

Εργασία #2 (GPU programming)

cuinfo (50%)

Για συσκευές OpenCL υπάρχει συνήθως ένα μικρό utility που ονομάζεται `clinfo` το οποίο βρίσκει όλες τις σχετικές συσκευές που υπάρχουν στο σύστημα και παρουσιάζει βασικές πληροφορίες για αυτές. Σας ζητείται να κάνετε κάτι αντίστοιχο για συσκευές CUDA, συμπληρώνοντας το πρόγραμμα `cuinfo`, το οποίο σας δίνεται κενό. Το `cuinfo` χρησιμοποιεί το CUDA Runtime API, ανακαλύπτει ποιες συσκευές CUDA υπάρχουν και για κάθε μία εμφανίζει τις εξής πληροφορίες:

- Όνομα συσκευής
- Έκδοση CUDA compute capability (π.χ. 3.5)
- Έκδοση CUDA driver και CUDA runtime (π.χ. 10.2)
- Πλήθος διαθέσιμων streaming multiprocessors (SMs)
- Συνολικό μέγεθος καθολικής μνήμης (global memory)
- Συνολικό μέγεθος μνήμης σταθερών (constant memory)
- Μέγεθος κοινόχρηστης μνήμης (shared memory) ανά block

Υπάρχει τρόπος να βρούμε το συνολικό πλήθος των cores μιας NVIDIA GPU? Εάν ναι, δώστε τις λεπτομέρειες στην αναφορά σας.

Θόλωση εικόνων με GPU offloading (50%)

Σας δίνεται ένα σειριακό πρόγραμμα το οποίο εφαρμόζει Gaussian blur προκειμένου να θολώσει (ή να ομαλοποιήσει) μία εικόνα (βλ. και Εργασία #1). 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 στη GPU χρησιμοποιώντας OpenMP ως εξής:

- Θα πρέπει συμπληρώσετε τη συνάρτηση `gaussian_blur_omp_device()` ώστε με τις κατάλληλες οδηγίες για offloading (βλ. Λεπτομέρειες παρακάτω), η θόλωση να γίνεται στην GPU
- Η συσκευή CUDA που σας παρέχεται διαθέτει 30 streaming multiprocessors με 128 πυρήνες ο καθένας. Θα πρέπει να πειραματιστείτε με όλα τα διαφορετικά πλήθη ομάδων και νημάτων ανά ομάδα με τους παρακάτω περιορισμούς:
 - Το πλήθος νημάτων ανά ομάδα που θα χρησιμοποιήσετε στην οδηγία OpenMP θα πρέπει να είναι πάντα πολλαπλάσιο του 32
 - Το πλήθος ομάδων θα πρέπει πάντα να είναι τέτοιο ώστε να χρησιμοποιούνται όλοι οι πυρήνες της συσκευής.

Λεπτομέρειες

Απαιτούμενα

- Θα πρέπει να παραδώσετε πλήρη αναφορά, περιλαμβάνοντας και γραφικές παραστάσεις χρονομετρήσεων καθώς και συζήτηση γύρω από τα αποτελέσματα, συγκρίνοντας και με τον αντίστοιχο σειριακό κώδικα.
- Τα προγράμματά σας (πηγαίοι κώδικες + αναφορά) θα πρέπει να τα παραδώσετε με `turnin set2@mye023`. Πληροφορίες στην ιστοσελίδα του μαθήματος.

- Για τη χρονομέτρηση της θόλωσης μπορείτε να χρησιμοποιήσετε κλήσεις χρονομέτρησης που παρέχει το ίδιο το OpenMP (`omp_get_wtime()` κλπ) ή την `gettimeofday()`.
- Το πρόγραμμα της θόλωσης θα πρέπει να συνδυάζει κατάλληλα τις οδηγίες `target`, `teams`, `distribute` και `parallel for collapse(2)`. Δοκιμάστε εμφωλευμένες οδηγίες (η μία μέσα στην άλλη) αλλά και συνδυασμένες οδηγίες (όπου σε μία γραμμή συνδυάζονται περισσότερες από μία οδηγίες). Βλέπετε κάποια διαφορά στους χρόνους εκτέλεσης;
- Για κάθε περίπτωση, ένα πρόγραμμα θα εκτελείται τουλάχιστον 4 φορές και ο τελικός χρόνος θα είναι ο μέσος όρος των τεσσάρων χρόνων.

Παρατηρήσεις

1. Η ανάπτυξη των προγραμμάτων σας μπορεί να γίνει οπουδήποτε αλλά η εκτέλεση και χρονομέτρηση των πειραμάτων σας θα πρέπει να γίνει στον υπολογιστή `parallax` χρησιμοποιώντας τους προσωρινούς κωδικούς και τις οδηγίες που σας έχουν δοθεί. Στην ιστοσελίδα του μαθήματος υπάρχουν οι σχετικές οδηγίες.
2. Τα αντίστοιχα σειριακά προγράμματα μπορείτε να τα βρείτε στην ιστοσελίδα του μαθήματος.
3. Το αρχείο πηγαίου κώδικα του προγράμματος `cuinfo` πρέπει να έχει κατάληξη `.cu` ή `.c`.
4. Για τη θόλωση εικόνων δίνεται στην ιστοσελίδα του μαθήματος εικόνα με ανάλυση 500×500 , 1000×1000 και 1500×1500 . Πειραματιστείτε με όλες, αλλά δώστε αποτελέσματα μόνο για την μεγαλύτερη εικόνα και για ακτίνα θόλωσης ίση με 8.

Προθεσμία παράδοσης:

Δευτέρα, 12 Μαΐου 2025

Βασίλειος Δημακόπουλος