## Chapter 4.7 Buffer Management

Assume that the operator on Relation can be used to get M main memory buffer, and of course they can be used to store the required data. *Provide available main memory for Query on Database, this key task is assigned on Buffer Management.* The task for Buffer Management is to get the required main memory and decrease delay and unsatisfiable requirements.

### Chapter 4.7.1 The Structure of Buffer Management

***Principle:***

There has two main buffer management structures:

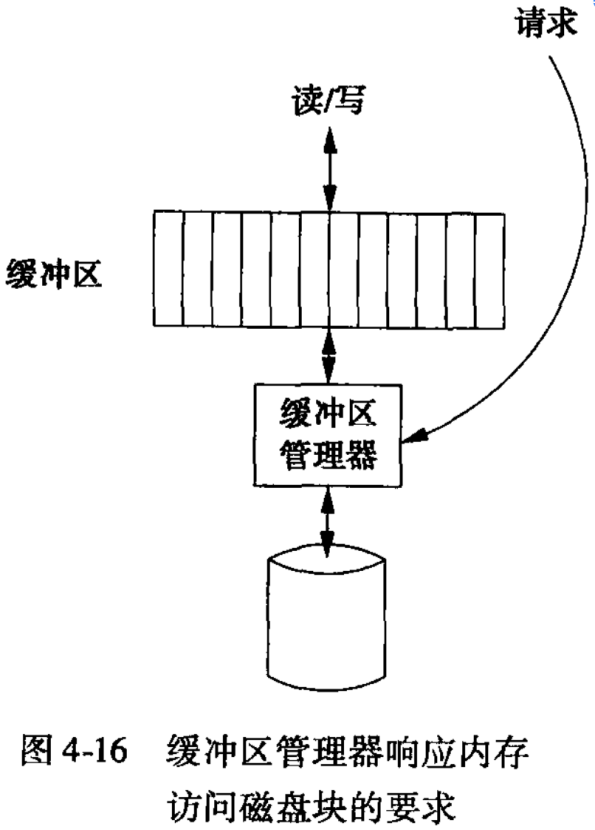
1. In majority Relational DBMS, the buffer management control the main memory directly.
2. Buffer Management is allowed to assign buffers in the virtual memory, it also allowed operation system to put which buffer areas into the main memory and which buffer areas in the ‘Swapping Space’.

***Question:***

No matter for which structure, the Buffer Management should be used to limit the number of buffer areas in order to make them adapt to the memory capacity.

***Solution:***

1. When Buffer Management control main memory directly, but if it exceeds the available space, then we need to return content of buffer into disk to clean buffer areas. If there has no change on the current buffer, then just clean it up; Otherwise if the block has changed, then write it back to the disk.
2. If the number of virtual memory that assigned for the main memory by Buffer Management is much larger than main memory capacity, then when we really use it, then *‘Jolt’* will happens, which means that there have a large bunch of main memory will be move in and move out. Then under this situation, the system will spend a large of time to swap the block other than finish the useful works.



***Conclusion:***

Normally, when initialization DBMS, the number of buffer memory is a parameter setting. In the chapters below, we just assume there has a fixed buffer pool, which is the collection by searching or other Database Operation.

### Chapter 4.7.2 The Strategy of Buffer Management

***Key Point:***

The key point that the Buffer Management do is to choose to throw which block out of Buffer Pool when a new block want to get a buffer area. -> So called *Buffer - Replace Strategy*.

***Least Recently Used (LRU)***

*Rule:*

LRU exchange the blocks that used least recently. This method requires that Buffer Management keeps one table that indicates the last used time of each blocks. It also requires that each time there has one access from database, then there should generate one column.

*Advantage:*

LRU strategy is an effective method, since the least used blocks will get less access than those blocks that have been visited recently.

*Disadvantage:*

LRU needs a large workload to maintain the table information.

***First In First Out (FIFO)***

*Rule:*

In this strategy, when needing a buffer block, then the one which has been occupied for the longest time is cleaned and new block is called and put into the buffer block. In this method, Buffer Management only needs to know when the current block has been filled with. So when read one block from the disk into the buffer block, just generate one column of the table, we should be noted that when the block is being visited, then the column do not need to be modified.

*Advantage:*

Compared with the LRU, FIFO only needs less maintenance.

*Disadvantage:*

FIFO may cause mistake, in the B - Tree index with root block, then it will turn the oldest block in the buffer area. The root block is wrote back to the disk and then read into another buffer block soon.

***Clock Algorithm (The Second Chance)***

***System Control***

### Chapter 4.7.3 The Relationship between Physical Operator Selection and Buffer Management