

README

Author: Yuhan Shao, Huan Wang, Muyuan Cao.

a) Tables

1. Trace File: simpleloop.ref; The number of unique virtual pages in the trace file: 2642.

command: ./sim -f traceprogs/tr-simpleloop.ref -m memsize -s 2642 -a [algorithm]

memsize = 50

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	70.6934	7239	3001	2951	247	2704
FIFO	71.0352	7274	2966	2916	202	2714
CLOCK	72.6367	7438	2802	2752	104	2648
LRU	72.8223	7457	2783	2733	88	2645
OPT	73.9453	7572	2668	2618	18	2600

memsize = 100

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	73.0176	7477	2763	2663	51	2612
FIFO	73.0859	7484	2756	2656	44	2612
CLOCK	73.7500	7552	2688	2588	4	2584
LRU	73.7891	7556	2684	2584	2	2582
OPT	74.1992	7598	2642	2542	0	2542

memsize = 150

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	73.4961	7526	2714	2564	19	2545
FIFO	73.4766	7524	2716	2566	16	2550
CLOCK	73.7988	7557	2683	2533	0	2533
LRU	73.8086	7558	2682	2532	0	2532
OPT	74.1992	7598	2642	2492	0	2492

memsize = 200

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	73.5352	7530	2710	2510	17	2493
FIFO	73.5547	7532	2708	2508	12	2496
CLOCK	73.7988	7557	2683	2483	0	2483
LRU	73.8086	7558	2682	2482	0	2482
OPT	74.1992	7598	2642	2442	0	2442

2. Trace File: matmul.ref; The number of unique virtual pages in the trace file: 1095.

command: ./sim -f traceprogs/tr-matmul.ref -m memsize -s 1095 -a [algorithm]

memsize = 50

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	65.5156	1892034	995878	995828	956353	39475
FIFO	60.9661	1760646	1127266	1127216	1083225	43991
CLOCK	63.9442	1846651	1041261	1041211	1040096	1115
LRU	63.9453	1846683	1041229	1041179	1040068	1111
OPT	79.6581	2300456	587456	587406	586318	1088

memsize = 100

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	88.7728	2563681	324231	324131	316686	7445
FIFO	62.4798	1804361	1083551	1083451	1061224	22227
CLOCK	63.9527	1846898	1041014	1040914	1039832	1082
LRU	65.1494	1881457	1006455	1006355	1005276	1079
OPT	96.7867	2795115	92797	92697	91611	1086

memsize = 150

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	96.6606	2791474	96438	96288	93941	2347
FIFO	98.8085	2853503	34409	34259	32944	1315
CLOCK	98.8501	2854705	33207	33057	31975	1082
LRU	98.8613	2855026	32886	32736	31657	1079
OPT	99.0784	2861298	26614	26464	25378	1086

memsize = 200

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	98.0426	2831383	56529	56329	54689	1640
FIFO	98.8266	2854024	33888	33688	32434	1254
CLOCK	98.8607	2855011	32901	32701	31622	1079
LRU	98.8616	2855037	32875	32675	31596	1079
OPT	99.3329	2868648	19264	19064	17978	1086

3. Trace File: blocked.ref; The number of unique virtual pages in the trace file: 1097.

command: ./sim -f traceprogs/tr-blocked.ref -m memsize -s 1097 -a [algorithm]

memsize = 50

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	99.6479	2409621	8515	8465	5910	2555
FIFO	99.7320	2411656	6480	6430	4171	2259
CLOCK	99.7824	2412873	5263	5213	2868	2345
LRU	99.7842	2412918	5218	5168	2814	2354
OPT	99.8466	2414427	3709	3659	2572	1087

memsize = 100

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	99.7802	2412821	5315	5215	3460	1755
FIFO	99.8206	2413799	4337	4237	2758	1479
CLOCK	99.8330	2414098	4038	3938	2623	1315
LRU	99.8434	2414350	3786	3686	2606	1080
OPT	99.8755	2415126	3010	2910	1834	1076

memsize = 150

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	99.8209	7522413804	4332	4182	2726	1456
FIFO	99.8252	2413910	4226	4076	2653	1423
CLOCK	99.8369	2414193	3943	3793	2575	1218
LRU	99.8441	2414367	3769	3619	2559	1060
OPT	99.8955	2415608	2528	2378	1300	1078

memsize = 200

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	99.8426	2414329	3807	3607	2287	1320
FIFO	99.8687	2414960	3176	2976	1878	1098
CLOCK	99.8681	2414946	3190	2990	1928	1062
LRU	99.8472	2414440	3696	3496	2436	1060
OPT	99.9058	2415858	2278	2078	1010	1068

4. Trace File:fibonacci.ref; The number of unique virtual pages in the trace file: 143.

command: ./sim -f traceprogs/tr-fibonacci.ref -m memsize -s 143 -a [algorithm]

memsize = 50

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	94.6217	6404	364	314	150	164
FIFO	94.9320	6425	343	293	126	167
CLOCK	95.9959	6497	271	221	84	137
LRU	96.2914	6517	251	201	68	133
OPT	97.5325	6601	167	117	19	98

memsize = 100

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	97.4291	6594	174	74	4	70
FIFO	97.4586	6596	172	72	0	72
CLOCK	97.7541	6616	152	52	0	52
LRU	97.8132	6620	148	48	0	48
OPT	97.8871	6625	143	43	0	43

memsize = 150

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	97.8871	6625	143	0	0	0
FIFO	97.8871	6625	143	0	0	0
CLOCK	97.8871	6625	143	0	0	0
LRU	97.8871	6625	143	0	0	0
OPT	97.8871	6625	143	0	0	0

memsize = 200

Algo	HitRate	HitCount	MissCount	OverallEvictionCount	CleanEvictionCount	DirtyEvictionCount
RAND	97.8871	6625	143	0	0	0
FIFO	97.8871	6625	143	0	0	0
CLOCK	97.8871	6625	143	0	0	0
LRU	97.8871	6625	143	0	0	0
OPT	97.8871	6625	143	0	0	0

b)

In general $\text{rand} < \text{fifo} < \text{clock} \leq \text{lru} < \text{opt}$, and algorithms perform better with larger memory size. The output of the clock and lru are very similar. Most of the time, lru is slightly better than clock. Since clock evicts the page that is "old enough" while lru evicts the page that has not been used for the longest time in the past. In the tr-matmul.ref: with smaller memory size [memsize = 50 & memsize = 100] rand could perform much better than fifo, and even better than clock and lru. The Opt always performs the best among all five algorithms.

The Forth Program Description: (fibonacci): Using while loop to calculate the fibonacci sequence, and returning the 10000th number in the fibonacci sequence. This algorithm is interesting since the number of unique virtual pages in the trace file is 143. Compare to the first three program:

1. simpleloop: the number of unique virtual pages in the trace file: 2642
2. matmul: the number of unique virtual pages in the trace file: 1095
3. blocked: the number of unique virtual pages in the trace file: 1097

The new program has relatively smaller number of unique virtual pages in the trace file, therefore, it can reflect the performance of each eviction algorithm in the situation where the number of unique virtual pages is small.

c)

Consider the HitRate for each trace file in increasing memory size:

memsize	50	100	150	200
simpleloop	72.8223	73.7891	73.8086	73.8086
matmul	63.9453	65.1494	98.8613	98.8616
blocked	99.7842	99.8434	99.8441	99.8472
fibonacci	96.2914	97.8132	97.8871	97.8871

For lru, as the memory size increasing, the algorithm performs better. As memory size increases, more pages could be stored in memory. Therefore, there are less evictions needed. In other words, the page which is evicted by the larger memory size lru has longer time period that it has not been referenced than the same page which is evicted by the smaller memory size. Thus, if the pages are

referenced periodically, larger memory size will lead to less evictions and lower miss rate since larger memory will keep more pages be held in the memory. In contrast, smaller memory size might lead to more page evictions and bringing backs since there is less space for the pages to be held in the memory.