Assignment 4

Index Tuning

Database Management and Tuning

Start date: April 16, 2013
Due date: April 30, 2013, 16:00
Grading: 1 point

In this assignment you will experiment with indexes using PostgreSQL 8.

1. Download `dblp.zip`. This archive contains two tab separated files (`publ.tsv` and `auth.tsv`) that store authors and their publications as found in the DBLP[1] bibliography. The imported tables have the following schemas:

   - `Auth(name(49),pubID(129))`
   - `Publ(pubID(129),type(13),title(700),booktitle(132),`
     `year(4),publisher(196))`

   You can assume that all attribute values are strings; the maximum string length is shown in brackets. `Publ.pubID` is a key.

2. Compare clustered B^+-tree, non-clustered B^+-tree, non-clustered hash index, and table scan (no index) for the following queries and measure the throughput:

   ```
   SELECT * FROM Publ WHERE pubID = ...
   SELECT * FROM Publ WHERE booktitle = ...
   SELECT * FROM Publ WHERE year = ...
   ```

   (a) Explain your experimental setup, in particular, the conditions that you used in your queries and the computation of the throughput.

   (b) Discuss your observations. Are the results expected? Why (not)?

   Notes:

   - Do *not* specify primary key, foreign key, or uniqueness constraints when you create the tables. PostgreSQL automatically creates an index to ensure uniqueness, which you want to avoid.
   - When you measure the throughput and repeat a query, do not to use the same condition in the `WHERE` clauses of the repeated queries since the database might buffer the results.

[1] `http://www.informatik.uni-trier.de/~ley/db/`

- To test the non-clustered indexes, cluster the table according to an attribute that is independent of the indexed attribute, e.g., cluster the table according to `title` for the condition on `year`.

3. Study index nested loop join, merge join, and hash join for the following queries:

```
SELECT name,title
FROM Auth, Publ
WHERE Auth.pubID=Publ.pubID;

SELECT title
FROM Auth, Publ
WHERE Auth.pubID=Publ.pubID AND Auth.name='Divesh Srivastava'
```

   (a) What join strategies does the system propose if you do not use an index, with a unique non-clustering index on `Publ.pubID`, with two clustering indexes on `pubID`?

   (b) Test the index nested loop join with an index on `Publ.pubID`, on `Auth.pubID`, and both `Publ.pubID` and `Auth.pubID`. Give the response times and discuss the query plans.

   (c) Test the merge join without index, with two non-clustering indexes, and with two clustering indexes. Give response times and discuss the query plans.

   (d) Test the hash join without index and give the response time.

   (e) Are the results (query plan and throughput) expected? Why (not)?

   Note: You can stop queries that run for more than 10 minutes on `alcor.inf.unibz.it`. Check the query plan to avoid queries with excessive runtime.

Notes about PostgreSQL

- *Clustering indexes*: You first create an index, then you use the index to cluster the table (i.e., physically sort the table by the index attribute):

```
CREATE INDEX year_idx ON publ(year);
ALTER TABLE publ CLUSTER ON year_idx;
```

- *Query plan*: The command `EXPLAIN` shows the query plan without executing the query. The command `EXPLAIN ANALYSE` also executes the query. Example:

```
EXPLAIN ANALYZE SELECT * FROM publ WHERE year='2006';
```

- *Join strategy*: You can influence the optimizer choice with the switches `enable_hashjoin`, `enable_mergejoin`, and `enable_nestloop`. Example:

```
SET enable_hashjoin TO true;
SHOW enable_hashjoin;
```

Please indicate the time that you spent solving this assignment in your report. The time that you indicate will have *no* impact on your grade.