Παράλληλος Προγραμματισμος 2019
Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: Ανδρεάνα Στυλίδου
ΑΜ: Π2015180

Αναφορά για την Υλοποίηση Πειραμάτων Υπολογισμού Απόδοσης Κρυφής Μνήμης

Παραλλαγή προγράμματος χωρίς SSE:

Στη πρώτη παραλλαγή γίνεται δέσμευση 3 συνολικά πινάκων διαστάσεων N*N (τύπου *float*) σε συνεχόμενες θέσεις μνήμης. Συγκεκριμένα, δύο πίνακες N*N (a και b) που πολλαπλασιάζονται μεταξύ τους και ένας (c) για την αποθήκευση των αποτελεσμάτων. Προσπελαύνονται έπειτα όλοι οι πίνακες για τον σκοπό της ανάθεσης αρχικών τιμών και εισάγονται έτσι στη κρυφή μνήμη (cache warmup). Κάνουμε την υπόθεση ότι ο πίνακας b είναι transposed, είναι ανεστραμμένες δηλαδή οι στήλες και οι γραμμές του, έτσι ώστε να επιτευχθεί η ευνοϊκή προσπέλαση κρυφής μνήμης.

Με τη βοήθεια δεικτών για κάθε έναν από τους πίνακες (**pa*, **pb* και **pc*), φτιάχνουμε τα τρία loops του πολλαπλασιασμού των πινάκων. Συγκεκριμένα, στο πρώτο loop, ορίζεται η εκάστοτε στήλη του πίνακα b, στο δεύτερο η γραμμή του a και στο τρίτο υπολογίζεται το εσωτερικό γινόμενο των πινάκων. Το αποτέλεσμα αποθηκεύεται στις θέσεις του c, οι οποίες προσπελαύνονται συνεχόμενα.

Με τη χρήση της *get_walltime(),* παίρνουμε μια μέτρηση χρόνου στη αρχή και μία στο τέλος του φορτίου. Με αυτόν τον τρόπο υπολογίζουμε τη συνολική διάρκεια του πειράματος. Η ταχύτητα απόδοσης υπολογίζεται σε *mflops*, διαιρώντας τη πολυπλοκότητα $o(n^3)$ (3 επαναλήψεις N*N), με το χρόνο εκτέλεσης.

Παραλλαγή προγράμματος με εντολές SSE κ SSE2:

Στη δεύτερη παραλλαγή δεσμεύονται οι ίδιοι πίνακες, με τη διαφορά ότι χρησιμοποιείται η *posix_memalign()* για ευθυγράμμιση των πινάκων στη μνήμη (16 bytes).

Οι πίνακες προσπελαύνονται με δείκτες τύπου __m128, έτσι ώστε να μπορέσουμε να χρησιμοποιήσουμε εντολές SSE για άθροιση ανά τετράδα. Χρησιμοποιείται μια βοηθητική temp μεταβλητή, στην οποία αποθηκεύονται προσωρινά τα αποτελέσματα των αθροισμάτων των γινομένων των τετράδων. Με χρήση της *mm_hadd_ps()* (SSE2) γίνεται οριζόντια άθροιση των 4 τιμών (όπως στο παράδειγμα sse-haddps.c) και το αποτέλεσμα τελικά εισάγεται στην κατάλληλη θέση του c.

Ο χρόνος εκτέλεσης και η απόδοση υπολογίζεται με τον ίδιο τρόπο.

Πίνακας αποτελεσμάτων

	N = 4	**N = 40**	**N = 400**	**N = 4000**
No-SSE	53.687091	571.139268	516.220045	512.841224
SSE	67.108864	2556.528152	2120.393501	2172.282424

Συμπέρασμα

Από τα αποτελέσματα διακρίνουμε την καλύτερη απόδοση των προγραμμάτων που εκμεταλλεύονται την αρχιτεκτονική των κρυφών μνημών (αρχή της τοπικότητας). Η δεύτερη παραλλαγή χρησιμοποιώντας τα intrinsics εκτελεί λιγότερες επαναλήψεις, κάνοντας πράξεις σε 4-άδες. Αυτό μειώνει σημαντικά τον χρόνο εκτέλεσης (ειδικά για μεγάλα N) και επιτρέπει την ταχύτερη προσπέλαση της μνήμης των υπολογιστών.