

Παράλληλος Προγραμματισμός 2019
Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: Μαρία Νεφέλη Νικηφόρου
ΑΜ: Π2015054



- Συνοπτική περιγραφή των δύο παραλλαγών του κώδικα

Η πρώτη παραλλαγή του κώδικα υλοποιείται στο αρχείο "matmul-normal.c". Εκτελείται πολλαπλασιασμός δύο float πινάκων (a και b), διαστάσεων $N \times N$, με το αποτέλεσμα αποθηκεύεται σε τρίτο πίνακα (c) διαστάσεων $N \times N$, χωρίς τη χρήση εντολών SSE2. Αυτό επιτυγχάνεται με τη χρήση κατάλληλων δεικτών για την προσπέλαση των δεδομένων, τύπου `float *`. Η χρήση τριπλού `for loop` οδηγεί στην αντιμετάθεση γραμμών-στηλών του πίνακα b , επιτρέποντας έτσι την αξιοποίηση της αρχής της τοπικότητας της κρυφής μνήμης, ώστε να αυξηθεί η απόδοση.

Η δεύτερη παραλλαγή του κώδικα υλοποιείται στο αρχείο "matmul-sse.c". Εκτελείται πολλαπλασιασμός δύο float πινάκων (a και b), διαστάσεων $N \times N$, με το αποτέλεσμα αποθηκεύεται σε τρίτο πίνακα (c) διαστάσεων $N \times N$, με τη χρήση εντολών SSE2 ανά τετράδες. Υλοποιείται, επιπλέον, ένας float πίνακας (sum) 4 θέσεων για την άθροιση των αποτελεσμάτων της τελικής τετράδας που προκύπτει σε έναν μόνο αριθμό, πριν την αποθήκευση του τελικού αποτελέσματος στον c . Χρησιμοποιούνται κατάλληλοι δείκτες για την προσπέλαση των δεδομένων, τύπου `float *` και `__m128 *`. Οι δείκτες fc και $fsum$ (τύπου `float *`) δημιουργήθηκαν έχοντας βοηθητικό ρόλο, για την αποφυγή σφαλμάτων προσπέλασης εκτός μνήμης (segmentation fault και aborted (core dumped)) κατά την άθροιση των αποτελεσμάτων της τελικής τετράδας που προκύπτει σε έναν μόνο αριθμό (τέταρτο `for loop`). Με την τεχνική `loop unrolling` επιλέχθηκαν οι κατάλληλες εντολές SSE2 για τον υπολογισμό του γινομένου ανά τετράδες.

Η μέτρηση της απόδοσης και για τις δύο παραλλαγές γίνεται με τη χρήση της συνάρτησης `get_walltime()`, με τη λήψη χρονικών τιμών αρχής και τέλους αμέσως πριν και αμέσως μετά το φορτίο, αντίστοιχα. Ο υπολογισμός των $Mflops/s$ εκτελείται από την εντολή: `mflop = (N*N*N)/((te-ts) * 1e6);`, όπου ο αριθμητής προκύπτει από τον αριθμό των προσπελάσεων. Για τον έλεγχο σχετικά με το αν ο μεταγλωττιστής απαλείφει τα `loops` (μεταξύ των κλήσεων της `get_walltime()`), χρησιμοποιήθηκαν τα αρχεία σε Assembly που προκύπτουν για το κάθε αρχείο, με εντολή τύπου: `gcc -O2 matmul-....c -S -DN=...`

- Πίνακας αποτελεσμάτων για τα διάφορα μεγέθη πίνακα και για τις δύο παραλλαγές κώδικα

N	1η παραλλαγή		2η παραλλαγή (SSE2)	
	Διάρκεια εκτέλεσης	Mflops/s	Διάρκεια εκτέλεσης	Mflops/s
4	0.000002	33.554432	0.000001	67.108864
40	0.000313	204.444369	0.000048	1335.499781
400	0.351280	182.190858	0.139724	458.045809
1000	4.721647	211.790503	1.338104	747.326062

Σημειώνεται πως οι εκτελέσεις των δύο παραλλαγών έγιναν σε εικονική μηχανή με λειτουργικό σύστημα Linux. Οι μετρήσεις έγιναν για τιμές του N: 4, 40, 400 και 1000. Δυστυχώς, στο συγκεκριμένο σύστημα, τα συγκεκριμένα προγράμματα δεν μπορούν να εκτελεστούν ομαλά για τιμές του N (πολ/σιες του 4) πάνω από 1000.

- *Εξήγηση των αποτελεσμάτων*

Μεταξύ των δύο παραλλαγών παρατηρείται πως η δεύτερη είναι πιο αποδοτική, καθώς για την ίδια τιμή του N , προκύπτει μικρότερη χρονική διάρκεια εκτέλεσης αλλά και περισσότερα *Mflops/s* σε σύγκριση με την πρώτη. Αυτό συμβαίνει διότι με τη χρήση εντολών SSE2 η προσπέλαση γίνεται ανά τετράδες στοιχείων, αντί για στοιχείο - στοιχείο. Έτσι, αξιοποιείται η αρχή της τοπικότητας της κρυφής μνήμης, με αποτέλεσμα να χρειάζονται συνολικά λιγότερες προσπελάσεις της κύριας μνήμης.

Τέλος, παρατηρείται πως και στις δύο παραλλαγές ο αριθμός των *Mflops/s* για N : 400 μειώνεται σημαντικά σε σχέση με τις προηγούμενες και τις επόμενες τιμές του N .

- *Αναφορά σε πηγές που πιθανόν χρησιμοποιήσατε*

The Intel Intrinsic Guide, <https://software.intel.com/sites/landingpage/IntrinsicGuide/>