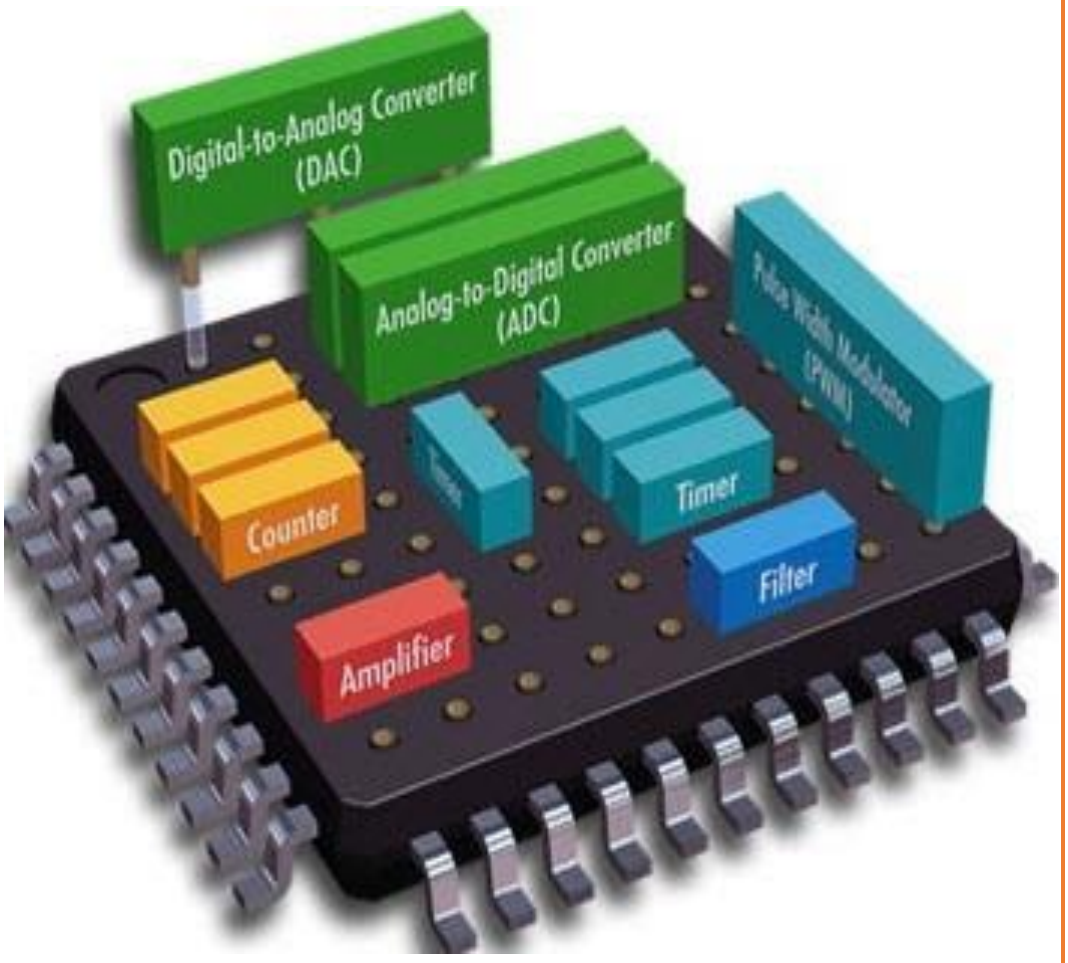
ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ



Περιεχόμενο

Σκοπός της συγκεκριμένης εργασίας είναι η τροποποίηση ενός αλγορίθμου producer-consumer έτσι ώστε να επιτευχθούν καλύτεροι χρόνοι με τη χρήση νημάτων.

Χριστιανού Γεωργία | 8419

gkchristi@ece.auth.gr

Εισαγωγή

Το ζητούμενο της συγκεκριμένης εργασίας είναι η υλοποίηση του προβλήματος producer-consumer μέσω μια ουράς fifo. Μέσω νημάτων μας ζητήθηκε να υλοποιήσουμε την επικοινωνία του producer με τον consumer για την αποστολή και λήψη αντίστοιχα, δεικτών σε συναρτήσεις.

Το σύστημα που χρησιμοποιήθηκε για την δημιουργία του κώδικα είναι ubuntu 20.10 σε επεξεργαστή Intel Core i5-8300H CPU με 4 πυρήνες (8 logical processors) και συχνότητα 2.3Ghz.

Για να τρέξει ο κώδικας χρησιμοποιήθηκε ο compiler gcc και πιο αναλυτικά η εντολή:

gcc -prthread prod-cons.c -lm

Ο κώδικας μπορεί να βρεθεί στον παρακάτω σύνδεσμο:

https://github.com/geochristi/embedded-systems

Η υλοποίηση

Ο αλγόριθμος που μας δινόταν είχε ήδη υλοποιημένη την επικοινωνία μεταξύ producer-consumer καθώς και την ουρά fifo στο struct της οποίας προστέθηκε ο πίνακας τύπου workFunction.

Το πρόγραμμα σε αυτό το σημείο δουλεύει για ίσο αριθμό νημάτων producer, consumer ο οποίος δηλώνεται στη μεταβλητή num_threads.

Πιο αναλυτικά, στην συνάρτηση main αρχικά δηλώνουμε τους δύο πίνακες νημάτων που θα χρησιμοποιήσουμε (tidpro[] και tidcon[]). Στη συνέχεια ελέγχουμε ότι η ουρά fifo εχει αρχικοποιηθεί και δημιουργούμε τα αντίστοιχα νήματα που καλούν την συνάρτηση producer και consumer.

Στη συνέχεια, στην συνάρτηση producer αποθηκεύονται οι αρχικές μετρήσεις χρόνου με το κάλεσμα της συναρτήσεως queueAdd και έπειτα, αρχικοποιούνται οι τιμές της workFunction για τη συγκεκριμένη λούπα. Πιο συγκεκριμένα, δηλώνεται σαν arg ένας αριθμό με βάση το άθροισμα της θέσης του τελευταίου στοιχείου της ουράς και του αριθμού της λούπας. Τελικά, επιλέγεται και την συνάρτηση που θα σταλθεί στον consumer μέσω της ουράς. Η συνάρτηση (void) *random_function() είναι η συνάρτηση που χρησιμοποιεί ο producer για να αναθέσει, τυχαία, στον consumer μια από τις 5 ενδεικτικές συναρτήσεις που έχουν δημιουργηθεί.

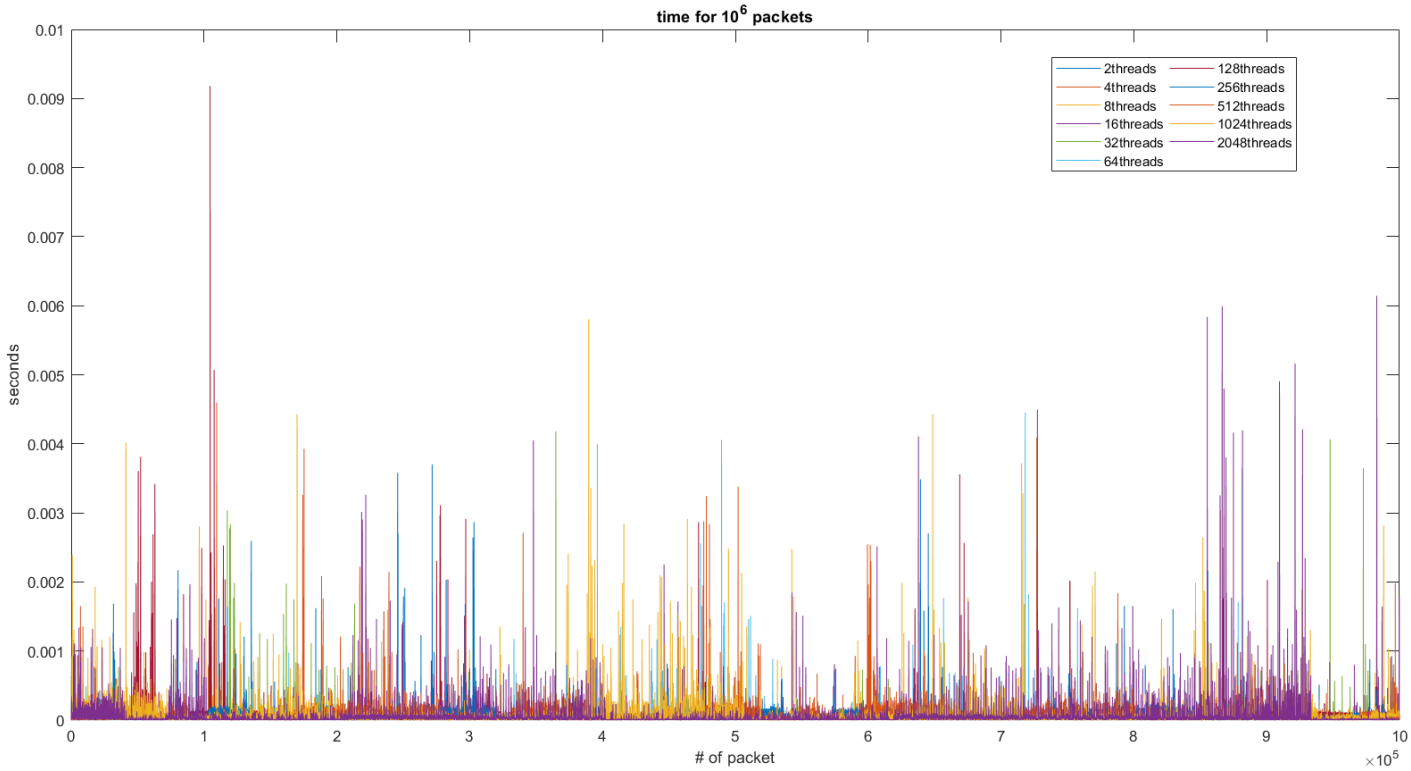
Ο consumer αρχικά μόλις λάβει το στοιχείο που του έχει αποσταλεί, παίρνει μέτρηση χρόνου και υπολογίζει το χρονικό περιθώριο από το συγκεκριμένο στοιχείο την ώρα που στάλθηκε από τον producer μέχρι την στιγμή που έφτασε σε αυτόν. Έπειτα, εκτελεί την συνάρτηση που του έχει ανατεθεί και στη συνέχεια διαγράφει το στοιχείο που έχει μόλις χρησιμοποίησει από την ουρά fifo.

Μετρήσεις και Αποτελέσματα

Για την εξαγωγή των αποτελεσμάτων περιορίσαμε τον αριθμό των επαναλήψεων της διαδικασίας αποστολής δεδομένων από τον producer στον consumer χρησιμοποιήθηκε ένας πίνακας αποθήκευσης χρόνων και στη συνέχεια οι χρόνοι εκτυπώθηκαν σε αρχεία txt.

Η μετρήσεις έγιναν για ίσο αριθμό threads από τους producer-consumer. Οι αριθμοί threads στα διαγράμματα αναφέρονται στον αριθμό threads μόνο της μιας συνάρτησης, με αποτέλεσμα τα συνολικά threads που χρησιμοποιούνται να είναι τα διπλάσια.

Στο παρακάτω διάγραμμα παρουσιάζονται οι μετρήσεις για 10^6 ανταλλαγές δεδομένων, σε ουρά 20 θέσεων, μεταξύ producer και consumer. Είναι ήδη φανερό ότι οι χρόνοι ανταλλαγής δεδομένων είναι κατά κύριο λόγο μικρότεροι από $0{,}002s$ για όλες τις μετρήσεις που πραγματοποιήθηκαν.



Για να γίνει καλύτερη ανάλυση των δεδομένων χρησιμοποιήθηκε η μέση τιμή η διάμεσος καθώς και η τυπική απόκλιση των αποτελεσμάτων για την κάθε ομάδα thread. Όπως είναι φανερό από το παρακάτω διάγραμμα, η ελαχιστοποίηση του χρόνου γίνεται για 64-1024 νήματα με μέσο χρόνο $2.6 * 10^{-5}$s.