



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΗΡΥ 418 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΑΡΑΛΛΗΛΩΝ
ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017-18

Άσκηση 1 : Χρήση OpenMP και pthreads

Εαρινό εξάμηνο 2017-18
N. Αλαχιώτης

Περιγραφή

Στην άσκηση αυτή θα χρησιμοποιήσετε το OpenMP Application Protocol Interface (API) και ένα υποσύνολο των συναρτήσεων του POSIX threads standard (γνωστό και ως pthreads) για να επιταχύνετε τον υπολογισμό αποστάσεων Hamming μεταξύ δύο συνόλων συμβολοσειρών.

Η απόσταση Hamming μεταξύ δύο συμβολοσειρών ίδιου μεγέθους είναι ο αριθμός των θέσεων στις οποίες τα αντίστοιχα σύμβολα διαφέρουν.

Παράδειγμα

Σ1 : 10101

Σ2 : 10110

Απόσταση Hamming = 2

Γράψτε τον αρχικό (σειριακό) κώδικα σε C, ο οποίος θα δέχεται ως είσοδο τον αριθμό (m) των συμβολοσειρών του συνόλου A, τον αριθμό (n) των συμβολοσειρών του συνόλου B, και το μέγεθος (l) κάθε συμβολοσειράς. Στη συνέχεια, ο κώδικας θα δημιουργεί και θα αρχικοποιεί τις m και n συμβολοσειρές με τυχαίους χαρακτήρες. Τέλος, θα υπολογίζει όλες τις αποστάσεις Hamming, θα αποθηκεύει τα αποτελέσματα σε ένα πίνακα m*n, και θα τυπώνει στην οθόνη το συνολικό άθροισμα όλων των m*n αποστάσεων.

Αφού ολοκληρώσετε τον σειριακό κώδικα, να υπολογίσετε χρόνους εκτέλεσης για διαφορετικά μεγέθη συνόλων συμβολοσειρών: m = 1, 100, 1000, 10000, n = 1, 100, 1000, 10000, και μέγεθος συμβολοσειράς l = 10, 100, 1000.

Ένας απλός τρόπος μέτρησης του χρόνου σε Linux και bash shell είναι να χρησιμοποιήσετε την time:

time ./<όνομα εκτελέσιμου>

Χρήση OpenMP και Pthreads

Υπάρχουν (τουλάχιστον) τρία σημεία στον κώδικα σας όπου μπορείτε και πρέπει να εκμεταλευτείτε παραλληλισμό με διαφορετικό granularity, και κατά συνέπεια διαφορετικό computation-to-communication ratio.

Χρησιμοποιείτε αρχικά το OpenMP για να επιταχύνετε τον υπολογισμό του κώδικα σας. Τροποποιήστε τον κώδικα σας ώστε να δέχετε τον αριθμό των threads που θα δημιουργηθούν κατά την εκτέλεση του προγράμματος από τον χρήστη μαζί με τα m, n, και l.

Σε Linux, για να κάνετε compile για OpenMP, χρησιμοποιείτε τον gcc ως εξής:

```
gcc -fopenmp -o <όνομα εκτελέσιμου> <όνομα αρχείου.c>
```

Στη συνέχεια αντικαταστήστε τον OpenMP κώδικα με pthreads και επαναλάβετε τις ίδιες μετρήσεις. Οι βασικές προσθήκες που θα πρέπει να γίνουν προκειμένου να τρέξετε ένα πρόγραμμα που έχει pthreads είναι οι εξής:

1. Να προσθέσετε το header file `#include <pthread.h>` στο πρόγραμμα.
2. Για να τρέξετε το πρόγραμμα, χρησιμοποιήστε το switch `-lpthread`, για παράδειγμα:
gcc -lpthread <όνομα αρχείου.c> -o <όνομα εκτελέσιμου>

Παραδοτέα:

1. source code για τη σειριακή υλοποίηση
2. source code για τις παράλληλες υλοποιήσεις με OpenMP
3. source code για τις παράλληλες υλοποιήσεις με pthreads
4. run script που θα καλεί όλες τις υλοποιήσεις για τις ίδιες παραμέτρους εισόδου
5. Αναφορά (PDF) που να περιγράφει και να σχολιάζει τις παράλληλες υλοποιήσεις, να δείχνει τα αποτελέσματα σας σε σύγκριση με τον σειριακό κώδικα (χρόνοι εκτέλεσης και speedups για διαφορετικό αριθμό από threads), καθώς και συμπεράσματα και παρατηρήσεις για τη χρήση OpenMP και pthreads.

Πληροφορίες για το OpenMP:

- Tutorial βήμα-προς-βήμα μαζί με βασική ιστορία πάνω στο OpenMP:
<https://computing.llnl.gov/tutorials/openMP/>
- Έντυπο από το University of Edinburgh (μαζί με παράδειγμα πάνω στο game of life σε fortran)
<http://anusef.anu.edu.au/~dbs900/OpenMP/openmp.pdf>
- OpenMP specifications:
<http://www.openmp.org/mp-documents/OpenMP3.1.pdf>
- Το επίσημο site του OpenMP:
<http://www.openmp.org>

Πληροφορίες για τα pthreads:

- Tutorials βήμα-προς-βήμα μαζί με βασική ιστορία πάνω στα pthreads:

<https://computing.llnl.gov/tutorials/pthreads/>

Στην ίδια σελίδα, μπορείτε να δείτε το παρακάτω tutorial για την ανταλλαγή δεδομένων μεταξύ pthreads και του βασικού προγράμματος:

<https://computing.llnl.gov/tutorials/pthreads/#Management>

- <http://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>
- <http://www.ibm.com/developerworks/library/l-posix1/>