



Παράλληλος Προγραμματισμός

Προγραμματιστική Εργασία 1

Πάτρικ Βάις Α.Μ: Π2014090

Περιγραφή Κώδικα

Δημιούργησα 2 εκδόσεις προγράμματος το οποίο είχε ως στόχο τον πολλαπλασιασμό δύο float πινάκων NxN και την αποθήκευση του αποτελέσματος σε τρίτο πίνακα NxN . Στο αρχείο [matmul-normal.c](#) βρίσκεται η κανονική έκδοση του κώδικα ενώ στο αρχείο [matmul-sse.c](#) βρίσκεται η έκδοση του κώδικα με εντολές SSE2. Και οι δύο εκδόσεις του κώδικα ακολουθούν τα εξής βήματα:

1. Δήλωση μεταβλητών
2. Δέσμευση πινάκων (+ έλεγχος ότι έγινε)
3. Αρχικοποίηση (cache warm-up)
4. Έναρξη καταμέτρησης του χρόνου
5. Φορτίο
6. Τερματισμός καταμέτρησης του χρόνου
7. Έλεγχος
8. Υπολογισμός και εκτύπωση χρόνου
9. Αποδέσμευση πινάκων

Η καταμέτρηση του χρόνου γίνεται με την βοήθεια της συνάρτησης `get_walltime` την οποία δημιουργήσαμε στο εργαστήριο. Ο πολλαπλασιασμός των πινάκων a και b γίνεται με την βοήθεια ενός for loop το οποίο δημιουργήσαμε επίσης στο εργαστήριο και γίνεται τμηματικά. Η διαφοροποίηση της δεύτερης έκδοσης του κώδικα σε σχέση με την πρώτη είναι το γεγονός ότι καταφέραμε με την βοήθεια των εντολών SSE2 να κάνουμε 4 πράξεις ταυτόχρονα σε κάθε επανάληψη.

Αποτελέσματα

	Απλή Υλοποίηση		Υλοποίηση με SSE	
N	Χρόνος	MFlops	Χρόνος	MFlops
4	0.000001	inf	0.000001	inf
40	0.000218	7.334302	0.000024	66.444420
400	0.071647	2.233173	0.026291	6.085702
2000	8.647821	0.462544	6.068971	0.659090
4000	70.060515	0.228374	66.185487	0.241745

Πίνακας 1. Αποτελέσματα

```
File Edit View Search Terminal Help
root@Kali:~/Desktop/Askisi1# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          39 bits physical, 48 bits virtual
CPU(s):                 4
On-line CPU(s) list:    0-3
Thread(s) per core:     2
Core(s) per socket:     2
Socket(s):              1
NUMA node(s):           1
Vendor ID:              GenuineIntel
CPU family:              6
Model:                  69
Model name:              Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz
Stepping:                1
CPU MHz:                 884.422
CPU max MHz:             3000.0000
CPU min MHz:             800.0000
BogoMIPS:                4789.24
Virtualization:          VT-x
L1d cache:               32K
L1i cache:               32K
L2 cache:                 256K
L3 cache:                 4096K
NUMA node0 CPU(s):       0-3
```

Φωτογραφία 1. Αποτέλεσμα εντολής lscpu

Επεξήγηση αποτελεσμάτων

Θεωρητικά θα πρέπει να πετύχουμε πιο γρήγορη εκτέλεση του προγράμματος στην δεύτερη περίπτωση καθώς σε μία επανάληψη καταφέραμε να κάνουμε 4 πράξεις ταυτόχρονα. Πράγματι αν και ελάχιστη διαφορά βλέπουμε πως υπάρχει καλύτερη απόδοση όταν χρησιμοποιούμε τις εντολές SSE. Βλέπουμε μεγαλύτερη αποδοτικότητα και ελάχιστα πιο γρήγορους χρόνους.