Building Wavelet Histograms on Large Data in MapReduce

Jeffrey Jestes[1] Ke Yi[2] Feifei Li[1]

THE UNIVERSITY OF UTAH®

[1]School of Computing
University of Utah

HKUST

[2]Department of Computer Science
Hong Kong University of Science and Technology

November 16, 2011

Introduction

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮

- For large data we often wish to obtain a concise summary.

Introduction

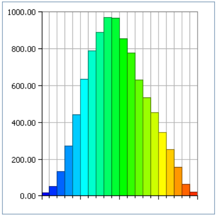
Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮

- For large data we often wish to obtain a concise summary.

Introduction

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮

→



- For large data we often wish to obtain a concise summary.

Outline

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.
  - Let A be an attribute of dataset R.

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.
  - Let A be an attribute of dataset R.
    - Values of A are drawn from finite domain $[u] = \{1, \cdots, u\}$.

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.
  - Let A be an attribute of dataset R.
    - Values of A are drawn from finite domain $[u] = \{1, \cdots, u\}$.
  - Define for each $x \in \{1, \cdots, u\}$, $\mathbf{v}(x) = \{count(R.A = x)\}$.

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.
  - Let A be an attribute of dataset R.
    - Values of A are drawn from finite domain $[u] = \{1, \cdots, u\}$.
  - Define for each $x \in \{1, \cdots, u\}$, $\mathbf{v}(x) = \{count(R.A = x)\}$.
  - Define frequency vector $\mathbf{v}$ of $R.A$ as $\mathbf{v} = (\mathbf{v}(1), \ldots, \mathbf{v}(u))$.

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.
  - Let A be an attribute of dataset R.
    - Values of A are drawn from finite domain $[u] = \{1, \cdots, u\}$.
  - Define for each $x \in \{1, \cdots, u\}$, $\mathbf{v}(x) = \{count(R.A = x)\}$.
  - Define frequency vector $\mathbf{v}$ of $R.A$ as $\mathbf{v} = (\mathbf{v}(1), \ldots, \mathbf{v}(u))$.
  - A *histogram* over $R.A$ is any compact (lossy) representation of $\mathbf{v}$.

Introduction: Histograms

- A widely accepted and utilized summarization tool is the Histogram.
  - Let A be an attribute of dataset R.
    - Values of A are drawn from finite domain $[u] = \{1, \cdots, u\}$.
  - Define for each $x \in \{1, \cdots, u\}$, $\mathbf{v}(x) = \{count(R.A = x)\}$.
  - Define frequency vector $\mathbf{v}$ of $R.A$ as $\mathbf{v} = (\mathbf{v}(1), \ldots, \mathbf{v}(u))$.
  - A *histogram* over $R.A$ is any compact (lossy) representation of $\mathbf{v}$.

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮

↓

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

↓

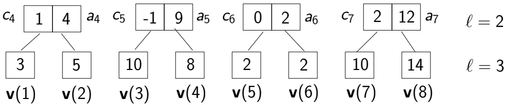
3	5	10	8	2	2	10	14	$\ell = 3$
$\mathbf{v}(1)$	$\mathbf{v}(2)$	$\mathbf{v}(3)$	$\mathbf{v}(4)$	$\mathbf{v}(5)$	$\mathbf{v}(6)$	$\mathbf{v}(7)$	$\mathbf{v}(8)$	

Original data signal at level $\ell = \log_2 u$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



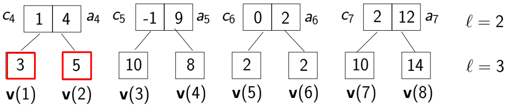
Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 2$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

↓



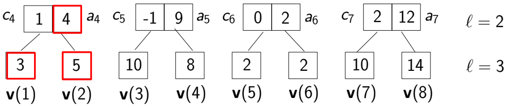
Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 2$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

$a_4 = (\mathbf{v}(2) + \mathbf{v}(1))/2$



Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 2$.

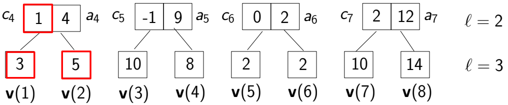
Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

↓

$c_4 = (\mathbf{v}(2) - \mathbf{v}(1))/2$

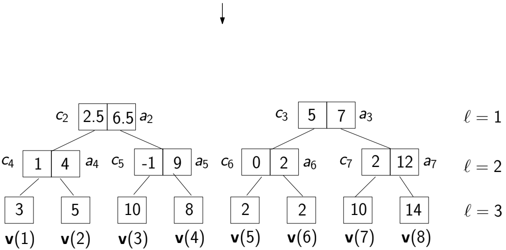


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 2$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

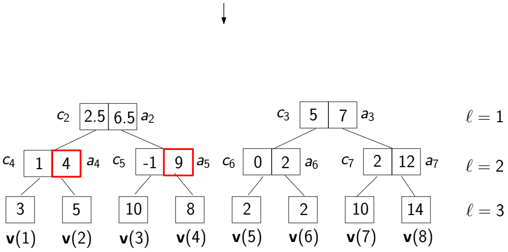


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 1$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



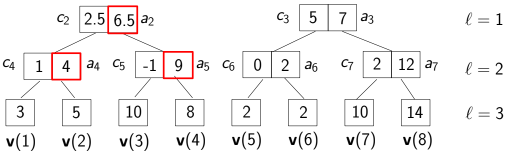
Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 1$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

$a_2 = (a_5 + a_4)/2$



Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 1$.

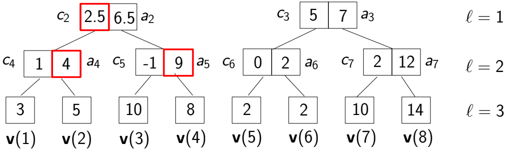
Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

↓

$c_2 = (a_5 - a_4)/2$

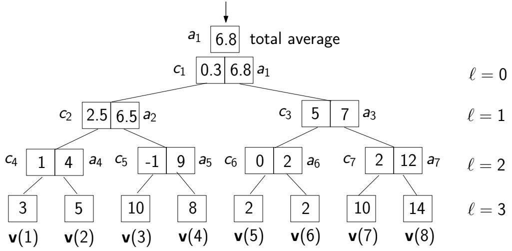


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 1$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

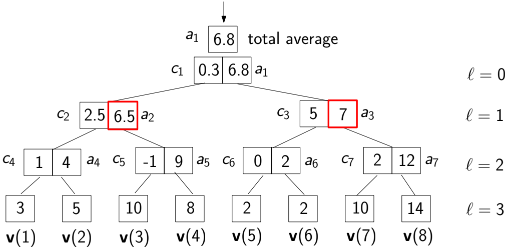


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 0$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

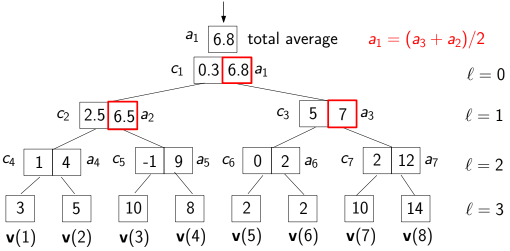


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 0$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

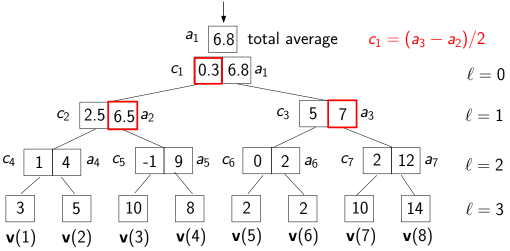


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 0$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

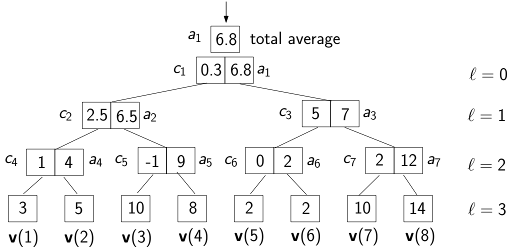


Compute *detail coefficients* c_i and *average coefficients* a_i for level $\ell = 0$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

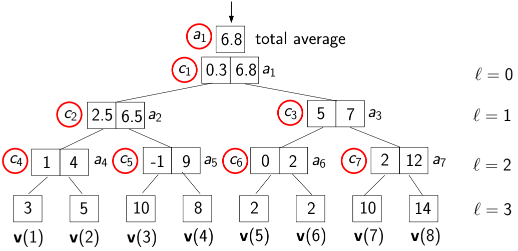


The *wavelet coefficients* w_i
are $[\ a_1, c_1, \ldots, c_{u-1}\ ]$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

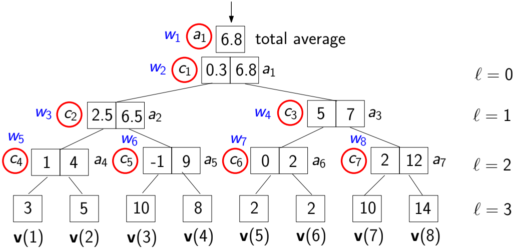


The *wavelet coefficients* w_i
are $[\ a_1, c_1, \ldots, c_{u-1}\ ]$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

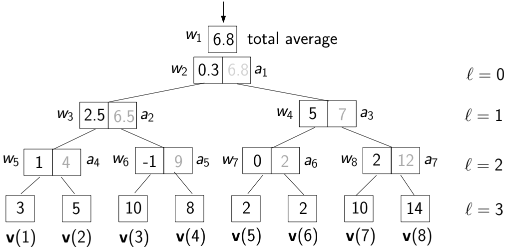


The *wavelet coefficients* w_i
are [$a_1, c_1, \ldots, c_{u-1}$].

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



One scales w_i by $\sqrt{u/2^\ell}$ to preserve energy, i.e.
$\|\mathbf{v}\|_2^2 = \Sigma_{i=1}^{u} \mathbf{v}(i)^2 = \Sigma_{i=1}^{u} w_i^2$

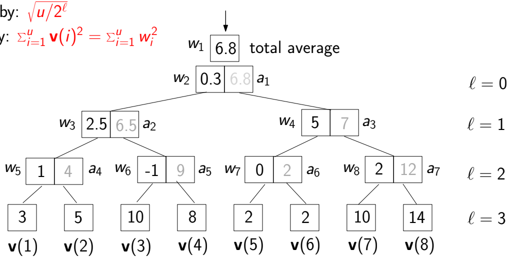
Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

scale by: $\sqrt{u/2^\ell}$

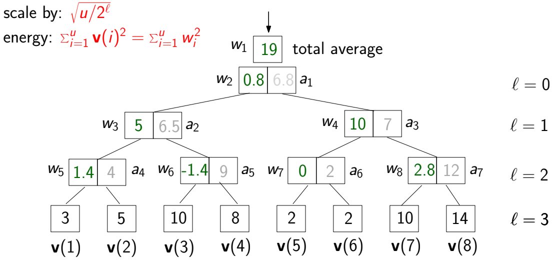
energy: $\Sigma_{i=1}^{u} \mathbf{v}(i)^2 = \Sigma_{i=1}^{u} w_i^2$



One scales w_i by $\sqrt{u/2^\ell}$ to preserve energy, i.e.
$\|\mathbf{v}\|_2^2 = \Sigma_{i=1}^{u} \mathbf{v}(i)^2 = \Sigma_{i=1}^{u} w_i^2$

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



One scales w_i by $\sqrt{u/2^\ell}$ to preserve energy, i.e.
$\|\mathbf{v}\|_2^2 = \Sigma_{i=1}^{u} \mathbf{v}(i)^2 = \Sigma_{i=1}^{u} w_i^2$

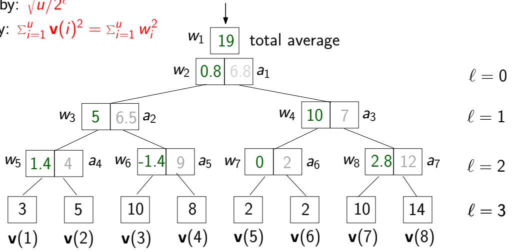
Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

scale by: $\sqrt{u/2^\ell}$

energy: $\Sigma_{i=1}^{u} \mathbf{v}(i)^2 = \Sigma_{i=1}^{u} w_i^2$

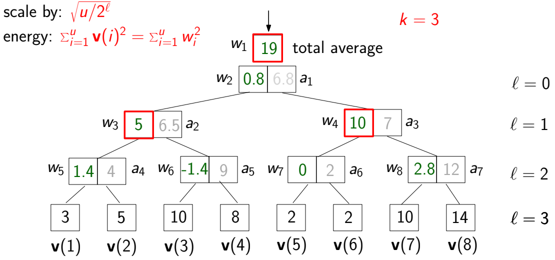


Select top-k w_i in the *absolute value* to obtain best k-term representation minimizing error in energy, i.e. minimize $\Sigma_{i=1}^{u} \mathbf{v}(i)^2 - \Sigma_{i=1}^{u} w_i^2$

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

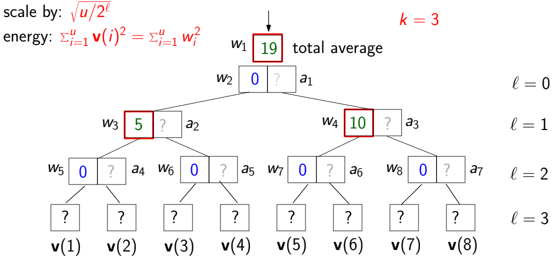


Select top-k w_i in the *absolute value* to obtain best k-term representation minimizing error in energy, i.e. minimize $\Sigma_{i=1}^{u} \mathbf{v}(i)^2 - \Sigma_{i=1}^{u} w_i^2$

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

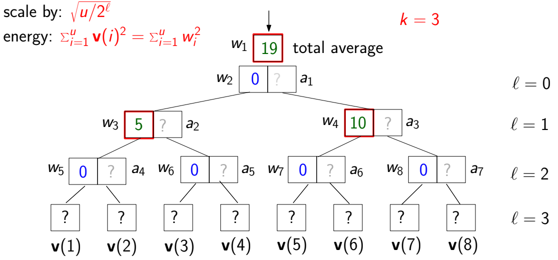


We maintain the best k-term w_i.
Other w_i are treated as 0.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

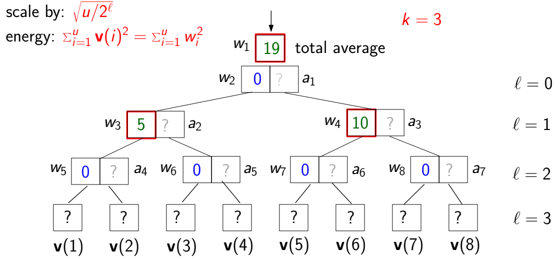


The error in energy is
$\Sigma_{i=1}^{u} \mathbf{v}(i)^2 - \Sigma_{i=1}^{u} w_i^2 = 502 - 489.5 = 12.5$.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

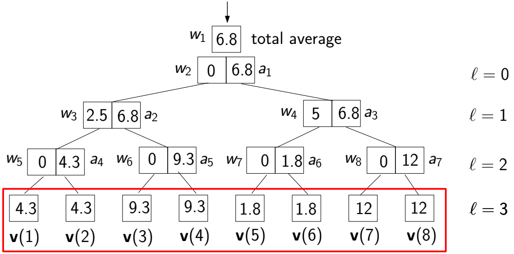


To reconstruct the original signal we compute the *average* and *difference coefficients* in reverse, i.e. top to bottom.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

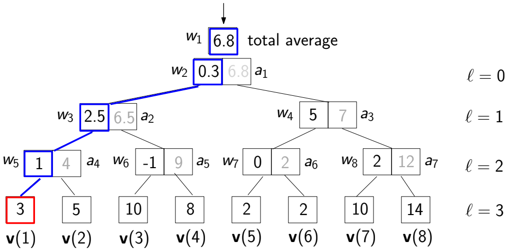


The reconstructed signal is a reasonably close approximation.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

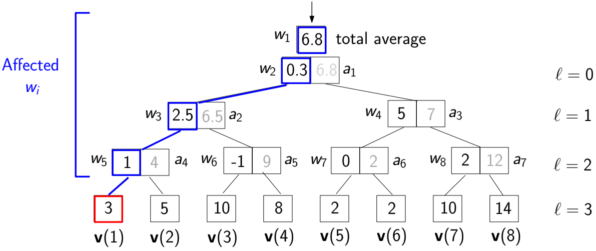
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

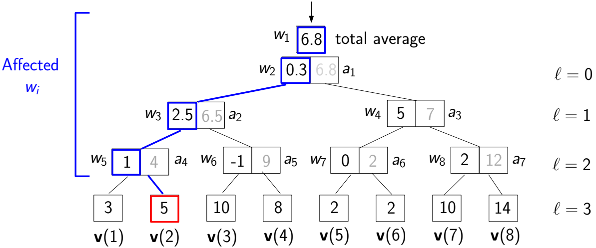
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

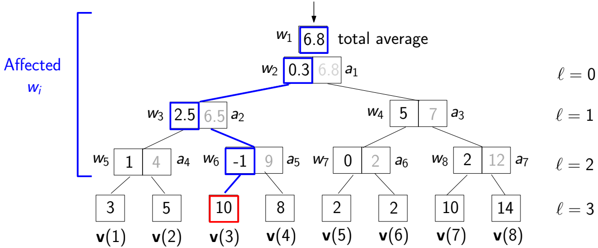
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

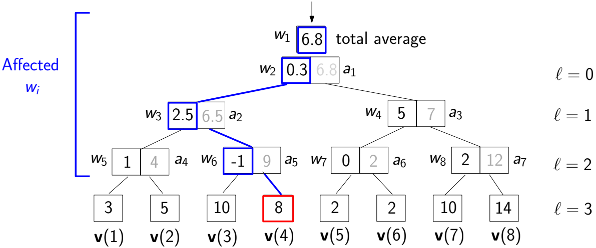
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

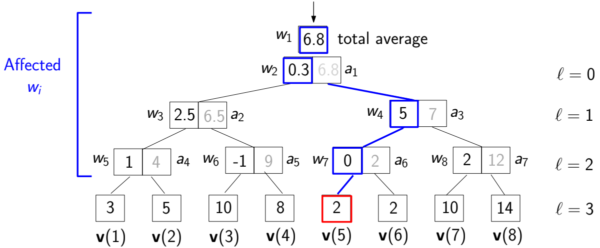
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

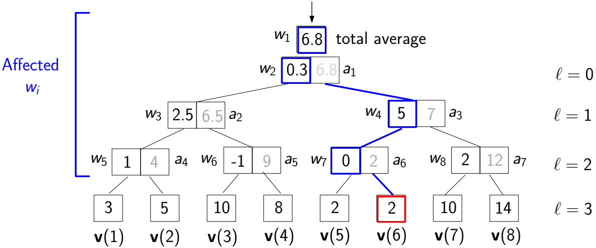
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

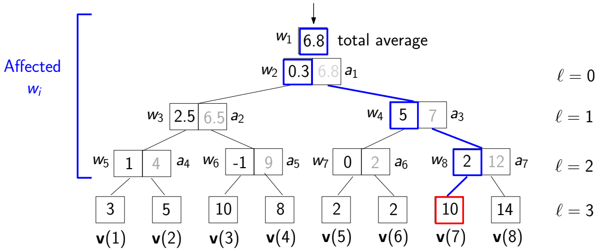
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

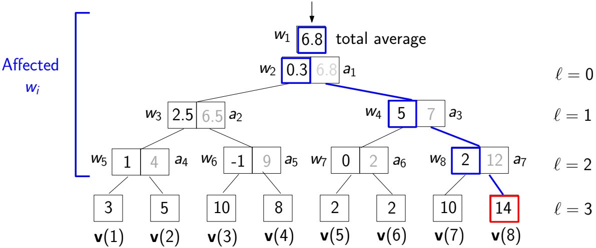
1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients w_i recursively as follows:

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



All w_i can be calculated in $O(u \log u)$ time:

1. We maintain $O(\log u)$ partial w_is at a time.
2. Compute affected w_i and contribution from each $\mathbf{v}(x)$ in $O(\log u)$ time.
2. Process $\mathbf{v}(x)$s in sorted order. [GKMS01]

[GKMS01] A.C. Gilbert, et al. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

Introduction: Histograms

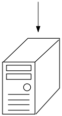
- We may also compute w_i with the wavelet basis vectors ψ_i.
  - $w_i = \mathbf{v} \cdot \psi_i$ for $i = 1, \ldots, u$

Outline

Introduction: MapReduce and Hadoop

R

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮

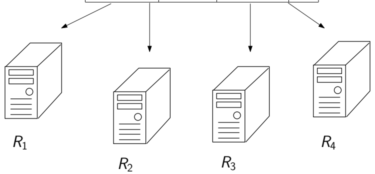


- Traditionally data is stored in a centralized setting.

Introduction: MapReduce and Hadoop

R

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮



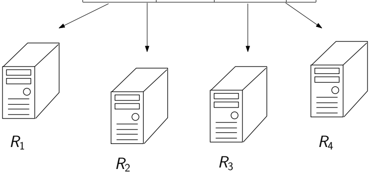
$$R = R_1 \cup R_2 \cup R_3 \cup R_4$$

- Traditionally data is stored in a centralized setting.
- Now stored data has sky rocketed, and is increasingly distributed.

Introduction: MapReduce and Hadoop

R

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮



$$R = R_1 \cup R_2 \cup R_3 \cup R_4$$

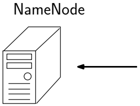
- Traditionally data is stored in a centralized setting.
- Now stored data has sky rocketed, and is increasingly distributed.
- We study computing the top-k coefficients efficiently on distributed data in MapReduce using Hadoop to illustrate our ideas.

Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).

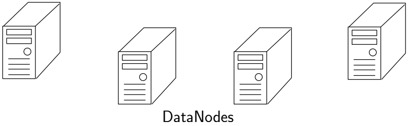
Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).



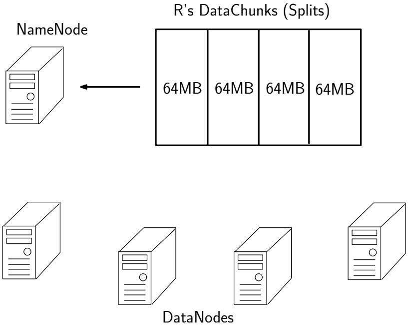
R

Record ID	User ID	Object ID	...
1	1	12872	...
2	8	19832	...
3	4	231	...
⋮	⋮	⋮	⋮



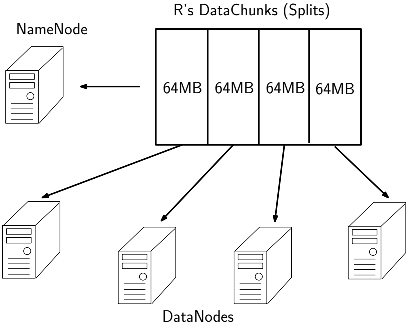
Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).



Background: Hadoop Distributed File System (HDFS)

- Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).



Background: Hadoop Core

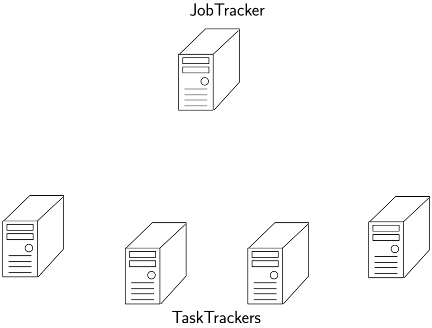
- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.

Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.

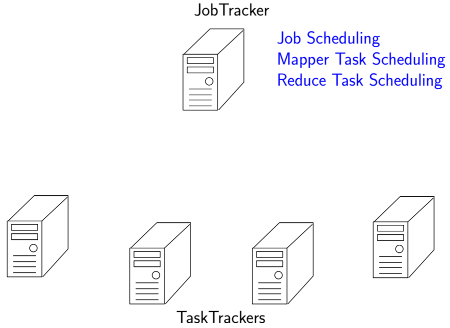
Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.



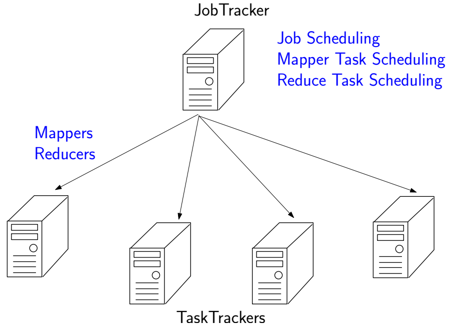
Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.



Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.



Background: Hadoop Cluster

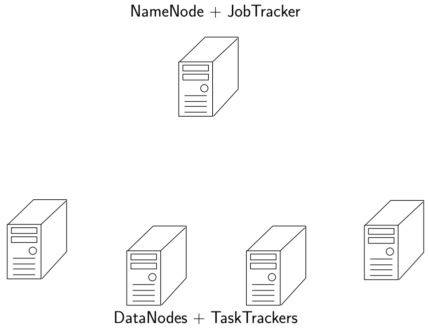
- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.

Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.
- The other machines run DataNode and TaskTracker tasks and are called *slaves*.

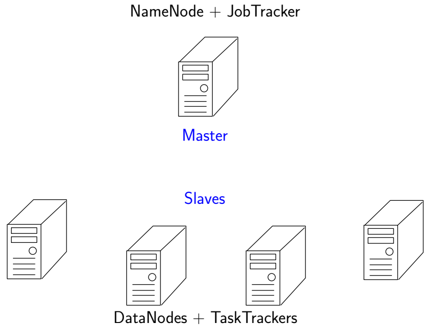
Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.
- The other machines run DataNode and TaskTracker tasks and are called *slaves*.

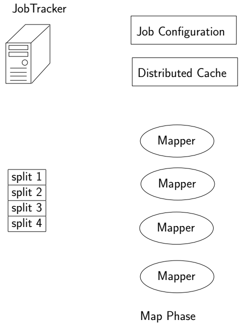


Background: Hadoop Cluster

- In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.
- The other machines run DataNode and TaskTracker tasks and are called *slaves*.

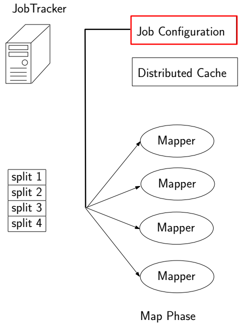


Background: MapReduce Job Overview



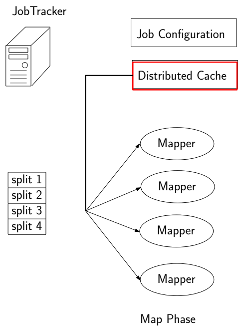
- Next we look at an overview of a typical MapReduce Job.

Background: MapReduce Job Overview



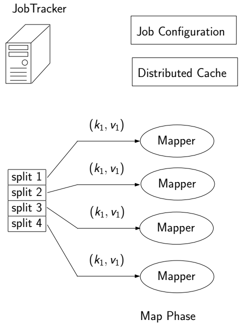
- Job specific variables are first placed in the *Job Configuration* which is sent to each *Mapper Task* by the *JobTracker*.

Background: MapReduce Job Overview



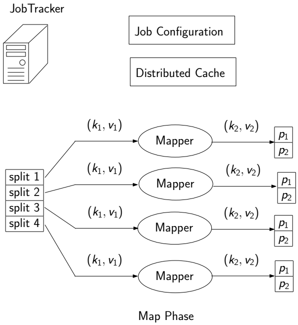
- Large data such as files or libraries are then put in the *Distributed Cache* which is copied to each *TaskTracker* by the *JobTracker*.

Background: MapReduce Job Overview



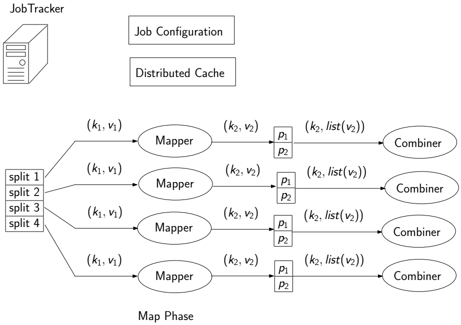
- The JobTracker next assigns each *InputSplit* to a *Mapper* task on a TaskTracker, we assume m Mappers and m InputSplits.

Background: MapReduce Job Overview



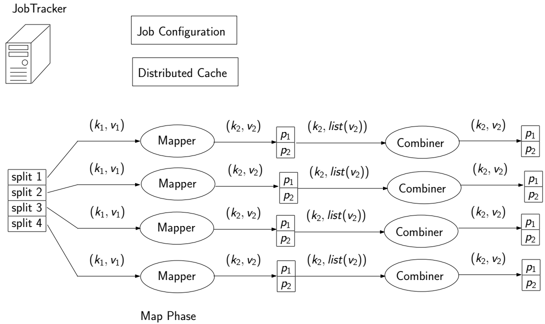
- Each Mapper maps a (k_1, v_1) pair to an intermediate (k_2, v_2) pair and partitions by k_2, i.e. $hash(k_2) = p_i$ for $i \in [1, r]$, $r = |reducers|$.

Background: MapReduce Job Overview



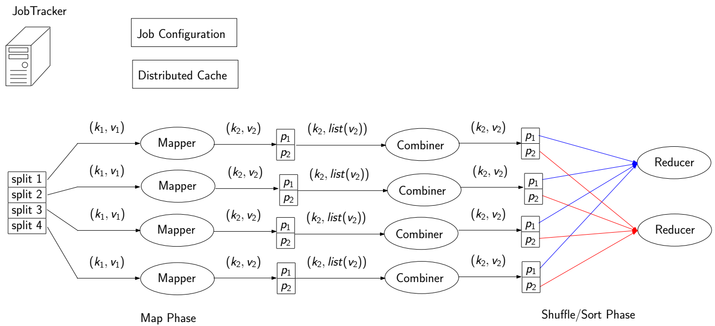
- An optional *Combiner* is executed over $(k_2, list(v_2))$.

Background: MapReduce Job Overview



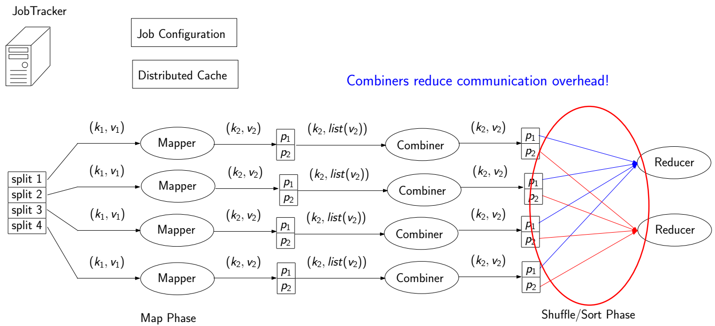
- The *Combiner* aggregates v_2 for a k_2 and a (k_2, v_2) is written to a partition on disk.

Background: MapReduce Job Overview



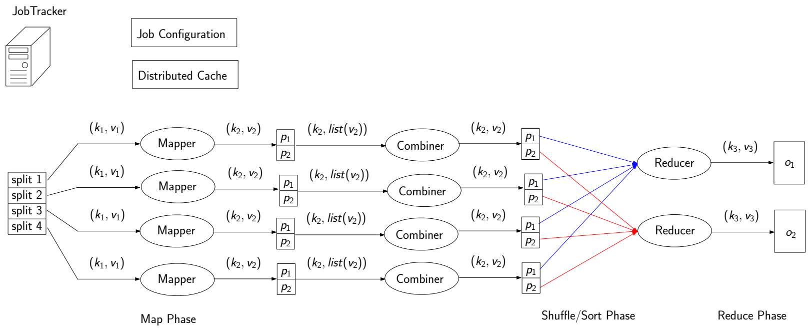
- The JobTracker assigns two TaskTrackers to run the Reducers, each Reducer copies and sorts it's inputs from corresponding partitions.

Background: MapReduce Job Overview



- The JobTracker assigns two TaskTrackers to run the Reducers, each Reducer copies and sorts it's inputs from corresponding partitions.

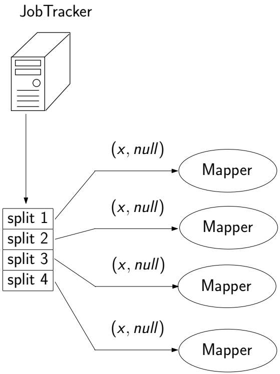
Background: MapReduce Job Overview



- Each Reducer reduces a $(k_2, list(v_2))$ to a single (k_3, v_3) and writes the results to a DFS file, o_i for $i \in [1, r]$.

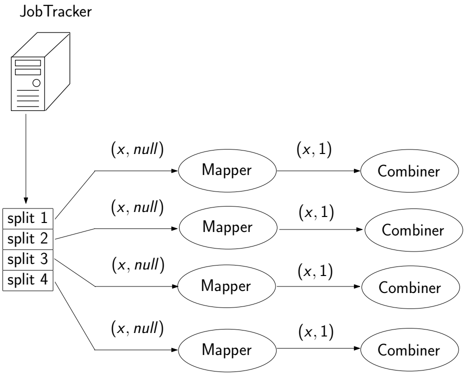
Outline

Exact Top-k Wavelet Coefficients: Naive Solution



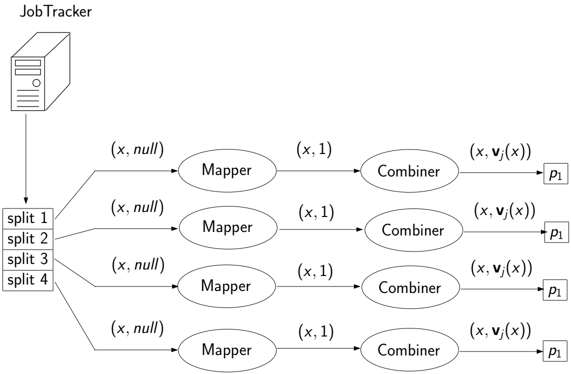
- Each of the m Mappers reads the input key x from its input split.

Exact Top-k Wavelet Coefficients: Naive Solution



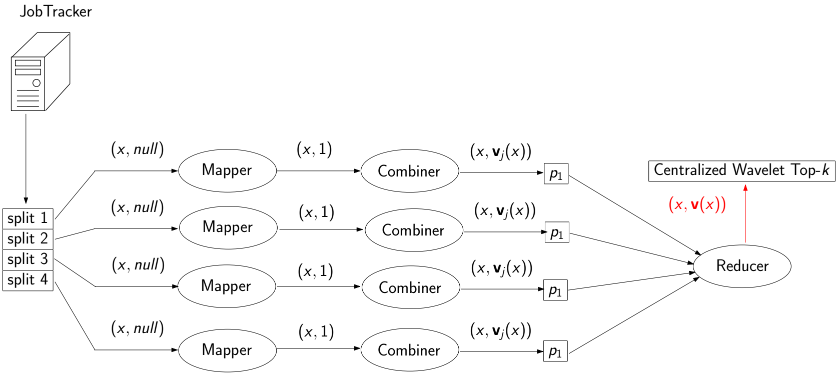
- Each Mapper emits $(x, 1)$ for combining by the Combiner.

Exact Top-k Wavelet Coefficients: Naive Solution



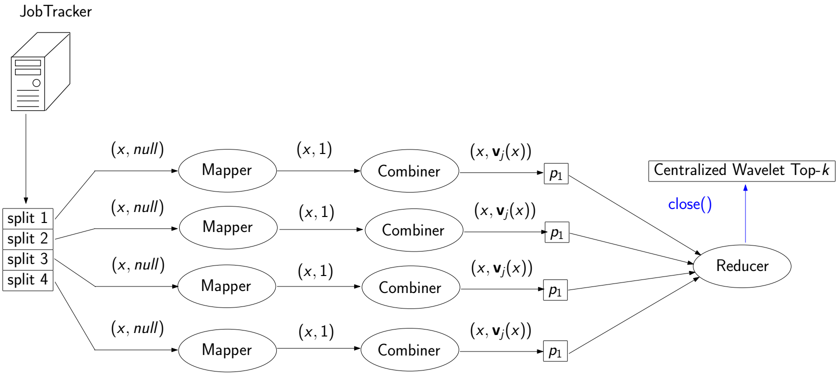
- Each Combiner emits $(x, v_j(x))$, where $v_j(x)$ is the local frequency of x.

Exact Top-k Wavelet Coefficients: Naive Solution



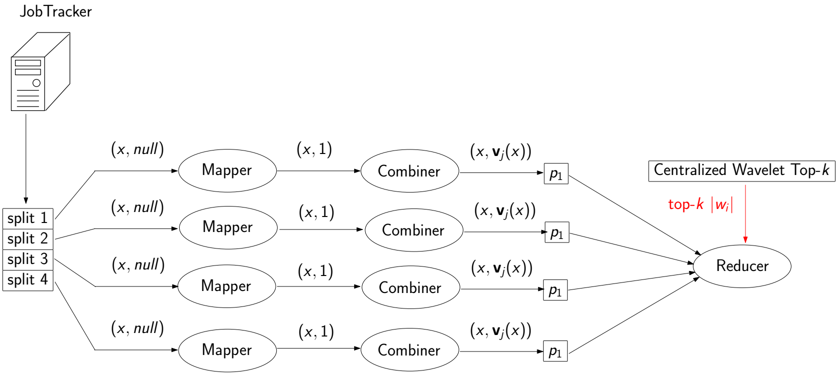
- The Reducer utilizes a Centralized Wavelet Top-k algorithm, supplying the $(x, v(x))$ in a streaming fashion.

Exact Top-k Wavelet Coefficients: Naive Solution



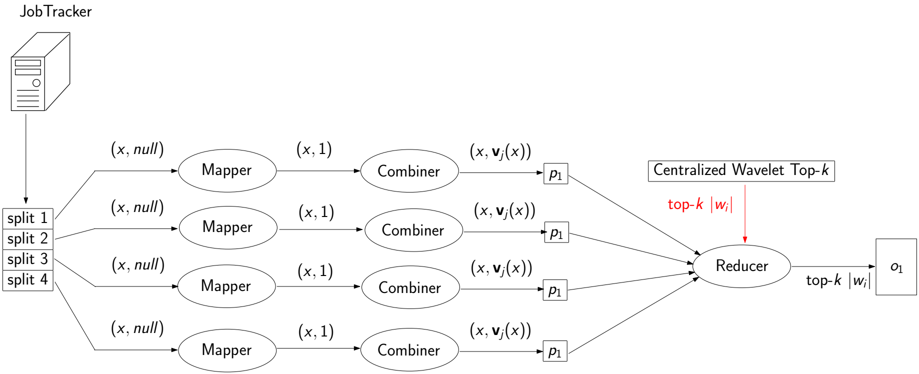
- At the end of the Reduce phase, the Reducer's *close()* method is invoked. The Reducer then requests the top-k $|w_i|$.

Exact Top-k Wavelet Coefficients: Naive Solution



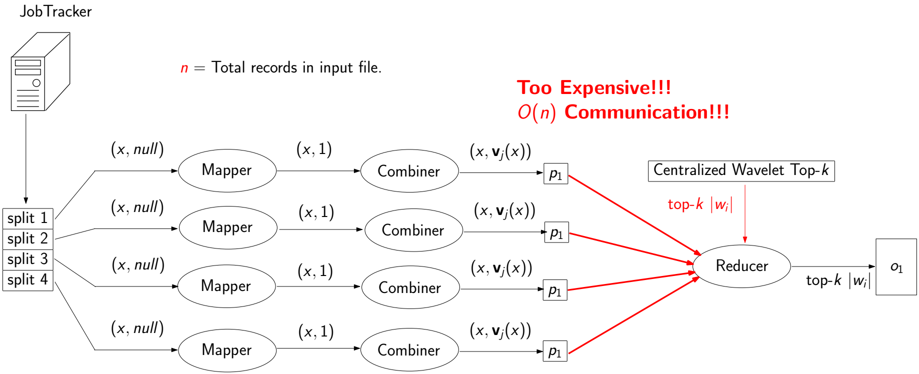
- The centralized algorithm computes the top-k $|w_i|$ and returns these to the Reducer.

Exact Top-k Wavelet Coefficients: Naive Solution



- Finally, the Reducer writes the top-k $|w_i|$ to its HDFS output file o_1.

Exact Top-k Wavelet Coefficients: Naive Solution



Outline

Exact Top-k Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-k: $w_i = \mathbf{v} \cdot \psi_i = (\sum_{j=1}^{m} \mathbf{v}_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}$.

Exact Top-k Wavelet Coefficients: Our Solution

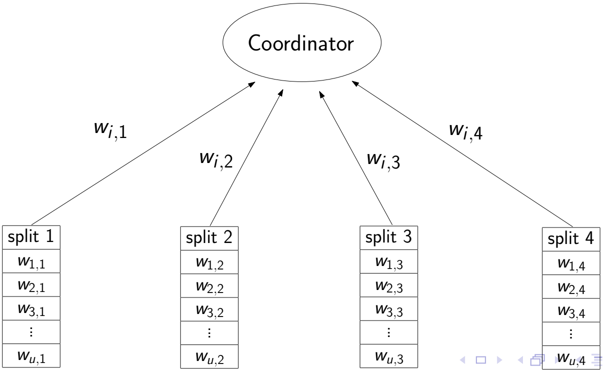
- We can try to model the problem as a distributed top-k: $w_i = \mathbf{v} \cdot \psi_i = \left(\sum_{j=1}^{m} \mathbf{v}_j\right) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}$.

$w_{i,j}$ is the local value of w_i in split j.

split 1	split 2	split 3	split 4
$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,4}$
$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$
$w_{3,1}$	$w_{3,2}$	$w_{3,3}$	$w_{3,4}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w_{u,1}$	$w_{u,2}$	$w_{u,3}$	$w_{u,4}$

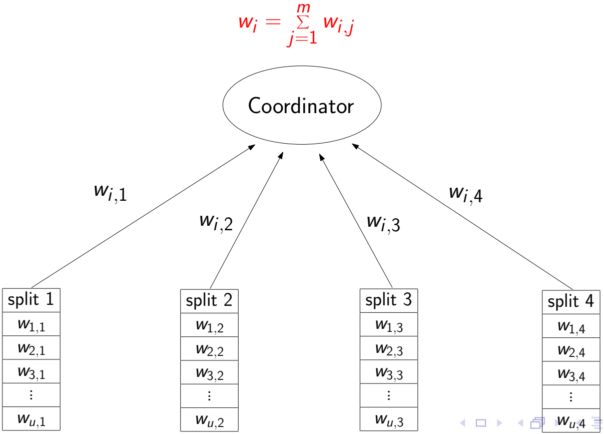
Exact Top-k Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-k: $w_i = \mathbf{v} \cdot \psi_i = \left(\sum_{j=1}^{m} \mathbf{v}_j\right) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}$.



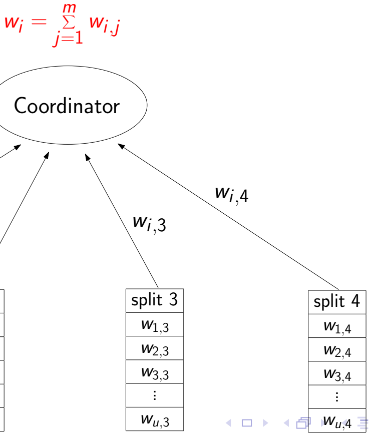
Exact Top-k Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-k: $w_i = \mathbf{v} \cdot \psi_i = (\sum_{j=1}^{m} \mathbf{v}_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}$.



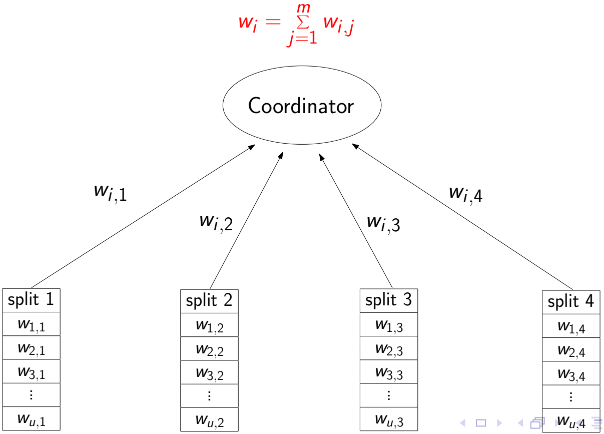
Exact Top-k Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-k: $w_i = \mathbf{v} \cdot \psi_i = (\sum_{j=1}^{m} \mathbf{v}_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}$.
- Previous solutions assume local score $s_{i,j} \geq 0$ and want the largest $s_i = \sum_{j=1}^{m} s_{i,j}$.



Exact Top-k Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-k: $w_i = \mathbf{v} \cdot \psi_i = (\sum_{j=1}^{m} \mathbf{v}_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}$.
- Previous solutions assume local score $s_{i,j} \geq 0$ and want the largest $s_i = \sum_{j=1}^{m} s_{i,j}$.
- We have $w_{i,j} < 0$ and $w_{i,j} \geq 0$ and want the largest $|w_i|$.



Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$

- An item x has a local score $s_i(x)$ at node i $\forall i \in [1 \dots m]$, where if x does not appear $s_i(x) = 0$

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$

- Each node sends:
  the top-k most positive scored items
  the top-k most negative scored items.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- The coordinator computes useful bounds for each received item.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\hat{s}(x)$ denotes the partial score sum for x

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\hat{s}(x)$ denotes the partial score sum for x

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- F_x is a receipt indication bit vector,
  if $s_i(x)$ is received $F_x(i) = 1$,
  else $F_x(i) = 0$.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- F_x is a receipt indication bit vector,
  if $s_i(x)$ is received $F_x(i) = 1$,
  else $F_x(i) = 0$.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\tau^+(x)$ is an upper bound on the total score $s(x)$,
  if $s_i(x)$ received then $\tau^+(x) = \tau^+(x) + s_i(x)$
  else $\tau^+(x) = \tau^+(x) + k$'th most positive from node i

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\tau^+(x)$ is an upper bound on the total score $s(x)$,
  if $s_i(x)$ received then $\tau^+(x) = \tau^+(x) + s_i(x)$
  else $\tau^+(x) = \tau^+(x) + k$'th most positive from node i

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\tau^-(x)$ is a lower bound on the total score sum $s(x)$,
  if $s_i(x)$ received then $\tau^-(x) = \tau^-(x) + s_i(x)$
  else $\tau^-(x) = \tau^-(x) + k$'th most negative score from node i

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\tau^-(x)$ is a lower bound on the total score sum $s(x)$,
  if $s_i(x)$ received then $\tau^-(x) = \tau^-(x) + s_i(x)$
  else $\tau^-(x) = \tau^-(x) + k$'th most negative score from node i

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\tau(x)$ is a lower bound on $|s(x)|$ computed as,
  $\tau(x) = 0$ if $sign(\tau^+(x)) \neq sign(\tau^-(x))$
  $\tau(x) = \min(|\tau^+(x)|, |\tau^-(x)|)$ otherwise.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

- $\tau(x)$ is a lower bound on $|s(x)|$ computed as,
  $\tau(x) = 0$ if $sign(\tau^+(x)) \neq sign(\tau^-(x))$
  $\tau(x) = \min(|\tau^+(x)|, |\tau^-(x)|)$ otherwise.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22,\ T_1/m = 22/3$

- We select the item with the kth largest $\tau(x)$.
  $\tau(x)$ is a lower bound T_1 on the top-k $|s(x)|$ for unseen item x.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22,\ T_1/m = 22/3$

- Any unseen item x must have at least:
  one $s_i(x) > T_1/m$ or
  one $s_i(x) < -T_1/m$
  To get into the top-k.

node 1		
id	x	$s_1(x)$
$e_{1,1}$	5	20
$e_{1,2}$	2	7
$e_{1,3}$	1	6
$e_{1,4}$	4	-2
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30

node 2		
id	x	$s_2(x)$
$e_{2,1}$	5	12
$e_{2,2}$	4	7
$e_{2,3}$	1	2
$e_{2,4}$	2	-5
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20

node 3		
id	x	$s_3(x)$
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,3}$	4	5
$e_{3,4}$	2	-3
$e_{3,5}$	5	-6
$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22,\ T_1/m = 22/3$

Round 1 End

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22, \; T_1/m = 22/3$

- T_1/m sent to each site.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22, \; T_1/m = 22/3$

- Each site finds items with
  $s_i(x) > T_1/m$ or
  $s_i(x) < T_1/m$.

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22, \; T_1/m = 22/3$

- Items with $|s_i(x)| > T_1/m$ are sent to coordinator.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R					
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22,\ T_1/m = 22/3$

- Items with $|s_i(x)| > T_1/m$ are sent to coordinator.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

→

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22, \ T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- Partial score sum $s(5) = 20 + 12$

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22, \; T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- Receipt vector $F_5 = [110]$

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22$, $T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau^+(x)$ is now tighter,
  if $s_i(x)$ received then $\tau^+(x) = \tau^+(x) + s_i(x)$
  else $\tau^+(x) = \tau^+(x) + T_1/m$

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22$, $T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau^-(x)$ is also tighter,
  if $s_i(x)$ received then $\tau^-(x) = \tau^-(x) + s_i(x)$
  else $\tau^-(x) = \tau^-(x) - T_1/m$

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- Score absolute value bound $\tau(5) = \min(39.3, 24.6)$.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau'(x)$ is an upper bound on $|s(x)|$, $\tau'(x) = \max\{|\tau^+(x)|, |\tau^-(x)|\}$

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22, \ T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau'(x)$ is an upper bound on $|s(x)|$, $\tau'(x) = \max\{|\tau^+(x)|, |\tau^-(x)|\}$

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22, \ T_1/m = 22/3$

- We select the item x with the kth largest $\tau(x)$, which serves as a new lower bound T_2 on $|s(x)|$ for any item.

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

- We select the item x with the kth largest $\tau(x)$, which serves as a new lower bound T_2 on $|s(x)|$ for any item.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

- Any item with $\tau'(x) < T_2$ cannot be in the top-k and is pruned from R.

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

- Any remaining items with a 0 in vector F_x are selected.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

Round 2 End

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

- The coordinator asks for missing scores for items still in R.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,6}$	6	-10

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22, \; T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,6}$	6	-10

→

R						
x	$\hat{s}(x)$	F_x	$\tau^+(x)$	$\tau^-(x)$	$\tau(x)$	$\tau'(x)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,6}$	6	-10

R		
x	$\hat{s}(x)$	F_i
~~1~~	~~10~~	~~001~~
3	-38	111
~~5~~	~~32~~	~~110~~
6	-45	111

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

- After collecting all scores, the coordinator can determine top-k $|s(x)|$.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	**3**	**6**
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,6}$	6	-10

R		
x	$\hat{s}(x)$	F_i
~~1~~	~~10~~	~~001~~
3	-38	111
~~5~~	~~32~~	~~110~~
6	-45	111

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

- After collecting all scores, the coordinator can determine top-k $|s(x)|$.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,6}$	6	-10

R		
x	$\hat{s}(x)$	F_i
~~1~~	~~10~~	~~001~~
3	-38	111
~~5~~	~~32~~	~~110~~
6	-45	111

$T_1 = 22,\ T_1/m = 22/3$

$T_2 = 45$

$\star$ $\mathbf{s(6) = -45}$

$s_3(3) = ?$

- After collecting all scores, the coordinator can determine top-k $|s(x)|$.

	node 1		
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

	node 2		
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

	node 3		
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Exact Top-k Wavelet Coefficients: Our Solution

$k = 1$

R		
id	x	$s_j(x)$
$e_{1,1}$	5	20
$e_{1,5}$	6	-15
$e_{1,6}$	3	-30
$e_{2,1}$	5	12
$e_{2,5}$	3	-14
$e_{2,6}$	6	-20
$e_{3,1}$	1	10
$e_{3,2}$	3	6
$e_{3,6}$	6	-10

R		
x	$\hat{s}(x)$	F_i
~~1~~	~~10~~	~~001~~
3	-38	111
~~5~~	~~32~~	~~110~~
6	-45	111

$T_1 = 22, \; T_1/m = 22/3$

$T_2 = 45$

$\star$ $\mathbf{s(6) = -45}$

Round 3 End

node 1			
	id	x	$s_1(x)$
√	$e_{1,1}$	5	20
	$e_{1,2}$	2	7
	$e_{1,3}$	1	6
	$e_{1,4}$	4	-2
√	$e_{1,5}$	6	-15
√	$e_{1,6}$	3	-30

node 2			
	id	x	$s_2(x)$
√	$e_{2,1}$	5	12
	$e_{2,2}$	4	7
	$e_{2,3}$	1	2
	$e_{2,4}$	2	-5
√	$e_{2,5}$	3	-14
√	$e_{2,6}$	6	-20

node 3			
	id	x	$s_3(x)$
√	$e_{3,1}$	1	10
	$e_{3,2}$	3	6
	$e_{3,3}$	4	5
	$e_{3,4}$	2	-3
	$e_{3,5}$	5	-6
√	$e_{3,6}$	6	-10

Outline

Approximate Top-k Wavelet Coefficients

- Hadoop Wavelet Top-k is a good solution if the exact top-k $|w_i|$ must be retrieved, but requires multiple phases.

Approximate Top-k Wavelet Coefficients

- Hadoop Wavelet Top-k is a good solution if the exact top-k $|w_i|$ must be retrieved, but requires multiple phases.
- If we are allowed an approximation, we could further improve:

Approximate Top-k Wavelet Coefficients

- Hadoop Wavelet Top-k is a good solution if the exact top-k $|w_i|$ must be retrieved, but requires multiple phases.
- If we are allowed an approximation, we could further improve:
  1. communication cost

Approximate Top-k Wavelet Coefficients

- Hadoop Wavelet Top-k is a good solution if the exact top-k $|w_i|$ must be retrieved, but requires multiple phases.
- If we are allowed an approximation, we could further improve:
  1. communication cost
  2. number of MapReduce rounds

Approximate Top-k Wavelet Coefficients

- Hadoop Wavelet Top-k is a good solution if the exact top-k $|w_i|$ must be retrieved, but requires multiple phases.
- If we are allowed an approximation, we could further improve:
    1. communication cost
    2. number of MapReduce rounds
    3. amount of I/O incurred

Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:

Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:
  1. Approximate distributed top-k.

Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:
  1. Approximate distributed top-k.
  2. Approximating local coefficients with a linearly combinable sketch.

Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:
  1. Approximate distributed top-k.
  2. Approximating local coefficients with a linearly combinable sketch.
     - For set $A = A_1 \cup A_2$,
       Sketch(A) = Sketch(A_1) *op* Sketch(A_2) for operator *op*.

Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:
  1. Approximate distributed top-k.
  2. Approximating local coefficients with a linearly combinable sketch.
     - For set $A = A_1 \cup A_2$,
       Sketch(A) = Sketch(A_1) *op* Sketch(A_2) for operator *op*.
     - The state of the art *wavelet* sketch is the GCS Sketch [CGS06].

[GCS06] G. Cormode, et al. Fast approximate wavelet tracking on streams. In *EDBT*, 2006.

Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:
  1. Approximate distributed top-k.
  2. Approximating local coefficients with a linearly combinable sketch.
     - For set $A = A_1 \cup A_2$,
       Sketch(A) = Sketch(A_1) *op* Sketch(A_2) for operator *op*.
     - The state of the art *wavelet* sketch is the GCS Sketch [CGS06].
     - The GCS gives us, for $\mathbf{v} = \mathbf{v_1} + \mathbf{v_2}$
       $\mathrm{GCS}(\mathbf{v}) = \mathrm{GCS}(\mathbf{v_1}) + \mathrm{GCS}(\mathbf{v_2})$

[GCS06] G. Cormode, et al. Fast approximate wavelet tracking on streams. In *EDBT*, 2006.

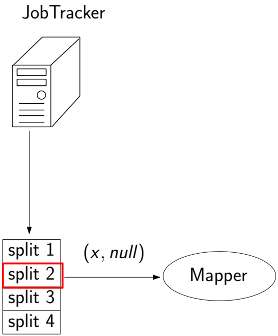
Approximate Top-k Wavelet Coefficients

- Some natural improvement attempts:
  1. Approximate distributed top-k.
  2. Approximating local coefficients with a linearly combinable sketch.
     - For set $A = A_1 \cup A_2$,
       Sketch(A) = Sketch(A_1) *op* Sketch(A_2) for operator *op*.
     - The state of the art *wavelet* sketch is the GCS Sketch [CGS06].
     - The GCS gives us, for $\mathbf{v} = \mathbf{v_1} + \mathbf{v_2}$
       $\text{GCS}(\mathbf{v}) = \text{GCS}(\mathbf{v_1}) + \text{GCS}(\mathbf{v_2})$
  3. Random sampling techniques.

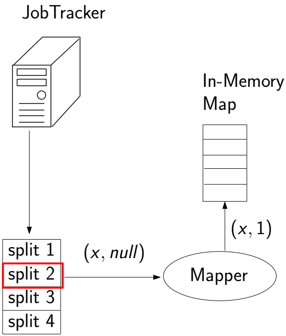
[GCS06] G. Cormode, et al. Fast approximate wavelet tracking on streams. In *EDBT*, 2006.

Outline

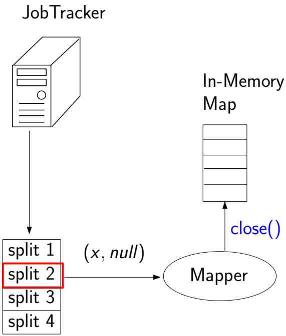
Approximate Top-k Wavelet Coefficients: Sketch



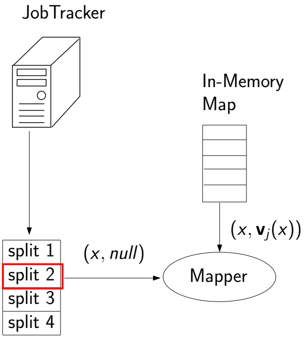
Approximate Top-k Wavelet Coefficients: Sketch



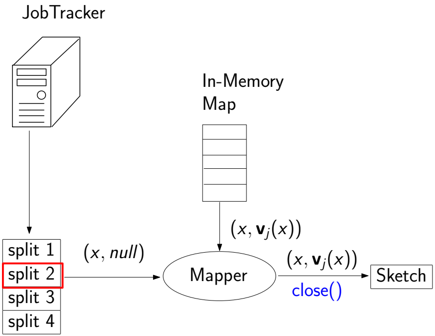
Approximate Top-k Wavelet Coefficients: Sketch



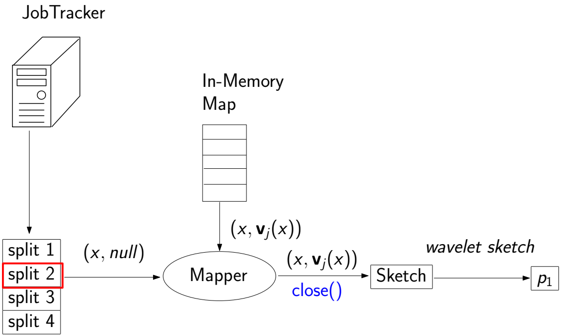
Approximate Top-k Wavelet Coefficients: Sketch



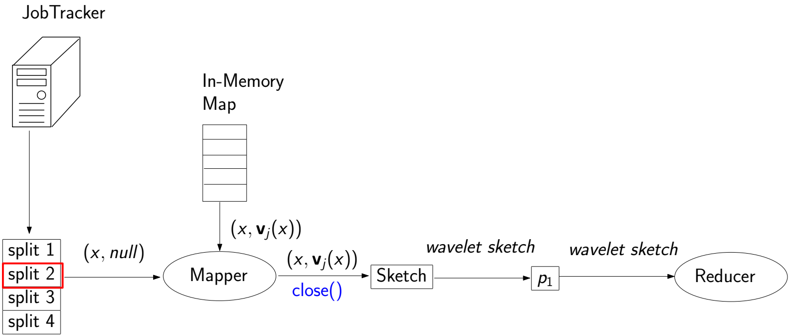
Approximate Top-k Wavelet Coefficients: Sketch



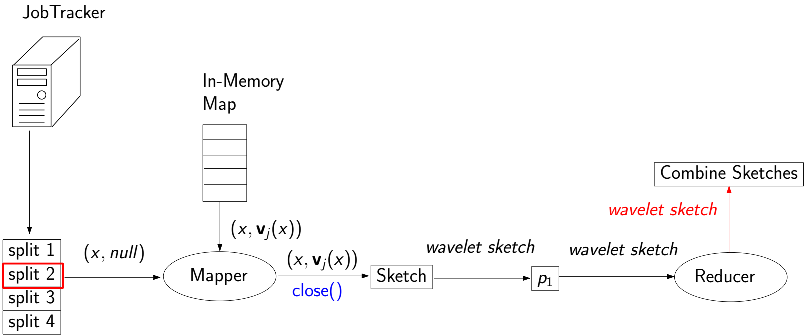
Approximate Top-k Wavelet Coefficients: Sketch



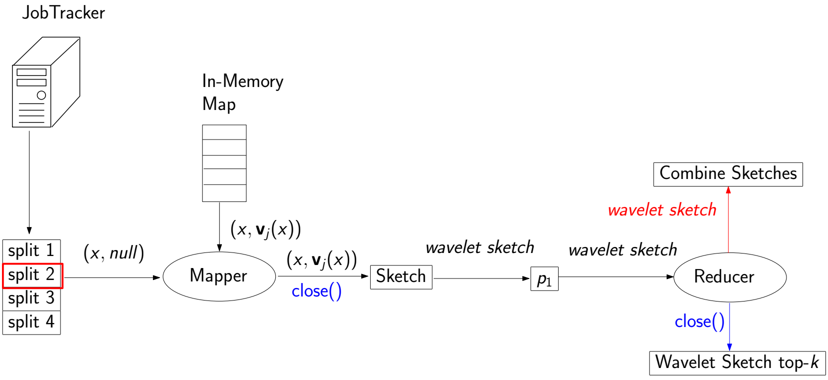
Approximate Top-k Wavelet Coefficients: Sketch



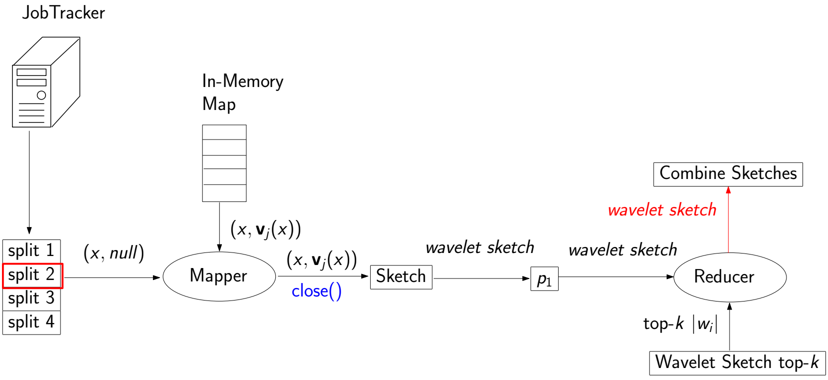
Approximate Top-k Wavelet Coefficients: Sketch



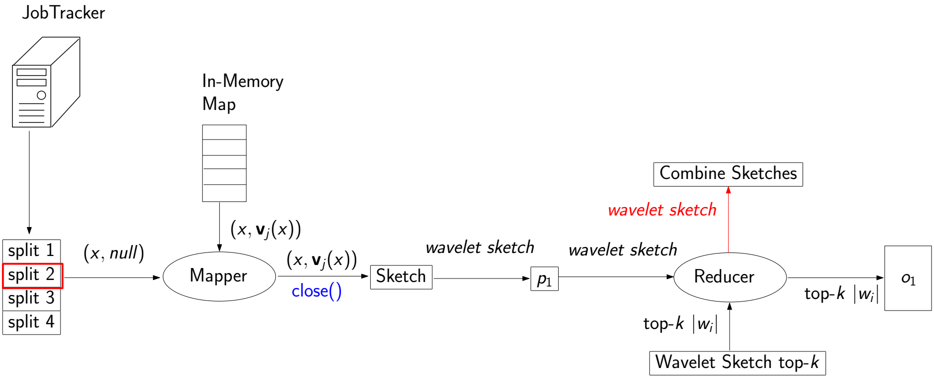
Approximate Top-k Wavelet Coefficients: Sketch



Approximate Top-k Wavelet Coefficients: Sketch

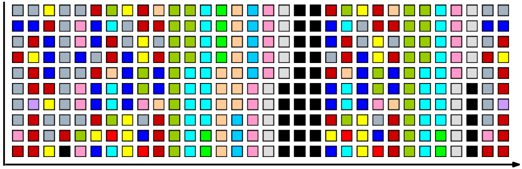


Approximate Top-k Wavelet Coefficients: Sketch



Outline

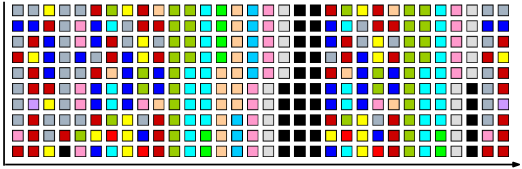
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j Records in split j

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

Well known fact: to approximate each $\mathbf{v}(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.

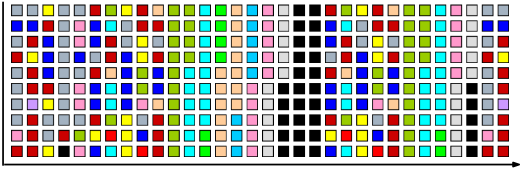


n_j Records in split j

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

Well known fact: to approximate each $\mathbf{v}(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.

Node j samples $t_j = n_j \cdot p$ records where $p = 1/\varepsilon^2 n$.

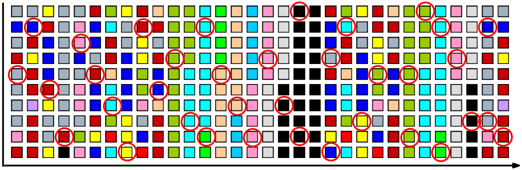


n_j Records in split j

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

Well known fact: to approximate each $\mathbf{v}(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.

Node j samples $t_j = n_j \cdot p$ records where $p = 1/\varepsilon^2 n$.

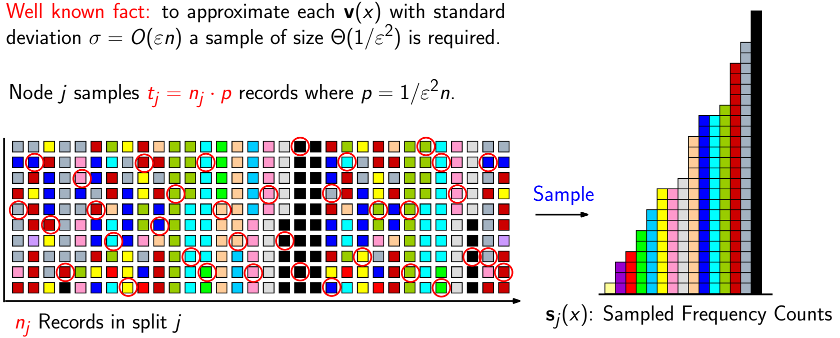


n_j Records in split j

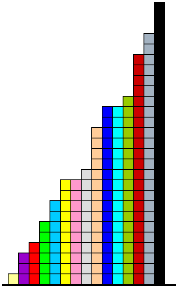
Approximate Top-k Wavelet Coefficients: Basic Random Sampling

Well known fact: to approximate each $\mathbf{v}(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.

Node j samples $t_j = n_j \cdot p$ records where $p = 1/\varepsilon^2 n$.

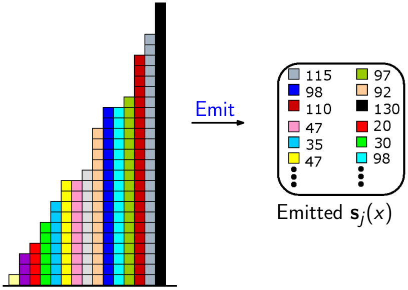


Approximate Top-k Wavelet Coefficients: Basic Random Sampling



$\mathbf{s}_j(x)$: Sampled Frequency Counts

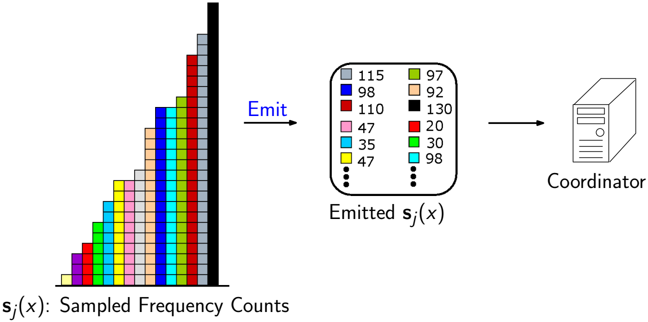
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



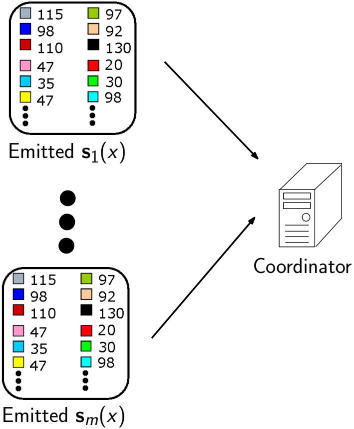
Emitted $\mathbf{s}_j(x)$

$\mathbf{s}_j(x)$: Sampled Frequency Counts

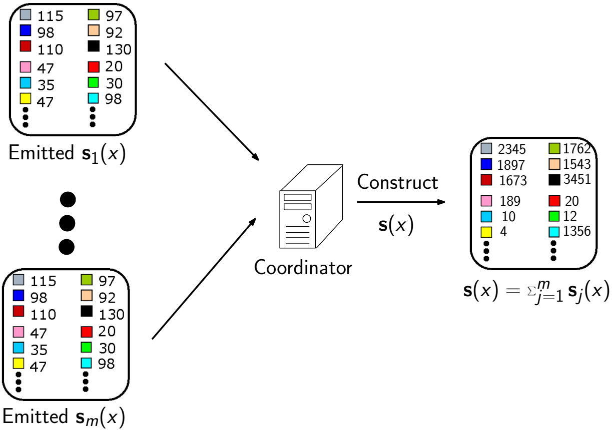
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



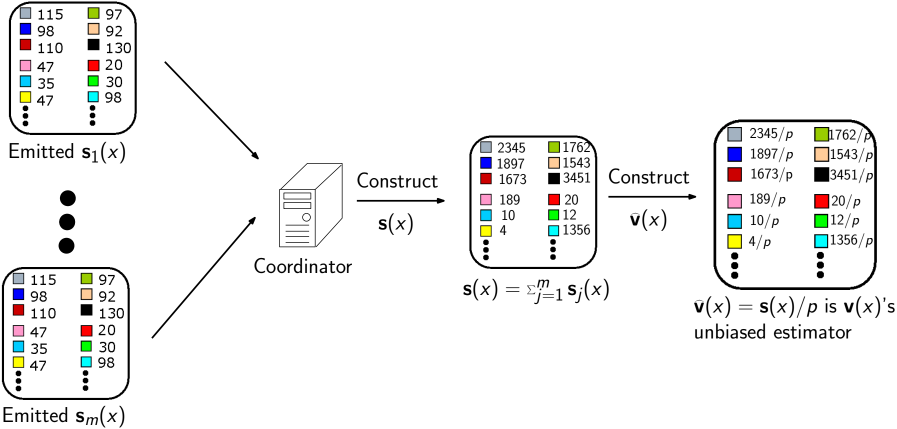
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



Approximate Top-k Wavelet Coefficients: Basic Random Sampling



Approximate Top-k Wavelet Coefficients: Basic Random Sampling



Approximate Top-k Wavelet Coefficients: Basic Random Sampling

- Note: ε must be small for $\widehat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

- Note: ε must be small for $\widehat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.
  - Typical values for ε are 10^{-4} to 10^{-6}.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

- Note: ε must be small for $\widehat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.
  - Typical values for ε are 10^{-4} to 10^{-6}.
- The communication for basic sampling is $O(1/\varepsilon^2)$.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

- Note: ε must be small for $\widehat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.
  - Typical values for ε are 10^{-4} to 10^{-6}.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - With 1 byte keys, 100MB to 1TB of data must be communicated!

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

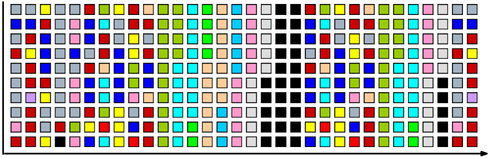
- Note: ε must be small for $\widehat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.
  - Typical values for ε are 10^{-4} to 10^{-6}.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - With 1 byte keys, 100MB to 1TB of data must be communicated!
- We improve basic random sampling with *Improved Sampling*.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

- Note: ε must be small for $\widehat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.
  - Typical values for ε are 10^{-4} to 10^{-6}.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - With 1 byte keys, 100MB to 1TB of data must be communicated!
- We improve basic random sampling with *Improved Sampling*.
  - Key idea: ignore sampled keys with small frequencies in a split.

Outline

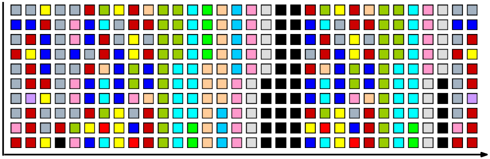
Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j Records in split

Approximate Top-k Wavelet Coefficients: Improved Sampling

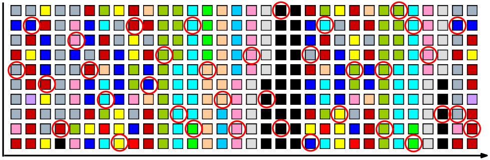
Node j samples $t_j = n_j \cdot p$ records using *Basic Sampling*, where $p = 1/\varepsilon^2 n$.



n_j Records in split

Approximate Top-k Wavelet Coefficients: Improved Sampling

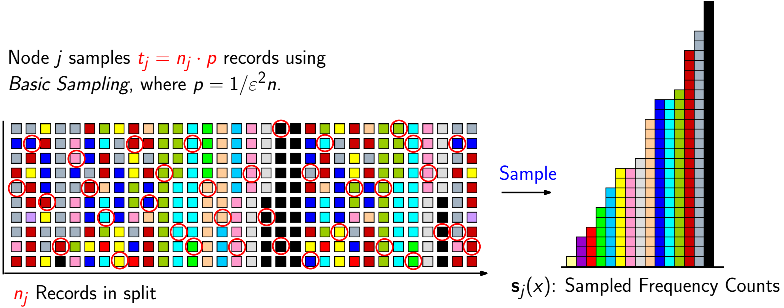
Node j samples $t_j = n_j \cdot p$ records using *Basic Sampling*, where $p = 1/\varepsilon^2 n$.



n_j Records in split

Approximate Top-k Wavelet Coefficients: Improved Sampling

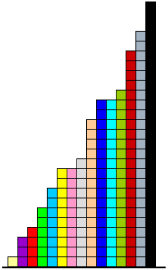
Node j samples $t_j = n_j \cdot p$ records using *Basic Sampling*, where $p = 1/\varepsilon^2 n$.



Approximate Top-k Wavelet Coefficients: Improved Sampling

Node j sends $(x, \mathbf{s}_j(x))$ only if $\mathbf{s}_j(x) > \varepsilon t_j$.

- The error in $\mathbf{s}(x)$ is $\leq \Sigma_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$.

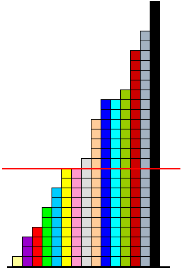


$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Improved Sampling

Node j sends $(x, \mathbf{s}_j(x))$ only if $\mathbf{s}_j(x) > \varepsilon t_j$.

- The error in $\mathbf{s}(x)$ is $\leq \Sigma_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$.

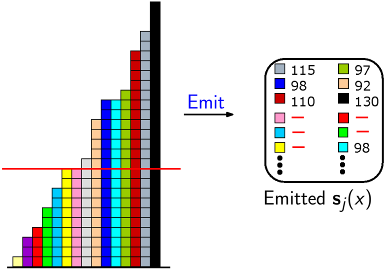


$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Improved Sampling

Node j sends $(x, \mathbf{s}_j(x))$ only if $\mathbf{s}_j(x) > \varepsilon t_j$.

- The error in $\mathbf{s}(x)$ is $\leq \Sigma_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$.



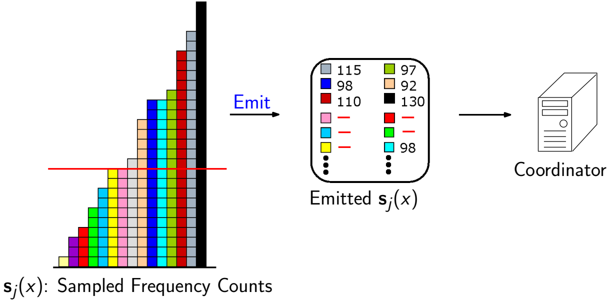
$\mathbf{s}_j(x)$: Sampled Frequency Counts

Emitted $\mathbf{s}_j(x)$

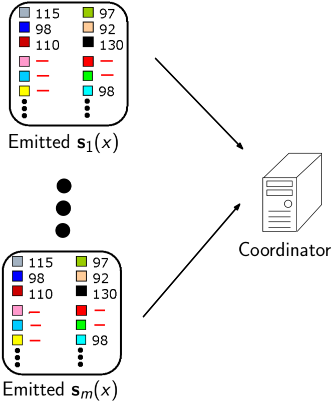
Approximate Top-k Wavelet Coefficients: Improved Sampling

Node j sends $(x, \mathbf{s}_j(x))$ only if $\mathbf{s}_j(x) > \varepsilon t_j$.

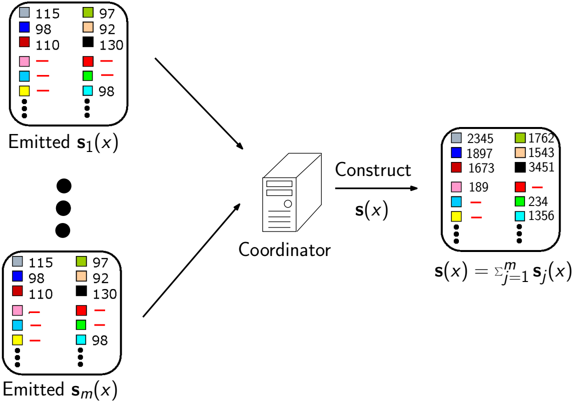
- The error in $\mathbf{s}(x)$ is $\leq \Sigma_{j=1}^{m} \varepsilon t_j = \varepsilon p n = 1/\varepsilon$.



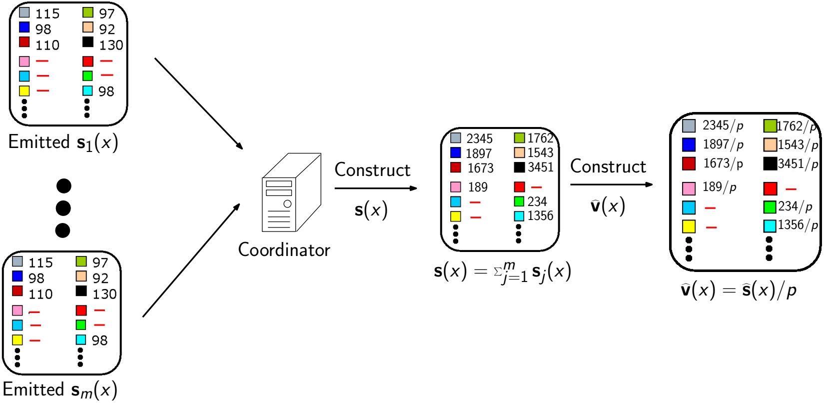
Approximate Top-k Wavelet Coefficients: Improved Sampling



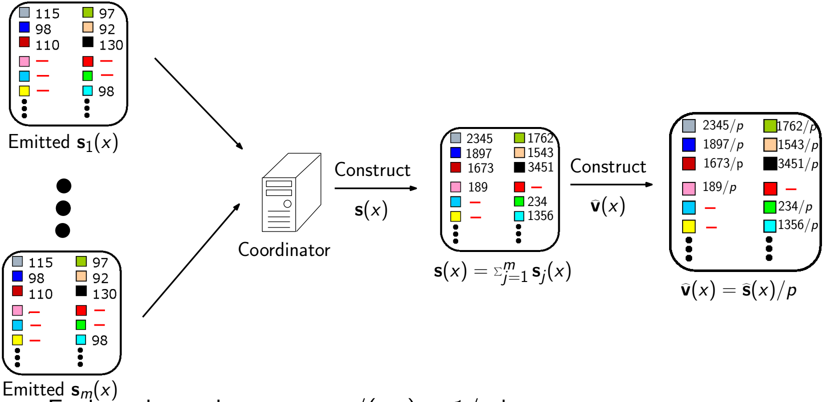
Approximate Top-k Wavelet Coefficients: Improved Sampling



Approximate Top-k Wavelet Coefficients: Improved Sampling

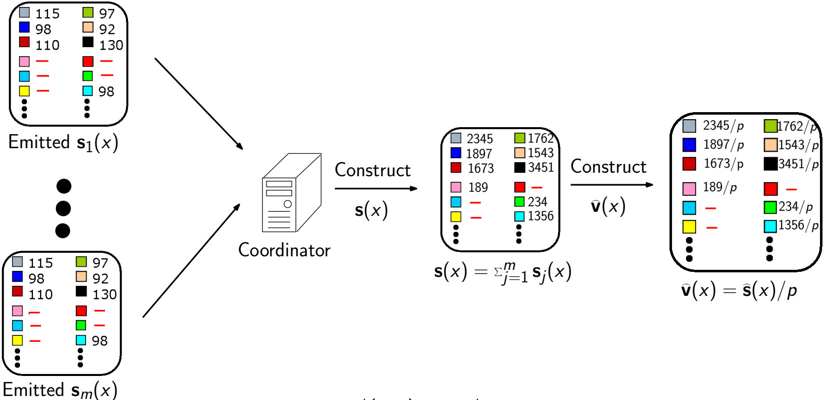


Approximate Top-k Wavelet Coefficients: Improved Sampling



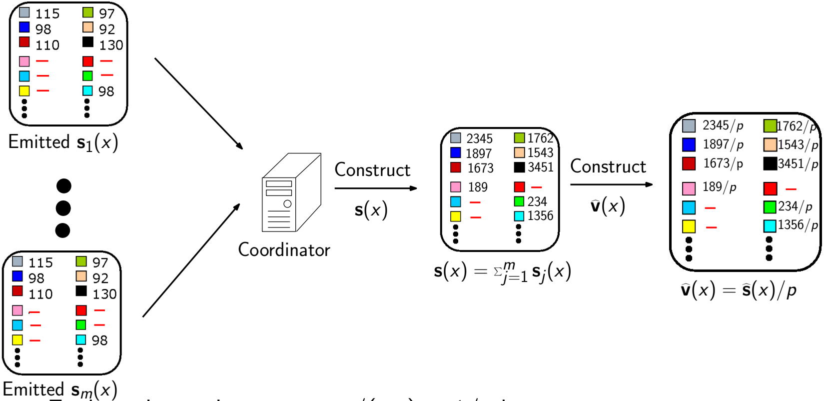
- Each node sends at most $t_j/(\varepsilon t_j) = 1/\varepsilon$ keys.

Approximate Top-k Wavelet Coefficients: Improved Sampling



- Each node sends at most $t_j/(\varepsilon t_j) = 1/\varepsilon$ keys.
    - The total communication is $O(m/\varepsilon)$.

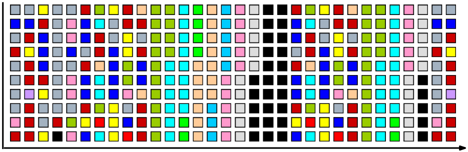
Approximate Top-k Wavelet Coefficients: Improved Sampling



- Each node sends at most $t_j/(\varepsilon t_j) = 1/\varepsilon$ keys.
  - The total communication is $O(m/\varepsilon)$.
- $\mathbf{E}[\widehat{\mathbf{v}}(x)]$ may be εn away from $\mathbf{v}(x)$ as $s_j(x) < \varepsilon t_j$ are ignored.

Outline

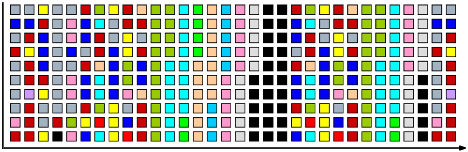
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j Records in split

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

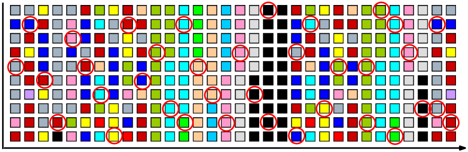
Node j samples $t_j = n_j \cdot p$ records using *Basic Sampling*, where $p = 1/\varepsilon^2 n$.



n_j Records in split

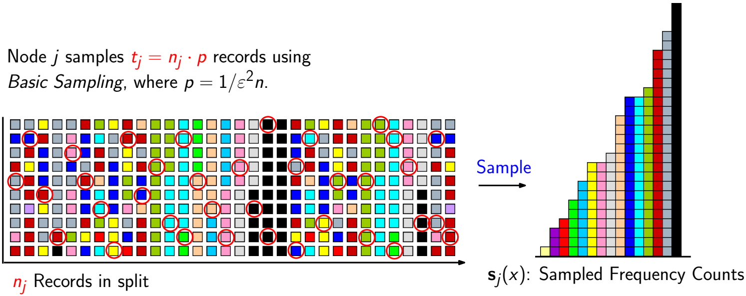
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Node j samples $t_j = n_j \cdot p$ records using *Basic Sampling*, where $p = 1/\varepsilon^2 n$.

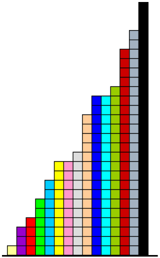


n_j Records in split

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



Approximate Top-k Wavelet Coefficients: Two-Level Sampling

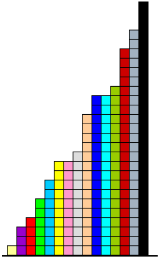


$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Sample record x with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.

- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
- Else emit (x, null) with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

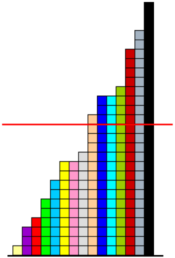


$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Sample record x with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.

- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
- Else emit $(x, null)$ with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

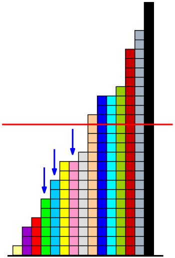


$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Sample record x with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.

- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
- Else emit (x, null) with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

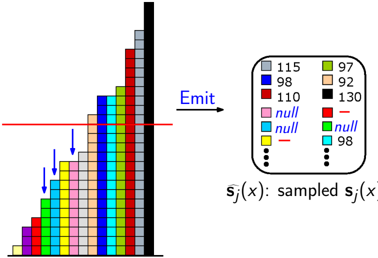


$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Sample record x with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.

- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
- Else emit (x, null) with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.



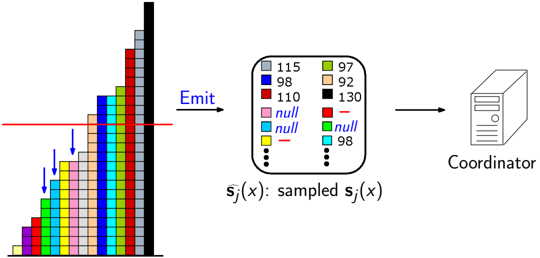
$\widehat{\mathbf{s}_j}(x)$: sampled $\mathbf{s}_j(x)$

$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Sample record x with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.

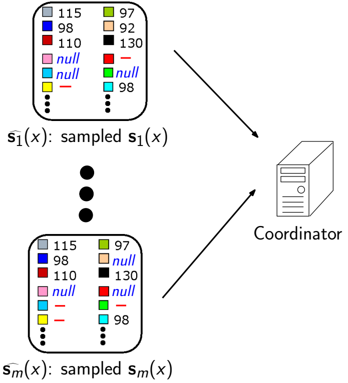
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
- Else emit (x, null) with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.



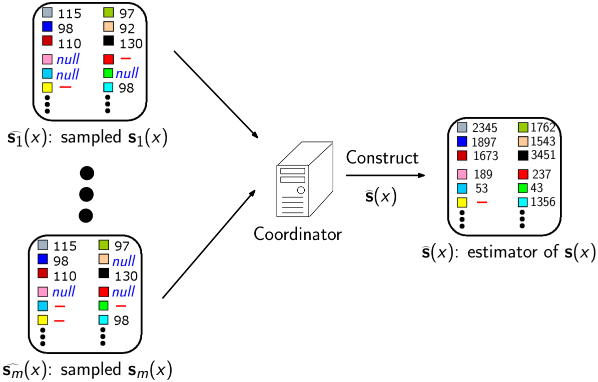
$\widehat{\mathbf{s}_j}(x)$: sampled $\mathbf{s}_j(x)$

$\mathbf{s}_j(x)$: Sampled Frequency Counts

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

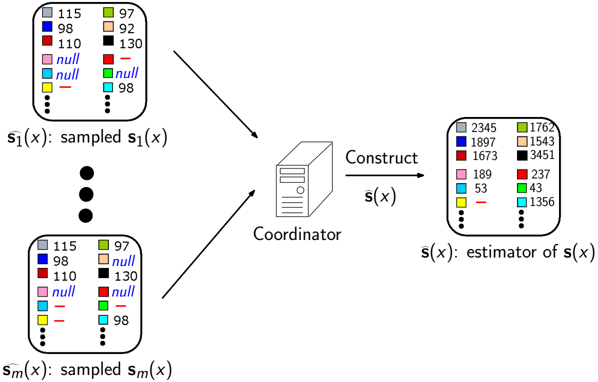


Approximate Top-k Wavelet Coefficients: Two-Level Sampling



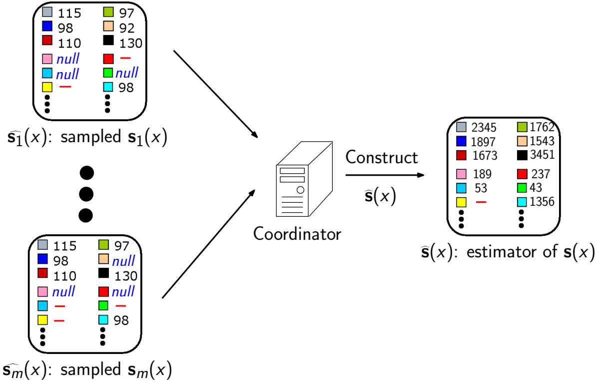
- To construct $\widehat{s}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



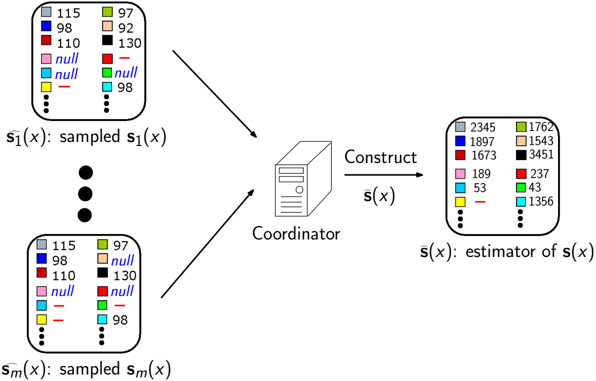
- To construct $\widehat{\mathbf{s}}(x)$.
  - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



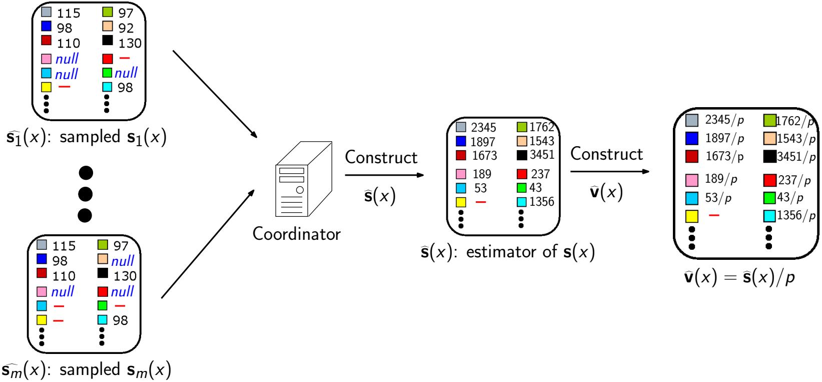
- To construct $\widehat{\mathbf{s}}(x)$.
  - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
  - Else if (x, null) received, $M(x) = M(x) + 1$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



- To construct $\widehat{\mathbf{s}}(x)$.
  - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
  - Else if (x, null) received, $M(x) = M(x) + 1$.
- Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



- To construct $\widehat{\mathbf{s}}(x)$.
  - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
  - Else if $(x, null)$ received, $M(x) = M(x) + 1$.
- Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.
- Then, $\widehat{\mathbf{v}}(x) = \widehat{\mathbf{s}}(x)/p$ is an unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Corollary

$\widehat{\mathbf{v}}(x)$ *is an unbiased estimator of* $\mathbf{v}(x)$ *with standard deviation at most* εn.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Corollary

$\widehat{\mathbf{v}}(x)$ *is an unbiased estimator of* $\mathbf{v}(x)$ *with standard deviation at most* εn.

Theorem

- $\widehat{w}_i$ *is an unbiased estimator for any* w_i.
- *Recall* $w_i = \langle \mathbf{v}, \psi_i \rangle$, *for* $\psi_i = (-\phi_{j+1,2k} + \phi_{j+1,2k+1})/\sqrt{u/2^j}$ *where* ϕ *is a* $[0,1]$ *vector defined for* $j = 1, \ldots, \log u$ *and* $k = 0, \ldots, 2^j - 1$. *The variance of* $\widehat{w}_i$ *is bounded by* $\frac{\varepsilon 2^j n}{u\sqrt{m}} \sum_{x=2ku/2^{j+1}+1}^{(2k+2)u/2^{j+1}} \mathbf{s}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an unbiased estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Corollary

$\widehat{\mathbf{v}}(x)$ is an unbiased estimator of $\mathbf{v}(x)$ with standard deviation at most εn.

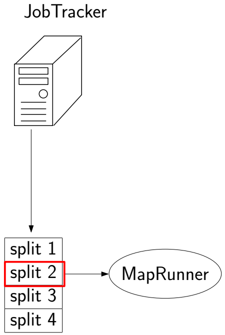
Theorem

- *$\widehat{w}_i$ is an unbiased estimator for any w_i.*
- *Recall $w_i = \langle \mathbf{v}, \psi_i \rangle$, for $\psi_i = (-\phi_{j+1,2k} + \phi_{j+1,2k+1})/\sqrt{u/2^j}$ where ϕ is a $[0,1]$ vector defined for $j = 1, \ldots, \log u$ and $k = 0, \ldots, 2^j - 1$. The variance of $\widehat{w}_i$ is bounded by $\frac{\varepsilon 2^j n}{u\sqrt{m}} \sum_{x=2ku/2^{j+1}+1}^{(2k+2)u/2^{j+1}} \mathbf{s}(x)$.*

Theorem

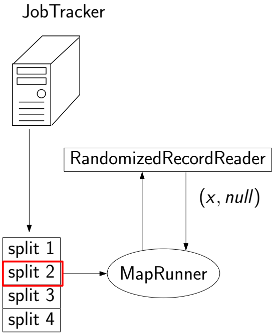
The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

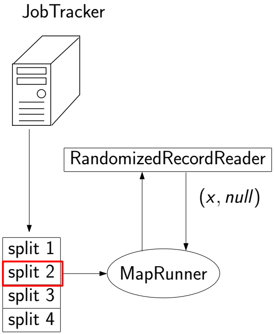
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.

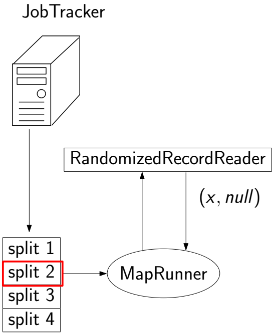
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.

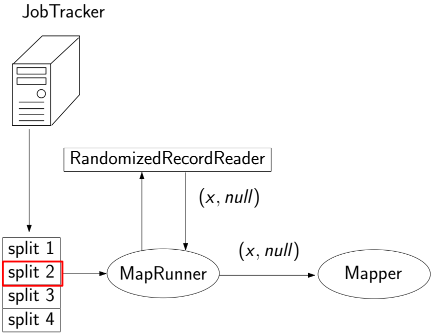
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

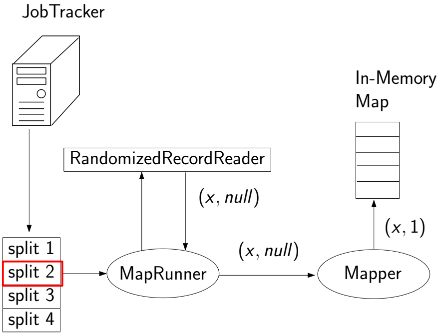
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
    2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

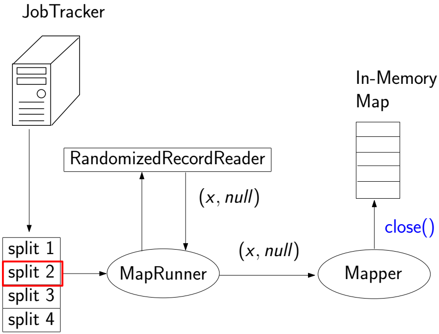
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
    2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

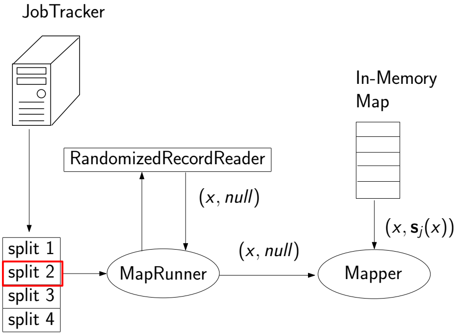
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

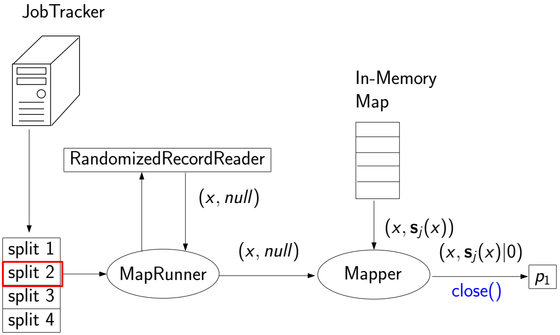
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
    2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

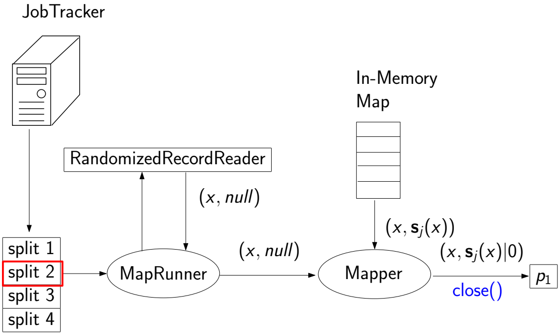
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

2. Mapper j samples key x from $\mathbf{s}$ with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.

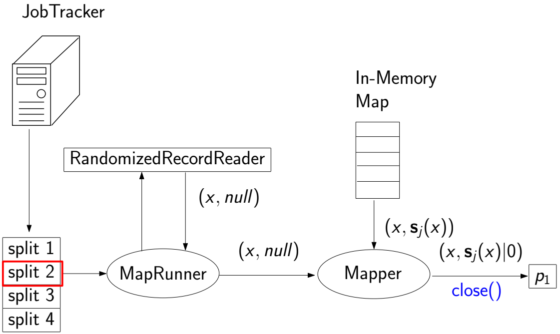
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

2. Mapper j samples key x from $\mathbf{s}$ with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.
   - If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.

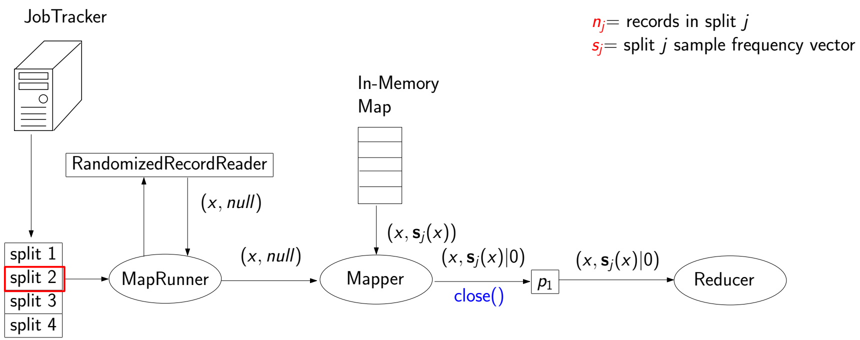
Approximate Top-k Wavelet Coefficients: Two-Level Sampling



n_j= records in split j
s_j= split j sample frequency vector

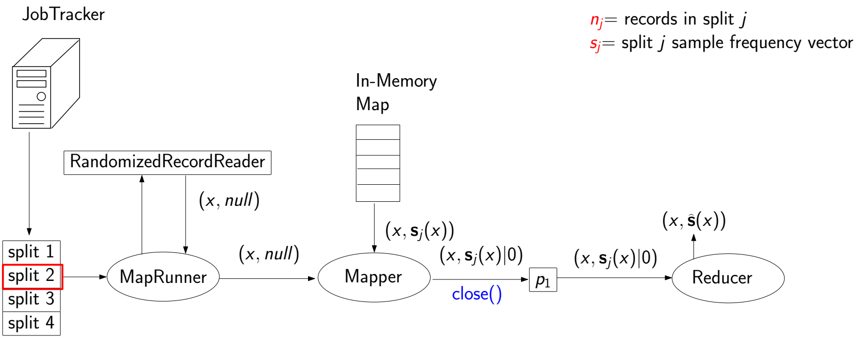
2. Mapper j samples key x from $\mathbf{s}$ with probability $\min\{\varepsilon\sqrt{m}\cdot \mathbf{s}_j(x), 1\}$.
   - If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
   - Else emit $(x, 0)$ with probability $\varepsilon\sqrt{m}\cdot \mathbf{s}_j(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



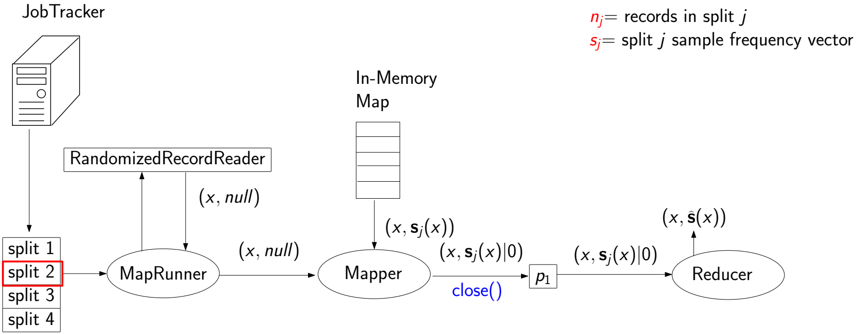
2. Mapper j samples key x from $\mathbf{s}$ with probability $\min\{\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x), 1\}$.
    - If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$, emit $(x, \mathbf{s}_j(x))$.
    - Else emit $(x, 0)$ with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



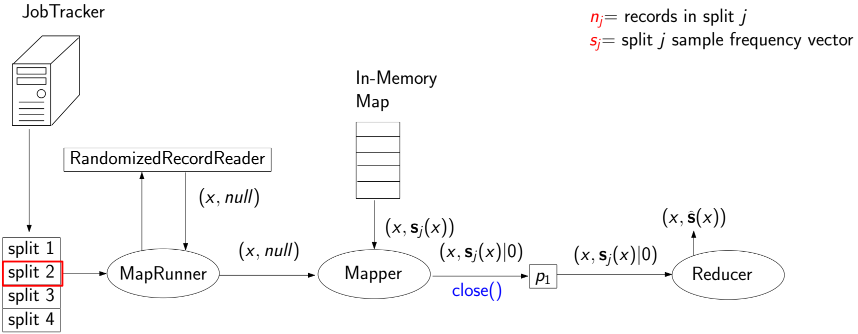
3. Construct $\widehat{\mathbf{s}}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



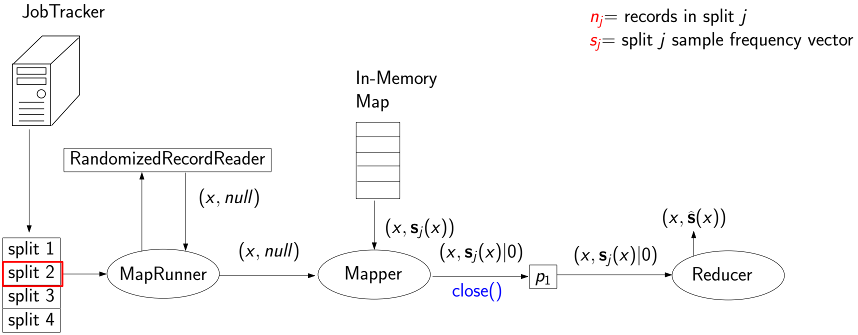
3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



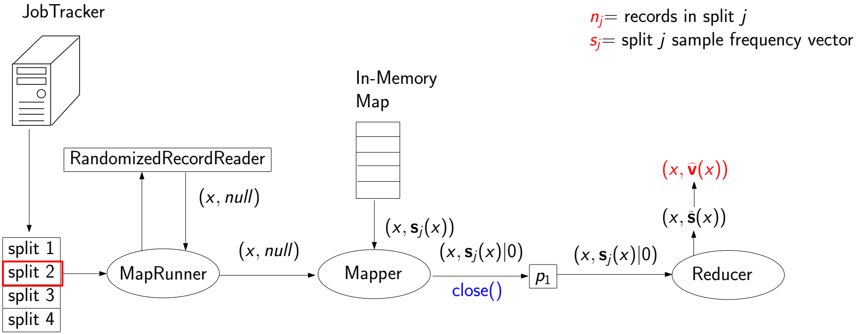
3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



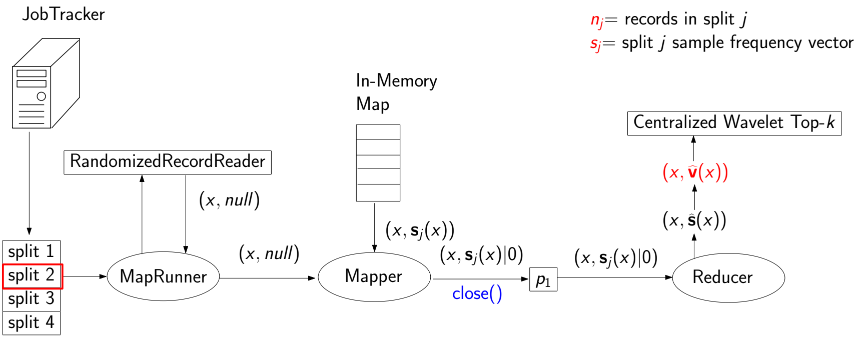
3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
4. Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



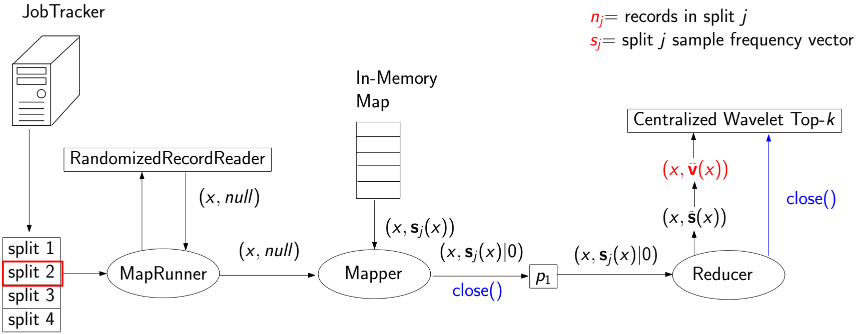
3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
4. Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.
5. Reducer uses $\widehat{\mathbf{v}}(x) = \widehat{\mathbf{s}}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



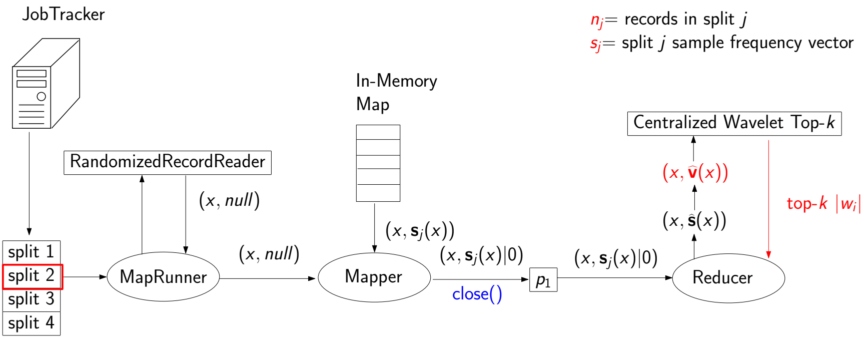
3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
4. Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.
5. Reducer uses $\widehat{\mathbf{v}}(x) = \widehat{\mathbf{s}}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



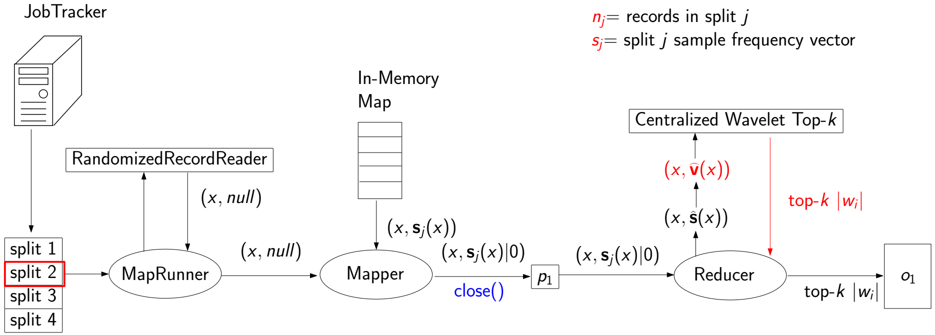
3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
4. Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.
5. Reducer uses $\widehat{\mathbf{v}}(x) = \widehat{\mathbf{s}}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
4. Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.
5. Reducer uses $\widehat{\mathbf{v}}(x) = \widehat{\mathbf{s}}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling



3. Construct $\widehat{\mathbf{s}}(x)$.
   - If $(x, \mathbf{s}_j(x))$ received, $\rho(x) = \rho(x) + \mathbf{s}_j(x)$.
   - Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
4. Finally, $\widehat{\mathbf{s}}(x) = \rho(x) + M(x)/\varepsilon\sqrt{m}$.
5. Reducer uses $\widehat{\mathbf{v}}(x) = \widehat{\mathbf{s}}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.
  - Only 1.2MB of data needs to be communicated!

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys .
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.
  - Only 1.2MB of data needs to be communicated!
  - 330-fold reduction over basic sampling and 33-fold reduction over improved sampling!

Outline

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods:
    - The baseline solution is denoted *Send-V*,

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods:
    - The baseline solution is denoted *Send-V*,
    - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-*k*").

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods:
    - The baseline solution is denoted *Send-V*,
    - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-k").
  - Approximate Methods:

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods:
    - The baseline solution is denoted *Send-V*,
    - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-*k*").
  - Approximate Methods:
    - *Improved Sampling* is denoted *Improved-S*.

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods:
    - The baseline solution is denoted *Send-V*,
    - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-k").
  - Approximate Methods:
    - *Improved Sampling* is denoted *Improved-S*.
    - *Two-Level Sampling* is denoted *TwoLevel-S*.

Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods:
    - The baseline solution is denoted *Send-V*,
    - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-*k*").
  - Approximate Methods:
    - *Improved Sampling* is denoted *Improved-S*.
    - *Two-Level Sampling* is denoted *TwoLevel-S*.
    - The *Sketch-Based Approximation* using the GCS-Sketch is denoted *Send-Sketch*.

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
     - One is reserved for the master (running JobTracker and NameNode).

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
     - One is reserved for the master (running JobTracker and NameNode).
  3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
     - One is reserved for the master (running JobTracker and NameNode).
  3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
     - One is reserved for the (only) Reducer.

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
    1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
    2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
        - One is reserved for the master (running JobTracker and NameNode).
    3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
        - One is reserved for the (only) Reducer.
    4. 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU

Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
     - One is reserved for the master (running JobTracker and NameNode).
  3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
     - One is reserved for the (only) Reducer.
  4. 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU
- All machines are directly connected to a 1000Mbps switch.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4 byte integer attributes including a client id and object id.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4 byte integer attributes including a client id and object id.
  - We assign each record a *clientobject* 4 byte integer id in $u = 2^{29}$ which is distinct for unique parings of a client id and object id.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4 byte integer attributes including a client id and object id.
  - We assign each record a *clientobject* 4 byte integer id in $u = 2^{29}$ which is distinct for unique parings of a client id and object id.
  - *WorldCup* is stored in binary format, in total it is 50GB.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4 byte integer attributes including a client id and object id.
  - We assign each record a *clientobject* 4 byte integer id in $u = 2^{29}$ which is distinct for unique parings of a client id and object id.
  - *WorldCup* is stored in binary format, in total it is 50GB.
- We utilize large synthetic Zipfian datasets to evaluate all methods.

Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4 byte integer attributes including a client id and object id.
  - We assign each record a *clientobject* 4 byte integer id in $u = 2^{29}$ which is distinct for unique parings of a client id and object id.
  - *WorldCup* is stored in binary format, in total it is 50GB.
- We utilize large synthetic Zipfian datasets to evaluate all methods.
  - Keys are randomly permuted and discontiguous in a dataset.

Experiments: Datasets

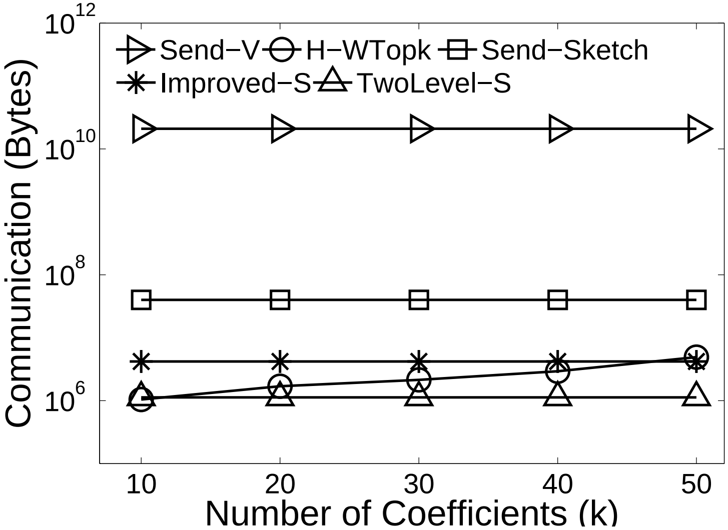
- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4 byte integer attributes including a client id and object id.
  - We assign each record a *clientobject* 4 byte integer id in $u = 2^{29}$ which is distinct for unique parings of a client id and object id.
  - *WorldCup* is stored in binary format, in total it is 50GB.
- We utilize large synthetic Zipfian datasets to evaluate all methods.
  - Keys are randomly permuted and discontiguous in a dataset.
  - Each key is a 4-byte integer and stored in binary format.

Experiments: Defaults

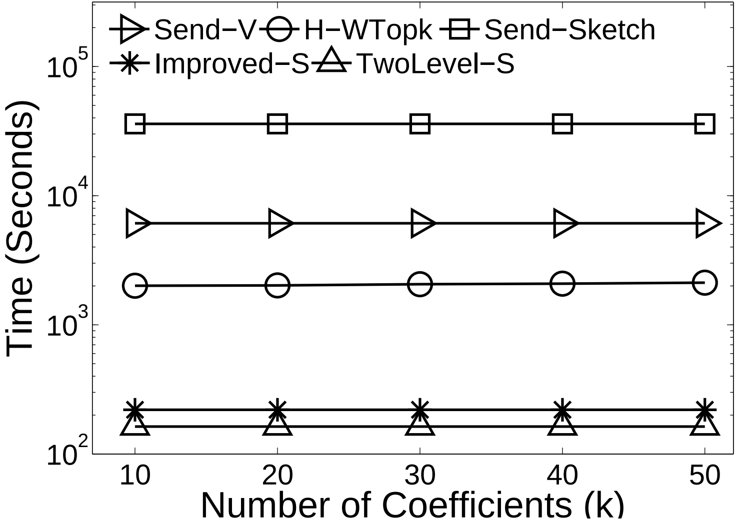
- Default values:

Symbol	Definition	Default
α	Zipfian skewness	1.1
u	max key in domain	$\log_2 u = 29$
n	total records	13.4 billion
	dataset size	50GB
β	split size	256MB
m	number of splits	200
B	network bandwidth	500Mbps

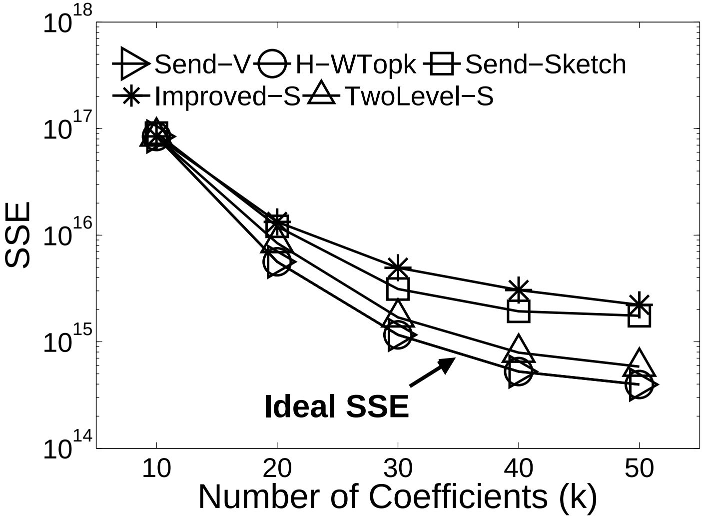
Experiments: Vary k



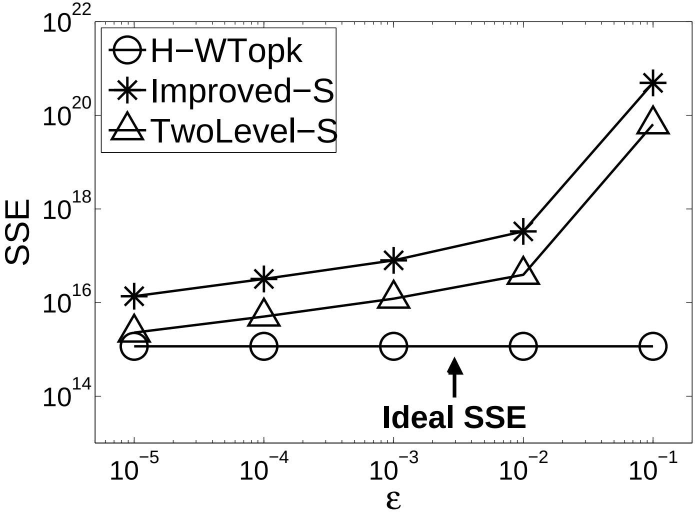
Experiments: Vary k



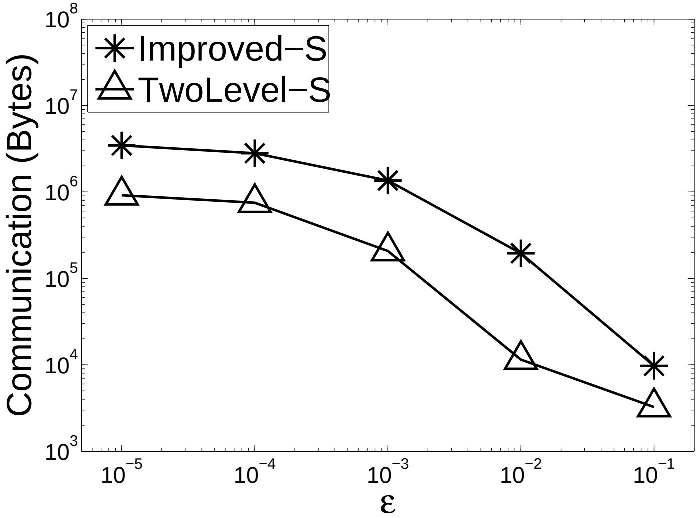
Experiments: Vary k



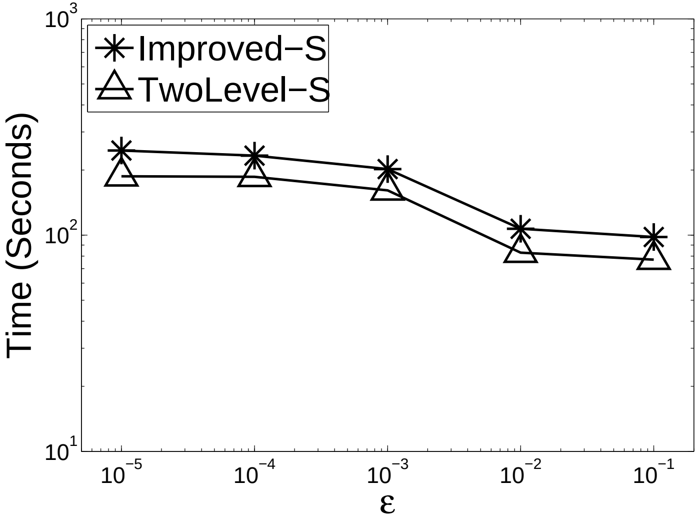
Experiments: Vary ε



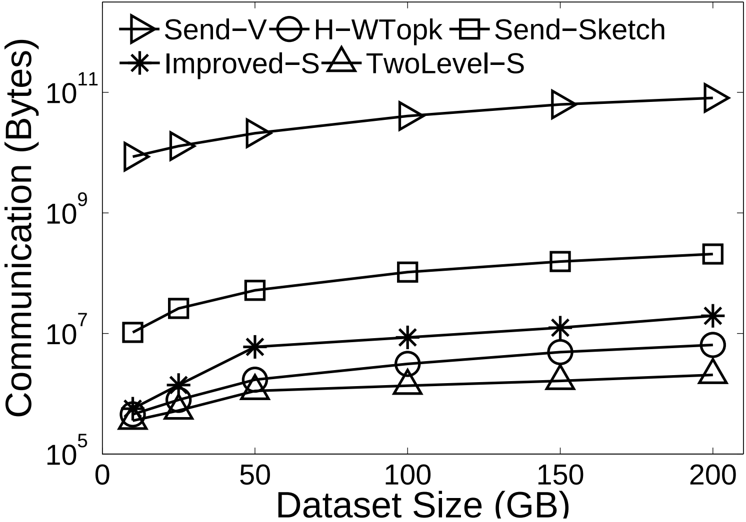
Experiments: Vary ε



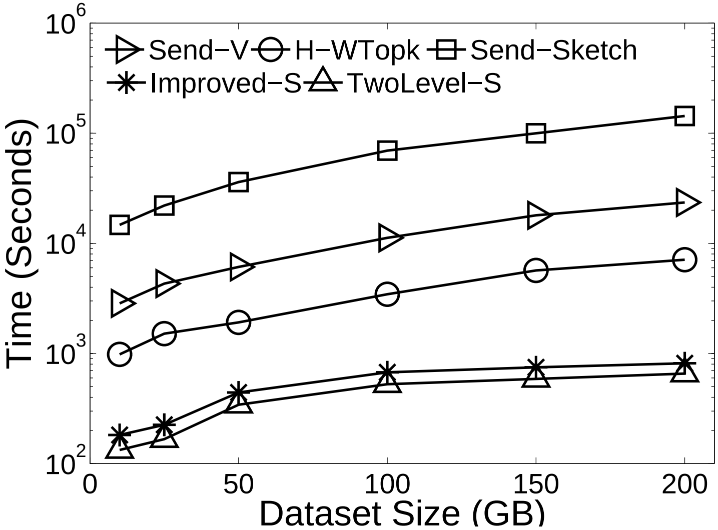
Experiments: Vary ε



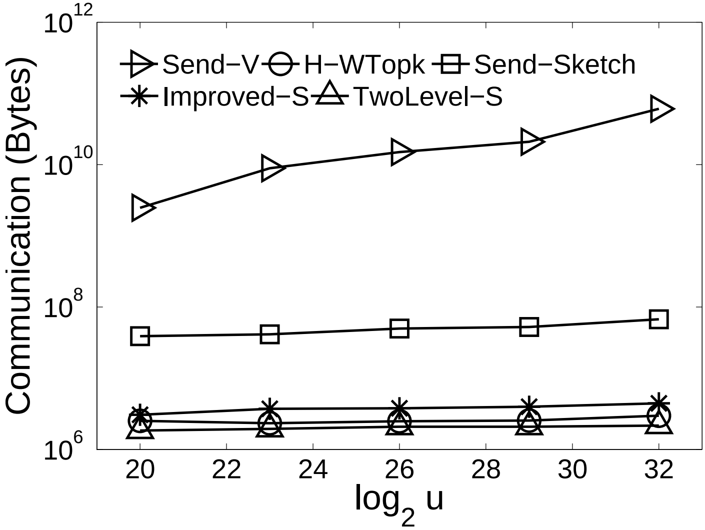
Experiments: Vary n



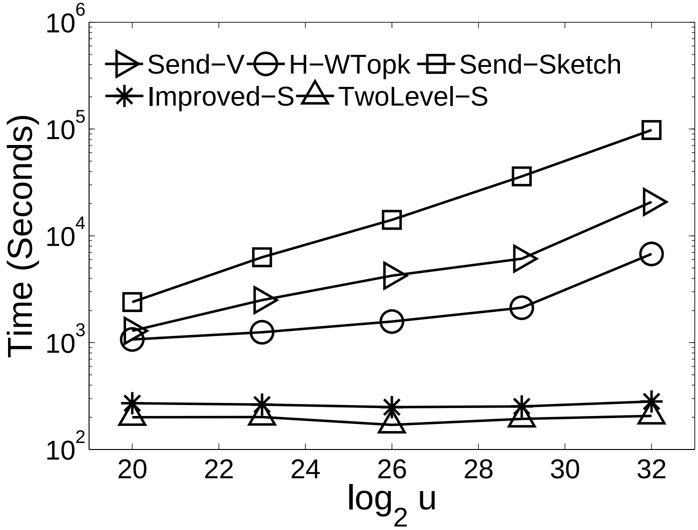
Experiments: Vary n



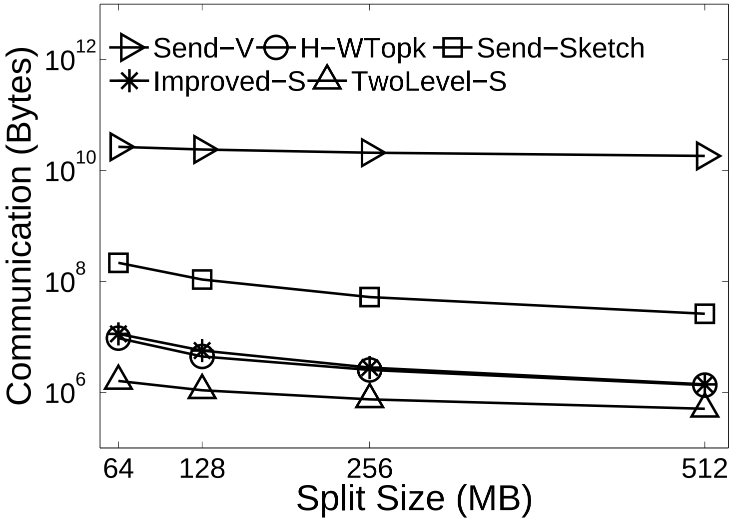
Experiments: Vary u



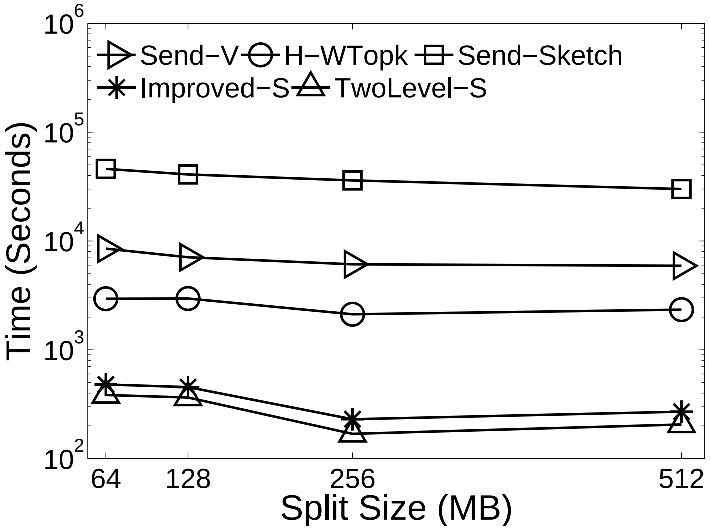
Experiments: Vary u



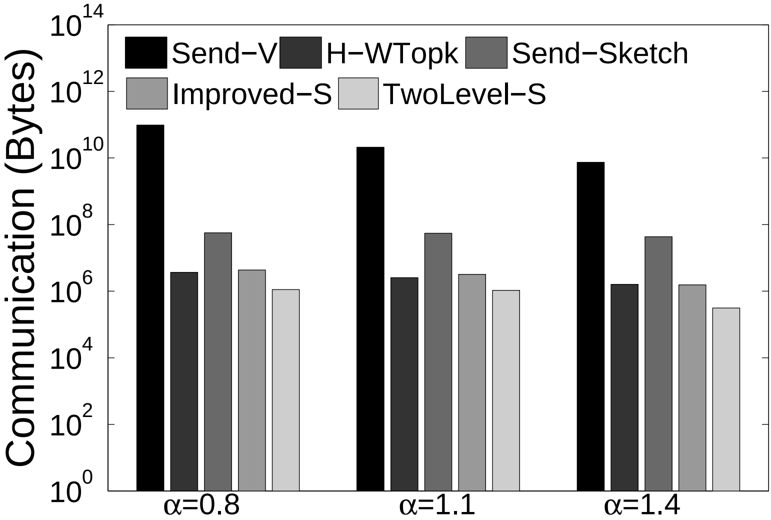
Experiments: Vary β



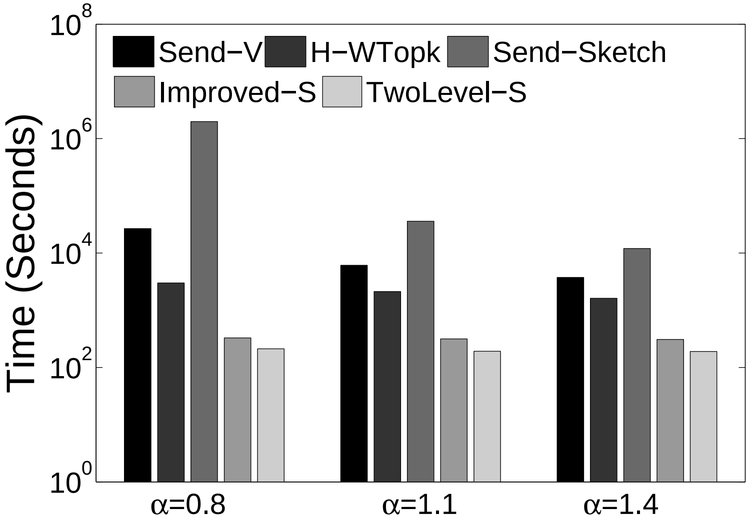
Experiments: Vary β



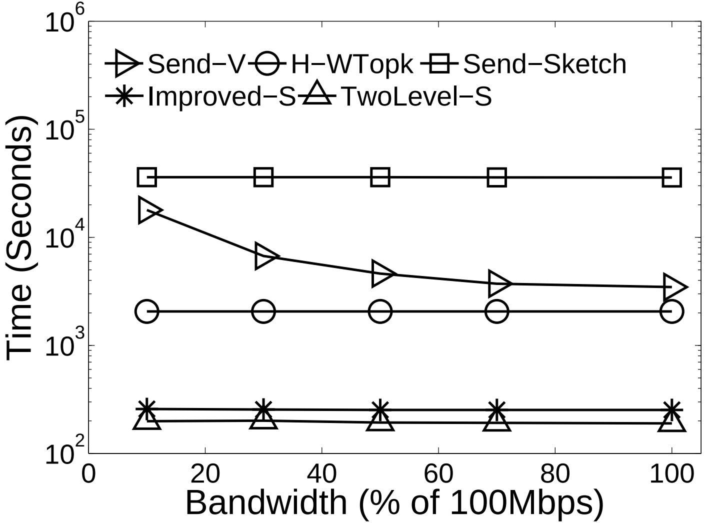
Experiments: Vary α



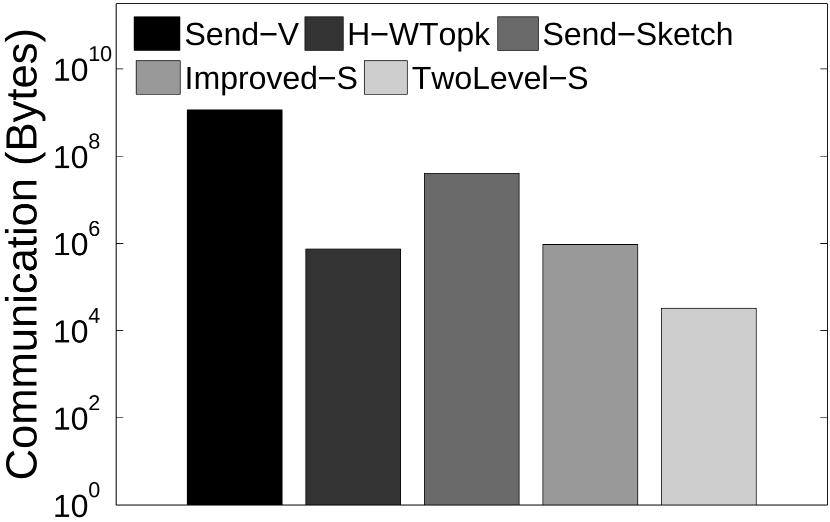
Experiments: Vary α



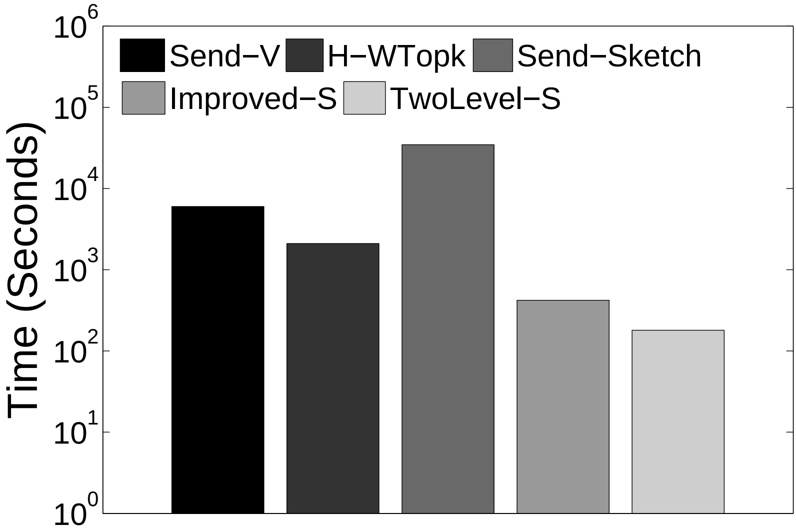
Experiments: Vary B



Experiments: *WorldCup* Dataset



Experiments: *WorldCup* Dataset



Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.

Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.

Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.
  - *TwoLevel-S* is especially easy to implement and ideal in practice.

Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
    - We present both exact and approximate algorithms.
    - *TwoLevel-S* is especially easy to implement and ideal in practice.
        - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!

Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
    - We present both exact and approximate algorithms.
    - *TwoLevel-S* is especially easy to implement and ideal in practice.
        - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!
- Our work is just the tip of the iceberg for data summarization techniques in MapReduce.

Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.
  - *TwoLevel-S* is especially easy to implement and ideal in practice.
    - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!
- Our work is just the tip of the iceberg for data summarization techniques in MapReduce.
- Many others remain including:

Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.
  - *TwoLevel-S* is especially easy to implement and ideal in practice.
    - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!
- Our work is just the tip of the iceberg for data summarization techniques in MapReduce.
- Many others remain including:
  - other histograms including the V-optimal histogram,
  - sketches and synopsis,
  - geometric summaries (ε-approximations and coresets),
  - graph summaries (distance oracles).

The End

Thank You

Q and A

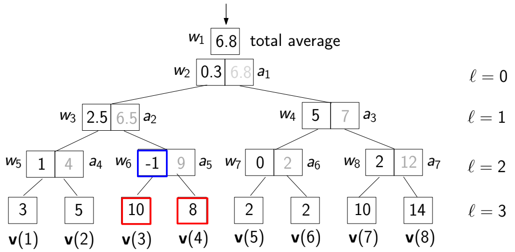
Introduction: Histograms

- We may also compute w_i with the wavelet basis vectors ψ_i.

Introduction: Histograms

- We may also compute w_i with the wavelet basis vectors ψ_i.

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

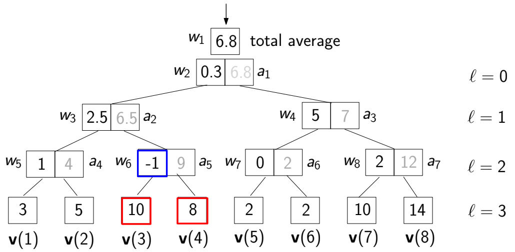


$$w_6 = \langle \mathbf{v}, \psi_6 \rangle$$

Introduction: Histograms

- We may also compute w_i with the wavelet basis vectors ψ_i.

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

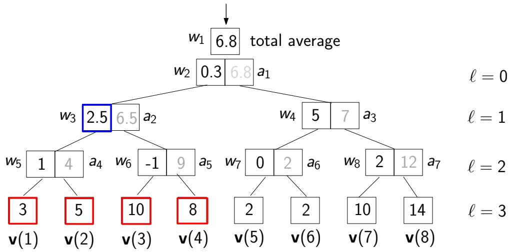


$$w_6 = \langle \mathbf{v}, \psi_6 \rangle$$

$w_6 = \langle \mathbf{v}, \psi_6 \rangle =$
$\langle [0\ 0\ \mathbf{v}(3)\ 0\ 0\ 0\ 0\ 0], \psi_6 \rangle + \langle [\ 0\ 0\ 0\ \mathbf{v}(4)\ 0\ 0\ 0\ 0], \psi_6 \rangle$

- We may also compute w_i with the wavelet basis vectors ψ_i.

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

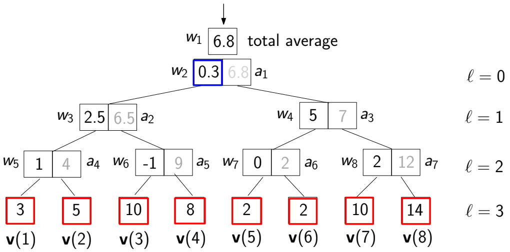


$$w_3 = \langle \mathbf{v}, \psi_3 \rangle$$

Introduction: Histograms

- We may also compute w_i with the wavelet basis vectors ψ_i.

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14

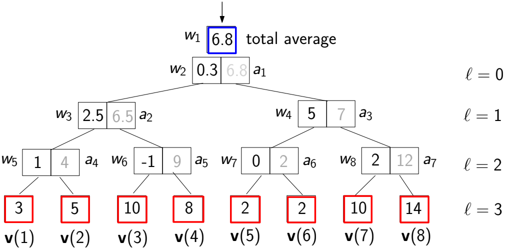


$$w_2 = \langle \mathbf{v}, \psi_2 \rangle$$

Introduction: Histograms

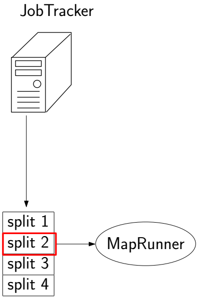
- We may also compute w_i with the wavelet basis vectors ψ_i.

x	1	2	3	4	5	6	7	8
$\mathbf{v}(x)$	3	5	10	8	2	2	10	14



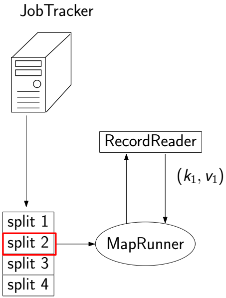
$$w_1 = \langle \mathbf{v}, \psi_1 \rangle$$

Background: Hadoop MapReduce, Map Phase



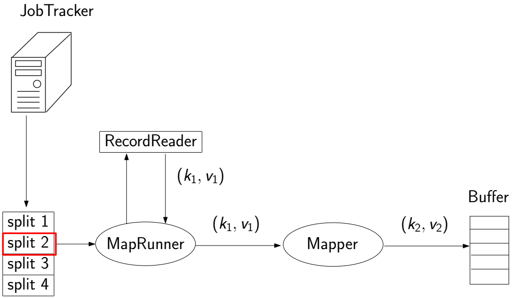
- The JobTracker assigns an InputSplit to a TaskTracker, a MapRunner task runs on the TaskTracker to process the split.

Background: Hadoop MapReduce, Map Phase



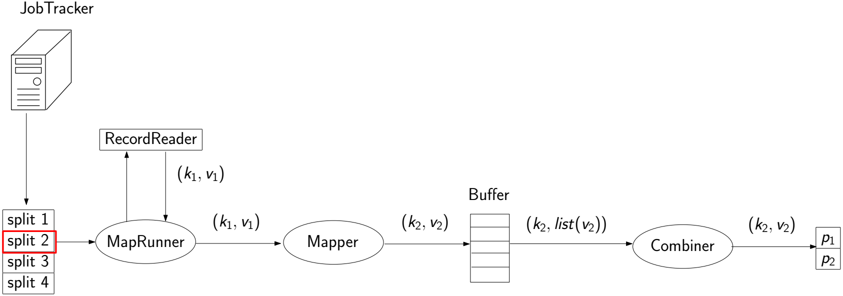
- The MapRunner acquires a RecordReader from the InputFormat for the file to view the InputSplit as a stream of records, (k_1, v_1).

Background: Hadoop MapReduce, Map Phase



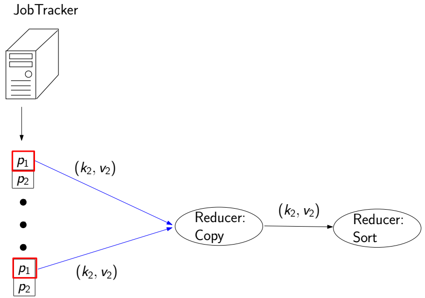
- The MapRunner invokes the user specified *Mapper* for each (k_1, v_1), the Mapper emits (k_2, v_2) and stores in an in-memory buffer.

Background: Hadoop MapReduce, Map Phase



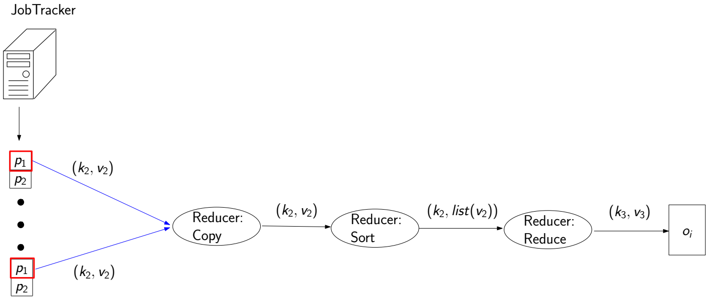
- When the buffer fills, the optional *Combiner* is executed over $(k_2, list(v_2))$, and a (k_2, v_2) is dumped to a partition on disk.

Background: Hadoop MapReduce, Shuffle and Sort Phase



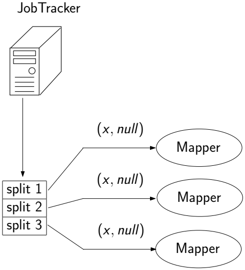
- The JobTracker assigns Reducers to TaskTrackers for each partition, each reducer first copies on (k_2, v_2) and then sorts on k_2.

Background: Hadoop MapReduce, Reduce Phase

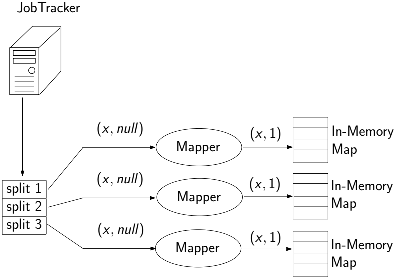


- The sorting output $(k_2, list(v_2))$ is processed one k_2 at a time and reduced, the reduced output (k_3, v_3) is written to reducer output o_i.

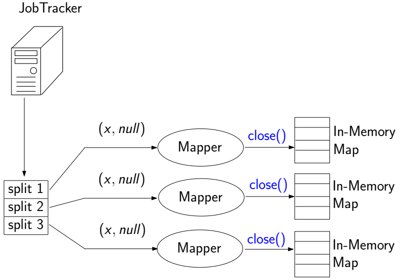
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



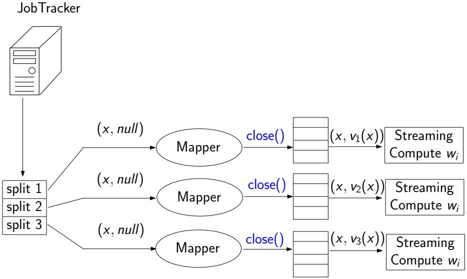
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



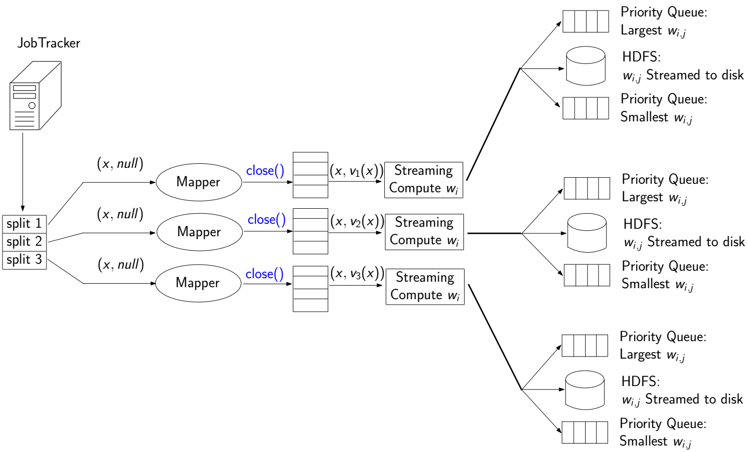
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



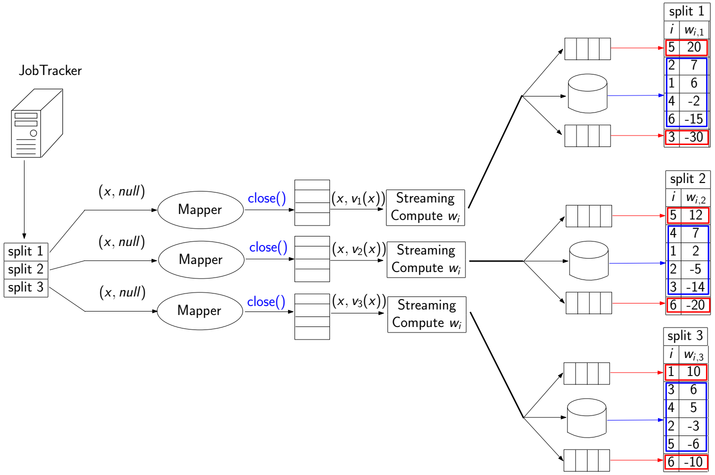
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



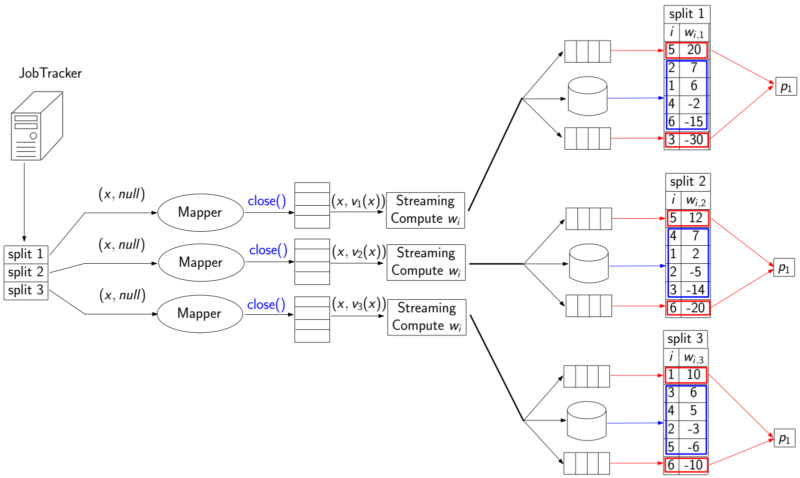
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



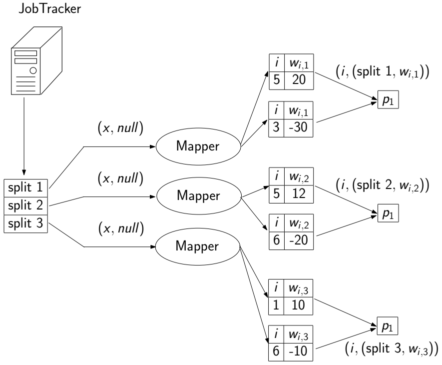
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



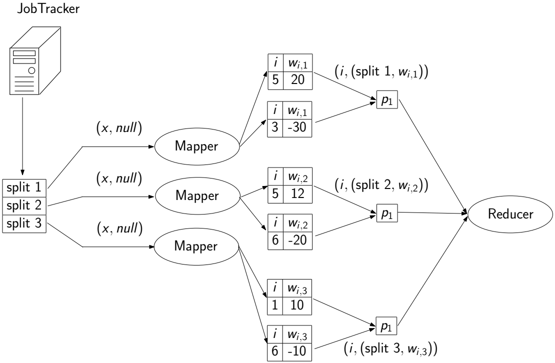
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



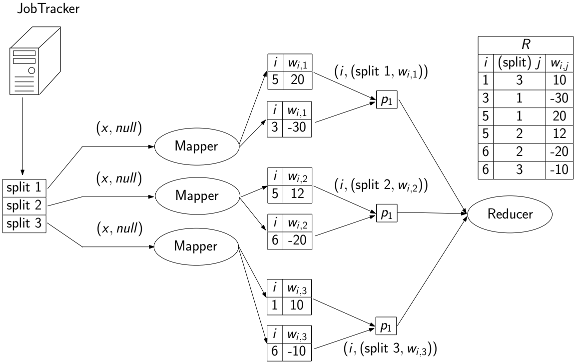
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



Exact Top-k Wavelet Coefficients: Hadoop Phase 1

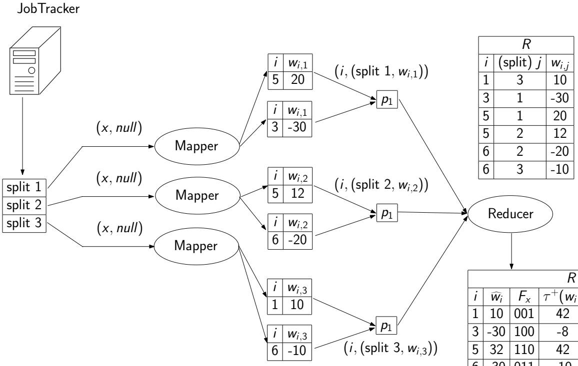


Exact Top-k Wavelet Coefficients: Hadoop Phase 1



R		
i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

Exact Top-k Wavelet Coefficients: Hadoop Phase 1



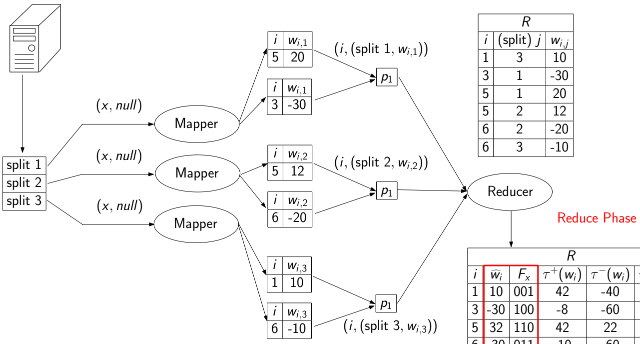
R

i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

R

i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

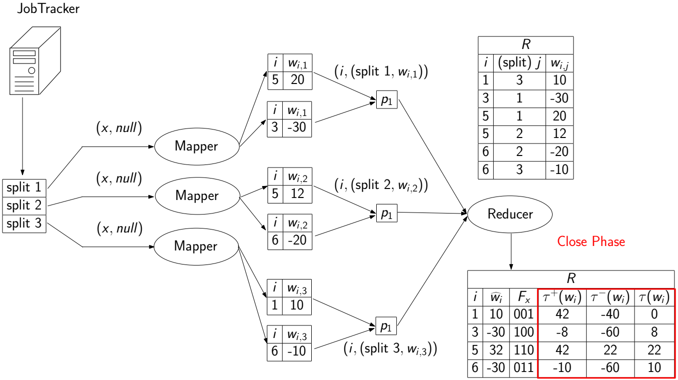
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



R		
i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

R					
i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

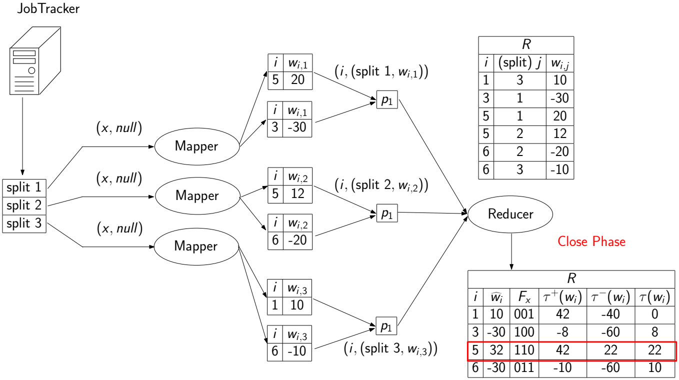
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



R		
i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

R					
i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

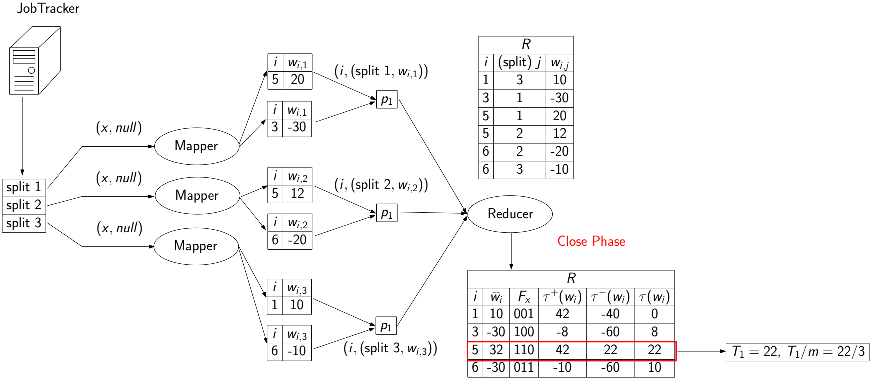
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



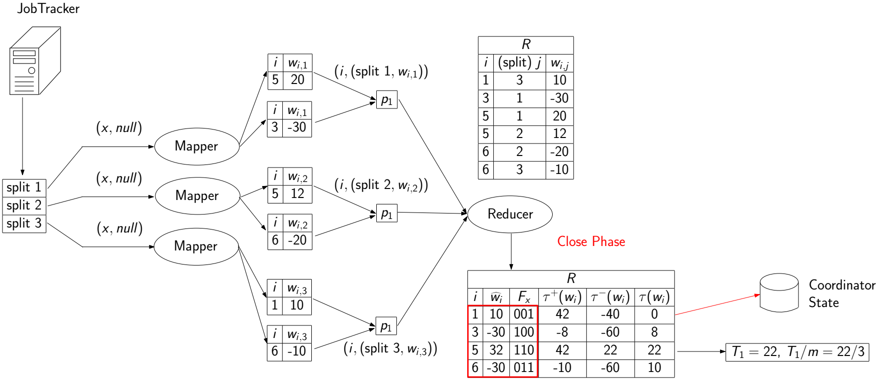
R		
i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

R					
i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

Exact Top-k Wavelet Coefficients: Hadoop Phase 1



Exact Top-k Wavelet Coefficients: Hadoop Phase 1

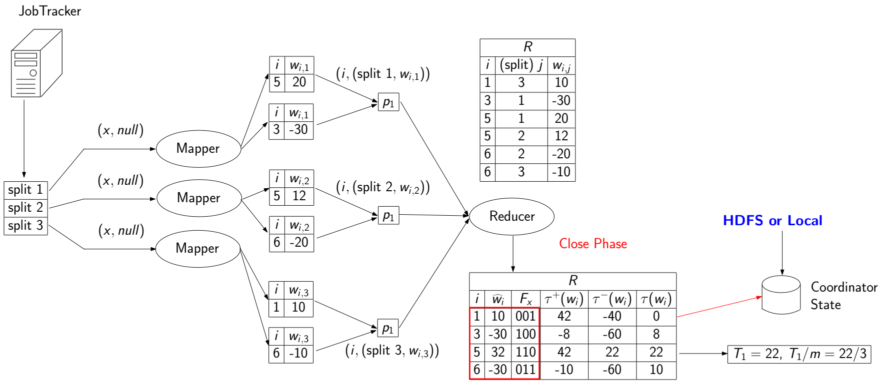


R		
i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

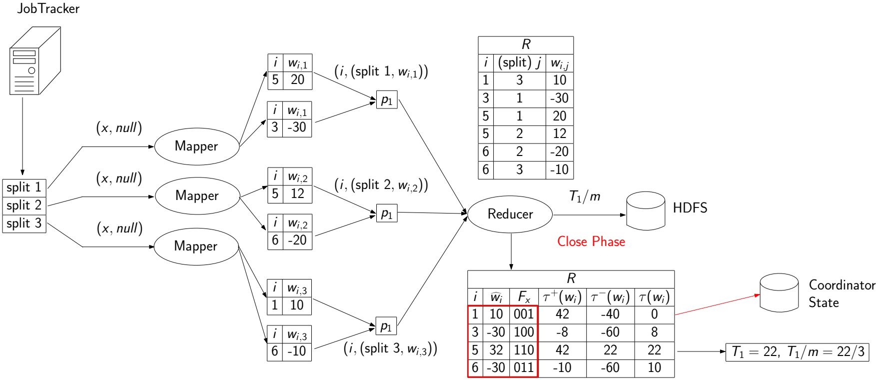
R					
i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

$T_1 = 22,\ T_1/m = 22/3$

Exact Top-k Wavelet Coefficients: Hadoop Phase 1



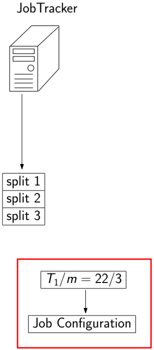
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



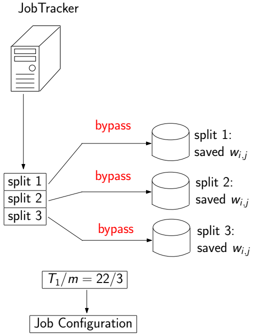
R		
i	(split) j	$w_{i,j}$
1	3	10
3	1	-30
5	1	20
5	2	12
6	2	-20
6	3	-10

R					
i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$
1	10	001	42	-40	0
3	-30	100	-8	-60	8
5	32	110	42	22	22
6	-30	011	-10	-60	10

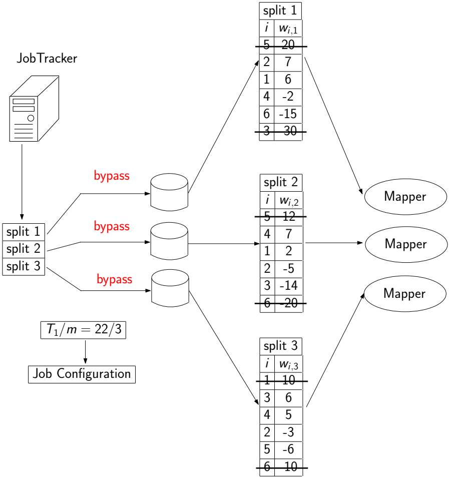
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



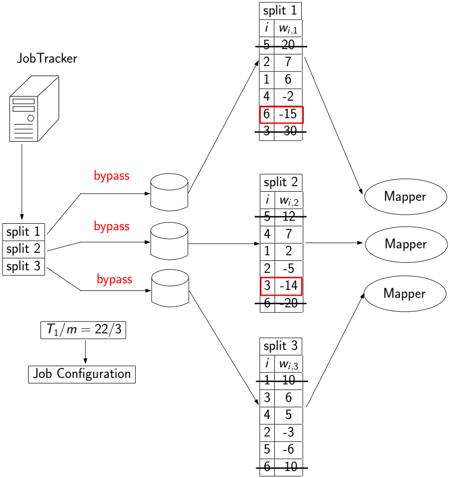
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



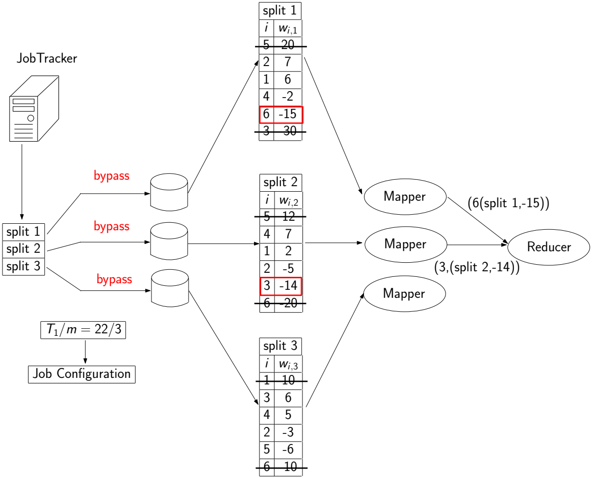
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



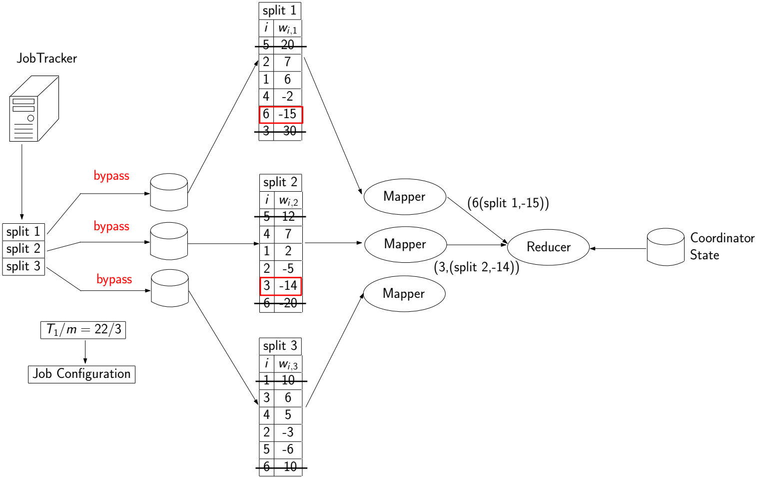
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



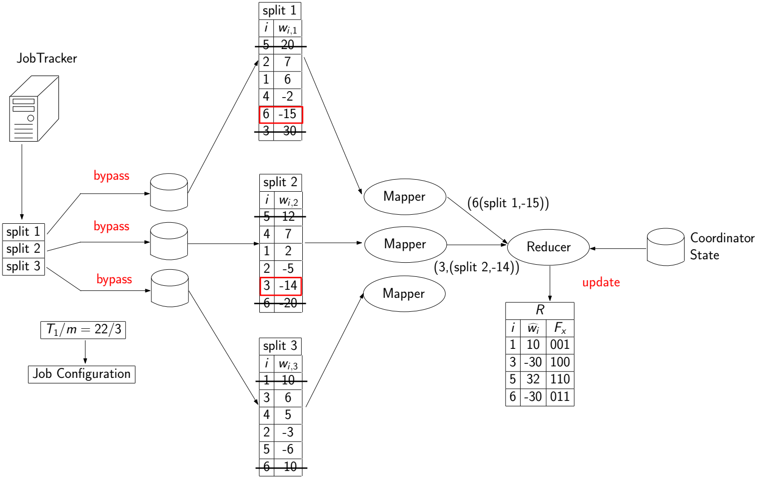
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



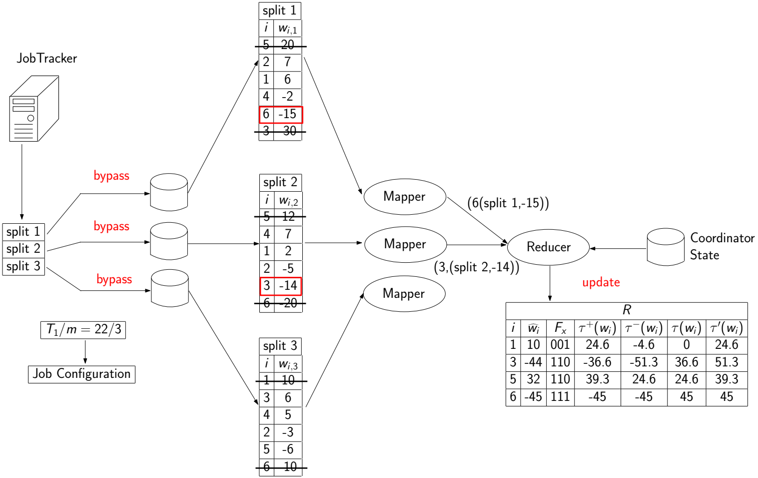
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



Exact Top-k Wavelet Coefficients: Hadoop Phase 3



Exact Top-k Wavelet Coefficients: Hadoop Phase 3



split 1

i	$w_{i,1}$
~~5~~	~~20~~
2	7
1	6
4	-2
6	-15
~~3~~	~~30~~

split 2

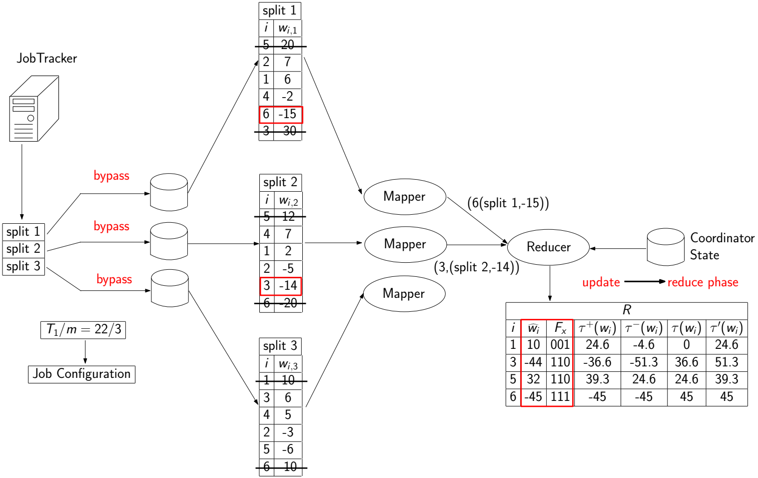
i	$w_{i,2}$
~~5~~	~~12~~
4	7
1	2
2	-5
3	-14
~~6~~	~~-20~~

split 3

i	$w_{i,3}$
~~1~~	~~10~~
3	6
4	5
2	-3
5	-6
~~6~~	~~10~~

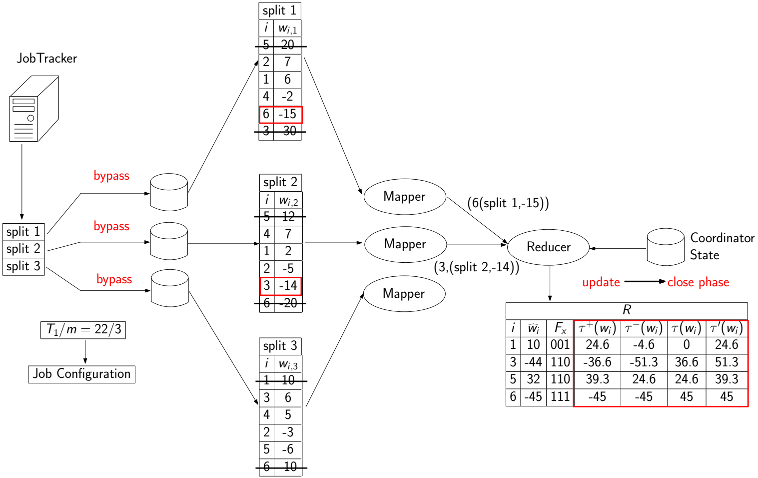
R

i	$\widehat{w}_i$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$	$\tau'(w_i)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45



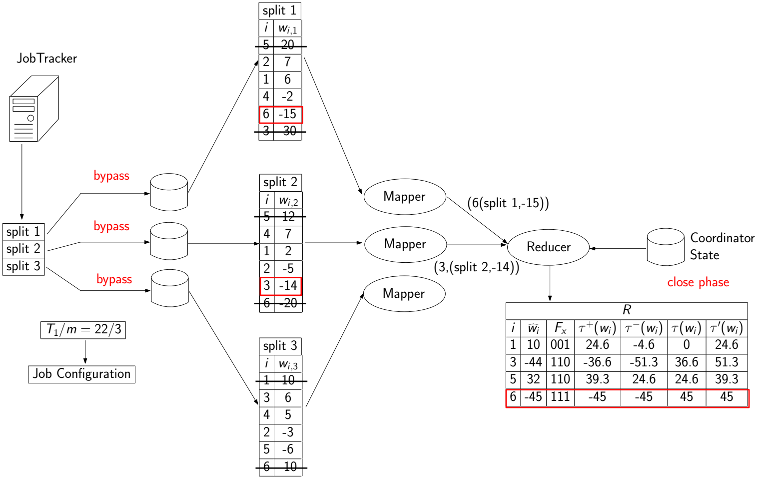
R						
i	$\widehat{w}_i$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$	$\tau'(w_i)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

Exact Top-k Wavelet Coefficients: Hadoop Phase 3



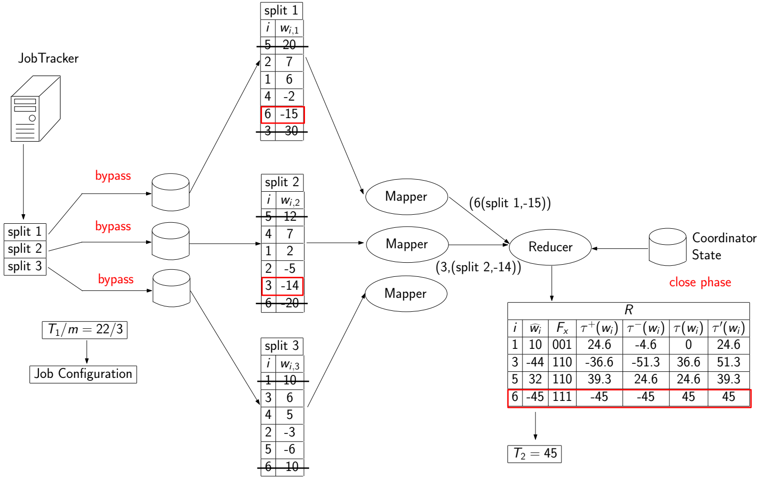
R						
i	$\widehat{w}_i$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$	$\tau'(w_i)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

Exact Top-k Wavelet Coefficients: Hadoop Phase 3

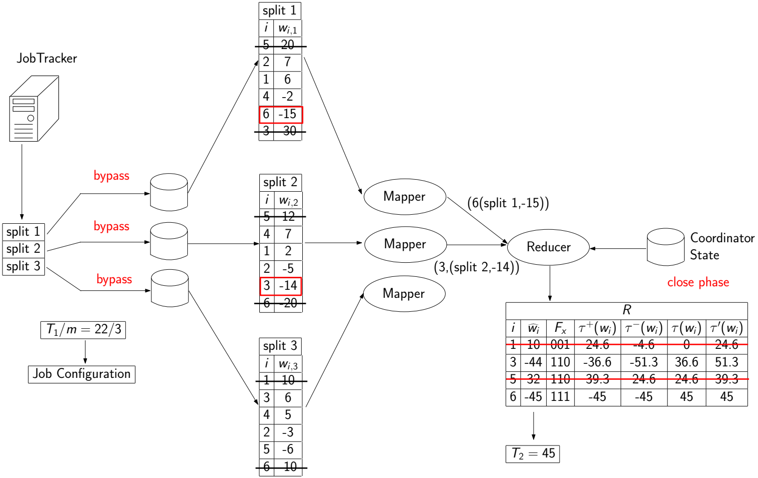


i	$\widehat{w}_i$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$	$\tau'(w_i)$
1	10	001	24.6	-4.6	0	24.6
3	-44	110	-36.6	-51.3	36.6	51.3
5	32	110	39.3	24.6	24.6	39.3
6	-45	111	-45	-45	45	45

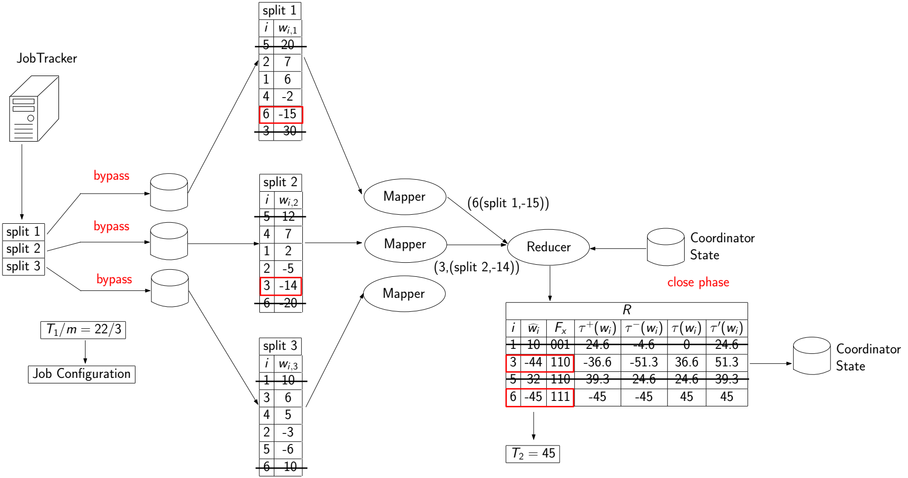
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



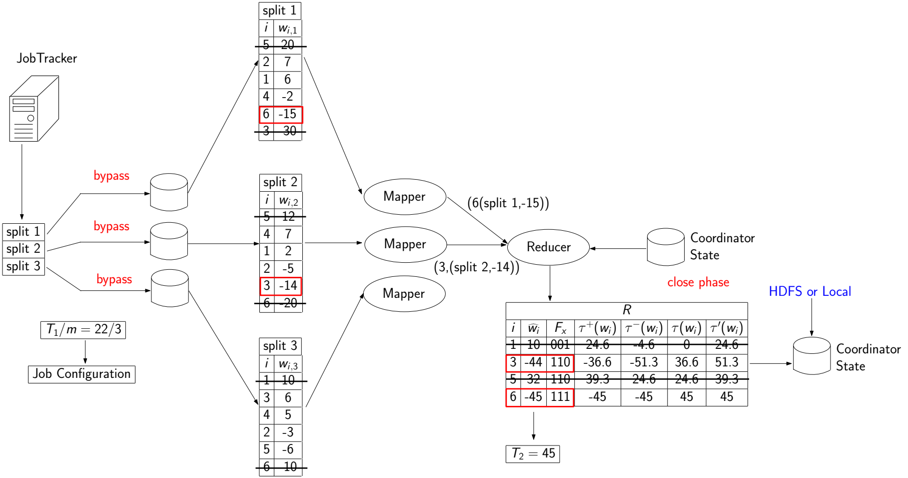
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



Exact Top-k Wavelet Coefficients: Hadoop Phase 3



Exact Top-k Wavelet Coefficients: Hadoop Phase 3



split 1

i	$w_{i,1}$
~~5~~	~~20~~
2	7
1	6
4	-2
6	-15
~~3~~	~~30~~

split 2

i	$w_{i,2}$
~~5~~	~~12~~
4	7
1	2
2	-5
3	-14
~~6~~	~~-20~~

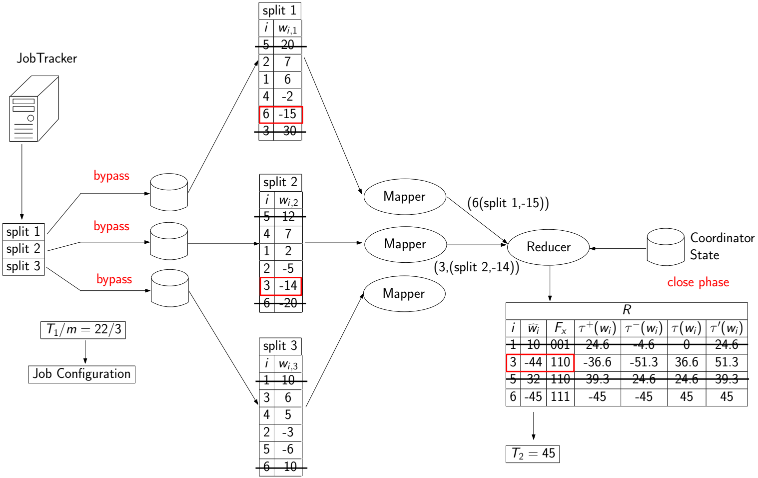
split 3

i	$w_{i,3}$
~~1~~	~~10~~
3	6
4	5
2	-3
5	-6
~~6~~	~~10~~

R

i	$\widehat{w}_i$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$	$\tau'(w_i)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

Exact Top-k Wavelet Coefficients: Hadoop Phase 3



split 1	
i	$w_{i,1}$
~~5~~	~~20~~
2	7
1	6
4	-2
6	-15
~~3~~	~~30~~

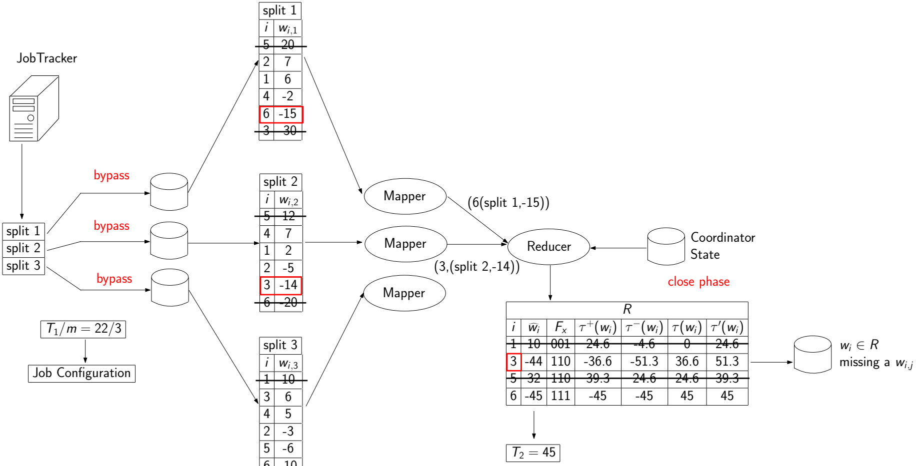
split 2	
i	$w_{i,2}$
~~5~~	~~12~~
4	7
1	2
2	-5
3	-14
~~6~~	~~-20~~

split 3	
i	$w_{i,3}$
~~1~~	~~10~~
3	6
4	5
2	-3
5	-6
~~6~~	~~10~~

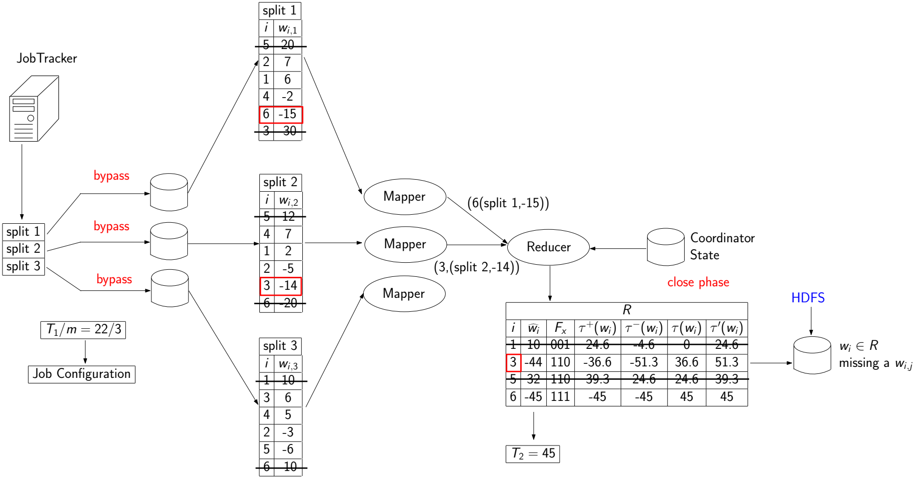
R

i	$\widehat{w_i}$	F_x	$\tau^+(w_i)$	$\tau^-(w_i)$	$\tau(w_i)$	$\tau'(w_i)$
~~1~~	~~10~~	~~001~~	~~24.6~~	~~-4.6~~	~~0~~	~~24.6~~
3	-44	110	-36.6	-51.3	36.6	51.3
~~5~~	~~32~~	~~110~~	~~39.3~~	~~24.6~~	~~24.6~~	~~39.3~~
6	-45	111	-45	-45	45	45

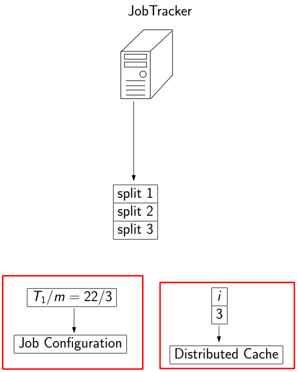
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



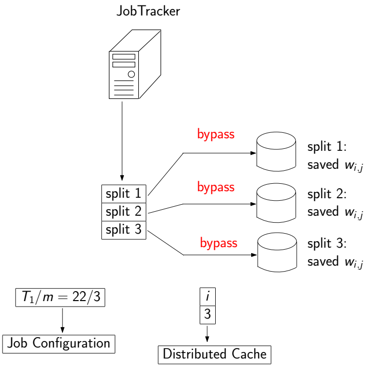
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



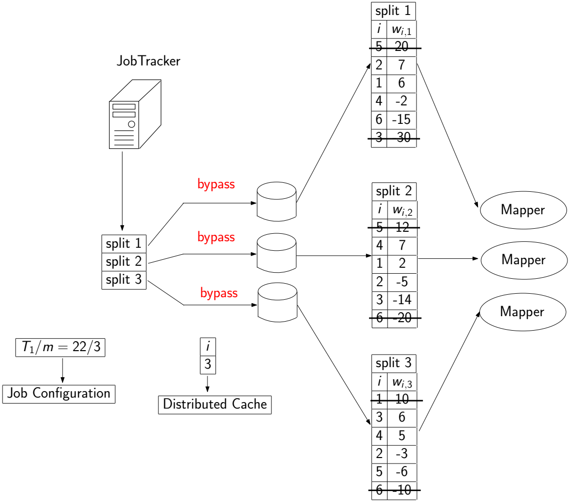
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



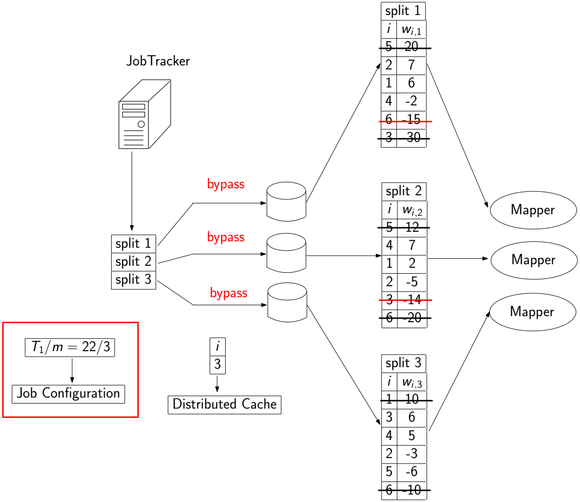
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



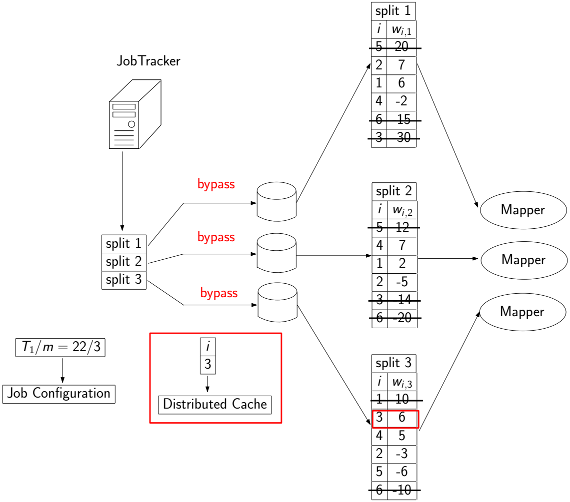
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



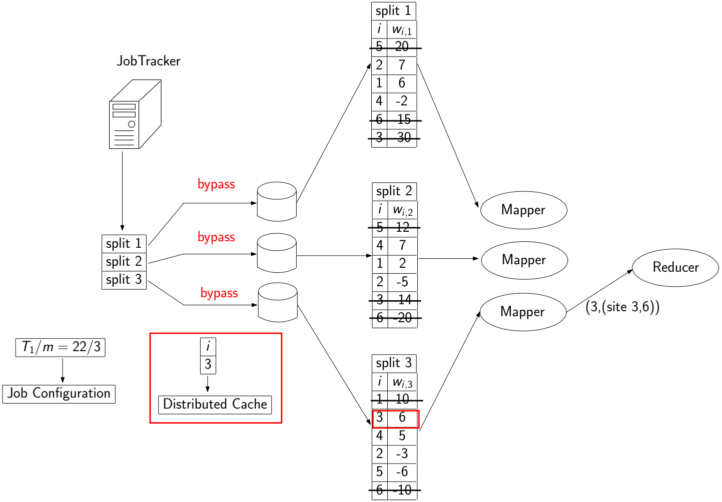
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



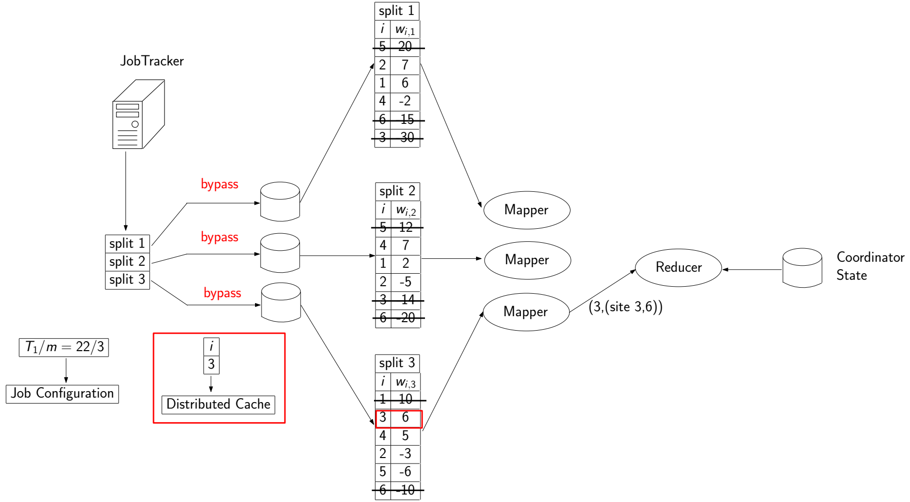
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



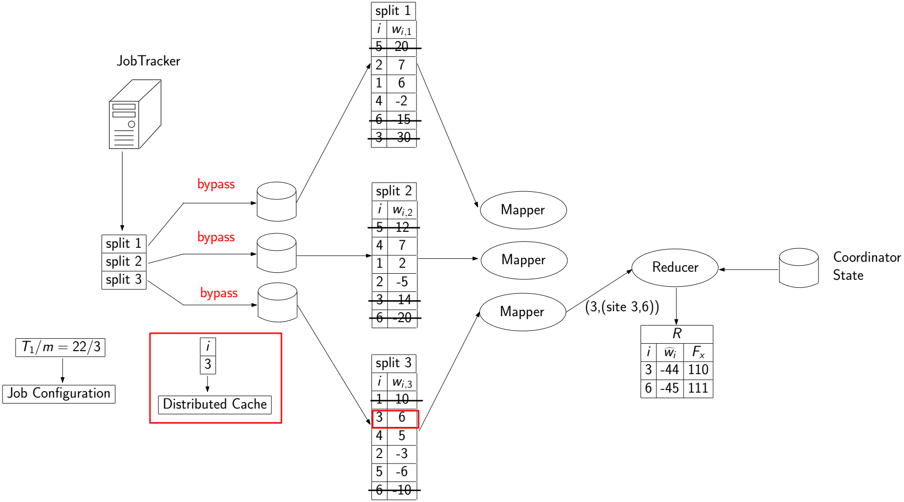
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



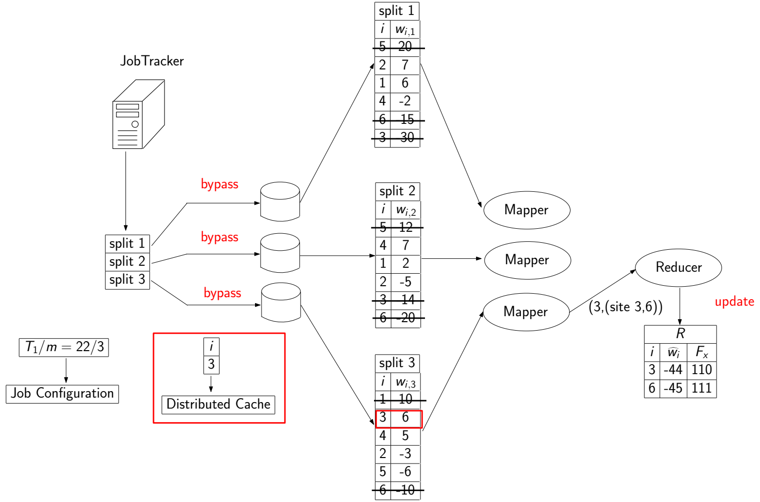
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



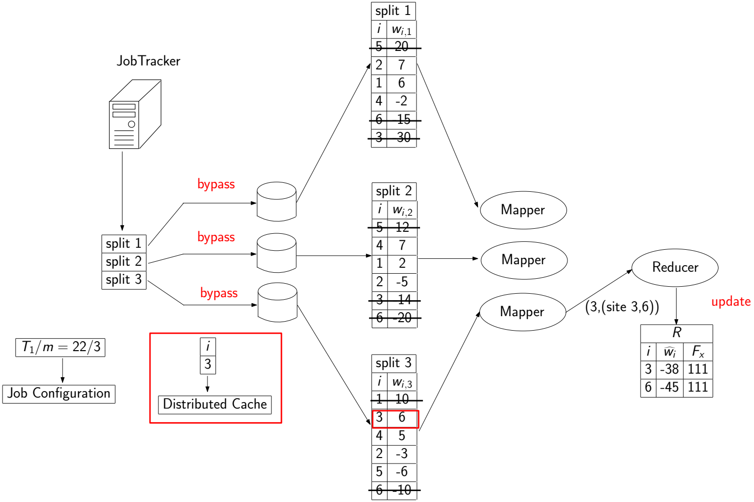
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



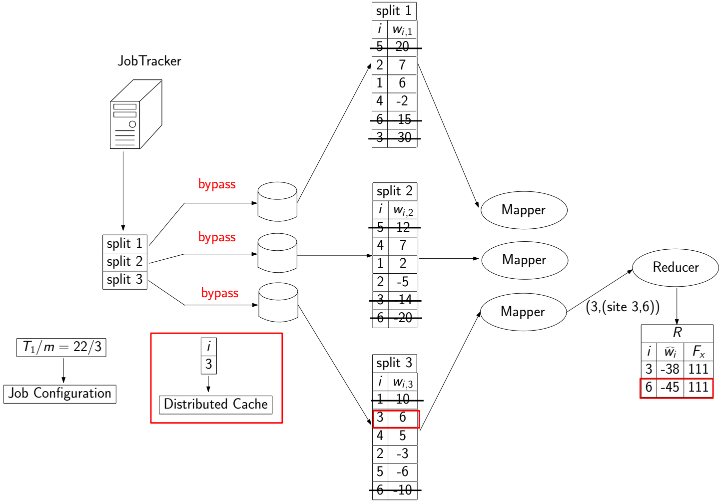
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



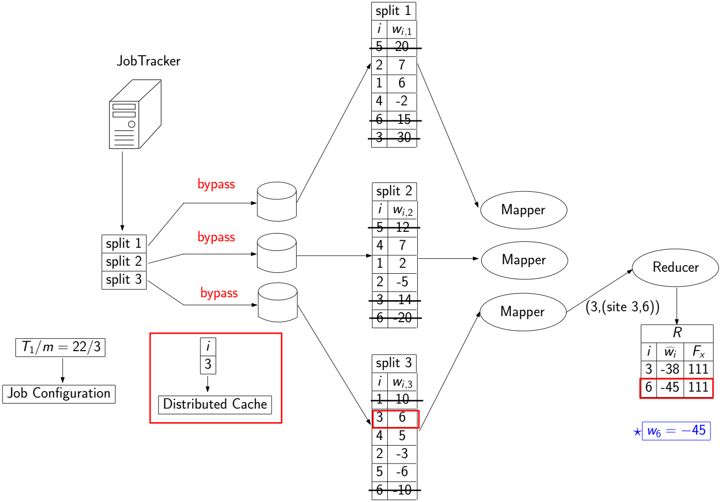
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



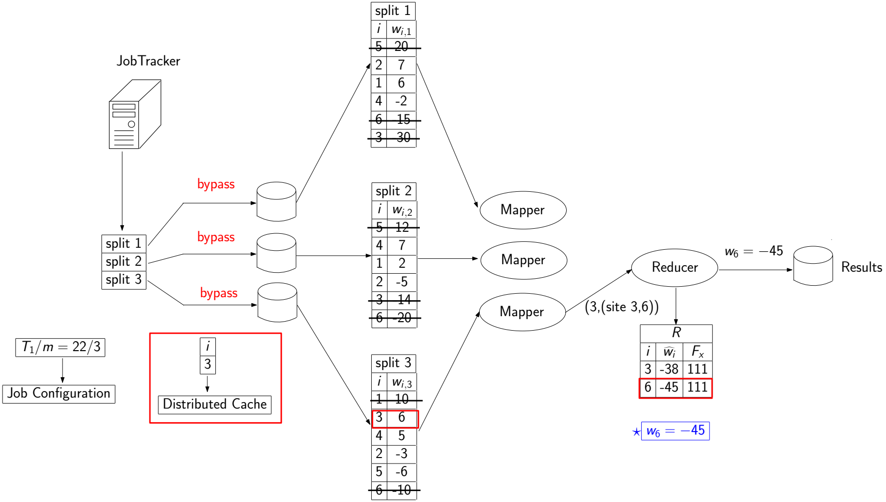
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



Exact Top-k Wavelet Coefficients: Hadoop Phase 3

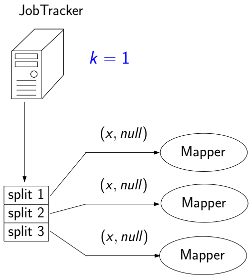


Exact Top-k Wavelet Coefficients: Hadoop Phase 3

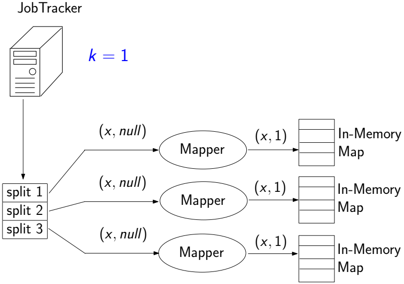


Outline

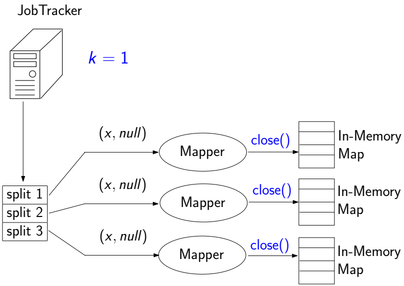
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



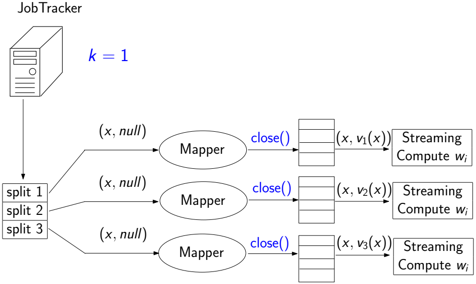
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



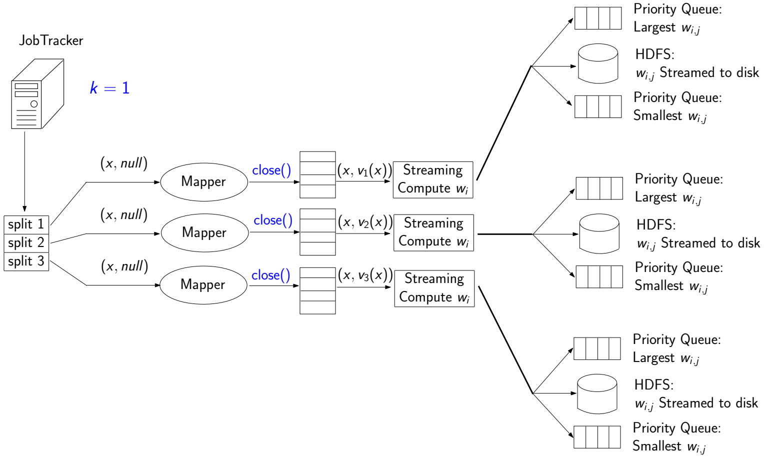
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



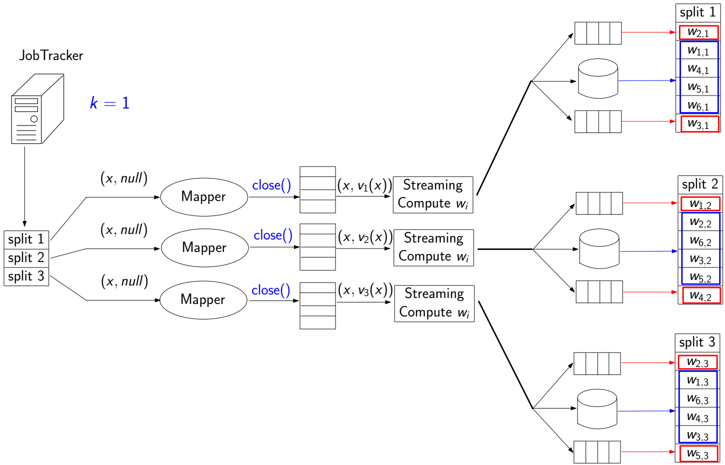
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



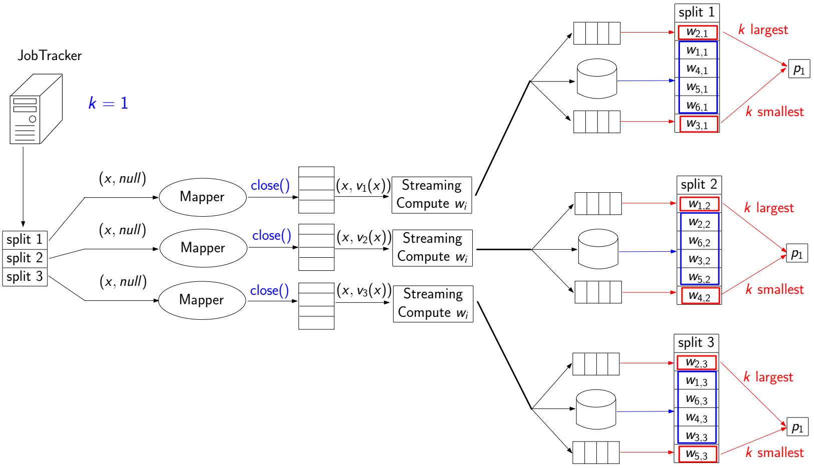
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



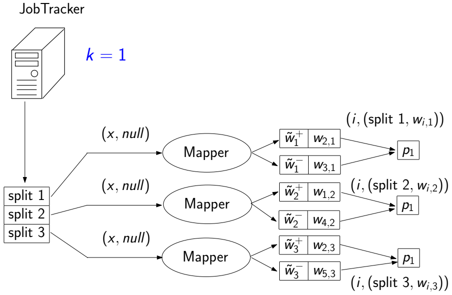
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



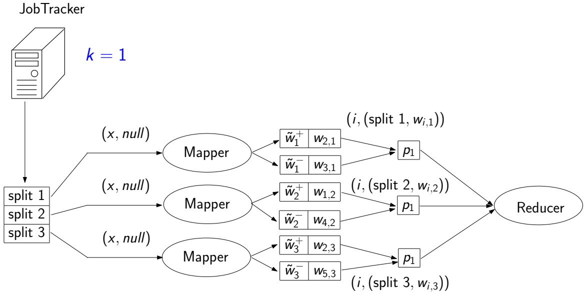
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



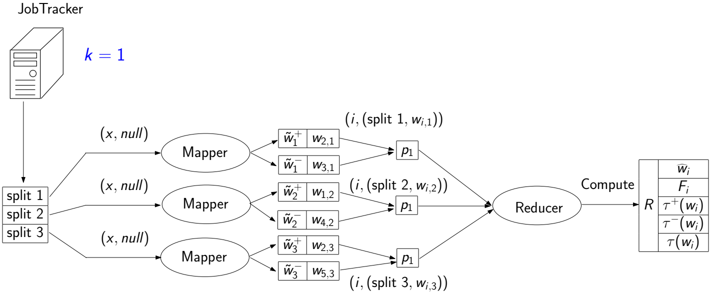
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



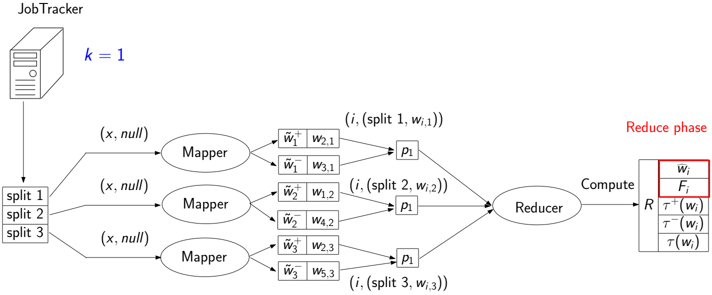
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



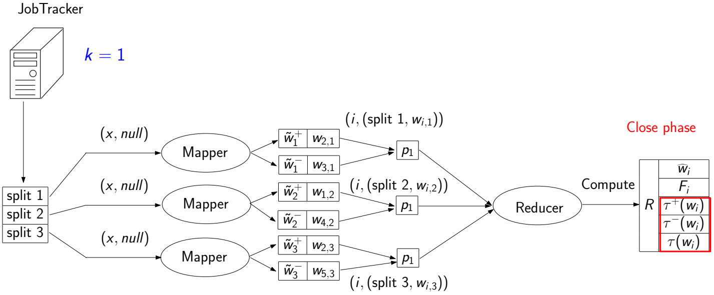
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



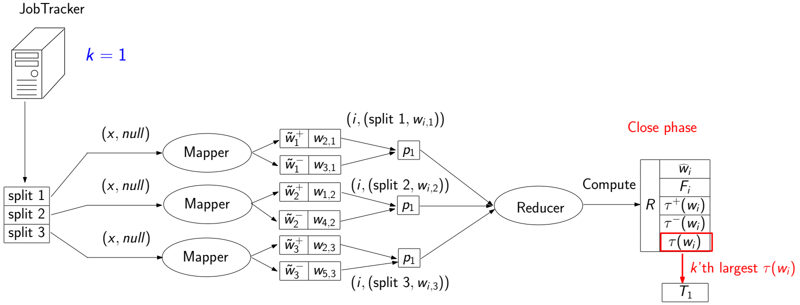
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



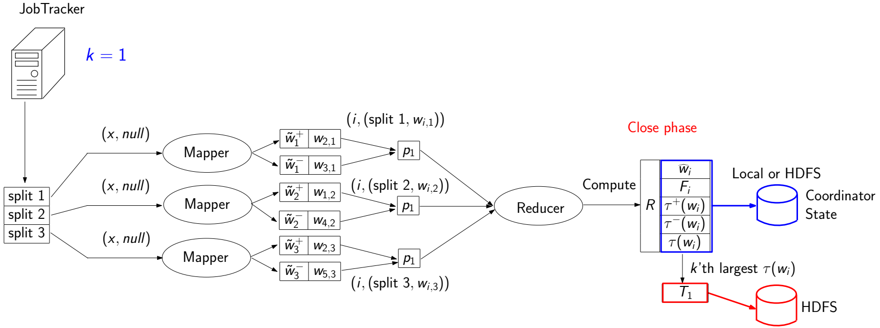
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



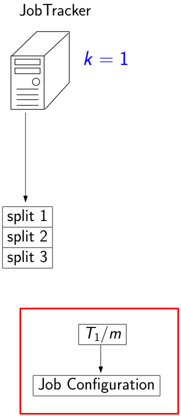
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



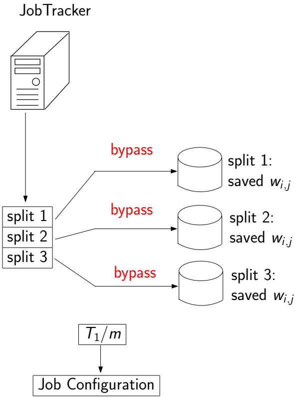
Exact Top-k Wavelet Coefficients: Hadoop Phase 1



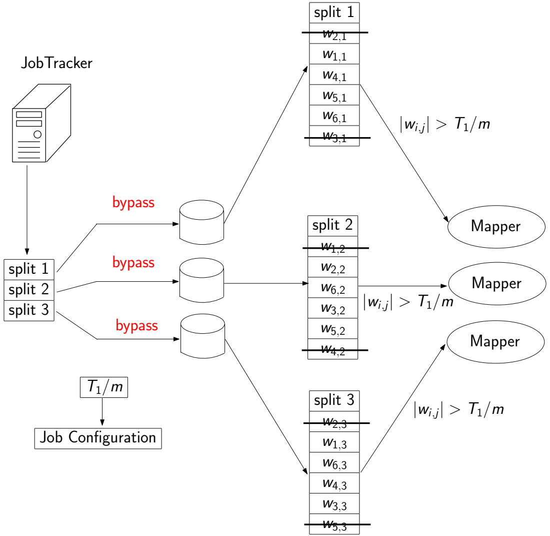
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



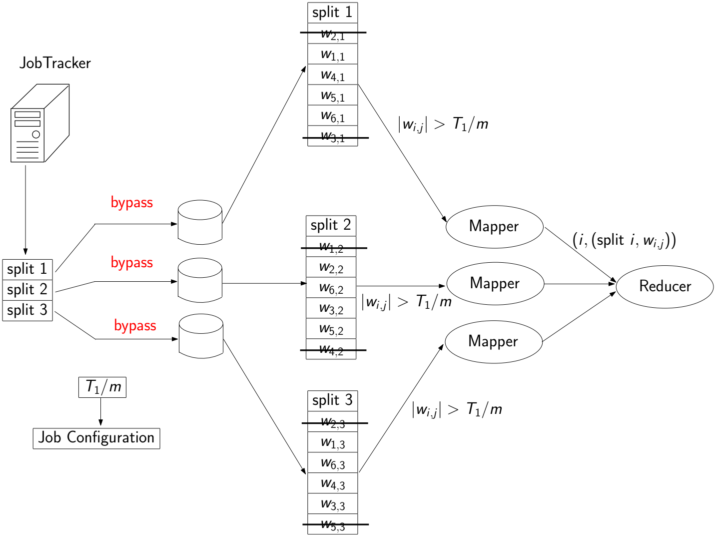
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



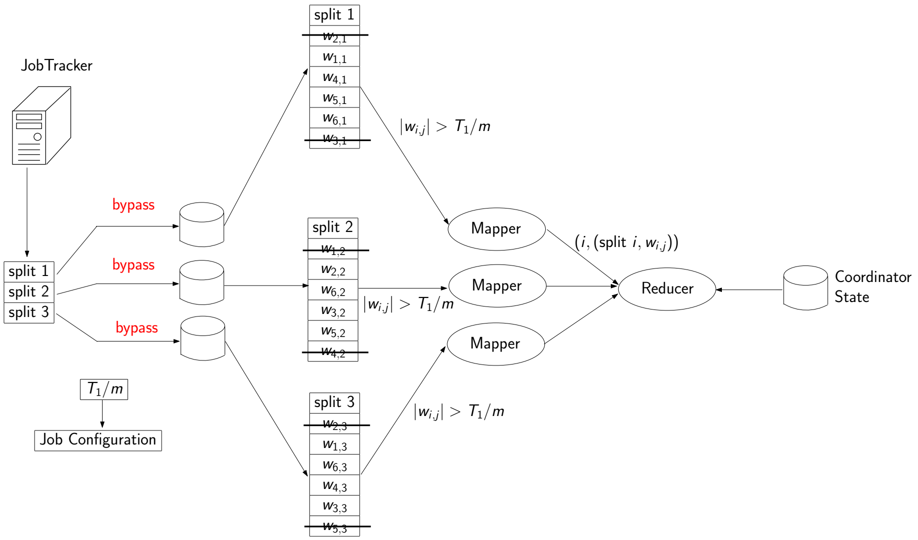
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



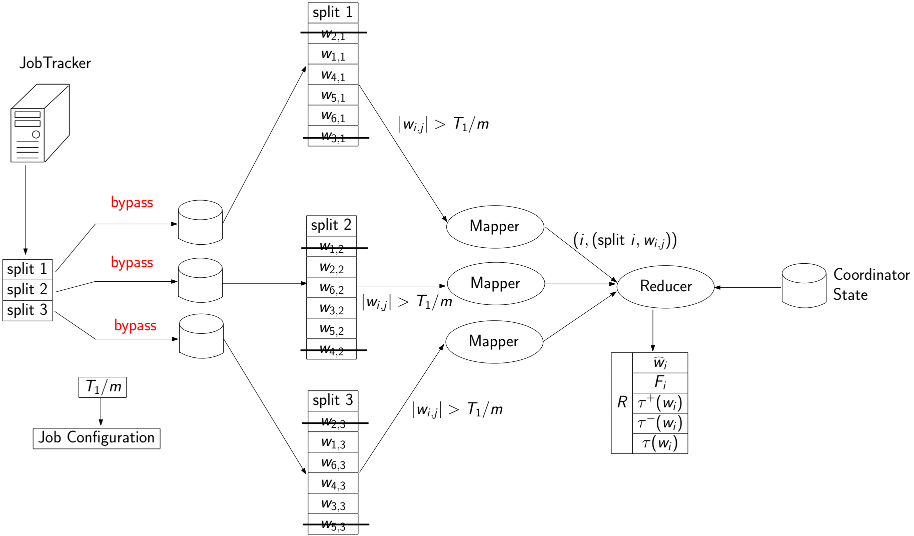
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



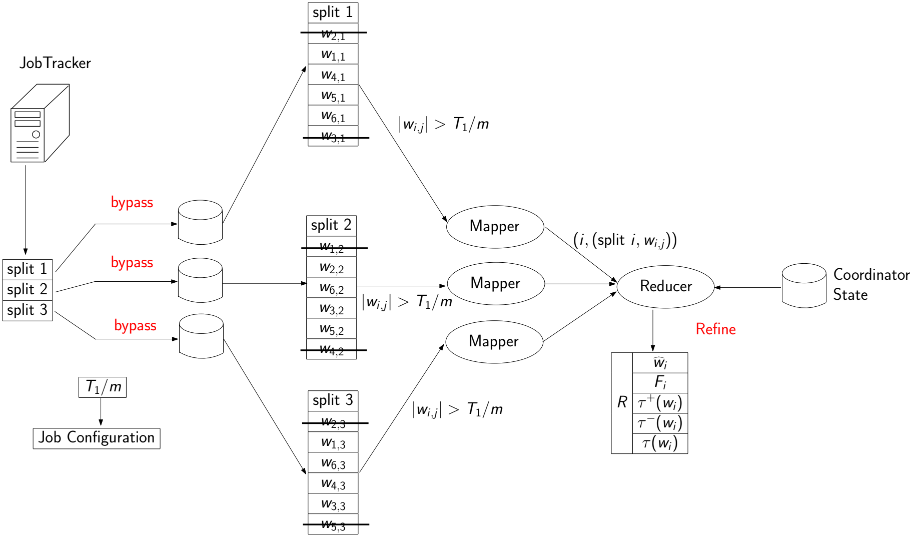
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



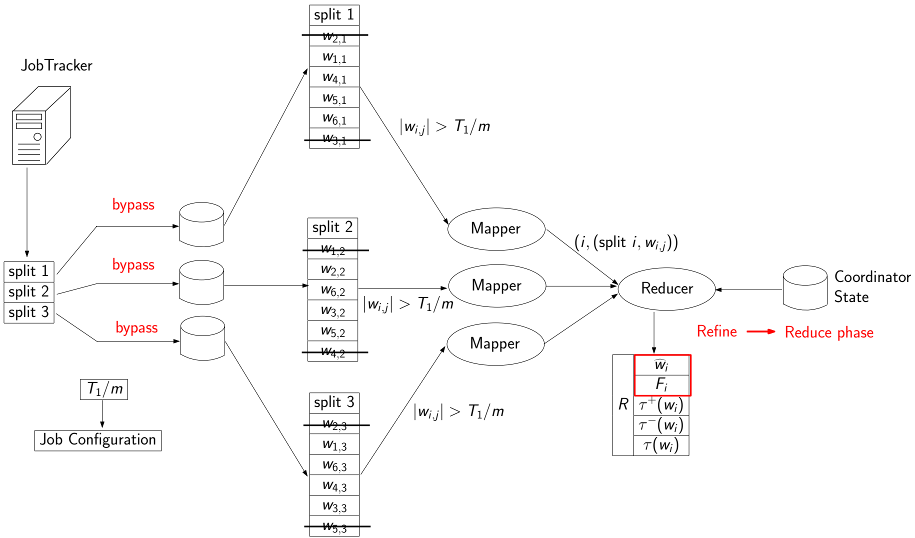
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



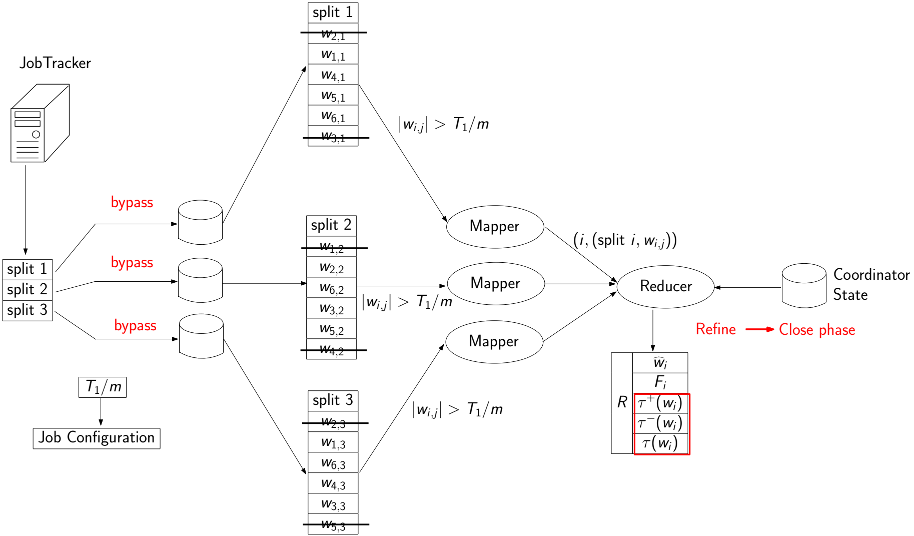
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



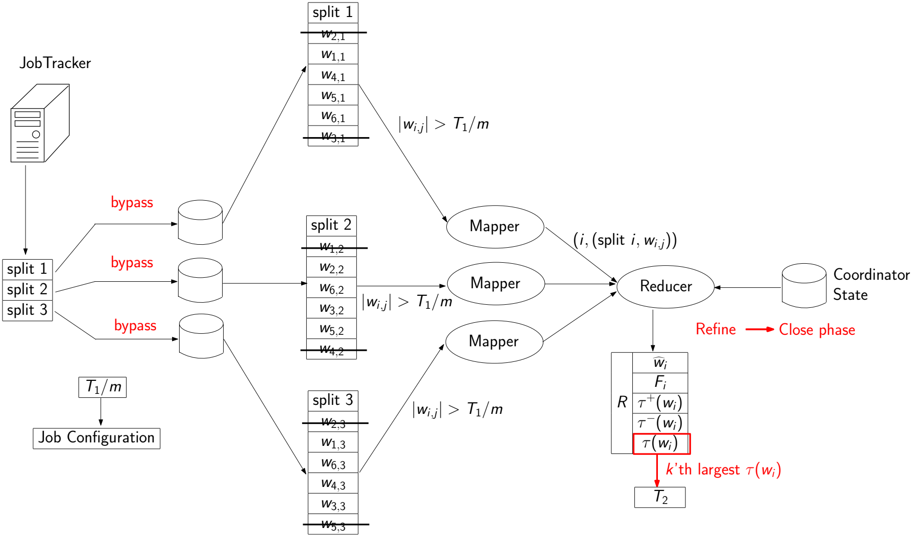
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



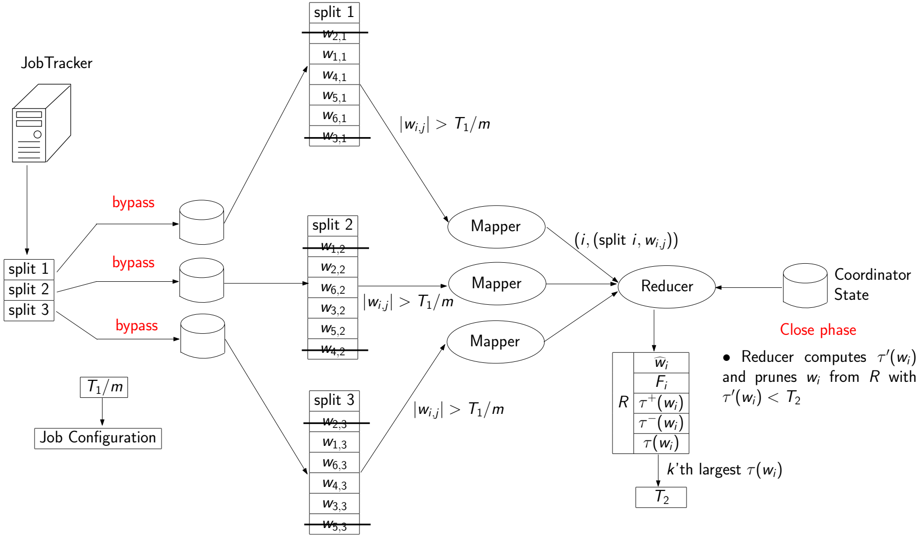
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



Exact Top-k Wavelet Coefficients: Hadoop Phase 2



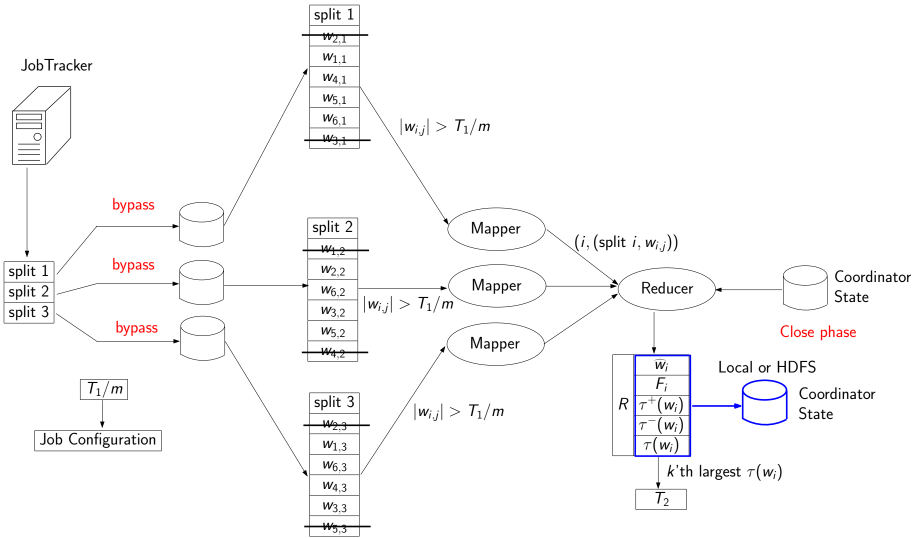
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



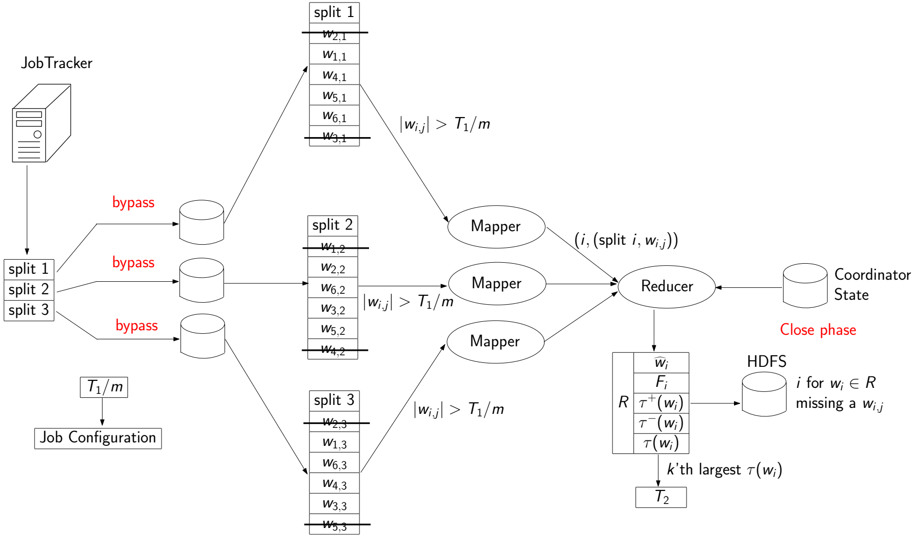
Close phase

- Reducer computes $\tau'(w_i)$ and prunes w_i from R with $\tau'(w_i) < T_2$

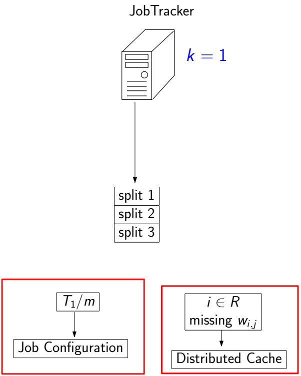
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



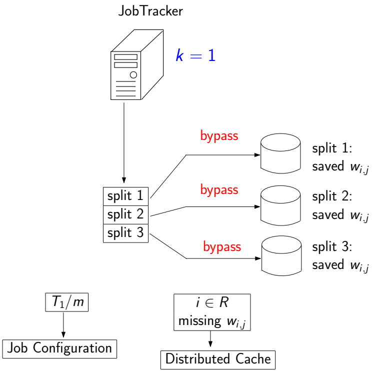
Exact Top-k Wavelet Coefficients: Hadoop Phase 2



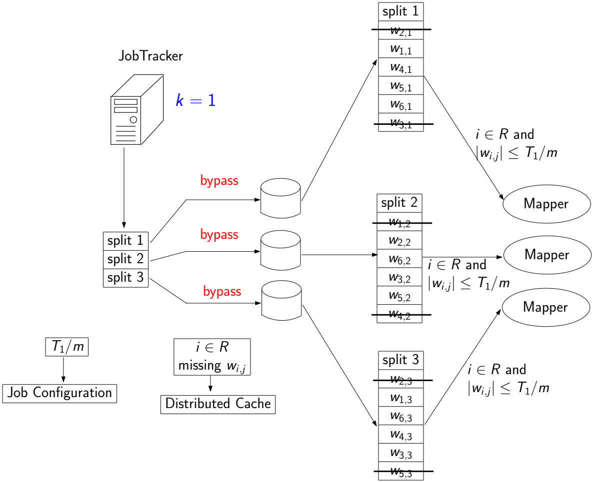
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



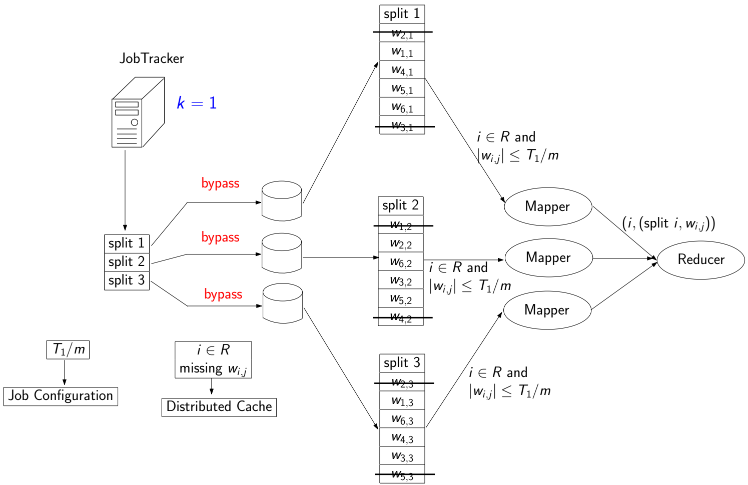
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



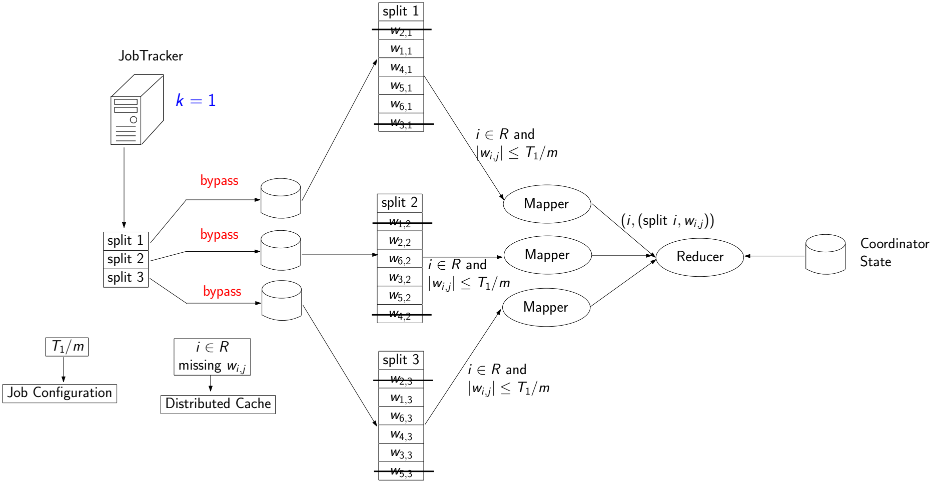
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



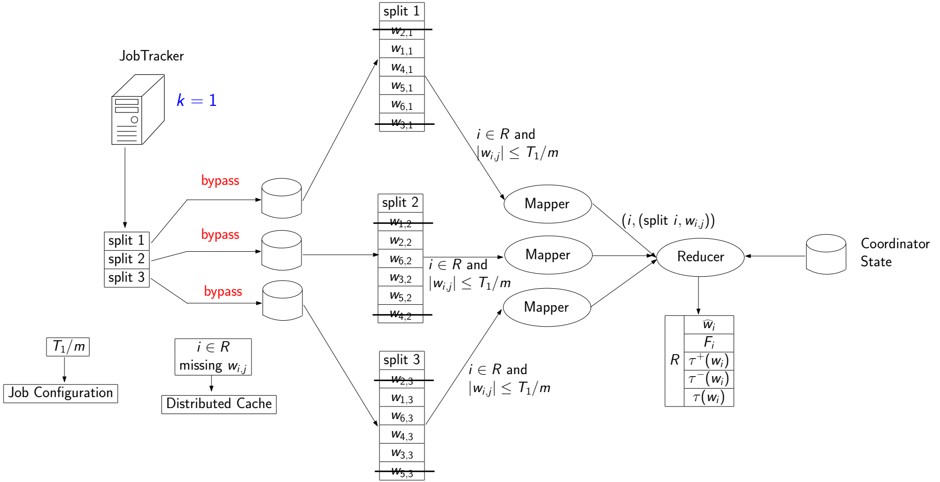
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



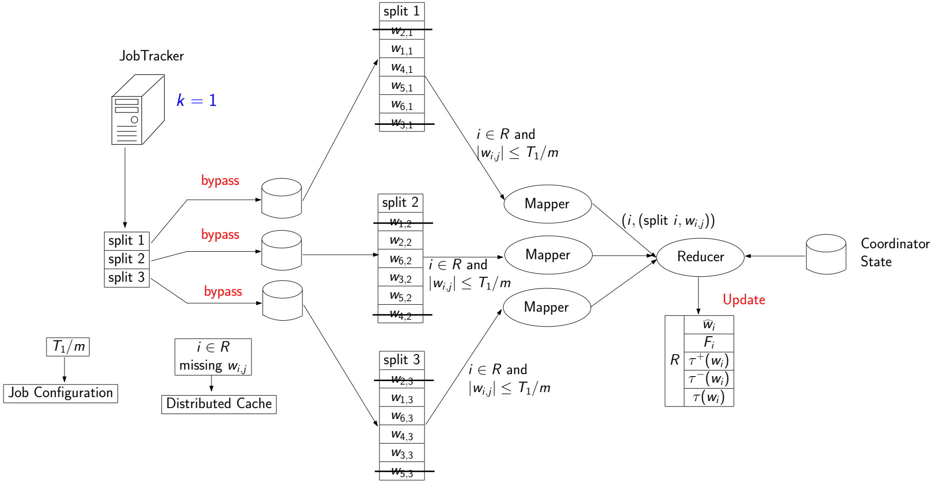
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



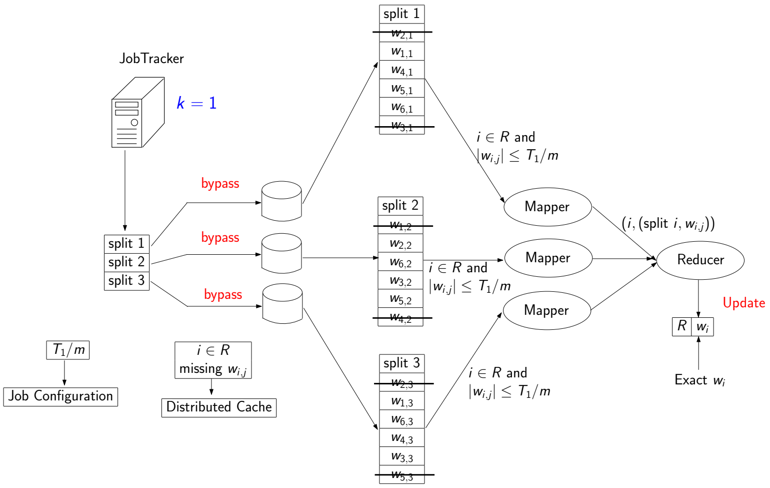
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



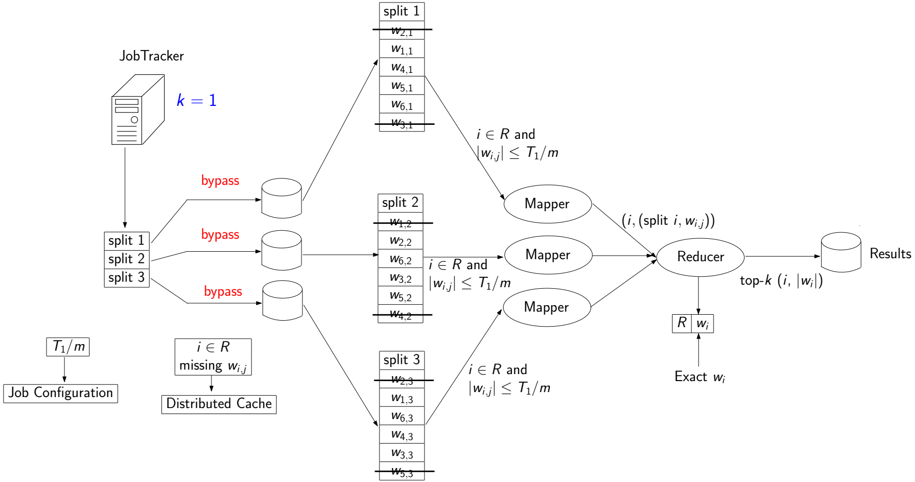
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



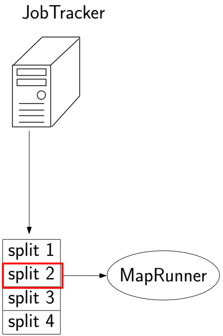
Exact Top-k Wavelet Coefficients: Hadoop Phase 3



Exact Top-k Wavelet Coefficients: Hadoop Phase 3



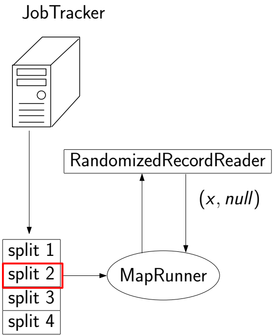
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j= records in split j

s_j= split j sample frequency vector

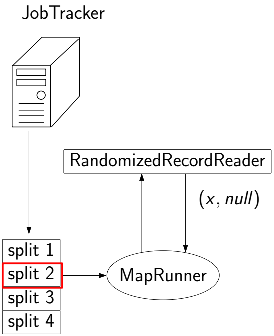
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.

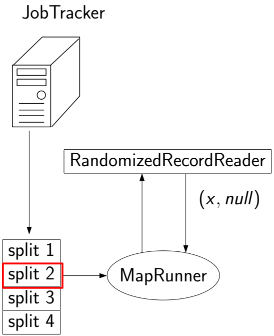
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.

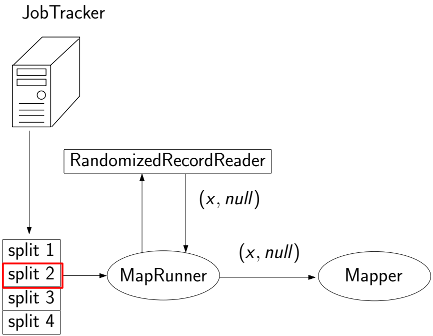
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
    2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

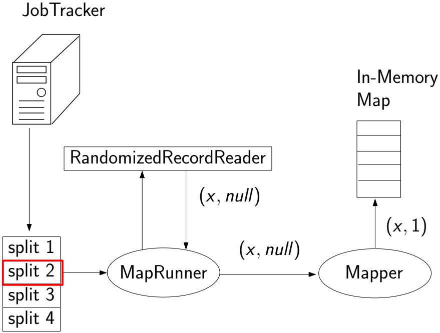
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling

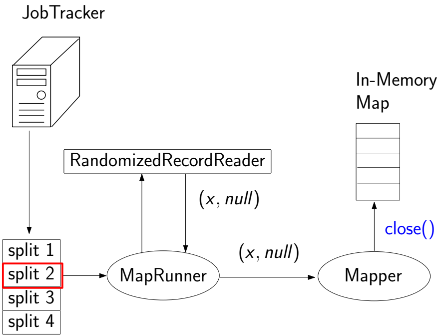


n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

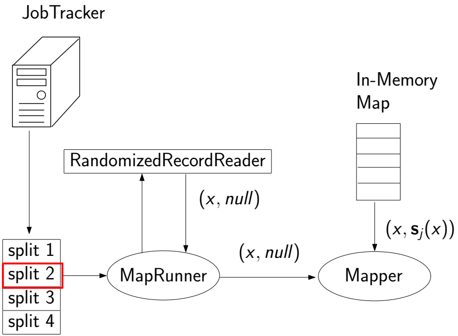
Approximate Top-k Wavelet Coefficients: Basic Random Sampling

n_j= records in split j
s_j= split j sample frequency vector



1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

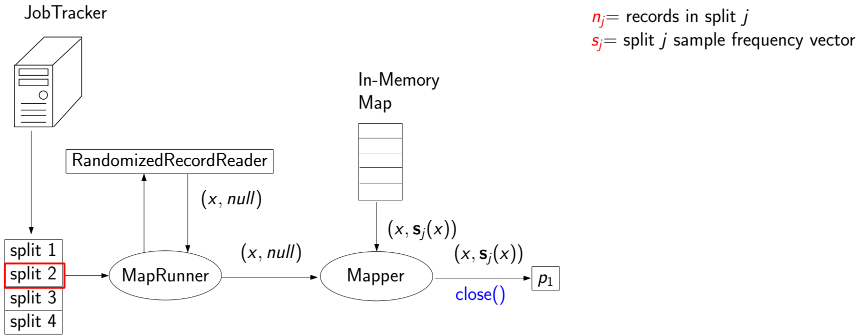
Approximate Top-k Wavelet Coefficients: Basic Random Sampling



n_j= records in split j
s_j= split j sample frequency vector

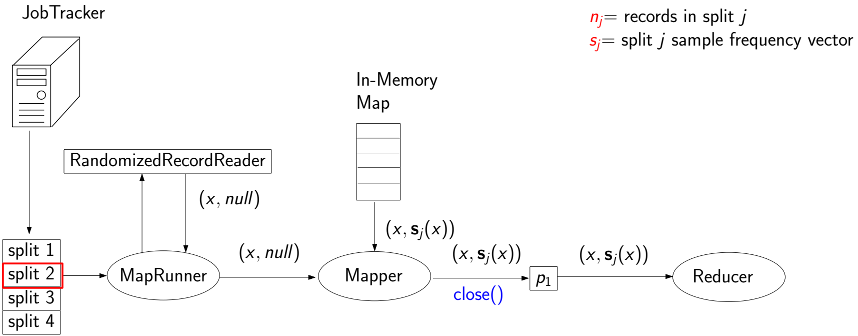
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



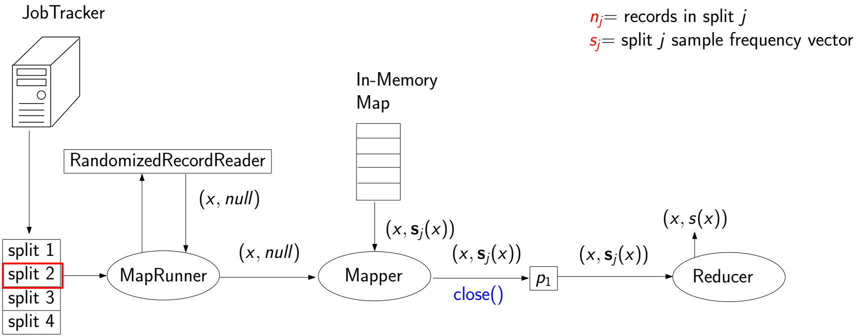
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
    1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
    2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



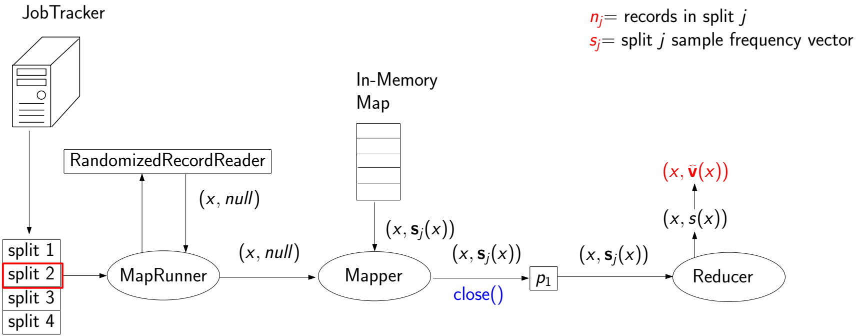
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



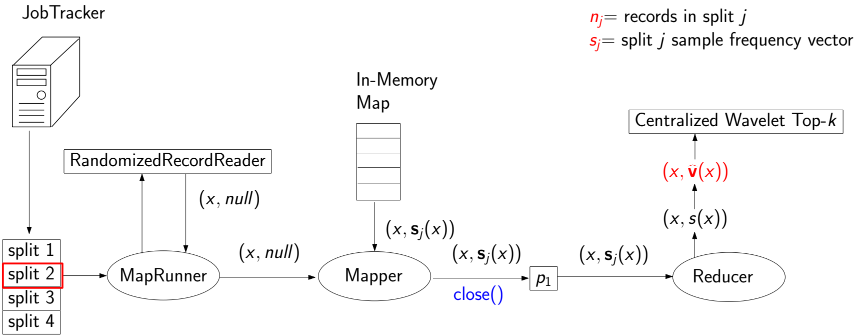
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



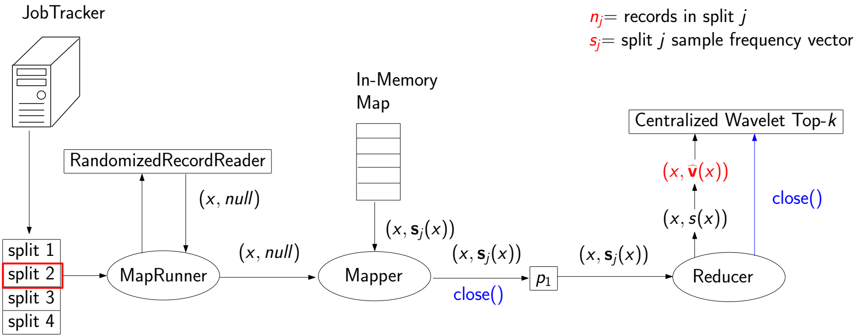
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.
2. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



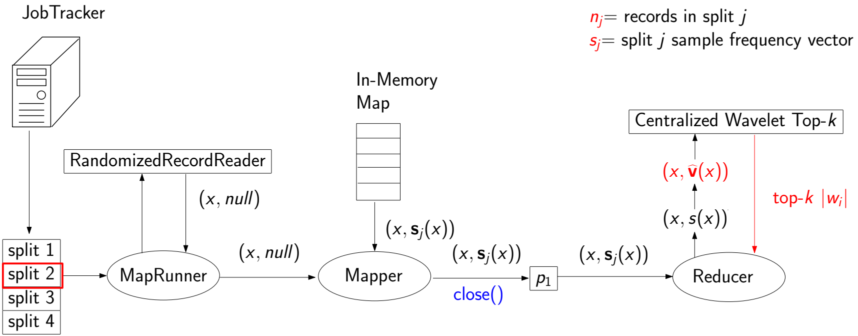
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.
2. Reducer uses $\hat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



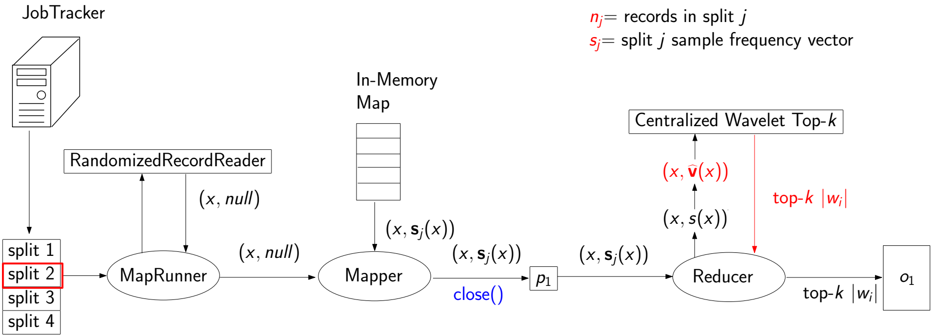
1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.
2. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.
2. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Basic Random Sampling



1. RandomizedRecordReader j (RR_j) samples $n_j/\varepsilon^2 n$ records.
   1. RR_j randomly selects $n_j/\varepsilon^2 n$ offsets in split j.
   2. RR_j sorts the offsets in ascending order then seeks the record at each sampled offset.
2. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our unbiased estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an unbiased estimator of $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an unbiased estimator of $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.
  3. $\mathbf{E}[M] = \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an unbiased estimator of $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.
  3. $\mathbf{E}[M] = \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) = \varepsilon\sqrt{m}(\mathbf{s}(x) - \rho(x))$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.
   - Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
   - Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
     2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
   - Let $M = \sum_{j=1}^{m'} X_j$.
     3. $\mathbf{E}[M] = \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) = \varepsilon\sqrt{m}(\mathbf{s}(x) - \rho(x))$.
4. $\mathbf{E}[\widehat{\mathbf{s}}(x)] = \mathbf{E}[\rho(x) + M/\varepsilon\sqrt{m}]$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.
   - Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
   - Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
     2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
   - Let $M = \sum_{j=1}^{m'} X_j$.
     3. $\mathbf{E}[M] = \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) = \varepsilon\sqrt{m}(\mathbf{s}(x) - \rho(x))$.
4. $\mathbf{E}[\widehat{\mathbf{s}}(x)] = \mathbf{E}[\rho(x) + M/\varepsilon\sqrt{m}] = \rho(x) + (\mathbf{s}(x) - \rho(x))$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an unbiased estimator of* $\mathbf{s}(x)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.
   - Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
   - Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
     2. $\mathbf{E}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
   - Let $M = \sum_{j=1}^{m'} X_j$.
     3. $\mathbf{E}[M] = \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) = \varepsilon\sqrt{m}(\mathbf{s}(x) - \rho(x))$.
4. $\mathbf{E}[\widehat{\mathbf{s}}(x)] = \mathbf{E}[\rho(x) + M/\varepsilon\sqrt{m}] = \rho(x) + (\mathbf{s}(x) - \rho(x)) = \mathbf{s}(x)$.

□

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$*.*

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x))$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
    2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.
  3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j]$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.
  3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ *is an estimator of* $\mathbf{s}(x)$ *with standard deviation at most* $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.

2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

- Let $M = \sum_{j=1}^{m'} X_j$.

3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m})$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.

2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

- Let $M = \sum_{j=1}^{m'} X_j$.

3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m}) = m'$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.

2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m}\cdot\mathbf{s}_j(x)(1-\varepsilon\sqrt{m}\cdot\mathbf{s}_j(x)) \leq \varepsilon\sqrt{m}\cdot\mathbf{s}_j(x)$.

- Let $M = \sum_{j=1}^{m'} X_j$.

3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'}\mathbf{Var}[X_j] \leq \sum_{j=1}^{m'}\varepsilon\sqrt{m}\cdot\mathbf{s}_j(x) \leq m'\cdot\varepsilon\sqrt{m}\cdot 1/(\varepsilon\sqrt{m}) = m'$.

4. $Var[\widehat{\mathbf{s}}(x)] = \mathbf{Var}[M/\varepsilon\sqrt{m}]$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.
  3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m})$ $= m'$.

4. $Var[\widehat{\mathbf{s}}(x)] = \mathbf{Var}[M/\varepsilon\sqrt{m}] = \mathbf{Var}[M]/\varepsilon^2 m$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
  2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
- Let $M = \sum_{j=1}^{m'} X_j$.
  3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m}) = m'$.

4. $Var[\widehat{\mathbf{s}}(x)] = \mathbf{Var}[M/\varepsilon\sqrt{m}] = \mathbf{Var}[M]/\varepsilon^2 m \leq m'/\varepsilon^2 m$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.
   - Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
   - Let $X_j = 1$ if x is sampled in split j and 0 otherwise.
2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.
   - Let $M = \sum_{j=1}^{m'} X_j$.
3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m})$
   $= m'$.
4. $Var[\widehat{\mathbf{s}}(x)] = \mathbf{Var}[M/\varepsilon\sqrt{m}] = \mathbf{Var}[M]/\varepsilon^2 m \leq m'/\varepsilon^2 m \leq 1/\varepsilon^2$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\widehat{\mathbf{s}}(x)$ is an estimator of $\mathbf{s}(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- Assume in the first m' splits $\mathbf{s}_j(x) < 1/(\varepsilon\sqrt{m})$.
- Let $X_j = 1$ if x is sampled in split j and 0 otherwise.

2. $\mathbf{Var}[X_j] = \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)(1 - \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)) \leq \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x)$.

- Let $M = \sum_{j=1}^{m'} X_j$.

3. $\mathbf{Var}[M] \leq \sum_{j=1}^{m'} \mathbf{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m}) = m'$.

4. $Var[\widehat{\mathbf{s}}(x)] = \mathbf{Var}[M/\varepsilon\sqrt{m}] = \mathbf{Var}[M]/\varepsilon^2 m \leq m'/\varepsilon^2 m \leq 1/\varepsilon^2 \leq 1/\varepsilon$.

□

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.
   2. There are $\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon$ such keys.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.
  2. There are $\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon$ such keys.
- If $s_j(x) < 1/(\varepsilon\sqrt{m})$,
  we emit $(x, null)$ with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j x$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.
  2. There are $\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon$ such keys.
- If $s_j(x) < 1/(\varepsilon\sqrt{m})$,
  we emit $(x, null)$ with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j x$.
  3. On expectation there are,
     $\sum_j \sum_x \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq \varepsilon\sqrt{m} \cdot 1/\varepsilon^2$

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.
  2. There are $\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon$ such keys.
- If $s_j(x) < 1/(\varepsilon\sqrt{m})$,
  we emit (x, null) with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j x$.
  3. On expectation there are,
     $\sum_j \sum_x \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq \varepsilon\sqrt{m} \cdot 1/\varepsilon^2 = \sqrt{m}/\varepsilon$.

Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

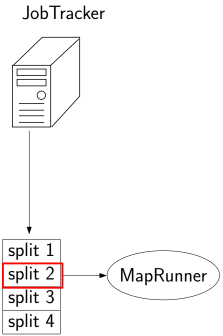
Proof.

1. Our estimator is $\widehat{\mathbf{s}}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

- The first-level sample size is $pn = 1/\varepsilon^2$.
- If $\mathbf{s}_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.
  2. There are $\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon$ such keys.
- If $s_j(x) < 1/(\varepsilon\sqrt{m})$,
  we emit (x, null) with probability $\varepsilon\sqrt{m} \cdot \mathbf{s}_j x$.
  3. On expectation there are,
     $\sum_j \sum_x \varepsilon\sqrt{m} \cdot \mathbf{s}_j(x) \leq \varepsilon\sqrt{m} \cdot 1/\varepsilon^2 = \sqrt{m}/\varepsilon$.
- By (2) and (3), the total number of emitted keys is $O(\sqrt{m}/\varepsilon)$.

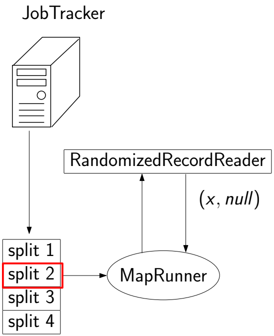
□

Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j= records in split j
s_j= split j sample frequency vector

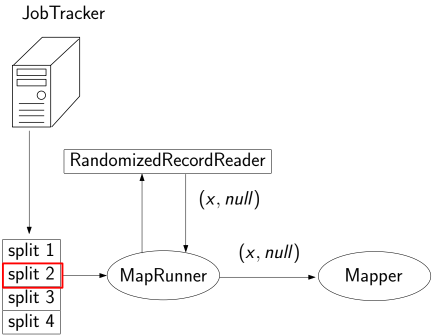
Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.

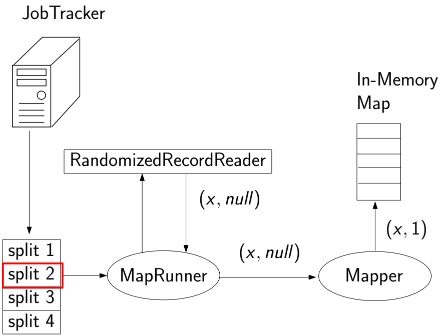
Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.

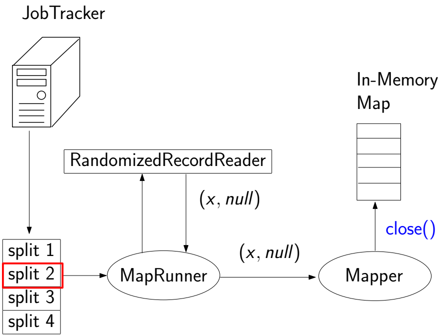
Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.

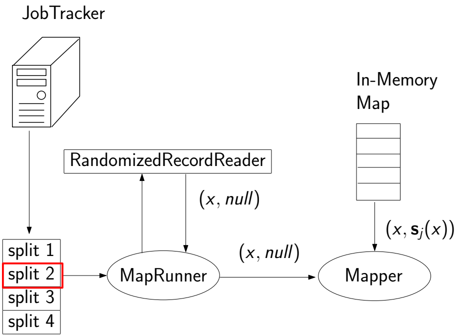
Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j= records in split j
s_j= split j sample frequency vector

1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.

Approximate Top-k Wavelet Coefficients: Improved Sampling

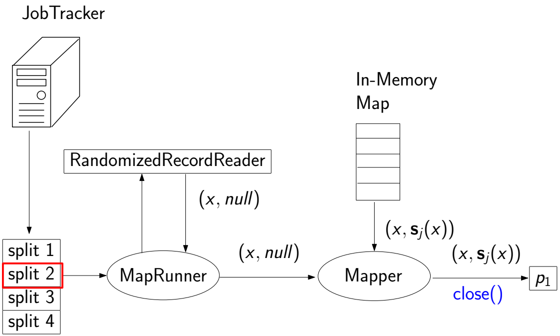


n_j= records in split j

s_j= split j sample frequency vector

1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.

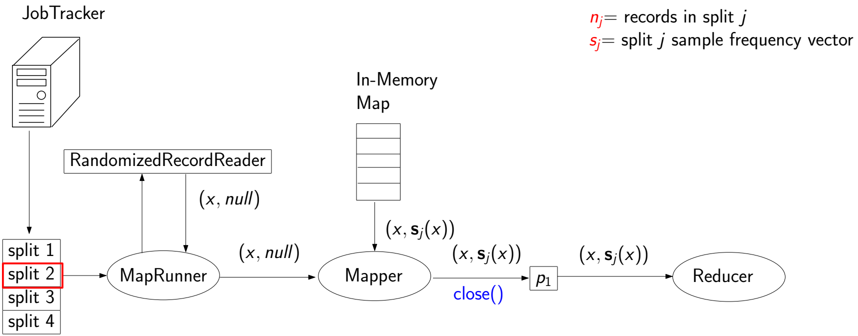
Approximate Top-k Wavelet Coefficients: Improved Sampling



n_j= records in split j
s_j= split j sample frequency vector

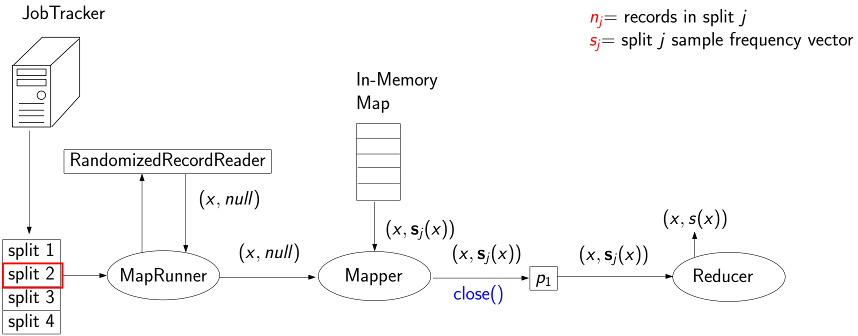
1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



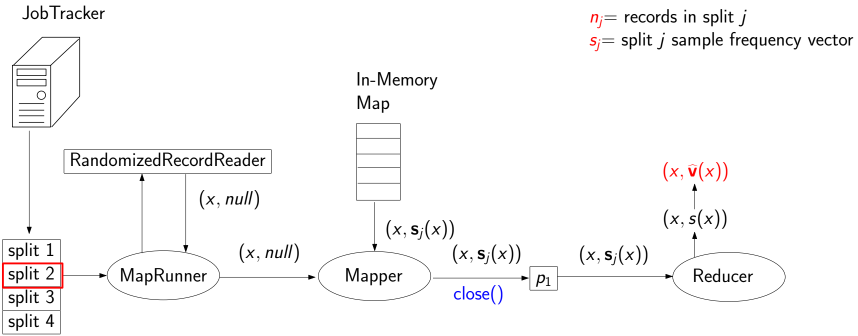
1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



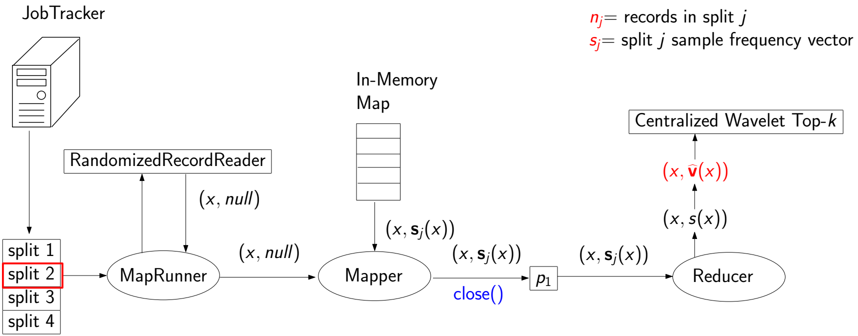
1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



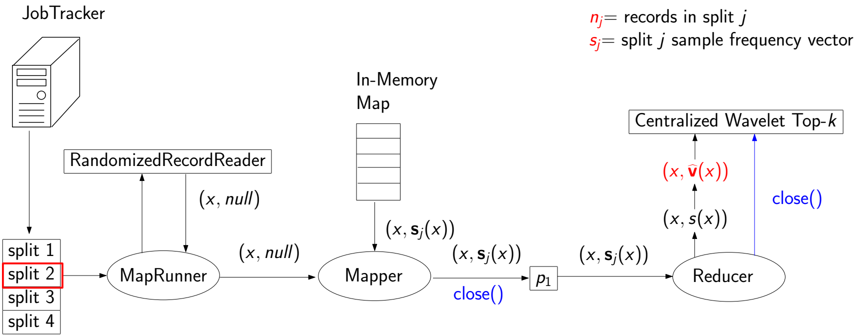
1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.
3. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



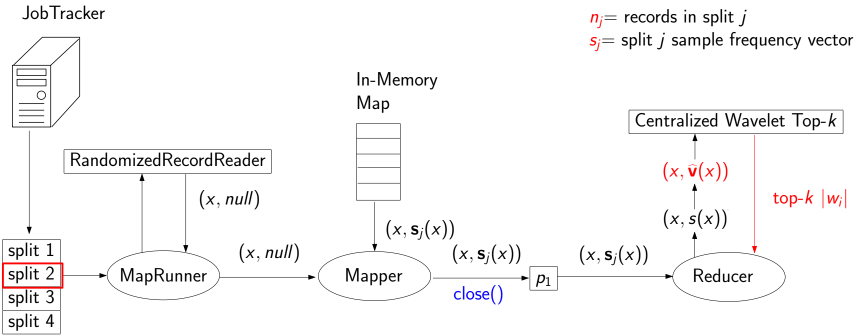
1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.
3. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



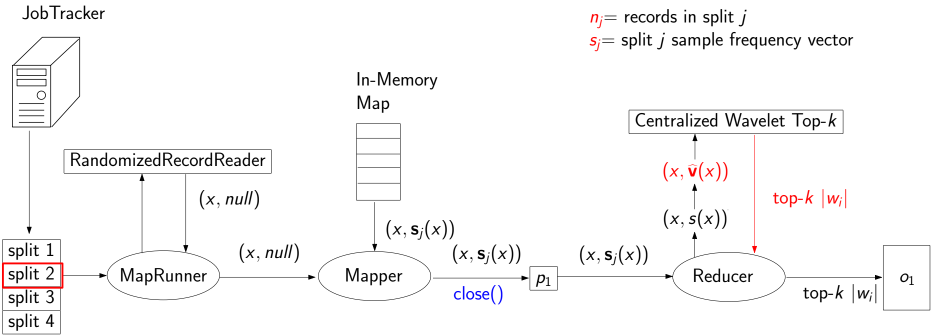
1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.
3. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.
3. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our estimator for $\mathbf{v}(x)$.

Approximate Top-k Wavelet Coefficients: Improved Sampling



1. RandomizedRecordReader j samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $\mathbf{s}_j(x) > \varepsilon t_j$, the Mapper emits $(x, \mathbf{s}_j(x))$.
3. Reducer uses $\widehat{\mathbf{v}}(x) = \mathbf{s}(x)/p$, our estimator for $\mathbf{v}(x)$.