Ειδικά Θέματα Παράλληλου Προγραμματισμού 2022-23

Εργασία #2 (GPU Programming)

A. Θόλωση εικόνων (100%)

Σας δίνεται ένα σειριακό πρόγραμμα (**gaussian-blur.c**) το οποίο εφαρμόζει Gaussian blur προκειμένου να θολώσει (ή να ομαλοποιήσει) μία εικόνα. Η συνάρτηση που κάνει τη θόλωση είναι η *gaussian_blur_serial()*, η οποία παίρνει μία εικόνα **imgin** και παράγει τη θολωμένη της εκδοχή **imgout**, βάσει μίας ακτίνας θόλωσης **radius** (όσο μεγαλύτερη η ακτίνα, τόσο πιο έντονο το θόλωμα). Σας ζητείται να γίνεται η θόλωση παράλληλα χρησιμοποιώντας το **OpenCL** ως εξής (χωρίς να αλλάξετε τον αλγόριθμο, απλά να μοιράσετε σωστά τη δουλειά):

- να συμπληρώσετε τη συνάρτηση *gaussian_blur_opencl()* (και όπου αλλού χρειάζεται για αρχικοποίηση και δέσμευση μνήμης), παραλληλοποιώντας κατάλληλα την συνάρτηση θόλωσης στην GPU.

- βρείτε τον βέλτιστο μέγεθος ομάδας εργασίας (work group size). Μπορείτε να πειραματιστείτε με διάφορες τιμές για το *local_work_size*. Το μέγεθος της ομάδας εργασίας πρέπει να επιλεγεί έτσι ώστε να μεγιστοποιεί τη χρήση των πόρων στην κάρτα γραφικών και να ελαχιστοποιεί την καθυστέρηση πρόσβασης στη μνήμη.

Για τη θόλωση εικόνων δίνεται εικόνα με ανάλυση 500x500, 1000x1000 και 1500x1500. Πειραματιστείτε με όλες, αλλά δώστε αποτελέσματα μόνο για την μεγαλύτερη εικόνα και για ακτίνα θόλωσης ίση με >=8.

Απαιτούμενα

- Ο **πηγαίος κώδικας** που δίνετε για τις υλοποιήσεις σας θα πρέπει να είναι σωστά δομημένος, στοιχισμένος και σχολιασμένος (προτεινόμενη γλώσσα τα Αγγλικά).

- Θα πρέπει να παραδώσετε **πλήρη αναφορά**, περιλαμβάνοντας και γραφικές παραστάσεις χρονομετρήσεων καθώς και συζήτηση γύρω από τα αποτελέσματα.

- Τα προγράμματά σας (πηγαίοι κώδικες + αναφορά) θα πρέπει να τα παραδώσετε στο eclass του μαθήματος σε μορφή **zip αρχείου**. Στο όνομα του αρχείου θα πρέπει να περιλαμβάνεται ο **αριθμός μητρώου** του φοιτητή.

- Οι ασκήσεις ελέγχονται για κοινό κώδικα και **αντιγραφή**. Τέτοιες περιπτώσεις φυσικά θα **μηδενίζονται** και δεν θα υπάρχει δικαίωμα εξέτασης στις επόμενες εξεταστικές περιόδους.

- Για τη **χρονομέτρηση** μπορείτε να χρησιμοποιήσετε κλήσεις χρονομέτρησης που παρέχει το ίδιο το OpenCL (με χρήση Events, δείτε τον κώδικα που σας δίνεται).

- Τα προγράμματά σας να τα δοκιμάσετε σε **διαφορετική συσκευή (CPU και GPU)** και να τα συγκρίνετε με τον καθαρό σειριακό και OpenMP κώδικα σας από την πρώτη εργασία. Πειραματιστείτε με διάφορες τιμές για το **local_work_size** και πραγματοποιήσετε (profiling) για να βρείτε την καλύτερη ρύθμιση για τη συγκεκριμένη κάρτα γραφικών και την εφαρμογή σας.

- Για κάθε περίπτωση, ένα πρόγραμμα θα εκτελείται τουλάχιστον **4 φορές** και ο τελικός χρόνος θα είναι ο μέσος όρος των τεσσάρων χρόνων.

Παρατηρήσεις

- Η παράδοση της εργασίας είναι **υποχρεωτική** και χρειάζεται βαθμολογία > 50% για να μπορείτε να δώσετε στις τελικές γραπτές εξετάσεις.

- Για διευκόλυνση σας, σας δίνεται ένα απλό OpenCL παράδειγμα (**array-addition.c**) το οποίο υλοποιεί την πρόσθεση 2 μονοδιάστατων πινάκων Ν θέσεων. Για οποιοδήποτε πρόβλημα εγκατάστασης της OpenCL στο μηχάνημα σας, παρακαλώ επικοινωνήστε με τον διδάσκοντα (abasilak@aueb.gr).

Προθεσμία παράδοσης: Κυριακή, 4 Ιουνίου 2022

Καλή Επιτυχία!

Ανδρέας-Αλέξανδρος Βασιλάκης

20/05/2023