Buffer Manager: Project 1 Assignment
Due Date: 27-Aug-2007

Overview of Project 1 Buffer Manager

In this project you will add functionality to the backend server of PostgreSQL. You need to delve into the actual server code. In this first assignment we limit our investigation to one core module, the buffer manager. The actual coding for this project is minimal, but you will need to figure out what code to add.

The major parts of this project are:
• Understanding different memory management strategies
• Examining, understanding and modifying existing code
• Measuring the effects of your code in a real system

Tasks and Steps
(1) Learn different buffer management strategies.
(2) Compile and install PostgreSQL.
(3) Implement MFU, LFU
(4) Compare performance of MFU, LFU with LRU (make a graph).

The actual implementation of LRU is rather straightforward once you understand the existing code. The existing code is not extremely clear, but it is understandable.

The actual buffer manager code is neatly separated from the rest of the codebase. It is located in
src/backend/storage/buffer and src/include/storage.

There are two main files which will be modified primarily in addition to other files (if required), i.e., src/backend/storage/buffer/freelist.c , manages which pages are not pinned in memory, and which pages are eligible for replacement and src/include/storage/buf_internals.h, which contains the definition for each buffer description (called BufferDesc). Most of the fields in BufferDesc are managed by other parts of the code, but for efficiency some fields are used to manage the buffer space while it is eligible for replacement (when it's on the existing LRU freelist). Some initialization of the queue occurs in src/backend/storage/buffer/buf_init.c. However, you must do all your initialization in freelist.c.

You will modify these two files to change the implementation from the existing LRU queue to the LFU and MFU strategy. You may find that you do not need to edit both files or change many of the functions.

To do performance test you have to change block size to a low value in order to generate more I/Os. While initialization of server you can also set the number of buffers.

Note:

(i) There is lot of material available on internet for postgresql. There are many versions of Postgresql and you have to do this assignment on version 7.3.15. This version implements LRU. You can see clock implementation in version 8.1.0.

(ii) You have to compile Postgresql source code in your account to do this assignment. Demos will be most probably in intel lab.

(iii) For submission you have to mail to Navneet pandey a tar archive :

1. Give a hand written report describing the strategies, load characteristics and the performance for the given load characteristics and the reasoning behind the performance. Please describe the problems you have faced in it.
2. freelist.c.mfu and freelist.c.lfu (please change your implemented file to this name)
3. buf_internals.h.mfu and buf_internals.h.lfu

(iv) Name your tar archive as your entry number.

(v) Can do this assignment in group

(vi) Both the candidates should not what has been done in the assignment.

(vii) Some bonus point can be given to an individual if he/she does this assignment alone.

(viii) This assignment will have weightage as ¼ th of the total assignment weightage.

(ix) Copy case if caught would be given –ve marks.

(x) For further query please mail at vkgoyal@cse.iitd.ernet.in.