ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τεχνολογίας Υπολογιστών και Υπολογιστών
Εργαστήριο Μικροϋπολογιστών και Ψηφιακών Συστημάτων

Σχεδιασμός Ενσωματωμένων Συστημάτων
9ο Εξάμηνο ΗΜΜΥ

1η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΧΑΜΗΛΗ ΚΑΤΑΝΑΛΩΣΗ ΕΝΕΡΓΕΙΑΣ ΚΑΙ ΥΨΗΛΗ ΑΠΟΔΟΣΗ

1.1 Περιγραφή του υπό εξέταση αλγορίθμου

Σε αυτή την παράγραφο θα αναλυθεί ο αλγόριθμος Parallel Hierarchical One Dimensional Search (PHODS), ένας αλγόριθμος που ανήκει στην περιοχή των πολυμέσων. Ο αλγόριθμος PHODS είναι ένας αλγόριθμος εκτίμησης κίνησης (Motion Estimation), ο οποίος έχει στόχο να ανιχνεύσει τη κίνηση των αντικειμένων μεταξύ δύο διαδοχικών εικόνων (frame) του βίντεο. Οι αλγόριθμοι ανίχνευσης της κίνησης είναι καθοριστικοί για την συμπίεση βίντεο και αποτελούν τον πυρήνα κάθε εφαρμογής που περιέχει βίντεο. Ο PHODS έχει σαν είσοδο δύο διαδοχικές εικόνες από μια ακολουθία εικόνων βίντεο (διαστάσεων M×N), χωρίζει τις εικόνες σε block (διαστάσεων B×B pixel) σε κάθε μια από τις εικόνες αυτές και προσπαθεί να βρει την μετατόπιση του κάθε block από τη μια εικόνα (frame) στην επόμενη (frame). Για την εύρεση της μετατόπισης κάθε μπλόκ από το ένα frame στο επόμενο frame εκτελείται σύγκριση του μπλόκ από το πρώτο frame με όλα τα μπλόκ που βρίσκονται στην γύρω περιοχή από το επόμενο frame. Βάσει ενός συγκεκριμένου κριτηρίου επιλέγεται το μπλοκ εκείνο που εμφανίζει την μικρότερη τιμή στο κριτήριο.

Τον αρχικό αλγόριθμο PHODS σε γλώσσα C μπορείτε να τον βρείτε στα συνημμένα αρχεία της εργαστηριακής άσκησης, στο αρχείο `phods.c`.

1.2 Ζητούμενο 1ο – Loop Optimizations & Design Space Exploration

Στο **προσωπικό σας υπολογιστή** να πραγματοποιήσετε τα παρακάτω ερωτήματα:

1. Μέσω εντολών του Unix, να καταγράψετε τα χαρακτηριστικά του υπολογιστή στον οποίο πραγματοποιήσατε την εργαστηριακή άσκηση. Για κάθε χαρακτηριστικό να σημειώσετε και την αντίστοιχη εντολή με την οποία βρήκατε την ζητούμενη πληροφορία. Συγκεκριμένα, βρείτε και σημειώστε τα παρακάτω:

- Έκδοση λειτουργικού και έκδοση πυρήνα linux που χρησιμοποιείτε
- Ιεραρχία μνήμης του μηχανήματος, ξεκινώντας από L1 cache μέχρι και την RAM μνήμη.
- Αριθμός πυρήνων και συχνότητα ρολογιού των πυρήνων

2. Μετασχηματίστε τον δοθέντα κώδικα ώστε να μετράται ο χρόνος που χρειάζεται η συνάρτηση phods_motion_estimation για να εκτελεστεί. Η μέτρηση του χρόνου να γίνει με την συνάρτηση gettimeofday με ακρίβεια μικροδευτερολέπτων. Να εκτελέσετε τον κώδικα 10 φορές και να μετρήσετε το μέσο όρο, μέγιστο και ελάχιστο χρόνο εκτέλεσης. Παρουσιάστε τα αποτελέσματα της μέτρησης.

3. Με δεδομένη την ύπαρξη της υποδομής για την μέτρηση του χρόνου που χρειάζεται το κρίσιμο κομμάτι της εφαρμογής για να εκτελεστεί, εφαρμόστε μετασχηματισμούς στον κώδικα ώστε να μειωθεί ο χρόνος εκτέλεσης. Για κάθε μετασχηματισμό, να καταγραφεί η αντίστοιχη αλλαγή και να δικαιολογήσετε το λόγο για τον οποίο τον πραγματοποιήσατε. Αφού έχετε ολοκληρώσει τους μετασχηματισμούς στον κώδικα, να ξαναεκτελέστε τον κώδικα 10 φορές και να μετρήσετε μέσο όρο, μέγιστο και ελάχιστο χρόνο εκτέλεσης, αντίστοιχα με το ερώτημα 2. Να συγκρίνετε τους χρόνους εκτέλεσης με αυτούς του ερωτήματος 2.

4. Στον κώδικα που προέκυψε από το ερώτημα 3, εφαρμόστε Design Space Exploration αναφορικά με την εύρεση του βέλτιστου μεγέθους μπλοκ B (μεταβλητή στον κώδικα) για την αρχιτεκτονική του υπολογιστή σας. Η αναζήτηση να πραγματοποιηθεί με εξαντλητική αναζήτηση και μόνο για τιμές του B που είναι κοινοί διαιρέτες του M και του N. Για κάθε τιμή του B, το πρόγραμμα να εκτελεστεί 10 φορές και ως μετρική απόδοσης να οριστεί ο μέσος όρος του χρόνου εκτέλεσης σε αυτές τις 10 εκτελέσεις. Καταγράψτε την τιμή του B για την οποία πετυχαίνετε την καλύτερη απόδοση. Υπάρχει κάποια σχέση μεταξύ αυτής της τιμής και κάποιου χαρακτηριστικού του μηχανήματός σας από αυτά που καταγράψατε στο ερώτημα 1;

5. Στον κώδικα που προέκυψε από το ερώτημα 3, εφαρμόστε Design Space Exploration αναφορικά με την εύρεση του βέλτιστου μεγέθους μπλοκ θεωρώντας ορθογώνιο μπλοκ διαστάσεων B_x και B_y για την αρχιτεκτονική του υπολογιστή σας. Το μέγεθος του B_x να είναι διαιρέτης του N και το μέγεθος B_y να είναι διαιρέτης του M. Εφαρμόστε εξαντλητική αναζήτηση προκειμένου να βρείτε το βέλτιστο ζεύγος B_x, B_y το οποίο ελαχιστοποιεί τον χρόνο εκτέλεσης του προγράμματος. Για κάθε ζεύγος B_x και B_y, το πρόγραμμα να εκτελεστεί 10 φορές και ως μετρική απόδοσης να οριστεί ο μέσος όρος του χρόνου εκτέλεσης σε αυτές τις 10 εκτελέσεις. Καταγράψτε το ζεύγος B_x, B_y για το οποίο πετυχαίνετε την καλύτερη απόδοση. Προτείνετε κάποιο ευριστικό τρόπο περιορισμού της αναζήτησης αν θεωρείτε ότι υπάρχει.

6. Να απεικονίσετε τα αποτελέσματα των ερωτημάτων 2-5 σε ένα διάγραμμα

boxplot[1,2]. Να συγκρίνετε και να αναλύσετε τα πειραματικά αποτελέσματα που λάβατε.

1.3 Ζητούμενο 2ο : Αυτοματοποιημένη βελτιστοποίηση κώδικα

Σε αυτό το ζητούμενο καλείστε να κάνετε εγκατάσταση του εργαλείου αυτόματης βελτιστοποίησης κώδικα για πολλαπλές αρχιτεκτονικές Orio. Αναλυτικές οδηγίες για την εγκατάσταση και τον τρόπο χρήσης του εργαλείου μπορείτε να βρείτε στον εξής σύνδεσμο https://brnorris03.github.io/Orio/ . Προκειμένου να χρησιμοποιήσουμε το Orio, πρέπει να ακολουθήσουμε τα εξής βήματα:

1. Δημιουργούμε ένα αρχείο με τα κατάλληλα settings για το exploration. Αυτό το αρχείο περιλαμβάνει τόσο τις παραμέτρους για το design space exploration, όσο και το κομμάτι κώδικα προς μετασχηματισμό. Παράδειγμα τέτοιου αρχείου θα βρείτε στο **tables_orio.c** για loop unrolling transformation με αλγόρθμο exhaustive search.
2. Εκτελούμε το αρχείο αυτό με την εντολή

```
:~$ sudo orcc tables_orio.c
```

3. Αφού το εργαλείο πραγματοποιήσει το exploration, παράγεται το αρχείο _tables_orio.c, το οποίο περιλαμβάνει το μετασχηματισμένο κομμάτι κώδικα.
4. Τέλος, αυτό το κομμάτι κώδικα το ενσωματώνουμε στο αρχείο που θέλουμε (αντικαθιστώντας το παλιό κομμάτι κώδικα) και έχουμε τον τελικό κώδικα προς εκτέλεση.

Στον **προσωπικό σας υπολογιστή** να πραγματοποιήσετε και να απαντήσετε στα εξής ερωτήματα:

1. Αφού έχετε εγκαταστήσει επιτυχώς το εργαλείο, καλείστε να το χρησιμοποιήσετε για βελτιστοποίηση του κώδικα **`tables.c`**, του οποίου η λειτουργία αφορά απλές προσπελάσεις και πράξεις με πίνακες. Το κομμάτι κώδικα που αφορά αυτή την εργασία είναι κατάλληλα σχολιασμένο. Αρχικά, μετρήστε τον χρόνο εκτέλεσης του. Η μέτρηση του χρόνου να γίνει με την συνάρτηση gettimeofday με ακρίβεια μικροδευτερολέπτων. Να εκτελέσετε τον κώδικα 10 φορές και να μετρήσετε το μέσο όρο, μέγιστο και ελάχιστο χρόνο εκτέλεσης. Παρουσιάστε τα αποτελέσματα της μέτρησης.
2. Προσαρμόστε το αρχείο **tables_orio.c**, για να εφαρμόσετε τον μετασχηματισμό loop unrolling στο κομμάτι κώδικα του ερωτήματος 1. Για κάθε έναν από τους παρακάτω τρόπους που αναγράφονται στη σελίδα του Orio (Exhaustive, Randomsearch, Simplex) πραγματοποιήστε Design Space Exploration για να βρείτε το κατάλληλο loop unrolling factor για αυτό το κομμάτι κώδικα. Στη περίπτωση του αλγορίθμου simplex, χρησιμοποιήστε τις ακόλουθες παραμέτρους:

[1] https://en.wikipedia.org/wiki/Box_plot
[2] https://seaborn.pydata.org/generated/seaborn.boxplot.html

```
def search {
arg algorithm = 'Simplex';
arg time_limit = 10;
arg total_runs = 10;
arg simplex_local_distance = 2;
arg simplex_reflection_coef = 1.5;
arg simplex_expansion_coef = 2.5;
arg simplex_contraction_coef = 0.6;
arg simplex_shrinkage_coef = 0.7;
}
```

Σε κάθε μία περίπτωση καταγράψτε το unroll factor που προέκυψε από την εξερεύνηση. Διαφέρουν μεταξύ τους; Εξηγήστε το γιατί.

3. Αφού έχει παραχθεί ο μετασχηματισμένος κώδικας για κάθε έναν από τους τρεις τρόπους εξερεύνησης του χώρου σχεδιασμού, να αντικαταστήσετε το αντίστοιχο κομμάτι κώδικα στο αρχείο **`tables.c`**. Μετρήστε τον χρόνο εκτέλεσης του. Η μέτρηση του χρόνου να γίνει με την συνάρτηση gettimeofday με ακρίβεια μικροδευτερολέπτων. Να εκτελέσετε τον κώδικα 10 φορές και να μετρήσετε το μέσο όρο, μέγιστο και ελάχιστο χρόνο εκτέλεσης. Παρουσιάστε τα αποτελέσματα της κάθε μέτρησης. Συγκρίνετε και σχολιάστε αυτά τόσο μεταξύ τους, όσο και με τα αποτελέσματα της μη μετασχηματισμένης έκδοσης του κώδικα.

Υποσημείωση 1: Όλα τα προγράμματα να γίνουν compile με flag –Oo ώστε να μην γίνονται optimizations από τον gcc στον κώδικα σας.

Υποσημείωση 2: Στα ερωτήματα 2-5 του 1ου Ζητούμενου, η εκτέλεση των πειραμάτων θα πρέπει να γίνεται με χρήση κάποιου script που θα καλεί το τελικό πρόγραμμα για τα διάφορα configurations. Το script θεωρείται μέρος της άσκησης και θα παραδοθεί και αυτό με τον τελικό κώδικα. Μη αυτόματη εκτέλεση των προγραμμάτων θεωρείται μη αποδοτική και δεν θα πάρει τον πλήρη βαθμό.

Υποσημείωση 3: Σε περίπτωση σφαλμάτων κατά την εκτέλεση του εργαλείου Orio, πιθανόν να χρειάζεται να το εκτελέσετε με δικαιώματα sudo.