Homework Assignment 4

15-415 Database Applications
Carnegie Mellon University

March 13, 2003
Due: March 21, 2003 (Noon)

1. Introduction

In this assignment, you will carry out a number of exercises involving the optimization of relational queries using the postgresql optimizer and the visualization command `EXPLAIN`.

You need to read parts of the "PostgreSQL 7.3 Documentation" to be able to complete this assignment. To be specific, you need to get familiar with the `EXPLAIN` and `SET` command of PostgreSQL (specific links are provided in the subsections).

2. Administrivia

This is another non-programming project. **It must be done INDIVIDUALLY.** Please read the entire assignment before beginning.

3. Setup

We provided some scripts to help you setup the experimental environment. Copy the whole directory "/usr0/dbclass/hw4" to your home directory.

Scripts:
setup.sh and table.dss

Relation Schema:
We will use three tables in this experiment: part, supplier, and partsupp.
part (p_partkey integer, p_name varchar(55));
p_partkey is the primary key of part.
supplier (s_suppkey integer, s_name char(25));
s_suppkey is the primary key of supplier.
partsupp (ps_pskey integer, ps_partkey integer, ps_suppkey integer, ps_availqty integer, ps_placed date, ps_ship date);
ps_pskey is the primary key of partsupp; ps_partkey is the foreign key of part.p_partkey; ps_suppkey is the foreign key of supplier.s_suppkey.

Data files:
Three data files are in hw4/: part.data, supplier.data, and partsupp.data

Setup Steps:
Just run "./setup.sh". It will first initialize a data cluster (default directory is hw4data at hw4/); then start the server; create a database (default name is hw4); create tables; import data into tables; update statistics. (You can change the default values when running setup.sh)

We will use the shared installation of PostgreSQL at "/usr0/local/pgsql7_3" which was also used in homework 0 and 3.

Useful documentation you need for this project.
Syntax of EXPLAIN command:
http://www.postgresql.org/docs/view.php?version=7.3&idoc=1&file=sql-explain.html
How to use EXPLAIN command and understand its output:
http://www.postgresql.org/docs/view.php?version=7.3&idoc=1&file=performance-tips.html
Check the statistics collected by PostgreSQL:
http://www.postgresql.org/docs/view.php?version=7.3&idoc=1&file=planner-stats.html
http://www.postgresql.org/docs/view.php?version=7.3&idoc=1&file=catalogs.html
Change the run-time configurations:
http://www.postgresql.org/docs/view.php?version=7.3&idoc=1&file=runtime-config.html

Remember to update statistics after adding or deleting indexes using "vacuum" and "analyze".

4. Exercises

4.1 Statistics of the tables (15%)

We will first examine the statistics for table "partsupp". Answer the following questions.

1) How many records are there actually in "partsupp"? What is the estimated value by the query optimizer? How do you find these values (command or SQL)?
2) Use PostgreSQL catalog to find the number of distinct values of each of the attributes in the "partsupp" relation. Write down the query you used to find the above information.

4.2 Index on perfect match query (30%)

We will check how index affects query optimization and performance.
Examine the following query:

```
SELECT * FROM partsupp
WHERE ps_availqty = 30;
```

1) What is the estimated total cost of executing the best plan? What does the cost of a plan mean?
2) What is the estimated result cardinality for this plan? How does the query optimizer obtain this value? Is it a reasonable one?
3) Which access method does the optimizer choose?
4) In what order would the tuples be returned by this plan? Why?

Create an index "ps_availqty_idx" on the attribute "ps_availqty". Execute "VACUUM" and "ANALYZE" to update the statistics.

5) Which access method does the optimizer consider to be the best now?

6) Compare the two plans (without and with index). Explain briefly why access method in 5) is cheaper than the previous one.

4.3 Index on range select (30%)

Consider the following query:

```
SELECT * FROM partsupp
WHERE ps_availqty < 150
```

1) How many tuples does the query optimizer think will be returned? What is the total cost?
2) How does the optimizer get this number? That is, what calculations does it perform?
3) What is the access method?
4) Disable the access method used by the optimizer in step 3). What is the total cost now? In what order would the tuples be returned by this plan? Is it the same as step 1)?
5) Explain why one of the access methods is more expensive than the other.

4.4 Join algorithm (25%)

Consider the following query:

```
SELECT DISTINCT (s_name)
FROM supplier, partsupp
WHERE s_suppkey = ps_suppkey and
        ps_availqty < 4;
```

Answer the follow questions:

1) Write down the best plan estimated by the optimizer (in plan tree form). What is the estimated total cost?
2) What is the join algorithm used in the plan?
3) According to the optimizer, how many tuples will be retrieved from partsupp?
4) Disable the join type used by the optimizer. What kind of join algorithm will be used now? What is the total cost now?
5) Disable the join type used in 2) and 4). What kind of join algorithm will be used now? What is the cost now?

5. Submission instructions

You should turn in brief answers to questions on **the template** provided. Print out the template and fill it in with your answers. *NOTE: the plans must be written down in the tree form*. Submit your answer sheet to **Minglong Shao** at **Wean Hall 1315** (if she is not there, slide your answer sheet under the door) before the deadline (**March 21st, noon**).