

Άσκηση 1:

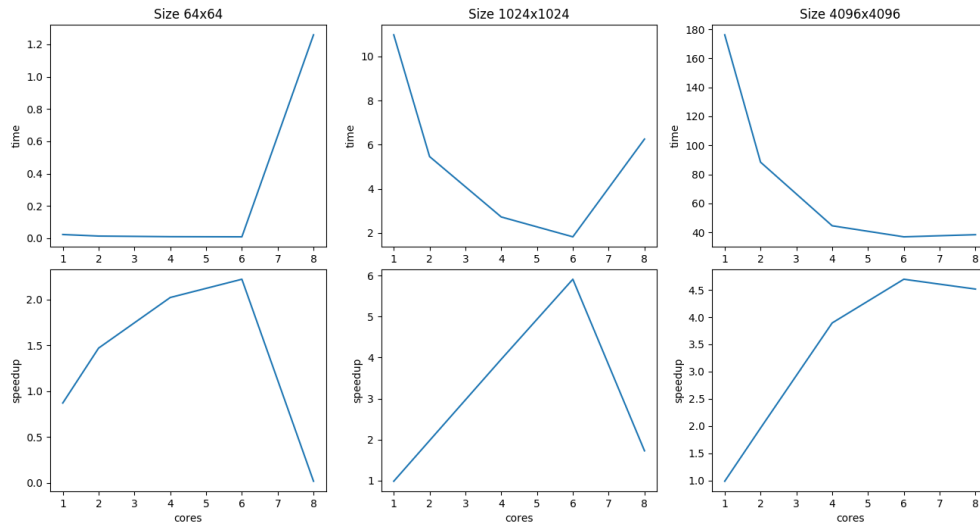
Παραλληλοποίηση αλγορίθμων σε πολυπύρηνες
αρχιτεκτονικές κοινής μνήμης
(ενδιάμεση αναφορά)

Ομάδα 12

Σαριδάκης Μπίτος Δημήτρης

AM: 03114606

1 Game Of Life



Παρατηρούμε ότι σε όλα τα μεγέθη του προβλήματος ο χρόνος πέφτει (και το speedup μεγαλώνει) μέχρι τους 6 επεξεργαστές.

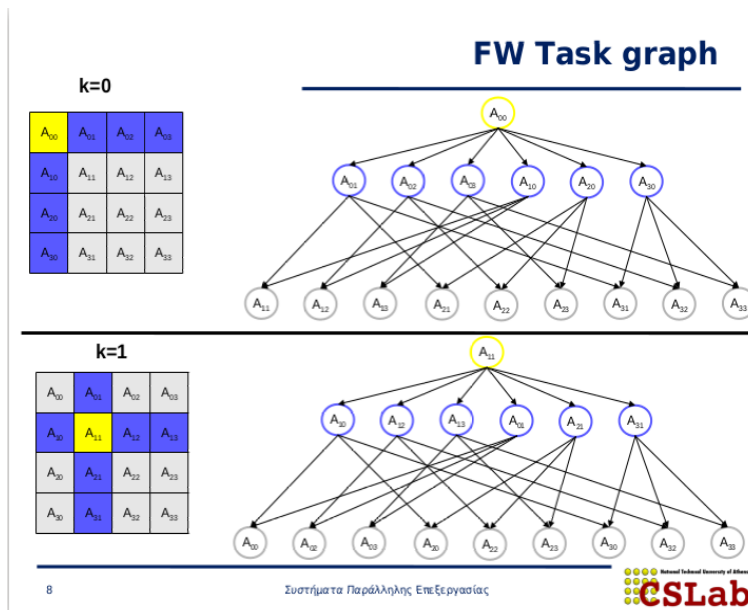
Επομένως, το κόστος συγχρονισμού/επικοινωνίας και διαχείρισης των νημάτων φαίνεται να υπερβαίνει το κέρδος της παραλληλοποίησης στους 8 επεξεργαστές (σπάει η κλιμακωσιμότητα). Μάλιστα, στο 64x64 λόγω πολύ μικρού μεγέθους το φαινόμενο αυτό παίζει τόσο μεγάλο ρόλο που οι 8 επεξεργαστές έκαναν (εντυπωσιακά) πιο πολύ χρόνο από το σειριακό πρόγραμμα!

2 Στρατηγική Παραλληλοποίησης Floyd-Warshall

Θα δημιουργήσουμε 1 task graph για κάθε έκδοση του αλγορίθμου.

2.1 Standard Έκδοση

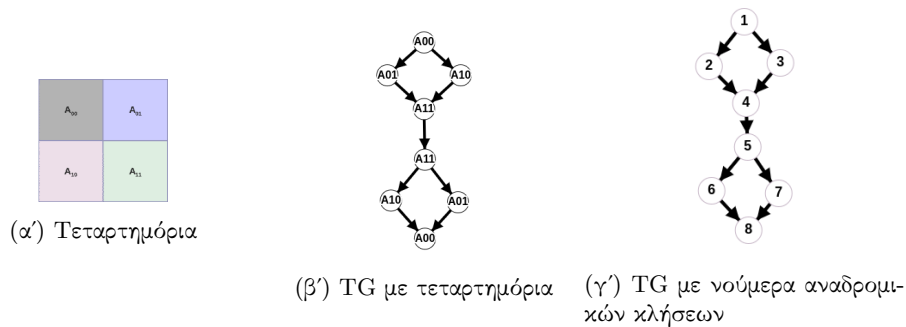
Για την standard έκδοση το task graph είναι αυτό των διαφανειών όπου για κάθε k κάθε στοιχείο της γραμμής και της στήλης k εξαρτάται από το στοιχείο $A_{k,k}$ και τα υπόλοιπα στοιχεία εξαρτώνται από στοιχεία της γραμμής και της στήλης k .



Σχήμα 1: Task graph standard έκδοσης

2.2 Recursive Έκδοση

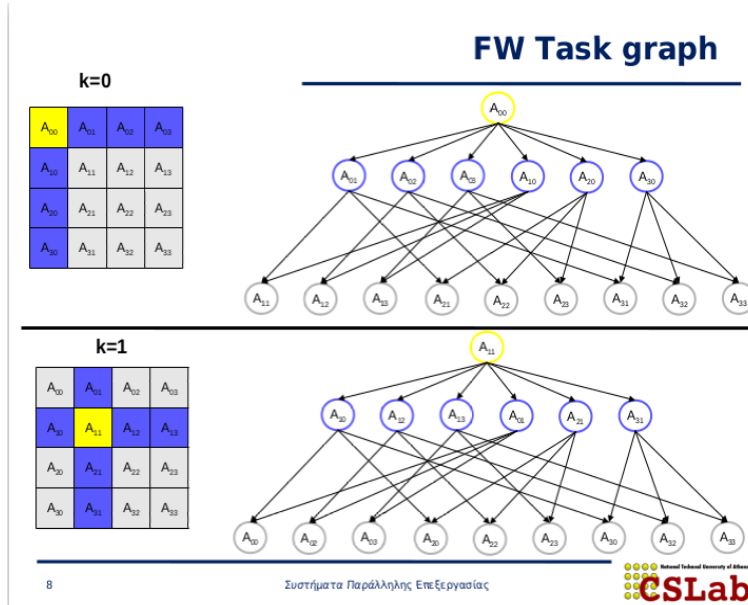
Για την recursive έκδοση τα tasks είναι οι αναδρομικές κλήσεις. Στο σχήμα 2 έχουμε ένα task graph με κόμβους τα τεταρτημόρια στα οποία γράφει η αντίστοιχη αναδρομική κλήση και ένα όπου οι κόμβοι είναι το νούμερο της αναδρομικής κλήσεις με την σειρά που εμφανίζεται στον κώδικα.



Σχήμα 2: Task graph recursive έκδοσης

2.3 Tiled Έκδοση

Στην tiled έκδοση το task graph έχει την ίδια μορφή με εκείνο της standard έκδοσης, με την διαφορά ότι το A_{ij} είναι το αντίστοιχο block και όχι απλά ένα στοιχείο του πίνακα.



Σχήμα 3: Task graph tiled έκδοσης