Advanced Data Engineering: Assignment 8

NGUYEN T. Hoang - SID: 15M54097
(ホアン)

Fall 2015, W831 Tue. Period 5-6
Due date: 2015/12/15

Problem

- Estimate costs (rough excution time) for Horizontal Partition.
- Estimate costs (rough excution time) for Vertical Partition.

Assuming:

- Disk transfer bandwidth: 10MB/s.
- Cardinality of a relation R: 100,000.
- Search condition:
  - (a) Attribute X = 123 (4B integer)
  - (b) Attribute Y = 456 (4B integer)
- The number of tuples satisfying condition a: 1,000.
- The number of tuples satisfying condition b: 500.
- The number of tuples satisfying both condition: 100.
- The size for TID: 4B
- Attributes included in output: X and Y
- The number of processors: 2
- Network bandwidth: 10MB/s.

Question 1: Horizontal Partition

Estimate cost for selection operation with Horizontal Partition scheme.

Answer: Assume the disk used in this database system is a standard Hard Disk Drive with $4KB$ page size and $4.7ms$ page access time. Also, I consider horizontal Round-Robin Partitioning.
In the Round-Robin Partitioning, I assume there are two possible configuration: unsorted data and B-Tree Indexing for X and Y. The cost formula for estimating selection is stated as follow:

$$cost_t = P_{\text{scan}} + P_{\text{compare}} + P_{\text{write to disk}} + S_{\text{communicate}} + S_{\text{join}}$$

Here, P denotes parallel processes that run on both processors and S denotes sequential processes that run on one processor. The cost includes time to scan the table, time to compare, time to write result to disk, time to send result to other processor and time for the other process join the results.

1.1 Linear scan

The data is scan sequentially in each processors. The number of data tuples scanned for each processor is:

$$\#\text{Scan} = \{R'\} = \frac{R}{\#\text{processor}} = \frac{R}{2} = \frac{100,000}{2} = 50,000 \text{ tuples}$$

$$\#\text{Page Access} = \frac{\#\text{Scan} \times \mathit{length}\ \text{tuple}}{\text{Page Size}} = \frac{50,000 \times 1,000B}{4KB} \approx 12,500 \text{ times}$$

Therefore, we have the cost for parallel linear scan:

$$P_{\text{scan}} = \#\text{Page Access} \times T_{\text{ disk page access}} = 12,500 \times 4.7ms = 58,750\ ms$$

The number of comparision to derive tuples satisfying both condition (a) and (b) is $\{R'\} \times 2$, since there are two comparisons for each tuple scanned. We have:

$$P_{\text{ compare}} = \{R'\} \times 2 \times T_{\text{ cpu}} = 100,000T_{\text{ cpu}}$$

Assume that the data is well distributed for X and Y all over 2 machines. Each machine will have 500 types satify condition (a), 250 tuples satisfy condition (b) and 50 tuples satifying both. The disk I/O cost to write these results to disk is:

$$P_{\text{ write to disk}} = \frac{(50) \times 8B}{PageSize} \times T_{\text{ disk page access}} \approx 0.23\ ms$$

Time to send one machine's result to another is:

$$S_{\text{communicate}} = \frac{50 \times 8B}{10MB/s} = 0.04\ ms$$

Time to concatinate result in one machine is:

$$S_{\text{join}} = P_{\text{write to disk}} = 0.23\ ms$$

In conclusion, the selection cost for this scheme is:

$$\begin{aligned} cost_{\text{RR\_L}} &= 58,750ms + 100,000\ T_{\text{cpu}} + 0.23ms + 0.04ms + 0.23ms \\ &= 58,750.5ms + 100,000\ T_{\text{cpu}} \end{aligned}$$

1.2 B-Tree Index for X and Y

For each processor, we have 50,000 tuples, therefore we need 2 B-Trees of height 2 for storing all TIDs for attributes X and Y. According to the previous assignment, each TID deriviation costs $9.8ms$. In each machine, we search for tuples that satisfy condition (a) and (b), then join the result, and send the final result to other machine. I assume join operation is search-based join.

$$P_{\text{scan}} = 9.8ms \times 750 = 7,350\ ms$$

Cost for join (a) and (b):

$$P_{\text{compare}} = 500 \times 250 \times T_{\text{cpu}}$$

$$P_{\text{write to disk}} = \frac{50 \times 8B}{PageSize} \times T_{\text{disk page access}} \approx 0.46\ ms$$

Time to send one machine's result to another is:

$$S_{\text{communicate}} = \frac{50 \times 8B}{10MB/s} = 0.04\ ms$$

Time to concatinate result in one machine is:

$$S_{\text{join}} = P_{\text{write to disk}} = 0.46\ ms$$

In conclusion, the selection cost for this scheme is:

$$\begin{aligned} cost_{\text{RR\_B}} &= 7,350ms + 125,000\ T_{\text{cpu}} + 0.23ms + 0.04ms + 0.23ms \\ &= 7,351ms + 125,000\ T_{\text{cpu}} \end{aligned}$$

Question 2: Vertical Partition

Estimate cost for selection operation with Horizontal Partition scheme.

Answer: We have the same assumption about hardware as in **Question 1**. In this vertical partition scheme, I assume that X and Y are divided and stored in two different machines. Same as before, I will consider two configuration for data in disk: unsorted and B-Tree Indexing. Also I assume the join operation is search-based since the data is small. The cost function for this scheme is:

$$cost_t = P_{\text{scan}} + P_{\text{compare}} + P_{\text{write to disk}} + S_{\text{communicate}} + S_{\text{join}}$$

2.1 Linear scan

The data is scan sequentially in each processors. The number of data sub-tuples scanned for each processor is:

$$\#\text{Scan} = \{R\} = 100,000 \text{ sub-tuples}$$

$$\#\text{Page Access} = \frac{\#\text{Scan} \times length_{\text{ tuple}}}{\text{Page Size}} = \frac{100,000 \times 8B}{4KB} \approx 200 \text{ times}$$

Therefore, we have the cost for parallel linear scan:

$$P_{\text{scan}} = \#\text{Page Access} \times T_{\text{disk page access}} = 200 \times 4.7ms = 940\ ms$$

The number of comparision to derive tuples satisfying condition (a) and (b) is:

$$P_{\text{compare}} = \{R\} \times T_{\text{cpu}} = 100,000 T_{\text{cpu}}$$

There is 1,000 sub-tuples satisfy (a) in one machine and 500 sub-tuples satisfy (b) in the other machine, therefore I use the larger set as the time for both.

$$P_{\text{write to disk}} = \frac{(1000) \times 8B}{PageSize} \times T_{\text{disk page access}} \approx 9.4\ ms$$

I assume the machine with smaller result set will send the data:

$$S_{\text{communicate}} = \frac{500 \times 8B}{10MB/s} = 0.4\ ms$$

Time to concatinate result in one machine is:

$$S_{\text{join}} = 1000 * 500 * T_{\text{cpu}} + \frac{100 \times 8B}{PageSize} \times T_{\text{disk page access}} = 500,000 T_{\text{cpu}} + 0.94$$

In conclusion, the selection cost for this scheme is:

$$cost_{\text{V\_L}} = 950ms + 600,000\ T_{\text{cpu}}$$

2.2 B-Tree Index for X and Y

For each processor, we have 100,000 tuples, therefore we need 2 B-Trees of height 2 for storing all TIDs for attributes X and Y. According to the previous assignment, each TID deriviation costs $0.8ms$. Here, I take the time of the larger set.

$$P_{\text{scan}} = 0.8ms \times 1000 = 800\ ms$$

$$P_{\text{write to disk}} = \frac{1000 \times 8B}{PageSize} \times T_{\text{disk page access}} \approx 9.2\ ms$$

Time to send one machine's result to another is:

$$S_{\text{communicate}} = \frac{500 \times 8B}{10MB/s} = 0.4\ ms$$

Time to concatinate result in one machine is:

$$S_{\text{join}} = 1000 * 500 * T_{\text{cpu}} + \frac{100 \times 8B}{PageSize} \times T_{\text{disk page access}} = 500,000T_{\text{cpu}} + 0.94$$

In conclusion, the selection cost for this scheme is:

$$cost_{\text{V\_B}} = 811ms + 500,000\ T_{\text{cpu}}$$