

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάθημα: Κατανεμημένα και δικτυακά συστήματα.



Παράλληλος προγραμματισμός σε openmp και
openmpi.

Υπεύθυνος καθηγητής : κ. Μηνάς Δασυγένης

Στοιχεία φοιτητών:

Μηνάς Αδαμάντιος ΑΕΜ:2373 adamantmc@csd.auth.gr

Μουστάκας Αριστείδης ΑΕΜ:2380 moustakas@csd.auth.gr

Τσανακτσίδης Γιώργος ΑΕΜ:2418 gtsanakt@csd.auth.gr

Github repository: <https://github.com/adamantmc/distributed.git>

ΘΕΣΣΑΛΟΝΙΚΗ 06/04/2016

Για την συνολική εκπόνηση της εργασίας μας υλοποιήσαμε τα παρακάτω αρχεία:

--Makefile: Το Makefile υλοποιεί τις παρακάτω λειτουργίες.

1. **generate:** χρησιμοποιεί την ετικέτα generator.o η οποία δημιουργεί ένα αρχείο object για το generator.c. Έπειτα χρησιμοποιώντας τον compiler gcc μεταγλωτίζει (με όρισμα -c -Wall) αυτό το αρχείο object και παράγει το εκτελέσιμο αρχείο generator.out. Στην συνέχεια εκτελεί αυτό αρχείο με ορίσματα το όνομα του αρχείου που θα δημιουργήσει ("data.txt") και τον αριθμό των γραμμών που θέλουμε να παράγει(15.000.000 στην συγκεκριμένη περίπτωση).
2. **examine** : μεταγλωτίζει μέσω της εντολής mpicc το αρχείο examine.c (χρησιμοποιώντας σαν παράμετρους -c -Wall και -fopenmp) και παράγει το εκτελέσιμο αρχείο examine.out. Στην συνέχεια εκτελεί αυτό το αρχείο δίνοντας του σαν ορίσματα (-1 -1 data.txt -1 -1) δηλαδή κανένα όριο στον αριθμό συγκρούσεων που θα εκτελεστούν, κανένα όριο στον χρόνο εκτέλεσης του προγράμματος, θα διαβάσει τις συγκρούσεις από το αρχείο data.out, θα χρησιμοποιήσει όσα είναι τα διαθέσιμα threads του συστήματος και τέλος θα χρησιμοποιεί όλους τους διαθέσιμους επεξεργαστές του συστήματος.
3. **clean** : διαγράφει με την εντολή "rm" όλα τα object file και τα μεταγλωττισμένα αρχεία.
4. **all** : εκτελεί τις λειτουργίες generate και examine μαζί.

--generator.c: Ο generator δέχεται σαν ορίσματα το όνομα του αρχείου που θέλουμε να δημιουργήσει και των αριθμό των γραμμών που θέλουμε να γράψει σε αυτό. Κάθε γραμμή του αρχείου .txt που θα παραχθεί αποτελείται από 3 πραγματικούς αριθμούς που παράγονται με ψευδοτυχαίο τρόπο. Επίσης, κάθε γραμμή του αρχείου είναι ακριβώς 31 bytes και όταν τυχαίνει αριθμός μικρότερος του 10 τότε συμπληρώνουμε με ένα μηδενικό, π.χ. το 9.123456 θα γίνει 09.123456. Αυτό γίνεται ώστε να ξέρουμε ακριβώς το μέγεθος της κάθε γραμμής για να μπορούμε να διαβάζουμε πιο εύκολα το παραγόμενο αρχείο.

--examinempi.c (παραλληλοποιημένο)

Το examinempi.c περιέχει τον κώδικα που ελέγχει τις συντεταγμένες ενός αρχείου με σκοπό να δει αν ανήκουν σε κάποιο εύρος τιμών. Δέχεται σαν ορίσματα τον αριθμό των γραμμών (που ονομάζονται συγκρούσεις - collisions), τον μέγιστο χρόνο που μπορεί να τρέξει το πρόγραμμα, το όνομα του αρχείου, τον αριθμό των νημάτων που μπορεί να χρησιμοποιήσει καθώς και τον αριθμό των διεργασιών που μπορεί να χρησιμοποιήσει. Αρχικά το πρόγραμμα αρχικοποιεί τις mpi διεργασίες και ελέγχει αν τα ορίσματα είναι σωστά. Έπειτα ανάλογα με τα ορίσματα, ορίζονται:

- τα νήματα μέσω της συνάρτησης `omp_set_num_threads()`

- ο χρόνος εκτέλεσης (μεταβλητή `runTime`)
- η μεταβλητή `size` της οποίας η αρχική τιμή δίνετε από την συνάρτηση `MPI_Comm_size()` και έπειτα αλλάζει στην περίπτωση που ο χρήστης ζητήσει έναν συγκεκριμένο αριθμό διεργασιών
- ένας νέος `communicator` που περιλαμβάνει τις διεργασίες στις οποίες γίνεται `MPI_Reduce()` (πιο κάτω), που χρησιμεύει στην περίπτωση που δεν θέλουμε να κάνουμε `reduce` σε όλες τις διεργασίες αν ο χρήστης πει ότι θέλει λιγότερες διεργασίες από αυτές που έχει το πρόγραμμα
- μια μεταβλητή `FILE*` που δείχνει στο αρχείο του οποίου το όνομα έδωσε ο χρήστης σαν παράμετρο
- μια μεταβλητή `limit` που περιέχει τον αριθμό των γραμμών (συγκρούσεων) που ζητάει ο χρήστης να ελεγχθούν από το πρόγραμμα

Έπειτα ανοίγει μια `for` από `i=0` μέχρι `i<size`, δηλαδή για κάθε διεργασία. Έτσι ελέγχουμε σε ποιες διεργασίες θα ελεγχθούν οι συγκρούσεις. Αρχικά μοιράζουμε τις συγκρούσεις στις διεργασίες, διαιρώντας τον αριθμό των γραμμών στο αρχείο (ή τον αριθμό των γραμμών που δίνει ο χρήστης μέσω του `limit`). Αν η διαίρεση δίνει υπόλοιπο, τότε προσθέτουμε το υπόλοιπο στον αριθμό των γραμμών που θα διαβάσει η τελευταία διεργασία. Ο αριθμός των γραμμών που θα διαβάσει κάθε διεργασία αποθηκεύεται στην μεταβλητή `lines`.

Μετά δεσμεύουμε μνήμη για έναν `buffer` που θα περιέχει τις γραμμές. Το μέγεθος του `buffer` είναι `BUF_SIZE*STR_LEN`. Το `BUF_SIZE` δείχνει τον αριθμό των γραμμών, και το πολλαπλασιάζουμε με το `STR_LEN` που δείχνει το μέγεθος της κάθε γραμμής σε bytes (ή chars). Επίσης δηλώνουμε έναν πίνακα `nums` που περιέχει τους αριθμούς της κάθε γραμμής και έναν πίνακα `cords` που περιέχει την αναπαράσταση κάθε αριθμού σε χαρακτήρες.

Η ανάγνωση από το αρχείο εισόδου γίνεται μέσω μίας `while`. Πριν ξεκινήσει η `while` καλούμε την `checkTime`, αν ο χρόνος εκτέλεσης έχει περάσει τον επιτρεπτό χρόνο, τότε τερματίζει το πρόγραμμα, αλλιώς συνεχίζει στη `while`. Εκεί μέσα διαβάζουμε τα περιεχόμενα του αρχείου και τα αποθηκεύουμε στον `buffer` μέχρι να έχουν διαβαστεί όσες γραμμές ορίζει η `lines`. Στην κάθε διεργασία υπάρχει και η μεταβλητή `offset` που ορίζει το σημείο από το οποίο θα διαβάσει η διεργασία και μετά. Για παράδειγμα, για ένα αρχείο 400 γραμμών η πρώτη διεργασία θα έχει `offset 0` και θα διαβάσει μέχρι το 100, ενώ η δεύτερη θα έχει `offset 100` και θα διαβάσει μέχρι το 200. Αφού διαβαστούν οι γραμμές (καλείται πάλι η `checkTime`, εφόσον περάσει τον έλεγχο) γίνεται μια `for` (η οποία παραλληλοποιείται με `openmp`) όπου γίνεται το parsing των γραμμών και αποθηκεύονται οι αριθμοί σε γραμματοσειρές στον πίνακα `cords`. Καλείται η `atoi` για τον πρώτο αριθμό, και αν χρειαστεί (αν είναι μέσα στα όρια ο αριθμός αυτός) καλείται για τον επόμενο κ.ο.κ.. Αυτό συμβαίνει γιατί η `atoi` κοστίζει αρκετά, και δεν μπορούμε να την καλούμε για όλους τους αριθμούς. (με το που τελειώσει ο έλεγχος για κάθε τριάδα αριθμών καλείται η `checkTime`)

Τέλος, γίνονται 2 reduce, ένα για τον αριθμό των γραμμών που διαβάστηκαν, και 1 για τον αριθμό των γραμμών που περιέχουν συντεταγμένες μέσα στα όρια, και τυπώνονται τα αποτελέσματα.

Οι συναρτήσεις που χρησιμοποιούνται είναι οι εξής:

- `checkArgs(char **argv)`, ελέγχει τα ορίσματα που δίνει ο χρήστης
- `read(FILE *data, char *buffer, long lines, long offset)`, διαβάζει από ένα αρχείο, του οποίου ο file pointer είναι στο offset, lines γραμμές, και τις αποθηκεύει στον buffer, επιστρέφοντας τον αριθμό των γραμμών που διάβασε.
- `parse(char *data, int nums, char returnBuffer[3][TOK_LEN+1], long line)`, αποθηκεύει στο returnBuffer nums αριθμούς (3 στην προκειμένη περίπτωση) TOK_LEN ψηφίων ο καθένας (το +1 για το \0) από την γραμμή line
- `checkTime(double runTime, struct timespec start)`, ελέγχει αν ο χρόνος εκτέλεσης έχει ξεπεράσει τον επιτρεπτό χρόνο εκτέλεσης, στην οποία περίπτωση επιστρέφει 1, αλλιώς 0
- `getElapsedTime(struct timespec start, struct timespec end)`, επιστρέφει τον χρόνο μεταξύ 2 timespecs.

--examine.c (σειριακό)

Το `examine.c` περιέχει τον κώδικα που κάνει ακριβώς την ίδια δουλειά με το παραπάνω, απλά εκτελείται σειριακά. Οι διαφορές είναι ότι δέχεται 3 ορίσματα (δεν δέχεται τα 2 τελευταία που είναι ο αριθμός των νημάτων και των διεργασιών), δεν έχει καμία παραλληλοποίηση με `openmp` και δεν δημιουργούνται παραπάνω `mpi` διεργασίες.

Στιγμιότυπα από εκτελέσεις

Σειριακό, για 1 node και 4 ppn:

```
Total time: 8.200888  
Valid collisions: 2226396  
Total lines read: 15000000  
Lines read/sec: 1829070.241180
```

Μόνο OpenMP, για 1 node, 4 ppn και 8 threads:

```
Total time: 2.266062  
Valid collisions: 2226396  
Total lines read: 15000000  
Lines read/sec: 6619412.633339
```

Μόνο OpenMPI, για 2 nodes, 4 ppn:

```
Total time: 1.106703  
Valid collisions: 2226396  
Total lines read: 15000000  
Lines read/sec: 13553766.992632
```

OpenMP & OpenMPI, 2 nodes, 4 ppn, 4 threads:

```
Total time: 0.607685  
Valid collisions: 2226396  
Total lines read: 15000000  
Lines read/sec: 24683821.353134
```

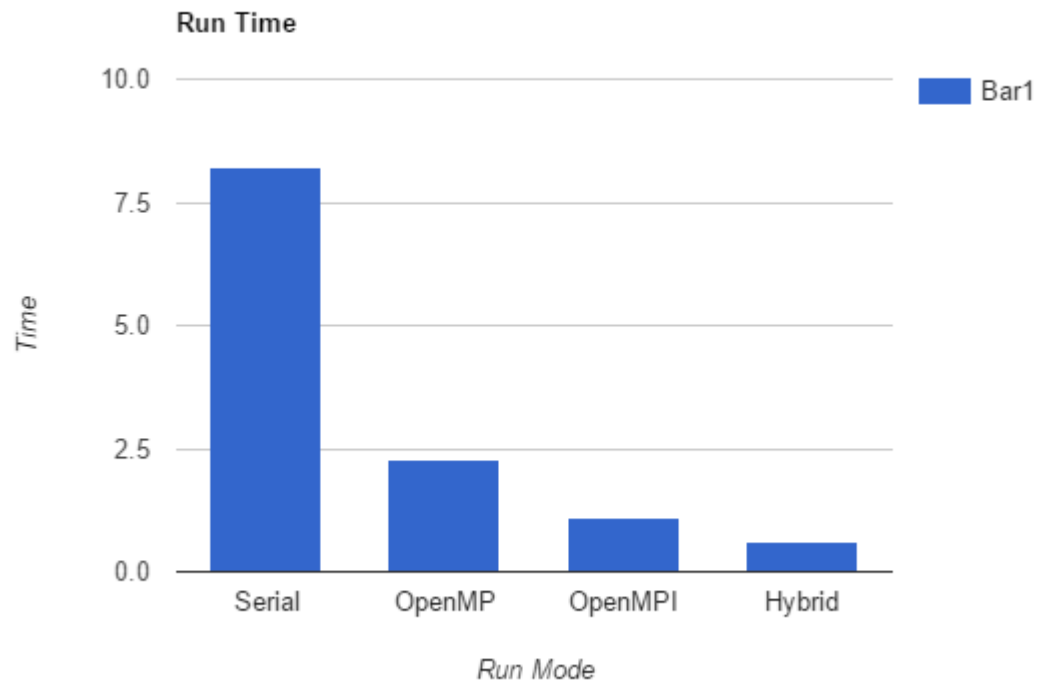
OpenMP & OpenMPI, 2 nodes, 4 ppn, 4 threads (επανάληψη):

```
Total time: 0.395242  
Valid collisions: 2226396  
Total lines read: 15000000  
Lines read/sec: 37951468.294869
```

Ο χρόνος εκτέλεσης κυμαίνεται από 0,6 sec έως και 0.39 sec σε κάποιες περιπτώσεις.

Γραφήματα

Χρόνος Εκτέλεσης:



Speedup:

