Homework 4 - Cache Experiments

CS2323 - Computer Architecture, Autumn 2023

Soham Rajesh Pawar
CS22BTECH11055

November 9, 2023

1 Question 1 :

Cache reads: Execute the assembly programs 1 and 2 and observe the hit rates for various data cache configurations. You should vary the configurations atleast as per the following (you are free to explore more combinations as well): Lines: 1, 2, 3 keeping Blocks as 2 and Ways as 0. In RIPES, Lines refers to the block size Blocks: 3, 4, 5 keping Lines as 3 and Ways as 0. In RIPES, Blocks refers to the number of blocks Ways: 0, 1 keeping Blocks as 2 and Lines as 3. In RIPES, Ways refers to the associativity.

1.1 Variation with Lines :

Varying Lines, Keeping Blocks = 2 and Ways = 0

Setting	*Program*	*Lines*	*Hit Rate*	*Number of Hits*	*Number of Misses*	*Total Acceses*
L1: (8, 8)	**1**	1	0.7424	49	17	66
		2	0.7424	49	17	66
		3	0.7424	49	17	66
	2	1	0.0303	2	64	66
		2	0.0303	2	64	66
		3	0.04545	3	63	66
L1: (16, 16)	**1**	1	0.7481	193	65	258
		2	0.7481	193	65	258
		3	0.7481	193	65	258
	2	1	0.007752	2	256	258
		2	0.007752	2	256	258
		3	0.007752	2	256	258

Inference :

a) Program 1: The number of lines doesn't significantly impact the hit rate. This is because the memory access is contiguous increasing spatial locality.

b) Program 2: A slight impact is observed on the hit rate, reducing the number of collisions.

1.2 Variation with Blocks :

Varying Blocks, Keeping Lines = 3 and Ways = 0						
Setting	**Program**	**Blocks**	**Hit Rate**	**Number of Hits**	**Number of Misses**	**Total Acceses**
L1: (8, 8)	1	3	0.8636	57	9	66
		4	0.9242	61	5	66
		5	0.9545	63	3	66
	2	3	0.8636	57	9	66
		4	0.9242	61	5	66
		5	0.9545	63	3	66
L1: (16, 16)	1	3	0.8721	225	33	258
		4	0.9341	241	17	258
		5	0.9651	249	9	258
	2	3	0.007752	2	256	258
		4	0.01163	3	255	258
		5	0.9574	247	11	258

Inference :

a) Program 1: Larger block size increases the hit rate due to contiguous memory access.

b) Program 2: Larger block size enhances hit rate by increasing the chances of the requested word being present in the cache.

Note :

In the latter setting for program 2, the hit rate jumps suddenly due to the huge increase in block size and increased spatial locality.

1.3 Variation with Ways :

Varying Ways, Keeping Blocks = 2 and Lines = 3						
Setting	**Program**	**Ways**	**Hit Rate**	**Number of Hits**	**Number of Misses**	**Total Acceses**
L1: (8, 8)	1	0	0.7424	49	17	66
		1	0.7424	49	17	66
	2	0	0.04545	3	63	66
		1	0.7424	49	17	66
L1: (16, 16)	1	0	0.7481	193	65	258
		1	0.7481	193	65	258
	2	0	0.007752	2	256	258
		1	0.007752	2	256	258

Inference :

a) Program 1: The number of ways doesn't significantly affect the hit rate, as once a block is exhausted, it is not accessed again.

b) Program 2: Increases the hit rate by utilizing data fetched in the past, taking advantage of the temporal locality of the program.

2 Question 2 :

Write-policies: Modify programs 1 and 2 to replace "ld x20, 0(x12)" to "sd x20, 0(x12)". Use the preset cache configuration (32-entry 4-word direct mapped). Run program-1 and program-2 for various combinations of Write-through and Write-back policies (with allocate and without allocate).

Preset : Cache Configuration (32 Entry 4 Word Direct Mapped)							
Setting	*Program*	*Write Policy*	*Allocate*	*Hit Rate*	*Number of Hits*	*Number of Misses*	*Writebacks*
L1: (8, 8)	1	WriteThrough	YES	0.7424	49	17	64
			NO	0.04545	3	63	64
		WriteBack	YES	0.7424	49	17	0
			NO	0.04545	3	63	62
	2	WriteThrough	YES	0.7424	49	17	64
			NO	0.04545	3	63	64
		WriteBack	YES	0.7424	49	17	0
			NO	0.04545	3	63	62
L1: (16, 16)	1	WriteThrough	YES	0.7481	193	65	256
			NO	0.01163	3	255	256
		WriteBack	YES	0.7481	193	65	33
			NO	0.01163	3	255	254
	2	WriteThrough	YES	0.01163	3	255	256
			NO	0.01163	3	255	256
		WriteBack	YES	0.01163	3	255	223
			NO	0.01163	3	255	254

Inference :

a) Program 1: Allocation allowed increases the number of hits compared to when disallowed, regardless of the policy. This is because a block is allocated in the cache on a miss, increasing subsequent hits due to increased spatial locality.

b) Program 2: In the former setting, the above logic applies. However, in the latter setting, the data is too large to benefit from spatial locality i.e before it can be accessed again it's replaced by another block due to collisions.

3 Question 3 :

Associativity: Execute program 3 and vary the following configurations:

a) 32-entry 4-word direct mapped

b) 32-entry 4-word 2-way set associative

c) 32-entry 4-word fully associative

Setting	*Preset Configuration*	*Hit Rate*	*Number of Hits*	*Number of Misses*	*Total Acceses*
L1: (8, 8)	32 Entry 4 Word Direct Mapped	0.01538	2	128	130
	32 Entry 4 Word 2 Way Set Associative	0.7385	96	34	130
	32 Entry 4 Word Fully Associative	0.7385	96	34	130
L1: (16, 16)	32 Entry 4 Word Direct Mapped	0.06809	35	479	514
	32 Entry 4 Word 2 Way Set Associative	0.7471	384	130	514
	32 Entry 4 Word Fully Associative	0.7471	384	130	514

Inference :

As the data is the same, having more ways would be beneficial, allowing retention of older blocks and possibly taking advantage of any temporal locality the program may have.