

Όνομα : Διακουμάκος Βασίλειος
AM: Π2015088
Μάθημα : Παράλληλος Προγραμματισμός

Στην παρούσα εργασία και οι δύο κώδικες βασίστηκαν στα εργαστήρια του μαθήματος. Αφού ακολούθησα τις παρακάτω τεχνικές

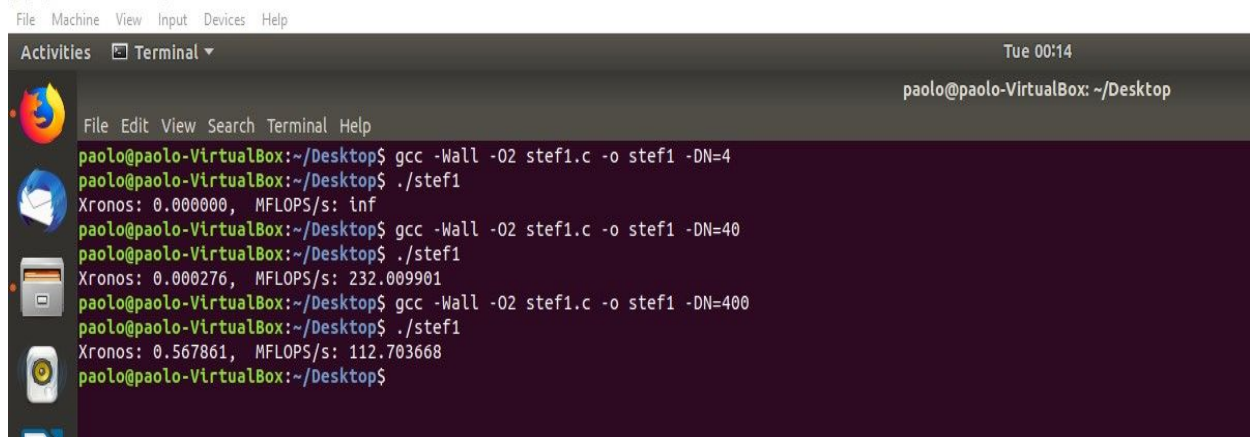
- 1) Δέσμευση πινάκων
- 2) Αρχικοποίηση για την κατάλληλη προετημασία οσον αφορά την κρυφή μνήμη
- 3) Φορτίο δηλαδή ο πολλαπλασιασμός των πινάκων a, b αποθηκεύοντας τα αποτελέσματα σε έναν τρίτο πίνακα C
- 4) Έλεγχος των αποτελεσμάτων
- 5) Χρήση mlops, και get_walltime
- 6) Αποδέσμευση των πινάκων που δεσμεύσαμε στην αρχή με την χρήση malloc και posix_memalign αντίστοιχα

Compile lines: gcc -Wall -O2 matmul-normal.c -o matmul-normal - DN= (ανάλογη τιμή N)
gcc -Wall -O2 matmul-sse.c -o matmul-sse - DN= (ανάλογη τιμή N)

Για να βεβαιωθώ ότι δεν έγινε απαλοιφή των for χρησιμοποίησα στην γραμμή του compile την εντολή -S η οποία μας εμφανίζει το assembly αρχείο του κώδικα

Η διαφορά στα δύο προγράμματα που γράφτηκαν είναι ότι στο matmul-sse.c χρησιμοποιήθηκαν τεχνικές βελτιστοποίησης με διάφορες εντολές που μας παρέχει η βιβλιοθήκη #include <emmintrin.h>. Ορισμένες εντολές που χρησιμοποιήθηκαν είναι η (__m128), posix_memalign, _mm_add_ps και άλλες sse εντολές που βοηθούν στην καλύτερη επεξεργασία των δεδομένων. Στο matmul-normal ο πολλαπλασιασμός έγινε κανονικά row * column, ενώ στο matmul-sse χρησιμοποιήθηκε η τεχνική ανα τετράδες και χρειάστηκε ένας βοηθητικός πίνακας D για τον υπολογισμό.

Τα αποτελέσματα είναι τα εξής : matmul-normal.c



```
File Machine View Input Devices Help
Activities Terminal Tue 00:14
paolo@paolo-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
paolo@paolo-VirtualBox:~/Desktop$ gcc -Wall -O2 stef1.c -o stef1 -DN=4
paolo@paolo-VirtualBox:~/Desktop$ ./stef1
Xronos: 0.000000, MFLOPS/s: inf
paolo@paolo-VirtualBox:~/Desktop$ gcc -Wall -O2 stef1.c -o stef1 -DN=40
paolo@paolo-VirtualBox:~/Desktop$ ./stef1
Xronos: 0.000276, MFLOPS/s: 232.009901
paolo@paolo-VirtualBox:~/Desktop$ gcc -Wall -O2 stef1.c -o stef1 -DN=400
paolo@paolo-VirtualBox:~/Desktop$ ./stef1
Xronos: 0.567861, MFLOPS/s: 112.703668
paolo@paolo-VirtualBox:~/Desktop$
```

matmul-sse.c

```
paolo@paolo-VirtualBox:~/Desktop$ gcc -Wall -O2 stef2.c -o stef2 -DN=4
paolo@paolo-VirtualBox:~/Desktop$ ./stef2
Xronos: 0.000000, MFLOPS/s: inf
paolo@paolo-VirtualBox:~/Desktop$ gcc -Wall -O2 stef2.c -o stef2 -DN=40
paolo@paolo-VirtualBox:~/Desktop$ ./stef2
Xronos: 0.000104, MFLOPS/s: 615.677651
paolo@paolo-VirtualBox:~/Desktop$ gcc -Wall -O2 stef2.c -o stef2 -DN=400
paolo@paolo-VirtualBox:~/Desktop$ ./stef2
Xronos: 0.149374, MFLOPS/s: 428.454728
paolo@paolo-VirtualBox:~/Desktop$
```

Η διαφορά στον χρόνο είναι προφανής και βλέπουμε την αποδοτικότητα των SSE εντολών αφού ο χρόνος είναι σαφώς καλύτερος στην δεύτερη περίπτωση. Να σημειωθεί ότι το συγκεκριμένο παράδειγμα έγινε σε VM.

Δόθηκαν οι εξής τιμές στο N: 4,40,400

Για κάθε τιμή μελέταμε τον χρόνο και τα mfloaps

Και στα δύο προγράμματα δόθηκαν οι ίδιες τιμές για να γίνει καλύτερη σύγκριση των αποτελεσμάτων.

Χαρακτηριστικά :

```
paolo@paolo-VirtualBox:~/Desktop$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                 60
Model name:             Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz
Stepping:               3
CPU MHz:               2494.224
BogoMIPS:               4988.44
Hypervisor vendor:     KVM
Virtualization type:    full
L1d cache:             32K
```