計算機結構 Exercise 05

B03902062 資工三 董文捷

5.13.1

address	hit / miss	evicted block	Set 0		Set 1
0	miss		Mem[0]		
2	miss		Mem[0]	Mem[2]	
4	miss	0	Mem[4]	Mem[2]	
8	miss	2	Mem[4]	Mem[8]	
10	miss	4	Mem[10]	Mem[8]	
12	miss	8	Mem[10]	Mem[12]	
14	miss	10	Mem[14]	Mem[12]	
16	miss	12	Mem[14]	Mem[16]	
0	miss	14	Mem[0]	Mem[16]	

0 hit

5.13.2

address	hit / miss	evicted block	Set 0		Set 1
0	miss		Mem[0]		
2	miss		Mem[0]	Mem[2]	
4	miss	2	Mem[0]	Mem[4]	
8	miss	4	Mem[0]	Mem[8]	
10	miss	8	Mem[0]	Mem[10]	
12	miss	10	Mem[0]	Mem[12]	
14	miss	12	Mem[0]	Mem[14]	
16	miss	14	Mem[0]	Mem[16]	
0	hit		Mem[0]	Mem[16]	

1 hit

5.13.3

address	hit / miss	flipping coin result	evicted block	Set 0		Set 1
0	miss			Mem[0]		
2	miss			Mem[0]	Mem[2]	
4	miss	tails	2	Mem[0]	Mem[4]	
8	miss	tails	4	Mem[0]	Mem[8]	
10	miss	heads	0	Mem[10]	Mem[8]	
12	miss	heads	10	Mem[12]	Mem[8]	
14	miss	heads	12	Mem[14]	Mem[8]	
16	miss	tails	8	Mem[14]	Mem[16]	
16	miss	tails	16	Mem[14]	Mem[0]	

0 hit

5.13.4 The only address that is accessed twice in this address sequence is `0`. The optimal policy should keep `Mem[0]` in `Set 0` and replace the other block in the set when there is a miss. In this case, the optimal policy is the same as an MRU replacement policy, the address sequence exhibits 1 hit by following this policy.

5.13.5 To implement a cache replacement policy that is optimal for all address sequences, we must know which address will be accessed in the future. However, the cache controller is not able to know the future, it can only make a prediction and the policy may not be optimal.

5.13.6 If we are able to know which address will not be accessed in the future, we can decide not to cache it, some other address can be kept in the cache, and the miss rate may be improved. However, if we do not know the address sequence and decide not to cache an address which is accessed again soon, the miss rate may become worse.

5.15.1 $CPI = 1.5 + \frac{120}{10000} \times (15 + 175) = 3.78$

$CPI_{double} = 1.5 + \frac{120}{10000} \times (15 + 175 \times 2) = 5.88$

$CPI_{half} = 1.5 + \frac{120}{10000} \times (15 + 175 \times 0.5) = 2.73$

$(1.5 + \frac{120}{10000} \times (15 + 175)) \times 0.9 = 1.5 + \frac{120}{10000} \times (15 + penalty)$

$\longrightarrow$ longest possible penalty to trap to the VMM $= 143.5$

5.15.2 non-virtualized : $CPI = 1.5 + \frac{30}{10000} \times 1100 = 4.8$

virtualized : $CPI = 1.5 + \frac{120}{10000} \times (15 + 175) + \frac{30}{10000} \times (1100 + 175)$ $= 7.605$

half the I/O accesses

non-virtualized : $CPI = 1.5 + \frac{15}{10000} \times 1100 = 3.15$

virtualized : $CPI = 1.5 + \frac{120}{10000} \times (15 + 175) + \frac{15}{10000} \times (1100 + 175)$ $= 5.6925$

Compared to the I/O access time, performance impact to trap to the guest O/S and VMM is relatively small. Therefore, virtualization has a smaller percentage of impact on I/O bound applications.