Άσκηση 4
Παράλληλα & Κατανεμημένα Υπολογιστικά Συστήματα
22 Ιανουαρίου 2019

Να υλοποιήσετε σε CUDA[1] αλγόριθμο για την εύρεση του πλήθους των τριγώνων σε μη-κατευθυνόμενο, απλό γράφο $G(\mathcal{V}, \mathcal{E})$, όπου $\mathcal{V}$ το σύνολο των κόμβων, πλήθους $N = |\mathcal{V}|$ και $\mathcal{E}$ το σύνολο των ακμών, πλήθους $M = |\mathcal{E}|$. Το τρίγωνο $\mathcal{C}_3$ ορίζεται ως ο πλήρης μη-κατευθυνόμενος γράφος 3 κορυφών.

Η καταμέτρηση του αριθμού των τριγώνων μπορεί να υλοποιηθεί με βασικές πράξεις γραμμικής άλγεβρας χρησιμοποιώντας τον *πίνακα γειτνίασης* (adjacency matrix) $\mathbf{A}$ του γράφου G, μεγέθους $N \times N$. Το στοιχείο $\mathbf{A}_{ij}$ έχει τιμή TRUE αν υπάρχει ακμή μεταξύ των κόμβων i και j, αλλιώς έχει τιμή FALSE. Συγκεκριμένα, ο αριθμός των τριγώνων μπορεί να υπολογιστεί με τον παρακάτω αλγόριθμο:

Αλγόριθμος 1 Εύρεση πλήθους τριγώνων n_T σε γράφο $G(\mathcal{V}, \mathcal{E})$

Είσοδος: Πίνακας γειτνίασης $\mathbf{A}$ του γράφου $G(\mathcal{V}, \mathcal{E})$
Έξοδος: Αριθμός τριγώνων n_T στον γράφο G

1: $\mathbf{C} = (\mathbf{A} \cdot \mathbf{A}) \odot \mathbf{A}$, $\cdot$: γινόμενο πινάκων $\odot$: γινόμενο Hadamard
2: $n_T = \frac{1}{6} \sum_{ij} \mathbf{C}_{ij}$

Ο πίνακας $\mathbf{A}$ πρέπει να αποθηκευτεί σε *αραιή* (sparse) μορφή, ώστε να είναι δυνατή η επεξεργασία μεγάλων δεδομένων. Σας δίνεται κώδικας MATLAB που υλοποιεί τον αλγόριθμο.

Το πρόγραμμά σας θα πρέπει να:

- Διαβάζει τον πίνακα γειτνίασης για έναν γράφο $G(\mathcal{V}, \mathcal{E})$ σε αραιή μορφή, από αρχείο στον σκληρό δίσκο.
- Μεταφέρει τα δεδομένα που χρειάζονται στην GPU.
- Υπολογίζει το πλήθος των τριγώνων που υπάρχουν στον γράφο, με τον αλγόριθμο που περιγράφτηκε.
- Ελέγχει την ορθότητα των αποτελεσμάτων.
- Δουλεύει σωστά για: $M \leq 2^{25}$, $N \leq 2^{23}$.

Χρησιμοποιήστε πολλαπλά νήματα και `shared memory` για να αυξήσετε την ταχύτητα του προγράμματος σας. Χρησιμοποιήστε διαθέσιμες βιβλιοθήκες όπως η cuSPARSE[2]. Για να αυξήσετε την απόδοση της υλοποίησής σας, υπολογίστε το γινόμενο των πινάκων μόνο στις μη-μηδενικές θέσεις του $\mathbf{A}$ (εξαιτίας του γινομένου Hadamard, μόνο αυτές οι θέσεις μπορεί να είναι μη μηδενικές στον πίνακα $\mathbf{C}$). Επιπλέον, δεν απαιτείται να αποθηκεύσετε τον πίνακα $\mathbf{C}$. Μπορείτε να υπολογίσετε απευθείας το τελικό άθροισμα.

Χρησιμοποιήστε τους εξής γράφους από την συλλογή `SuiteSparse Matrix Collection`[3]: i) `auto`, ii) `great-britain_osm` και iii) `delaunay_n22` για τα πειράματά σας.

Παραδώστε:

- Αναφορά 3−4 σελίδων που να περιγράφει τη μέθοδο του παραλληλισμού καθώς και τους ελέγχους ορθότητας που χρησιμοποιήσατε.
- Σχόλια και συμπεράσματα για την ταχύτητα υπολογισμών συγκριτικά με την σειριακή έκδοση του αλγορίθμου, για το εύρος παραμέτρων που δουλεύει σωστά ο παράλληλος αλγόριθμος.
- Τον κώδικα του προγράμματος.

Δεοντολογία: Εάν χρησιμοποιήσετε κώδικες από το διαδίκτυο ή αλλού, να αναφέρετε την πηγή και τις αλλαγές που κάνατε.

Σημείωση: Ομαδικές εργασίες γίνονται δεκτές. Ο μέγιστος αριθμός φοιτητών που μπορούν να συνεργαστούν σε μία ομάδα είναι δύο, αρκεί κανένα ζευγάρι να μην έχει συνεργαστεί σε προηγούμενη εργασία.

Ημερομηνία παράδοσης: 19 Φεβρουαρίου 2019.

---

[1] https://docs.nvidia.com/cuda/
[2] https://developer.nvidia.com/cuBLAS
[3] https://sparse.tamu.edu