

CSC369: Assignment 3

Yuvrender Gill and Gaojin

Algorithm Comparison Tables:

I. Trace File: Simpleloop

Memory Size = 50	FIFO	CLOCK	LRU	RAND
Hit Rate	22.4822	25.1481	25.2073	22.4822
Hit Count	759	849	851	759
Miss Count	2617	2527	2525	2617
Overall eviction count	2567	2477	2475	2567
Clean eviction count	60	15	14	60
Dirty eviction count	2507	2462	2461	2507

Memory Size = 100	FIFO	CLOCK	LRU	RAND
Hit Rate	23.7855	25.1481	25.2073	23.8744
Hit Count	803	849	851	806
Miss Count	2573	2527	2525	2570
Overall eviction count	2473	2427	2425	2470
Clean eviction count	38	15	14	31
Dirty eviction count	2435	2412	2411	2439

II. Trace File: Matmul

Memory Size = 50	FIFO	CLOCK	LRU	RAND
Hit Rate	62.9391	65.7665	65.7665	67.2663
Hit Count	1913768	1999739	1999739	2045342
Miss Count	1126896	1040925	1040925	995322

Overall eviction count	1126846	1040875	1040875	995272
Clean eviction count	1104475	1039933	1039933	974769
Dirty eviction count	22371	942	942	20503

Memory Size = 100	FIFO	CLOCK	LRU	RAND
Hit Rate	64.3709	67.0594	66.9061	89.3805
Hit Count	1957302	2039051	2034389	2717762
Miss Count	1083362	1001613	1006275	322902
Overall eviction count	1083262	1001513	1006175	3044828
Clean eviction count	1071730	1000572	1005234	3040664
Dirty eviction count	11532	941	941	4164

III. Trace File: Blocked

Memory Size = 50	FIFO	CLOCK	LRU	RAND
Hit Rate	99.8245	99.8473	99.8618	99.7716
Hit Count	3507538	3508339	3508848	3505678
Miss Count	6166	5365	4856	8026
Overall eviction count	6116	5315	4806	7976
Clean eviction count	4142	3061	2623	5796
Dirty eviction count	1974	2254	2183	2180

Memory Size = 100	FIFO	CLOCK	LRU	RAND
Hit Rate	99.8817	99.8866	99.8971	99.8563
Hit Count	3509549	3509721	3510089	3508654
Miss Count	4155	3983	3615	5050

Overall eviction count	4055	3883	3515	4950
Clean eviction count	2759	2590	2587	3409
Dirty eviction count	1296	1293	928	1541

Discussion:

FIFO:

The **hit rate** of FIFO is lower than CLOCK and LRU for both 50 and 100 memory sizes on all three types of trace files. The above tables show that when the memory size is increased from 50 to 100 the hit rate for FIFO increases for all three cases. The **overall eviction count** for FIFO is the highest as well when compared to CLOCK and LRU with both clean eviction count highest in all cases and dirty eviction count higher in most cases. The overall space and time complexity of the algorithm is constant whereas that's not the case for both LRU and CLOCK.

CLOCK:

Clock always outperforms FIFO in terms of hit rate. Clock's hit rate is slightly lower than that of LRU's. In case of matmul and memory size 100, it outperforms LRU but by a very small margin. This could only be possible by chance that the LRU evicts some page and CLOCK does not so we get a higher hit rate for CLOCK. Overall we can see that the data supports that the LRU is better than CLOCK for most of the cases. Overall eviction count for both of the CLOCK and LRU algorithms is pretty much the same. The CLOCK algorithm behaves very similar to LRU overall, and it supports our intuition that the CLOCK is an approximation of LRU and the data supports that our implementation of CLOCK is a good approximation of LRU.

LRU:

Over all LRU beats both CLOCK and FIFO in terms of hit rate. The overall hit rate increases with increase in memory from 50 to 100 for most of the cases. It is because we have more pages in memory and there only the least recently used one is evicted thus there is a higher chance for hits and hence the higher hit rate. Finally the overall eviction count decreases or remains the same when the memory is increased. This data makes sense as we know that the larger the memory the lower the need for the evictions will be.

Custom Trace-file Analysis:

Trace 1	FIFO	CLOCK	LRU
Hit Count	11	12	13
Miss Count	25	24	23
Hit Rate	30.5556	33.3333	36.1111
Miss Rate	69.4444	66.6667	63.8889

Trace 2	FIFO	CLOCK	LRU
Hit Count	38	38	38
Miss Count	2	2	2
Hit Rate	95.0000	95.0000	95.0000
Miss Rate	5.0000	5.0000	5.0000

Trace 3	FIFO	CLOCK	LRU
Hit Count	0	0	0
Miss Count	33	33	33
Hit Rate	0.0000	0.0000	0.0000
Miss Rate	100.0000	100.0000	100.0000
