Άσκηση 4
Παράλληλα & Κατανεμημένα Συστήματα Υπολογιστών
28 Ιανουαρίου 2016

Επέκταση της $3^{\eta\varsigma}$ άσκησης (επίλυση του *All Pair Shortest Path*), ώστε να βελτιωθεί περαιτέρω ο συνολικός χρόνος επίλυσης του προβλήματος. Επιλέξτε μία από τις δύο εναλλακτικές προτάσεις που ακολουθούν:

Επιλογή 1: Σε κάθε βήμα του αλγορίθμου *All Pair Shortest Path*, το πρόβλημα εύρεσης της ελάχιστης απόστασης μπορεί να χωριστεί σε υποπροβλήματα και το κάθε υποπρόβλημα να λυθεί ανεξάρτητα από τα υπόλοιπα, ως το τέλος του βήματος. Ο χωρισμός του προβλήματος γίνεται τεμαχίζοντας τον πίνακα γειτνίασης (*adjacency matrix*) σε *blocks*. Το κάθε *block* αποτελεί και ένα υποπρόβλημα. Οι υπολογισμοί μπορεί να γίνουν παράλληλα, αν τα *blocks* ανατεθούν σε διαφορετικά *MPI processes*.

Να γραφεί πρόγραμμα σε *MPI-CUDA* το οποίο:

- Δέχεται σαν είσοδο τον πίνακα γειτνίασης (όπως στην 3^{η} άσκηση) και τον τεμαχίζει σε ίσα *blocks*. Εναλλακτικά, μπορεί κάθε *process* να διαβάζει εξ' αρχής διαφορετικό τμήμα του πίνακα από αρχείο.
- Υπολογίζει κάθε βήμα του αλγορίθμου παράλληλα σε διαφορετικά *MPI processes*.
- Υλοποιεί τις κατάλληλες επικοινωνίες ανάμεσα στα *MPI processes*, στο τέλος κάθε βήματος.
- Επιστρέφει το τελικό αποτέλεσμα σε ένα *process*

Για περισσότερες πληροφορίες και ιδέες, δείτε την εργασία [2].

Επιλογή 2: Η πρόσβαση στη μνήμη διευκολύνεται, αν χρησιμοποιηθούν δύο αντίγραφα του πίνακα γειτνίασης. Ένα αντίγραφο του οποίου τα στοιχεία προσπελάζονται συνεχόμενα κατά γραμμές και ένα του οποίου τα στοιχεία προσπελάζονται συνεχόμενα κατά στήλες.

Να γράφει πρόγραμμα σε *CUDA* το οποίο:

- Δέχεται ως είσοδο τον πίνακα γειτνίασης $\mathbf{A}$ και τον αναστρέφει δημιουργώντας το πίνακα $\mathbf{A^T}$. Αν τα στοιχεία του $\mathbf{A}$ είναι αποθηκευμένα συνεχόμενα κατά γραμμές, τότε τα στοιχεία του $\mathbf{A^T}$ θα είναι αποθηκευμένα συνεχόμενα κατά στήλες.
- Χρησιμοποιεί τους πίνακες $\mathbf{A}$ και $\mathbf{A^T}$ ώστε η πρόσβαση στη μνήμη να είναι συνεχόμενη, σε κάθε βήμα του αλγορίθμου (*coalesced memory access*).
- Ανανεώνει τα περιεχόμενα των $\mathbf{A}$ και $\mathbf{A^T}$ έτσι ώστε στο τέλος κάθε βήματος οι δυο πίνακες να περιέχουν την ίδια πληροφορία και να μπορούν να χρησιμοποιηθούν στο επόμενο βήμα.
- Υλοποιεί τις διαφορετικές εκδόσεις πύρινων *CUDA* που ζητούταν στην 3^{η} άσκηση.

Για περισσότερες πληροφορίες και ιδέες, δείτε την εργασία [1].

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 έκδοση του αλγορίθμου της $3^{\eta\varsigma}$ άσκησης, για μέγεθος προβλήματος $n = 2^{[7:12]}$ και πιθανότητα ύπαρξης ακμής $p = [0.33 \quad 0.45 \quad 0.66]$ (διαγράμματα με χρόνους). Στο χρόνο σύγκρισης πρέπει να συμπεριληφθεί και η μεταφορά των δεδομένων προς και από την `device memory`.
- 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 τρεις, αρκεί κανένα ζευγάρι να μην έχει συνεργαστεί σε προηγούμενη εργασία.

Ημερομηνία παράδοσης: 11:59μμ, Κυριακή 28 Φεβρουαρίου 2016.

Αναφορές

[1] Bo Wu, Zhijia Zhao, Eddy Zheng Zhang, Yunlian Jiang, and Xipeng Shen. Complexity Analysis and Algorithm Design for Reorganizing Data to Minimize Non-coalesced Memory Accesses on GPU. *SIGPLAN Not.*, 48(8):57–68, February 2013.

[2] Qingshuang Wu, Chunya Tong, Qiang Wang, and Xiangfu Cheng. All-pairs Shortest Path Algorithm based on MPI+CUDA Distributed Parallel Programming Model. *Journal of Networks*, 8(12):2797–2803, 2013.