

Παράλληλος προγραμματισμός 2019

Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: Δημήτριος Τριανταφύλλου

ΑΜ: Π2015077

Ι) Συνοπτική περιγραφή αρχείων κώδικα

Τα αρχεία κώδικα όπως και οι υλοποιήσεις που έγιναν είναι δύο: “matmul-normal.c” και “matmul-sse.c”. Τα βήματα που τηρείθηκαν και στα δύο προγράμματα είναι τα εξής:

- Δέσμευση πινάκων καθώς και έλεγχος δημιουργίας του.
- Αρχικοποίησή τους για να γίνει προετοιμασία στο επίπεδο της κρυφής μνήμης. (ίδιες τιμές και για τα δύο προγράμματα)
- Φορτίο, πολλαπλασιασμός πίνακα ‘a’ με ‘b’ και αποθήκευση στον ‘c’.
- Έλεγχος τιμών των αποτελεσμάτων από το προηγούμενο στάδιο (ίδιες τιμές και για τα δύο προγράμματα).
- Αποδέσμευση δεσμευμένης δυναμικής μνήμης από τους πίνακες.

Στο πρώτο αρχείο, “matmul-normal.c” δεν χρησιμοποιήθηκε καμία τεχνική βελτίωσης και ο πολλαπλασιασμός των πινάκων έγινε κανονικά, γραμμή επί στήλη και το άθροισμά τους υπολογίστηκε στοιχείο προς στοιχείο. Για ευνοϊκή προσπέλαση στην κρυφή μνήμη χρησιμοποιήθηκε ένα τριπλό for loop ώστε να γίνει αντιμετάθεση των γραμμών και των στηλών στον πίνακα ‘b’ για τον πολλαπλασιασμό και η προσπέλαση των δεδομένων έγινε με τη χρήση δεικτών. Όταν το πρόγραμμα ολοκληρωθεί με επιτυχία εμφανίζεται ο χρόνος που χρειάστηκε να υπολογιστεί το φορτίο καθώς η απόδοση του σε μονάδα μέτρηση flops και συγκεκριμένα mega-flops.

Στο δεύτερο αρχείο, “matmul-sse.c”, χρησιμοποιήθηκαν τεχνικές βελτιστοποίησης όπως το loop-unrolling μαζί με το σύνολο εντολών SSE για τον ζητούμενο υπολογισμό. Η αποτελεσματικότητα και γενική ιδέα βημάτων παραμένει ίδια με το προηγούμενο πρόγραμμα. Μία διαφορά είναι η δέσμευση της μνήμης με τη συνάρτηση `posix_memalign` ώστε να έχουμε καλύτερη, εγγυημένη και ορισμένη από το χρήστη ευθυγράμμιση, στη μνήμη, σε σχέση με τη συνάρτηση `malloc`. Οι άλλες διαφορές, που έχουν προαναφερθεί, είναι η τεχνική του loop-unrolling, ανά τετράδες, όπου επιτυγχάνουμε ταχύτητα εκτέλεσης στο πρόγραμμα αυξάνοντας ελάχιστα το μέγεθος της μνήμης που καταλαμβάνει το διαδυκό αρχείο και οι εντολές SSE όπου βοηθούν στην επεξεργασία (πρόσθεση, πολλαπλασιασμό, ...) πολλαπλών δεδομένων (τετράδων) και έχουν σαν αποτέλεσμα καλύτερη απόδοση.

Για να βεβαιωθώ ότι το πρόγραμμα δεν έκανε απαλοιφή τα βασικά for loops (αυτά στην περιοχή του φορτίου) χρειάστηκε να ελέγξω τα αρχεία του κώδικα που προκύπτουν από τον Assembler για το κάθε πρόγραμμα. Αυτό επιτυγχάνεται με τις ακόλουθες εντολές και για τα δύο αρχεία: “gcc -S matmul-normal.c -S -DN=?” και “gcc -O2 matmul-sse.c -S -DN=?”, με το N να είναι το μέγεθος των πινάκων. Σε αυτά ελέγχουμε τις εντολές τύπου `jmp` στην περιοχή που εκτελούνται οι εντολές του φορτίου, για την ανίχνευση επαναλήψεων τμημάτων στο πρόγραμμα.

II) Πίνακες αποτελεσμάτων

Στις δοκιμές το N που επιλέχθηκε διαιρείται με τον αριθμό τέσσερα. Οι εκτελέσεις των προγραμμάτων δεν έγιναν τοπικά σε υπολογιστή με λειτουργικό σύστημα Linux, αλλά με Windows. Παρόλα αυτά, ο μεταγλωτιστής που χρησιμοποιήθηκε ήταν ο gcc της GNU με τη χρήση του λογισμικού Cygwin (4).

Μέγεθος πινάκων (N)	Χρόνος Εκτέλεσης (secs)	Πράξεις κινητής υποδιαστολής το δευτερόλεπτο (megaflops/sec)
4	0.000000	inf
20	0.000005	1597.830078
40	0.000061	1048.576050
200	0.009097	879.401184
400	0.083340	767.939209
2000	11.332314	705.945862
4000	91.514610	699.341858

Table 1: Αποτελέσματα δοκιμών του "matmul-normal.c"

Μέγεθος πινάκων (N)	Χρόνος Εκτέλεσης (secs)	Πράξεις κινητής υποδιαστολής το δευτερόλεπτο (megaflops/sec)
4	0.000000	inf
20	0.000003	2796.202637
40	0.000014	4628.197754
200	0.001814	4410.414062
400	0.018153	3525.597412
2000	3.108613	2573.494873
4000	24.979746	2562.075684

Table 2: Αποτελέσματα δοκιμών του "matmul-sse.c"

III) Εξήγηση των αποτελεσμάτων

Από τα παραπάνω αποτελέσματα είναι φανερό το γεγονός πως το πρόγραμμα που χρησιμοποιεί εντολές SSE είναι αποδοτικότερο από αυτό χωρίς αυτές. Όσο αυξάνεται το μέγεθος των πινάκων (N) και στις δύο περιπτώσεις μειώνονται τα megflops. Χρονικά, φαίνεται να διαφέρουν εκθετικά. Επιπλέον, μπορεί να παρατηρηθεί πως μετά από ένα μέγεθος του N, στη συγκεκριμένη περίπτωση περίπου μετά από N=400 και N=2000, φαίνεται τα megaflops να συγκλίνουν μεταξύ τους και η διαφορά να είναι μικρής τάξης μεγέθους.

Στον παρακάτω πίνακα (Table 3) παρατίθενται τα βασικά χαρακτηριστικά του επεξεργαστή όπου έγιναν οι προηγούμενες δοκιμές. Για περισσότερες πληροφορίες αυτού βρίσκονται στο σύνδεσμο (3) που αναφέρονται από τον κατασκευαστή.

Product Collection	7th Generation Intel® Core™ i5 Processors
Code Name	Products formerly Kaby Lake
Vertical Segment	Mobile
Processor Number	i5-7200U
Lithography	14 nm
Number of Cores	2
Number of Threads	4
Processor Base Frequency	2.50 GHz
Max Turbo Frequency	3.10 GHz
Cache	3 MB SmartCache
Instruction Set	64-bit
Instruction Set Extensions	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2

Table 3: Βασικά χαρακτηριστικά αρχιτεκτονικής επεξεργαστή (i5-7200u)

IV) Πηγές - Βιβλιογραφία

- 1) Intel Intrinsics Guide: <https://software.intel.com/sites/landingpage/IntrinsicsGuide/>
- 2) <https://inf.ethz.ch/personal/markusp/teaching/263-2300-ETH-spring11/slides/class02.pdf>
- 3) <https://ark.intel.com/content/www/us/en/ark/products/95443/intel-core-i5-7200u-processor-3m-cache-up-to-3-10-ghz.html>
- 4) Cygwin: <https://sourceware.org/cygwin/>