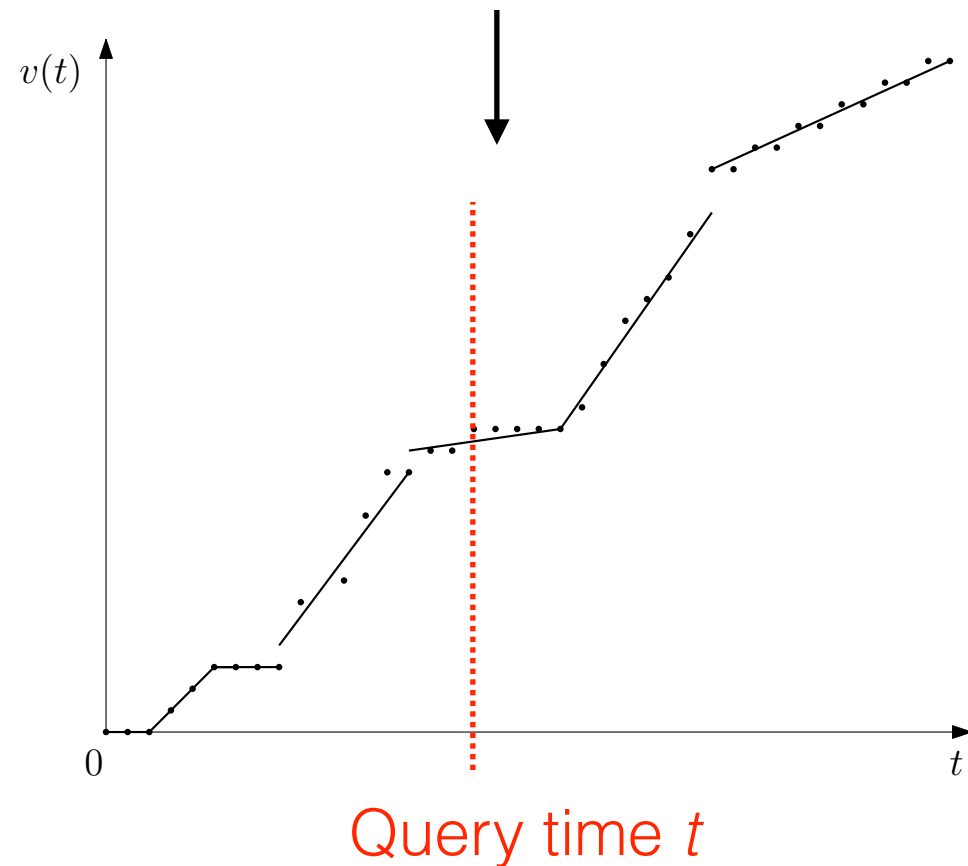
PLA generator:

PLA-based Persistent Sketch

Ephemeral sketch:



PLA generator:



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: $\epsilon \|\mathbf{f}(s, t)\|_1$ (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: $\epsilon \|\mathbf{f}(s, t)\|_1$ (ephemeral error) + Δ (persistent error)
- Space: proportional to $(1/\epsilon + m/\Delta^2)$ in **random stream model**

Estimating Join Size

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

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

Estimating Join Size

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

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

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

Estimating Join Size

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

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

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

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

Estimating Join Size

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

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

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

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

- Bias will amplify error significantly

Estimating Join Size

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

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

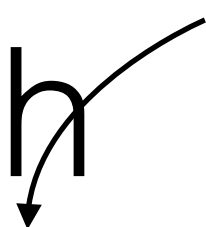
- Estimating (self) join size in a persistent sketch:

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

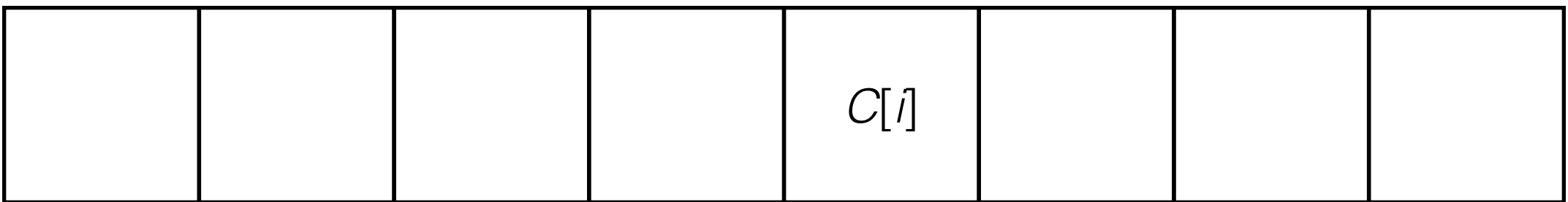
- Bias will amplify error significantly
- Need unbiased estimator of the counter

Sampling Based Persistent Sketch

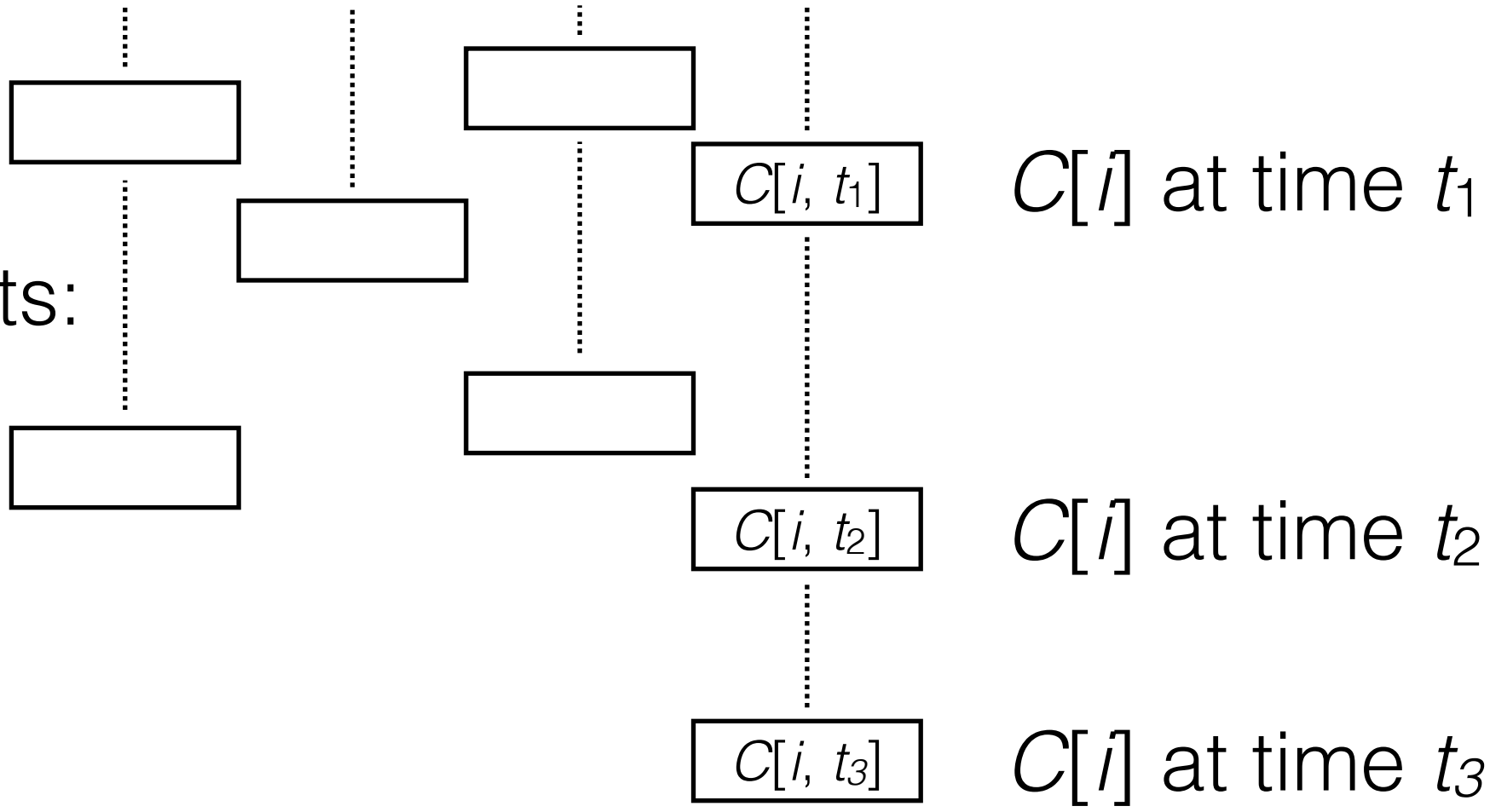
$h(i)$



Ephemeral sketch:



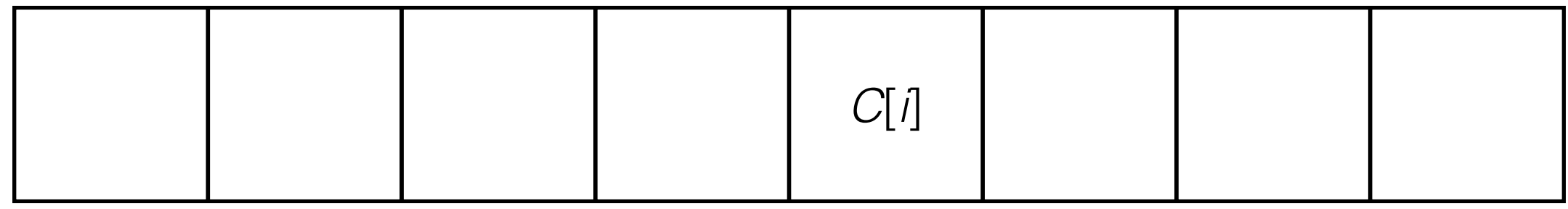
Historical Lists:



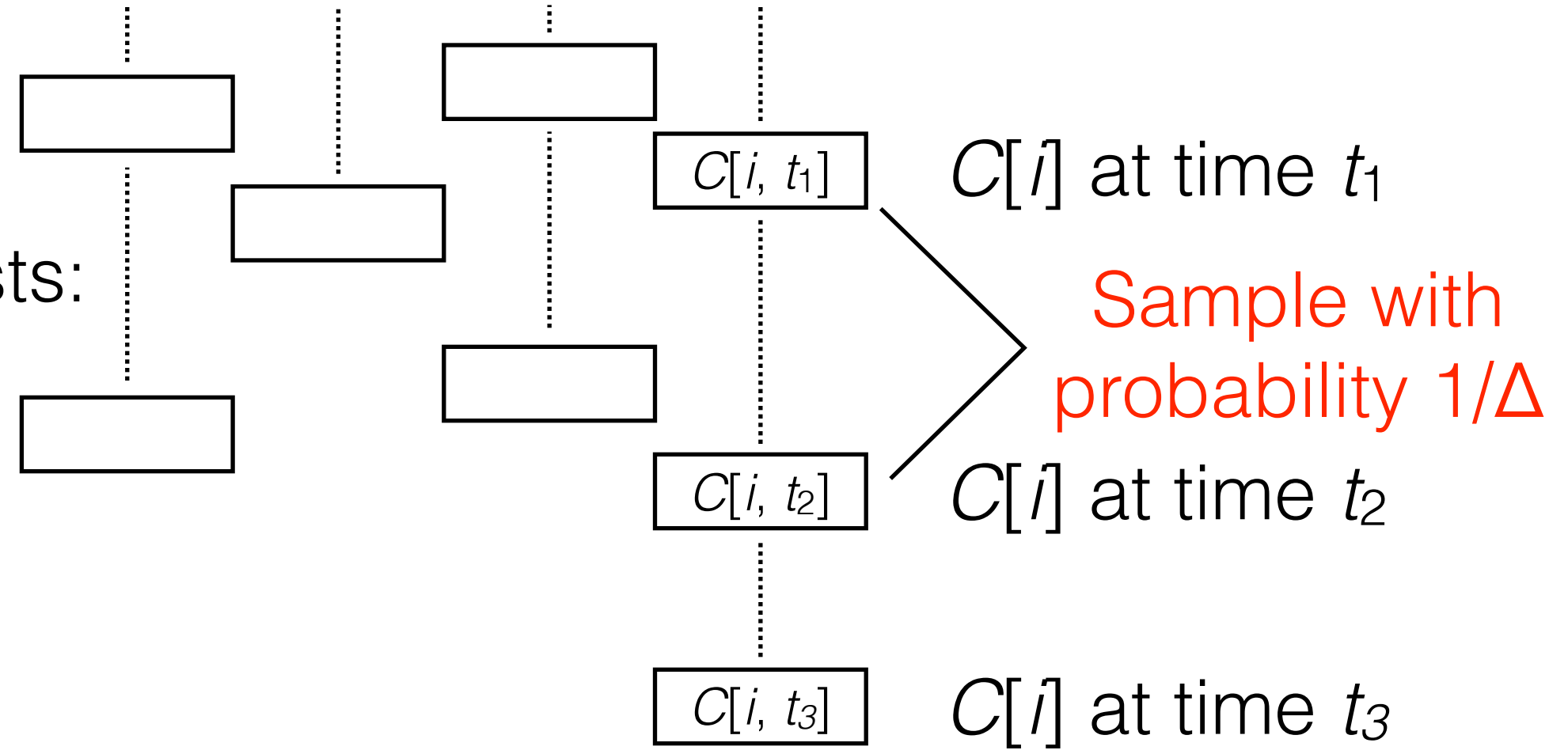
Sampling Based Persistent Sketch

$h(i)$

Ephemeral sketch:



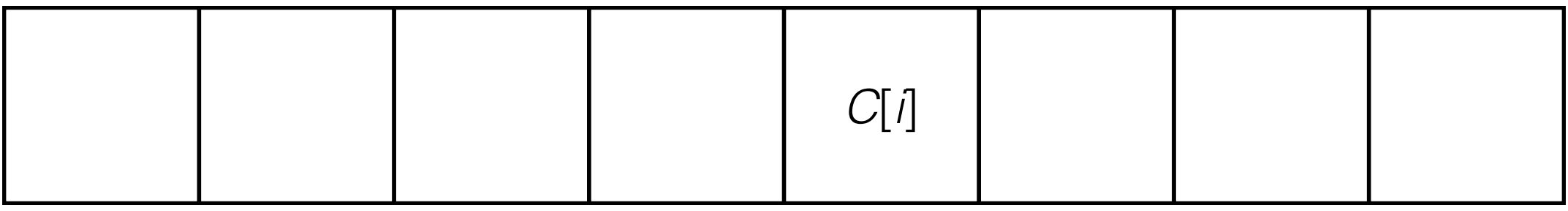
Historical Lists:



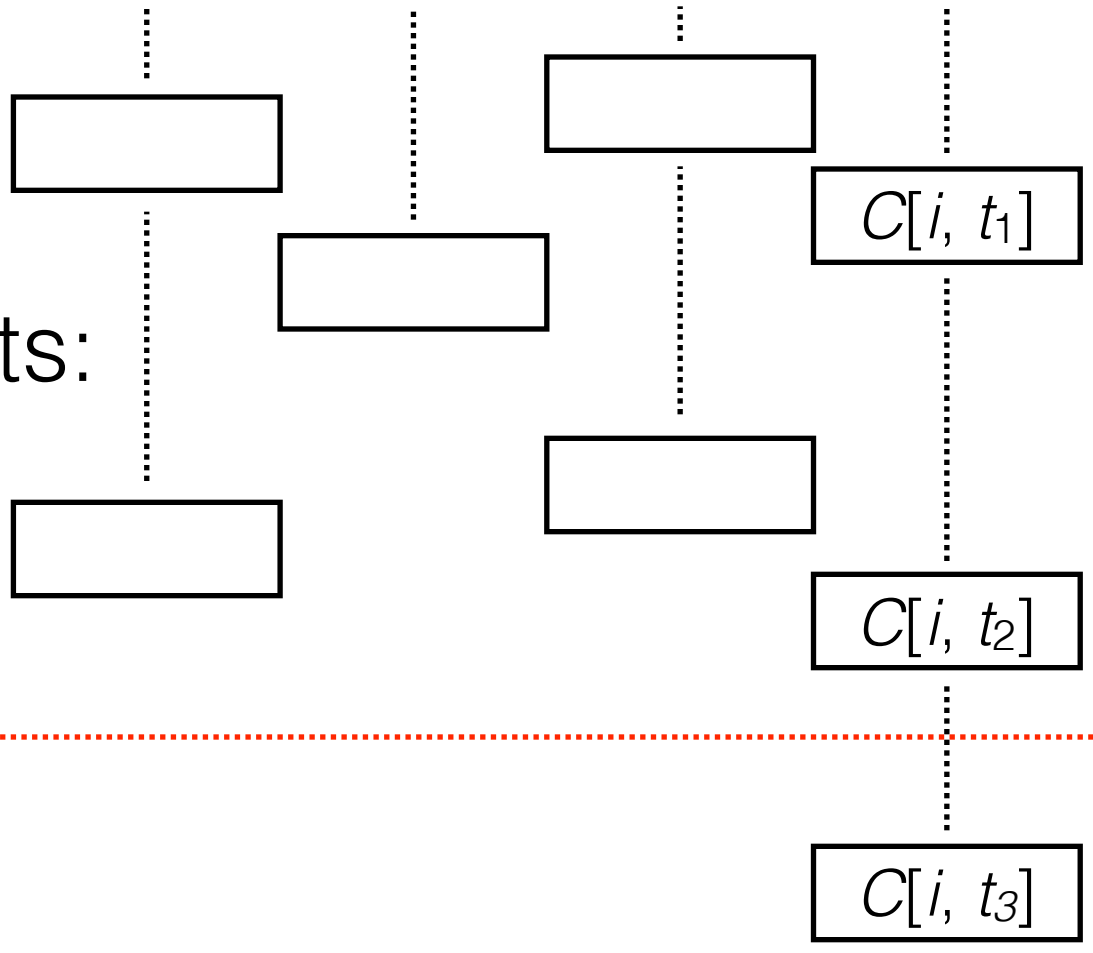
Sampling Based Persistent Sketch

$h(i)$

Ephemeral sketch:



Historical Lists:



$C[i]$ at time t_1

Sample with probability $1/\Delta$

$C[i]$ at time t_2

$C[i]$ at time t_3

Query time t

0

$C[i, t_3] + \Delta - 1$

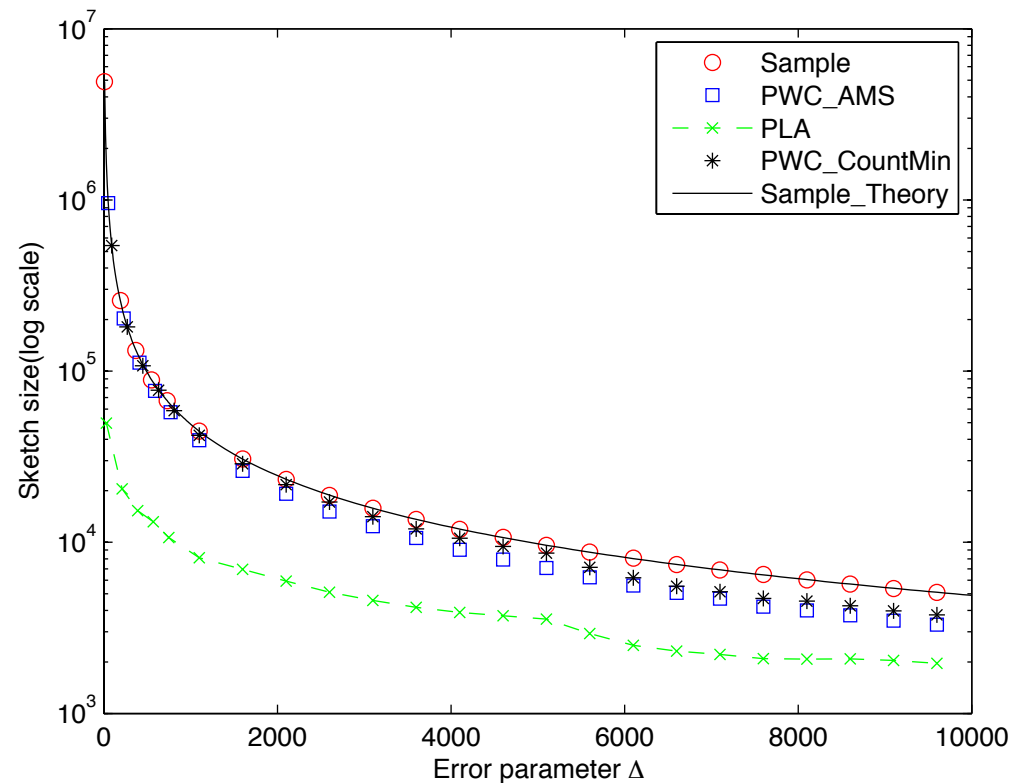
Sampling based AMS Sketch

- 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: $\varepsilon \sqrt{\left(\|\mathbf{f}_{s,t}\|_2^2 + \left(\frac{\Delta \mathbf{f}}{\varepsilon}\right)^2 \right) \left(\|\mathbf{g}_{s,t}\|_2^2 + \left(\frac{\Delta \mathbf{g}}{\varepsilon}\right)^2 \right)}$
- Space: proportional to $(1/\varepsilon + m/\Delta)$

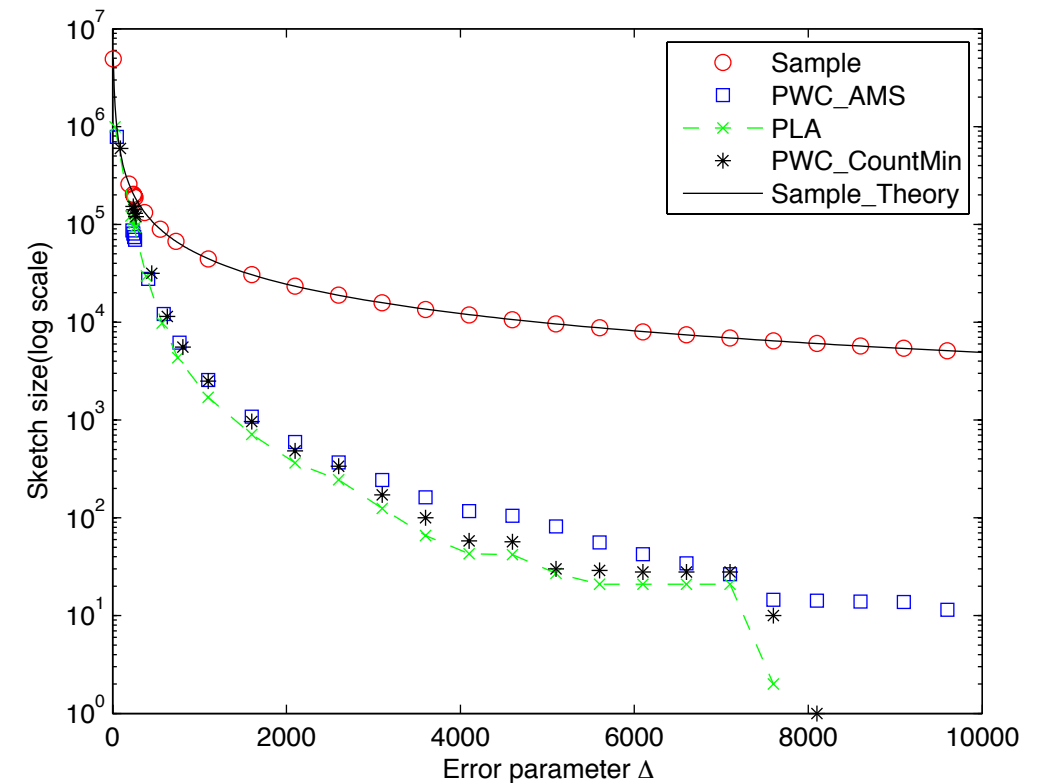
Experimental Study

- 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



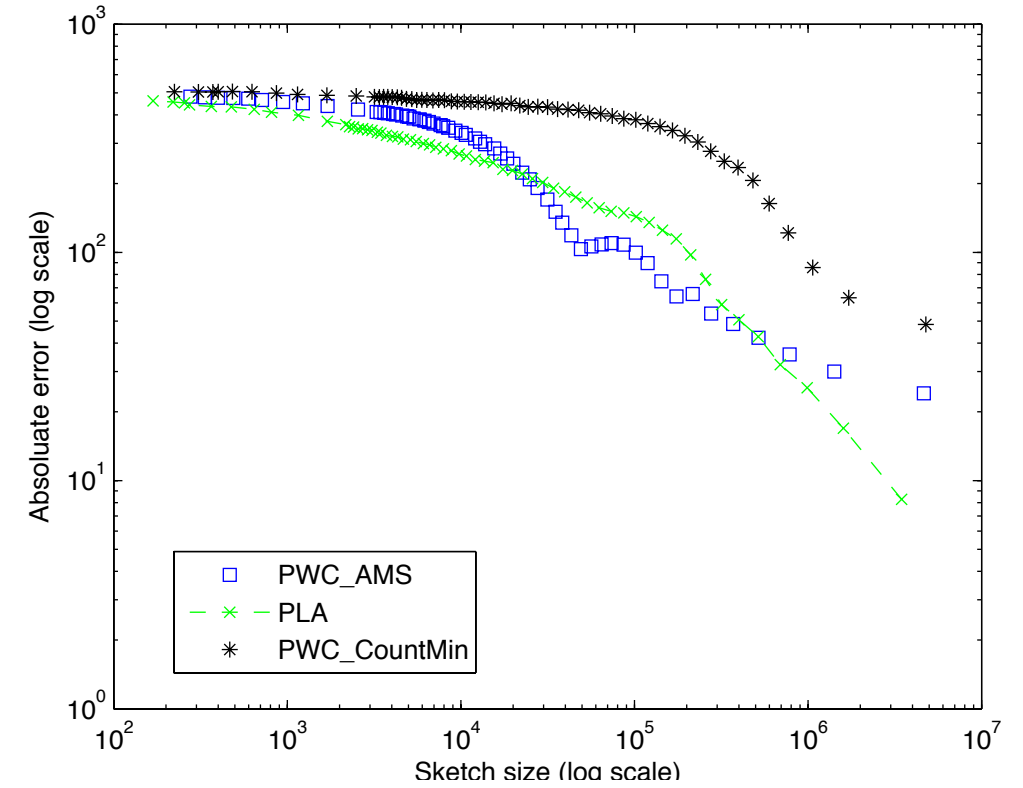
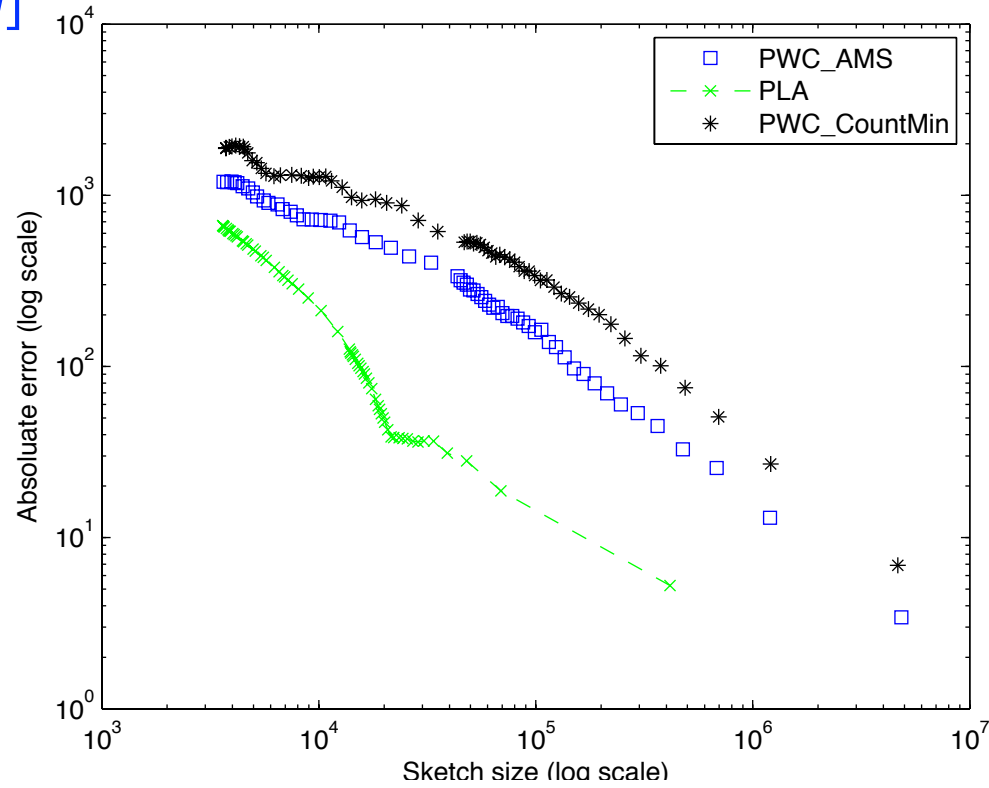
Experimental Study

Query range
(0.2N, 0.6M]

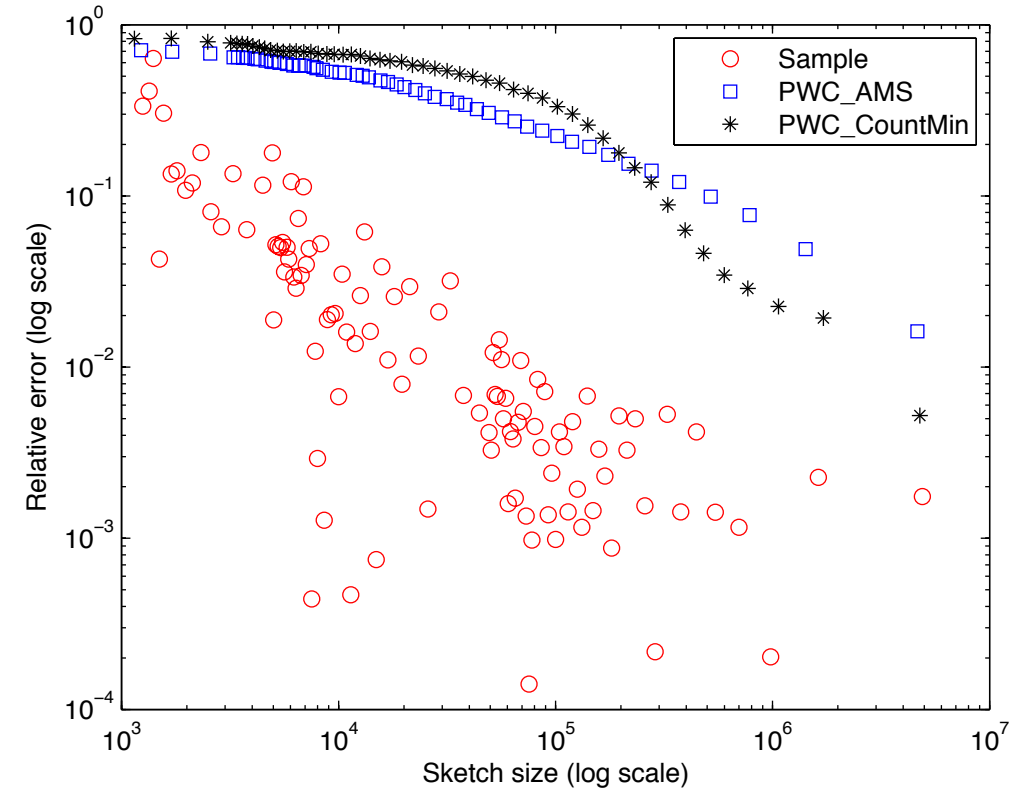
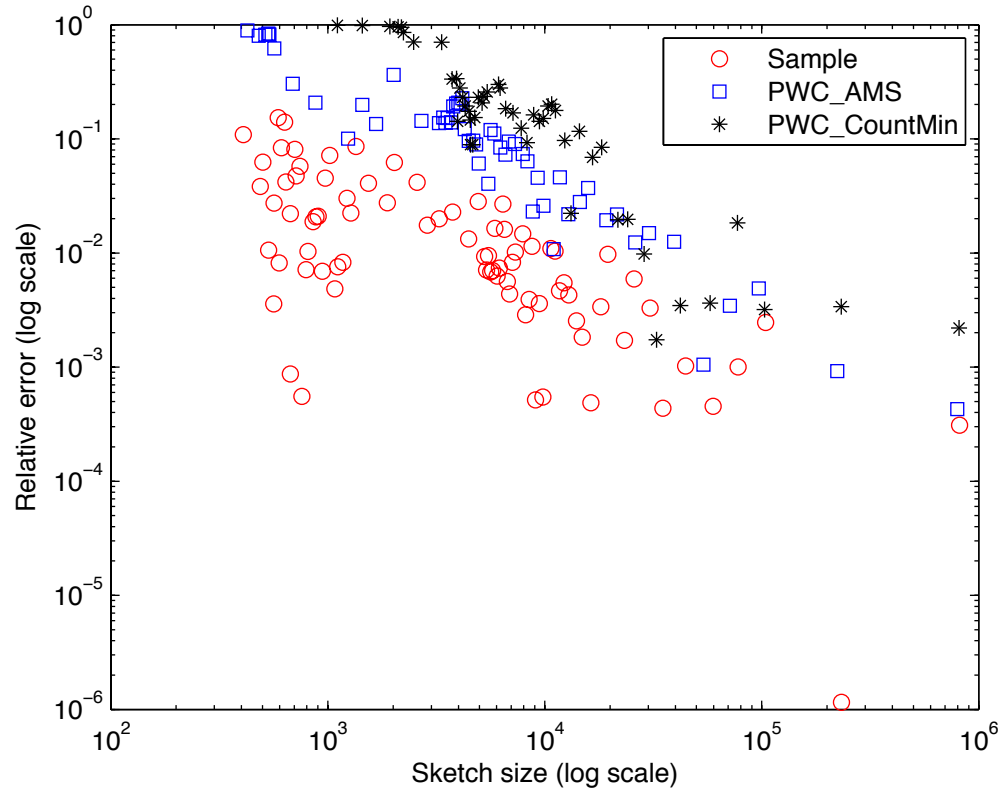
Requested URL

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!