1]

OS Test

Fernando Vazquez
3/4/19

Direct Map

Reference	Binary Address	Hit/miss	index	tag
6	0000 0110	miss	0011	0
26	0001 1010	miss	1101	0
7	0000 0111	~~miss~~ Hit	0011	0
24	0001 1000	miss	1100	0
17	0001 0001	Miss	1000	0
66	0100 0010	Miss	0001	0
22	0001 0110	Miss	1011	0
6	0000 0110	~~Miss~~ Miss	0011	0
8	0000 1000	miss	0100	0
118	0111 0110	miss	1011	0
68	0100 0100	Miss	0010	0

index = $\log_2 16 = 4$
offset = $\log_2 8 = 3$

of tag bits = 8 − 4 − 3 = 1 (index bits)

cache address		TAG	Data
0	0000		
1	0001	0001	0001 0001
2	0010	0100	0100 0010
3	0011		
4	0100	0100	0100 0100
5	0101		
6	0110	~~0000~~ 0111	~~0000 0110~~ 0111 0110
7	0111	0000	0000 0111
8	1000	0000	0000 1000
9	1001		
10	1010	0001	0001 1010
11	1011		
12	1100		
13	1101		
14	1110		
15	1111		

2-way associative

	cache address
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

	Bin address		Hit/Miss	index	tag
6	0000	0110	Miss	011	00
26	0001	1010	Miss	101	00
7	0000	0111	Miss	011	00
24	0001	1000	Miss	100	00
17	0001	0001	Miss	000	00
66	0100	0010	Miss	001	01
22	0001	0110	Miss	011	00
6	0000	0110	~~Miss~~ Hit	011	00
8	0000	1000	Miss	100	00
118	0111	0110	Miss	011	01
68	0100	0100	Miss	010	01

~~Index~~ : # of index $= \log_2 8 = 3$

# of offset $= \log_2 8 = 3$

# of Tag $= 8-3-3 = 2$

2.

	arrival	Burst	Priority
P1	0	3	2
P2	1	5	4
P3	2	6	1
P4	3	2	3

Quantum = 2 units

a) preemptive SJF

1	1	1	2	5	6
P1	P1	P1	P4	P2	6

0 1 2 3 5 10 16

B) preemptive priority

2	6	1	2	5
P1	P3	P1	P4	P2

0 2 8 9 11 16

C RR

2	2	2	2	1	2	2	1	2
P1	P2	P3	P4	P1	P2	P3	P2	P3

0 2 4 6 8 9 11 13 14 16

2a. P2 wait : 4
P3 wait : 8

average wait time for preemptive SJF is 3 time units

2b. P1 6
P2 10
P3 0
P4 6

average wait time for preemptive priority is $(6+10+0+6)/4$ = 5.5 units of time

P4: 3+

$\frac{3+6+8+8}{4} = 25/4 = 6.25$

2C. P1: 6
P2: 5+2 (+1) = 8
P3: 2+5+1 = 8

average wait time for RR is 6.25 time units

3. Parallelizable = 40%

$$\text{speedup} = \frac{1}{(1-F_p) + F_p/N}$$

For speedup of 1.5
$N=6$ cores is needed.

$$1.5 = \frac{1}{1-.4 + .4/N}$$

$$1.5((1-.4) + .4/N) = 1$$

$$.6 + .4/N = 1/1.5$$

$$.4/N = (1/1.5) - .6$$

$$\frac{.4}{(\frac{1}{1.5}) - .6} = N$$

$$N = 6$$

For the speedup of 2.2, it is not possible to reach because the limit of Amdahl's law is $1/_{1-F_p} = 1/_{1-.4} = 1.6666$. So the maximum spedup possible with 40% of code parrallelizable is 1.6667

$$2.2 = \frac{1}{1-.4 + .4/N}$$

$$2.2 = \frac{1}{.6 + .4/N}$$

$$.6 + .4/N = \frac{1}{2.2}$$

$$.4/N = .4545 - .6$$

$$\frac{.4}{.4545 - .6} = N$$

4. step 1: assign 4 numbers from array A and 4 numbers to array B to the 16 cores.

step2: multiply each A value to its corresponding B value. 4 steps

$\square + \square$ $\square + \square$ step 3: add the numbers together $a_1b_1 + a_2b_2$ and $a_3b_3 + a_4b_4$, for each of the processers 4 numbers.
2 steps

step4: Each processer adds its last 2 numbers. 1 step

step5: 8 processers add the remaining 16 numbers. 1step

step6: 4 processers add the remaining 8 numbers. 1 step

step 7: 2 processers add the remaining 4 numbers.
1 step

step 8: 1 processor adds the last 2 numbers, thus resulting in the dot product of 2 arrays of 64 integers each.

Total steps is 11 steps to calculate the dot product.