A Novel Approach for Approximate Aggregations Over Arrays

Yi Wang, Yu Su, Gagan Agrawal

The Ohio State University

SSDBM 2015

June 29th, San Diego, California

Outline

- Introduction
- Aggregation Method
- V-Optimized Binning
- Experimental Results
- Conclusion

Motivation

- Aggregation is a Key Functionality
- Approximate Aggregation
  - 100% accuracy may be unnecessary
    - E.g., real-time decision making
  - Based on a summary structure/data synopsis
    - Sample, histogram, wavelet, etc.
  - Well studied on relational data but not array data
    - Each array element has not only a value but also a position (subscript)

Challenges in Approximate Aggregations over Array Data

- Flexible Aggregation Over Any Subset
  - Dim-based predicate: *depth < 5*
  - Val-based predicate: *temp < 4*
  - Combined predicate: *depth < 5 AND temp < 4*
- Aggregation Accuracy
  - Value distribution
  - Spatial distribution (not in relational data)
- Aggregation without Data Reorganization
  - Should avoid extra processing and storage costs

Existing Techniques and Limitations

- All Problematic for Array Data
  - Sample-Based
    - Unable to capture both distributions
  - Histogram-Based
    - No spatial distribution
  - Wavelet-Based
    - No value distribution (by mapping a data cube to an array)
    - Restricted to SUM, COUNT, and AVG
- Need for Another Data Synopsis
  - We choose **bitmap indices**

Outline

- Introduction
- Aggregation Method
- V-Optimized Binning
- Experimental Results
- Conclusion

Bitmap Indexing and Pre-Aggregation

- Bitmap Indices

ID	Value	e_0	e_1	e_2	e_3	e_4	e_5	i_0	i_1	i_2
		=1	=2	=3	=4	=5	=6	[1, 2]	[3, 4]	[5, 6]
0	5	0	0	0	0	1	0	0	0	1
1	4	0	0	0	1	0	0	0	1	0
2	2	0	1	0	0	0	0	1	0	0
3	5	0	0	0	0	1	0	0	0	1
4	6	0	0	0	0	0	1	0	0	1
5	1	1	0	0	0	0	0	1	0	0
6	3	0	0	1	0	0	0	0	1	0
7	1	1	0	0	0	0	0	1	0	0
Dataset		Low Level Indices						High Level Indices		

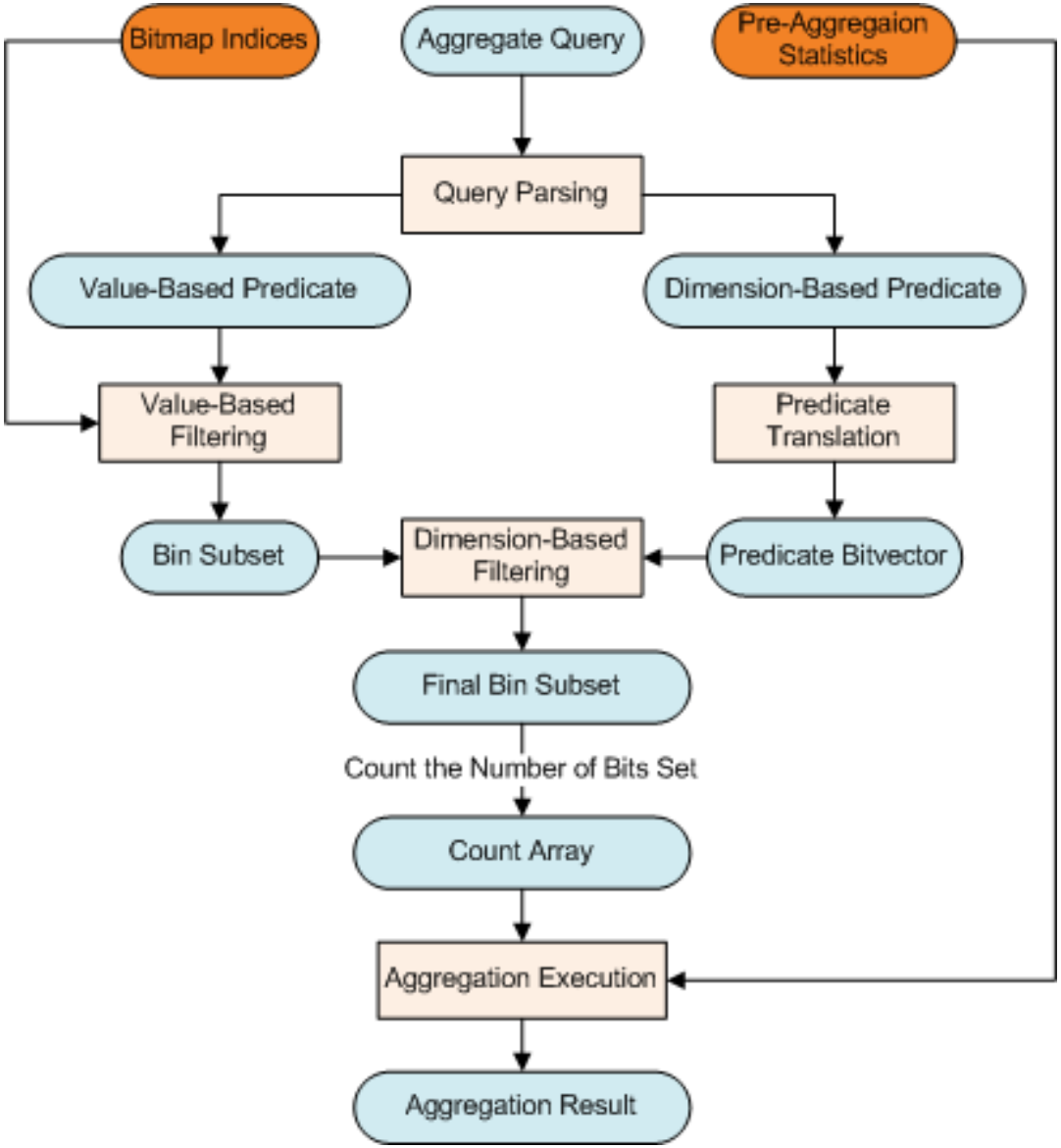
- Pre-Aggregation Statistics

Statistics \ Bin #	i_0	i_1	i_2
COUNT	3	2	3
SUM	4	7	16
SQUARED SUM	6	25	86
MIN	1	3	5
MAX	2	4	6
User-Defined Pre-Aggregation Statistics			

Key Insight

- Bitvectors
  - Preserve spatial info for a bin
    - Any subarea can be represented as a bitvector
  - Support Fast bitwise operations on its compressed format
    - AND, OR, COUNT, etc.
- Pre-Aggregation Statistics
  - Preserve value info for each bin
    - Equivalent to a histogram
  - Cheap
    - No extra data scan: generated during bitmap index generation

Approximate Aggregation Workflow



Example

- Bitmap Indices

SELECT SUM(Array) WHERE Value > 3 AND ID < 4;

ID	Value	e_0	e_1	e_2	e_3	e_4	e_5	i_0	i_1	i_2
		=1	=2	=3	=4	=5	=6	[1, 2]	[3, 4]	[5, 6]
0	5	0	0	0	0	1	0	0	0	1
1	4	0	0	0	1	0	0	0	1	0
2	2	0	1	0	0	0	0	1	0	0
3	5	0	0	0	0	1	0	0	0	1
4	6	0	0	0	0	0	1	0	0	1
5	1	1	0	0	0	0	0	1	0	0
6	3	0	0	1	0	0	0	0	1	0
7	1	1	0	0	0	0	0	1	0	0
Dataset		Low Level Indices						High Level Indices		

Predicate Bitvector: 11110000

i_1': 01000000
i_2': 10010000

- Pre-Aggregation Statistics

Statistics \ Bin #	i_0	i_1	i_2
COUNT	3	2	3
SUM	4	7	16
SQUARED SUM	6	25	86
MIN	1	3	5
MAX	2	4	6
User-Defined Pre-Aggregation Statistics			

Count1: 1
Count2: 2

Estimated Sum: 7 × 1/2 + 16 × 2/3 = 14.167
Precise Sum: 14

Outline

- Introduction
- Aggregation Method
- V-Optimized Binning
- Experimental Results
- Conclusion

Accuracy Concern

- Conventional Bitmap Indices
  - Mainly designed for accelerating selection, not aggregation
  - 100% accuracy relies on the extra validation on the raw data
  - Now the raw data is unavailable
- Need for Accurate Approximation
  - Similar problems can be found in histogram literature

A Novel Binning Strategy

- Conventional Binning Strategies
  - Equi-width/equi-depth binning
  - Not necessarily a good approximation
- V-Optimized Binning Strategy
  - Inspired by v-optimal histogram
  - Goal: approximately minimizes ***Sum Squared Error*** **(SSE)**

$$SSE = \sum_{i=1}^{B} \sum_{j=1}^{n_i} (\overline{v}_i - v_{ij})^2$$

  - Unbiased v-optimized binning: data is randomly queried
  - Weighted v-optimized binning: certain subareas are frequently queried

Unbiased V-Optimized Binning

- 3 Steps:
  1) Initial Binning: uses equi-depth binning
  2) Iterative Refinement: adjusts bin boundaries
  3) Bitvector Generation: marks spatial positions

Algorithm 2 partition(*num_elements* N, *num_bins* B)

1: Initialize $B-1$ bin boundaries with equi-depth binning
2: Initialize two priority queues Q^+ and Q^-, where E^+_{min} is the minimum element in Q^+ and E^-_{max} is the maximum element in Q^-
3: **while** $E^+_{min} < E^-_{max}$ **do**
4: remove the ith boundary that corresponds to E^+_{min}
5: update Q^+ and Q^-
6: split the jth bin that corresponds to E^-_{max} in two
7: **if** $i = j$ **then**
8: break
9: **end if**
10: update Q^+ and Q^-
11: **end while**

Weighted V-Optimized Binning

- Difference: minimizes WSSE instead of SSE

$$WSSE = \sum_{i=1}^{B} \sum_{j=1}^{n_i} w_{ij} \times (\overline{v'}_i - v_{ij})^2$$

- Similar binning algorithm
- Major Modification
  - Representative value of each bin is weighted by querying probabilities

$$\overline{v'}_i = \frac{\sum_{j=1}^{n_i} w_{ij} \times v_{ij}}{\sum_{j=1}^{n_i} w_{ij}}$$

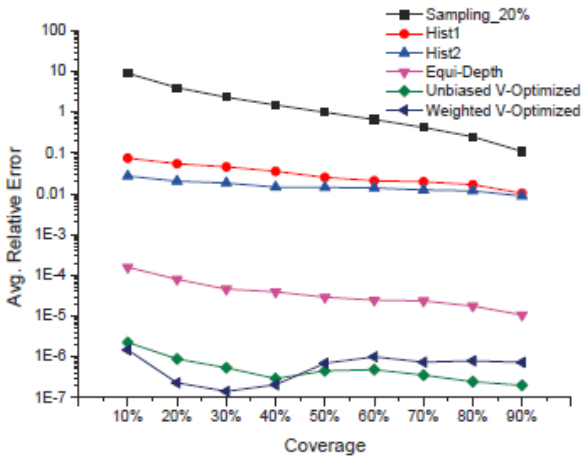
Outline

- Introduction
- Aggregation Method
- V-Optimized Binning
- Experimental Results
- Conclusion

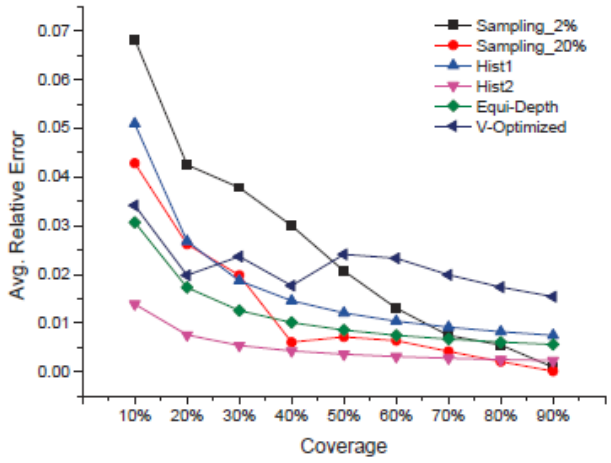
Experimental Setup

- Data Skew
  1) Dense Range: less than 5% space but over 90% data
  2) Sparse Range: less than 95% space but over 10% data
- 5 Types of Queries
  1) DB: with **dimension-based** predicates
  2) VBD: with **value-based** predicates over **dense** range
  3) VBS : with **value-based** predicates over **sparse** range
  4) CD: with **combined** predicates over **dense** range
  5) CS : with **combined** predicates over **sparse** range
- Ratio of Querying Possibilities – 10 : 1
  - 50% synthetic data is frequently queried
  - 25% real-world data is frequently queried

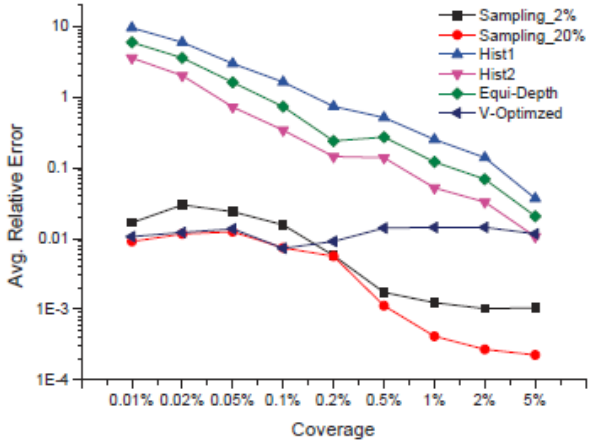
SUM Aggregation Accuracy of Different Methods



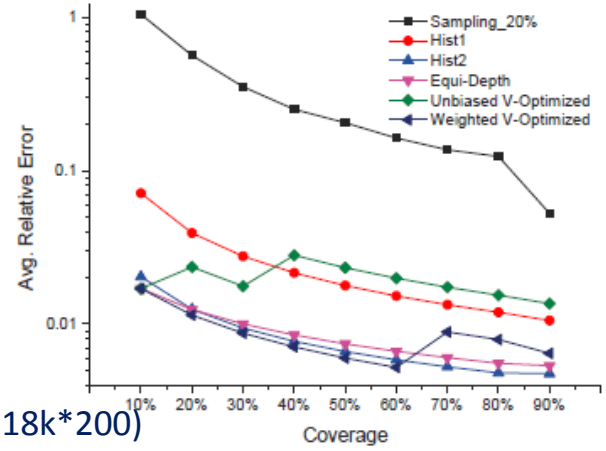
(a) DB



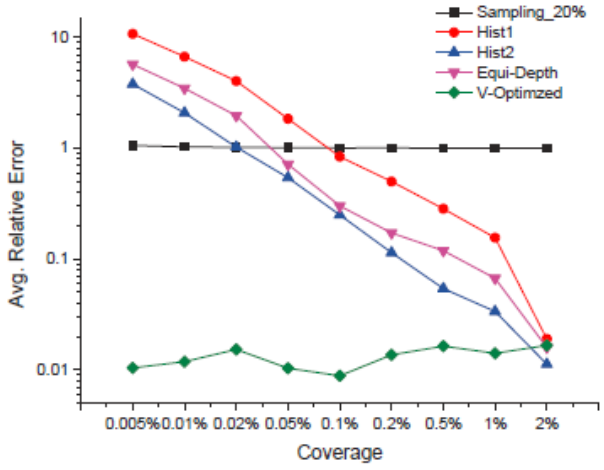
(b) VBD



(c) VBS



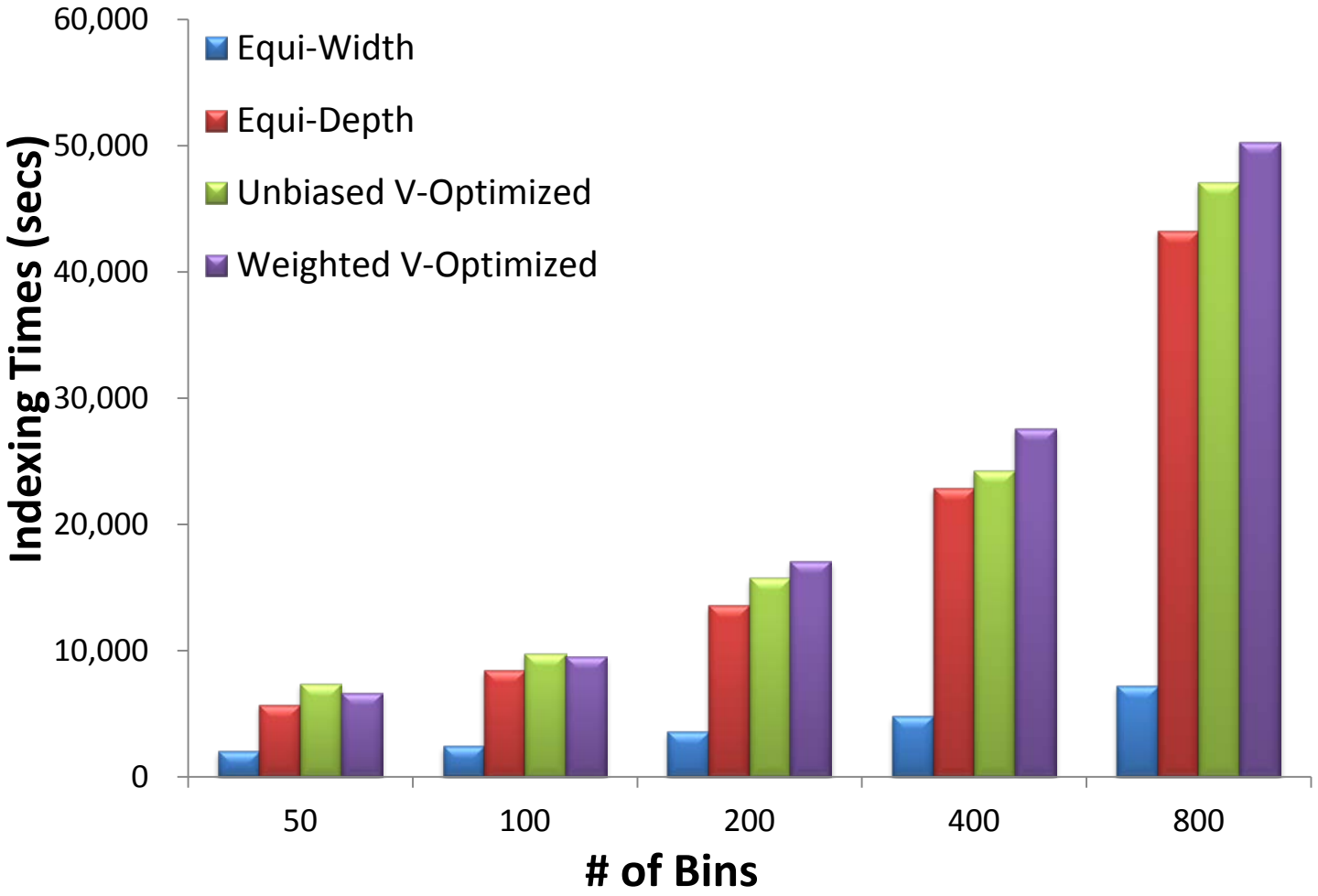
(d) CD



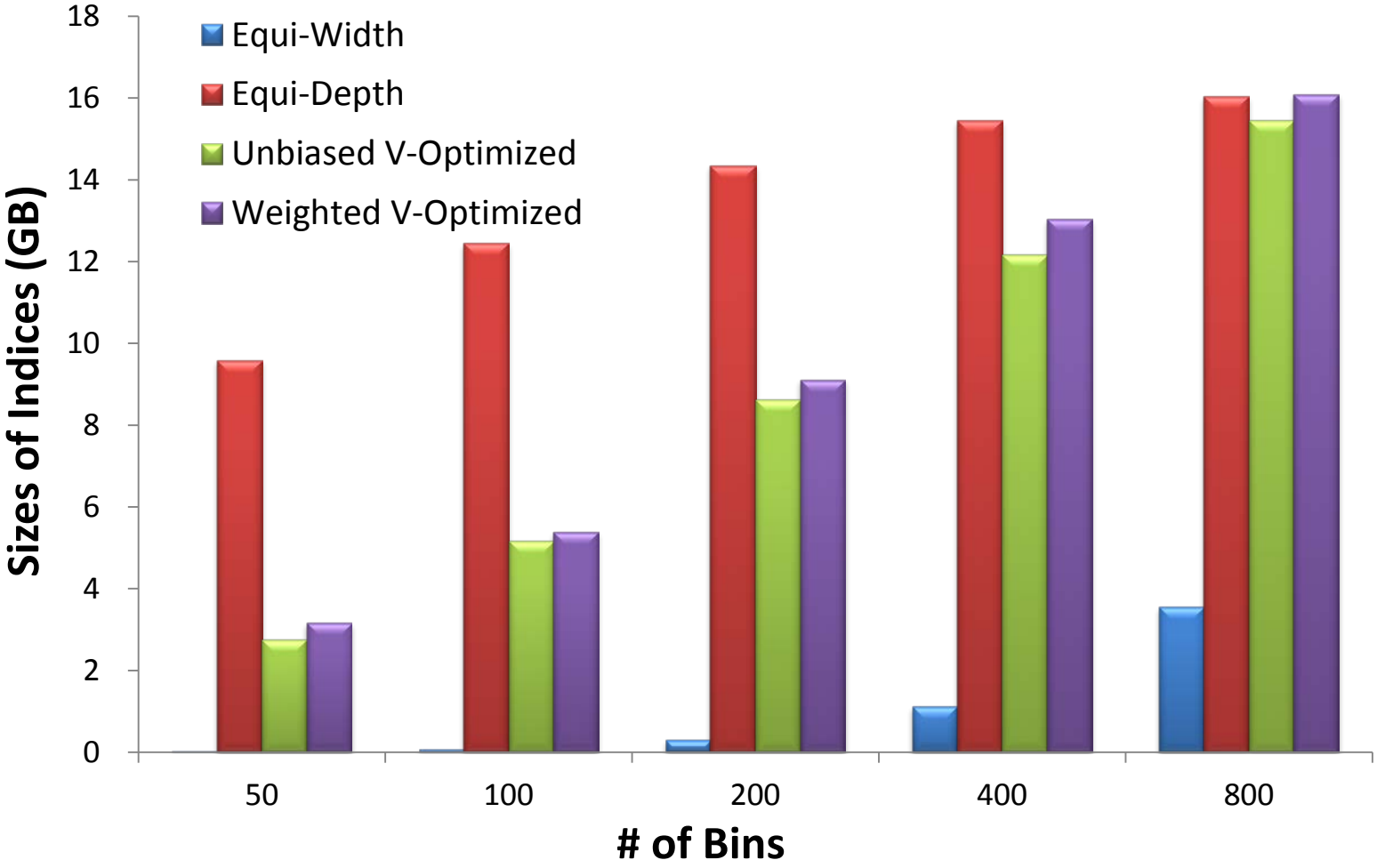
(e) CS

Sampling_2%
Sampling_20%
(Equi-Depth) Hist1 (200*18k*200)
(Equi-Depth) Hist2 (200*72k*800)
Unbiased V-Optimized
Weighted V-Optimized

Indexing Creation Times



Space Requirements of Indexing



Conclusion

- Bitmap Indices Good for Approximate Aggregation over Array Data
  - Allow dim-based and/or val-based predicates
  - Preserve both value and spatial distribution
  - Cheap creation and no data reorganization
- Novel Binning Strategy for High Accuracy
  - Unbiased v-optimized binning
  - Weighted v-optimized binning