



## Εργασία 3 (υποχρεωτική) – Προγραμματισμός με MPI

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2025 – 2026

(ΕΚΦΩΝΗΣΗ) ΣΑΒΒΑΤΟ 20 ΔΕΚΕΜΒΡΙΟΥ 2025

(ΠΑΡΑΔΟΣΗ ΣΤΟ ECLASS ΜΕΧΡΙ) ΠΑΡΑΣΚΕΥΗ 16 ΙΑΝΟΥΑΡΙΟΥ 2026

### Πληροφορίες για τις Υποχρεωτικές Εργασίες του μαθήματος

- Κάθε ομάδα μπορεί να αποτελείται από 1 ή 2 φοιτητές. Όλα τα μέλη της ομάδας πρέπει να έχουν ισότιμη συμμετοχή και να γνωρίζουν τις λεπτομέρειες της υλοποίησης της ομάδας.
- Για την εξεταστική Σεπτεμβρίου δε θα δοθούν άλλες εργασίες. Τον Σεπτέμβριο εξετάζεται μόνο το γραπτό.
- Επίσημη υπολογιστική πλατφόρμα του μαθήματος είναι το δίκτυο των υπολογιστών του Τμήματος με λειτουργικό σύστημα Linux Ubuntu (linux01.di.uoa.gr έως linux30.di.uoa.gr).
- Μαζί με τον κώδικά σας καλείστε να υποβάλετε και τα σχετικά αρχεία Makefile.
- Στην αναφορά σας καλείστε να δώσετε πληροφορίες σχετικά με το όνομα του υπολογιστικού συστήματος που χρησιμοποιείτε, καθώς επίσης και το μοντέλο επεξεργαστή, τον αριθμό των πυρήνων, την έκδοση του λειτουργικού συστήματος, και την έκδοση του μεταγλωττιστή. Για τα πειραματικά δεδομένα που παρουσιάζετε, καλείστε να αναφέρετε ρητά στα εκάστοτε σημεία της αναφοράς τις εισόδους που χρησιμοποιήσατε. Καλείστε να εκτελέσετε κάθε πείραμα (το οποίο ορίζεται ως η εκτέλεση ενός προγράμματος για συγκεκριμένο αριθμό νημάτων και παραμέτρους εισόδου) πολλές φορές (για παράδειγμα, 4 φορές) και να παρουσιάσετε τον μέσο όρο των αποτελεσμάτων (σύσταση: δημιουργήστε scripts για την εκτέλεση των πειραμάτων, ακόμα και για την επεξεργασία των αποτελεσμάτων και την δημιουργία γραφημάτων).
- Προαιρετικά, μπορείτε να υποβάλετε και τα scripts που χρησιμοποιήσατε για να τρέξετε τα πειράματα και να δημιουργήσετε τα σχετικά γραφήματα.
- Καλείστε να προσεγγίσετε την κάθε άσκηση στην αναφορά σας ως εξής: περιγραφή προβλήματος, σύντομη περιγραφή της λύσης σας, παράθεση πειραματικών αποτελεσμάτων (χρήση πινάκων ή γραφημάτων), και σχολιασμός αποτελεσμάτων.
- Σε περίπτωση αντιγραφής θα μηδενίζονται όλες οι ομάδες που μετέχουν σε αυτή.
- Η παράδοση της Εργασίας πρέπει να γίνει μέχρι τα μεσάνυχτα της προθεσμίας ηλεκτρονικά και μόνο στο eclass (να ανεβάσετε ένα μόνο αρχείο zip ή rar με την αναφορά σας σε PDF και τον κώδικά σας). Μην περιμένετε μέχρι την τελευταία στιγμή.

### Άσκηση 3.1 (35%)

Σε αυτή την άσκηση καλείστε να γράψετε ένα πρόγραμμα MPI το οποίο υπολογίζει τον πολλαπλασιασμό πολυωνύμων. Πιο συγκεκριμένα, καλείστε να γράψετε ένα πρόγραμμα στο οποίο η διεργασία 0 δημιουργεί δύο τυχαία πλήρη πολυώνυμα η βαθμού οι συντελεστές των οποίων είναι ακέραιοι μη-μηδενικοί αριθμοί, και στέλνει στις υπόλοιπες διεργασίες μόνο τα απαραίτητα δεδομένα για να πραγματοποιηθεί ο παράλληλος υπολογισμός. Στο τέλος, η διεργασία 0 συγκεντρώνει το τελικό αποτέλεσμα και το τυπώνει.

Το πρόγραμμά σας θα λαμβάνει σαν όρισμα τον βαθμό  $n$  του κάθε πολυωνύμου. Σαν έξοδο, το πρόγραμμά σας θα τυπώνει: (i) τον χρόνο αποστολής των δεδομένων από τη διεργασία 0 στις υπόλοιπες διεργασίες, (ii) τον χρόνο εκτέλεσης του παράλληλου υπολογισμού, και (iii) τον χρόνο λήψης των αποτελεσμάτων των διεργασιών από τη διεργασία 0, και (iv) τον συνολικό χρόνο εκτέλεσης που περιλαμβάνει όλα τα προηγούμενα βήματα (χωρίς τον χρόνο δέσμευσης και αρχικοποίησης των πολυωνύμων).

Συγκρίνετε την απόδοση του παράλληλου προγράμματος σας όσον αφορά τα επιμέρους βήματα της εκτέλεσης (εκτός του χρόνου δημιουργίας και αρχικοποίησης των πολυωνύμων) για διάφορους βαθμούς πολυωνύμων  $n$  (για παράδειγμα,  $10^5$  ή  $10^6$ ) και για διαφορετικό αριθμό MPI διεργασιών, καθώς επίσης και όσον αφορά τον συνολικό χρόνο εκτέλεσης σε σχέση με τη σειριακή υλοποίηση. Στην άσκηση αποφύγετε την επίλυση μέσω Fast Fourier Transform. Τέλος, συγκρίνετε την απόδοση του παράλληλου MPI προγράμματος σας όταν τρέχει σε έναν κόμβο με διαφορετικό αριθμό νημάτων με τη γρηγορότερη εκδοχή κοινόχρηστης μνήμης (Pthreads/OpenMP) που υλοποιήσατε στις Ασκήσεις 1.1/2.1. Παρατηρείτε διαφορές στην επίδοση;

### Άσκηση 3.2 (65%)

Σε αυτή την άσκηση καλείστε να γράψετε ένα πρόγραμμα MPI που να υλοποιεί αποδοτικά τον πολλαπλασιασμό αραιού πίνακα με διάνυσμα (sparse matrix-vector multiplication), με τη χρήση του Compressed Sparse Row (CSR) σχήματος αναπαράστασης.

Πιο συγκεκριμένα, η διεργασία 0 αρχικά θα κατασκευάζει με τυχαίο τρόπο τον πίνακα και το διάνυσμα με ακέραιους αριθμούς φροντίζοντας να υπάρχει κατάλληλο ποσοστό μηδενικών στοιχείων στον πίνακα. Στη συνέχεια, θα

κατασκευάζει την αναπαράσταση CSR του πίνακα και θα στέλνει στις υπόλοιπες διεργασίες μόνο τα απαραίτητα δεδομένα για να πραγματοποιηθεί ο παράλληλος υπολογισμός. Ο παράλληλος υπολογισμός είναι ο πολλαπλασιασμός του πίνακα με το διάνυσμα για έναν συγκεκριμένο αριθμό επαναλήψεων (το διάνυσμα του αποτελέσματος της κάθε επανάληψης είναι το διάνυσμα εισόδου της επόμενης επανάληψης). Στο τέλος, η διεργασία ο συγκεντρώνει το τελικό αποτέλεσμα και το τυπώνει. Τέλος, το πρόγραμμά σας θα πραγματοποιεί τον παράλληλο πολλαπλασιασμό μέσω MPI με τη χρήση της πυκνής μορφής για τον ίδιο αριθμό επαναλήψεων και θα τυπώνει τον αντίστοιχο χρόνο εκτέλεσης.

Το πρόγραμμά σας θα λαμβάνει σαν ορίσματα: (i) τον αριθμό των στοιχείων γραμμής/στήλης του πίνακα (θεωρείστε ότι ο πίνακας είναι τετραγωνικός), (ii) το ποσοστό μηδενικών στοιχείων του πίνακα (sparsity), και (iii) τον αριθμό επαναλήψεων του πολλαπλασιασμού.

Σαν έξοδο, το πρόγραμμά σας θα τυπώνει: (i) τον χρόνο κατασκευής της αναπαράστασης CSR, (ii) τον χρόνο αποστολής των δεδομένων από τη διεργασία ο στις υπόλοιπες διεργασίες, και (iii) τον χρόνο εκτέλεσης του παράλληλου υπολογισμού με CSR αναπαράσταση, (iv) τον συνολικό χρόνο εκτέλεσης με CSR αναπαράσταση που περιλαμβάνει όλα τα προηγούμενα βήματα (δηλαδή χωρίς τον χρόνο δέσμευσης και αρχικοποίησης του αρχικού πίνακα σε πυκνή μορφή και του διανύσματος), και (v) τον συνολικό χρόνο εκτέλεσης με την πυκνή αναπαράσταση.

Πειραματιστείτε με διαφορετικές διαστάσεις του πίνακα/διανύσματος (για παράδειγμα,  $10^3$  έως  $10^4$  στοιχεία ανά γραμμή), διαφορετικά ποσοστά μηδενικών στοιχείων (για παράδειγμα, 0% έως 99%), διαφορετικό αριθμό επαναλήψεων (για παράδειγμα, 1 έως 20) και διαφορετικό αριθμό MPI διεργασιών. Αρχικά επικεντρωθείτε στο πώς κλιμακώνει η παράλληλη υλοποίηση σας με χρήση της CSR αναπαράστασης καθώς αλλάζει ο αριθμός των διεργασιών σε σχέση με την αντίστοιχη σειριακή εκτέλεση. Στη συνέχεια, επικεντρωθείτε στη σύγκριση της επίδοσης μεταξύ της αναπαράστασης CSR και της πυκνής αναπαράστασης. Για ποιες τιμές των παραμέτρων παρατηρείτε καλύτερο χρόνο επίδοσης με τη χρήση της αναπαράστασης CSR; Στην αναφορά σας καλείστε να παραθέσετε γραφήματα (ή πίνακες) που να δείχνουν τον αντίκτυπο των παραπάνω παραμέτρων και να σχολιάσετε τα αποτελέσματα, τόσο όσον αφορά τα επιμέρους βήματα της εκτέλεσης, καθώς επίσης και όσον αφορά τον συνολικό χρόνο εκτέλεσης.

### Άσκηση 3.3 (προαιρετική)

Τροποποιήστε κατάλληλα το πρόγραμμα που γράψατε στην Άσκηση 3.2 ώστε να χρησιμοποιεί υβριδικά το MPI (για εκμετάλλευση της παραλληλίας των κόμβων) και το OpenMP (για εκμετάλλευση της παραλληλίας των πυρήνων που βρίσκονται μέσα σε κάθε κόμβο). Συγκρίνετε την απόδοση αυτής της παράλληλης υλοποίησης σε σχέση με την υλοποίησή σας της Άσκησης 3.2 κρατώντας την συνολική χρήση πόρων (κόμβοι και πυρήνες ανά κόμβο) σταθερή (για παράδειγμα, υλοποίηση MPI με 2 κόμβους και 4 διεργασίες ανά κόμβο vs υβριδική υλοποίηση MPI+OpenMP με 2 κόμβους και 4 νήματα ανά κόμβο).