

Παράλληλος Προγραμματισμός 2018

Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: Ηρώ Σπύρου

ΑΜ: Π2014120

Συνοπτική περιγραφή κώδικα

Στον κώδικα των αρχείων matrix1.c και matrix2.c γίνεται δέσμευση συνεχόμενων θέσεων μνήμης για την υλοποίηση ενός πίνακα δύο διαστάσεων, μεγέθους $NROWS * NCOLS * 8$. Στη συνέχεια αρχικοποιούμε όλες τις θέσεις του πίνακα στην τιμή 1.0 και το sum στην τιμή 0.0. Έπειτα λαμβάνουμε τον χρόνο εκκίνησης, προσπελάζουμε τον πίνακα με την χρήση δύο loops, μέσα στα οποία υπολογίζεται ένα άθροισμα (sum) ώστε να μην γίνεται απαλοιφή των loops, και ύστερα λαμβάνουμε τον χρόνο τερματισμού της διαδικασίας αυτής. Η προσπέλαση του πίνακα γίνεται γραμμή-προς-γραμμή στο matrix1.c, ενώ στο matrix2.c γίνεται στήλη-προς-στήλη. Στη συνέχεια γίνεται ένας έλεγχος του αποτελέσματος για να βεβαιωθούμε ότι έχουν εκτελεστεί όλα τα loops. Τέλος υπολογίζουμε τον χρόνο εκτέλεσης που χρειάστηκε, καθώς και τα Maccesses/sec, και τα τυπώνουμε.

Πίνακας και εξήγηση αποτελεσμάτων

	Matrix1	Matrix2
100	Time: 0,000018 Maccesses/sec: 1124,552925	Time: 0,000021 Maccesses/sec: 1007,390488
1.000	Time: 0,000347 Maccesses/sec: 732,829788	Time: 0,000845 Maccesses/sec: 536,076367
10.000	Time: 0,004034 Maccesses/sec: 760,521346	Time: 0,011167 Maccesses/sec: 244,844603
100.000	Time: 0,026773 Maccesses/sec: 836,007720	Time: 0,115034 Maccesses/sec: 187,432995

Παρατηρώντας τον πίνακα βλέπουμε ότι στο πείραμα με 100 και στο πείραμα με 1.000 γραμμές ο κώδικας του matrix2.c εκτελέστηκε ελαφρώς πιο αργά από εκείνον του matrix1.c. Όμως, στο πείραμα με 10.000 και στο πείραμα με 100.000 γραμμές παρατηρούμε ότι ο χρόνος εκτέλεσης είναι μεγαλύτερος κατά μία τάξη μεγέθους στον κώδικα του matrix2.c από εκείνον στον matrix1.c.

Αυτή η χρονική διαφορά οφείλεται στο ότι η γλώσσα προγραμματισμού C είναι row-major, το οποίο σημαίνει ότι με κάθε κλήση για επιστροφή μίας τιμής του πίνακα το σύστημα επιστρέφει την τιμή που ζητήσαμε συν τις επόμενες 7 στην γραμμή, αφού έχουμε τιμές τύπου double, οι οποίες αποθηκεύονται στην cache. Αυτό σημαίνει ότι η προσπέλαση γραμμή-προς-γραμμή θα είναι πιο αποδοτική, καθώς θα μειωθούν οι αιτήσεις προς την κύρια μνήμη εφόσον πολλές από τις τιμές θα βρίσκονται ήδη στην cache. Στην προσπέλαση στήλη-προς-στήλη, όμως, θα έχουμε $NROWS * NCOLS$ αιτήσεις προς την κύρια μνήμη, καθώς οι cache δεν περιέχει τις τιμές που απαιτεί το πρόγραμμά μας.

Πηγές

1. https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Row-major_order.html
2. <https://stackoverflow.com/questions/9936132/why-does-the-order-of-the-loops-affect-performance-when-iterating-over-a-2d-array>