Page Replacement

- the *page replacement policy* is critical for performance
  - disks are slow – page faults are expensive
  - swapping out pages that must soon be swapped back in can effectively bring the system to a halt
    - *thrashing* refers to a system that spends too much time handling page faults rather than running processes

> THE CRUX: HOW TO DECIDE WHICH PAGE TO EVICT
> How can the OS decide which page (or pages) to evict from memory? This decision is made by the replacement policy of the system, which usually follows some general principles (discussed below) but also includes certain tweaks to avoid corner-case behaviors.

Cache Management

- main memory holds a subset of all of the pages
  - it can be viewed as a *cache* for the virtual memory pages stored on disk
  - the goal for a page replacement policy is to minimize the number of cache misses – or to maximize the number of cache hits

Metrics

- *average memory access time* (AMAT)

$$AMAT = T_M + (P_{Miss} \cdot T_D)$$

always pay the cost of accessing memory, plus sometimes have to also pay the cost of loading it from disk first

  - T_M = cost (time) of accessing memory
  - T_D = cost (time) of accessing disk
  - P_{Miss} = probability of a cache miss (0-1)

Metrics

1µs = 1000ns
1ms = 1000µs

$$AMAT = T_M + (P_{Miss} \cdot T_D)$$

assume T_M = 100ns = 0.1µs, T_D = 10ms = 10000µs (HDD speed)

90% hit rate → P_{Miss} = 0.1 → AMAT = 100ns + (0.1)(10ms) = 1.0001ms
99.9% hit rate → P_{Miss} = 0.001 → AMAT = 100ns + (0.001)(10ms) = 10.1µs

---

assume T_M = 100ns = 0.1µs, T_D = 100µs (SSD speed)

90% hit rate → P_{Miss} = 0.1 → AMAT = 100ns + (0.1)(100µs) = 10.1µs
99.9% hit rate → P_{Miss} = 0.001 → AMAT = 100ns + (0.001)(100µs) = 200ns

---

the hit rate matters – very high values are needed to avoid running at disk speed

Optimal Replacement Policy

- the optimal replacement policy is the one resulting in the fewest cache misses
  - cache hit rate = (hits)/(hits+misses)
- this is achieved by replacing the page accessed farthest in the future first

process uses four pages: 0, 1, 2, 3
assume cache holds three pages
cache begins empty (*cold start*)

first misses for any page is a *compulsory* (*cold-start*) *miss* because pages aren't swapped in until needed

later misses are *capacity misses* because the cache is full

hit rate = 6/(6+5) = 54.5%
without first miss for each page, hit rate = 6/(6+1) = 85.7%

Access	Hit/Miss?	Evict	Resulting Cache State
0	Miss		0
1	Miss		0, 1
2	Miss		0, 1, 2
0	Hit		0, 1, 2
1	Hit		0, 1, 2
3	Miss	2	0, 1, 3
0	Hit		0, 1, 3
3	Hit		0, 1, 3
1	Hit		0, 1, 3
2	Miss	3	0, 1, 2
1	Hit		0, 1, 2

Optimal Replacement Policy

- optimal policy isn't implementable, but provides a benchmark for evaluating other policies

FIFO

- pages are placed in a queue when swapped in
- first page in the queue is the first evicted
  - oldest first

FIFO hit rate = 4/(4+7) = 36.3%
without compulsory misses, FIFO hit rate = 4/(4+3) = 57.1%

far worse than 54.5% (85.7%) for optimal policy

Access	Hit/Miss?	Evict	Resulting Cache State	
0	Miss		First-in→	0
1	Miss		First-in→	0, 1
2	Miss		First-in→	0, 1, 2
0	Hit		First-in→	0, 1, 2
1	Hit		First-in→	0, 1, 2
3	Miss	0	First-in→	1, 2, 3
0	Miss	1	First-in→	2, 3, 0
3	Hit		First-in→	2, 3, 0
1	Miss	2	First-in→	3, 0, 1
2	Miss	3	First-in→	0, 1, 2
1	Hit		First-in→	0, 1, 2

Belady's Anomaly

- references: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- FIFO replacement
- hit rate with cache size 3?
- hit rate with cache size 4?

→ hit rate is *worse* with a larger cache

- replacement policies that obey the *stack property* avoid this anomaly