

ΚΑΤΑΝΕΜΗΜΕΝΑ ΚΑΙ ΔΙΑΔΥΚΤΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

6ο Εξάμηνο - Ακαδημαϊκό Έτος 2015-2016

ΕΚΦΩΝΗΣΗ 1ης ΕΡΓΑΣΙΑΣ

ΥΠΕΥΘΥΝΟΣ ΕΡΓΑΣΙΑΣ: Μηνάς Δασυγένης (mdasyg@ieee.org)
<http://arch.icte.uowm.gr/mdasyg>

Μέλος ΔΕΠ, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Δυτικής Μακεδονίας

Περιγραφή - Σκοπός

Το πείραμα του CERN για την ανίχνευση του σωματιδίου του Higgs έχει ξεκινήσει. Ένα θέμα που απασχολεί τους επιστήμονες είναι η ικανότητα της συστοιχίας εκατοντάδων ισχυρών υπολογιστών να επεξεργάζονται, στη πρώτη φάση του πειράματος, τα δεδομένα που στέλνουν οι ανιχνευτές. Τα δεδομένα αυτά προκύπτουν από τις συγκρούσεις των πρωτονίων και στην ουσία κάθε σύγκρουση αφήνει ένα αποτύπωμα στους ανιχνευτές υπό τη μορφή καρτεσιανών συντεταγμένων. Ορίζεται σαν σημείο αναφοράς το σημείο της σύγκρουσης $C(0,0,0)$ των πρωτονίων, ενώ οι ανιχνευτές που περιβάλλουν το σημείο C έχουν διαστάσεις 40×25 μέτρα. Ένα τέτοιο αποτύπωμα είναι π.χ.: $S1(24.120345, 21.056291, 16.563074)$ όπου σε συνδυασμό με την ταχύτητα πρόσκρουσης μπορούν να εξαχθούν σημαντικά συμπεράσματα. Οι φυσικοί σύμφωνα με τους υπολογισμούς τους αναμένουν να γίνουν 5.000.000 συγκρούσεις το δευτερόλεπτο.

Συμμετέχετε σε μια ομάδα που θα δοκιμάσει ένα υποσύνολο του cluster για την δυνατότητά του να επεξεργάζεται τα πρώτα κρίσιμα δεδομένα των ανιχνευτών. Έχετε στη διάθεσή σας πρόσβαση στην κατανεμημένη συστοιχία του Τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών (ΠΔΜ), για να λάβετε πειραματικά αποτελέσματα. Θα πρέπει να αναπτύξετε κατάλληλα πρόγραμμα σε OpenMP και σε OpenMPI για τη συνολική επεξεργασία αυτών των συγκρούσεων, δηλαδή πόσες συγκρούσεις αναλύονται στη μονάδα του χρόνου. Δεδομένου ότι απαιτούνται 5.000.000 / δευτερόλεπτο, ο σκοπός είναι να εξετάσετε αν το παράλληλο και κατανεμημένο πρόγραμμα που θα αναπτύξετε, μπορεί να ανταπεξέλθει σε αυτό το φόρτο. Η περιοχή ενδιαφέροντος για το συγκεκριμένο πείραμα είναι αυτή στην οποία η απόσταση R των αποτυπωμάτων από το σημείο σύγκρουσης είναι στην περιοχή τιμών: $[12, 30]$. Πρέπει και οι 3 συντεταγμένες (x,y,z) να βρίσκονται μέσα σε αυτήν την περιοχή. Οι τιμές των ορίων θα δοθούν ως `#define` μέσα στο C πρόγραμμα, ώστε να μπορούν να τροποποιηθούν.

Ζητούμενα

- Ανάπτυξη ενός προγράμματος C που θα χρησιμοποιεί OpenMP και OpenMPI και θα εξετάζει ένα αρχείο που θα περιέχει έναν μεγάλο αριθμό συγκρούσεων ως τριάδες συντεταγμένων (π.χ. 15000000 εκατομμύρια τριάδες) και θα αναφέρει το συνολικό χρόνο που χρειάστηκε για την εξέταση των συντεταγμένων, υπολογίζοντας στο τέλος το ρυθμό επεξεργασίας (συγκρούσεις ανά δευτερόλεπτο). Το πρόγραμμα θα δέχεται τις παρακάτω παραμέτρους γραμμής εντολών.
 - Παράμετρο μέγιστου αριθμού συγκρούσεων που θα εξεταστούν, ακέραιος αριθμός, με -1 να σηματοδοτεί ότι δεν υπάρχει όριο.
 - Παράμετρο μέγιστου χρόνου εκτέλεσης προγράμματος, ακέραιος αριθμός, με -1 να σηματοδοτεί ότι δεν υπάρχει όριο.
 - Παράμετρο που δείχνει το αρχείο που θα φέρει τις συντεταγμένες.
 - Παράμετρο που δηλώνει το μέγεθος των νημάτων openmp που θα χρησιμοποιηθούν ανά κόμβο, με -1 να δηλώνει τη χρησιμοποίηση όλων των διαθέσιμων νημάτων. Αν ο αριθμός υπερβαίνει τα διαθέσιμα νήματα, τότε χρησιμοποιούνται όλα τα νήματα.
 - Παράμετρο που δηλώνει το μέγεθος των διεργασιών openmpi που θα χρησιμοποιηθούν, με -1 να δηλώνει τη χρησιμοποίηση όλων των διαθέσιμων διεργασιών. Αν ο αριθμός υπερβαίνει τις διαθέσιμες διεργασίες, τότε χρησιμοποιούνται όλες οι διαθέσιμες διεργασίες.

Παραδείγματα εκτέλεσης:

```
./myprogram -1 -1 datafile.txt -1 -1
```

```
./myprogram 15000000 -1 datafile2 -1 10
```

- Μια που δεν είμαστε στο CERN και δεν έχουμε στη διάθεσή μας τους ανιχνευτές, για την παραγωγή των καρτεσιανών συντεταγμένων, θα πρέπει να αναπτύξετε ένα σειριακό πρόγραμμα που θα δημιουργεί το αρχείο με τις τυχαίες συντεταγμένες. Το πρόγραμμα θα δέχεται 2 παραμέτρους στη γραμμή εντολών: (α) το όνομα του αρχείου που θα δημιουργηθεί και θα αποθηκευτούν οι συντεταγμένες και (β) το πλήθος των συντεταγμένων που θα χρησιμοποιηθούν.

Παράδειγμα εκτέλεσης:

```
./mygenerator datafile 15000000
```

Θα πρέπει να χρησιμοποιήσετε το παρακάτω τμήμα κώδικα για δημιουργία τυχαίων συντεταγμένων.

```
int utime;  
long int ltime;  
int i;  
float cords[coordinate_index][3];  
ltime = time(NULL);  
utime = (unsigned int) ltime/2;  
srand(utime);
```

```
for (i=0;i<25000;i++) {
cords[i][0]=(float) 34*rand() / (RAND_MAX-1) ;
cords[i][1]=(float) 34*rand() / (RAND_MAX-1) ;
cords[i][2]=(float) 34*rand() / (RAND_MAX-1) ;
```

- Όπως κάθε project, θα πρέπει να συνοδεύεται από ένα αρχείο Makefile, το οποίο κατ ελάχιστο θα έχει τις παρακάτω ετικέτες και λειτουργίες:
 - **clean**: θα διαγράφει τα object file και τα μεταγλωττισμένα αρχεία
 - **generate**: θα μεταγλωττίζει και θα εκτελεί το generator και θα δημιουργεί ένα αρχείο datafile με 15000000 συντεταγμένες.
 - **examine**: θα μεταγλωττίζει και θα εκτελεί το πρόγραμμα ελέγχου με είσοδο το αρχείο datafile χωρίς κανένα περιορισμό.
 - **all**: θα εκτελεί τη μεταγλώττιση, το generate και το examine
- Όπως κάθε ομαδικό project, θα πρέπει η ανάπτυξη να υποστηρίζεται από ένα αποθετήριο προσβάσιμο δικτυακά. Θα πρέπει να δημιουργήσετε ένα λογαριασμό σε ένα δικτυακό αποθετήριο (αν δεν έχετε), π.χ. το github (for education/academic), και να κάνετε commit κάθε αλλαγή που γίνεται. Στο τέλος ο διδάσκων θα εξετάσει τα commits στο αποθετήριο από την έναρξη του project. Θα πρέπει να υπάρχουν commits από κάθε μέλος της ομάδας, ώστε να φαίνεται ότι όλοι έχουν εργασθεί στο project. Ο διδάσκων μπορεί να δώσει μειωμένη βαθμολογία σε μέλη της ομάδας που δεν έχουν συμμετέχει στην ανάπτυξη (σύμφωνα με τα commits).
- Η παράδοση του project θα συνοδεύεται από μια αναφορά 3 -6 σελίδων στην οποία θα υπάρχει η περιγραφή της εργασίας με δικά σας λόγια, η μεθοδολογία ανάπτυξης, διεύθυνση αποθετηρίου git (προσβάσιμη από το διδάσκοντα, να δώσετε πρόσβαση στο mdasyg@ieee.org), στιγμιότυπα εκτέλεσης, περιγραφή 2-3 σημείων του κώδικά σας και γραφήματα επιτάχυνσης (π.χ. μόνο OpenMPI, μόνο OpenMP και συνδυασμός των 2 για τιμές από 2 ροές εκτέλεσης, έως το μέγιστο μέγεθος).

Διευκρινήσεις

- Μπορείτε να επιλέξετε οποιαδήποτε τεχνική παραλληλοποίησης (master/slaves, peers, ...). Το cluster που θα σας δοθεί χρησιμοποιεί κοινό δικτυακό χώρο αποθήκευσης και μπορείτε να το εκμεταλλευτείτε για την ανάγνωση των τιμών.
- Η μέτρηση του χρόνου να γίνει με wall time POSIX (σύμφωνα με τις οδηγίες της 1ης διάλεξης).
- Ο μέγιστος βαθμός της εργασίας είναι 1 μονάδα.
- Η εργασία μπορεί να γίνει είτε ατομικά είτε σε ομάδες που θα αποτελούνται το πολύ από 4 άτομα. Η βαθμολόγηση θα είναι ίδια ανεξαρτήτως των μελών της ομάδας.
- Κάθε ομάδα θα πρέπει να παραδώσει ένα αρχείο zip με ονομασία AEM1_AEM2_AEM3_AEM4.zip.

- Μέσα στο zip θα πρέπει να υπάρχουν τα εξής:
 - Ο κώδικας της εργασίας, επαρκώς σχολιασμένος.
 - Σειριακός Κώδικας υλοποίησης του ζητούμενου αλγορίθμου.
 - Παράλληλος Κώδικας υλοποίησης του ζητούμενου αλγορίθμου (openmp + openMPI).
 - Κώδικας δημιουργίας του αρχείου συντεταγμένων.
 - Makefile
 - Αναφορά σε PDF.

Προτεινόμενος Χρονοπρογραμματισμός

1η εβδομάδα:

- Δημιουργία ομάδας.
- Εύρεση μηχανήματος Linux που θα γίνει η ανάπτυξη του κώδικα πριν σταλεί στη συστοιχία. Μπορεί να είναι και live USB linux ή VPS.
- Δημιουργία λογαριασμού στο github for education. Όλα τα μέλη της ομάδας κάνουν ένα sample commit ([http://zafora.ict.e.uowm.gr/GitHubSimpleGuide by Dasygenis.pdf](http://zafora.ict.e.uowm.gr/GitHubSimpleGuide_by_Dasygenis.pdf)).
- Δημιουργία generator σε C που δέχεται command line και δημιουργεί αρχείο με συντεταγμένες.
- Δημιουργία σειριακού προγράμματος σε C examine που ανοίγει το αρχείο και αναφέρει στο τέλος πόσες συντεταγμένες αναγνώστηκαν, πόσες ήταν εντός της περιοχής ενδιαφέροντος, και συνολικό ποσοστό.
- Τοποθέτηση συνάρτησης μέτρησης χρόνου wall time POSIX στο examine και εκτύπωση σχετικού μηνύματος ρυθμού επεξεργασίας (coordinates/second). (<http://arch.ict.e.uowm.gr/courses/parallel/posixwall.pdf>).
- Επανάληψη OpenMP από τις διαφάνειες του διδάσκοντα ([http://arch.ict.e.uowm.gr/courses/parallel/lab-openmp oc.pdf](http://arch.ict.e.uowm.gr/courses/parallel/lab-openmp_oc.pdf)).

2η εβδομάδα:

- Ο διδάσκων θα δώσει πρόσβαση σε κάθε ομάδα στη συστοιχία για πραγματικές μετρήσεις.
- Τροποποίηση σειριακού κώδικα και παραλληλοποίηση κατά OpenMP. Δε μας ενδιαφέρουν τα όρια οπότε χρησιμοποιούμε όλα τα νήματα. Δοκιμή για διάφορα νήματα από 2 έως 24.
- Επανάληψη openMPI από τις διαφάνειες του διδάσκοντα ([http://arch.ict.e.uowm.gr/courses/parallel/lab-mpi-01 oc.pdf](http://arch.ict.e.uowm.gr/courses/parallel/lab-mpi-01_oc.pdf) έως [http://arch.ict.e.uowm.gr/courses/parallel/lab-mpi-05 oc.pdf](http://arch.ict.e.uowm.gr/courses/parallel/lab-mpi-05_oc.pdf))

3η εβδομάδα

- Τροποποίηση σειριακού κώδικα με τις συναρτήσεις OpenMPI. Δε μας ενδιαφέρουν τα όρια οπότε χρησιμοποιούμε όλες τις διεργασίες. Εκτύπωση αποτελεσμάτων. Δοκιμή για διάφορα ranks από 2 έως 24.

4η εβδομάδα

- Μίξη των προγραμμάτων κατά OpenMP και OpenMPI. Στον κώδικα για τα OpenMPI τοποθετούμε κατάλληλα τον κώδικα για OpenMP. Τοποθέτηση ορίων.
- Συγγραφή αναφοράς .
(http://arch.icte.uowm.gr/mdasyg/pdfs/writing_diploma_thesis.pdf).

5η εβδομάδα

- Παράδοση της εργασίας (ως Κυριακή 3/4/2016 23:59) με τον τρόπο που θα ανακοινωθεί .
- Δυνατότητα παράτασης 1ας εβδομάδας σε φοιτητές που έχουν ολοκληρώσει πάνω από 60% και το ζητήσουν ως την Κυριακή 3/4.

Βαθμολογία

- 30% της βαθμολογίας αφορά την παραλληλοποίηση σε OpenMP.
- 30% της βαθμολογίας αφορά την παραλληλοποίηση σε OpenMPI.
- 10% της βαθμολογίας αφορά το Makefile.
- 10% της βαθμολογίας αφορά τη χρήση ορίων.
- 5% της βαθμολογίας αφορά τη σωστή μέτρηση του χρόνου κατά Posix Wall Clock.
- 5% της βαθμολογίας αφορά τη σωστή χρήση του github (κατ ελάχιστο 1 commit / ημέρα ενασχόλησης).
- 10% της βαθμολογίας αφορά τη σωστή αναφορά χωρίς γραμματικά και συντακτικά λάθη.

Bonus +1 μονάδα, αν κάποια ομάδα υλοποιήσει εκτός από τα παραπάνω και τον ίδιο κώδικα σε Nvidia/Cuda Threads (ημερομηνία παράδοσης μόνο για Cuda threads 9/5).