



ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ ΣΤΑ ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΛΛΗΛΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ

Ασκηση 1

Η προσθήκη που κάναμε στον κώδικα είναι η εξής:

```
/* ... */  
for ( t = 0 ; t < T ; t++ ) {  
    #pragma omp parallel for collapse(2) shared(previous, current) private(i,  
j, nbrs)  
    for ( i = 1 ; i < N-1 ; i++ )  
        for ( j = 1 ; j < N-1 ; j++ ) {  
/* ... */
```

Χρησιμοποιούμε το `parallel for` του `omp` για την παραλληλοποίηση των δυο `for-loop` που ακολουθούν. Το `collapse(2)` γνωστοποιεί στο `omp` ότι ακολουθούν τα εν λογο `loops`. Οι μεταβλητές `previous` και `current` χαρακτηρίζονται ως `shared` γιατί αποτελούν κομμάτι των κοινών πόρων των διάφορων εργατών. Οι μεταβλητές `i`, `j`, `nbrs` είναι ιδιωτικές ώστε να αποφευχθούν τυχών `race conditions`.

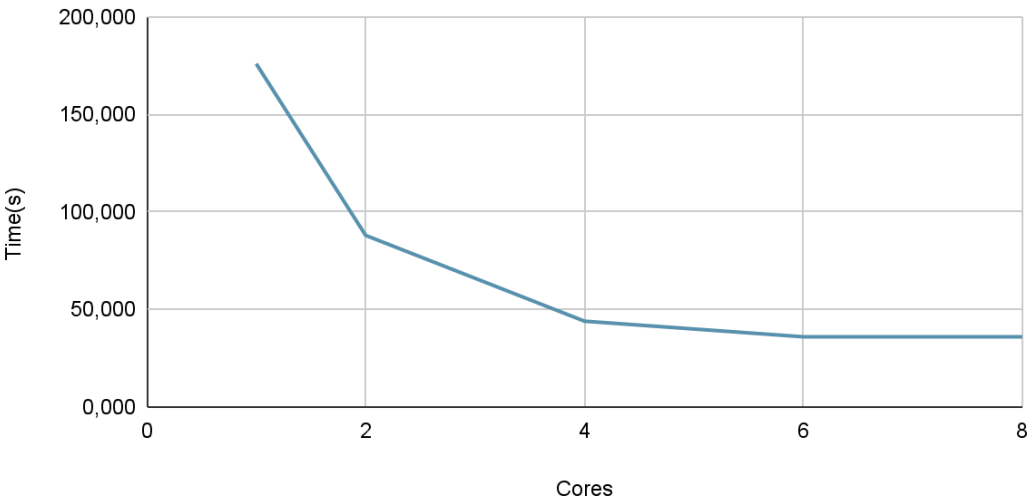
Οι μετρήσεις σε sec:

cores	1	2	4	6	8
64	0.023	0.013	0.01	0.009	0.008
1024	11	5.47	2.72	1.82	1.37
4096	176	88	44	36	36

Παρακάτω παραθέτουμε τα διαγράμματα χρόνου συναρτήσεων των πυρήνων για τα διάφορα μεγέθη:

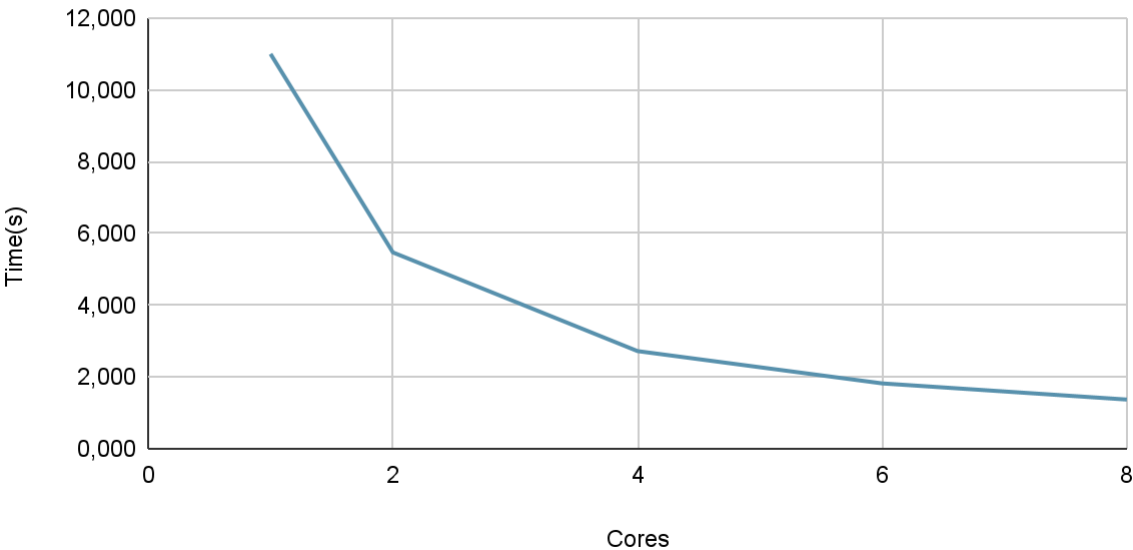
Execution times

size: 4096x4096



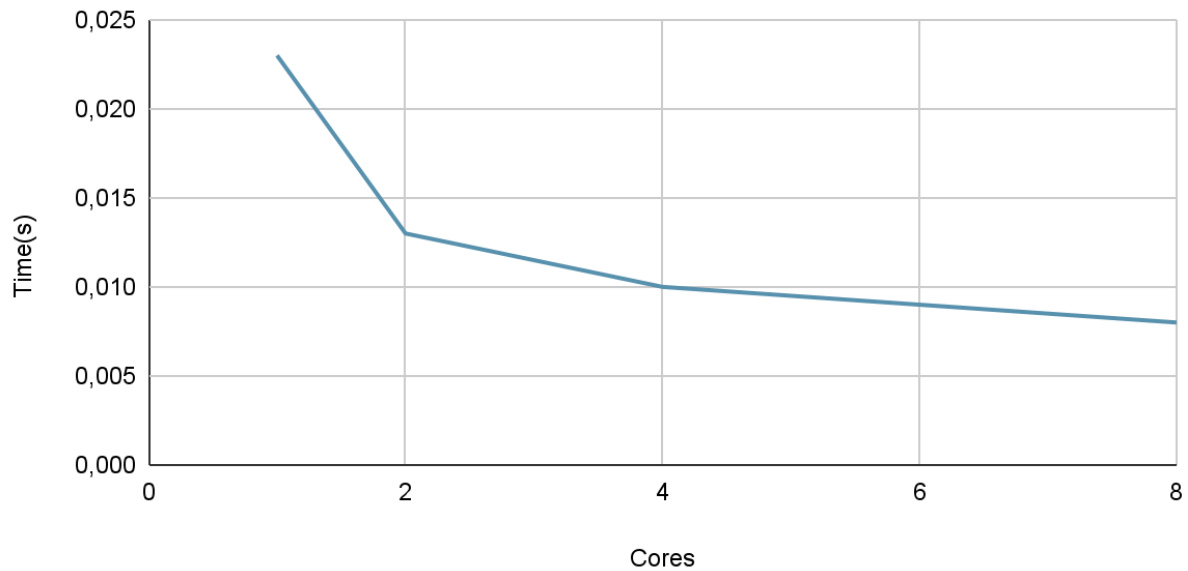
Execution times

size: 1024x1024



Execution times

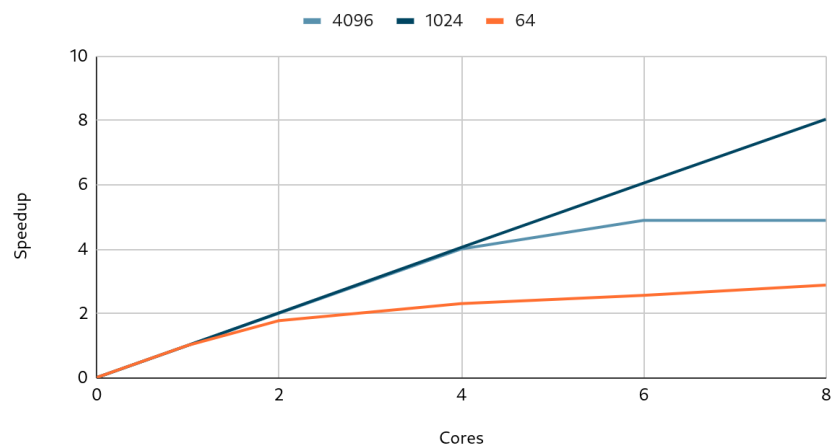
size: 64x64



Παρατηρούμε ότι οι βελτιώσεις στους χρόνους εκτέλεσης είναι πιο εμφανείς σε μεγαλύτερου μεγέθους πινάκων, δηλαδή στα προγράμματα με μεγαλύτερο φόρτος υπολογισμών, ενώ η προσθήκη ακόμα περισσότερων πυρήνων δεν εμφανίζει ιδιαίτερη επιρροή στην επίδοση, πχ η διαφορά επίδοσης από τους 6 στους 8 πυρήνες για μέγεθος πίνακα 4096x4096 είναι ελάχιστη σε σύγκριση με την διαφορά από τον 1 πυρήνα στους 2.

Παρακάτω παραθέτουμε το διάγραμμα επιτάχυνσης:

Speedup



Ομάδα 19 - Μέλη:

Αλταν Αβτζη - 03119241

Τζοναταν Λουκai - 03119230

Σταυρος Λαζαρου - 03112642