Program: Blocked

Algorithm: LRU

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	96.2269	7498	294	264	94	150
100	98.0621	7641	151	51	0	51
150	98.1520	7648	144	0	0	0
200	98.1520	7648	144	0	0	0

Algorithm: FIFO

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	95.0975	7410	382	332	149	183
100	97.7156	7614	178	78	0	78
150	98.1520	7648	144	0	0	0
200	98.1520	7648	144	0	0	0

Algorithm: Random

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	94.7382	7382	410	360	189	171
100	97.4589	7594	198	98	11	87
150	98.1520	7648	144	0	0	0
200	98.1520	7648	144	0	0	0

Algorithm: Optimal

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	97.5744	7603	189	139	30	109
100	98.1520	7648	144	44	0	44
150	98.1520	7648	144	0	0	0
200	98.1520	7648	144	0	0	0

Algorithm: Clock

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	95.8932	7472	320	270	114	156
100	97.9979	7636	156	56	0	56
150	98.1520	7648	144	0	0	0
200	98.1520	7648	144	0	0	0

Program: Matmul

Algorithm: LRU

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	96.2905	7372	284	234	87	147
100	98.0408	7506	150	50	0	50
150	98.1322	7513	143	0	0	0
200	98.1322	7513	143	0	0	0

Algorithm: FIFO

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	95.1280	7283	373	323	143	180
100	97.6881	7479	177	77	0	77
150	98.1322	7513	143	0	0	0
200	98.1322	7513	143	0	0	0

Algorithm: Random

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	94.6447	7246	410	360	193	167
100	97.4660	7462	194	94	10	84
150	98.1322	7513	143	0	0	0
200	98.1322	7513	143	0	0	0

Algorithm: Optimal

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	97.6097	7473	183	133	28	105
100	98.1322	7513	143	43	0	43
150	98.1322	7513	143	0	0	0
200	98.1322	7513	143	0	0	0

Algorithm: Clock

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	95.9117	7343	313	263	110	153
100	97.9885	7502	154	54	0	54
150	98.1322	7513	143	0	0	0
200	98.1322	7513	143	0	0	0

Program: Simpleloop

Algorithm: LRU

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	74.6820	8221	2787	2737	88	2649
100	75.5541	8317	2691	2591	2	2589
150	75.5814	8320	2688	2538	0	2538
200	75.5814	8320	2688	2488	0	2488

Algorithm: FIFO

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	72.9288	8028	2980	2930	202	2728
100	74.9001	8245	2763	2663	44	2619
150	75.2725	8286	2722	2572	16	2556
200	75.3452	8294	2714	2514	12	2502

Algorithm: Random

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	72.8743	8022	2986	2936	227	2709
100	74.7729	8231	2777	2677	61	2616
150	75.2816	8287	2721	2571	20	2551
200	75.3543	8295	2713	2513	21	2492

Algorithm: Optimal

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	75.6450	8327	2681	2631	27	2604
100	76.0084	8367	2641	2541	0	2541
150	76.0084	8367	2641	2541	0	2491
200	76.0084	8367	2641	2541	0	2441

Algorithm: Clock

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	74.5730	8209	2799	2749	98	2651
100	75.5723	8319	2689	2589	0	2589
150	75.5814	8320	2688	2538	0	2538
200	75.5814	8320	2688	2488	0	2488

Program: Heaploop (the fourth program chosen our own, from exercise 8, you can use "runit" on this program to generate the trace file, you can find the code and executable files of heaploop in the repo)

Algorithm: LRU

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	94.7651	7422	410	360	88	272
100	95.9908	7518	314	214	2	212
150	96.1440	7530	302	152	0	152
200	96.4632	7555	277	77	0	77

Algorithm: FIFO

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	93.4116	7316	516	466	162	304
100	95.6588	7492	340	240	21	219
150	95.9908	7518	314	164	0	164
200	96.1440	7530	302	102	0	102

Algorithm: Random

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	92.8115	7269	563	513	220	293
100	95.6205	7489	343	243	35	208
150	96.2589	7539	293	143	3	140
200	96.4888	7557	275	75	0	75

Algorithm: Optimal

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	96.1185	7528	304	254	27	227
100	96.6292	7568	264	164	0	164
150	96.6292	7568	264	114	0	114
200	96.6292	7568	264	64	0	64

Algorithm: Clock

Memory	Hit rate	Hit Count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count
50	94.4714	7399	433	383	100	283
100	95.9525	7515	317	217	3	214
150	96.0163	7520	312	162	0	162
200	96.1696	7532	300	100	0	100

Comparison paragraph: For all programs, Optimal algorithm always has the best performance and Random algorithm always has the worst. FIFO algorithm has better performance than Random, but the improvement is limited. Clock algorithm has better performance than FIFO and is little bit worse than LRU. The performance difference between Clock and LRU is small, since Clock is approximate LRU. To conclude, the performance order for all programs is Optimal > LRU > Clock > FIFO > Random.

LRU description: The performance of LRU is getting better when the memory size increases. However, the performance improvement is very small when the memory size is greater than 100. This may depends on how the program access memory, if the memory access has locality, then certain size of memory can have good performance and increase in memory size only has limited improvement. Besides, for these programs, since the number of unique vaddr in Blocked and Matmul are 144 and 143 which are smaller than 150, then when the memory size is greater than 150, the performance doesn’t change.