ΠΜΣ “Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης”

Αλγοριθμικές Τεχνικές και Εφαρμογές

Καθηγητής: Χαράλαμπος Κωνσταντόπουλος

Πάνου Γιώργος - Κώστας Σπυρόπουλος

2η Εργασία

Τα σύνορα και οι περιφέρειες μιας χώρας συχνά μπορούν να αναπαρασταθούν με την χρήση πολυγραμμών (polylines). Μια πολυγραμμή είναι το σχήμα που προκύπτει από την ένωση πολλών σημείων με ευθύγραμμα τμήματα, θεωρώντας τα σημεία αυτά ως τις συντεταγμένες που αποτελούν τα σύνορα της κάθε χώρας.

Συχνά, η απεικόνιση των συνόρων σε ένα χάρτη μπορεί να είναι αρκετά λεπτομερής. Για αυτό τον λόγο γίνεται χρήση του αλγορίθμου Ramer-Douglas-Pecker, που σκοπός του είναι η απλοποίηση των πολυγραμμών που αποτελούν τα σύνορα αφαιρώντας σημεία έτσι ώστε να επιτευχθεί προσέγγιση της αρχικής πολυγραμμής με μια νέα απλούστερη πολυγραμμή.

Ο αλγόριθμος Ramer-Douglas-Pecker θεωρεί την ευθεία που ενώνει το πρώτο και το τελευταίο σημείο της πολυγραμμής. Από τα υπόλοιπα σημεία, βρίσκει το σημείο που απέχει περισσότερο απο την ευθεία, και αν η απόστασή του είναι μικρότερη από την επιτρεπόμενη απόκλιση, τότε η πολυγραμμή αντικαθιστάται από την ευθεία. Εάν η απόσταση είναι μεγαλύτερη από την επιτρεπόμενη απόκλιση, τότε η πολυγραμμή χωρίζεται στο σημείο αυτό και ο αλγόριθμος επαναλαμβάνεται αναδρομικά για τις δυο νέες πολυγραμμές.

Στα πλαίσια της εργασίας αυτής υλοποιήσαμς τον παραπάνω αλγόριθμο με την χρήση του προτύπου POSIX Threads. Με την χρήση “νημάτων” κάποια τμήματα του αλγορίθμου παραλληλοποιούνται και επιτυγχάνεται μείωση του χρόνου υπολογισμού του αλγορίθμου.

Ως βάση της υλοποίησης μας χρησιμοποιήσαμε τον κώδικα “2D implementation of the Ramer-Douglas-Peucker algorithm - By Tim Sheerman-Chase, 2016” στη γλώσσα προγραμματισμού C++ και χρησιμοποιώντας την βιβλιοθήκη pthreads.

Η λογική της παραλληλοποίηση έχει ως εξής:

Αρχικά ορίζουμε τον αριθμό των threads που θα πραγματοποιήσουν τους υπολογισμούς. Το σύνολο των πολυγραμμών ορίζουν ορίζουν ένα work-pool ώστε κάθε πολυγραμμή να ανατίθεται σε όποιο από τα threads είναι αδρανές. Υπάρχει ένας κοινός thread counter που ορίζει ποια πολυγραμμή είναι διαθέσιμη προς απλοποίηση. Τόσο ο thread counter όπως και το vector των πολυγραμμών προστατεύονται με mutex ώστε να μην υπάρξουν ασυνέπειες στο τελικό αποτέλεσμα.

Στη συνέχεια, για καλύτερη εξισορρόπηση φόρτου, ορίζουμε ένα threshold ως το όριο του σημείων που θέλουμε να έχει μια πολυγραμμή. Εάν δεν υπερβαίνει το threshold τότε η υλοποίηση του αλγορίθμου γίνεται σειριακά από το thread το οποίο έχει αναλάβει την ανάλυσή του. Εάν υπερβαίνει το threshold, τότε η πολυγραμμή χωρίζεται σε δύο μικρότερα κομμάτια, από τα οποία το πρώτο υλοποιείται από το thread κανονικά και το δεύτερο προστίθεται στην δεξαμενή έργων για μετέπειτα απλοποίηση του από κάποιο thread που θα είναι αδρανές.

Για την ένωση των κομματιών που έχουν προκύψει από αυτή τη διαδικασία χρησιμοποιήθηκαν σημαφόροι ώστε να είναι βέβαιο ότι ανεξάρτητα από την σειρά εκτέλεσης του κάθε thread η τελική απλοποιημένη πολυγραμμή θα διατηρεί την αρχική διάταξή της. Επειδή η C++ δεν παρέχει εγγενώς σημαφόρους τους υλοποιήσαμε με ένα counter array και mutex.

Μέτρηση και αξιολόγηση μείωσης χρόνου ανάμεσα σε σειριακή και παράλληλη υλοποίηση:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threads # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Time (sec) | **2.14** | **1.167** | **0.823** | **0.667** | **0.574** | **0.498** | **0.472** | **0.443** |