

## Σχεδιασμός, Ανάπτυξη και Αξιολόγηση Παράλληλων Προγραμμάτων σε MPI, Υβριδικό Mpi+OpenMp-Cuda που υλοποιούν την Jacobi with successive over-relaxation

Σκοπός της εργασίας είναι να αναπτύξετε και να αξιολογήσετε παράλληλα προγράμματα σε MPI, υβριδικό MPI+OpenMp και CUDA που υλοποιούν την jacobi μέθοδο (με τη παραλλαγή successive over-relaxation) για να λύσει αριθμητικά την εξίσωση Poisson.

$$(\nabla^2 - \alpha)u = \frac{d^2}{dx^2}u + \frac{d^2}{dy^2}u - \alpha u = f$$

Στον κατάλογο στην ΑΡΓΩ /etc/skel2/Project υπάρχουν 2 υποκατάλογοι, ο seqJacobi και ο challenge. Ο κατάλογος seqJacobi περιέχει πηγαίο ακολουθιακό πρόγραμμα, που υλοποιεί την μέθοδο και υπολογίζει τις τιμές του πίνακα  $u$  στο διάστημα  $[-1, 1] \times [-1, 1]$  και το σφάλμα σε σχέση με την αναλυτική λύση  $u(x,y) = (1-x^2)(1-y^2)$ . Θα το χρησιμοποιήσετε ως βάση για παραλληλοποίηση ακολουθώντας τις συνοδευτικές οδηγίες. Ο κατάλογος challenge περιέχει εκτελέσιμο MPI κώδικα με το οποίο θα συγκρίνετε την δική σας υλοποίηση. Και οι δυο κατάλογοι περιέχουν αρχείο εισόδου input και PBS script.

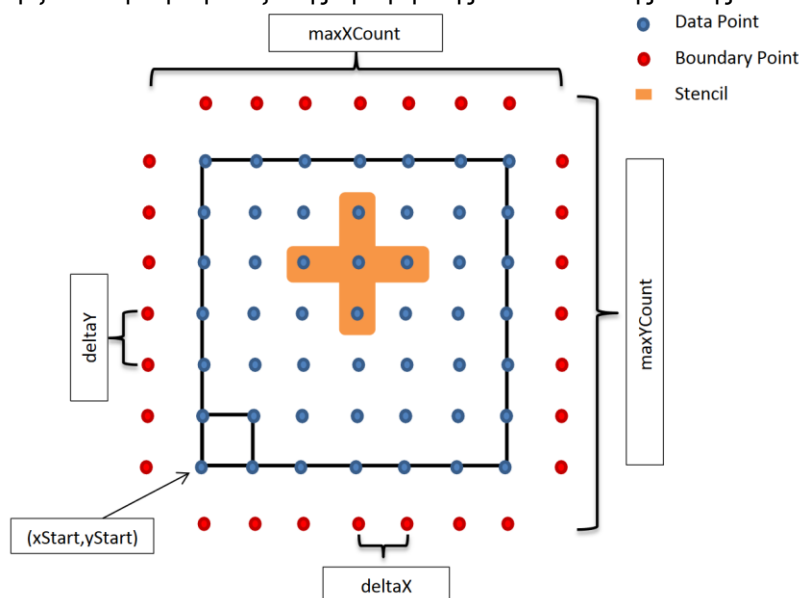
Ο αλγόριθμος δουλεύει ως εξής:

1. Έχουμε 2 πίνακες  $u$ ,  $u\_old$ . Ο πίνακας  $u\_old$  αρχικοποιείται σε μηδέν.
2. Η νέα προσέγγιση της λύσης υπολογίζεται στον πίνακα  $u$  για κάθε σημείο  $(x,y)$  χρησιμοποιώντας τις τιμές γειτονικών σημείων του  $u\_old$  (stencil 5 σημείων) και υλοποιείται στην συνάρτηση **one\_jacobi\_iteration** του κώδικα.
3. Οι πίνακες  $u$  και  $u\_old$  ανταλλάσσονται
4. Τα βήματα 2 και 3 επαναλαμβάνονται έως ότου η μέθοδος συγκλίνει (δηλαδή οι τιμές μετά από κάθε επανάληψη δεν αλλάζουν σημαντικά - error tolerance for the iterative solver -tol) ή μέχρι να φτάσουμε έναν προκαθορισμένο αριθμό επαναλήψεων (mits - maximum solver iterations). Στην περίπτωση μας διατηρείστε σταθερό τον αριθμό επαναλήψεων σε mits=50 και μικρό tol =  $1e-13$  ώστε να εκτελούνται πάντα οι 50 επαναλήψεις για σωστή μελέτη κλιμάκωσης.
5. Στο τέλος η συνάρτηση **checkSolution** υπολογίζει το σφάλμα μεταξύ της αριθμητικής και αναλυτικής λύσης.

Δεν θα χρειαστείτε ιδιαίτερες μαθηματικές γνώσεις ή άλλο υπόβαθρο. Τα σχόλια και ο κώδικας του ακολουθιακού προγράμματος αρκούν για την εργασία.

Η μαθηματική σημασία μεταβλητών που χρησιμοποιεί το πρόγραμμα απεικονίζονται στο διπλανό σχήμα.

Το αρχείο input περιέχει τιμές  $n$ ,  $m$  για το μέγεθος των  $u$ ,  $u\_old$  (οι μοναδικές τιμές που θα αλλάξετε),  $\alpha$  - Helmholtz constant (0.8),  $\text{relax}$  - successive over-relaxation parameter (1.0),  $\text{tol}$  - error tolerance for the iterative solver ( $1e-13$ ),  $\text{mits}$  - maximum solver iterations (50)



Ζητείται

1. Να βελτιώσετε και να προετοιμάσετε τον ακολουθιακό κώδικα για παραλληλοποίηση
2. Να σχεδιάσετε, αναπτύξετε και να αξιολογήσετε αποδοτικό Παράλληλο MPI κώδικα. Ειδικότερα
  - a. να μελετήσετε την κλιμάκωση (δεδομένα-μέγεθος πίνακα-, διεργασίες)
    - i. Χωρίς τον έλεγχο σύγκλισης (πρακτικά απλά δεν εκτελείτε την εντολή Allreduce)
    - ii. Με τον έλεγχο σύγκλισης σε κάθε βήμα της επανάληψης
  - b. Να συγκρίνετε με την αντίστοιχη κλιμάκωση του challenge.
3. Να εξελίξετε το MPI πρόγραμμα σε υβριδικό MPI+OpenMp, να μελετήσετε την κλιμάκωση του (δεδομένα, διεργασίες) και να τον συγκρίνετε με το καθαρό MPI που αναπτύξατε.
4. Να αναπτύξετε κώδικα CUDA και να μελετήσετε τους χρόνους σε μία και δυο GPU.
5. Να ακολουθήσετε τις συνοδευτικές οδηγίες Σχεδιασμού και Ανάπτυξης για βελτιστοποίηση κώδικα.
6. Να παρουσιάσετε, τον σχεδιασμό των προγραμμάτων, την μελέτη κλιμάκωσης, συγκρίσεις κλπ σε αναλυτική Έκθεση με Τυπική Δομή και Περιεχόμενο:

- I. **Εισαγωγή** (αντικείμενο εργασίας, δομή εργασίας, κλπ)
- II. **Ακολουθιακό Πρόγραμμα.** Βελτιστοποίηση, προετοιμασία για παραλληλοποίηση (δείτε συνοδευτικές οδηγίες). Σύγκριση χρόνων με το αρχικό. Μετρήστε με διαφορετικά μεγέθη πίνακα (αλλάζοντας **μόνο** τις δυο πρώτες τιμές αρχείου input). Ξεκινήστε με 840 X 840 και συνεχίστε διπλασιάζοντας την πλευρά (το 840 διαιρείται ακριβώς με 2,3,4,5,6,7,8,10 που θα βοηθήσει στον διαμερισμό των πινάκων σε διεργασίες στο MPI). Δεν αλλάζετε τις άλλες παραμέτρους σε καμία μέτρηση στην εργασία, ώστε να συγκρίνετε μόνο την επίπτωση λόγω του μεγέθους του πίνακα. Σταματάτε τον διπλασιασμό πλευράς (840, 1680, 3360, 6720, 13440, 26880) όταν δεν υπάρχει αρκετή μνήμη ή όταν εκτελούνται λιγότερες από 50 επαναλήψεις. Τα ίδια μεγέθη πίνακα θα χρησιμοποιήσετε σε όλες τις μετρήσεις των MPI προγραμμάτων. Οι αντίστοιχοι χρόνοι του αρχικού προγράμματος για τα 6 μεγέθη, ώστε να αποφύγετε χρονοβόρα τρεξίματα, μεταγλωττισμένο με -O3 είναι σε δεύτερα (0.837, 3.337, 13.336, 53.336, 213.313 (3.56 λεπτά), 853.351 (14+ λεπτά!!!)). Παρατηρήστε στο PBS script την επιλογή **mem** #PBS -l select=1:ncpus=8:mpiprocs=1:**mem=16400000kb** (ορίζει την συνολική χρήση μνήμης και όχι ανά διεργασία, ώστε αν έχετε λιγότερες από 8 διεργασίες να μοιράζεται όλη η διαθέσιμη μνήμη). Παράκληση, δοκιμάστε τις βελτιώσεις σας με μικρά μεγέθη πίνακα και για τα μεγάλα χρονομετρήστε **μόνο** μια φορά γιατί είναι πολύ χρονοβόρα (το 26880 X 26880 είναι 1024 φορές πιο αργό από το 840 X 840). Σημειώνουμε ότι το Challenge πρόγραμμα δεν μπορεί να επιλύσει μεγαλύτερο πρόβλημα από 26880 X 26880. Το δικό σας, με καλό σχεδιασμό, ίσως να μπορεί να το λύσει, χωρίς μάλλον να εκτελεί 50 επαναλήψεις. Να το δοκιμάσετε για μεγαλύτερα προβλήματα ΜΟΝΟ για 80 διεργασίες.
- III. **Σχεδιασμός MPI Παραλληλοποίησης.** Διαμοιρασμός Δεδομένων πινάκων u και u\_old (Block-Block) σε καρτεσιανή τοπολογία NXN διεργασιών και οργάνωση βασικής δομής προγράμματος βάσει της μεθοδολογία του Foster (άλως, stencil, επικοινωνία, τοπολογία διεργασιών, κλπ.) (δείτε συνοδευτικές οδηγίες). Η διεργασία 0 διαβάζει το input και με Bcast ενημερώνει τις υπόλοιπες.
- IV. **Βελτιστοποίηση MPI κώδικα.** Μείωση αδρανούς χρόνου ή περιττών υπολογισμών (επιλογές επικοινωνίας, επικάλυψη επικοινωνίας με υπολογισμούς, αποφυγή πολλαπλών αντιγραφών με χρήση datatypes, κλπ). Να αναφέρετε **ρητά** ποιες βελτιώσεις έχετε εφαρμόσει. Επίσης αν έχετε κάνει δικές βελτιώσεις που δεν έχω αναφέρει. Χρησιμοποιείστε το mpiP για να δείτε το λόγο τυχόν καθυστερήσεων (δείτε συνοδευτικές οδηγίες).
- V. **Μετρήσεις MPI κώδικα.** Δύο μελέτες κλιμάκωσης (χρόνοι, επιτάχυνση, αποδοτικότητα) α) με απενεργοποιημένη την εντολή Allreduce και β) μια πλήρης. Για τα ανωτέρω μεγέθη πίνακα να τρέξετε για 4, 9, 16, 25, 36, 49, 64 και 80 (8X10) διεργασίες (NXN) και παρουσίαση αποτελεσμάτων. Σταθερός αριθμός επαναλήψεων (50) και των άλλων δεδομένων εισόδου για **όλες** τις μετρήσεις, ώστε να παίρνετε συγκρίσιμους και μετρήσιμους χρόνους. Σχολιάζετε την συμπεριφορά με αναφορά σε mpiP αρχεία. Σύγκριση με το Challenge πρόγραμμα που υλοποιεί διαμοιρασμό κατά σειρές, επομένως το δικό σας θα πρέπει να πλεονεκτεί λόγω καλύτερου διαμοιρασμού (βάσει του Foster) και των άλλων βελτιώσεων (δείτε συνοδευτικές οδηγίες).
- VI. **Υβριδικό MPI+OpenMp.** Σχεδιασμός παραλληλοποίησης τμημάτων (for) του MPI με openmp. Μετρήσεις και σύγκριση με το καθαρό MPI. Για τα ανωτέρω μεγέθη πίνακα να τρέξετε για 4, 16, 36, 64 και 80 νήματα (π.χ. 2 διεργασίες και 4 νήματα, 8δ-16ν, 18δ-36ν, 32δ-64ν, 40δ-80) και παρουσίαση αποτελεσμάτων (δείτε συνοδευτικές οδηγίες).
- VII. **Πρόγραμμα CUDA.** Σχεδιασμός, αξιολόγηση καθαρού προγράμματος cuda σε 1 και 2 GPU. Δείτε CUDA οδηγό.
- VIII. **Συμπεράσματα**

- ❖ Μην κάνετε “μετρήσεις για τις μετρήσεις”. Αν για το κάποιο μέγεθος προβλήματος μετά από κάποιον αριθμό πυρήνων δεν έχετε επιτάχυνση >1 να μην κάνετε τρεξίματα με περισσότερους πυρήνες. Είναι περιττές και σπαταλούν πόρους.
- ❖ Να τρέχετε πάντα με ενεργοποιημένο το mpiP, να βάλετε όμως εντολές για ενεργοποίηση του μόνο στο κεντρικό for, εκεί δηλαδή που έχετε βάσει WTime και όχι για όλο το πρόγραμμα. Οδηγίες θα βρείτε στο κεφ. "Controlling the Scope of mpiP Profiling in your Application" των σημειώσεων mpiP (όπως στην εργαστηριακή άσκηση). Αν δεν παίρνετε καλά αποτελέσματα να εξετάσετε το output του mpiP. Θα σας βοηθήσει να δείτε τι δεν πάει καλά. Δείτε π.χ. που ξοδεύει χρόνο το πρόγραμμά σας (waits, sends, recvs). Μερικά προβλήματα είναι απλά, π.χ. τα waits σε λάθος θέση.

#### Φοιτητές ΠΜΣ - ΔΠΜΣ

Οι μεταπτυχιακοί ΠΜΣ Πληροφορικής και ΔΠΜΣ ΗΑ θα πρέπει να μελετήσουν, αναπτύξουν, κλπ κάτι επιπλέον, όπως:

1. Parallel I/O. έξοδος του Τελικού πίνακα. Ξεχωριστές μετρήσεις για το I/O με Wtime πριν και μετά το I/O.
2. Μεθοδολογία Foster (μετρήσεις latency, bandwidth συστήματος με omb benchmarks).
3. Εφαρμογή Μεθοδολογίας POP με το MPIP.
4. Πειράματα με διαφορετικά υβριδικά προγράμματα (π.χ. με νήματα που επικαλύπτουν με υπολογισμούς την κλήση SendRecv από άλλο νήμα).
5. Υβριδικό πρόγραμμα MPI+CUDA ή MPI+OpenMp+CUDA
6. Σύγκριση OpenMp vs. MPI Shared memory access σε έναν κόμβο.
7. Πειράματα με Thread affinity, caching
8. Κατόπιν συνεννόησης, άλλο θέμα.

## Ομαδική Εργασία

Επιτρέπεται και μάλιστα συνιστάται η συνεργασία 2 ατόμων. Ομάδες των 3ων επιτρέπεται κατ' εξαίρεση, όμως στην περίπτωση αυτή η εργασία θα αξιολογηθεί με πιο αυστηρά κριτήρια και θα πρέπει να περιλαμβάνει κάποια επιπλέον δουλειά από την ανωτέρω λίστα των μεταπτυχιακών.

## Παραδοτέα

1. Κατάθεση της έκθεσης/κειμένου σε pdf στο eclass με όνομα αρχείου τους ΑΜ των μελών της ομάδας σας.
2. Δημιουργείτε στον λογαριασμό σας στην argo τον κατάλογο `ProjectSubmission` με 4 υποκαταλόγους
  - a. `Sequential` Περιέχει το βελτιωμένο ακολουθιακό σας πρόγραμμα, Makefile (ή txt αρχείο με την εντολή μεταγλώττισης) το `input` και το `PBSseqJ.sh`.
  - b. `ParallelMPI` Περιέχει το παραλληλοποιημένο MPI πρόγραμμα, Makefile (ή txt αρχείο με την εντολή μεταγλώττισης), το `input`, το `PBSmpiJ.sh` καθώς και `mpiP` αρχεία που έχετε επιλέξει και έχετε σχολιάσει στο κείμενό σας.
  - c. `HybridMPI` Περιέχει το παραλληλοποιημένο υβριδικό MPI+OpenMp πρόγραμμα, Makefile (ή txt αρχείο με την εντολή μεταγλώττισης), το `input` και το `PBSHybridJ.sh`
  - d. `CUDA` Περιέχει το παραλληλοποιημένο CUDA πρόγραμμα, Makefile (ή txt αρχείο με την εντολή μεταγλώττισης), το `input` και το `PBScudaJ.sh`

Στην κατάθεση σας στο eclass να αναφέρετε τον λογαριασμό argo που βρίσκονται οι ανωτέρω κατάλογοι και αρχεία. Δίνετε δικαιώματα Read-Write-Execute των ανωτέρω καταλόγων στον λογαριασμό cotronis (<https://unix.stackexchange.com/questions/507571/how-to-allow-folder-permission-for-another-user-in-linux>)

## Προθεσμίες Κατάθεσης

Παράδοση (αυστηρές προθεσμίες-χωρίς δυνατότητα παράτασης):

Για εξεταστική Ιανουαρίου Κυριακή 21/02/2021

Για εξεταστική Σεπτεμβρίου Κυριακή 10/10/2021

## Αξιολόγηση

Η εργασία συνολικά συνεισφέρει στον τελικό βαθμό 50%, τα γραπτά 30% και η εργαστηριακή άσκηση 20%. Στην **αξιολόγηση** της εργασίας **εκτιμώνται η απόδοση με τις βελτιστοποιήσεις κώδικα και η παρουσίαση έκθεσης**, όπως προσδιορίζονται από τα ζητούμενα 1-7 πιο πάνω. Αξιολογούνται: Σχεδιασμός, υλοποίηση, μελέτη κλιμάκωσης, συγκρίσεις και παρουσίαση. Επιμέρους κατανομή βαθμολογίας στα 100 (με το bonus **105!**)

Ακολουθιακό	10
MPI	50
MPI+openMp	10
Cuda	20
Συνολική εικόνα (Παρουσίαση δομή, πληρότητα, κλπ)	10
<i>Bonus για εξαιρετικές περιπτώσεις</i>	<i>05</i>