CSCI317 – Database Performance Tuning

Tutorial - Indexing (Read Blocks)

Sionggo Japit
sjapit@uow.edu.au

17 August 2022

Question

Assume that relational table SHIPMENT (supplier#, part#, quantity, sdate) contains 10^6 rows, the attributes (supplier#, part#, sdate) form a composite primary key.

Assume that all shipments have been done by 20 suppliers, quantities of shipment vary from 1 to 100 with the same probability of each value of quantity, average number of rows per disk block is equal to 10, block size is 2 K.

Question

Assume that all primary keys are automatically indexed by a database system and that such index is implemented as non-clustered B*-Tree. The height of this index is equal to 3.

Moreover, a database administrator created a non-clustered B*-Tree index on attribute QUANTITY and found that height of this index is equal to 1.

Understanding of the question

Assume that relational table SHIPMENT (supplier#, part#, quantity, sdate) contains 10^6 rows, the attributes (supplier#, part#, sdate) form a composite primary key.

- The SHIPMENT table consists of 1,000,000 rows (records), and its primary key is a composite primary key consisting of the attributes supplier#, part#, and sdate.

Understanding of the question

SHIPMENT

Supplier#	**Part#**	Quantity	**SDate**	
S1	P1	90	10 Jan 2011	1
S1	P4	10	31 Jan 2011	2
S2	P3	5	27 Jan 2011	3
.	.	.	.	.
.	.	.	.	.
S20	P1	50	9 Feb 2011	.
.	.	.	.	.
.	.	.	.	.
S7	P50	20	1 Dec 2011	10^6 rows

Understanding of the question

Primary Key Index

Supplier#	**Part#**	**Sdate**	RowPointer
S1	P1	10 Jan 2011	1
S1	P1	27 Jan 2011	13
S1	P3	20 JN 2011	8
.	.	.	.
.	.	.	.
S2	P2	5 Jan 2011	79
S2	P3	2 Jan 2011	7
.	.	.	.
.	.	.	.

Understanding of the question

Assume that all shipments have been done by 20 suppliers, quantities of shipment vary from 1 to 100 with the same probability of each value of quantity, average number of rows per disk block is equal to 10, block size is 2 K.

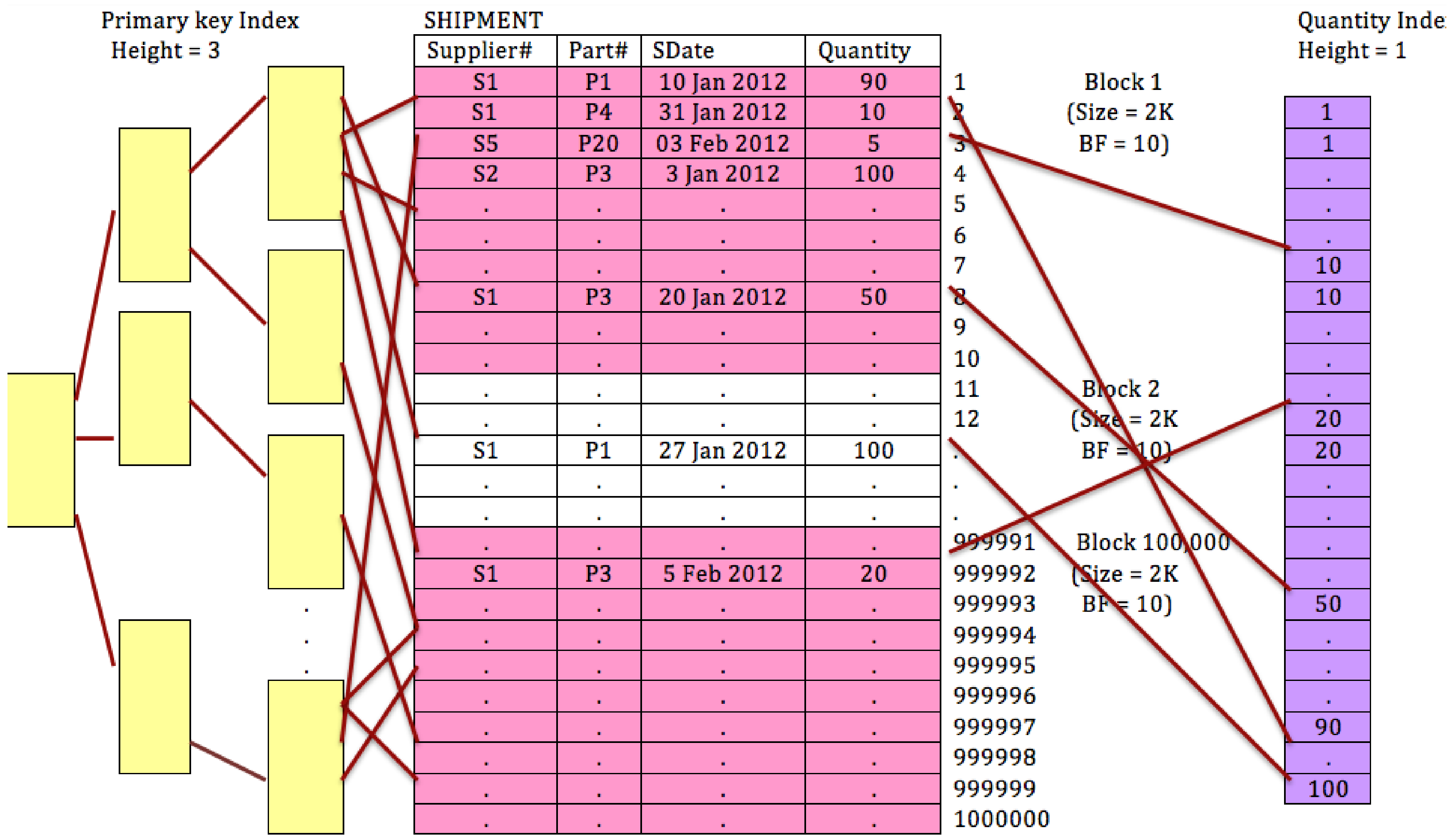
- **20 suppliers supplying parts, with quantities of shipment vary from 1 to 100, with <u>same probability</u>.**
- **Thus on an average, each supplier has 50,000 shipment records (1,000,000/20 = 50,000)**

Understanding of the question

Assume that all primary keys are automatically indexed by a database system and that such index is implemented as non-clustered B*-Tree. The height of this index is equal to 3.

Moreover, a database administrator created a non-clustered B*-Tree index on attribute QUANTITY and found that height of this index is equal to 1.

Understanding of the question



SHIPMENT

Supplier#	Part#	SDate	Quantity	
S1	P1	10 Jan 2012	90	1
S1	P4	31 Jan 2012	10	2
S5	P20	03 Feb 2012	5	3
S2	P3	3 Jan 2012	100	4
.	.	.	.	5
.	.	.	.	6
.	.	.	.	7
S1	P3	20 Jan 2012	50	8
.	.	.	.	9
.	.	.	.	10
.	.	.	.	11
.	.	.	.	12
S1	P1	27 Jan 2012	100	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	999991
S1	P3	5 Feb 2012	20	999992
.	.	.	.	999993
.	.	.	.	999994
.	.	.	.	999995
.	.	.	.	999996
.	.	.	.	999997
.	.	.	.	999998
.	.	.	.	999999
.	.	.	.	1000000

Understanding of the question

Primary key Index

A block of primary key index. ➔

Multiple blocks to contain the keys of all shipment records (about 50000) supplied by supplier S1 for the various parts on different shipment date.

Supplier#	Part#	Sdate	Row Pointer
S1	P1	10 Jan 2012	1
S1	P1	27 Jan 2012	13
S1	P3	20 Jan 2012	8
S1	P3	5 Feb 2012	999992
S1	P4	31 Jan 2012	2
S1	P10	28 Jan 2012	.
.	.	.	.
.	.	.	.
.	.	.	.
S1	P45	02 Feb 2012	38

Understanding of the question

Quantity Index

A block of quantity index. ➔

Quantity	Row Pointer
1	10
1	3
1	28
1	1
2	2
2	.
2	999992
.	.
.	.
100	45

(1) - Question

How many read block operations are needed to compute the following query?

```
SELECT    *
FROM      SHIPMENT
WHERE     SUPPLIER# = 123456;
```

Show your calculations. There is no need to compute a value of log function.

(1) – Suggested Solution:

- We access an index on primary key (SUPPLIER#, PART#, SDATE).
- Next, we vertically traverse the index to find a leaf-level block with a key that starts from123456 (3 block read operations).
- Next we use the pointers associated with the key to access a relational table SHIPMENT. (average number pointers associated with a key that starts from 123456 equals $10^6/20 = 5*10^4$,
- then in the best case when all rows with the same value of SUPPLIER# are grouped in the adjacent blocks we need $(5*10^4)/10 = 5 * 10^3$ read block operations,
- in the worst case each row is in a different block, then we need $5*10^4$ read block operations, average: $(5*10^3 + 5*10^4)/2$)
- Hence total number of read block operations = $3 + (5*10^3 + 5*10^4)/2) = 27{,}503$.

(2) - Question

How many read block operations are needed to compute the following query?

```
SELECT      *
FROM        SHIPMENT
WHERE SDATE > ‘1-JAN-2000’ ;
```

Show your calculations. There is no need to compute a value of log functions.

(2) - Suggested Solution

Algorithm

- We are unable to use an index (please don't ask me why). So, sequential scan of table SHIPMENT is the only option (total number of rows equal to 10^6, with 10 rows per block means that we have to read 10^5 blocks).

(3) - Question

How many read block operations are needed to compute the following query?

```
SELECT    QUANTITY
FROM      SHIPMENT
WHERE     SUPPLIER# = ‘123456’
AND       SDATE = ‘1-JAN-2000’
AND       PART#= ‘777888’;
```

Show your calculations. There is no need to compute a value of log function.

(3) – Suggested Solution

Algorithm

- We access an index on primary key (SUPPLIER#, PART#, SDATE).
- Next, we vertically traverse the index to find a leaf-level block with a key [123456 777888 1-JAN-2000] (3 block read operations).
- Next we use a pointer associated with the key to access a relational table SHIPMENT. (1 read block operation)
- Hence a total of (3 + 1) = 4 read blocks are carried out.

(4) - Question

```
SELECT    *
FROM      SHIPMENT
WHERE     SUPPLIER# = ‘123456’
AND       SDATE = ‘10-JAN-2012’
AND       PART# = ‘7777888’
AND       QUANTITY = 100;
```

(4) – Suggested Solution

Algorithm

- The attributes (Supplier#, Part#, Sdate) is a composite primary key, and an index on quantity is also exist, the system will vertically traverse the primary key index to locate the key in leaf level (height of primary key index = 3 read block operations). Once the key is located, the pointer (Rowid) is used to read the required row from data file (1 read block operation).
- If the row just read satisfies the second condition (quantity=100), the row is return, otherwise it is discarded.
- Hence a total of (3 + 1) = 4 read blocks are carried out.

(4) - Question

```
SELECT      *
FROM        SHIPMENT
WHERE       SUPPLIER# = ‘123456’
AND         QUANTITY = 100;
```

(4) – Suggested Solution

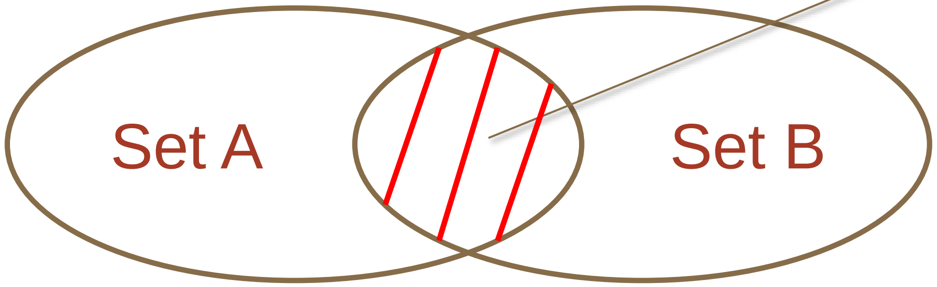
Algorithm

- Both the logically ‘AND’ conditions can be satisfied using indexes, but none of the indexes used is a primary key index, hence the system will access the subsets and fetch the ‘**intersected**’ rows.
- The condition supplier# = 123456 can be satisfied using primary key index. The system will vertically traverse the primary key index to find a set of rows that satisfy the condition supplier# = 123456, (we call this set A).
- The system next traverse the quantity index to find a second set of rows that satisfy the condition quantity = 100, and we call this set B.

(4) – Suggested Solution

Algorithm (continue…)

- Next, the system will compute an intersection of the two sets (set A and set B), and using the set of row identifiers obtained from the intersection to read data blocks from the relational table SHIPMENT.



The intersection consists of rows that meet the logically AND condition specified in the WHERE clause.

(4) – Suggested Solution

Algorithm (continue…)

- Total number of blocks read = total number of index blocks read to vertically traverse the primary key index + total number of index blocks read to vertically travers the quantity index + total number of data blocks read from the relational table SHIPMENT as a result of the intersection.

(4) – Suggested Solution

Algorithm (continue…)

- Total number of index blocks read to vertically traverse the primary key index = height of the primary key index = 3.
- Total number of index blocks read to vertically traverse the quantity index = height of the quantity index = 1.

(4) – Suggested Solution

Algorithm (continue…)

- Total number of data blocks read from the result of the intersection = (best case + worst case)/2.
- Best case for the first condition, that is, supplier# = 123456 = ((1/20) * 10^6) / blocking factor = 50,000 / 10 = 5,000.
- Worst case for the first condition = ((1/20) * 10^6) / 1) = 50,000.
- Best case for quantity index = ((1/100) * 10^6) / blocking factor = 10,000/10 = 1000.
- Worst case for quantity index = 10^6/100 = 10^4 = 10,000.

(4) – Suggested Solution

Algorithm (continue…)

- The best case for Set A:
  - ((1/20) * 1,000,000) / blocking factor
- The best case for Set B:
  - ((1/100) * 1,000,000) / blocking factor
- The combined best case:
  - = (((1/20) * 1,000,000) * ((1/100) * 1,000,000)) / blocking factor
  - = (((1/20) * (1/100)) * 1,000,000,000,000) / blocking factor
  - = ((1/2,000) * 1,000,000,000,000) / 10
  - = (500,000,000 / 10) = 50,000,000.

(4) – Suggested Solution

Algorithm (continue…)

- The worst case for Set A:
  - ((1/20) * 1,000,000) / 1
- The worst case for Set B:
  - ((1/100) * 1,000,000) / 1

[Note: in one block only 1 record satisfies the condition.]

- The combined worst case:
  - = (((1/20) * 1,000,000) * ((1/100) * 1,000,000)) / 1
  - = (((1/20) * (1/100)) * 1,000,000,000,000) / 1
  - = ((1/2,000) * 1,000,000,000,000) / 1
  - = (500,000,000 / 1) = 500,000,000.

(4) – Suggested Solution

Algorithm (continue…)

- The average case = (best case + worst case) / 2
  = (50,000,000 + 500,000,000) / 2
  = 275,000,000 read blocks
- Hence, total read block = 3 + 1 + 275,000,000
  = 275,000,004 blocks.

(5) - Question

```
SELECT     *
FROM       SHIPMENT
WHERE      SUPPLIER# = '123456'
OR         QUANTITY = 100;
```

(5) – Suggested Solution

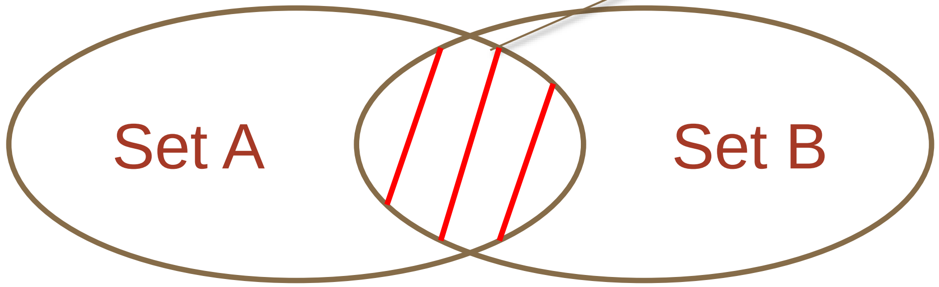
Algorithm

- Both the logically ‘OR’ conditions can be satisfied using indexes, but none of the indexes used is a primary key index, hence the system will access the subsets and fetch the ‘**unioned**’ rows.
- The condition supplier# = 123456 can be satisfied using primary key index. The system will vertically traverse the primary key index to find a set of rows that satisfy the condition supplier# = 123456, (we call this set A).
- The system next traverse the quantity index to find a second set of rows that satisfy the condition quantity = 100, and we call this set B.

(5) – Suggested Solution

Algorithm (continue…)

- Next, the system will compute a union of the two sets (set A and set B), and using the set of row identifiers obtained from the union to read data blocks from the relational table SHIPMENT.



Overlapping, and hence two instance of each rows exist. Need to remove one of the instance.

(5) – Suggested Solution

Algorithm (continue…)

- Total number of blocks read = total number of index blocks read to vertically traverse the primary key index + total number of index blocks read to vertically travers the quantity index + total number of data blocks read from the relational table SHIPMENT as a result of the union.

(5) – Suggested Solution

Algorithm (continue…)

- Total number of index blocks read to vertically traverse the primary key index = height of the primary key index = 3.
- Total number of index blocks read to vertically traverse the quantity index = height of the quantity index = 1.

(5) – Suggested Solution

- Instances (rows) that are overlapping = ((1/20) x (1/100)) x 10^6 = 500.
- These number of rows need to be removed from the combined cases.

(5) – Suggested Solution

Algorithm (continue…)

- Total number of data blocks read from the result of the union = (best case + worst case)/2.
- Best case for the first condition, that is, supplier# = 123456 = ((1/20) * 10^6) / blocking factor = 50,000 / 10 = 5,000.
- Worst case for the first condition = ((1/20) * 10^6) / 1) = 50,000.
- Best case for quantity index = ((1/100) * 10^6) / blocking factor = 10,000/10 = 1000.
- Worst case for quantity index = 10^6/100 = 10^4 = 10,000.

(5) – Suggested Solution

Algorithm (continue…)

- The best case for Set A:
  - ((1/20) * 1,000,000) / blocking factor
- The best case for Set B:
  - ((1/100) * 1,000,000) / blocking factor
- Combined best case = (((1/20) x 10^6) + ((1/100) x 10^6)) – overlapping instances) / 10

  = ((1/20 + 1/100) x 10^6)) – 500) / 10

  = ((6/100) x 10^6 – 500) / 10

  = 59,500 / 10

  = 5,950 blocks

(5) – Suggested Solution

Algorithm (continue…)

- The worst case for Set A:
  - ((1/20) x 1,000,000) / 1
- The worst case for Set B:
  - ((1/100) x 1,000,000) / 1

[Note: in one block only 1 record satisfies the condition.]

- The combined worst case:
  - = (((1/20) x 1,000,000) + ((1/100) x 1,000,000) - overlapping) / 1
  - =((((1/20) + (1/100)) x 1,000,000) – 500) / 1
  - = ((6/100) x 1,000,000) – 500 / 1
  - = (59,500 / 1) = 59,500.

(5) – Suggested Solution

Algorithm (continue…)

- The average case = (best case + worst case) / 2
  = (5,950 + 59,500) / 2
  = 32,725 read blocks
- Hence, total read block = 3 + 1 + 32,725
  = 32,729 blocks.

(6) - Question

```
SELECT    *
FROM      SHIPMENT
WHERE     SDATE = ’01-Jan-2001’
OR        QUANTITY = 100;
```

Since there is no index on sdate, and the conditions in the ‘WHERE’ clause is ‘OR’ed, the system will perform a full-table scan. Hence the total number of data block read = $10^6/10 = 10^5$.

(7) - Question

```
Select quantity
From		SHIPMENT;
```

No ‘where’ clause, but the attribute quantity can be obtained from the quantity index. Hence the system will traverse the quantity index horizontally at leaf level. Display the same value of a key as many times as many row identifiers are associated with a key.

Total number of read block operations = 1.

(8) - Question

```
Select count(distinct quantity)
From        SHIPMENT;
```

No ‘where’ clause, however, the information on quantity can be obtained from the quantity index. Hence the system will traverse the quantity index horizontally at leaf level and count the total number of keys.

Total number of read block operations = 1.

(9) - Question

```
Select count(*)
From		SHIPMENT
Where	quantity = 90;
```

Query contains ‘where’ clause, and the condition can be obtained through the index on quantity. Hence the system will traverse the quantity index vertically and find a key 90. The system will then count the identifiers of rows associated with the index key.

Total number of read block operations = 1.

(10) - Question

```
Select    count(*)
From      SHIPMENT;
```

There is no ‘where’ clause in the query, however, the information for the count(*) can be obtained from the primary key index. The system will horizontally traverse the leaf level of the primary key index and count the total number of key.

Total number of read block operation = 10,000.   (How to obtain 10,000 ?)

(10) - Question

Height of a tree = $\log_b n$

$$3 = \log_b 1000000$$

$$3 = \log_b 1000000$$

$$b^3 = 1000000$$

$$b = \sqrt[3]{1000000}$$

$$b = \sqrt[3]{\left(10^2\right)^3} = 100$$

Level 0:	1
Level 1:	100
Level 2:	10,000

(11) - Question

```
Select max(sdate)
From        SHIPMENT;
```

No ‘where’ clause, and information on sdate cannot be found from any of the indexes available. The system will perform a full table scan.

Total number of read block operations = 10^5.

(12) - Question

```
Select Max(supplier#)
From          SHIPMENT;
```

No ‘where’ clause, but the information about maximum supplier# can be found from the primary key index. Since the primary key index is ordered in ascending order at leaf level, the system will access the last leaf block and get the key (supplier#) from the last entry of the block.

Total number of read block operation = 1.

(13) - Question

```
Select *
From        SHIPMENT
Order by    supplier#;
```

There is no ‘where’ clause and the information to be retrieved are from the data file. However, the output need to be ordered by supplier#, and this information can be obtained from the primary key index. The system will traverse horizontally at the leaf level of the primary key, which are sorted by supplier# in ascending order. With each key, the system will make use of the row identifier to access the row (record) from the data file.

Total number of read block operation = total number of leaf block of primary key index + total number of records.
= 10,000 + 1,000,000 = 1,010,000

(14) - Question

```
Select      supplier#
From        SHIPMENT
Order by    supplier#;
```

There is no ‘where’ clause and the information to be retrieved are from the data file. However, the output need to be ordered by supplier#, and this information can be obtained from the primary key index. The system will traverse horizontally at the leaf level of the primary key, which are sorted by supplier# in ascending order. Since the supplier# can be obtained from the leaf nodes, the system will not access the data file.

Total number of read block operation = total number of leaf block of primary key index.
= 10,000