

Παράλληλος Προγραμματισμός 2023

Προγραμματιστική Εργασία #2

Ζητούμενο

Δίνεται ο κώδικας για τη σειριακή εύρεση του μέγιστου και ελάχιστου στοιχείου ενός μονοδιάστατου πίνακα με αριθμούς τύπου double στα εξής αρχεία:

<https://mixstef.github.io/courses/parprog/minmax.c> (αρχείο C)

και

<https://mixstef.github.io/courses/parprog/minmax.cpp> (αρχείο C++).

Χρησιμοποιώντας τα αρχεία αυτά ως βάση, παραλληλίστε τον σειριακό κώδικα με τη βοήθεια α) των pthreads, β) του OpenMP και γ) του Intel TBB.

Προσοχή: Η εύρεση του μέγιστου και ελάχιστου θα πρέπει να γίνεται ταυτόχρονα, στο ίδιο «πέρασμα» του πίνακα. Δεν πρέπει να γίνεται π.χ. σε δύο ανεξάρτητες φάσεις ούτε σε δύο ανεξάρτητα threads (ένα το min και ένα το max)!

Υπόδειξη: Για κάθε ένα από τα ζητούμενα, συμβουλευτείτε τα αντίστοιχα παραδείγματα από τη σελίδα του εργαστηρίου.

Διαδικασία

1) Για την υλοποίηση των pthreads ξεκινήστε από το αρχείο C που σας δίνεται. Προσθέστε τις δομές και συναρτήσεις που απαιτούνται για την υλοποίηση της λύσης. Προσθέστε τον κώδικα διαχείρισης των threads (create-join) αντικαθιστώντας το σειριακό φορτίο ανάμεσα στις δύο κλήσεις του get_walltime(). Μην ξεχάσετε να προσθέσετε όποια δομή συγχρονισμού απαιτείται. Ο κώδικας στο main() για τη δέσμευση, αρχικοποίηση, έλεγχο και αποδέσμευση του πίνακα θα πρέπει να διατηρηθεί ως έχει.

2) Για την υλοποίηση με το OpenMP ξεκινήστε και πάλι από το αρχείο C που σας δίνεται. Προσθέστε τα κατάλληλα pragmas στο σειριακό φορτίο ανάμεσα στις δύο κλήσεις του get_walltime() για την υλοποίηση της λύσης. Ελέγξτε την αποκλειστική πρόσβαση σε κοινές μεταβλητές μέσω κρίσιμης περιοχής (critical) ή/και βεβαιωθείτε ότι τα threads συγχρονίζονται όπου απαιτείται. Ο κώδικας στο main() για τη δέσμευση, αρχικοποίηση, έλεγχο και αποδέσμευση του πίνακα θα πρέπει να διατηρηθεί ως έχει.

3) Για την υλοποίηση με το Intel TBB ξεκινήστε από το αρχείο C++ που σας δίνεται. Μετατρέψτε το σειριακό φορτίο ανάμεσα στις δύο κλήσεις του chrono::high_resolution_clock::now() στον κατάλληλο αλγόριθμο (parallel_...) του TBB. Ο κώδικας στο main() για τη δέσμευση, αρχικοποίηση, έλεγχο και αποδέσμευση του πίνακα θα πρέπει να διατηρηθεί ως έχει.

Υπόδειξη: Χρησιμοποιήστε blocked_range τύπου vector<double>::iterator για να προσπελάσετε το vector με iterators αρχής/τέλους (δείτε και το σχετικό παράδειγμα στο υλικό του εργαστηρίου).

Υπόδειξη 2: Μπορείτε να χρησιμοποιήσετε ένα std::pair<double,double> για να κρατάτε το μέγιστο και ελάχιστο στοιχείο ταυτόχρονα.

4) Ετοιμάστε **αναφορά σε μορφή pdf**, η οποία θα περιέχει:

- Σύντομη περιγραφή των τριών παραλλαγών κώδικα που δημιουργήσατε.
- Σύντομο σχολιασμό της απόδοσής τους σε σχέση με τον αρχικό κώδικα.
- Αναφορές σε πηγές (εκτός του υλικού του εργαστηρίου) που τυχόν χρησιμοποιήσατε.

5) Τοποθετήστε **την αναφορά σας** (αρχείο pdf) και **τον κώδικά σας** (τρία αρχεία C ή C++) σε **ένα (και μοναδικό) αρχείο zip**.

6) Ανεβάστε το αρχείο zip στο opencourses (**Εργασία 2**) έως και τη **Δευτέρα 5/6**.