Computer Architecture (CS423A), Spring 2023
Indian Institute of Technology Kanpur
Assignment 1

Members: Manish (190477), Sarthak Rout(190772)
Group Number: 10
Date: February 17, 2023

REPORT
1

Problem 1

Results for Inclusive Policy

Name	L2 Hit	L2 Miss	Total L2	L3 Hit	L3 Miss	Total L3
bzip2	5259461(49%)	5398166(51%)	10657627	3951778(73%)	1446388(27%)	5398166
gcc	11574350(79%)	3036461(21%)	14610811	1663059(54%)	1373402(46%)	3036461
gromacs	3094660(90%)	336851(10%)	3431511	166320(49%)	170531(51%)	336851
h264ref	1378895(58%)	969678(42%)	2348573	627532(64%)	342146(36%)	969678
hmmer	1766344(50%)	1743421(50%)	3509765	1352195(77%)	391226(23%)	1743421
sphinx3	1933098(17%)	8820349(83%)	10753447	612987(6%)	8207362(94%)	8820349

Results for NINE Policy

Name	L2 Hit	L2 Miss	Total L2	L3 Hit	L3 Miss	Total L3
bzip2	5260051(49%)	5397576(51%)	10657627	3951730(73%)	1445846(27%)	5397576
gcc	11581002(79%)	3029809(21%)	14610811	1663561(54%)	1366248(46%)	3029809
gromacs	3094787(90%)	336724(10%)	3431511	166265(49%)	170459(51%)	336724
h264ref	1382949(58%)	965624(42%)	2348573	632041(65%)	333583(35%)	965624
hmmer	1774443(50%)	1735322(50%)	3509765	1358978(78%)	376344(22%)	1735322
sphinx3	1938317(18%)	8815130(82%)	10753447	609986(6%)	8205144(94%)	8820349

Results for Exclusive Policy

Name	L2 Hit	L2 Miss	Total L2	L3 Hit	L3 Miss	Total L3
bzip2	5260051(49%)	5397576(51%)	10657627	4508355(83%)	889221(17%)	5397576
gcc	11581002(79%)	3029809(21%)	14610811	1786985(58%)	1242824(42%)	3029809
gromacs	3094787(90%)	336724(10%)	3431511	177422(52%)	159302(48%)	336724
h264ref	1382949(58%)	965624(42%)	2348573	821943(85%)	143681(15%)	965624
hmmer	1774443(50%)	1735322(50%)	3509765	1435276(82%)	300046(18%)	1735322
sphinx3	1938317(18%)	8815130(82%)	10753447	1594354(18%)	7220776(82%)	8820349

Observation and Analysis

- **Comparing L2 Hits:**
  NINE and Exclusive got more L2 Hit than Inclusive for all 6 Benchmarks. This is because an L3 eviction invalidates the block in L2 in the case of Inclusive. A block which got a lot of hits in L2 will be maintained as MRU in the L2 cache but the same block in L3 can easily become LRU as an L2 hit doesn't change L3 LRU metadata. When we need a new block in L3 this block is evicted due to the LRU replacement policy. Only in the case of Inclusive L2 block invalidation occurs resulting in more

L2 misses than NINE and Exclusive. NINE and Exclusive have the same policy regarding this so both of them have exactly the same L2 Hits.

- **Comparing L3 Hits:**
  Exclusive has a lot more hits than Inclusive and NINE. In the case of Inclusive L3 has L2's complete data. In the case of NINE L3 and L2 also shares some common data as L3 misses copying the block to both L2 and L3. It reduces the L3 capacity in both cases because the common data don't get any hit from L2 miss. Exclusive has more blocks in L3 to match the L2 miss because it had some more blocks in place of the common data resulting in more L3 hits.

  L3 hits are almost the same in the case of Inclusive and NINE because they only differ in L2 invalidation of the block on L3 eviction. So it only affects the L2 hits and misses.

Problem 2

Classifying L3 Misses:

So we have L3 Misses for a 16-way Associative 2MB cache following Inclusive and LRU Replacement Policy. We want to classify the misses into cold, capacity and Conflict Misses.

- **Cold Misses**: Cold misses are when access for a cache block comes for the first time(cannot be reduced).
- **Capacity Misses**: Capacity Misses are misses due to the limited size of the cache. These can be calculated as:
  Capacity misses = Total misses in Fully Associative Cache - Cold Misses
- **Conflict Misses**: Conflict misses are misses which occur due to limited associativity of cache resulting in a conflict in the set. These can be calculated as:
  Conflict Misses = Total misses in N-way associative cache - Total misses in a Fully associative Cache.

In the problem statement, we have to follow 2 replacement policies for fully associative. So Misses can be calculated as

- Cold Misses are the same for both policies.
- Capacity Misses(FA-LRU) = Total Misses in FA Cache following LRU replacement policy - Cold Misses.
- Capacity Misses(FA-Belady) = Total Misses in FA Cache following Belady replacement policy - Cold Misses.
- Conflict Misses(FA-LRU) = Total Misses in 16-way cache - Total Misses in FA Cache following LRU replacement policy
- Conflict Misses(FA-Belady) = Total Misses in 16-way cache - Total Misses in FA Cache following Belady replacement policy

Name	Total L3 Misses	Cold Misses	Capacity (FA-LRU)	Conflict (FA-LRU)	Capacity (FA-Belady)	Conflict (FA-Belady)
bzip2	1446388(27%)	119753	1241648	84987	417083	909552
gcc	1373402(46%)	773053	596871	3478	166236	434113
gromacs	170531(51%)	107962	61406	1163	35292	27277
h264ref	342146(36%)	63703	272177	6266	47902	230541
hmmer	391226(23%)	75884	301140	14202	77563	237779
sphinx3	8207362(94%)	122069	8265179	**-179886**	2946511	5138782

Observation and Analysis

- Cold misses will be the same in all caches. They cannot be reduced by changing size, associativity or replacement policy. These misses occur when the first access to a block happens. The block must be brought into the cache.
- Fully associative case with Belady replacement cache always has fewer total misses than Fully associative with LRU replacement cache as we know that in case of replacement Belady is optimal.
- Capacity Misses which are calculated using FA cache which uses LRU replacement policy will always be more than FA cache which uses Belady replacement policy because in the formula Capacity Misses = Total Misses in Fully Associative Cache - Cold Misses. Cold misses are the same(Point 1). Total Misses in Fully Associative Cache will be more in the case of LRU than Belday(Point 2).

- Conflict misses which are calculated using FA cache which uses LRU replacement policy will always be less than FA cache which uses Belady replacement policy because in the formula Conflict Misses = Total misses in n-way associative cache - Total misses in a Fully associative Cache Total misses in the n-way cache are the same but the misses in the Fully associative cache which uses the LRU replacement policy are more than the Fully associative cache which uses the Belady replacement policy(Point 2).
- We can observe that conflict misses can be negative. This can happen when there are more misses in fully associative cache as compared to set-associative caches. This can happen due to certain edge case behavior where the LRU block in the fully associative case may be used soon after evicting, in which doesn't happen in the set-associative case.