README

Nimit Sachdeva

July 2018

Tables

Algorithm FIFO

tr-simpleloop.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	70.8855	73.0551	73.4461	73.5242
Hit count	7253	7475	7515	7523
Miss count	2979	2757	2717	2709
Overall evictions count	2929	2657	2565	2509
Clean evictions	128	32	8	6
Dirty evictions	2801	2625	2559	2503

tr-matmul.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	60.9658	62.4796	98.8085	98.8265
Hit count	1760634	1804352	2853494	2854015
Miss count	1127270	1083552	34410	33889
Overall evictions count	1127220	1083452	34260	33689
Clean evictions	541683	530671	16665	16250
Dirty evictions	585537	552781	17595	17439

tr-blocked.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	99.7318	99.8206	99.8252	99.8686
Hit count	2411643	2413790	2413901	2414951
Miss count	6485	4338	4227	3177
Overall evictions count	6435	4238	4077	2977
Clean evictions	2117	1398	1367	1003
Dirty evictions	4318	2840	2710	1974

tr-simpleloop.ref from stackloop				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	94.4162	97.0958	97.4102	97.5299
Hit count	6307	6486	6507	6515
Miss count	373	194	173	165
Overall evictions count	323	94	23	0
Clean evictions	98	0	0	0
Dirty evictions	225	94	23	0

Algorithm Clock

tr-simpleloop.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	72.6642	73.7002	73.7588	73.7686
Hit count	7435	7541	7547	7548
Miss count	2797	2691	2685	2684
Overall evictions count	2747	2591	2535	2484
Clean evictions	70	5	0	0
Dirty evictions	2677	2586	2535	2484

tr-matmul.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	63.9438	63.9525	98.8500	98.8607
Hit count	1846637	1846887	2854694	2855001
Miss count	1041267	1041017	33210	32903
Overall evictions count	1041217	1040917	33060	32703
Clean evictions	520117	519983	16130	15988
Dirty evictions	521100	520934	16930	16715

tr-blocked.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	99.7819	99.8329	99.8369	99.8681
Hit count	2412855	2414088	2414184	2414938
Miss count	5273	4040	3944	3190
Overall evictions count	5223	3940	3794	2990
Clean evictions	1466	1337	1327	1062
Dirty evictions	3757	2603	2467	1928

tr-simpleloop.ref from stackloop				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	95.5090	97.3353	97.4251	97.5299
Hit count	6380	6502	6508	6515
Miss count	300	178	172	165
Overall evictions count	250	78	22	0
Clean evictions	68	0	0	0
Dirty evictions	182	78	22	0

Exact LRU

tr-simpleloop.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	72.7912	73.7588	73.7783	73.7783
Hit count	7448	7547	7549	7549
Miss count	2784	2685	2683	2683
Overall evictions count	2734	2585	2533	2483
Clean evictions	65	2	0	0
Dirty evictions	2669	2583	2533	2483

tr-matmul.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	63.9451	65.1493	98.8612	98.8616
Hit count	1846674	1881448	2855017	2855028
Miss count	1041230	1006456	32887	32876
Overall evictions count	1041180	1006356	32737	32676
Clean evictions	520102	502791	16018	15985
Dirty evictions	521078	503565	16719	16691

tr-blocked.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	99.7842	99.8434	99.8441	99.8471
Hit count	2412909	2414341	2414358	2414431
Miss count	5219	3787	3770	3697
Overall evictions count	5169	3687	3620	3497
Clean evictions	1436	1328	1319	1281
Dirty evictions	3733	2359	2301	2216

tr-simpleloop.ref from stackloop				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	95.8084	97.3802	97.5150	97.5299
Hit count	6400	6505	6514	6515
Miss count	280	175	166	165
Overall evictions count	230	75	16	0
Clean evictions	65	0	0	0
Dirty evictions	165	75	16	0

RAND

tr-simpleloop.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	70.9050	72.8206	73.4949	73.4949
Hit count	7255	7451	7520	7520
Miss count	2977	2781	2712	2712
Overall evictions count	2927	2681	2562	2512
Clean evictions	137	37	11	11
Dirty evictions	2790	2644	2551	2501

tr-matmul.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	65.5208	88.8183	96.6502	98.0466
Hit count	1892179	2564987	2791164	2831493
Miss count	995725	322917	96740	56411
Overall evictions count	995675	322817	96590	56211
Clean evictions	478289	157860	47274	27479
Dirty evictions	517386	164957	49316	28732

tr-blocked.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	99.6530	99.7785	99.8168	99.8405
Hit count	2409737	2412772	2413699	2414272
Miss count	8391	5356	4429	3856
Overall evictions count	8341	5256	4279	3656
Clean evictions	3078	1870	1537	1335
Dirty evictions	5263	3386	2742	2321

tr-simpleloop.ref from stackloop				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	94.4461	96.9012	97.5000	97.5299
Hit count	6309	6473	6513	6515
Miss count	371	207	167	165
Overall evictions count	321	107	17	0
Clean evictions	94	10	0	0
Dirty evictions	227	97	17	0

OPT

tr-simpleloop.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	73.9152	74.1693	74.1693	74.1693
Hit count	7563	7589	7589	7589
Miss count	2669	2643	2643	2643
Overall evictions count	2619	2543	2493	2443
Clean evictions	18	0	0	0
Dirty evictions	2601	2543	2493	2443

tr-matmul.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	79.6580	96.7867	99.0784	99.3329
Hit count	2300447	2795106	2861289	2868639
Miss count	587457	92798	26615	19265
Overall evictions count	587407	92698	26465	19065
Clean evictions	293414	46008	12924	9238
Dirty evictions	293993	46690	13541	9827

tr-blocked.ref				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	99.8466	99.8755	99.8954	99.9058
Hit count	2414418	2415117	2415599	2415849
Miss count	3710	3011	2529	2279
Overall evictions count	3660	2911	2379	2079
Clean evictions	1321	1043	830	660
Dirty evictions	2339	1868	1549	1419

tr-simpleloop.ref from stackloop				
Statistics	memsize: 50	memsize: 100	memsize: 150	memsize: 200
Hit rate	97.1856	97.5299	97.5299	97.5299
Hit count	6492	6515	6515	6515
Miss count	188	165	165	165
Overall evictions count	138	65	15	0
Clean evictions	17	0	0	0
Dirty evictions	121	65	15	0

Note: For an interesting program, I choose to comment out the call to heaploop function in the simpleloop.c file and instead run the stackloop in that file to produce a new tr-simpleloop.ref file. It is interesting because there is no page eviction in any of the algorithm when the memory size is 200. Also, each algorithm behaves like OPT when the memory size is 200 for this tracefile.

Comparison Paragraph

I noticed that the for every algorithm and tracefile, the hit rate and hit count increases with the increase in memory. On the other hand, miss count, overall evictions count, clean evictions and dirty evictions decrease with the increase in the memory. One thing was also interesting was that every algorithm, except RAND, got its clean eviction count eventually to 0 with increase in memory for the original file tr-simpleloop.ref(generated from heaploop function).
In general, it can also be observed that the results for each algorithm as memory size increases becomes similar no matter how we access the memory.
One thing that was also evident was that the hit rate for tr-blocked.ref for every algorithm became almost 100 percent as there were a lot of memory references (2418128 in total). Furthermore, this is due to larger traces having more memory references.

LRU as the size of memory increases

With the increase, the performance of LRU improved. This phenomenon was evident as Hit rate and hit count increased with increase in memory size (hit count and hit rate were directly proportional to memory size). Subsequently, all of the eviction count decreased with respect to the increase in the memory(they are inversely proportional). Also, there was an anomaly if we compare the hit rate of RAND and LRU on the trace file tr-matmul.ref when the memory size was 50 and 100. Here, we can observe that RAND outperformed slighly when

memory size was 50 and significantly when the memory size was 100. This could be due to a fact that instructions could be laid out in a very random manner, and hence RAND had better performance. Otherwise, compared to other algorithms except OPT, LRU gave better performance.
Also, as noticed before LRU gave poor performance on the very large trace called tr-matmul.ref when the memory size but it performed exceptionally when the memory size was 150 and 200. Hence, the performace of LRU improved with increase in memory.