Data Warehouse

——Data Warehouse Implementation——

徐华

清华大学 计算机系 智能技术与系统国家重点实验室

xuhua@tsinghua.edu.cn

1

Data Warehouse

- Review the basic concepts of database
- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

2

Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L levels?

$$T = \prod_{i=1}^{n} (L_i + 1)$$

- Materialization of data cube
  - Materialize every (cuboid) (full materialization), none (no materialization), or some (partial materialization)
  - Selection of which cuboids to materialize
    - Based on size, sharing, access frequency, etc.

3

Cube Operation

- Cube definition and computation in DMQL

  define cube sales[item, city, year]: sum (sales_in_dollars)

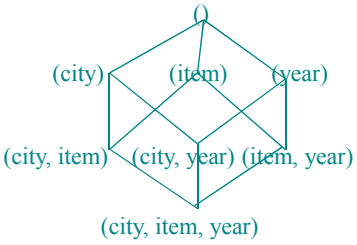
  compute cube sales

- Transform it into a SQL-like language (with a new operator **cube by**, introduced by Gray et al.' 96)

  SELECT item, city, year, SUM (amount)
  FROM SALES
  CUBE BY item, city, year

- Need compute the following Group-Bys

  (date, product, customer),
  (date, product),(date, customer), (product, customer),
  (date), (product), (customer)
  ()



4

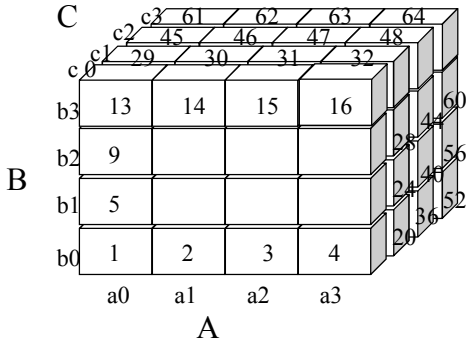
Cube Computation: ROLAP-Based Method

- ROLAP-based cubing algorithms
  - Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
  - Grouping is performed on some sub-aggregates as a "partial grouping step"
  - Aggregates may be computed from previously computed aggregates, rather than from the base fact table

5

Multi-way Array Aggregation for Cube Computation

- Partition arrays into chunks (a small subcube which fits in memory).
- Compressed sparse array addressing: (chunk_id, offset)
- Compute aggregates in "multiway" by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost.

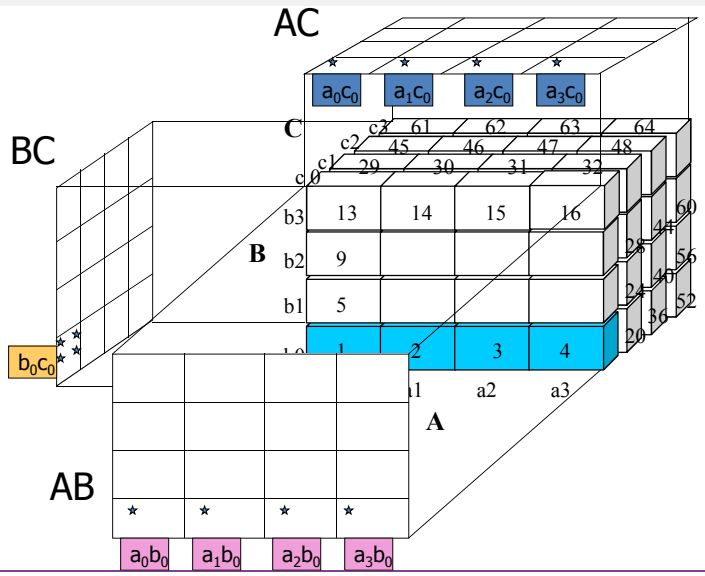


A: 40
B: 400
C: 4000

What is the best traversing order to do multi-way aggregation?

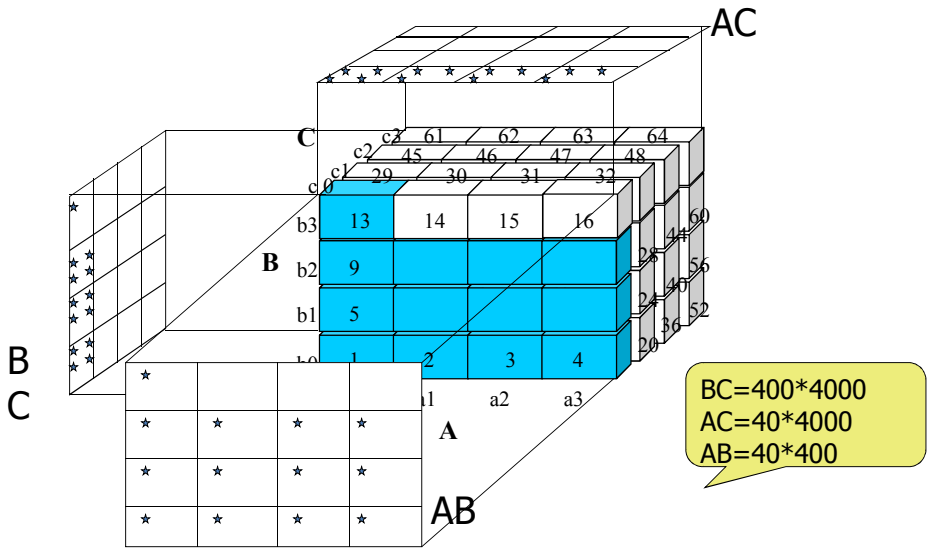
6

Multi-way Array Aggregation for Cube Computation



7

Multi-way Array Aggregation for Cube Computation



8

Data Warehouse — Subject-Oriented

为完成计算内存中保持所有的2 - D平面所需最小存储为：

最佳的是AB平面40*400+AC平面的一行40*1000+BC平面的一块100*1000

总计是156000.

如果次序是BC,AC,AB则所需内存是400*4000+40*1000+10*100

=1641000,是最佳策略的10倍以上.

- Limitation of the method: computing well only for a small number of dimensions
  - If there are a large number of dimensions, "bottom-up computation" and iceberg cube computation methods can be explored

9

Indexing OLAP Data: Bitmap Index

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The i-th bit is set if the i-th row of the base table has the value for the indexed column
- Not suitable for high cardinality domains

Base table

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail
C5	Europe	Dealer

Index on Region

RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

10

Efficient Processing OLAP Queries

- Determine which operations should be performed on the available cuboids:
  - transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g, dice = selection + projection
- Determine to which materialized cuboid(s) the relevant operations should be applied.
- Exploring indexing structures and compressed vs. dense array structures in MOLAP

11

Thanks !

12