Παράλληλος Προγραμματισμός 2019
Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: Αστέριος Παπαμιχαήλ
ΑΜ: Π2015059

Περιγραφή κώδικα

Απλός Πολλαπλασιασμός πινάκων

Αρχικά γίνεται δέσμευση των πινάκων χρησιμοποιώντας την βοηθητική συνάρτηση `alocate_mem` και σε περίπτωση μη επιτυχημένης δεσμευσης σταματάει την ροή του προγράμματος. Στην συνέχεια γίνεται αρχικοποίηση των τριών πινάκων. Αφού γίνει η αρχικοποίηση γίνεται η πρώτη μέτρηση χρόνου `ts`. Για να γίνει πολλαπλασιασμός των πινάκων έχουμε ένα `for loop # 1` το οποίο είναι υπεύθυνο για την γραμμή του `a` καθώς και για την επαναφορά του `pb` στην αρχική του θέση. Το `for loop # 2` είναι υπεύθυνο για την επαναφορά του pointer που δείχνει στο στοιχείο του πίνακα `a` καθώς και για την αύξηση του pointer που δείχνει στο αποτέλεσμα. Και το εσωτερικο `for loop # 3` είναι που γινεται και ο πολλαπλασιασμός των στοιχείων της γραμμής του `a` με αυτή του `b`. Αφού τελειώσει η διαδικασία του πολλαπλασιασμού παίρνουμε και τον δεύτερο χρόνο `te`. Στην συνέχεια πραγματοποιείται και ο έλεγχος του αποτελέσματος και τέλος απελευθερώνεται η μνήμη η οποία δεσμεύτηκε με την συνάρτηση `malloc`.

Πολλαπλασιασμός πινάκων χρησιμοποιώντας τις εντολές "sse2"

Αρχικά γίνεται δέσμευση μνήμης χρησιμοποιώντας την εντολή `posix_memalign` που παίρνει ως είσοδο τον pointer του πίνακα που θέλουμε να δεσμεύσουμε, το μέγεθος της στοίχισης καθώς και το μέγεθος του πίνακα. Έπειτα γίνεται αρχικοποίηση χρησιμοποιώντας την βοηθητική συνάρτηση `set_array` που παίρνει ως είσοδο τον pointer του πίνακα, την τιμή που θέλουμε να έχει ο πίνακας, καθώς και το μέγεθος του πίνακα. Στην συνέχεια ακολουθεί η διαδικασία του πολλαπλασιασμού των πινάκων έτσι όπως γίνεται και παραπάνω με μία διαφορά, ότι για τον πολλαπλασιασμό και για την πρόσθεση χρησιμοποιούνται συναρτήσεις `sse2` που εκμεταλλεύονται το hardware του επεξεργαστή.

Πολλαπλασιασμός πινάκων χρησιμοποιώντας τις εντολές "sse2" και loop unrolling

Ο κώδικας δεν άλλαξε πολύ απλά αντί να έχω ενάν `__m128` pointer για κάθε πίνακα έχω έναν πίνακα από pointers `__m128` που δείχνουν σε συνεχόμενες τετράδες.

Αποτελέσματα

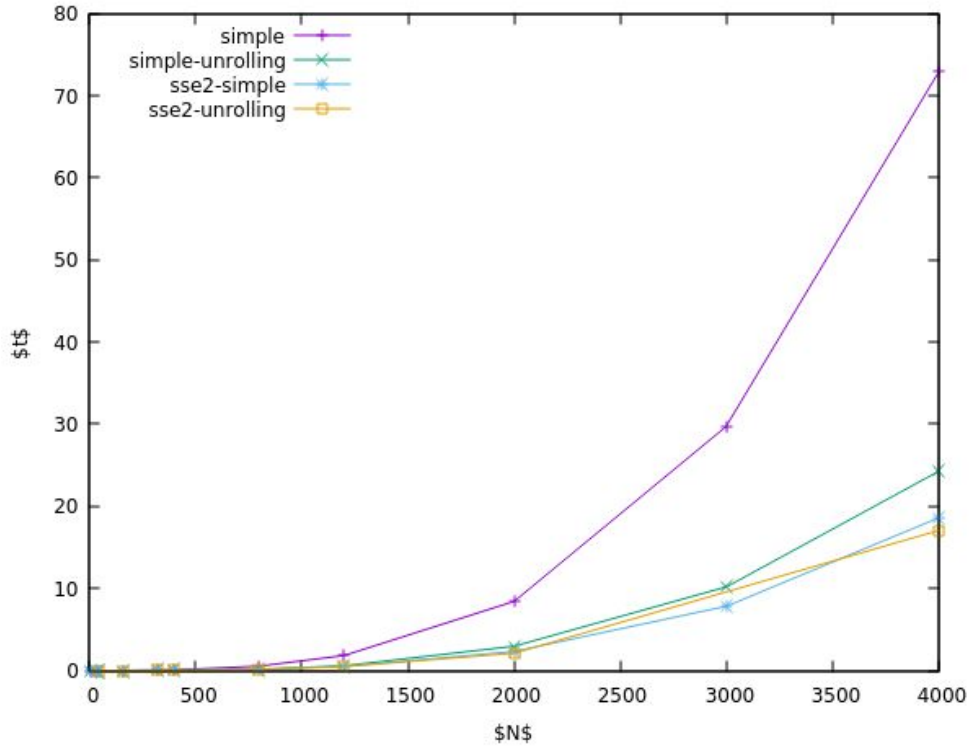
Ακολουθεί συγκεντρωτικός πίνακας των χρόνων.

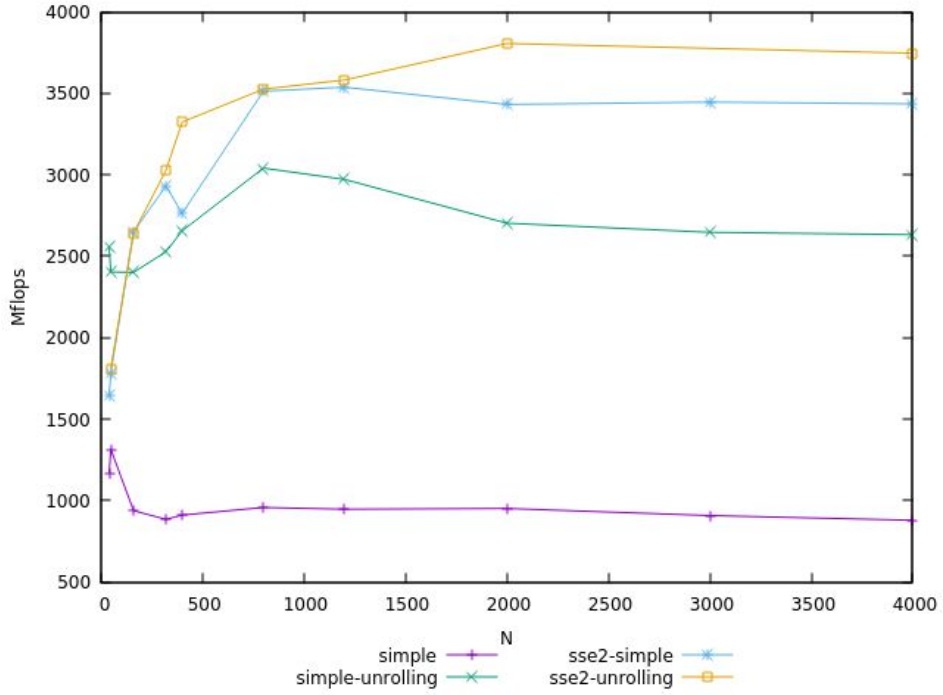
N	Simple	Simple-unrolling	SSE2-simple	SSE2-unrolling
4	0	0	0	
40	0,000055	0,000025	0,000039	
48	0,000084	0,000046	0,000062	0,000061
160	0,004372	0,001706	0,001545	0,001549
320	0,037073	0,012966	0,01118	0,010823
400	0,070324	0,024102	0,02316	0,019252
800	0,535246	0,168348	0,145736	0,145199
1200	1,826081	0,581548	0,488481	0,482416
2000	8,420488	2,959728	2,331128	2,10194
3000	29,779496	10,202712	7,83432	
4000	72,905873	24,313478	18,627765	17,081156

Ακολουθεί συγκεντρωτικός πίνακας των ***Mflops*.**

N	Simple	Simple-unrolling	SSE2-simple	SSE2-unrolling
4	inf	inf	inf	
40	1167.110678	2556.528152	1646.843288	
48	1314.040986	2403.401388	1784.063338	1811.939328
160	936.896394	2400.764279	2650.805305	2644.277233
320	883.880211	2527.194643	2930.968043	3027.623163
400	910.074098	2655.358050	2763.387441	3324.319261
800	956.569690	3041.317836	3513.202459	3526.193616
1200	946.288782	2971.379959	3537.496528	3581.971382
2000	950.063715	2702.951080	3431.814807	3806.007503
3000	906.664103	2646.355189	3446.374489	
4000	877.844230	2632.284859	3435.731566	3746.819006

Ακολουθούν και τα αντίστοιχα διαγράμματα των αποτελεσμάτων με την ίδια σειρά.





Όπως παρατηρείται από τα παραπάνω διαγράμματα το loop unrolling βελτιώνει την απόδοση του προγράμματος κατά πολύ ακόμα και στην περίπτωση του απλού πολλαπλασιασμού. Αυτό συμβαίνει γιατί το πρόγραμμα “προσπαθεί” να τρέξει τον πολλαπλασιασμό σε hardware level.