Assignment 4

Computer Architecture

Nidhi Hegde (180472)

Part A:

LRU Policy

	No of L1 accesses	No of L2 accesses	No of L1 Cache Misses	No of L2 Cache Misses	Percentage of L2 DoF Blocks	Percentage of L2 Blocks (>= 2 hits)
perl	602217256	873034	873034	16704	8.59076	53.7751
bzip	701200790	8745365	8745365	4548977	67.2544	52.2912
gcc	530755808	23224339	23224339	5035921	94.777	49.6113
mcf	539049280	68828431	68828431	33576159	60.981	18.1957
soplex	530837767	19552677	19552677	18814562	97.9148	22.377
hmmer	669811655	3285938	3285938	1619858	95.3233	29.6265
omnetpp	591902829	14092639	14092639	10471631	80.9856	36.7517
xalanc	569708840	16548039	16548039	2415618	49.7791	61.8453

Formula for calculating % of DoF: (#evicted entries with 0 hits + # entries with 0 hits finally present in cache) / # L2 blocks filled

We see that the LRU policy offers the lowest L1 miss rate for the Perl benchmark. The highest miss rate is seen in mcf benchmark. Perl shows the lowest percentage of dead-on-fill blocks. Xalanc shows the highest percentage of L2 blocks experiencing at least 2 hits before getting evicted.

Part B:

SRRIP Policy

	No of L1 Cache Misses	No of L2 Cache Misses
perl	873035	16705
bzip	8747090	4601535
gcc	23223777	5026432
mcf	68830487	34482580
soplex	19548330	18759545
hammer	3279062	1555564
omnetpp	14088666	10456956
xalanc	16547131	2337919

We can see that this policy performs **better** than LRU Policy as it suffers a lesser number of L2 and L1 misses than LRU policy for most of the benchmarks.

REASON:

SRRIP policy addresses 2 of the problems – dead-on-fill misses and less frequently accessed blocks - encountered in the LRU Policy. Thus, we may obtain a better performance for this policy.

NRU Policy

	No of L1 Cache Misses	No of L2 Cache Misses
perl	873038	16724
bzip	8745671	4570825
gcc	23224605	5039692
mcf	68829771	33802251
soplex	19553118	18818585
hammer	3286045	1603119
omnetpp	14094205	10486398
xalanc	16549366	2427413

This policy offers a slightly **worse** performance than LRU and SRRIP Policy on most of the benchmarks.

REASON:

We just maintain the most recently accessed way index in the NRU policy along with the REF bits. As soon as all REF bits are set to 1(upon a hit or fill), we reset all of them except the most recently used entry. We do not differentiate between the second most recently used entry or the least recently used entry. If the former has a lower way index, we choose to evict it, though it may not be optimal. However, in case of LRU, we maintain additional information about the LRU state of all the entries, and only choose to evict the entry that was least recently accessed if need arises.

Possible Reason for Different L1 misses:

The 3 cache replacement policies are different for L2 cache only. But the policies affect the number of misses in L1 because of the inclusive cache hierarchy. When a block is evicted from L2, its corresponding entry in L1 is also evicted. Since different replacement policies may choose to evict different blocks in L2, the L1 cache state is not conserved across different policies. Hence, when a particular block that was indirectly evicted from L1 by the policy in the past is requested again, it would add an extra L1 miss.