HW-4: Cache Experiments

Nitesh Gokul - AI23BTECH11006

Running the given programs in RIPES to observe Cache and draw conclusions.

Program-1:

Cache Observations:

L1(8,8):

Case-1: Fixing Blocks as 2 and Ways as 0 and varying Lines-1,2,3:

Increasing the number of lines does not change the hit-rate in program-1. This is because the code's reads are spatially localized, i.e, we access data from sequential memory locations. Adding more lines would help us prevent losing temporal locality but there is no change in Spatial locality accesses.

Hit-Rate with lines=1: 0.7424
Hit-Rate with lines=2: 0.7424
Hit-Rate with lines=3: 0.7424

Case-2: Fixing Lines as 3, Ways as 0 and varying Blocks- 3,4,5:

Increasing the number of Blocks increases the hit-rate in program-1. This is because the code's reads are spatially localized, i.e, we access data from sequential memory locations. Adding more Blocks would help accessing spatially local data easier and faster, since adding more blocks will retrieve data to cache faster.

Hit-Rate with Blocks=3: 0.8636
Hit-Rate with Blocks=4: 0.9242
Hit-Rate with Blocks=5: 0.9545

Case-3: Fixing Lines as 3, Blocks as 2 and varying Ways- 0,1:

Changing the number of Ways does not change the hit-rate in program-1. This is because the code's reads are spatially localized, i.e, we access data from sequential memory locations. Adding more Ways would help us prevent losing temporal locality but there is no change in Spatial locality accesses.

Hit-Rate with Ways=0: 0.7424
Hit-Rate with Ways=1: 0.7424

L1(16,16):

The hit rate trend continues to be the same although hit rate values change.

Hit-rate in case-1, Lines = 1: 0.7481
Hit-rate in case-1, Lines = 2: 0.7481
Hit-rate in case-1, Lines = 3: 0.7481
Hit-rate in case-2, Blocks = 3: 0.8721
Hit-rate in case 2, Blocks = 4: 0.9341
Hit-rate in case 2, Blocks = 5: 0.9651

Hit-rate in case 3, Ways = 0: 0.7481
Hit-rate in case 3, Ways = 1: 0.7481

	Case 11	Case 12	Case 13	Case 23	Case 24	Case 25	Case 30	Case 31
Lines	1	2	3	3	3	3	3	3
Ways	0	0	0	0	0	0	0	1
Blocks	2	2	2	3	4	5	2	2
Hits(8,8)|(16,16)	49 | 193	49 | 193	49 | 193	57 | 225	61 | 241	63 | 249	49 | 193	49 | 193
Misses(8,8) | (16,16)	17 | 65	17 | 65	17 | 65	9 | 33	5 | 17	3 | 9	17 | 65	17 | 65
Accesses (8,8) | (16,16)	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258
Hit-Rate (8,8)|(16,16)	0.7424 | 0.7481	0.7424 | 0.7481	0.7424 | 0.7481	0.8636 | 0.8721	0.9242 | 0.9341	0.9545 | 0.9651	0.7424 | 0.7481	0.7424 | 0.7481

	Write through and allocate	Write through and without allocate	Write back and allocate	Write back and without allocate
Lines	5	5	5	5
Ways	0	0	0	0
Blocks	2	2	2	2
Hits (8,8) | (16,16)	49 | 193	3 | 3	49 | 193	3 | 3
Misses (8,8) | (16,16)	17 | 65	63 | 255	17 | 65	63 | 255
Accesses (8,8) | (16,16)	66 | 258	66 | 258	66 | 258	66 | 258
Hit-Rate (8,8) | (16,160	0.7424 | 0.7481	0.04545 | 0.01163	0.7424 | 0.7481	0.04545 | 0.01163

Program-2:

Cache Observations:

Case-1: Fixing Blocks as 2 and Ways as 0 and varying Lines-1,2,3:

The hit rate in this case is very low and also does not change a lot with an increase in the number of lines. The reason for the low hit rate is because the code accesses elements of a 2D 8*8 array column by column.

Hit-Rate with Lines=1: 0.0303
Hit-Rate with Lines=2: 0.0303
Hit-Rate with Lines=3: 0.04545

Case-2: Fixing Lines as 3, Ways as 0 and varying Blocks- 3,4,5:

The hit rate in this case increases with increase in the number of blocks. The reason being we are accessing data from a 8*8 Array column by column.

Hit-Rate with Blocks=3: 0.8636
Hit-Rate with Blocks=4: 0.9242
Hit-Rate with Blocks=5: 0.9545

Case-3: Fixing Lines as 3, Blocks as 2 and varying Ways- 0,1:

The hit-rate in this case increases with increasing associativity. This is because the element access in this method happens column wise. Initially there were only 4 blocks and hence every miss will be replaced but in case of 2-way associativity, we have less replacements and hence making the access easy.

Hit-Rate with Ways=0: 0.04545
Hit-Rate with Ways=1: 0.7424

L1(16,16):

The hit rate trend continues to be the same although hit rate values change.

Hit-rate in case-1, Lines = 1: 0.007752
Hit-rate in case-1, Lines = 2: 0.007752
Hit-rate in case-1, Lines = 3: 0.007752
Hit-rate in case-2, Blocks = 3: 0.007752
Hit-rate in case 2, Blocks = 4: 0.01163
Hit-rate in case 2, Blocks = 5: 0.9574
Hit-rate in case 3, Ways = 0: 0.00752
Hit-rate in case 3, Ways = 1: 0.7424

	Case 11	Case 12	Case 13	Case 23	Case 24	Case 25	Case 30	Case 31
Lines	1	2	3	3	3	3	3	3
Ways	0	0	0	0	0	0	0	1
Blocks	2	2	2	3	4	5	2	2
Hits(8,8)|(16,16)	2 | 2	2 | 2	3 | 2	57 | 2	61 | 3	63 | 247	3 | 2	49 | 2
Misses(8,8) | (16,16)	64 | 256	64 | 256	63 | 256	9 | 256	5 | 255	3 | 11	63 | 256	17 | 256
Accesses (8,8) | (16,16)	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258	66 | 258
Hit-Rate (8,8)|(16,16)	0.0303 | 0.007752	0.0303 | 0.007752	0.0454 | 0.007752	0.8636 | 0.007752	0.9242 | 0.001163	0.9545 | 0.9574	0.0454 | 0.00752	0.7424 | 0.007752

	Write through and allocate	Write through and without allocate	Write back and allocate	Write back and without allocate
Lines	5	5	5	5
Ways	0	0	0	0
Blocks	2	2	2	2
Hits (8,8) | (16,16)	49 | 3	3 | 3	49 | 3	3 | 3
Misses (8,8) | (16,16)	17 | 255	63 | 255	17 | 255	63 | 255
Accesses (8,8) | (16,16)	66 | 258	66 | 258	66 | 258	66 | 258
Hit-Rate (8,8) | (16,160	0.7424 | 0.01163	0.04545 | 0.01163	0.7424 | 0.01163	0.04545 | 0.01163

Program-3:

L1(8,8):

Case-1: 32 entry 4-word direct map:

The hit rate in this case is very low since ld x20, 0(x12) and ld x20, 1024(x12) fall in the same index after the first load, all the next loads will be a miss and hence resulting in a very low hit rate.

Hit-Rate in this case: 0.01538

Case-2: 32 entry 4-word 2-way set associative map:

The hit rate in this case is higher compared to the previous case, since ld x20, 0(x12) and ld x20, 1024(x12) fall in the same index they are stored in the other set, i.e, the second set of the same index and hence the data is stored in cache and data loss does not occur.

Hit-Rate in this case: 0.7385

Case-3: 32 entry 4-word fully associative map:

The hit rate in this case is the same as 2 way associative, and the same reason is applied here although we know maximum accesses are 128.

Hit-Rate in this case: 0.7385

L1(16,16):

The hit rate trend continues to be the same although values change.
Hit-Rate in direct map: 0.06809
Hit-Rate in 2-way set associative map: 0.7471

Hit-Rate in fully associative map: 0.7471

	32 entry 4 word direct maps	32 entry 4 word 2 way set associative	32 entry 4 word fully associative cache
Lines	5	4	0
Ways	0	1	5
Blocks	2	2	2
Hits (8,8) | (16,16)	2 | 35	96 | 384	96 | 384
Misses (8,8) | (16,16)	128 | 479	34 | 130	34 | 130
Accesses (8,8) | (16,16)	130 | 514	130 | 514	130 | 514
Hit-rate (8,8) | (16,16)	0.01538 | 0.06809	0.7385 | 0.7471	0.7385 | 0.7471