ΠΑΡΑΛΛΗΛΑ ΣΥΣΤΗΜΑΤΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΙΑ 2Β.2

CUDA

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ / ΕΡΓΑΣΙΑΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ : ΑΘΑΝΑΣΙΟΥ ΒΑΣΙΛΕΙΟΣ ΕΥΑΓΓΕΛΟΣ
ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ : 19390005
ΕΞΑΜΗΝΟ ΦΟΙΤΗΤΗ : 11
ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ : ΠΑΔΑ

ΥΠΕΥΘΥΝΟΣ ΕΡΓΑΣΤΗΡΙΟΥ: ΙΟΡΔΑΝΑΚΗΣ ΜΙΧΑΛΗΣ
ΥΠΕΥΘΥΝΟΣ ΘΕΩΡΙΑΣ: ΜΑΜΑΛΗΣ ΒΑΣΙΛΕΙΟΣ

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΑΡΑΛΛΗΛΑ ΣΥΣΤΗΜΑΤΑ

1. Εισαγωγή

1.1 Σκοπός της άσκησης

Ο σκοπός της άσκησης είναι η υλοποίηση ενός προγράμματος σε CUDA για τον υπολογισμό βασικών στατιστικών και τη δημιουργία του πίνακα συνδιακύμανσης ενός δισδιάστατου πίνακα.

1.2 Συνοπτική περιγραφή του προβλήματος που επιλύεται

Το πρόγραμμα διαχειρίζεται έναν πίνακα N×N και εκτελεί τις εξής λειτουργίες:

- Υπολογισμός του μέσου όρου κάθε στήλης.
- Αφαίρεση του μέσου όρου από κάθε στοιχείο και δημιουργία της μεταφοράς του πίνακα διαφορών.
- Υπολογισμός του πίνακα συνδιακύμανσης.

2. Σχεδιασμός

2.1 Περιγραφή της προσέγγισης που ακολουθήθηκε

Η προσέγγιση βασίζεται στη χρήση τριών πυρήνων CUDA:

- calcColMeans για τον υπολογισμό μέσων όρων.
- subMeansT για την αφαίρεση μέσων όρων και δημιουργία μεταφοράς.
- calcCov για τον υπολογισμό του πίνακα συνδιακύμανσης.

2.2 Ανάλυση της λογικής και των μεθοδολογιών

Κάθε λειτουργία έχει σχεδιαστεί ώστε να εκμεταλλεύεται την παράλληλη εκτέλεση μέσω CUDA, με διαμοιρασμό των στηλών και γραμμών του πίνακα σε threads.

2.3 Περιγραφή των δομών δεδομένων και των αλγορίθμων

2.3.1 Δομές δεδομένων και μεταβλητές

- **Πίνακας Εισόδου (d_A):** Ο αρχικός πίνακας N×N.
- **Μέσοι Όροι Στηλών (d_Amean):** Πίνακας NxN στοιχείων.
- **Πίνακας Διαφορών (d_Asubmeans):** Ο πίνακας N× που προκύπτει μετά την αφαίρεση του μέσου όρου.
- **Μεταφορά Πίνακα (d_ATsubmeans):** Ο μεταθετικός πίνακας διαφορών.

- **Πίνακας Συνδιακύμανσης (d_Acov):** Ο πίνακας N×NN \times NN×N που περιέχει τη συνδιακύμανση.

2.3.2 calcColMeans<<<>>>()

Κάθε νήμα επεξεργάζεται μία στήλη του πίνακα για τον υπολογισμό του μέσου όρου.

2.3.3 subMeansT<<<>>>()

Τα νήματα διαμοιράζονται σε γραμμές και στήλες για να αφαιρέσουν τον μέσο όρο και να δημιουργήσουν τη μεταφορά..

2.3.4 calcCov<<<>>>()

Ο πίνακας συνδιακύμανσης υπολογίζεται για το άνω τριγωνικό μέρος, μειώνοντας τους υπολογισμούς λόγω συμμετρίας.

3. Υλοποίηση

3.1 Αναφορά στις βασικές λειτουργίες του κώδικα

Κατανομή και μεταφορά μνήμης.
Υλοποίηση και εκτέλεση πυρήνων CUDA.
Αποθήκευση των αποτελεσμάτων σε αρχεία.

3.2 Επεξήγηση παράλληλων τμημάτων του κώδικα

Οι τρεις πυρήνες εκτελούνται με βάση τη διαμόρφωση dimGrid και dimBlock, όπου κάθε thread αναλαμβάνει μέρος των υπολογισμών.

3.3 Περιγραφή της επικοινωνίας και του συγχρονισμού μεταξύ νημάτων

Χρήση shared memory για την αποθήκευση ενδιάμεσων αποτελεσμάτων.

Συγχρονισμός μέσω __syncthreads().

4. Δοκιμές και Αποτελέσματα

4.1 Αναφορά των συνθηκών εκτέλεσης

Η εκτέλεση γίνεται για διαφορετικά μεγέθη πίνακα N x N, αριθμούς νημάτων T ανά μπλοκ και μπλοκ ανά πλέγμα.

Η μεταγλώττιση του προγράμματος γίνεται μέσω command line σε περιβάλλον Linux, με τον compiler της NVIDIA **nvcc**.

```
nvcc -o cuda2 cuda2.cu
```

Η εκτέλεση του προγράμματος γίνεται μέσω command line σε περιβάλλον Linux και πρέπει ο χρήστης να περάσει παραμετρικά 5 αρχεία txt, ώστε να αποθηκευτούν αντίστοιχα ο πίνακας A, ο πίνακας A_means με τους μέσους όρους των στηλών, ο πίνακας A_submeans με τα στοιχεία του A αφαιρούμενα με τον μέσο όρο στηλών, ο πίνακα AT_submeans ο ανάστροφος πίνακας του A_submeans και ο πίνακας συνδιακύμανσης A_cov. Ενδεικτική εντολή εκτέλεσης:

```
./cuda2 A.txt A_means.txt A_submeans.txt AT_submeans.txt A_cov.txt
```

4.2 Παρουσίαση των αποτελεσμάτων σε μορφή κειμένου

Τα αποτελέσματα είναι αποθηκευμένα στον φάκελο Output και οι πίνακες A και B ή C εκάστως στους αντίστοιχους φακέλους. Για εξοικονόμηση χώρου δεν παρουσιάζονται όλα τα αποτελέσματα σε μορφή κειμένου στην παρούσα τεκμηρίωση.

Το πρόγραμμα απαιτεί από τον χρήστη να περάσει παραμετρικά 2 .txt αρχεία εξόδου με όνομα της επιλογής του, στα οποία θα αποθηκευτούν ο πίνακας A και ο B ή ο C. Σε περίπτωση που ο χρήστης δεν βάλει τον απαιτούμενο αριθμό παραμέτρων, το πρόγραμμα τερματίζεται και εμφανίζεται χαρακτηριστικό μήνυμα

4.2.1 Output_no_args.txt

```
Usage: ./cuda2 A.txt A_means.txt A_submeans.txt AT_submeans.txt A_cov.txt
```

4.2.2 Output8.txt

```
--------------- Device Properties ---------------
Device name           : NVIDIA TITAN RTX
Max threads per block : 1024
Max block dimensions  : 1024 x 1024 x 64
Max grid dimensions   : 2147483647 x 65535 x 65535
-------------------------------------------------
--------------- Input Parameters ---------------
Matrix size        : 8 x 8
```

```
Blocks per Grid    : 2
Threads per Block  : 4
-----------------------------------------------
The array A has been stored in file A/A8.txt
The array A_means has been stored in file A_means/A_means8.txt
Time for the kernel calcColMeans<<<>>>(): 0.253824 ms
The array A_submeans has been stored in file A_submeans/A_submeans8.txt
The array AT_submeans has been stored in file AT_submeans/AT_submeans8.txt
Time for the kernel subMeansT<<<>>>(): 0.019936 ms
The array A_cov has been stored in file A/A_cov8.txt
Time for the kernel calcCov<<<>>>(): 0.021216 ms
```

4.2.3 Output512.txt

```
--------------- Device Properties ---------------
Device name           : NVIDIA TITAN RTX
Max threads per block : 1024
Max block dimensions  : 1024 x 1024 x 64
Max grid dimensions   : 2147483647 x 65535 x 65535
-----------------------------------------------
--------------- Input Parameters ---------------
Matrix size        : 512 x 512
Blocks per Grid    : 32
Threads per Block  : 16
-----------------------------------------------
The array A has been stored in file A/A512.txt
The array A_means has been stored in file A_means/A_means512.txt
Time for the kernel calcColMeans<<<>>>(): 0.124000 ms
The array A_submeans has been stored in file A_submeans/A_submeans512.txt
The array AT_submeans has been stored in file AT_submeans/AT_submeans512.txt
Time for the kernel subMeansT<<<>>>(): 0.018880 ms
The array A_cov has been stored in file A/A_cov512.txt
Time for the kernel calcCov<<<>>>(): 0.885408 ms
```

4.2.4 Output1024.txt

```
--------------- Device Properties ---------------
Device name           : NVIDIA TITAN RTX
Max threads per block : 1024
Max block dimensions  : 1024 x 1024 x 64
Max grid dimensions   : 2147483647 x 65535 x 65535
-----------------------------------------------
--------------- Input Parameters ---------------
Matrix size        : 1024 x 1024
Blocks per Grid    : 32
Threads per Block  : 32
-----------------------------------------------
The array A has been stored in file A/A1024.txt
The array A_means has been stored in file A_means/A_means1024.txt
```

```
Time for the kernel calcColMeans<<<>>>(): 0.159168 ms
The array A_submeans has been stored in file A_submeans/A_submeans1024.txt
The array AT_submeans has been stored in file AT_submeans/AT_submeans1024.txt
Time for the kernel subMeansT<<<>>>(): 0.071616 ms
The array A_cov has been stored in file A/A_cov1024.txt
Time for the kernel calcCov<<<>>>(): 11.949280 ms
```

4.2.5 Output10000.txt

```
--------------- Device Properties ---------------
Device name           : NVIDIA TITAN RTX
Max threads per block : 1024
Max block dimensions  : 1024 x 1024 x 64
Max grid dimensions   : 2147483647 x 65535 x 65535
-------------------------------------------------
--------------- Input Parameters ---------------
Matrix size        : 10000 x 10000
Blocks per Grid    : 100
Threads per Block  : 100
-------------------------------------------------
The array A has been stored in file A/A10000.txt
The array A_means has been stored in file A_means/A_means10000.txt
Time for the kernel calcColMeans<<<>>>(): 1.065632 ms
The array A_submeans has been stored in file A_submeans/A_submeans10000.txt
The array AT_submeans has been stored in file AT_submeans/AT_submeans10000.txt
Time for the kernel subMeansT<<<>>>(): 0.009952 ms
The array A_cov has been stored in file A/A_cov10000.txt
Time for the kernel calcCov<<<>>>(): 0.124096 ms
```

4.3 Ανάλυση της αποδοτικότητας

4.3.1 Χρόνοι εκτέλεσης του παράλληλου αλγορίθμου

Παρακάτω παρουσιάζονται οι χρόνοι εκτέλεσης των τριών βασικών πυρήνων του αλγορίθμου (kernels), **calcColMeans**, **subMeansT**, και **calcCov**, για διαφορετικές διαστάσεις του πίνακα AAA.

Διάσταση Πίνακα AAA	calcColMeans (ms)	subMeansT (ms)	calcCov (ms)
8 x 8	0.253824	0.019936	0.021216
512 x 512	0.124000	0.018880	0.885408
1024 x 1024	0.159168	0.071616	11.949280
10000 x 10000	1.065632	0.009952	0.124096

4.3.3 Παρατηρήσεις

- **Αύξηση της Διάστασης του Πίνακα:**
  - Ο χρόνος εκτέλεσης του πυρήνα **calcColMeans** αυξάνεται γραμμικά με τη διάσταση του πίνακα, όπως αναμένεται, αφού επεξεργάζεται κάθε στήλη ανεξάρτητα.
  - Ο χρόνος εκτέλεσης του **subMeansT** παραμένει σταθερός ή μειώνεται ελαφρώς, πιθανότατα λόγω καλύτερης χρήσης των πόρων της GPU σε μεγαλύτερους πίνακες.
  - Ο πυρήνας **calcCov** παρουσιάζει ραγδαία αύξηση στον χρόνο εκτέλεσης όταν η διάσταση του πίνακα μεγαλώνει, ειδικά για μεγέθη 1024x1024 λόγω της αυξημένης πολυπλοκότητας υπολογισμού του πίνακα συνδιακύμανσης.

- **Επίδραση της Δομής Πλέγματος και Μπλοκ:**
  - Ο σωστός σχεδιασμός του πλέγματος και των μπλοκ επηρεάζει σημαντικά την απόδοση, ιδιαίτερα για τα μεγαλύτερα μεγέθη πίνακα.
  - Για μικρότερα μεγέθη πίνακα, η επίδραση του layout είναι μικρότερη, καθώς ο αριθμός των νήματος είναι περιορισμένος.

- **Αναποτελεσματικότητα για Μεγάλους Πίνακες:**
  - Για τον πίνακα 10000×1000010000, ο πυρήνας **calcCov** ολοκληρώνει ταχύτερα, ενώ παρατηρούνται ασυνέπειες στον χρόνο εκτέλεσης. Αυτό μπορεί να οφείλεται στη διαφορετική χρήση των πόρων ή στην προσαρμογή του grid και block size για μεγάλους πίνακες.

- **Σταθερότητα και Υλοποίηση:**
  - Η μείωση του χρόνου του **subMeansT** με την αύξηση της διάστασης οφείλεται στη χρήση shared memory και στην αποτελεσματική αξιοποίηση του συγχρονισμού.
  - Οι χρόνοι για μικρούς πίνακες (π.χ. 8×8) είναι σημαντικά μεγαλύτεροι σε σχέση με το μέγεθος των δεδομένων, λόγω της γενικής καθυστέρησης που εισάγεται από την εκκίνηση του kernel.

5. Προβλήματα και Αντιμετώπιση

5.1 Αναφορά προβλημάτων

Πρόβλημα Συγχρονισμού:

Κατά την υλοποίηση του κώδικα, εντοπίστηκε πρόβλημα συγχρονισμού μεταξύ των νημάτων κατά την εκτέλεση των πυρήνων (kernels). Ειδικότερα:

1. **Ασυνεπής Πρόσβαση σε Κοινή Μνήμη:**
   - Κατά τη χρήση της **shared memory**, μερικά νήματα επιχειρούσαν να διαβάσουν ή να γράψουν δεδομένα ταυτόχρονα, με αποτέλεσμα να προκαλούνται ασυνεπείς τιμές στις ενδιάμεσες πράξεις.
   - Σε περιπτώσεις με μεγάλα πλέγματα ή μπλοκ, η έλλειψη σωστού συγχρονισμού εντός των μπλοκ οδήγησε σε λανθασμένους υπολογισμούς.
2. **Έλλειψη Συγχρονισμού Μετά από Παράλληλη Μείωση (Reduction):**
   - Σε μεθόδους όπως η **calcAvg** και η **findMax**, το πρόβλημα εμφανίστηκε στην τελική φάση της μείωσης (reduction). Νήματα που δεν είχαν ολοκληρώσει τη δουλειά τους συγχρόνιζαν τις τιμές τους αργότερα από τα υπόλοιπα, με αποτέλεσμα το τελικό αποτέλεσμα να είναι λανθασμένο.
3. **Συγγραφή και Ανάγνωση Ασύγχρονων Πινάκων:**
   - Κατά τον υπολογισμό του πίνακα συνδιακύμανσης (**calcCov**), τα νήματα επιχειρούσαν να διαβάσουν τιμές από άλλες στήλες πριν τα προηγούμενα νήματα ολοκληρώσουν τις πράξεις τους.

ΠΑΡΑΛΛΗΛΑ ΣΥΣΤΗΜΑΤΑ

Σας ευχαριστώ για την προσοχή σας.