



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Παράλληλα Συστήματα
Εργασία 4

Ονοματεπώνυμο: ΙΩΑΝΝΟΥ ΝΙΚΟΛΑΣ

Αριθμός Μητρώου: 1115202200054

Παραδοτέο

Το αρχείο 01-nbody.cu περιέχει τον κώδικα που έχει ως αποτέλεσμα τρέχει η εφαρμογή σε λιγότερο από 0.9 δευτερόλεπτα για 4096 σώματα και σε λιγότερο από 1.3 δευτερόλεπτα για 65536 σώματα. Στο Readme περιλαμβάνεται το πιστοποιητικό, η περιγραφή του προβλήματος, η ανάλυση της λύσης, τα αποτελέσματα και ένας σχολιασμός αυτών που βρέθηκαν.

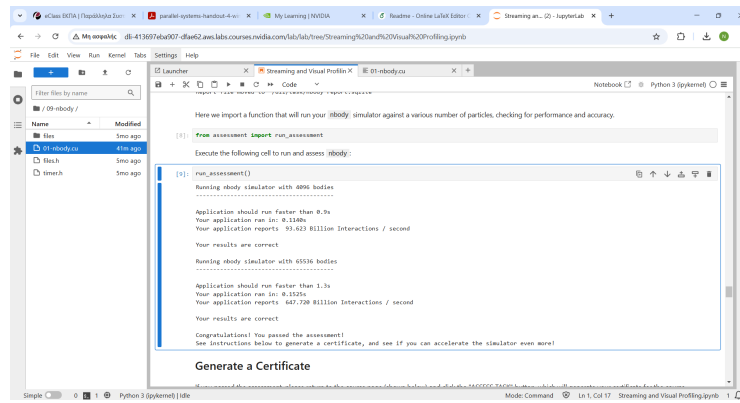
Περιγραφή προβλήματος

Η άσκηση ζητάει να επιταχύνουμε το αρχείο 01-nbody.cu. Πιο συγκεκριμένα, το αρχείο αυτό είναι ένας προσομοιωτής n-body για σώματα που κινούνται σε τρισδιάστατο χώρο, το οποίο βασίζεται στη CPU. Η χρήση της CPU έχει ως αποτέλεσμα να γίνεται αργή εκτέλεση της εφαρμογής. Η επιτάχυνση του προγράμματος θα γίνει με την χρήση της GPU και για να θεωρηθεί σωστή η μετατροπή θα πρέπει να έχουμε χρόνο εκτέλεσης λιγότερο από 0.9 δευτερόλεπτα για 4096 σώματα και λιγότερο από 1.3 δευτερόλεπτα για 65536 σώματα.

Ανάλυση λύσης

Η κύρια ιδέα πίσω από τη μετατροπή του προγράμματος από τη χρήση της CPU στη GPU είναι η αξιοποίηση της παράλληλης εκτέλεσης μέσω της CUDA. Αυτό επιτεύχθηκε με την αλλαγή του ορισμού των συναρτήσεων και τη χρήση της shared memory. Αρχικά, οι συναρτήσεις `bodyForce(Body *p, float dt, int n)` και `integratePosition(Body *p, float dt, int n)` μετατράπηκαν σε CUDA kernels με την προσθήκη του προθέματος `__global__`. Αυτή η αλλαγή επιτρέπει την εκτέλεση των συναρτήσεων στην GPU, επιτυγχάνοντας την παράλληλη επεξεργασία των δεδομένων από πολλά νήματα ταυτόχρονα. Στόχος αυτής της προσέγγισης είναι η μεγιστοποίηση της απόδοσης μέσω της εκμετάλλευσης της αρχιτεκτονικής πολλαπλών πυρήνων της GPU. Στη συνέχεια, στη συνάρτηση `bodyForce`, η οποία υπολογίζει τις βαρυτικές δυνάμεις μεταξύ των σωμάτων, χρησιμοποιήθηκε `shared memory` για να βελτιωθεί η απόδοση. Η `shared memory` επιτρέπει την αποθήκευση των σωμάτων που ανήκουν στο ίδιο μπλοκ μνήμης, μειώνοντας τις καθυστερήσεις κατά την ανάγνωση των δεδομένων. Με τον τρόπο αυτό, τα δεδομένα που χρησιμοποιούνται συχνά παραμένουν προσβάσιμα πολύ γρηγορότερα για τα νήματα του μπλοκ, ενισχύοντας τη δυνατότητα παράλληλης επεξεργασίας τους και μειώνοντας την καθυστέρηση. Επίσης, ορίστηκαν οι παράμετροι `blockSize` και `gridSize`, ώστε να διασφαλιστεί η βέλτιστη κατανομή των εργασιών στην GPU. Τέλος, για την επιτυχή λειτουργία της παράλληλης εκτέλεσης και την αποφυγή σφαλμάτων συγχρονισμού, χρησιμοποιήθηκε η εντολή `__syncthreads()`.

Αποτελέσματα



Σχολιασμός Αποτελεσμάτων

Η χρήση της GPU οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης του προγράμματος σε σύγκριση με την CPU . Αυτό οφείλεται κυρίως στην εκτεταμένη παράλληλη επεξεργασία, καθώς η GPU διαθέτει πολλαπλάσιους πυρήνες από την CPU , επιτρέποντας την ταυτόχρονη εκτέλεση μεγάλου αριθμού υπολογισμών. Επιπλέον, η GPU αξιοποιεί τη shared memory, προσφέροντας ταχύτερη πρόσβαση στα δεδομένα και μειώνοντας τον χρόνο αναμονής κατά τη διάρκεια των υπολογισμών. Συνολικά, αυτοί οι παράγοντες συμβάλλουν στη βελτίωση της απόδοσης και καθιστούν τη GPU την ιδανική επιλογή για εφαρμογές με υψηλές απαιτήσεις υπολογιστικής ισχύος, όπως η άσκηση αυτή.

Πιστοποιητικό

