

Άσκηση 3η: Ιεραρχία Μνήμης Αρχιτεκτονική Υπολογιστών

Ανδρέας Στάμος
Αριθμός Μητρώου: 03120***

Ιανουάριος 2023

Περιεχόμενα

Περιεχόμενα	1
1 Μέρος A	1
2 Μέρος B	1
2.1 Ερώτημα A	2
2.2 Ερώτημα B	3

1 Μέρος A

Σε ένα επίπεδο κρυφής μνήμης ο μέσος χρόνος πρόσβασης σε πλήθος κύκλων είναι γενικά:

Μέσος χρόνος πρόσβασης = Πιθανότητα hit · Χρόνος ευστοχίας + Πιθανότητα miss · Χρόνος αστοχίας (1)

Συνεπώς η δοθείσα κρυφή μνήμη με το 1 επίπεδο έχει μέσο χρόνο πρόσβασης:

$$\text{Μέσος χρόνος πρόσβασης}_{1 \text{ επίπεδο}} = 0.975 \cdot 1 + (1 - 0.975) \cdot 90 = 3.225 \text{ κύκλοι}$$

Σχετικά με την κρυφή μνήμη 2 επιπέδων, έστω t ο χρόνος πρόσβασης από το 2ο επίπεδο της κρυφής μνήμης σε πλήθος κύκλων. Τότε ο μέσος χρόνος πρόσβασης της συνολικής κρυφής μνήμης είναι:

$$\text{Μέσος χρόνος πρόσβασης}_{2 \text{ επιπέδων}} = 0.975 \cdot 1 + (1 - 0.975) \cdot \left(0.88 \cdot t + (1 - 0.88) \cdot 90 \right)$$

Ζητείται να ισχύει:

$$\text{Μέσος χρόνος πρόσβασης}_{2 \text{ επιπέδων}} \leq \frac{1}{2.24} \cdot \text{Μέσος χρόνος πρόσβασης}_{1 \text{ επίπεδο}}$$

Επιλύοντας την ανίσωση λαμβάνουμε $t \leq 4.49$ κύκλοι, δηλαδή ο χρόνος πρόσβασης στο 2ο επίπεδο της μνήμης πρέπει να είναι μικρότερος ή ίσος με 4 κύκλους.

2 Μέρος B

Στο ζητούμενο τμήμα κώδικα συμβαίνουν 24 επαναλήψεις και σε κάθε επανάληψη συμβαίνουν 4 προσβάσεις στην μνήμη (αναγνώσεις ή εγγραφές). Συνολικά, επομένως, συμβαίνουν 96 προσβάσεις στην μνήμη. Η εκτέλεση της προσομοίωσης της κρυφής μνήμης με το χέρι είναι επιπόνη και χρονοβόρα και για αυτό συντάχθηκε πρόγραμμα για τον υπολογισμό των hits και misses. Το πρόγραμμα είναι το εξής

```
1 import operator
2
3 maxint = int(1e9)
4
5 m = 4 #number of blocks in each set
6 sets = 2 #number of sets
```

```

7  blocksize = 4 #4 floating-point numbers per block (32 bytes per block / 8 bytes per fpv)
8
9  #cache[i] is the ith set
10 #cache[i][j] is the jth block of the ith set
11 #cache[i][j][0] is the number of block of the main memory
12 #cache[i][j][1] is the age
13 cache = [ [[-1,maxint] for _ in range(m)] for _ in range(sets)]
14
15 misses = 0
16 hits = 0
17
18 def access(loc):
19     global hits, misses
20
21     #Calculating in which block the requested location belongs
22     block = loc // blocksize
23     #Calculating in which set the requested block belongs
24     set_loc = block % sets
25
26     #Increasing the age of all blocks.
27     for i in range(sets):
28         for j in range(m):
29             cache[i][j][1] += 1
30
31     #Checking if the requested block is in the cache and if it is setting the age to 0.
32     for i in range(m):
33         if cache[set_loc][i][0] == block:
34             hits += 1
35             cache[set_loc][i][1] = 0
36             return
37
38     #From now on it is certainly a miss.
39     misses += 1
40
41     #Finding the least recently used block. (it is the one with the maximum age)
42     lru = cache[set_loc].index(max(cache[set_loc], key=operator.itemgetter(1)))
43
44     #"Loading" in the cache and setting the age to 0.
45     cache[set_loc][lru][0] = block
46     cache[set_loc][lru][1] = 0
47
48     for i in range(6):
49         for j in range(0,8,2):
50             access(64 + (i%4)*8 + j)
51             access((i%3)*8 + j)
52             access(128 + j)
53             access((i%3)*8 + j)
54
55     print("Hits: ", hits)
56     print("Misses: ", misses)
57     print("Total: ", hits+misses)
58     print("Hit rate: ", hits/(hits+misses))
59     print("Miss rate: ", misses/(hits+misses))

```

2.1 Ερώτημα Α

Έφροσον με το πρόγραμμα θα επαληθεύσουμε επακριβώς το hit rate, θα επιχειρήσουμε να δώσουμε μια προσεγγιστική έκφραση για το πού αναμένεται να κυμανθεί, προκειμένου να κατανοήσουμε και καλύτερα το πώς αξιοποιεί ο κώδικας την κρυφή μνήμη. Αν η μνήμη ήταν πλήρως συσχετιστική, θα είχαμε προσεγγιστικά 3 αστοχίες χωρητικότητας ανά 2 επαναλήψεις (σε 2 συνεχόμενες επαναλήψεις γίνεται επεξεργασία των ίδιων blocks). Τα blocks των A,B στα οποία γίνεται πρόσβαση έχουν πάντα ίδιο mod4. Είναι 2 και κάθε σύνολο χωράει 2 blocks. Αρά δεν

αναμένονται αστοχίες σύγκρουσης μεταξύ τους. Αστοχίες σύγκρουσης αναμένονται μόνο από αναγνώσεις του C αν τύχει να είναι στο ίδιο mod4. Αυτό όμως αναμένεται σπάνια. Θα το αγνοήσουμε στα πλαίσια της προσέγγισης. Συνεπώς αναμένονται $\frac{3}{2} \cdot 24 = 36$ misses. Συνολικά έχουμε 96 προσβάσεις στην μνήμη. Αρα αναμένεται miss rate $\frac{36}{96} = 35\% - 40\%$.

Η εκτέλεση όλης της ακολουθίας των προσβάσεων στην μνήμη, με χρήση του προγράμματος που δόθηκε προηγουμένως, έδωσε 38 misses και 56 hits, δηλαδή miss rate 39.6%. Η προσέγγιση μας επομένως σαν αποτέλεσμα ήταν σχετικά καλή.

2.2 Ερώτημα Β

Η αύξηση της συσχετιστικότητας προφανώς θα αυξήσει το hit rate εφόσον θα απαλείψει πλήρως τις αστοχίες σύγκρουσης. Συγκεκριμένα για μνήμη ίδιας χωρητικότητας αλλά 4-way set associative, θα έχουμε συνολικά 2 sets. Εισάγοντας τις παραμέτρους αυτές στο προηγούμενο πρόγραμμα, λαμβάνουμε 26 misses, 70 hits, δηλαδή miss rate 27.1%.

Η αύξηση της συσχετιστικότητας θα μειώσει τον χρόνο που απαιτείται για πρόσβασεις στην κυρία μνήμη, θα αυξήσει όμως τον χρόνο ευστοχίας. Ο μέσος χρόνος πρόσβασης στην μνήμη είναι:

$$\text{hit rate} \cdot \text{χρόνος ευστοχίας} + \text{miss rate} \cdot \text{χρόνος αστοχίας} \quad (2)$$

Έστω α_1 και $\alpha_2 = \lambda \cdot \alpha_1$ ο χρόνος ευστοχίας για 2-way και 4-way κρυφή μνήμη και β ο χρόνος αστοχίας σε πλήθος κύκλων. Η αύξηση της συσχετιστικότητας συμφέρει αν:

$$72.9\alpha_2 + 27.1\beta < 60.4\alpha_1 + 39.6\beta \Leftrightarrow \lambda < 0.171 \frac{\beta}{\alpha_1} + 0.83 \quad (3)$$

Ο λόγος $\frac{\beta}{\alpha_1}$ γνωρίζουμε ότι είναι σίγουρα μεγαλύτερος από 10, ενώ η επιβάρυνση λ στον χρόνο ευστοχίας λόγω διπλασιασμού της συσχετιστικότητας δεν θα είναι παραπάνω από 2-πλασια (το πολύ να χρειαστεί ένα επιπλέον επίπεδο στους συγκριτές). Συνεπώς από απόψη ταχύτητας συμφέρει ξεκάθαρα η αύξηση της συσχετιστικότητας. Ενεργειακά, η αύξηση της συσχετιστικότητας, από την στιγμή που η μνήμη είναι ήδη συσχετιστική, δηλαδή έχει το υλικό για την lru πολιτική κλπ, θα απαιτήσει απλά λίγες πύλες παραπάνω για την επιπλέον συγκρίσεις. Είναι σχεδόν αμελητέα η ενεργειακή επιβάρυνση.

Τελικά, λοιπόν, η αύξηση της συσχετιστικότητας συμφέρει.