Lab 3

Out: 10/20/15, Due 11/24/15

This lab will be your first experience modifying PostgreSQL, a major open-source DBMS. Your assignment is to implement a different buffer management policy. The current versions of PostgreSQL uses a clock sweep algorithm to determine which buffer page should be replaced. In this lab, you will need to implement a LRU (Least Recently Used) policy alongside the existing clock sweep policy.

This lab will not feature a large amount of coding; the entire assignment can be done in roughly 100 lines of code. Most of your time will be spent looking through and understanding the existing source code, ***so make sure you get an early start***. Unlike the first two labs, you will want to complete this lab in groups of two (a group of three needs a personal confirmation from Prof. Mokbel). At least one member of your group should be highly familiar with C.

Installation

Download version 9.2.1 of PostgreSQL from http://www.postgresql.org/ftp/source/v9.2.1/ and install it to your Linux CSELabs account. Detailed installation instructions can be found at http://www.postgresql.org/docs/9.2/static/install-procedure.html. PostgreSQL is a large program, so you will want **at least** about 200MB **of free space** on your CSELabs account.

As an additional note, the standard "./configure" command should not be used during installation, as students do not have permission to install anything to the user directory. You will want to specify another folder within your home directory to install PostgreSQL to, with the command

./configure --prefix=install_path

The prefix option requires an absolute path, not a relative one. One example would be

./configure --prefix=$HOME/PostgreSQL/installation

Setup

Once the program is fully installed, we need to initialize and create a database to use. To initialize databases to a certain folder, use the following command:

install_path/bin/initdb -D install_path/data

We can then start the PostgreSQL backend by using the following command:

install_path/bin/postgres -D install_path/data

We can then use this terminal to get information on what is happening in our DBMS. Open another terminal and use the following command to create a database to use:

install_path/bin/createdb -h localhost database_name

We interact with our DBMS system using a program called psql. To start psql, use this command:

install_path/bin/psql -h localhost database_name

You can now use this terminal to enter SQL commands and manage your database. After this, whenever you start up PostgreSQL, you should only need to run the postgres and psql commands, even after rebuilding and reinstalling.

Modification

All of PostgreSQL's code regarding buffer management is stored in src/backend/storage/buffer. The functions that control the buffer replacement policy are contained in freelist.c; most, if not all, of your modifications will be to this file. Figuring out how the source code works is part of your assignment, so the TA's can only give limited help in helping you understand the code. If you like, you can use the debugging program **gdb** on the PostgreSQL backend, to take a step-by-step look at how PostgreSQL replaces its buffers. ***It's a good idea to get an early start in deciphering just how the buffer replacement function works. Leaving this lab for the last minute is an extremely bad idea.***

Whenever you modify the source code of PostgreSQL, you can rebuild and reinstall it with the commands "make" and "make install", or preferably "gmake" and "gmake install". Make sure PostgreSQL is not running when you do this.

Do not completely eliminate the old buffer replacement code. The testing phase will need to compare the default clock algorithm and your LRU algorithm side by side. The best practice is to switch from one method to the other using a debug flag and an if statement.

The PostgreSQL online source code documentation will be an invaluable resource for this lab and the next. You can find it at http://doxygen.postgresql.org.

Testing

As stated above, we will be comparing the performance of your LRU strategy and the default clock strategy. You will also be provided with test cases to use, so you may see for yourself how performances compare. The exact test cases to be run are currently TBD.

Submission

You should submit a .zip file or .tar.gz file to Moodle by 01:00 p.m. on the due date. Your submission should contain the following:

1. Any source code files that you modified.
2. A short README file.
3. A short report (one page or less) with the results of your test cases, comparing the two approaches and explaining why one might have outperformed the other.

Your README file should include:

- The names and x500 accounts of all group members.
- A list of all the source code files you modified, as well as where they belong in the src hierarchy.
- A brief description of how you modified the code.

It is preferred that all files be in text-only format. Acceptable formats for the report include text-only, pdf, and anything that can be opened by OpenOffice. Any additional information will be posted on the Moodle forums.