ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

12 Φεβρουαρίου 2018

ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΛΛΗΛΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ

Εξετάσεις Κανονικής Περιόδου Ακ. Έτους 2017-2018

Η εξέταση γίνεται με κλειστά βιβλία και σημειώσεις. Μπορείτε να έχετε μαζί σας μόνο μία κόλλα Α4. Διάρκεια εξέτασης $2^{3/4}$ ώρες.

Θέμα 1ο (60%)

Στη συνέχεια δίνεται ψευδοκώδικας που υλοποιεί την παραγοντοποίηση Cholesky, δηλαδή παραγοντοποιεί έναν πίνακα A στο γινόμενο $A = LL^*$, όπου L κάτω τριγωνικός και L^* ο συζυγής ανάστροφός του.

```
for (j = 0; j < n; j++) {
    s = 0;
    for (k = 0; k < j; k++) s+= L[j,k] * L[j,k];
    L[j,j] = sqrt(A[j, j] - s);
    for (i = j+1; i < n; i++) {
        m = 0;
        for (k = 0; k < j; k++) m+= L[i, k] * L[j, k];
        L[i,j] = (1.0 / L[j,j] * (A[i, j] - m));
    }
}
```

Α. Καλείστε να σχεδιάσετε και να υλοποιήσετε παράλληλο πρόγραμμα για την εκτέλεση του παραπάνω αλγορίθμου σε υπολογιστική πλατφόρμα με πολλαπλούς επεξεργαστές που υποστηρίζει το μοντέλο κοινού χώρου διευθύνσεων.

I. Δώστε το γράφο των εξαρτήσεων για *n = 4* αναδεικνύοντας το μέγιστο δυνατό παραλληλισμό. Επισημάνετε σε κάθε κόμβο του γράφου τον υπολογισμό που εκτελείται (π.χ. sqrt, div, mult, add/sub). (20%)
II. Δώστε έκφραση για τη μέγιστη δυνατή επιτάχυνση λαμβάνοντας υπόψη τη σχετική διαφορά ανάμεσα στις πράξεις, δηλ.: πρόσθεση/αφαίρεση = 1, πολλαπλασιασμός = 2, διαίρεση = 10, υπολογισμός τετραγωνικής ρίζας = 20. (5%)
III. Υλοποιήστε παράλληλη έκδοση χρησιμοποιώντας *parallel for.* (5%)
IV. Σχεδιάστε το γράφο που εκτελείται από την υλοποίηση του παραπάνω ερωτήματος και σχολιάστε αν υπάρχουν διαφορές σε σχέση με το γράφο που σχεδιάσατε στο ερώτημα Α.Ι. (5%)
V. Υλοποιείστε παράλληλη έκδοση χρησιμοποιώντας *fork – join*. (5%)

Β. Αντίστοιχα με το ερώτημα Α, καλείστε να σχεδιάσετε και να υλοποιήσετε παράλληλο πρόγραμμα για σύστημα που υποστηρίζει προγραμματιστικό μοντέλο ανταλλαγής μηνυμάτων.

I. Επισημάνετε τις εξαρτήσεις των δεδομένων πάνω στον πίνακα L. (7%)
II. Επιλέξτε μία από τις τέσσερις δυνατές κατανομές του πίνακα που προκύπτουν από τους συνδυασμούς κατά γραμμές/στήλες και συνεχόμενα/κυκλικά και αιτιολογήστε την επιλογή σας. (5%)
III. Δώστε ψευδοκώδικα υλοποίησης στο μοντέλο της ανταλλαγής μηνυμάτων. (15%)

Θέμα 2ο (25%)

Παρακάτω σας δίνεται ψευδοκώδικας εκτέλεσης ενός νήματος που υπολογίζει τους αριθμούς που ανήκουν στην ακολουθία Fibonacci στο διάστημα [*myStart*, *myEnd*]. Η υλοποίηση αυτή λαμβάνει υπόψη το ενδεχόμενο ταυτόχρονης εκτέλεσης άλλων νημάτων που αναλαμβάνουν τον υπολογισμό σε διαφορετικά διαστήματα. Ο υπολογισμός περιλαμβάνει έλεγχο για κάθε αριθμό του διαστήματος αν ανήκει στην ακολουθία Fibonacci και εισαγωγή των αριθμών Fibonacci σε πίνακα που είναι κοινός ανάμεσα στα νήματα *(array)*. Σημειώνονται τα εξής:

- Οι αριθμοί δεν χρειάζεται να είναι ταξινομημένοι στο πίνακα.
- Το μέγεθος του πίνακα είναι αρκετά μεγάλο (δεν χρειάζεται να ασχοληθείτε με τη διαχείριση μνήμης για τον πίνακα).
- Η υλοποίηση περιλαμβάνει ένα κλείδωμα (*lock*) για κάθε θέση του πίνακα και έναν κοινό μετρητή (*counter*).
- Οι αναγνώσεις και οι εγγραφές σε μία θέση μνήμης εκτελούνται ατομικά.

```
 0: shared int counter = 0;
 1: struct elem {
 2:    int number;
 3:    spinlock lock;
 4: }
 5: shared struct elem *array;
 6:
 7: /* per thread function */
 8: void FindFibonacciNumbers (int myStart, int myEnd) {
 9:    int h; //private variables
10:
11:    for (int j = myStart; j <= myEnd; j++)
12:       if (isFibonacciNumber(j) == true) { // compute
13:          while (trylock(array[counter].lock) == false)
14:             ;
15:          /* start critical section */
16:          h = counter;
17:          counter++;
18:          array[h].number = j;
19:          /* end critical section */
20:          unlock(array[h].lock);
21:       }
22: }
```

i) Γιατί χρειάζεται να αποθηκεύουμε στην τοπική μεταβλητή *h* την τιμή του *counter* (line 16); Θα μπορούσαμε να αντικαταστήσουμε τη μεταβλητή *h* με τον ίδιο τον *counter*; Αν ναι, δικαιολογήστε, αν όχι, αναφέρετε ένα αντιπαράδειγμα για το οποίο δε θα δούλευε αυτό. (3%)

ii) Ποια γραμμή του κώδικα δίνει έναυσμα στα νήματα να εισέλθουν στο κρίσιμο τμήμα; Περιμένουν την απελευθέρωση του κλειδώματος (line 20) για να προχωρήσουν; (2%)

iii) Δώστε εναλλακτικές υλοποιήσεις με:

- Coarse-grain locking (2%)
- Transactional Memory (2%)
- Atomic operations (5%).

iv) Στην περίπτωση της υλοποίησης με Transactional Memory, υπάρχει κίνδυνος για μαζικά "aborts" και αν ναι πού οφείλεται αυτός ο κίνδυνος; (3%)

v) Κρίνετε καλή επιλογή τη χρήση Transactional Memory ως μηχανισμό συγχρονισμού για το συγκεκριμένο κώδικα; Δικαιολογήστε. (3%)

vi) Προτείνετε αλλαγές που χρειάζονται στην υλοποίηση του συγκεκριμένου προγράμματος για βελτίωση της εκτέλεσης σε NUMA αρχιτεκτονική. (3%)

vii) Θεωρήστε εναλλακτική υλοποίηση κατά την οποία τα νήματα δεν εισάγουν στον πίνακα έναν αριθμό Fibonacci αμέσως μόλις τον βρουν, αλλά έχουν τη δυνατότητα να αποθηκεύουν τοπικά τα αποτελέσματα και να κάνουν πιο μαζικές εισαγωγές. Δώστε ψευδοκώδικα για μία τέτοια υλοποίηση και εξηγήστε αν αναμένετε να είναι πιο αποδοτική ή όχι. (7%)

Θέμα 3ο (25%)

Α. Σας ζητείται να διασυνδέσετε 72 υπολογιστικούς κόμβους σε τοπολογία fat tree. Έχετε στη διάθεσή σας διακόπτες τεχνολογίας OmniPath με 48 θύρες (48-port switches). Πόσους διακόπτες θα χρησιμοποιήσετε; Πόσα επίπεδα έχει το fat tree; Δώστε σχηματικά την τοπολογία. Χρησιμοποιούνται όλες οι διαθέσιμες θύρες κάθε διακόπτη; Τι μπορείτε να κάνετε με τις πλεονάζουσες θύρες, αν υπάρχουν; (10%)

Β. Σας ζητείται να διασυνδέσετε τους 8208 υπολογιστικούς κόμβους ενός υπερυπολογιστή σε τοπολογία fat tree. Τώρα έχετε στη διάθεσή σας διακόπτες δύο τύπων: (α) διακόπτες τεχνολογίας OmniPath με 48 θύρες (48-port switches) και (β) διακόπτες τεχνολογίας OmniPath με 768 θύρες (768-port switches). Μπορείτε να χρησιμοποιήσετε όσους διακόπτες επιθυμείτε, ωστόσο στο κατώτερο επίπεδο του fat tree (σύνδεση με τους κόμβους) μπορείτε να χρησιμοποιήσετε μόνο τους διακόπτες τύπου (α) (διακόπτες 48 θυρών). Πόσους διακόπτες θα χρησιμοποιήσετε; Πόσα επίπεδα έχει το fat tree; Δώστε σχηματικά την τοπολογία. Ποιο είναι το εύρος τομής του συστήματος; (10%)

Γ. Η τοπολογία του ερωτήματος (Β) είναι η τοπολογία του συστήματος Oakforest-PACS, του 9ου ταχύτερου υπερυπολογιστή στον κόσμο. Στο σύστημα αυτό, οι κόμβοι οργανώνονται σε racks, με 72 κόμβους ανά rack. Σχεδιάζοντας την τοπολογία, οι μηχανικοί του συστήματος μελέτησαν μία εναλλακτική τοπολογία. Συγκεκριμένα, η αρχική ιδέα ήταν να διασυνδέονται σε fat tree με πλήρες εύρος τομής[1] οι κόμβοι κάθε rack (intra-rack) με διακόπτες 48 θυρών, ακριβώς όπως στο ερώτημα (Α), και στη συνέχεια τα racks να διασυνδέονται μεταξύ τους με διακόπτες 768 θυρών. Περιγράψτε και σχεδιάστε αυτή την τοπολογία. Συγκρίνετε την τοπολογία αυτή με την τοπολογία του ερωτήματος (Β), ως προς τη διάμετρο και το εύρος τομής. Αν κάθε διακόπτης 48 θυρών κοστίζει 9,000€ και κάθε διακόπτης 768 θυρών κοστίζει 160,000€, συγκρίνετε τις δύο τοπολογίες ως προς το κόστος. Γιατί πιστεύετε ότι οι μηχανικοί επέλεξαν την τοπολογία του ερωτήματος (Β); (10%)

[1] Εύρος τομής είναι ο ελάχιστος αριθμός ακμών/συνδέσμων που κόβουμε, χωρίζοντας το δίκτυο στα δύο. Στην τοπολογία fat tree Ν κόμβων, το εύρος τομής πρέπει να είναι ίσο με Ν/2.

Σημείωση για το σχήμα βαθμολόγησης:

- Γράφετε με άριστα το 110.
- Το άθροισμα των βαθμών των επιμέρους ερωτημάτων κάθε θέματος είναι μεγαλύτερο από το μέγιστο βαθμό του θέματος. Ο βαθμός που θα πάρετε σε κάθε θέμα είναι ο min(*βαθμοί που συλλέξατε*, *μέγιστος βαθμός θέματος*).