

## Buffer Replacement Strategy Comparison: FIFO vs. Clock Sweep

This experiment compares the performance of two buffer replacement strategies—FIFO and Clock Sweep—on the PostgreSQL buffer manager, using two test cases and varying buffer limits. The FIFO approach uses a fixed queue size, `BM_BUF_FIFO_QUEUE_SIZE`, initially set to 2048 buffers based on the test cases, which primarily utilized around 2000 buffers. Clock Sweep was tested by manually restricting free buffers to force the algorithm into action.

### Results:

#### Clock-Sweep Algorithm

NBuffers	16384	1024	500	200
buffertest1	0.182903	0.1724054	0.1876018	0.1811462
buffertest2	0.249881	0.2685784	0.2616376	0.2530688

#### FIFO Algorithm

FIFO Queue Size	2048	1024	500	200
buffertest1	0.1713482	0.1723276	0.17531	0.1783236
buffertest2	0.2470458	0.2419072	0.2702182	0.2794256

### Results Overview:

- The FIFO algorithm consistently outperformed the Clock Sweep algorithm across various buffer sizes in both test cases. This suggests that FIFO's straightforward eviction of the first unpinned buffer is efficient under these conditions.
- In Test Case 1, both algorithms generally pull buffers directly from the freelist, as the test only requires around 1998 buffers. However, when forced to go beyond the freelist:
  - The Clock Sweep approach had higher average access times compared to FIFO, which can be attributed to Clock Sweep's combination of clock and LRU, adding overhead in maintaining usage counts.
- In Test Case 2, FIFO's performance remains better across all buffer settings, showing that evicting the oldest unpinned buffer simplifies and speeds up buffer allocation, especially with limited buffers.

### Design Considerations:

- **Queue Size in FIFO:** The FIFO queue is capped at 2048 buffer nodes, appropriate for this test's requirements of around 2000 buffers. Adjusting this size will affect FIFO's performance, as a larger queue allows for more buffers to be cycled before eviction.

- **Clock Sweep's Complexity:** Clock Sweep employs both LRU and clock principles, which improves retention of frequently accessed pages but adds complexity. This overhead may contribute to slightly slower performance than FIFO in these controlled tests.
- **Free List Behavior:** Both algorithms initially use the freelist until exhausted. Forcing buffers beyond this list highlighted Clock Sweep's limitations in performance when handling increased eviction pressures compared to FIFO.

The results suggest that while Clock Sweep's hybrid approach might benefit specific access patterns requiring retention of frequently accessed buffers, FIFO's simplicity provides a clear performance advantage for the tested workloads. Further exploration with different workloads and buffer settings can provide deeper insights into when Clock Sweep's hybrid strategy might excel.