Project 1 : REPORT

Title : Implementation of **FIFO Buffer Replacement Policy** in PostgreSQL alongside Clock sweep Algorithm

Team Members

Roshni Anand	**Neelam Nand Somai**
USC ID: [redacted]	**USC ID**: [redacted]

Introduction

This report compares the performance of two buffer replacement algorithms, FIFO (First-In, First-Out) and CLOCK SWEEP, using a set of test cases executed on PostgreSQL database. The evaluation focuses on query execution time to determine which algorithm performs better under specific workloads and database querying operations. Two SQL scripts, ***buffertest1.sql*** and ***buffertest2.sql*** and two value files containing ***values10k.dat*** and ***values100k.dat values***, were used to perform similar insertions into the database, with execution time recorded for each test case as indicated in the table below.

Algorithms & their implementation

The two test case files ***buffertest1.sql*** and ***buffertest2.sql*** were executed on PostgreSQL with some configurational changes made in the buffer implementations as follows : **NBuffers = 16** and **shared_buffers = 128kB**.

Two algorithms were tested under identical workloads:

- **FIFO (First-In, First-Out):** This algorithm removes the oldest inserted buffer, determined by the order of insertion, whenever a new buffer is required.
- **CLOCK SWEEP:** Operating on a circular list of buffers, this algorithm searches for a buffer with a usage count of zero, decrementing the usage count of buffers it encounters along the way.

Test Query Execution Time Results

Test Cases	Time taken : FIFO	Time taken : CLOCK SWEEP
Buffer_test1 - **10k**	117.041 ms	168.445 ms
Buffer_test1 - **100k**	236.659 ms	127.805 ms
Buffer_test2 - **10k**	127.827 ms	149.402 ms
Buffer_test2 - **100k**	267.412 ms	112.29 ms

Observations from Test Query Execution Time Results

1. **Performance with Small Datasets (10k):**
    - FIFO outperforms Clock Sweep in both Buffer_test1 and Buffer_test2 for 10k records.

      Possible Reason: The overhead of managing reference bits and the clock-hand mechanism in Clock Sweep adds extra processing time, which becomes significant when the dataset is small.

2. **Performance with Large Datasets (100k):**
    - Clock Sweep outperforms FIFO significantly in both test cases when the dataset size increases to 100k.

      Possible Reason: Clock Sweep's ability to retain frequently accessed pages reduces page faults, improving cache efficiency and execution time. FIFO, on the other hand, replaces pages blindly, leading to more cache misses.

Conclusions:

- FIFO is faster for small datasets due to minimal overhead.

- Clock Sweep performs better with larger datasets- offering better performance as it effectively reduces cache misses.