OPTIMAL CHUNKING OF LARGE MULTIDIMENSIONAL ARRAYS FOR DATA WAREHOUSING

Reading club – Big Data Infrastructures for Neuroinformatics laboratory, Concordia University

https://dl.acm.org/doi/10.1145/1317331.1317337

Timothée Guédon

CONTEXT

Numerous applications in scientific domains such as Physics, Astronomy, Geology, Earth Sciences, Statistics, etc., **map their problems space onto matrices and multi-dimensional arrays** on which mathematical tools such as linear, non-linear equations solvers and differential equation solvers can be applied.

Such arrays are required to be:

- persistent on disks
- accessed efficiently for scientific analysis.

MOLAP?

OLAP

- Online analytical processing
- To analyze multidimensional data interactively
- Comprises business intelligence, data mining, databases

MOLAP

- « M » for multidimensional
- Classic form of OLAP
- Stores data in an optimized multi-dim. array storage
- As opposed to storage in relational databases

INTRODUCTION

ARRAY STORAGE OPTIONS

- Naive: seek into one file containing all the data
- Persistent storage is typically done by chunking

CHUNKS

Units of transfers between disk and memory

Two properties:

- Chunk size
- Chunk shape

INTRODUCTION

CHUNKING

- Multidimensional array divided into chunks
- Each chunk stored on disk contiguously
- Layout of the chunks can be done using other linear mapping

Example linear mapping: space-filling curves

BENEFITS

- array chunks with all zero entries are not stored
- chunks with few entries can be compressed.
- allows arbitrary array expansions without storage reorganization

INTRODUCTION – MORTON ORDER

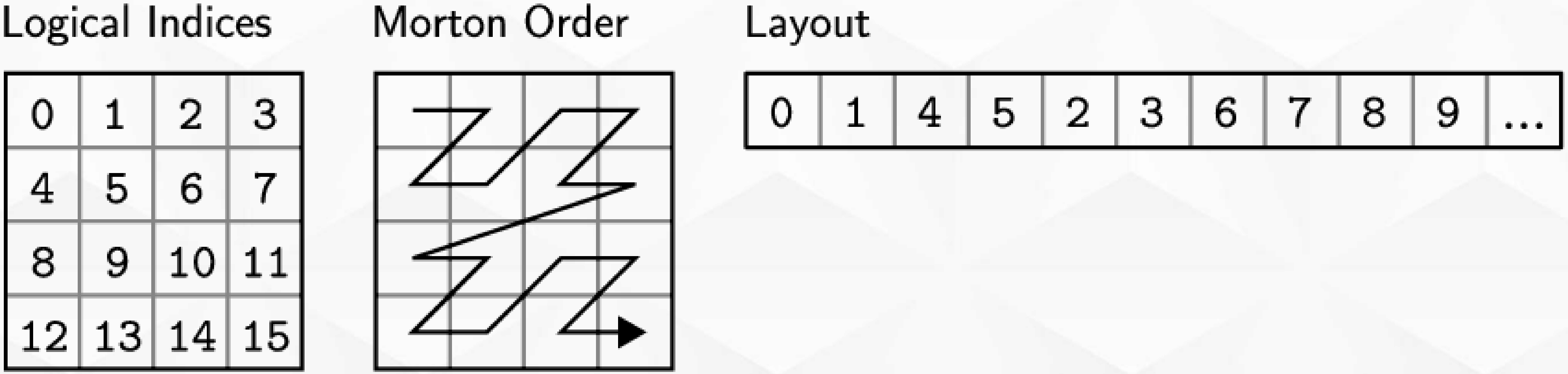

Illustration of the Morton order, indexed using the Z-curve.

INTRODUCTION

COST OF A QUERY

Related to the number of chunks that overlap the sub-array defined by the query.

CHUNK SIZE PROBLEM

- large chunk sizes => unnecessary reads
- small chunk sizes => more disk accesses

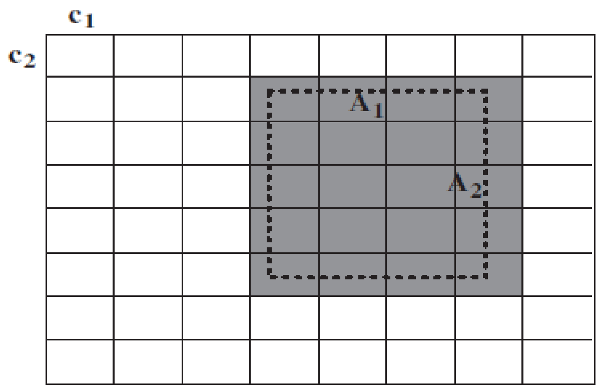

Figure 2: Example of a query and chunks retrieved by it.

OPTIMAL CHUNK PROBLEM

Given chunk size, which chunk shape to use to minimize expected number of chunks crossed by query

CONTRIBUTIONS

- Development of two accurate mathematical models of the problem
- Derivation of optimal solutions
- Experiments using synthetic workloads
- Experiments using real life datasets.

RELATED WORK

Problem introduced by Sarawagi and Stonebraker

- gave an approximate solution to this problem
- only approximate answer
- can deviate significantly from the true answer

Nearly all applications using disk resident large scale multidimensional arrays and MOLAP use chunking for physical organization.

RELATED WORK

- Chunks of size the block size of the disk storage system as tie breaker for the **chunk shape size problem**
- Chunk compression is further used to improve storage utilization
- Chunking schemes not driven by the query access pattern.
- Some research on handling extendibility in chunked arrays

The Block Compressed Row storage (BCRS)

- basis of a typical chunk addressing method
- Each block has a coordinate index <i,j>
- Linear mapping function maps coordinates to integer
- Reverse function used for computing neighbors

RELATED WORK

Prediction of query access patterns

- Based on query statistics (query history logs, sampling, etc.)
- Can be efficiently used to select chunk dimensions
- Significant reduction in the cost of answering queries

MODELS FOR QUERY ACCESS PATTERN PREDICTION

Query represented as <A1,A2, . . . ,Ak>

I-Independent Attribute Range (IAR)

- Compute probabilistic distribution of the possible range values
- query range in each dimension is independent from each other

II-Query Shape (QS)

- From Sarawagi and Stonebraker
- Probability distribution of complete query shapes

MODELS FOR QUERY ACCESS PATTERN PREDICTION

Example workload

Table 2: Queries

Query number	Query	shape
1	$< 1:3, 2:5 >$	$< 2, 3 >$
2	$< 4:7, 6:10 >$	$< 3, 4 >$
3	$< 5:9; 3:6 >$	$< 4, 3 >$
4	$< 6:8, 4:7 >$	$< 2, 3 >$

Dimension 1

Dimension 2

Table 3: Individual range probabilities under model (IAR)

Dimension #	Range value	Appears in query#	Range probability
1	2	1,4	1/2
1	3	2	1/4
1	4	3	1/4
2	3	1,3,4	3/4
2	4	2	1/4

MODELS FOR QUERY ACCESS PATTERN PREDICTION

Example workload

Table 2: Queries

Query number	Query	shape
1	$< 1:3, 2:5 >$	$< 2, 3 >$
2	$< 4:7, 6:10 >$	$< 3, 4 >$
3	$< 5:9; 3:6 >$	$< 4, 3 >$
4	$< 6:8, 4:7 >$	$< 2, 3 >$

First model

Second model

Table 4: Shape probabilities under the two models

Query Shape	Prob. Model (IAR)	In Query #	Prob. (QS)
$< 2, 3 >$	$1/2 \times 3/4 = 3/8$	1,4	$1/2$
$< 2, 4 >$	$1/2 \times 1/4 = 1/8$	-	0
$< 3, 3 >$	$1/4 \times 3/4 = 3/16$	-	0
$< 3, 4 >$	$1/4 \times 1/4 = 1/16$	2	$1/4$
$< 4, 3 >$	$1/4 \times 3/4 = 3/16$	3	$1/4$
$< 4, 4 >$	$1/4 \times 1/4 = 1/16$	-	0

MOTIVATING EXAMPLE

- 3D dataset
- Query shape: <40,60,120>
- Block size constraint of 212=4096.
- 2 chunk shapes tested

Formula of Sarawagi et al

Formula found in this study

Table 1: Cost summary

Option	Chunk Shape	SS Cost Eqn 4.1	Exact Cost Eqn 4.2	Relative Error
1	$< 8, 64, 8 >$	75	179.2449	139%
2	$< 8, 16, 32 >$	80	129.95	62%

According to Sarawagi and Stonebraker we should use chunk shape 1

MODELS FOR QUERY ACCESS PATTERN PREDICTION

Average query cost in Sarawagi and Stonebraker

Weighted sum of all query costs

$$\sum_{j=1}^{q} \boxed{\left(\prod_{i=1}^{k} \left\lceil \frac{A_{ij}}{c_i} \right\rceil \right)} p_j \qquad (4.1)$$

Probability of query shape

Query cost

Average query cost in this study

$$\sum_{j=1}^{q} p_j \prod_{i=1}^{k} \left(\frac{A_{ij} - 1}{c_i} + 1 \right); \qquad (4.2)$$

Chunk shape

Query shape

MINIMIZATION PROBLEMS - IAR

Average query cost →

$$\sum_{j=1}^{q} p_j \prod_{i=1}^{k} \left(\frac{A_{ij} - 1}{c_i} + 1 \right); \tag{4.2}$$

The chunk overlap minimization problem we wish to solve can be stated as follows:

$$\min \prod_{i=1}^{k} \left(\frac{\bar{A}_i}{c_i} + 1 \right) \tag{4.1}$$

Subject to

$$\prod_{i=1}^{k} c_i \leq C \tag{4.2}$$

We want to find the optimal chunk shape s.t. minimizes query cost and of size disk block

$\bar{A}_i = \sum_{j=1}^{m_i} p_{ij}(A_{ij} - 1)$

MINIMIZATION PROBLEMS - QS

Average query cost →

$$\sum_{j=1}^{q} p_j \prod_{i=1}^{k} \left(\frac{A_{ij} - 1}{c_i} + 1 \right); \tag{4.2}$$

$$\min \sum_{j=1}^{q} p_j \prod_{i=1}^{k} \left(\frac{A_{ij}}{2^{y_i}} + 1 \right); \quad s.t. \sum_{i=1}^{k} y_i \leq C'; \ \mathbf{y} \in S \tag{4.4}$$

with $C' = \log_2 C$ and let $y_i = \log_2 C_i$

OPTIMAL SOLUTIONS

Analysis of the Independent Attribute Range (IAR) Model

- closed form of optimization problem solved

Analysis of the shape model

- no closed form solution
- used greedy algorithm

A **greedy algorithm** is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage [wikipedia]

EXPERIMENTAL RESULTS - SIMULATION

- **Using QS model**: current study VS [Sarawagi and Stonebraker]
- Under an assumption of equal probability of shapes
- Same c
- Random values of A
- Tested the accuracy relative to the real number of overlaping chunks
- Random queries on 2, 3, 4 and 5 dimensions

Environment

- 1.8 GHz AMD Athlon 64 microprocessor
- 1 GB main memory
- Ubuntu 6.06.1 LTS Linux

EXPERIMENTAL RESULTS - SIMULATION

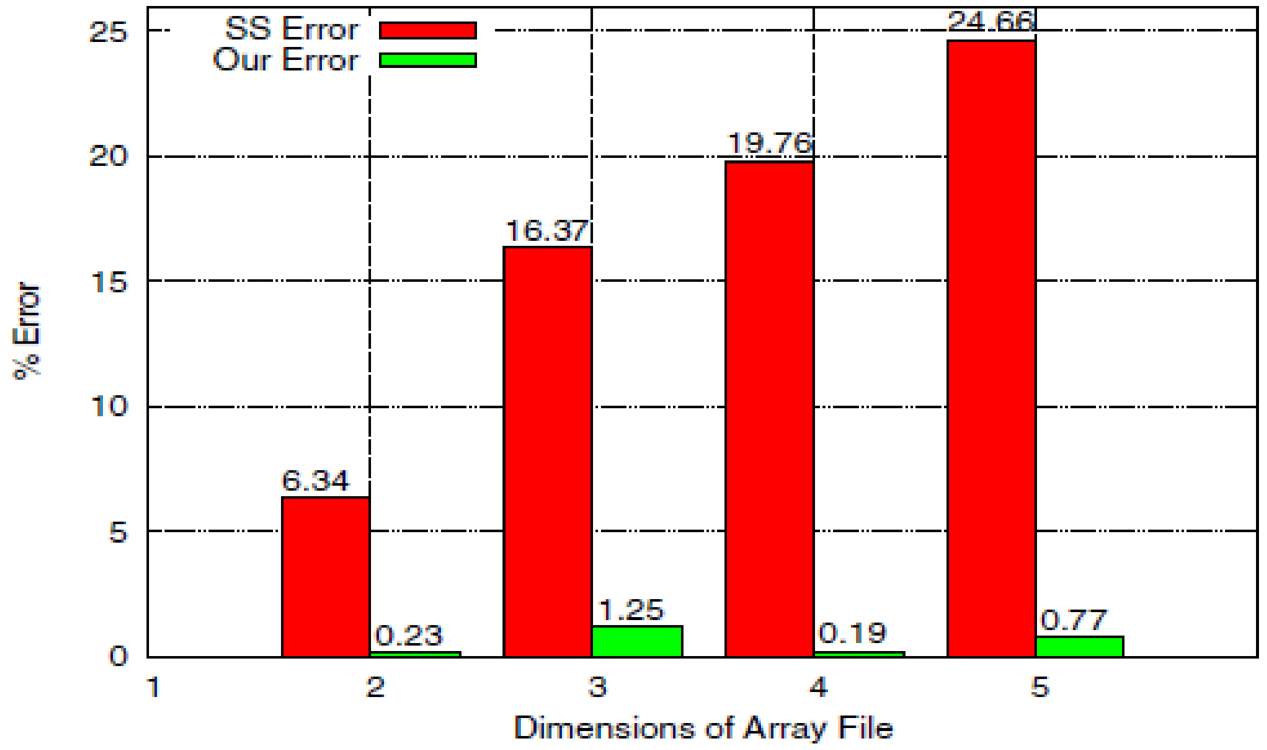

Figure 3: Bar Charts of Percentage Errors for Number of Chunks Retrieved

EXPERIMENTAL RESULTS – REAL DATA

Dataset

- Sloan Digital Sky Survey (SDSS)
- astronomical survey project
- 168 million records
- 500 attributes

« maps one quarter of the entire sky in order to determine the positions and absolute brightnesses of more than 100 million celestial objects »

EXPERIMENTAL RESULTS – REAL DATA

Feature selection

- selected 4 representative attributes -> most commonly used
- extensive study of the real query workloads from astronomers
- extracted 5,000 queries

EXPERIMENTAL RESULTS – REAL DATA

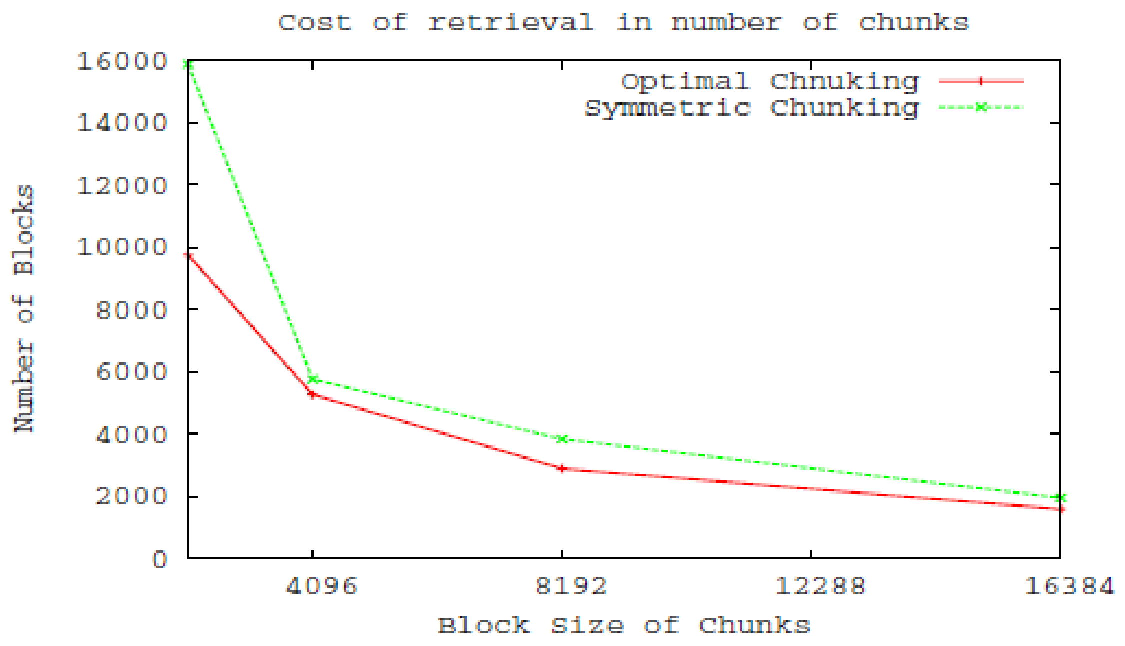

Figure 4: Comparison of Optimal and Symmetric chunking

Symmetric chunking: « chuck shapes in which all dimensions have equal sizes »

EXPERIMENTAL RESULTS – REAL DATA

Draw random query shapes

Test the queries on different block sizes (disk storage system)

attrs.	dec	ra	u	z	Opt. chunk cost	Sym. chunk. cost
adj. avg. range	22.7	54.79	146.04	71.5		
blk. 2048	2	8	16	8	9755.44	15862.39
blk. 4096	4	8	16	8	5272.677	5763.278
blk. 8192	4	8	32	8	2896.653	3846.639
blk. 16384	4	8	32	16	1594.07	1961.929

Table 8: Chunk sizes for different block sizes for SDSS data

CONLUSION

- Development of two accurate mathematical models of the problem
- Derivation of optimal solutions
- Experiments using synthetic workloads
- Experiments using real life datasets.