

Προγραμματισμός Συστημάτων Υψηλής Επίδοσης (ECE415)

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Πανεπιστήμιο Θεσσαλίας

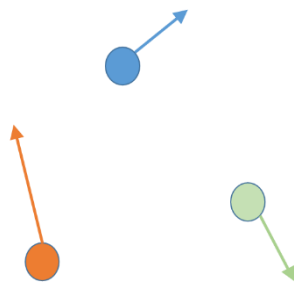
Διδάσκων: Χρήστος Δ. Αντωνόπουλος

5η Εργαστηριακή Άσκηση

Στόχος: Παραλληλοποίηση και βελτιστοποίηση εφαρμογής προσομοίωσης NBody σε GPU.

Εισαγωγή:

Οι προσομοιώσεις NBody προσομοιώνουν την εξέλιξη δυναμικών συστημάτων σωμάτων (σωματιδίων, ατόμων, πλανητών), υπό την επίδραση φυσικών δυνάμεων (ηλεκτροστατικών, βαρυτικών).



Εικόνα 1: Στιγμιότυπο τριών κινούμενων σωμάτων

Η χρονική εξέλιξη του φαινομένου προσομοιώνεται σε βήματα. Σε κάθε βήμα τα σώματα χαρακτηρίζονται:

- Από τη θέση τους στον 3D χώρο.
- Από το διάνυσμα της ταχύτητάς τους.

Επιπλέον, ανάλογα με τη φύση του φαινομένου, το κάθε σώμα μπορεί επίσης να χαρακτηρίζεται από άλλες φυσικές ποσότητες, όπως π.χ. η μάζα ή το φορτίο του.

Ως σημείο αναφοράς και εκκίνησης σας δίνεται μια ακολουθιακή υλοποίηση του κώδικα σε C.

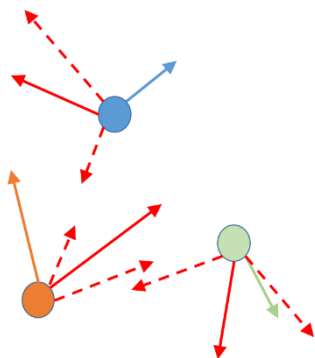
Υπόβαθρο:

Ο κώδικας που σας δίνεται υλοποιεί την απλούστερη δυνατή μορφή του αλγόριθμου.

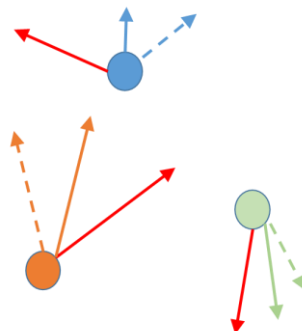
Για κάθε χρονικό βήμα:

α) Υπολογίζονται οι δυνάμεις που εξασκούνται σε κάθε σώμα (και η συνισταμένη τους), εξαιτίας της αλληλεπίδρασής του με καθένα από τα υπόλοιπα σώματα. Οι δυνάμεις αυτές προφανώς εξαρτώνται από τη μεταξύ τους απόσταση και τη μάζα / φορτίο τους.

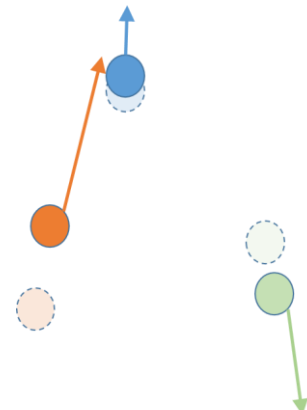
β) Η συνισταμένη δύναμη αξιοποιείται ώστε να υπολογιστεί η αντίστοιχη επιτάχυνση καθώς και η



Εικόνα 2: Βήμα (α)



Εικόνα 3: Βήμα (β)



Εικόνα 4: Βήμα (γ)

συνακόλουθη μεταβολή του διανύσματος ταχύτητας κάθε σώματος.

γ) Με βάση την ταχύτητα και το χρονικό διάστημα μεταξύ διαδοχικών βημάτων υπολογίζεται η νέα θέση των σωμάτων.

Ζητούμενα:

Σε 1^η φάση, θα πρέπει να παραλληλοποιήσετε για τη CPU τον κώδικα που σας δόθηκε με χρήση OpenMP. Λογικά δε θα χρειαστεί να αλλάξετε παραπάνω από 1-2 γραμμές κώδικα. Σημειώστε ότι το εκτελέσιμο περιμένει μία παράμετρο από τη γραμμή εντολών, τον αριθμό των σωμάτων που προσομοιώνονται. Τιμές της τάξης του 64K (65536) ή 128K (131072) είναι λογικές.

Σε 2^ο στάδιο, θα πρέπει να μεταφέρετε τον κώδικα στη GPU. Ο κώδικας θα πρέπει να είναι λειτουργικός για οποιοδήποτε μέγεθος δεδομένων (με την προϋπόθεση ότι επαρκεί η device memory της GPU). Βεβαιωθείτε ότι ο κώδικάς σας είναι σωστός συγκρίνοντας τις τελικές θέσεις των σωμάτων όπως αυτές υπολογίζονται μεταξύ CPU και GPU. Μπορείτε να θεωρήσετε ότι διαφορές πέρα από το 2^ο δεκαδικό είναι δικαιολογημένες.

Ο συγκεκριμένος κώδικας, παρότι μικρός, είναι αρκετά δεκτικός σε βελτιστοποιήσεις που έχουμε εξετάσει στο μάθημα. Ενδεικτικά αναφέρονται η κατανομή των δεδομένων στη μνήμη, προσεγγιστικές βελτιστοποιήσεις, tiling, unrolling κλπ. Μπορείτε να δείτε και αυτό το homework και ως διαγωνισμό, όπου "κερδίζει" η γρηγορότερη υλοποίηση. Εξυπακούεται ότι η υλοποίηση θα πρέπει πάνω απ' όλα να είναι ορθή. Για τις μετρήσεις χρόνου χρησιμοποιήστε 128K σώματα.

Για τη μεταγλώττιση σε CPU μπορείτε να χρησιμοποιήσετε την ακόλουθη εντολή (αντίστοιχα και για τους Intel Compilers αν προτιμάτε):

```
gcc -std=c99 -O3 -fopenmp -D_DEFAULT_SOURCE -o nbody nbody.c -lm
```

Για τη μεταγλώττιση σε GPU μπορείτε να χρησιμοποιήσετε την ακόλουθη εντολή:

```
nvcc -arch=sm_37 -I. -o nbody nbody.cu
```

Και στις 2 περιπτώσεις μπορείτε φυσικά να δώσετε και επιπλέον arguments στον compiler εάν απαιτείται, ανάλογα με την υλοποίησή σας.

Παράδοση:

Πρέπει να παραδώσετε:

- Τον τελικό κώδικα.
- Αναφορά στην οποία θα αναλύετε τη στρατηγική παραλληλοποίησης που ακολουθήσατε, τις βελτιστοποιήσεις που εφαρμόσατε και την επίδρασή τους στο χρόνο εκτέλεσης. Μην παραλείψετε να αναφέρετε και τυχόν βελτιστοποιήσεις που δοκιμάσατε χωρίς καλά αποτελέσματα στο χρόνο εκτέλεσης (ή στην ακρίβεια).

Δημιουργήστε ένα αρχείο .tar.gz με τα παραπάνω περιεχόμενα και όνομα <όνομα1>_<AEM1>_<όνομα2>_<AEM2>_lab5.tar.gz. Αποστέιλέτέ το στο e-class έως 23:59 της Κυριακής 12/1/2025.