CS 564: Database Management Systems

Lecture 15: Buffer Management

Xiangyao Yu

2/26/2024

Midterm Exam Logistics

Mid-term exam

- March 20th, Wednesday
- 2:30–3:45pm (in-class)
- Please arrive 5 min early
- Paper-based, closed-book
- Cheat sheet allowed, US letter size (8.5 × 11 inches), double-sided

Previous years' exam questions will be released next week

Final exam is cumulative

- You will be tested on everything you have learned so far
- More focus on the second half of class

Module B1: Basics of DB Internals

Data Storage

File Organization

Buffer Management

Outline

Buffer manager

Buffer replacement policy

- LRU
- Clock
- Other algorithms

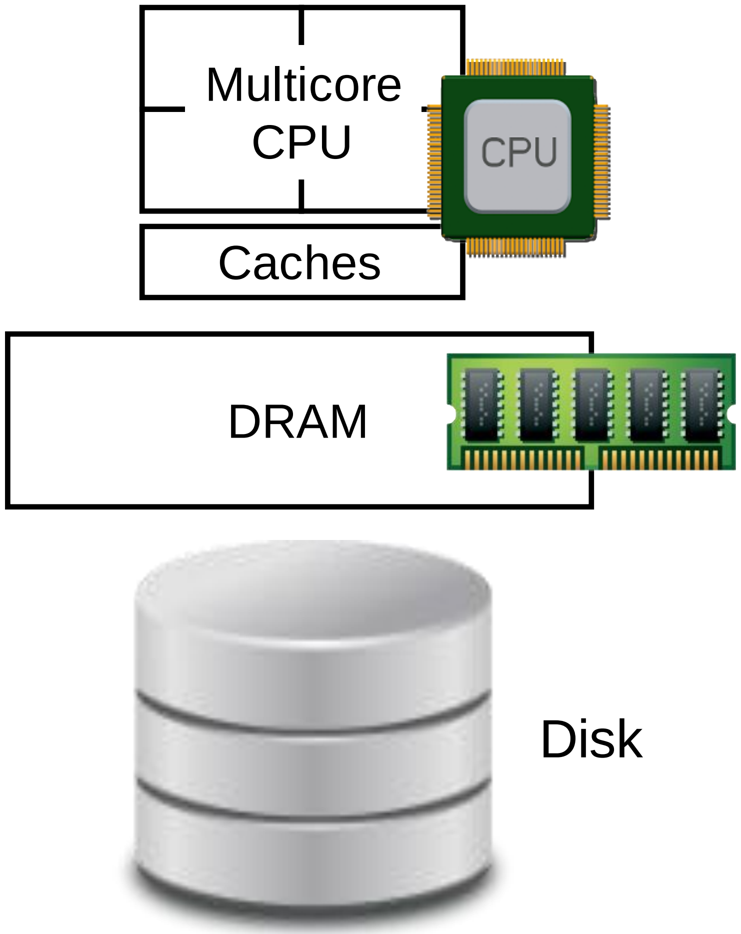
Outline

Buffer manager

Buffer replacement policy

- LRU
- Clock
- Other algorithms

Architecture – Hardware Perspective



How does a DBMS store and access data?

- **Primary storage**: Main memory (DRAM) for currently used data
- **Secondary storage**: Disk for the main database
- **Tertiary storage**: Tapes for archiving older versions of the data

How do we move data from disk to main memory?

- buffer manager

Buffer Manager

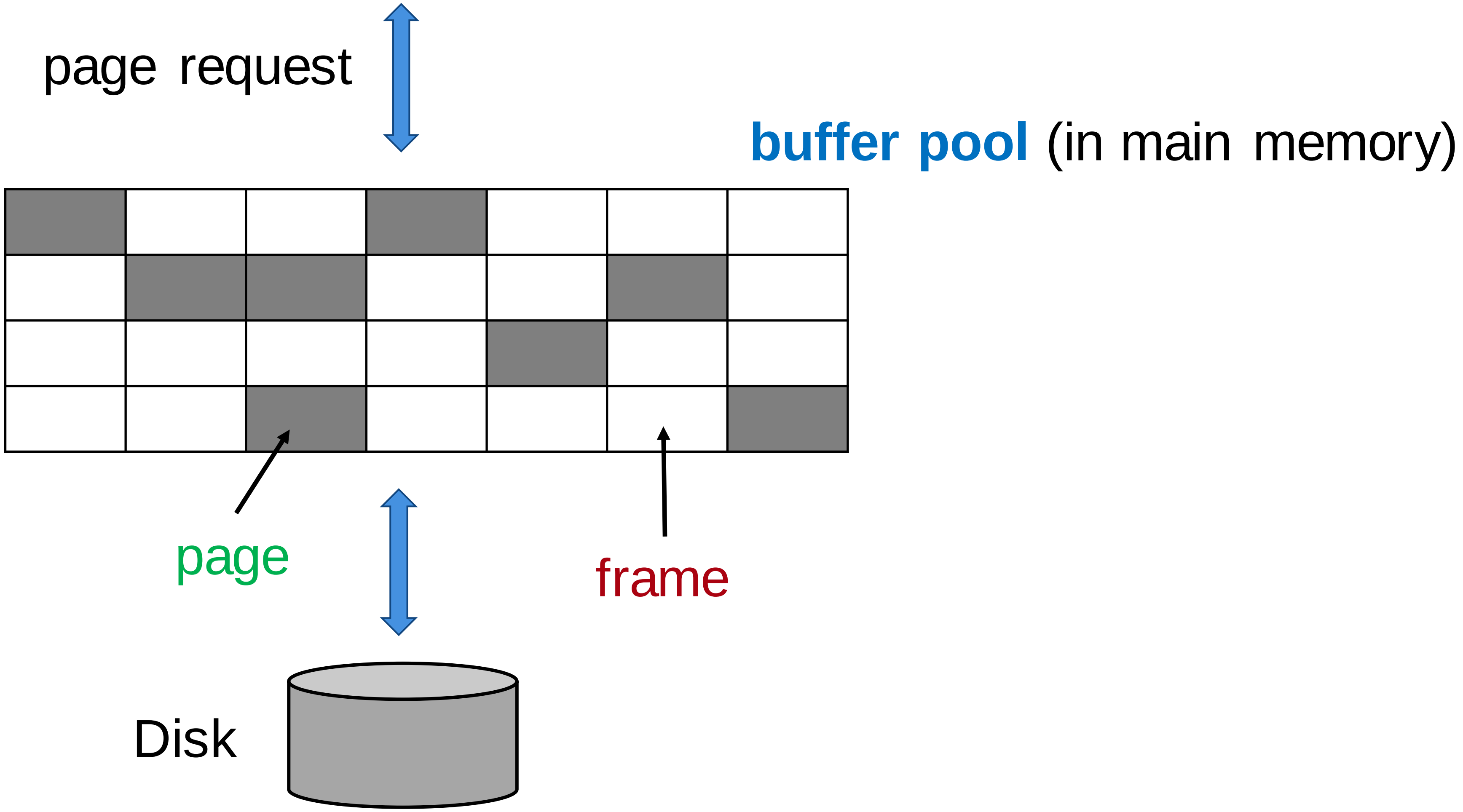
Data must be in RAM for DBMS to operate on it

Data pages may not entirely fit into main memory

Buffer manager: responsible for bringing pages from disk to main memory as needed

- Pages brought into main memory are in the buffer pool
- The buffer pool is partitioned into frames: slots for holding disk pages

Buffer Manager



Buffer Manager – Requests

Read (page): read a page from disk and add to the buffer pool (if not already in buffer). Increment the pin counter

Flush (page): evict page from buffer pool & write to disk if dirty

Release (page): decrement pin counter

Bookkeeping

Bookkeeping per frame:

pin count: # current users of the page

- *pinning* : increment the pin count
- *unpinning* : decrement the pin count

dirty bit: indicates if the page has been modified

- bit = 1 means that the changes to the page must be propagated to the disk

Page Request

Page is in the buffer pool:

- Return the address to the frame
- Increment the pin count

Page is not in the buffer pool:

- If exists empty frame, add the page to it
- Otherwise, choose a frame for replacement (with pin count = 0)
- If frame is dirty, write the page to disk
- Read requested page into chosen frame
- Pin the page and return the address

Buffer Management – Example

	frame	**dirty**	**pincount**
1		0	0
2		0	0
3		0	0

Sequence of requests:

request A, modify A, request B, request B, release A,
request C, release B, request D, modify D, release B,
request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	**A**	0	**1**
2		0	0
3		0	0

one I/O to read the page

Sequence of requests:

request A, modify A, request B, request B, release A, request C, release B, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	A	**1**	1
2		0	0
3		0	0

no I/O here!

Sequence of requests:

request A, **modify A**, request B, request B, release A, request C, release B, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	A	1	1
2	**B**	0	**1**
3		0	0

one I/O to read the page

Sequence of requests:

request A, modify A, **request B**, request B, release A, request C, release B, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	A	1	1
2	B	0	**2**
3		0	0

No I/O here
The pincount increases!

Sequence of requests:

request A, modify A, request B, **request B**, release A, request C, release B, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	A	1	**0**
2	B	0	2
3		0	0

no I/O yet!

Sequence of requests:

request A, modify A, request B, request B, **release A**, request C, release B, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	A	1	0
2	B	0	2
3	**C**	0	**1**

one I/O to read the page

Sequence of requests:

request A, modify A, request B, request B, release A, **request C**, release B, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	A	1	0
2	B	0	**1**
3	C	0	1

the pincount decreases

Sequence of requests:

request A, modify A, request B, request B, release A, request C, **release B**, request D, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	**D**	0	**1**
2	B	0	1
3	C	0	1

two I/Os: one to write A to disk and one to read D

Sequence of requests:

request A, modify A, request B, request B, release A, request C, release B, **request D**, modify D, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	D	**1**	1
2	B	0	1
3	C	0	1

no I/O here

Sequence of requests:

request A, modify A, request B, request B, release A, request C, release B, request D, **modify D**, release B, request A, request E

Buffer Management – Example

	frame	dirty	pincount
1	D	1	1
2	B	0	**0**
3	C	0	1

no I/O

Sequence of requests:

request A, modify A, request B, request B, release A, request C, release B, request D, modify D, **release B**, request A, request E

Buffer Management – Example

	frame	**dirty**	**pincount**
1	D	1	1
2	**A**	0	**1**
3	C	0	1

one I/O to read A

Sequence of requests:

request A, modify A, request B, request B, release A, request C, release B, request D, modify D, release B, **request A**, request E

Buffer Management – Example

	frame	dirty	pincount
1	D	1	1
2	A	0	1
3	C	0	1

The buffer pool is full, the request must wait!

Sequence of requests:

request A, modify A, request B, request B, release A, request C, release B, request D, modify D, release B, request A, **request E**

Buffer Replacement Policy

Choose a victim if multiple pages can be evicted

- LRU (**L**east **R**ecently **U**sed)
- Clock
- MRU (**M**ost **R**ecently **U**sed)
- LFU (**L**east **F**requently **U**sed)
- FIFO, random, …

The replacement policy has big impact on # of I/O's (depends on the access pattern)

Outline

Buffer manager

Buffer replacement policy

- **LRU (least-recently used)**
- Clock
- Other algorithms

Least Recently Used (LRU)

LRU (Least Recently Used)

- Uses a queue of pointers to frames that have pin count = 0
- A page request uses frames only from the head of the queue
- When the pin count of a frame goes to 0, it is added to the end of the queue

LRU – Example

	frame	dirty	pincount
1		0	0
2		0	0
3		0	0

LRU page MRU page

priority queue: 1, 2, 3

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, read E

For simplicity, consider only reads and each read is a request + release (so no need to worry about pincount)

LRU – Example

	frame	dirty	pincount
1	A	0	0
2		0	0
3		0	0

LRU page MRU page

priority queue: 2, 3, 1

Sequence of requests:

read A, read B, read C, read A, read B, read D, read A, read B, read E

Miss

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3		0	0

LRU page MRU page

priority queue: 3, 1, 2,

Sequence of requests:

read A, **read B**, read C, read A, read B, read D,
read A, read B, read E

Miss

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	C	0	0

LRU page MRU page

priority queue: 1, 2, 3

Sequence of requests:

read A, read B, **read C**, read A, read B, read D,
read A, read B, read E

Miss

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	C	0	0

LRU page MRU page

priority queue: 2, 3, 1

Sequence of requests:

read A, read B, read C, **read A**, read B, read D,
read A, read B, read E

Hit

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	C	0	0

LRU page MRU page

priority queue: 3, 1, 2

Sequence of requests:

read A, read B, read C, read A, **read B**, read D,
read A, read B, read E

Hit

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	~~C~~ **D**	0	0

LRU page

MRU page

priority queue: 1, 2, 3

Sequence of requests:

read A, read B, read C, read A, read B, **read D**, read A, read B, read E

Miss

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	D	0	0

LRU page MRU page

priority queue: 2, 3, 1

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, read E

Hit

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	D	0	0

LRU page MRU page

priority queue: 3, 1, 2

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, **read B**, read E

Hit

LRU – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	~~D~~ **E**	0	0

LRU page MRU page

priority queue: 1, 2, 3

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

5 misses out of 9 accesses

Miss ratio = 5/9 = 55.6%, hit ratio = 4/9 = 44.4%

Outline

- Buffer manager
- Buffer replacement policy
  - LRU
  - **Clock**
  - Other algorithms

Clock

Drawbacks of LRU

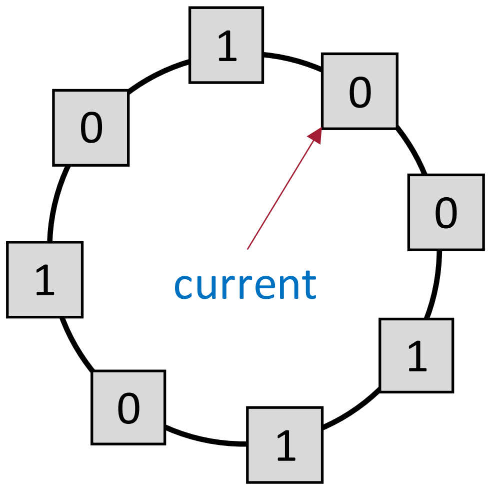
- LRU priority list consumes memory
- Needs to update the list for each page access

Clock algorithm is a variant of LRU with lower memory consumption and computation

Clock (The 2nd Chance Algorithm)

Clock algorithm: data structures

- Each frame has a referenced bit that is set to 1 when pin count becomes 0
- A current variable points to a frame



When a frame is considered for replacement:

- If pin count > 0, increment current
- If referenced = 1, set to 0 and increment current
- If referenced = 0 and pin count = 0, choose the page to replace

Clock – Example

	frame	dirty	pincount
1		0	0
2		0	0
3		0	0

current ↓

0	0	0

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, read E

Clock – Example

	frame	dirty	pincount
1	A	0	0
2		0	0
3		0	0

current

1	0	0

Sequence of requests:

read A, read B, read C, read A, read B, read D, read A, read B, read E

Miss

Clock – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3		0	0

current

1	1	0

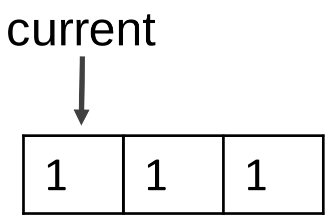
Sequence of requests:

read A, **read B**, read C, read A, read B, read D,
read A, read B, read E

Miss

Clock – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	C	0	0



Sequence of requests:

read A, read B, **read C**, read A, read B, read D, read A, read B, read E

Miss

Clock – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	C	0	0

current

1	1	1

Sequence of requests:

read A, read B, read C, **read A, read B**, read D,
read A, read B, read E

Hit Hit

Buffer hits, set referenced bits for frame1 and frame2 to 1 (they are already 1)

Clock – Example

	frame	**dirty**	**pincount**
1	A	0	0
2	B	0	0
3	C	0	0

current

1	1	1

Sequence of requests:

read A, read B, read C, read A, read B, **read D**, read A, read B, read E

Miss

Clock – Example

	frame	**dirty**	**pincount**
1	A	0	0
2	B	0	0
3	C	0	0

current

0	1	1

Sequence of requests:

read A, read B, read C, read A, read B, **read D**,
read A, read B, read E

Miss

Clock – Example

	frame	dirty	pincount
1	A	0	0
2	B	0	0
3	C	0	0

current

0	0	1

Sequence of requests:

read A, read B, read C, read A, read B, **read D**,
read A, read B, read E

Miss

Clock – Example

	frame	dirty	pincount
1	**~~A~~ D**	0	0
2	B	0	0
3	C	0	0

current

0	0	0

Sequence of requests:

read A, read B, read C, read A, read B, **read D**,
read A, read B, read E

Miss

Load page D into frame 1, replacing page A

Clock – Example

	frame	**dirty**	**pincount**
1	**~~A~~ D**	0	0
2	B	0	0
3	C	0	0

current

1	0	0

Sequence of requests:

read A, read B, read C, read A, read B, **read D**,
read A, read B, read E

Miss

Load page D into frame 1, replacing page A

Set Frame1’s referenced bit to 1, advance current

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	~~B~~ A	0	0
3	C	0	0

current

1	0	0

Sequence of requests:

read A, read B, read C, read A, read B, read D, **read A**, read B, read E

Miss

Replace page B

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	A	0	0
3	C	0	0

current

1	1	0

Sequence of requests:

read A, read B, read C, read A, read B, read D, **read A**, read B, read E

Miss

Replace page B

Set Frame2's referenced bit to 1, advance current

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	A	0	0
3	~~C~~ **B**	0	0

current

1	1	0

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, **read B**, read E

Miss

Replace page C

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	A	0	0
3	B	0	0

current

1	1	1

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, **read B**, read E

Miss

Replace page C

Set Frame3's referenced bit to 1, advance current

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	A	0	0
3	B	0	0

current

1	1	1

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	A	0	0
3	B	0	0

current

0	1	1

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Clock – Example

	frame	dirty	pincount
1	D	0	0
2	A	0	0
3	B	0	0

current

0	0	1

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Clock – Example

	frame	dirty	pincount
1	~~D~~ E	0	0
2	A	0	0
3	B	0	0

current

0	0	0

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Replace page D in frame 1

Clock – Example

	frame	dirty	pincount
1	E	0	0
2	A	0	0
3	B	0	0

current

1	0	0

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Replace page D in frame 1, set Frame1’s reference bit and advance current

Clock – Example

	frame	dirty	pincount
1	E	0	0
2	A	0	0
3	B	0	0

current

1	0	0

Miss ratio?

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Replace page D in frame 1, set Frame1's reference bit and advance current

Clock – Example

	frame	dirty	pincount
1	E	0	0
2	A	0	0
3	B	0	0

current

1	0	0

Miss ratio = 7/9

Sequence of requests:

read A, read B, read C, read A, read B, read D,
read A, read B, **read E**

Miss

Keep resetting reference bits and advancing current

Replace page D in frame 1, set Frame1's reference bit and advance current

Outline

Buffer manager

Buffer replacement policy

- LRU
- Clock
- **Other algorithms**
  - MRU (Most Recently Used)
  - LFU (Least Frequently Used)
  - FIFO
  - Random

MRU (Most-Recently Used)

Consider a buffer pool of size K and the following access pattern

```
while (true)
        access P1
        access P2
        …
        access Pk+1
```

What will happen for LRU replacement?

What will happen for MRU replacement?

MRU (Most-Recently Used)

Consider a buffer pool of size K and the following access pattern

```
while (true)
        access P_1
        access P_2
        …
        access P_k+1
```

What will happen for LRU replacement?

– Sequential flooding: all accesses are misses

What will happen for MRU replacement?

– After the first iteration, only one access per iteration is a miss

Other Replacement Policies

LFU (Least Frequently Used)

- Replace the least frequently used page
- Use a counter to track the number of per-page accesses

FIFO

- Replace the first page in the buffer
- Maintained using a first-in-first-out queue

Random

- Pick a random page for replacement

Hybrid Replacement Policy

Straight sequential pattern (file scan)

- Allocate one page in buffer.
- Replacement: Any replacement.

Looped sequential pattern

- Allocate file size.
- Replacement: LRU or MRU.

Random access pattern

- Allocate hot set size.
- Replacement: LRU or LFU.

Summary

Buffer manager
- Pages
- Dirty bit, pin count

Buffer replacement policy
- LRU
- Clock
- Other algorithms
  - MRU (Most Recently Used)
  - LFU (Least Frequently Used)
  - FIFO
  - Random