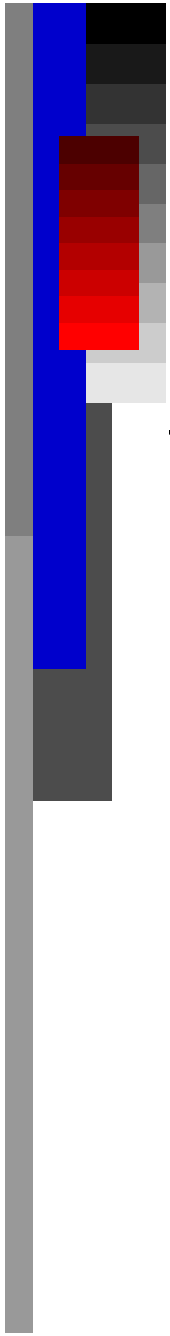
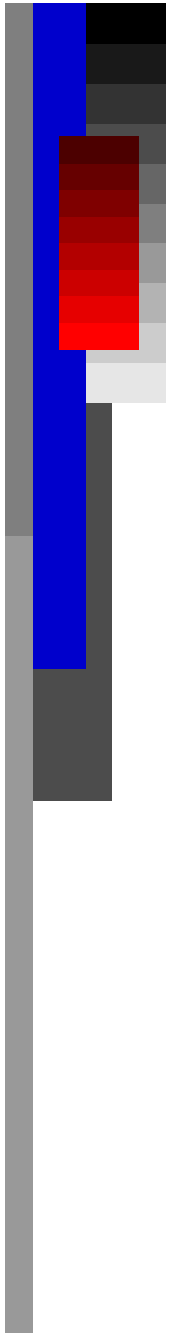


TERTIARY STORAGE ORGANIZATION FOR LARGE MULTIDIMENSIONAL DATASETS

Sachin More

Department of Electrical and Computer Engineering,
Northwestern University

March 28, 2000



Introduction

- Large multidimensional datasets are found in data warehousing, satellite data processing, high energy physics ...
- Large amounts of data ($>$ 1 terabyte), stored in tertiary storage
- Data permutation problem
  - Data exists/generated in *native order*
  - To be converted into *storage order*
  - Limited permutation space

Data permutation by multiple passes over input data

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A two dimensional 16 element dataset

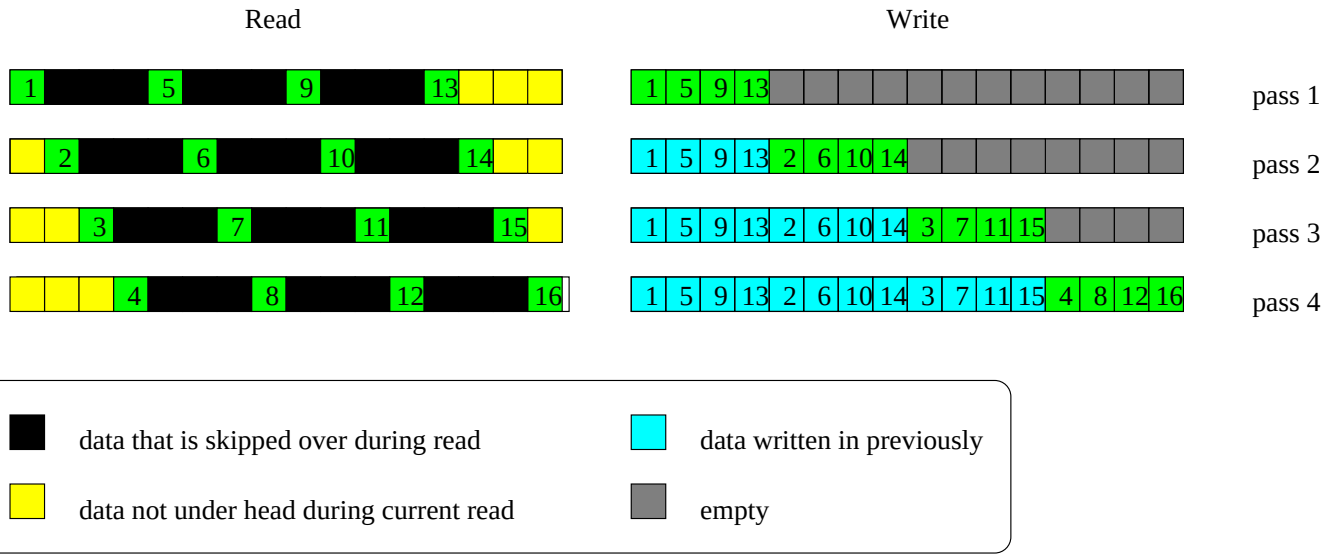
generated in native order

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

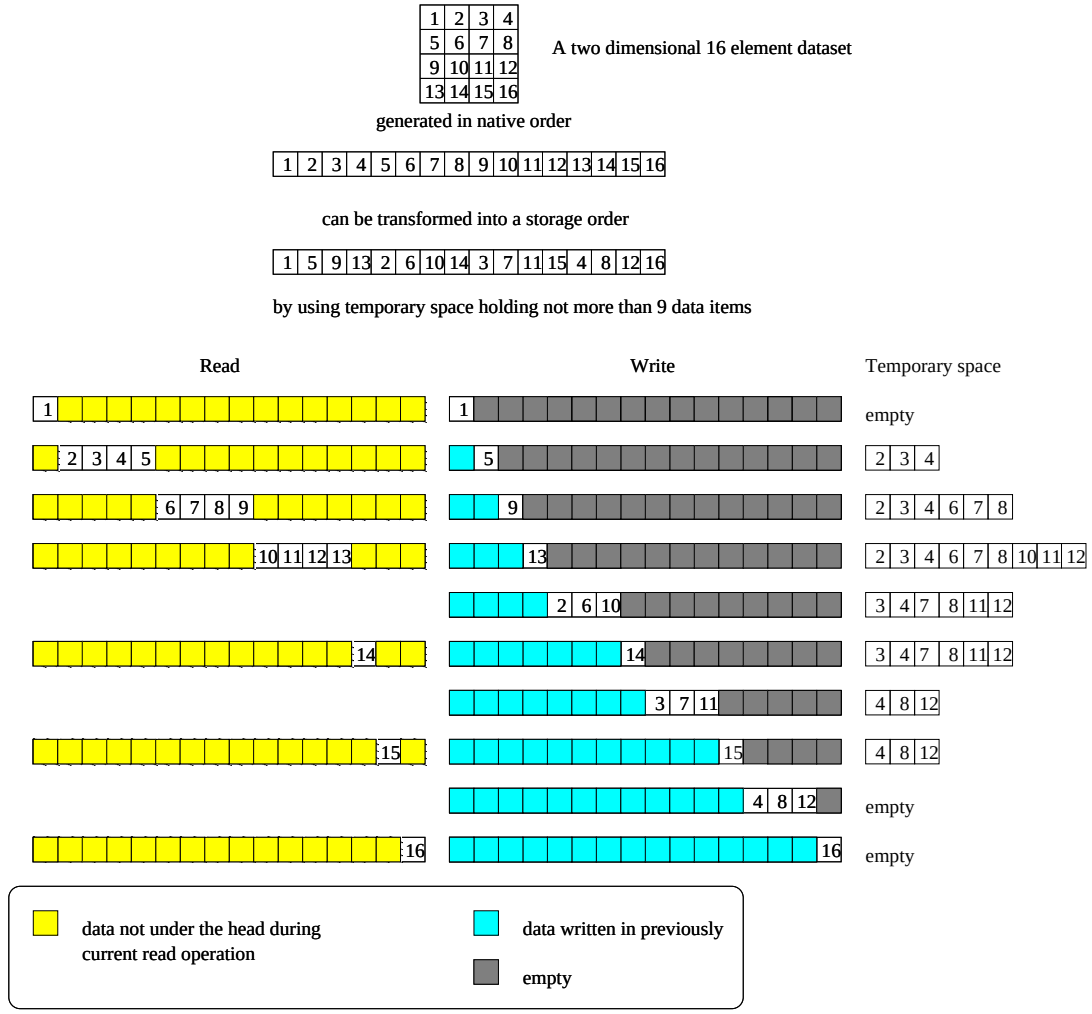
can be transformed into a storage order

1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

by making 4 passes over input dataset



Data permutation by using temporary space



Same access pattern, more than one suitable storage pattern

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

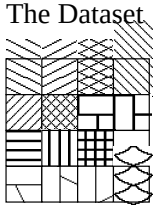
A two dimensional 16 element dataset

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Row-major ordering
2	1	3	4	5	7	6	8	10	11	12	9	16	15	14	13	Ordering 1
9	10	11	12	5	6	7	8	13	14	15	16	1	2	3	4	Ordering 2
10	11	12	9	5	7	6	8	16	15	14	13	2	1	3	4	Ordering 3

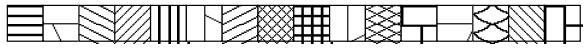
In Native Order

Can Produce the Storage Order

The Dataset



5	7	13	15
6	8	14	16
1	3	9	11
2	4	10	12



13	15	5	7
14	16	6	8
9	11	1	3
10	12	2	4

Different native orders for the same access pattern can mean different storage orders under constrained permutation space

Computing Affinities

Given two data items in the dataset, I_i and I_j, and a set of expected query types and their execution probabilities, we define $p(I_i, I_j)$ as the *affinity* between I_i and I_j. Affinity between two data items is the probability that if I_i is accessed by a query then I_j is also accessed and can be compute as:

$$\sum_{Q_i=Q_1}^{Q_k}(p_i \times \prod_{D_j=D_1}^{D_n} \lambda_i^j)$$

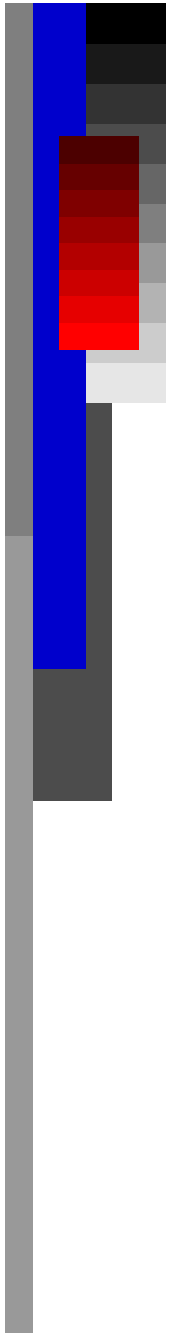
where,

Q_i	i^{th} query type	
p_i	execution probability of Q_i	
D_j	j^{th} dimension	
λ_i^j	$=$ 1	when $Q_i(j) = ALL$
	$=$ 1	when $Q_i(j) = ANY$ and $I_1(j) = I_2(j)$
	$=$ 1	when $Q_i(j) = VALUE$ and $I_1(j) = I_2(j) = v_i^j$
	$=$ 1	when $Q_i(j) = RANGE$ and $\|I_1(j) - I_2(j)\| \leq r_i^j$
	$=$ 0	otherwise
$Q_i(j)$	selection criteria of Q_i in D_j	
$I_1(j)$	coordinate of I_1 in D_j	
$I_2(j)$	coordinate of I_2 in D_j	
v_i^j	value parameter for Q_i from the domain of D_j	
r_i^j	range parameter for Q_i from the domain of D_j	

Greedy Heuristic

```
while (there are more data items)
  if (there is space left in the temporary storage)
    input "some" data items
  endif
  if (there are data items in the temporary storage)
    output "some" data items
  endif
endwhile
```

- If there are $d \leq D$ pages free in the temporary storage, how many data items of size S pages to input in each iteration?
- If there are k data items ($k \times S \leq D$) in the temporary storage, how many and which of the k data items to output in each iteration?



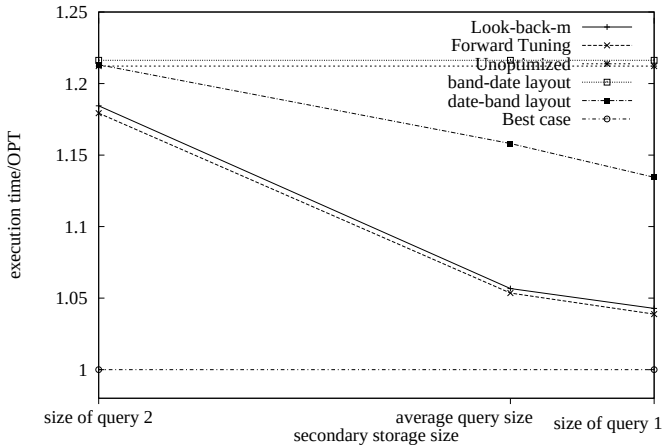
Algorithms

- Look-back m : remember last m data items
- Forward tuning : Look-back m with ties resolved
- Date-Band, Band-Date : fixed order strategies

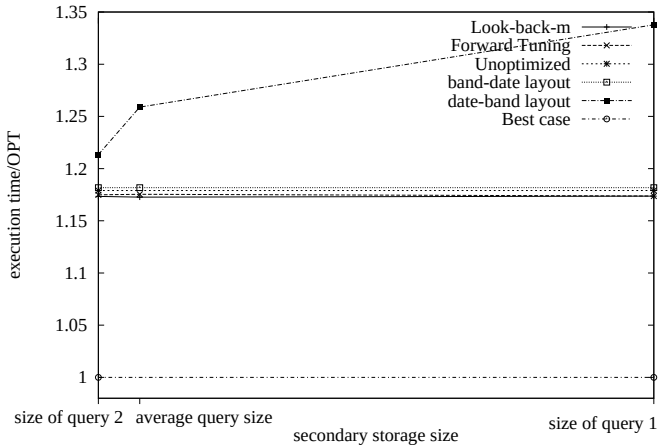
Workloads

- Workload 1 : native order differs substantially from storage order
- Workload 2 : native order differs very little from storage order

Performance Results



Workload 1



Workload 2