Επεξήγηση Κώδικα

Για το πρώτο πρόγραμμα έγινε δέσμευση των δεδομένων οι θέσεις που δεσμεύτηκαν είναι N * N , επίσης γίνονται έλεγχοι για την επιτυχή δέσμευση της μνήμης. Σε περίπτωση αποτυχία έχουμε απότομο τερματισμό του προγράμματός μας και αποδέσμευση όσης μνήμης δεσμεύτηκε. Όσο αφορά την τριπλέτα τον for loops:

Η σημασία που εξυπηρετούν είναι η εξής : το πρώτο loop αναλαμβάνει να κάνει τον pointer b (pb) να δείχνει στην αρχή του πίνακα b γιατί άμα δεν γίνει επαναφορά του στις επόμενες επαναλήψεις θα οδηγηθούμε σε λάθος προσπέλαση μνήμης και έτσι θα προκύψουν λάθος αποτελέσματα, σε περίπτωση που η εκτέλεση του προγράμματος συνεχιστεί κανονικά. Επίσης το πρώτο loop δείχνει την γραμμή στην οποία θα ξεκινήσει ο δείκτης για τον πίνακα A. Στο δεύτερο loop ο δείκτης A δείχνει στην γραμμή η οποία καθορίζεται από το κ δηλαδή το πρώτο loop στην συνέχεια στο τρίτο loop γίνονται οι κατάλληλες πράξεις για να υπολογιστεί ο πολλαπλασιασμός μεταξύ των πινάκων A και B. Το αποτελέσματα του dot product δίνεται στην κατάλληλη θέση του πίνακα c . Τέλος γίνονται οι κατάλληλοι έλεγχοι για να δούμε άμα τα αποτελέσματα του πίνακα έχουν πάρει τις σωστές αναμενόμενες τιμές, επίσης με αυτόν τον τρόπο προσπαθούμε να κάνουμε τον compiler να μην κάνει βελτιστοποιήσεις στον κώδικα μας, στο τέλος του προγράμματος μας γίνετε αποδέσμευση της μνήμης

Στον κώδικα ο οποίος υλοποιεί τα sse :

Ακολουθεί παρόμοια φιλοσοφία με το πρώτη υλοποίηση με τις εξής βασικές διαφορές:

Για την δέσμευση της μνήμης χρησιμοποιήθηκε η posix_memallign για την οποία οι δεσμευμένες θέσεις μνήμης θα έχουμε σίγουρη ευθυγράμμιση

Σε αντίθεση με το προηγούμενο κώδικα που είχαμε δείκτες σε floats τώρα έχουμε δείκτες σε τετράδες για τους πίνακες A και B. Εφόσον έχουμε δείκτες σε τετράδες στο τρίτο for loop το βήμα γίνεται με 4 καθώς γίνονται τέσσερις πράξεις ταυτόχρονα. Οι ταυτόχρονες πράξεις αυτές στην περίπτωσή μας γίνονται με την συνάρτηση mm_mul_ps τα αποτελέσματα αυτά αθροίζονται σε μια μεταβλητή τετράδας sum η οποία αρχικοποείται με την συνάρτηση mm_set_ps με μηδενικές τιμές σε κάθε δεύτερο for loop αξιοσημείωτο είναι ότι η αρχικοποίηση της τετράδας με την συνάρτηση αυτή γίνετε με την αντίστροφη σειρά με την οποία δηλώθηκαν οι τιμές. Τέλος το αποτέλεσμα του αθροίσματος αυτός είναι έναν πίνακας με τέσσερεις τιμές . Προκειμένου να πάρουμε μια τιμή , απλά αθροίζουμε τις τέσσερεις τιμές χρησιμοποιώντας τον τελεστή για την πρόσβαση στα στοιχεία ενός πίνακα από την θέση 0-3.

Αποτελέσματα:

Αρχικά μπορούμε να διαπιστώσουμε από τον πίνακα τον αποτελεσμάτων ότι η μέθοδος με τα SSE σημείωσε μεγαλύτερη επίδοση από την αντίστοιχη υλοποίηση χωρίς την χρήση sse.

Αρχικά το τρίτο Loop εκτελείτε λιγότερες φορές λόγο του loop unrolling, επιπρόσθετα οι εντολές με τις τετράδες εκτελούνται παράλληλα αυξάνοντας ακόμα περισσότερο την απόδοση.

Επίσης κάτι το οποίο παρουσιάζει ενδιαφέρον είναι ότι παρατηρούμε από τον πίνακα των αποτελεσμάτων πως η απόδοση του προγράμματος μας δεν παρουσιάζει κάποια ιδιαίτερη πρόοδο όσο μεγαλώνουμε το μέγεθος του πίνακα κάτι το οποίο μπορεί να μας οδηγεί στο συμπέρασμα πως ο επεξεργαστής δεν μπόρεσε να σημειώσει κάποια μεγάλη πρόοδο. Και ίσως θα πρέπει να ανευρεθεί διαφορετικός τρόπος υλοποίησης ο οποίος θα μειώσει και την πολυπλοκότητα του αλγόριθμου

Μέγεθος Πίνακα	Χρόνος	Mflops
4	0.000001	67.108864
40	0.000175	365.715880
400	0.090749	705.241734
800	0.4952554	1033.813367
1200	1.622937	1064.736363
1600	3.3824830	1070.897253
2000	7.405691	1080.250324
2400	12.763410	1083.096144
3400	24.833447	1087.243347
4000	58.908587	1086.429047

Πίνακας αποτελεσμάτων χωρίς SSE

Μέγεθος Πίνακα	Χρόνος	Mflops
4	0.0	inf
40	0.000050	1278.264076
400	0.042845	1493.756189
800	0.211659	2418.985773
1200	0.498807	3464.266061
1600	1.409531	2905.931373
2000	2.981403	2683.300560
2400	5.022734	2752.285947
3400	14.550608	2701.192938
4000	23.589666	2713.052387

Πίνακας αποτελεσμάτων με SSE

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 60
Model name:            Intel(R) Core(TM) i5-4460  CPU @ 3.20GHz
Stepping:              3
CPU MHz:               800.000
CPU max MHz:           3400,0000
CPU min MHz:           800,0000
BogoMIPS:              6396.75
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              6144K
NUMA node0 CPU(s):     0-3
```