CSCI 447 Homework 4

Isaac Boaz

May 29, 2024

Question 1

For an average page fault service of 5 milliseconds, and a memory access time of 300 nanoseconds, what is the maximum access(es) out of 2,000 attempts that can cause a page fault so that the effective access time is *at most* 6.6 microseconds?

$$\begin{aligned}
\frac{5,000,000 \cdot n + 300 \cdot (2,000 - n)}{2,000} &\leq 6,600 \\
5,000,000n + 600,000 - 300n &\leq 13,200,000 \\
4,999,700n &\leq 12,600,000 \\
n &\leq \frac{12,600,000}{4,999,700} \\
n &\leq 2 \text{ acceses causing page faults}
\end{aligned}$$

Question 2

Given the following page requests: `1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6`

Regardless of the eviction algorithm used, if we are only given 1 frame of memory, since there are no duplicate page requests (e.g. 1, 1), the number of page faults is equal to the number of page requests (i.e. **20**).

Similarly, since there are only 7 unique pages being requested, with 7 frames of memory we will always have a page fault on the first unique 7 pages, and no page faults on the remaining requests. Thus, the number of page faults is **7**.

Sadly we need to do more work for the 3 and 5 frame cases.

Strategy	3 Frames	5 Frames
LRU	15	8
FIFO	16	10
Optimal	11	7

L R U

Oldest	1	1	1	2	4	2	1	5	6	1	2	3	6	3	2
		2	2	3	2	1	5	6	2	2	3	7	3	2	3
Newest			3	4	1	5	6	2	1	3	7	6	2	1	6
				3					6			7		3	
				4					1			6		1	
				2					2			3		2	
														1	
														2	
														3	

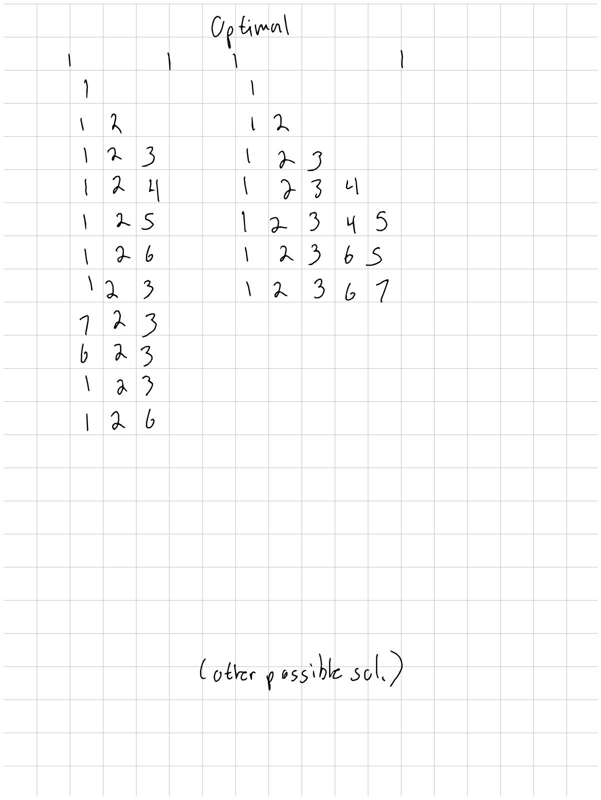
				3	4	5	~~6~~
			1	4	2	6	~~1~~
		1	2	2	1	1	~~2~~
	1	2	3	1	5	2	~~3~~
1	2	3	4	5	6	3	7
			1		4		~~6~~
			3		~~1~~		~~3~~
			4		5		~~2~~
			2		6		1
			3		~~2~~		2
			4		1		3
			2		2		6
			1				

FiFo

1 1 1 1

1		
2	1	
3	2	1
4	3	2
1	4	3
5	1	4
6	5	1
2	6	5
1	2	6
3	1	2
7	3	1
6	7	3
2	6	7
1	2	6
3	1	2
6	3	1

1				
2	1			
3	2	1		
4	3	2	1	
5	4	3	2	1
6	5	4	3	2
1	6	5	4	3
2	1	6	5	4
3	2	1	6	5
7	3	2	1	6



Optimal

1		
1	2	
1	2	3
1	2	4
1	2	5
1	2	6
1	2	3
7	2	3
6	2	3
1	2	3
1	2	6

1				
1	2			
1	2	3		
1	2	3	4	
1	2	3	4	5
1	2	3	6	5
1	2	3	6	7

(other possible sol.)

Question 3

- Faster CPU: Most likely not, as the CPU utilization is already low, and it appears the bottleneck is the paging disk.
- Install a bigger paging disk: This would likely help, as it would increase the number of pages that can be stored in memory, reducing the number of page faults.
- Increase the degree of multiprogramming: This could help, as it would allow the CPU to switch to another process while waiting for either the I/O or the paging disk.
- Decrease the degree of multiprogramming: This could help in a rare case if switching between processes is causing a significant amount of overhead (either due to limited memory or many page faults).
- Install more main memory: This would likely help, as it would reduce the number of page faults.
- Install a faster hard disk or multiple controllers with multiple hard disks: This would likely help, as it would reduce the time spent waiting for the paging disk.

- Increase the page size: This could help, since the I/O device is only at 5% utilization, it is likely that the bottleneck is the paging disk.