Plot Query Processing with Wavelets

Mehrdad Jahangiri, Cyrus Shahabi
University of Southern California
Dept. of Computer Science
Los Angeles, CA 90089-0781
{jahangir,shahabi}@usc.edu
http://infolab.usc.edu

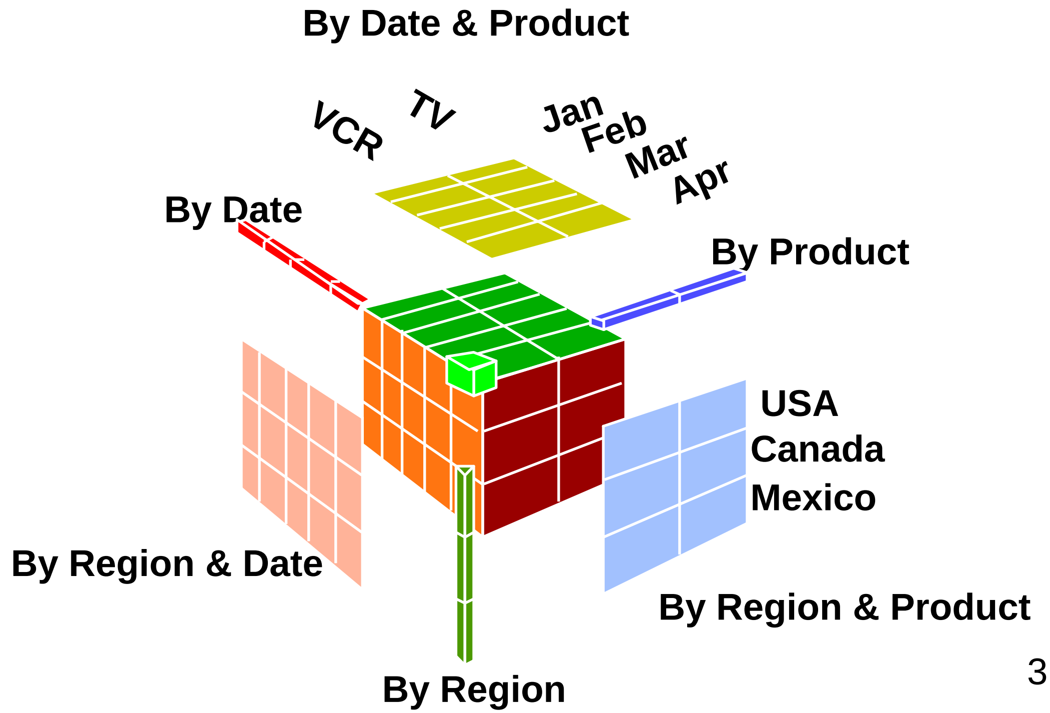
Outline

- Background
  - OLAP
  - Wavelets
- Wavelet-based OLAP
- Range Group-by Queries with Wavelets
- ProDA: An end-to-end WOLAP system
- Summary and Future Work

OLAP

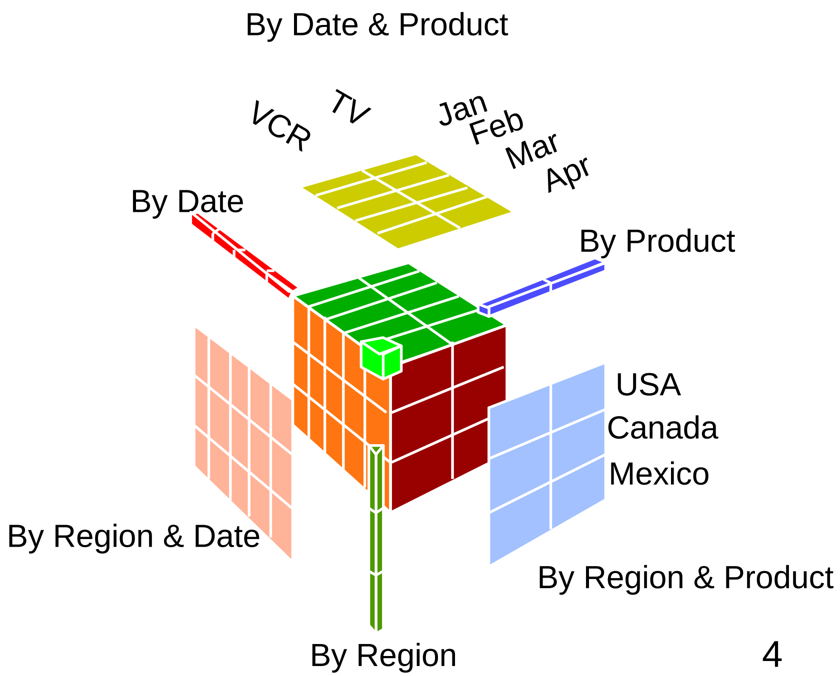
- OLAP: On-Line Analytical Processing
- Fast Analysis of Large Multidimensional Data
  - TV Sales in USA for Jan *(point queries)*
  - Total TV Sales in North America for Jan-Mar *(range aggregate queries)*
  - Variance of TV Sales in North America *(more complex queries)*

Product	Region	Date	Sales
VCR	USA	Jan	3
VCR	Canada	Feb	6
VCR	Mexico	Jan	2
PC	USA	Jan	4
PC	Mexico	Feb	4
TV	USA	Jan	5
TV	Canada	Feb	3
…		…	…

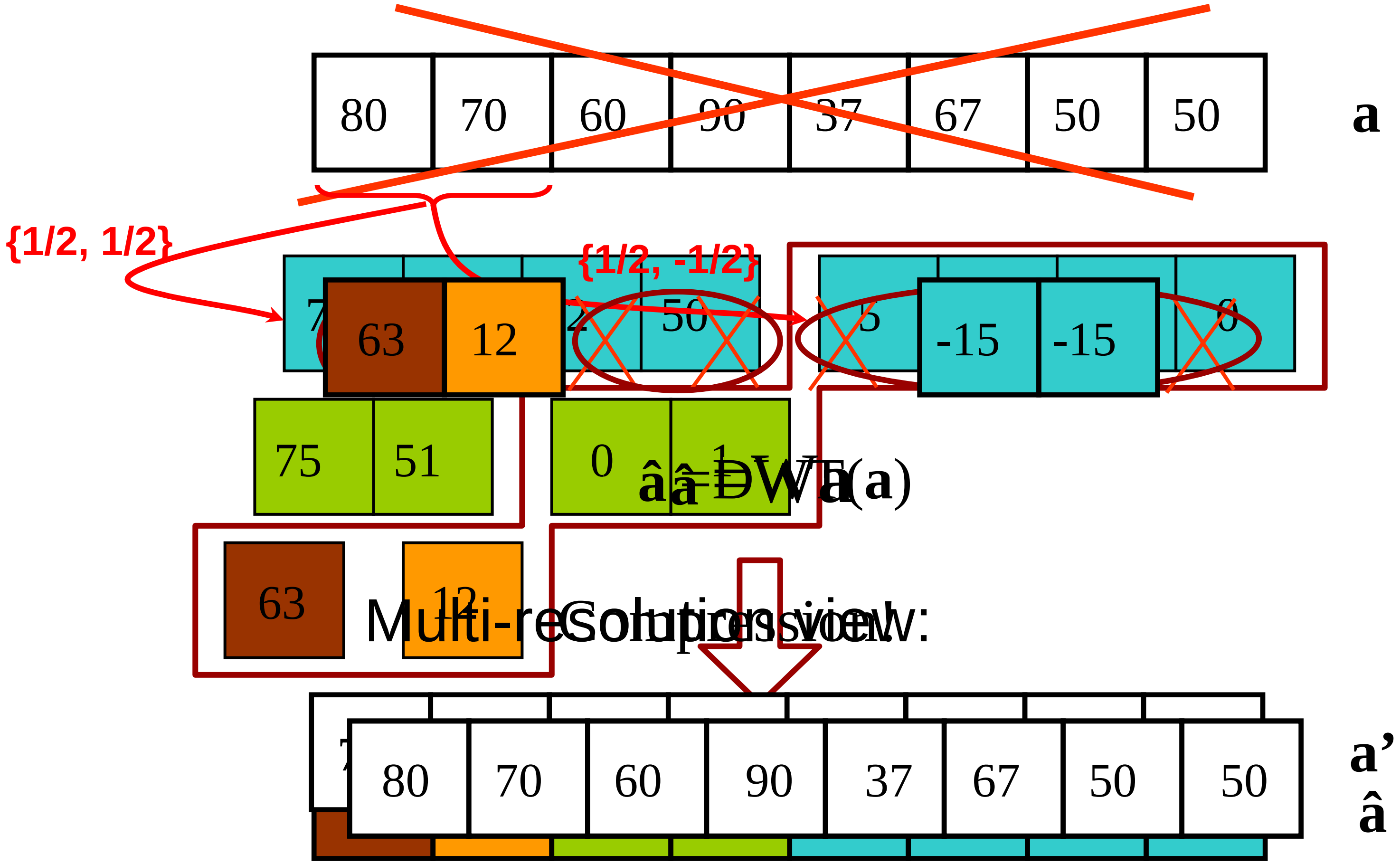


OLAP challenges

- Large multi-dimensional data
  - ✓ Scalability
- Fast response time
  - ✓ Fast exact, Approximate, or Progressive
- Aggregation
  - ✓ Pre-aggregation/transformation
- Ad-hoc ranges
  - ✓ Online computation
- Updates/Appends
  - ✓ Avoid re-doing
- More complex queries
  - ✓ Covariance, correlation, …



Discrete Wavelet Transform (Example)

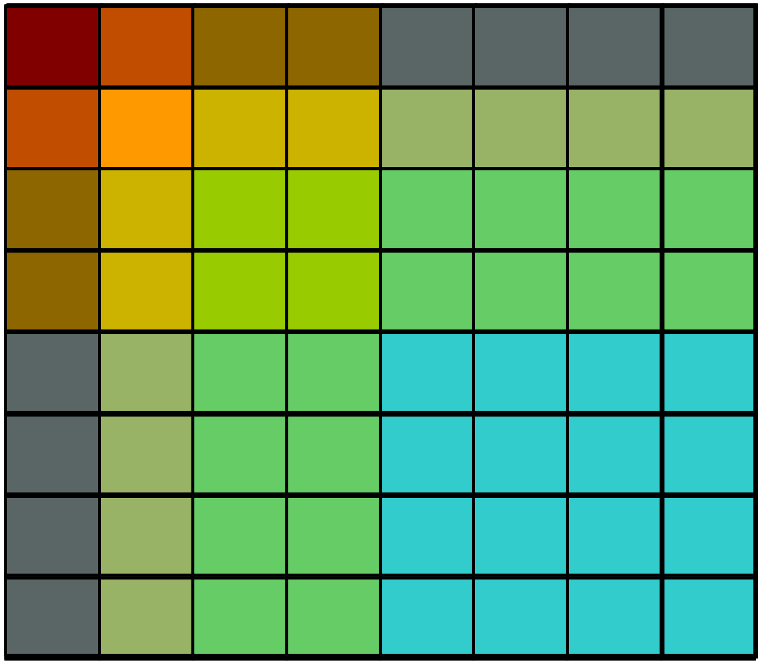


* We normalize our filters from **{1/2, 1/2}** and **{1/2, -1/2}** to **{1/√2, 1/√2}** and **{1/√2, -1/√2}**

Multidimensional DWT

- Series of one-dimensional transformations along each dimension with the order not being important
- W_x: matrix transformations along x
- W_y: matrix transformations along y
- DWT of a multidimensional D

$$\hat{D} = W_x W_y D$$



W_y $\hat{D}$ D_x D

WOLAP Example

Original	Wavelet*

a

80	70	60	90	37	67	50	50

â

178.1	33.9	0	2	7.07	-21.2	-21.2	0

1	1	1	1	1	1	1	1

2.83	0	0	0	0	0	0	0

Result=504

Result=178.19*2.83=504

(Parseval Theorem)

0	0	1	1	1	1	1	0

1.73	-.35	-1	.5	0	0	0	.71

Result=304

Result=178.19*1.73+33.94*(-.35)+2*.5

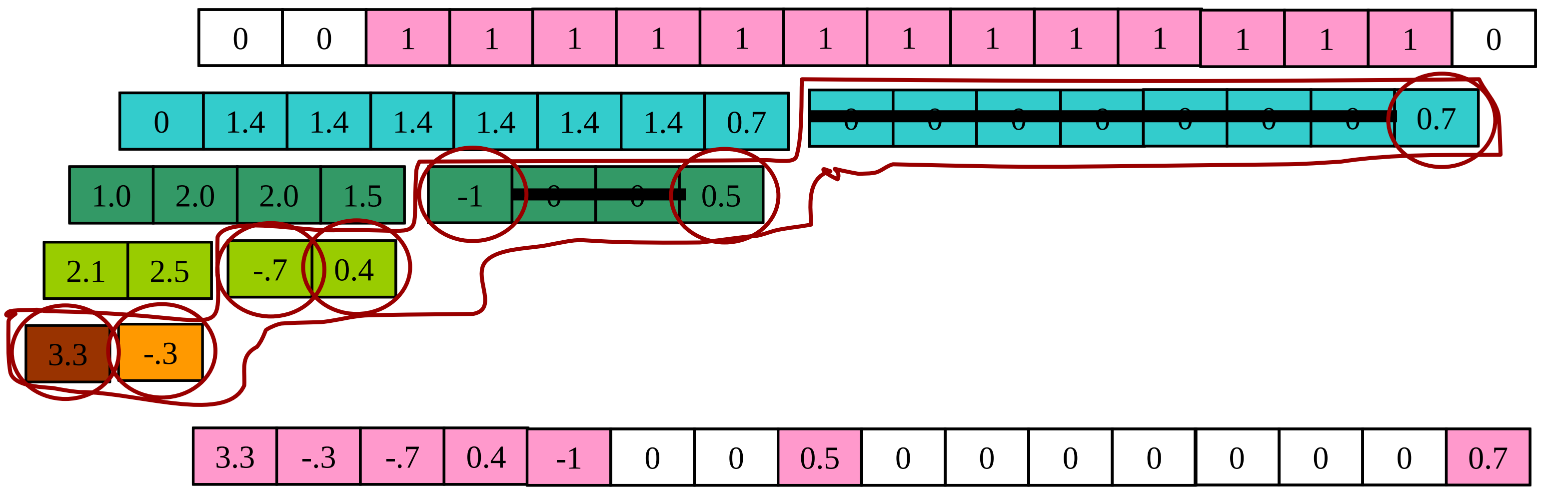
=304 (Parseval Theorem)

~303 (99% accuracy!)

O(N) << O(log N)

* Let’s normalize our filters from **{1/2, 1/2}** and **{1/2, -1/2}** to **{1/√2, 1/√2}** and **{1/√2, -1/√2}**

Aggregation Complexity is O(log N)



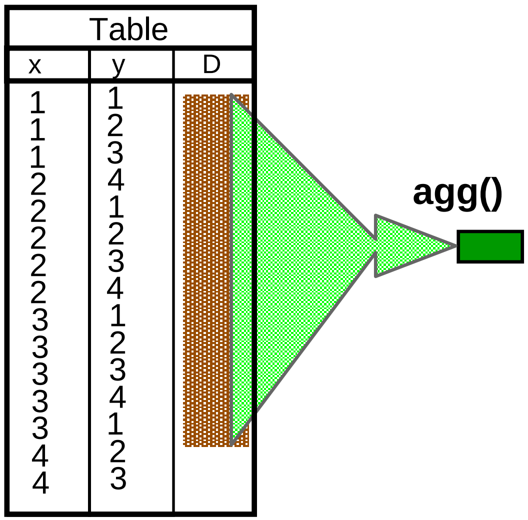
- Worst case: 2 non-zeros at each level
- Theorem:
  - If Size(Q)=N, $\hat{Q}$ has O(log N) non-zero values → O(log N) retrievals
  - Query Transformation is O(log N) by computing <u>only on the boundaries</u>:
    - Lazy Wavelet Transform

Outline

- ✓ Background
  - ✓ OLAP
  - ✓ Wavelets
- ✓ Wavelet-based OLAP
- Range Group-by Queries with Wavelets
- ProDA: An end-to-end WOLAP system
- Summary and Future Work

Range Group-by Query

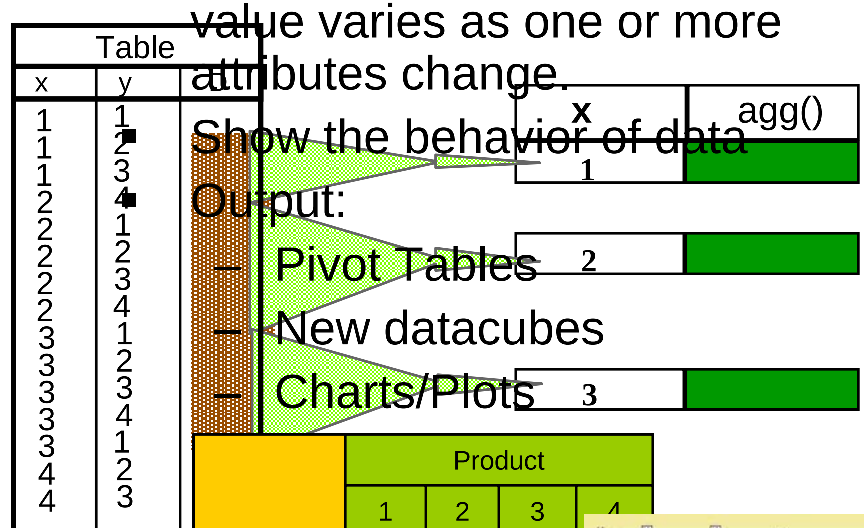
Range Aggregate Query

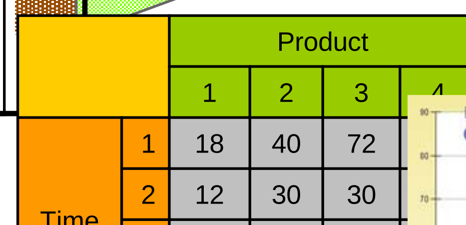


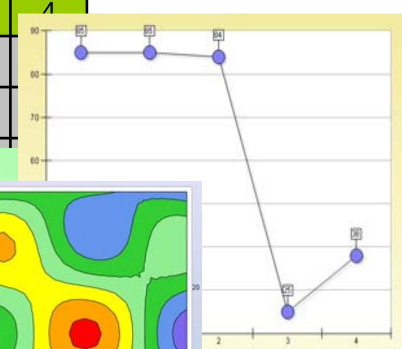
Select sum(D)
From Table
Where x<4 and y<5;

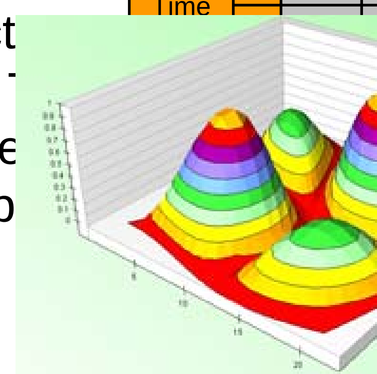
Range Group-by Query

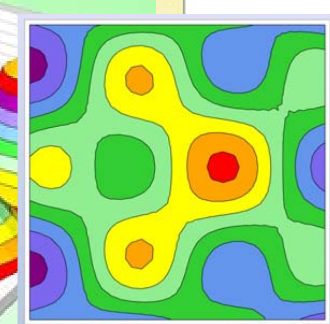
- Summarize how a measure value varies as one or more attributes change.
- Show the behavior of data
- Output:
  - Pivot Tables
  - New datacubes
  - Charts/Plots









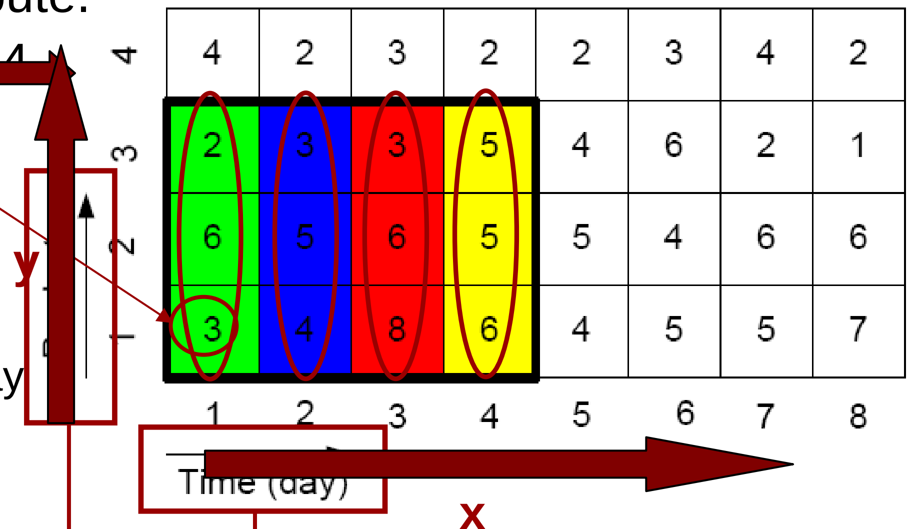


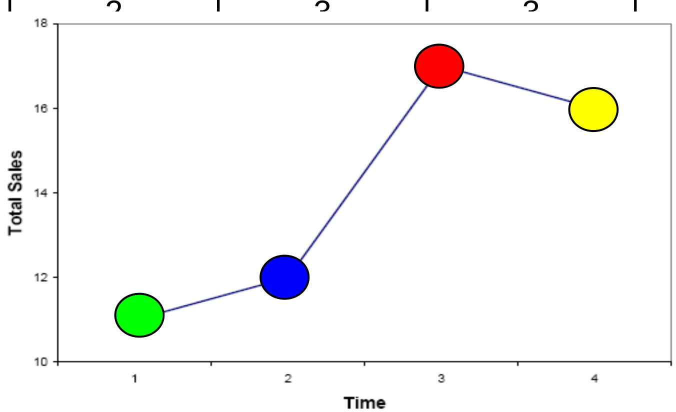
Select
From T
Where
Group

2-d Example

- Data: a 2-dimensional dataset with *product*, and *time* as the dimensions and *sales* as the measure attribute.
- Range: product<=3 and time<=4
- Query: Total Sales vs. Time
  - Time: Grouping dimension
  - Total: Aggregate function
- Result:
  - Aggregation of 3 products per day
  - **{(1,11),(2,12),(3,17),(4,16)}**

Time	Product	Sales
1	1	3
1	2	6
1	3	2
1	4	4
2	1	4
2	2	5





Time	Sales
1	11
2	12
3	17
4	16

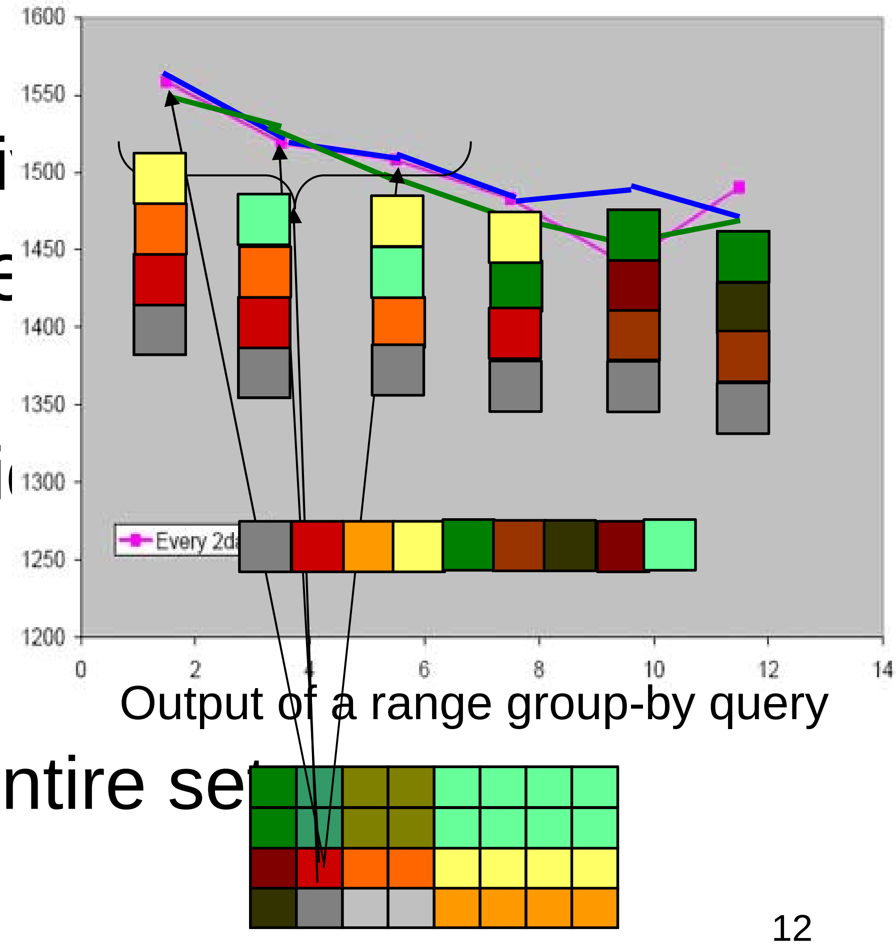
Grouping dimension (x)

Aggreg... (y)

11	12	17	16

Challenge

- Requirements:
  - Low-maintenance
  - Approximate/Progressi
- Set of individual querie
  - No I/O sharing
  - Approximation of indivi
- Single Query
  - One-pass algorithm
  - Approximation of the entire se



Output of a range group-by query

Query (Definition)

- x: grouping dimension with the range of $[l_x, h_x]$
- y: aggregating dimension with the range of $[l_y, h_y]$
- Query Definition:

$$\{(x, G) | l_x \leq x \leq h_x, G(x) = \sum_{l_y \leq y \leq h_y} D(x, y)\}$$

- Query Vector:

$$Q(y) = \begin{cases} 1 \ \mathit{if} \ l_y \leq y \leq h_y; \\ 0 \ \mathit{otherwise}. \end{cases}$$

- Query Definition :

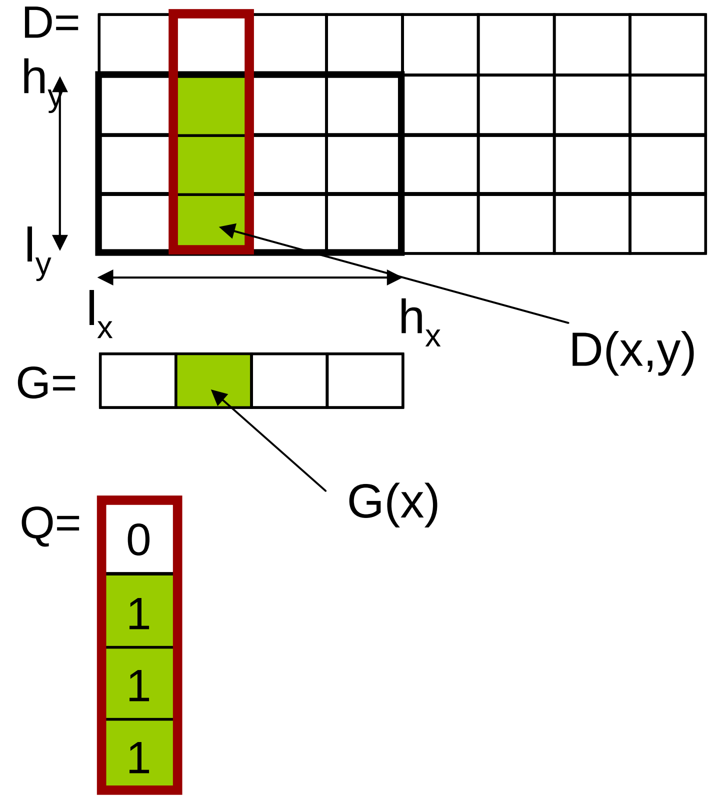
$$\{(x, G) | l_x \leq x \leq h_x, G(x) = \sum_y D(x, y) \cdot Q(y)\}$$

- Dot product of <the x column of D> and <Q>:

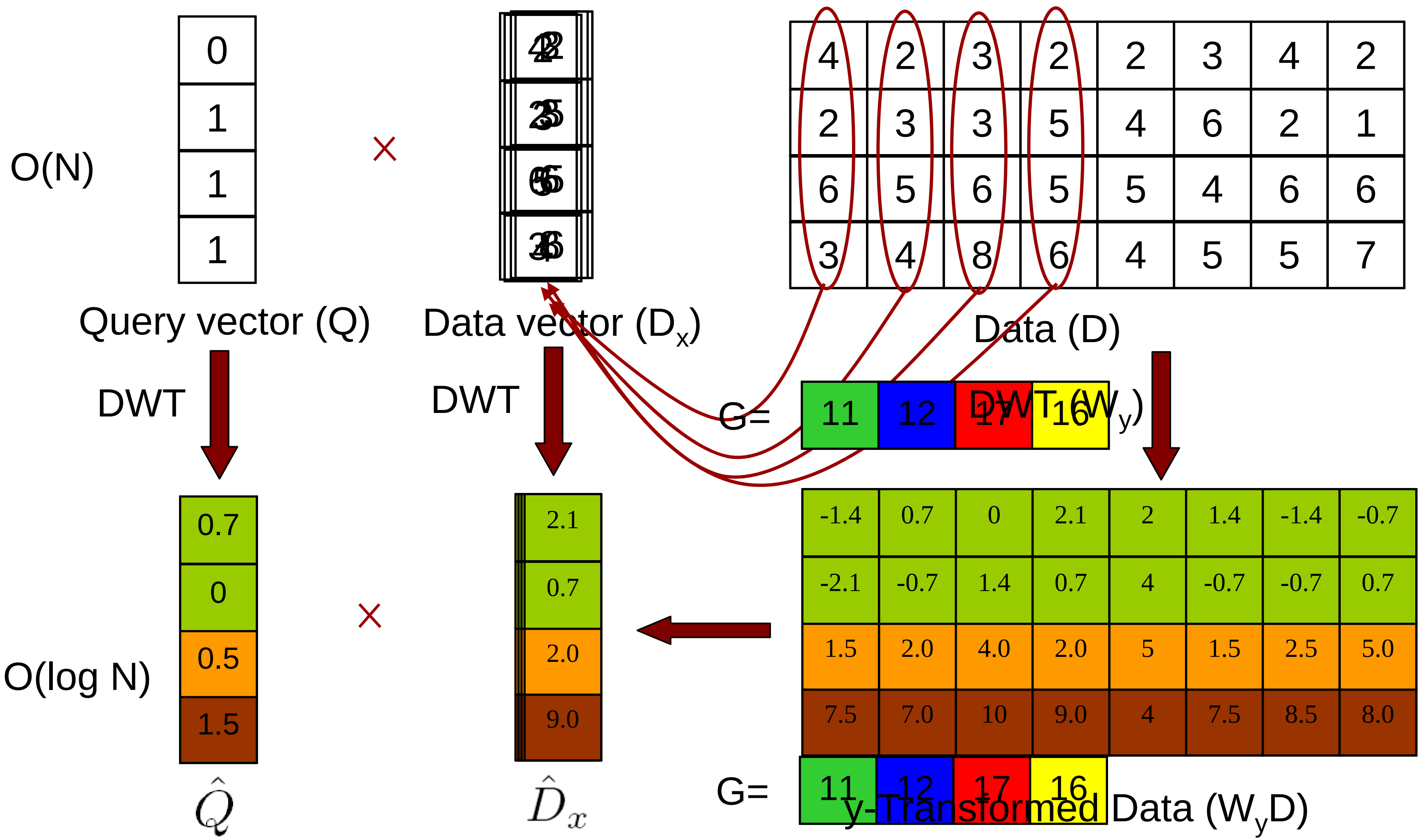
$$G(x) = \sum_y D_x(y) \cdot Q(y)$$

- Dot product of their wavelet-transform:

$$G(x) = \sum_y \hat{D}_x(y) \cdot \hat{Q}(y)$$



Dot product (Example)



$\hat{D}_x$: the x column of W_yD

Reconstruction + Aggregation

- We store $\hat{D}$, not W_yD
- Because: grouping dimensions are selected on-the-fly
  - We must store W_yD and W_xD
  - $O(2^d)$ for a d-dimensional dataset
- Solution: Online computation of W_yD from $\hat{D}$

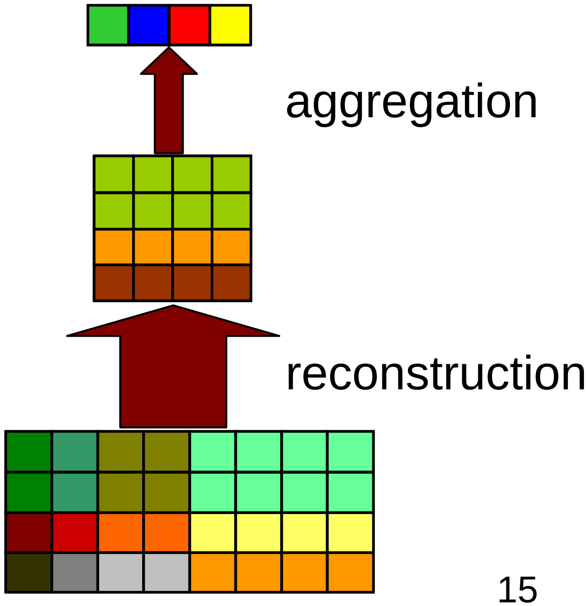
$$W_yD = W_x^{-1}\hat{D} \qquad (\hat{D} = W_xW_yD)$$

- Algorithm:
  - Step1 (reconstruction):

$$W_yD(x,y) = \boxed{\sum_{\alpha} W_x^{-1}(x,\alpha)\cdot\hat{D}(\alpha,y)}$$

  - Step2 (aggregation):

$$G(x) = \sum_{y} \boxed{W_yD(x,y)} \cdot \hat{Q}(y)$$



- Not efficient
  - D is Large → Online computation of W_yD is costly!

Aggregation + Reconstruction

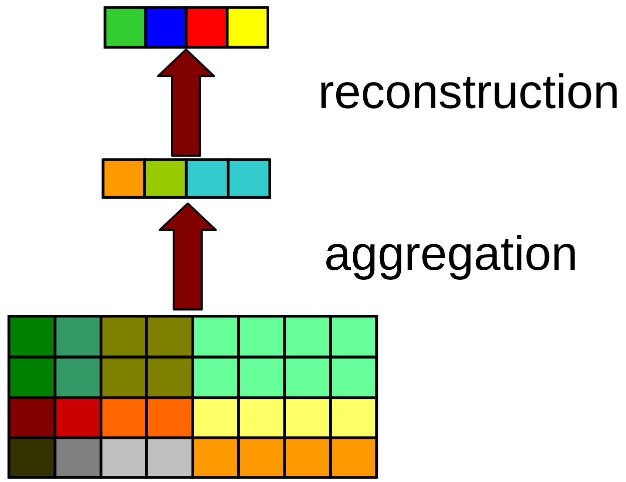
- We interchange the equations to push the aggregation down to the wavelet domain:

$$G(x) = \sum_y \left(\sum_\alpha W_x^{-1}(x,\alpha)\hat{D}(\alpha,y)\right)\hat{Q}(y)$$
$$= \sum_\alpha W_x^{-1}(x,\alpha)\boxed{\left(\sum_y \hat{D}(\alpha,y)\hat{Q}(y)\right)}$$

- Efficient plot query processing:

Step 1 (Aggregation) : $$\hat{G}(x) = \sum_x \hat{D}(x,y)\hat{Q}(y)$$

Step 2 (Reconsutrction) : $$G(x) = \sum_y W_x^{-1}(x,y)\hat{G}(y)$$

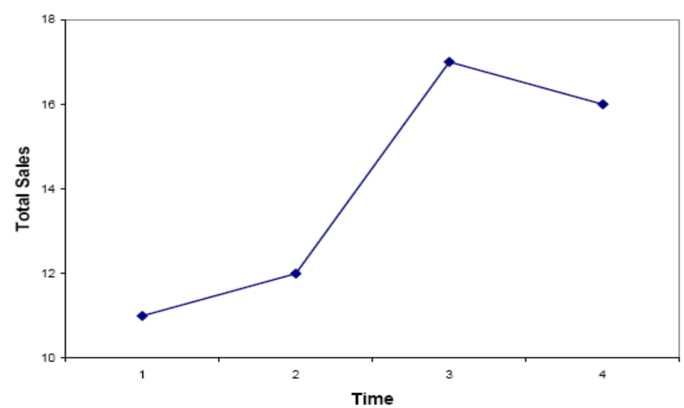


Complete Example

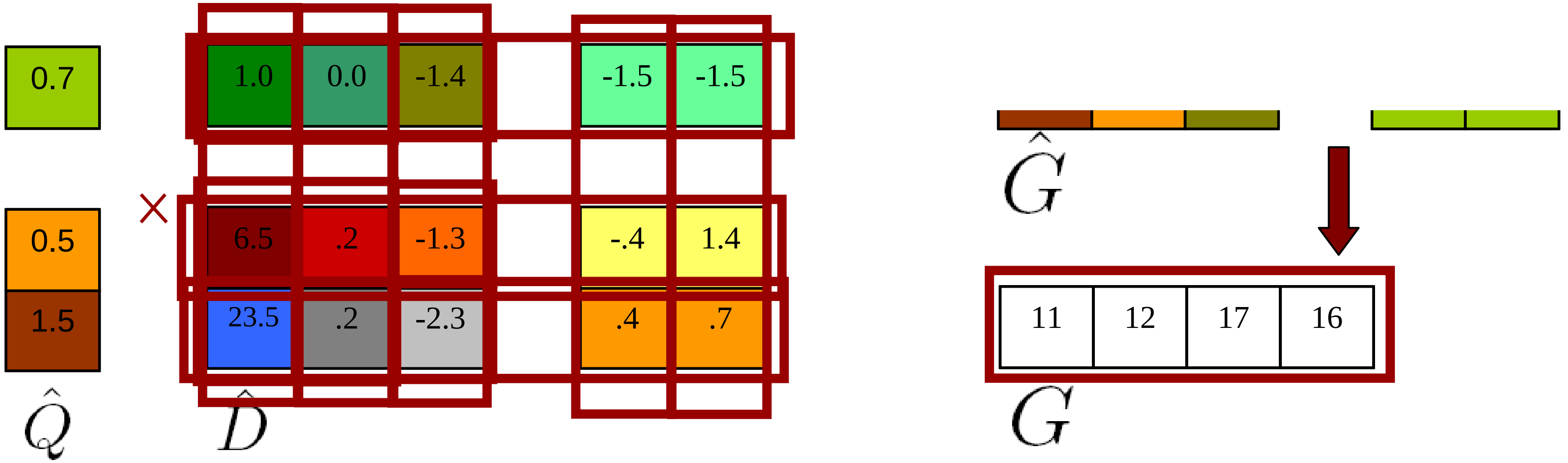
$$
\hat{Q} = \begin{bmatrix} 0.7 \\ \cancel{0} \\ 0.5 \\ 1.5 \end{bmatrix}
\times
\hat{D} = \begin{bmatrix}
1.0 & 0.0 & -1.4 & \cancel{2.8} & -1.5 & -1.5 & \cancel{-.5} & \cancel{-.5} \\
\cancel{-.3} & \cancel{-.3} & \cancel{-2.5} & \cancel{0.0} & \cancel{1.0} & \cancel{0.5} & \cancel{-1.0} & \cancel{-1.0} \\
6.5 & .2 & -1.3 & \cancel{-3.0} & -.4 & 1.4 & \cancel{1.1} & \cancel{-1.8} \\
23.5 & .2 & -2.3 & \cancel{0.0} & .4 & .7 & \cancel{1.1} & \cancel{.4}
\end{bmatrix}
$$

$$
\hat{G} = \begin{bmatrix} 39.2 & .35 & -5.0 & \cancel{0.5} & -.71 & .71 & \cancel{-1.4} & \cancel{-.71} \end{bmatrix}
$$

$$
G = \begin{bmatrix} \boxed{11 \;\; 12 \;\; 17 \;\; 16} & 13 & 15 & 13 & 14 \end{bmatrix}
$$

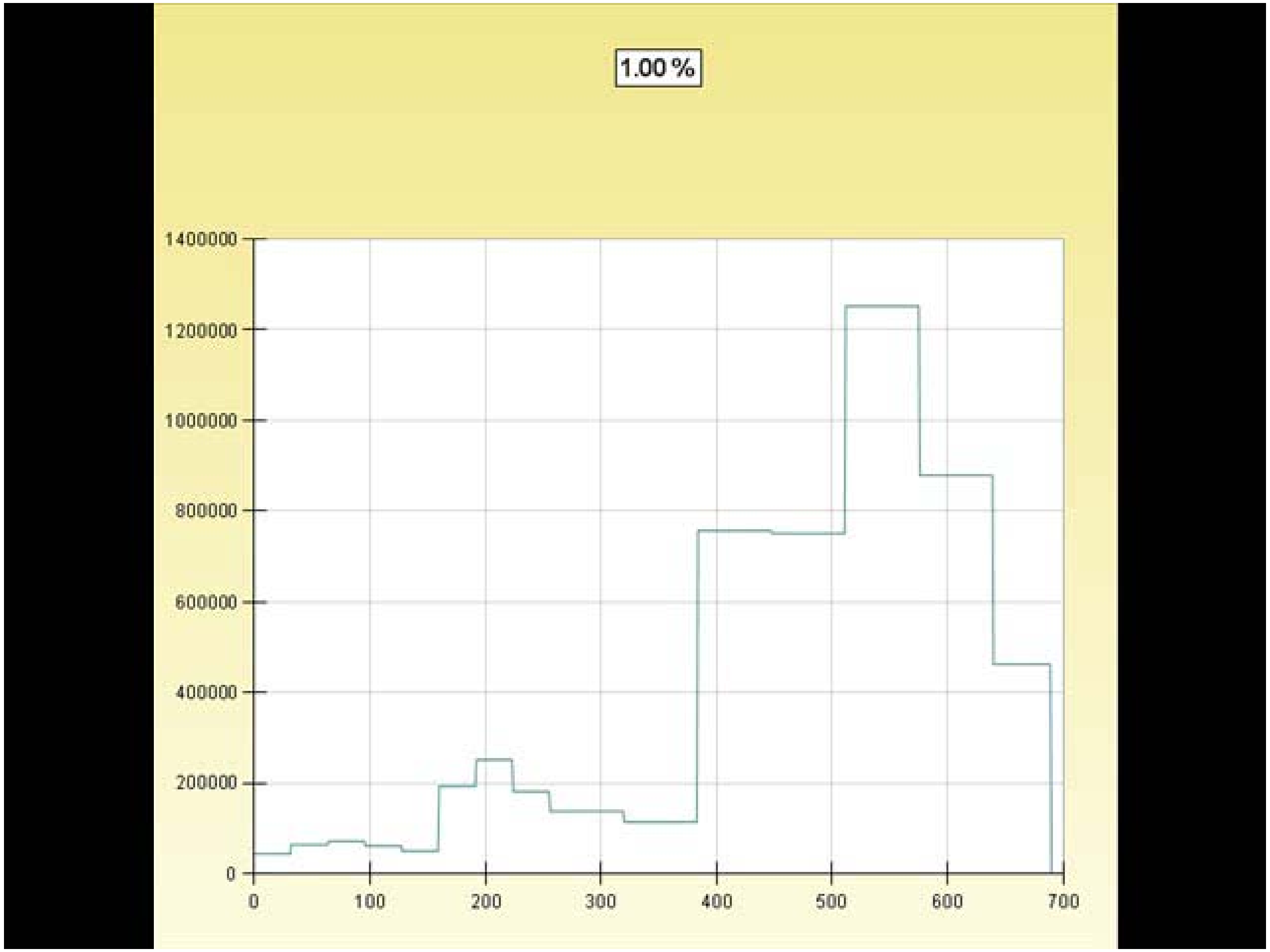


Progressiveness



- Favoring Aggregation
  1. First-B on Aggregating
  2. Highest-B on Aggregating
- Favoring Reconstruction
  3. First-B on aggregating
  4. Highest-B on aggregating
- Hybrid
  5. First-B on both aggregating and grouping
  6. Highest-B on aggregating and First-B on grouping

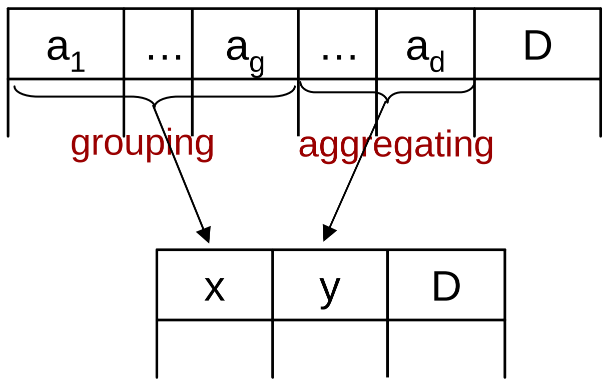
Progressive Output (Example)



d-dimensional Query

- Consider a d-dimensional dataset with $a_1,\ldots,a_d$ as its dimensions and $D(a_1,\ldots,a_d)$ as its measure.
- Let the query range be $[l_i,h_i]$ for each dimension i
- Let the first g dimensions be the grouping dimensions
- SQL statement:

$$
\begin{array}{rl}
SELECT & a_1, \ldots, a_g, SUM(D) \\
FROM & Data \\
WHERE & l_1 \le a_1 \le h_1 \\
 & \ldots \\
AND & l_d \le a_d \le h_d \\
GROUP\ BY & a_1, \ldots, a_g;
\end{array}
$$



- Query is defined as:

$$\{(a_1, \ldots, a_g, G) | \forall i \le g, l_i \le a_i \le h_i,$$
$$G(a_1, \ldots, a_g) = \sum_{l_{g+1} \le a_{g+1} \le h_{g+1}} \ldots \sum_{l_d \le a_d \le h_d} D(a_1, \ldots, a_d)\}$$

$$\hat{D} = \underbrace{W_1 \ldots W_g}_{W_x} \underbrace{W_{g+1} \ldots W_d}_{W_y} D \quad \{(x, G) | \hat{D} = W_x W_y D \sum_{l_y \le y \le h_y} D(x, y)\}$$

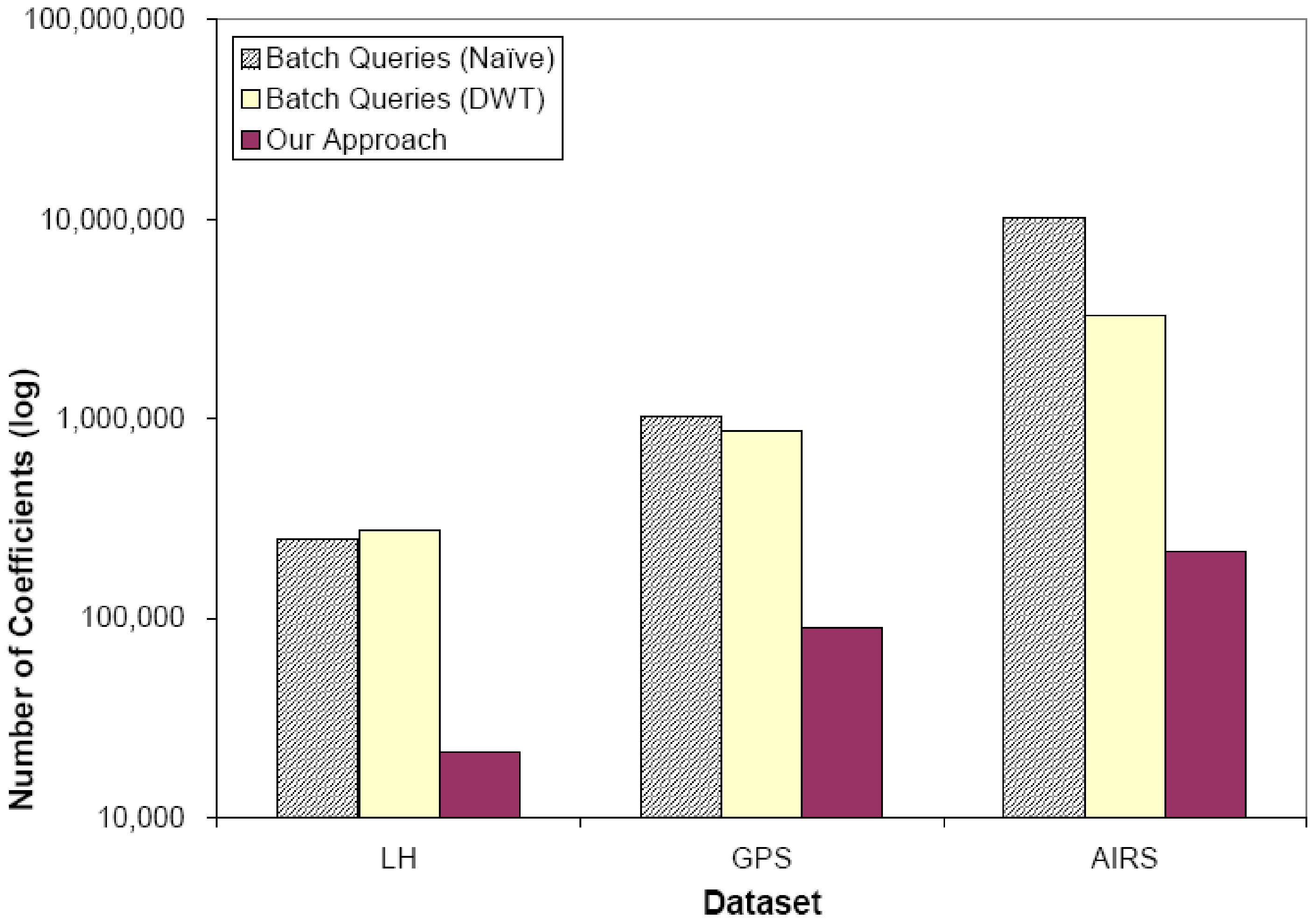
Performance Evaluation

- Experimental Datasets
- Query Performance
- Effect of Grouping Dimensions
- Progressiveness

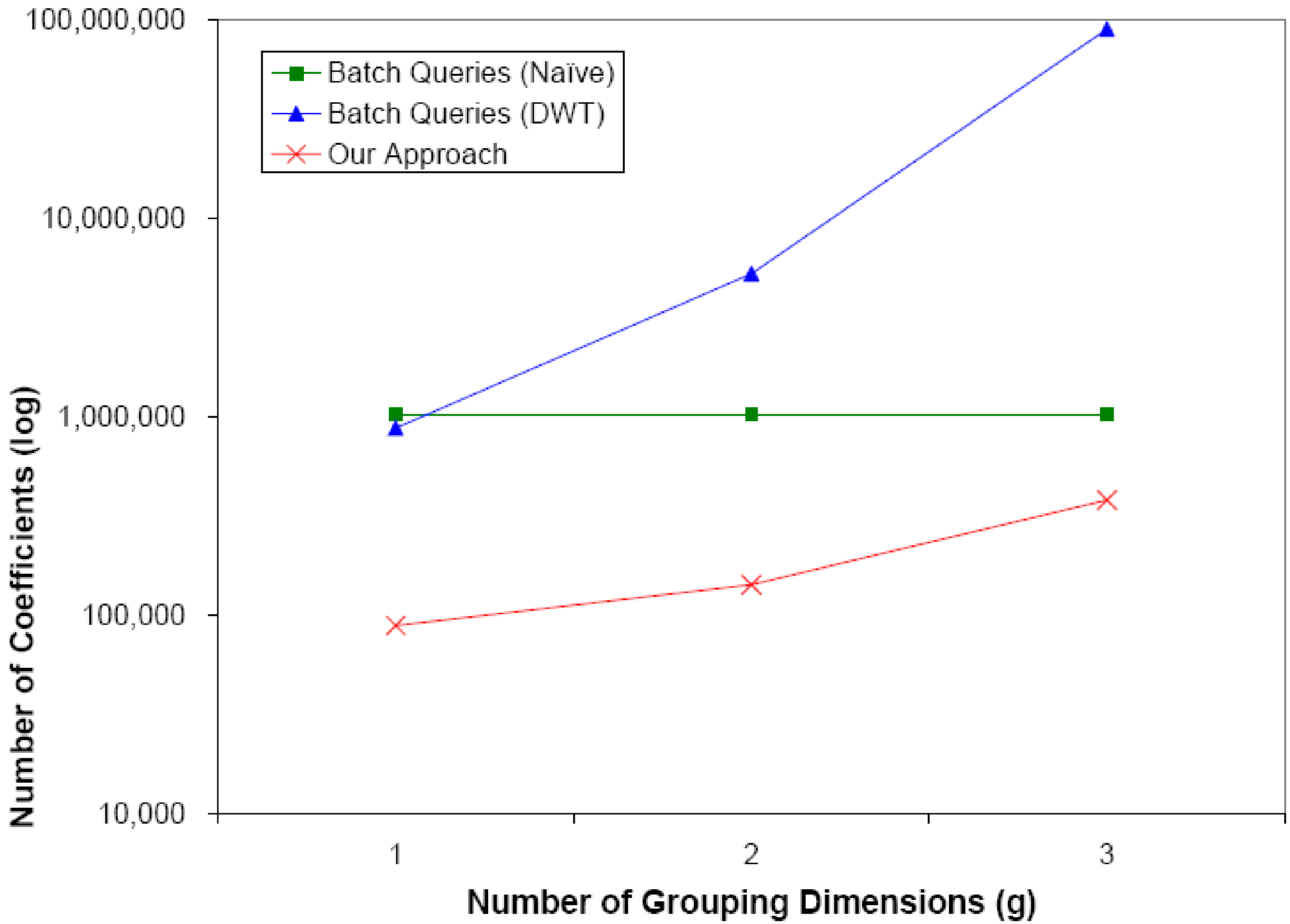
Experimental Datasets

- LH
  - Monthly production and injection history data for a waterflood oil reservoir for 57 years
  - Dimensions: well ID and time
  - Measure: oil production
  - Size: 1 GB
- GPS
  - Profiles of atmospheric water vapor pressure with resolution of about a kilometer, derived from radio occultation data for 9 months
  - Dimensions: latitude, longitude, pressure level, and time
  - Measure: water vapor pressure
  - Size: 2 GB
- AIRS
  - Earths atmospheric temperature profiles at a very high rate for one year
  - Dimensions: latitude, longitude, pressure level, and time
  - Measure: temperature
  - Size: 320 GB

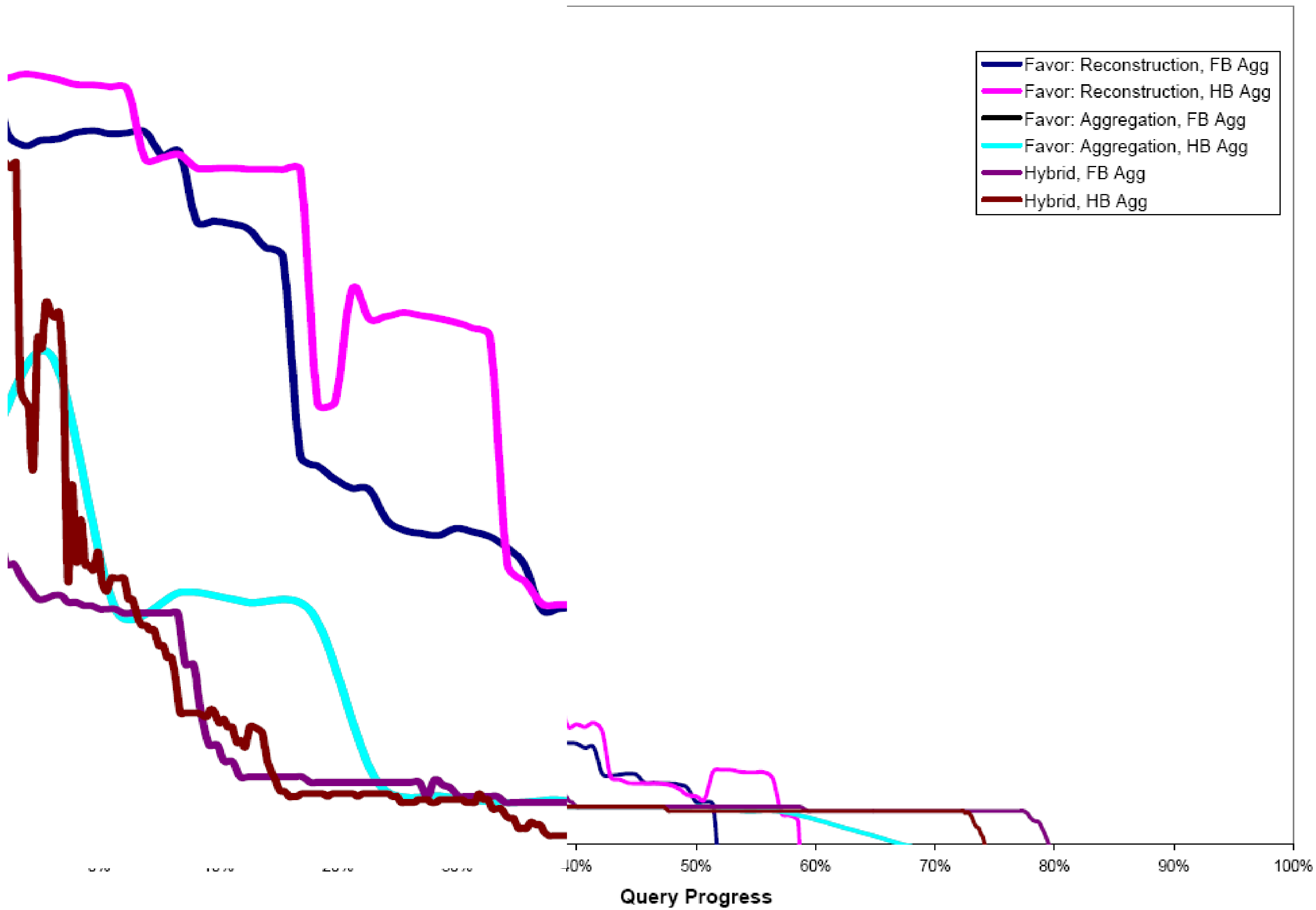
Query Performance



Effect of Grouping Dimensions



Progressive Processing



Summary and Future Work

- Summary:
  - We addressed an important class of queries, "range group-by query"
  - We employ wavelets to support exact, approximate, and progressive range group-by queries on large multidimensional datasets, while keeping update costs relatively low
  - An efficient range group-by query processing allows scientists to generate meaningful plots on large multidimensional datasets for arbitrary settings
- Future Work:
  - Including having into the range group-by query

References

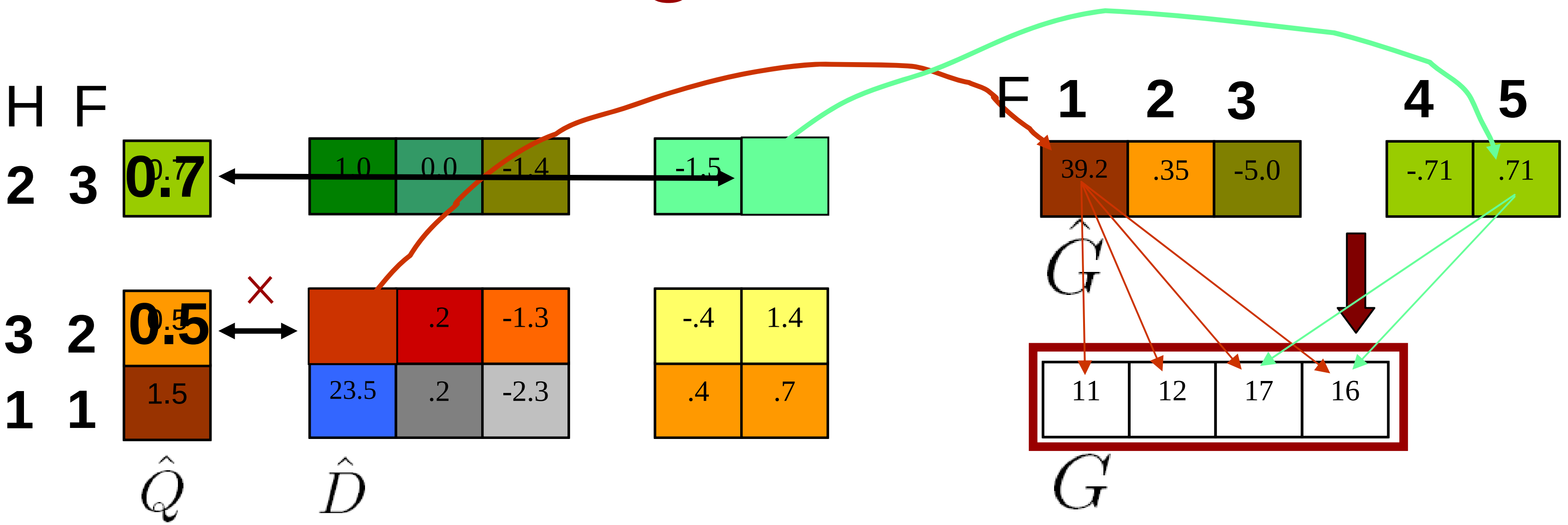
- R. R. Schmidt, C. Shahabi, ProPolyne: A Fast Wavelet-based Algorithm for Progressive Evaluation of Polynomial Range-Sum Queries, EDBT 2002, Prague, Czech
- [Shahabi-PODS'02]R. R. Schmidt, C. Shahabi, How to Evaluate Multiple Range-Sum Queries Progressively
- C. Shahabi, M. Jahangiri, D. Sacharidis, Hybrid Query and Data Ordering for Fast and Progressive Range-Aggregate Query Answering, International Journal of Data Warehousing and Mining, April'05.
- M. Jahangiri, D. Sacharidis, C. Shahabi, SHIFT-SPLIT: I/O Efficient Maintenance of Wavelet-Transformed Multidimensional Data, ACM SIGMOD 2005, Baltimore, Maryland
- M. Jahangiri, C. Shahabi, ProDA: A Suite of WebServices for Progressive Data Analysis, ACM SIGMOD 2005, Baltimore, Maryland, (demonstration)
- M. Jahangiri, C. Shahabi, Essentials for Modern Data Analysis Systems, Second NASA Data Mining Workshop 2006, Pasadena, California
- M. Jahangiri, C. Shahabi, Enabling Pivot Charts on Massive Multidimensional Datasets, Microsoft eScience Workshop, Chapel Hill, NC, Oct 2007.
- C. Shahabi, M. Jahangiri, F. Banaei-Kashani, ProDA: An End-to-End Wavelet-Based OLAP System for Efficient Analysis of Massive Datasets, IEEE Computer, April 2008.
- M. Jahangiri, C. Shahabi, WOLAP: Wavelet-Based Range Aggregate Query Processing, Submitted to VLDBj.
- M. Jahangiri, C. Shahabi, Plot Query Processing with Wavelets, Scientific and Statistical Database Management (SSDBM), July 2008.

Thank you

References

- [Gray-ICDE'96] J.~Gray, A.~Bosworth, A.~Layman, and H.~Pirahesh., Datacube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total
- [Kotidis-SIGMOD'02]Yannis Sismanis, Nick Roussopoulos, Antonios Deligianannakis, and Yannis Kotidis., Dwarf: Shrinking the petacube.
- [Ioannidis-VLDB'06]Konstantinos Morfonios and Yannis Ioannidis., Cure for cubes: cubing using a rolap engine.
- [Lakshmanan-SIGMOD'03]Laks V.~S. Lakshmanan, Jian Pei, and Yan Zhao., Qc-trees: an efficient summary structure for semantic olap.
- [Agrawal-SIGMOD'97]C.~Ho, R.~Agrawal, N.~Megiddo, and R.~Srikant., Range queries in OLAP data cubes.
- [Abbadi-ICDE'99]S.~Geffner, D.~Agrawal, A.~El Abbadi, and T.~Smith., Relative prefix sums: An efficient approach for querying dynamic OLAP data cubes.
- [Abbadi-DaWak'00] Mirek Riedewald, Divyakant Agrawal, and Amr~El Abbadi., Space-efficient datacubes for dynamic environments.
- [Vitter-CIKM'98]J.~S. Vitter, M.~Wang, and B.~R. Iyer.,Data cube approximation and histograms via wavelets.
- [Vitter-SIGMOD'99]J.~S. Vitter and M.~Wang.,Approximate computation of multidimensional aggregates of sparse data using wavelets.
- [Agrawal-CIKM'00]Yi-Leh Wu, Divyakant Agrawal, and Amr~El Abbadi. Using wavelet decomposition to support progressive and approximate range-sum queries over data cubes.
- [Garofalakis-VLDB'00]Kaushik Chakrabarti, Minos~N. Garofalakis, Rajeev Rastogi, and Kyuseok Shim., Approximate query processing using wavelets.
- [Lee-DAFSAA'06] Young-Koo Lee, Woong-Kee Loh, Yang-Sae Moon, Kyu-Young Whang, and Il-Yeol Song, An Efficient Algorithm for Computing Range-Groupby Queries

Progressiveness



- Grouping dimension (x):
  - Lowered frequencies are preferred (First-B ordering)
- Aggregating dimension (y):
  - Lower frequencies are preferred (First-B ordering)
  - Higher values of Q are preferred (Highest-B ordering)