CSCC69 A2

Tables

RAND

Program	Memory Size	Hit rate	Hit count	Miss count	Clean eviction count	Dirty eviction count	Overall eviction count
simpleloop	50	70.8756	7269	2987	345	2592	2937
simpleloop	100	73.1377	7501	2755	161	2494	2655
simpleloop	150	73.5862	7547	2709	134	2425	2559
simpleloop	200	73.6252	7551	2705	130	2375	2505
matmul	50	67.0289	1124452	553111	530929	22132	553061
matmul	100	89.2400	1497058	180505	176026	4379	180405
matmul	150	96.7910	1623730	53833	52148	1535	53683
matmul	200	98.1005	1645697	31866	30545	1121	31666
blocked	50	99.6539	2091066	7263	5100	2113	7213
blocked	100	99.7832	2093780	4549	2987	1462	4449
blocked	150	99.8164	2094476	3853	2425	1278	3703
blocked	200	99.8393	2094957	3372	2037	1135	3172
fourDimArr	50	99.9685	466865	147	67	30	97
fourDimArr	100	99.9803	466920	92	0	0	0
fourDimArr	150	99.9803	466920	92	0	0	0
fourDimArr	200	99.9803	466920	92	0	0	0

FIFO

Program	Memory Size	Hit rate	Hit count	Miss count	Clean eviction count	Dirty eviction count	Overall eviction count
simpleloop	50	71.0998	7292	2964	318	2596	2914
simpleloop	100	73.1572	7503	2753	158	2495	2653
simpleloop	150	73.5569	7544	2712	129	2433	2562
simpleloop	200	73.6349	7552	2704	125	2379	2504
matmul	50	62.6644	1051235	626328	601609	24669	626278
matmul	100	64.1172	1075607	601956	589287	12569	601856
matmul	150	98.8390	1658086	19477	18370	957	19327
matmul	200	98.8572	1658392	19171	18050	921	18971
blocked	50	99.7330	2092727	5602	3655	1897	5552
blocked	100	99.8200	2094553	3776	2427	1249	3676
blocked	150	99.8256	2094670	3659	2314	1195	3509
blocked	200	99.8671	2095540	2789	1657	932	2589
fourDimArr	50	99.9743	466892	120	48	22	70
fourDimArr	100	99.9803	466920	92	0	0	0
fourDimArr	150	99.9803	466920	92	0	0	0
fourDimArr	200	99.9803	466920	92	0	0	0

CLOCK

Program	Memory Size	Hit rate	Hit count	Miss count	Clean eviction count	Dirty eviction count	Overall eviction count
simpleloop	50	72.7867	7465	2791	213	2528	2741
simpleloop	100	73.8397	7573	2683	117	2466	2583
simpleloop	150	73.8885	7578	2678	112	2416	2528
simpleloop	200	73.8885	7578	2678	112	2366	2478
matmul	50	65.6980	1102126	575437	574548	839	575387
matmul	100	88.0240	1476658	200905	199982	823	200805
matmul	150	98.8816	1658801	18762	17790	822	18612
matmul	200	98.9069	1659226	18337	17316	821	18137
blocked	50	99.7849	2093815	4514	2536	1928	4464
blocked	100	99.8327	2094819	3510	2307	1103	3410
blocked	150	99.8478	2095135	3194	2134	910	3044
blocked	200	99.8637	2095469	2860	1770	890	2660
fourDimArr	50	99.9769	466904	108	48	10	58
fourDimArr	100	99.9803	466920	92	0	0	0
fourDimArr	150	99.9803	466920	92	0	0	0
fourDimArr	200	99.9803	466920	92	0	0	0

LRU

Program	Memory Size	Hit rate	Hit count	Miss count	Clean eviction count	Dirty eviction count	Overall eviction count
simpleloop	50	72.9037	7477	2779	202	2527	2729
simpleloop	100	73.8787	7577	2679	113	2466	2579
simpleloop	150	73.8885	7578	2678	112	2416	2528
simpleloop	200	73.8885	7578	2678	112	2366	2478
matmul	50	65.5174	1099096	578467	577582	835	578417
matmul	100	66.6718	1118462	559101	558179	822	559001
matmul	150	98.8908	1658955	18608	17637	821	18458
matmul	200	98.8915	1658967	18596	17575	821	18396
blocked	50	99.7848	2093814	4515	2482	1983	4465
blocked	100	99.8430	2095035	3294	2292	902	3194
blocked	150	99.8436	2095048	3281	2229	902	3131
blocked	200	99.8465	2095109	3220	2135	885	3020
fourDimArr	50	99.9790	466914	98	40	8	48
fourDimArr	100	99.9803	466920	92	0	0	0
fourDimArr	150	99.9803	466920	92	0	0	0
fourDimArr	200	99.9803	466920	92	0	0	0

OPT

Program	Memory Size	Hit rate	Hit count	Miss count	Clean eviction count	Dirty eviction count	Overall eviction count
simpleloop	50	73.9177	7581	2675	114	2511	2625
simpleloop	100	74.2200	7612	2644	44	2500	2544
simpleloop	150	74.2882	7619	2637	2	2485	2487
simpleloop	200	74.2882	7619	2637	2	2435	2437
matmul	50	80.4410	1349448	328115	327232	833	328057
matmul	100	96.9065	1625668	51895	50970	825	51795
matmul	150	99.0970	1662414	15149	14174	825	14999
matmul	200	99.3424	1666532	11031	10006	825	11030
blocked	50	99.8211	2094575	3754	2680	1024	3683
blocked	100	99.8536	2095258	3071	1956	1015	2955
blocked	150	99.8719	2095642	2687	1534	1003	2537
blocked	200	99.8844	2095903	2426	1227	999	2226
fourDimArr	50	99.9799	466918	94	30	14	44
fourDimArr	100	99.9803	466920	92	0	0	0
fourDimArr	150	99.9803	466920	92	0	0	0
fourDimArr	200	99.9803	466920	92	0	0	0

Comparison between different algorithms

The hit rates for FIFO in our tables is the lowest among all algorithms as expected since FIFO takes into account only the order in which pages enter when choosing which victim to evict. Also, it is interesting to note that in our results, FIFO is not suffering from Belady's Anomaly (increasing in memory sizes causes increasing page faults) Instead, the number of page faults in our results does not increase when memory sizes increases. Moreover, FIFO serves as a lower bound for hit rates for Clock and OPT, and these results make sense since that's how we expect FIFO to behave.

The Clock (Second Chance) page replacement algorithm improves from FIFO algorithm. Before evicting a page, the algorithm checks if the page was referenced recently. If so, then the page will not be evicted. We expect this modification to result in a same or higher hit rate than FIFO. In our results, the hit rates of clock are at least as high as the hit rates of FIFO in call cases as desired.

For the LRU algorithm, it evicts the least recent entry, which is a better strategy than FIFO (evicts the oldest page). Therefore, we expect that LRU usually has higher hit rates than FIFO, but there could be exceptions. We also expect LRU hit rates to be very similar to Clock since they both estimate the future references well. In our results, the majority of LRU has higher hit rates than FIFO to our expectation. There is one exception that FIFO does better (For the ‘Blocked’ program with 200 as the memory size). Also, it is generally close to the results of CLOCK (with the exception of matmul trace file, memory size = 100).

For the optimal algorithm, we see the future to evict the page that is (1) never referenced in the future or (2) referenced the farthest into the future (if all pages are referenced in the future) so we expect that this gives the highest hit rates. In our results, for all cases, the optimal algorithm gives the highest hit rate as expected. There are cases of ties, but the hit rates of OPT is never strictly lower than the hit rates of any other algorithm.

In all trace programs, using any algorithm, the hit rate increases as memory size increases, and the number of evictions are decreased . This is a good result to have because that means we are not suffering from Belady’s Anomaly.

LRU Description

As the memory size increases, LRU's hit rate increases while its total eviction count decreases.

LRU evicts the least recent entry. LRU is generally better than FIFO since the least recent entries are less likely to appear than the oldest entries. However, this is not guaranteed, so sometimes FIFO performs better.

LRU and CLOCK performs similarly. Except for matmul with memory size = 100 where clock is significantly better, but most of the results are very similar. The idea of both of them is trying to estimate the page referenced as late as possible, and both of these algorithms do a good estimate.

LRU has a lower hit rate than OPT for every case. The least recent page is not necessarily the one with furthest distance, so the only way for LRU to catch up to OPT is if every time, the least recent page is never referenced in the future or the one referenced furthest into the future.

For a large trace, as memory size increases, LRU's performance improves. When we ran the matmul trace, we noticed that memory size 50 is the slowest, memory size 100 is slightly faster, memory size 150 is even faster, and memory size 200 is the fastest. For the other algorithms, we didn't notice any significant changes in the performance. They all seem to finish in about the same time, only for LRU we can clearly observe that the performance is getting faster as memory increases.