Παράλληλος Προγραμματισμός

Εργασία #1

Δήμα Μαρία, ΑΜ.: Π2013029

Ερώτημα b)

Εκτέλεση των 2 εναλλακτικών προγραμμάτων (προσπέλαση ανά γραμμή και ανά στήλη) με διαφορετικά μεγέθη πίνακα (πλήθος γραμμών - NROWS):

	MACCESSES/SEC		TIME(SEC)	
NROWS	Ανά ΓΡΑΜΜΗ	Ανά ΣΤΗΛΗ	Ανά ΓΡΑΜΜΗ	Ανά ΣΤΗΛΗ
100,000	666	68.9	0.0150	0.1450
50,000	1000	100	0.0050	0.0500
20,000	400	200	0.0050	0.0100
10,000	INF	200	0.0000	0.0050
5,000	INF	100	0.0000	0.0050
2,000	INF	INF	0.0000	0.0000

Η προσπέλαση ανά στήλη έχει σημαντικά χαμηλότερη απόδοση (από 2 έως και 10 φορές χειρότερη) και αυτό οφείλεται στο ότι η αποθήκευση των δεδομένων των πινάκων στην μνήμη γίνεται σε συνεχόμενες θέσης όπως βρίσκονται τα δεδομένα στον πίνακα ανά γραμμή. Συνεπώς στην ανά γραμμή προσπέλαση τα δεδομένα διαβάζονται ή γράφονται με τη σειρά που είναι αποθηκευμένα στην μνήμη. Ενώ στην ανά στήλη προσπέλαση τα δεδομένα προσπελαύνονται συνεχώς από απομακρυσμένες μεταξύ τους θέσεις μνήμης. Αυτό επιβαρύνει τη λειτουργία της τοπικής μνήμης (cache) στην οποία έρχονται block δεδομένων από την βασική μνήμη και άρα οι απομακρυσμένες προσπελάσεις δεν είναι σίγουρο ότι θα βρίσκονται μέσα στο block των δεδομένων που έχουν έρθει ήδη στην cache. Αντίθετα, συνεχόμενα δεδομένα διαδοχικών προσπελάσεων (στην περίπτωση προσπέλασης ανά γραμμή) βρίσκονται ήδη στην cache, ανάλογα βέβαια και με το μέγεθος της διαθέσιμης cache και του block των δεδομένων που αντιγράφεται σε αυτή.

Μειώνοντας το μέγεθος του πίνακα (τον αριθμό των γραμμών του) μειώνεται και ο χρόνος που απαιτεί η προσπέλαση όλων των στοιχείων του πίνακα. Για πίνακα 10.000 γραμμών και κάτω, στην περίπτωση της προσπέλασης των στοιχείων ανά γραμμή, ο χρόνος είναι μηδενικός (με μετρίσιμος) γιατί ο συγκεκριμένος πίνακας καταλαμβάνει 10.000γραμμές x 100στήλες x8bytes (λόγω double) = 8MB και χωράει ολόκληρος μέσα στην cache του επεξεργαστή (που για τον υπολογιστή που έγινε η εργασία είναι μεγέθους 8MB). Και ο πίνακας βρίσκεται ήδη στην cache από τη διαδικασία της αρχικοποίησης του που προηγείται της έναρξης της μέτρησης του χρόνου. Συνεπώς ο χρόνος εκτέλεσης είναι

μικρότερος από τα μsec που μετράει ο κώδικάς μας (πολύ υψηλή ταχύτητα λειτουργίας της cache σε σχέση με την βασική μνήμη). Το ίδιο συμβαίνει και στην περίπτωση της προσπέλασης ανά στήλη αλλά για μικρότερο μέγεθος πίνακα (από 2.000 γραμμές και κάτω), καθώς οι διάσπαρτες προσπελάσεις των στοιχείων του πίνακα στην μνήμη δεν βοηθούν την αποδοτική λειτουργία της cache.

Για να πάρουμε μετρήσεις για ακόμα μικρότερα μεγέθη πινάκων, ταυτόχρονα με την εκτέλεση του προγράμματος βάλαμε να εκτελούνται κι άλλα προγράμματα που απασχολούσαν τον επεξεργαστή και χρησιμοποιούσαν μέρος της cache (και άρα δεν είναι διαθέσιμα και τα 8MB της). Σε αυτή τη περίπτωση το μέγεθος του πίνακα για το οποίο το πρόγραμμα μας μετρούσε μηδενικό χρόνο ήταν οι 500 γραμμές:

	MACCESSES/SEC		TIME(SEC)	
NROWS	Ανά ΓΡΑΜΜΗ	Ανά ΣΤΗΛΗ	Ανά ΓΡΑΜΜΗ	Ανά ΣΤΗΛΗ
100,000	640	61.5	0.0155	0.1620
50,000	713	73	0.0070	0.0680
20,000	665	190	0.0030	0.0105
10,000	665	222	0.0015	0.0045
5,000	995	250	0.0005	0.0020
2,000	400	400	0.0005	0.0005
1,000	INF	200	0.0000	0.0005
500	INF	INF	0.0000	0.0000

Οι περιπτώσεις που δεν χωράει ολόκληρος ο πίνακας στην cache έχουν παρόμοιες αποδόσεις με την περίπτωση που ήταν διαθέσιμη ολόκληρη η cache για την συγκεκριμένη εφαρμογή και εξακολουθεί να υπάρχει η ίδια διαφορά απόδοσης μεταξύ των προσπελάσεων των στοιχείων του πίνακα ανά γραμμή και ανά στήλη.

Για να γίνει πιο εμφανής η επίπτωση στη λειτουργία της ιεραρχίας της μνήμης του υπολογιστή (βασική μνήμη και cache) θα μπορούσαμε να υλοποιήσουμε μεγαλύτερο κώδικα με περισσότερες προσπελάσεις στον πίνακα