ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ – ΠΑΡΑΛΛΗΛΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΕΡΓΑΣΙΑ 1

Ονοματεπώνυμο: Τασιόπουλος Μανώλης

ΑΜ : Π2015046

Διδάσκοντας : Μιχάλης Στεφανιδάκης

Αριθμός σελίδων : 6

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΓΡΑΦΗ

MATMUL-NORMAL.C

- Το "matmul-normal.c" είναι ένα πρόγραμμα το οποίο εκτελεί έναν απλό πολλαπλασιασμό πινάκων 2 διαστάσεων και εκτυπώνει το χρόνο αλλά και τα Mega Flops που απαιτήθηκαν προκειμένου να εκτελεστεί η παραπάνω πράξη με πολυπλοκότητα O(n^3).Πιο συγκεκριμένα το πρόγραμμα εκτελείται ως εξής:

- Δήλωση και δέσμευση πινάκων
  - → Δυναμική δέσμευση μεγέθους N*N με malloc()
  - → Έλεγχος δέσμευσης
  - → Έξοδος σε περίπτωση αποτυχίας

- Αρχικοποίηση των πινάκων
  - → Χρήση δεικτών στην αρχή κάθε πίνακα
  - → Αρχικοποίηση του κάθε στοιχείου
  - → Αύξηση δεικτών κάθε φορά

- Δημιουργία του φορτίου του προγράμματος
  - → Χρονοσφραγίδα στην αρχή του φορτίου
  - → Οι πίνακες αντιμετωπίζονται ως μονοδιάστατοι
  - → Υπολογισμός του εσωτερικού γινομένου κάθε στοιχείου
  - → Αποθήκευση αποτελέσματος
  - → Χρονοσφραγίδα στο τέλος του φορτίου

- Έλεγχος εξόδου του προγράμματος και αποτελέσματα
  - → Ελέγχεται κάθε στοιχείο για την εγκυρότητά του
  - → Σε περίπτωση λάθους το πρόγραμμα τερματίζει
  - → Υπολογισμός του χρόνου εκτέλεσης του φορτίου
  - → Υπολογισμός των Mega Flops
  - → Αποδέσμευση πινάκων

- Το πρόγραμμα "matmul-normal.c" δεν εκτελεί καμία μέθοδο αύξησης απόδοσης του πολλαπλασιασμού των πινάκων προκειμένου να αποτελέσει ένα μετρό σύγκρισης για το επόμενο βελτιστοποιημένο πρόγραμμα μας

MATMUL-SSE.C

- Το "matmul-sse.c" αποτελεί μια διαφορετική παραλλαγή του προηγούμενου προγράμματος. Σκοπός της εκτέλεσης του είναι η αύξηση της απόδοσης στην πράξη του πολλαπλασιασμού δυσδιάστατων πινάκων. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε εντολές SSE2 , SSE3 αλλά και τη μέθοδο όπως του loop unrolling.
Πιο συγκεκριμένα το πρόγραμμα εκτελείται ως εξής:

- Δήλωση και δέσμευση πινάκων
  - → Δέσμευση μεγέθους N*N με posix_memalign()
  - → Δέσμευση βοηθητικού πίνακα 4ων θέσεων για τα αθροίσματα των 4αδων
  - → Έλεγχος δέσμευσης
  - → Έξοδος σε περίπτωση αποτυχίας

- Αρχικοποίηση των πινάκων
  - → Χρήση δεικτών στην αρχή κάθε πίνακα
  - → Αρχικοποίηση του κάθε στοιχείου
  - → Αύξηση δεικτών κάθε φορά
  - → Σύνδεση δείκτη (__m128) με τον πίνακα των αθροισμάτων

- Δημιουργία του φορτίου του προγράμματος
  - → Χρονοσφραγίδα στην αρχή του φορτίου
  - → Οι πίνακες αντιμετωπίζονται ως μονοδιάστατοι
  - → Σύνδεση δεικτών (__m128) στους 2 πίνακες για τον υπολογισμό
  - → Χρήση των _mm_add_pc() και _mm_add_pc() ανά 4 στοιχεία
  - → Υπολογισμός του εσωτερικού γινομένου
  - → Χρήση της _mm_hadd_pc() 2 φορές για το άθροισμα των 4 στοιχείων (SSE3 εντολή)
  - → Αποθήκευση αποτελέσματος
  - → Χρονοσφραγίδα στο τέλος του φορτίου

- Έλεγχος εξόδου του προγράμματος και αποτελέσματα
  - → Ελέγχεται κάθε στοιχείο για την εγκυρότητά του
  - → Σε περίπτωση λάθους το πρόγραμμα τερματίζει
  - → Υπολογισμός του χρόνου εκτέλεσης του φορτίου
  - → Υπολογισμός των Mega Flops
  - → Αποδέσμευση πινάκων

- Το πρόγραμμα "matmul-sse.c" με τις μεθοδολογίες αυτές καταφέρνει να αυξήσει την απόδοση του δηλαδή τα Mega Flops σε σχέση με το "matmul-normal.c"

-Προκειμένου να σιγουρευτούμε ότι τα προγράμματα μας δεν κάνουν απαλοιφή των for loops τα μετατρέπουμε σε Assembly με τη χρήση του GSS και του ορίσματος -S, ελέγχοντας τις εντολές "jmp" στο φορτίο μας για επαναλαμβανόμενα κομμάτια τα οποία δηλώνουν πως υπάρχει μια επανάληψη.

ΠΙΝΑΚΑΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

-Τα προγράμματα δοκιμάστηκαν σε ένα εύρος τιμών όσο αφορά το μέγεθος των πινάκων μας και έδωσαν τα παρακάτω αποτελέσματα:

N	Mega Flops	Time
	Matmul-normal.c	
4	67.108864	0.000001
40	241.616074	0.000265
400	225.080276	0.284343
800	244.184738	2.096773

N	Mega Flops	Time
	Matmul-sse.c	
4	67.108864	0.000001
40	972.592224	0.000066
400	732.082031	0.087422
800	967.963562	0.539536

ΠΕΡΙΓΡΑΦΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

-Από τα αποτελέσματα τα οποία πήραμε μπορούμε να διακρίνουμε αρκετά μεγάλες διάφορες για τις μετρήσεις τις οποίες κάναμε.
Συγκεκριμένα παρατηρούμε:

- Όσο αυξάνεται το N και στις 2 περιπτώσεις αυξάνεται ο χρόνος και τα Mflops
- Η διαφορά στα Mega Flops είναι αρκετά εμφανής
- Το ίδιο και στο χρόνο ολοκλήρωσης του φορτίου

-Αυτές οι διάφορες που προκύπτουν οφείλονται σε αρκετούς παράγοντες. Σιγουρά το hardware παίζει σημαντικό ρολό στον καθορισμό της απόδοσης ενός προγράμματος. Βέβαια οι τιμές αυτές και η μεγάλη αυτή διαφορά τους οφείλονται στον τρόπο με τον οποίο έχει γραφτεί το πρόγραμμα.

Οι συναρτήσεις posix_memalign() και maloc() μπορεί να έχουν τα ιδιά αποτελέσματα όμως στο θέμα της απόδοσης αποδεικνύεται πως η maloc() δεν είναι τόσο αποδοτική. Το ίδιο ισχύει και για τις συναρτήσεις SSE2 _mm_add_ps() και _mm_mul_ps() σε σύγκριση με την κλασική πρόσθεση αλλά και τον πολλαπλασιασμό. Τέλος η εντολή SSE3 _mm_hadd_ps() δουλεύει αποτελεσματικότερα από ένα κλασικό for loop για το οριζόντιο άθροισμα.

HARDWARE ΑΠΟΤΕΛΕΣΜΑΤΩΝ

-Οι δοκιμές των προγραμμάτων έγιναν σε ένα εικονικό περιβάλλον kali Linux το οποίο έτρεχε live από ένα usb. Αυτός είναι και ο λόγος για τον οποίο έχουμε πάρει τόσο μικρά αποτελέσματα ως έξοδο από τα προγράμματα μας όμως η διαφορά στην απόδοση υπάρχει.

CPU

```
root@kali:~/Desktop# lscpu
Architecture:        i686
CPU op-mode(s):      32-bit
Byte Order:          Little Endian
Address sizes:       39 bits physical, 48 bits virtual
CPU(s):              1
On-line CPU(s) list: 0
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):           1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               60
Model name:          Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
Stepping:            3
CPU MHz:             3600.016
BogoMIPS:            7200.03
Hypervisor vendor:   KVM
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            256K
L3 cache:            8192K
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxs
sse sse2 ht nx rdtscp constant_tsc xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq monitor ssse3 cx
 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf_lm abm pti fsgsbase avx2 invpcid
root@kali:~/Desktop#
```

Τέλος να επισημάνω πως οι δοκιμές έγιναν με μέγιστο N = 800 διότι η έλλειψη μνήμης δεν επιτρέπεται την εμφάνιση σωστών αποτελεσμάτων για μεγαλύτερες δηλώσεις

```
root@kali:~/Desktop# gcc -Wall -O2 EX1.c -o EX1 -DN=4000
EX1.c: In function 'main':
EX1.c:79:15: warning: integer overflow in expression of type 'int' results in '-
424509440' [-Woverflow]
  mflops = (N*N*N) / (time * 1e6);
               ^
root@kali:~/Desktop#
```

```
root@kali:~/Desktop# gcc -msse3 -O Ex1_v2.c -o EX1_V2 -DN=4000
Ex1_v2.c: In function 'main':
Ex1_v2.c:110:15: warning: integer overflow in expression of type 'int' results
n '-424509440' [-Woverflow]
  mflops = (N*N*N) / (time * 1e6);
               ^
root@kali:~/Desktop#
```