

## Εργασία 3 - Προγραμματισμός με MPI

Ονοματεπώνυμο: Μάριος Γιαννόπουλος

A.M.: 1115200000032

### Γενικές Πληροφορίες

#### Υπολογιστικό Σύστημα

Όλο το έργο υλοποιήθηκε στο ίδιο υπολογιστικό περιβάλλον:

- Όνομα Υπολογιστικού Συστήματος: Linux12
- Επεξεργαστής: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz
- Αριθμός Πυρήνων: 4
- Λειτουργικό Σύστημα: Linux Ubuntu 20.04.2 LTS
- Έκδοση Μεταγλωττιστή: gcc (Ubuntu 9.4.0-1ubuntu1 20.04.2) 9.4.0

#### Οδηγίες Εκτέλεσης Python Scripts

Για την εκτέλεση των Python scripts που επεξεργάζονται τα αποτελέσματα, ακολουθήστε τα εξής βήματα:

1. Μεταβείτε στον φάκελο `scripts`.
2. Εγκαταστήστε τις απαραίτητες βιβλιοθήκες:

```
pip install -r requirements.txt
```

3. Εκτελέστε το script που σας ενδιαφέρει:

```
python <test_script>.py
```

**Σημείωση:** Όλα τα αποτελέσματα στα γραφήματα είναι από την εκτέλεση των πειραμάτων στο εργαστήριο Linux. Κάθε πείραμα εκτελέστηκε 5 φορές και τα αποτελέσματα αναφέρονται στο μέσο όρο των επαναλήψεων.

### Άσκηση 3.1

#### Εισαγωγή

Το Παιχνίδι της Ζωής (Game of Life) είναι ένα κυτταρικό αυτόματο που αναπτύχθηκε από τον μαθηματικό John Horton Conway. Είναι ένα μοντέλο που προσομοιώνει την εξέλιξη ενός πληθυσμού κυττάρων σε ένα δισδιάστατο πλέγμα, όπου κάθε κύτταρο μπορεί να βρίσκεται σε μία από δύο καταστάσεις: ζωντανό ή

νεκρό. Οι κανόνες που διέπουν την εξέλιξη του πληθυσμού βασίζονται στον αριθμό των γειτόνων κάθε κυττάρου. Σε αυτή την άσκηση, υλοποιήθηκε μια παράλληλη έκδοση του Παιχνιδιού της Ζωής χρησιμοποιώντας την MPI (Message Passing Interface) για την κατανομή του υπολογιστικού φόρτου μεταξύ πολλαπλών διεργασιών. Η υλοποίηση βασίζεται στη σειριακή έκδοση του αλγορίθμου που αναπτύχθηκε στην Άσκηση 2.1, χωρίς τη χρήση OpenMP.

## Συγχρονισμός

Ο συγχρονισμός μεταξύ των διεργασιών είναι ένα κρίσιμο στοιχείο στην παράλληλη υλοποίηση του Παιχνιδιού της Ζωής. Στον κώδικα, ο συγχρονισμός επιτυγχάνεται με τις εξής λειτουργίες:

1. **MPI\_Scatter:** Η διεργασία 0 κατανέμει τα δεδομένα του πλέγματος στις υπόλοιπες διεργασίες.
2. **MPI\_Sendrecv:** Κάθε διεργασία ανταλλάσσει γραμμές ορίων (ghost rows) με τις γειτονικές της διεργασίες για να υπολογίσει σωστά την επόμενη γενιά.
3. **MPI\_Gather:** Η διεργασία 0 συγκεντρώνει τα αποτελέσματα από όλες τις διεργασίες για να ενημερώσει το τελικό πλέγμα.
4. **MPI\_Barrier:** Εξασφαλίζει ότι όλες οι διεργασίες έχουν ολοκληρώσει την τρέχουσα γενιά πριν προχωρήσουν στην επόμενη.

Αυτές οι λειτουργίες εξασφαλίζουν ότι οι διεργασίες συνεργάζονται αποτελεσματικά και ότι τα δεδομένα παραμένουν συνεπή κατά τη διάρκεια της εκτέλεσης.

## Πειραματική Διαδικασία

### • Παραμετροποίηση:

- Μέγεθος πλέγματος:  $64 \times 64$ ,  $1024 \times 1024$ ,  $4096 \times 4096$ .
- Αριθμός διεργασιών: 2, 4, 8, 16.
- Αριθμός γενιών: 1000.

### • Εκτέλεση:

- Τα πειράματα εκτελέστηκαν 5 φορές για κάθε συνδυασμό παραμέτρων.
- Καταγράφηκε ο χρόνος εκτέλεσης για κάθε πείραμα.
- Τα δεδομένα αποθηκεύτηκαν σε CSV αρχείο.

### • Αυτοματοποίηση:

- Αναπτύχθηκαν Python scripts για την εκτέλεση των πειραμάτων και την καταγραφή των δεδομένων.
- Χρησιμοποιήθηκαν Python scripts για την επεξεργασία των αποτελεσμάτων και τη δημιουργία γραφημάτων.

## Αποτελέσματα

Από τα αποτελέσματα των μετρήσεων, παρατηρούμε τα εξής:

1. Για μικρά μεγέθη πλέγματος ( $64 \times 64$ ):

- Η χρήση περισσότερων διεργασιών δεν οδηγεί πάντα σε μείωση του χρόνου εκτέλεσης. Για παράδειγμα, η χρήση 4 διεργασιών οδήγησε σε μεγαλύτερο χρόνο εκτέλεσης (2.20885 s) σε σύγκριση με τη χρήση 2 διεργασιών (1.72495 s). Αυτό οφείλεται στο γεγονός ότι το κόστος επικοινωνίας μεταξύ των διεργασιών είναι υψηλό σε σχέση με τον χρόνο υπολογισμού για τόσο μικρά πλέγματα.
- Η χρήση 8 και 16 διεργασιών οδήγησε σε ακόμη μεγαλύτερους χρόνους εκτέλεσης (7.92210 s και 10.79159 s αντίστοιχα), γεγονός που επιβεβαιώνει ότι για μικρά πλέγματα η παράλληλη εκτέλεση δεν είναι αποτελεσματική.

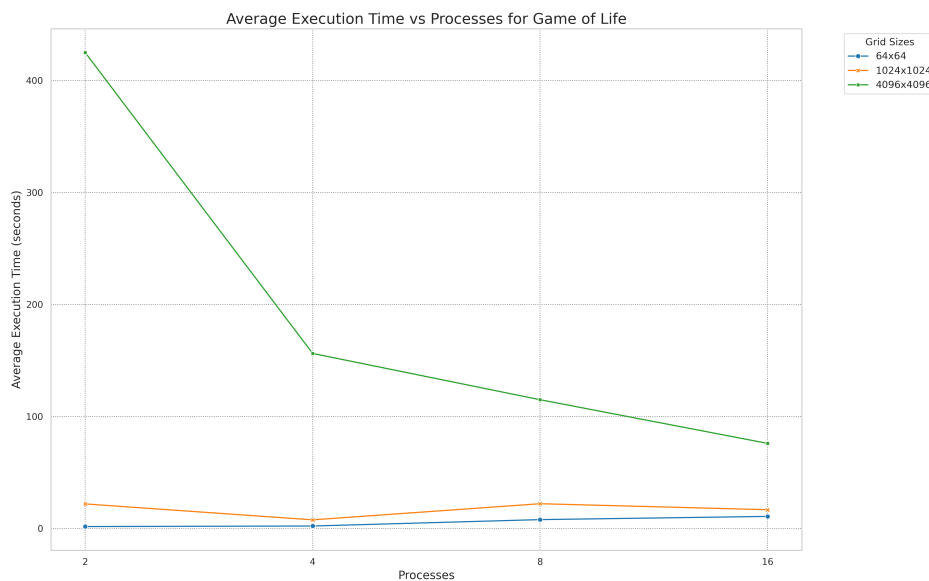
## 2. Για μεσαία μεγέθη πλέγματος (1024x1024):

- Η χρήση 4 διεργασιών οδήγησε σε σημαντική μείωση του χρόνου εκτέλεσης (7.71714 s) σε σύγκριση με τη χρήση 2 διεργασιών (21.93556 s). Αυτό δείχνει ότι για μεσαία μεγέθη πλέγματος, η παράλληλη εκτέλεση αρχίζει να γίνεται αποτελεσματική.
- Ωστόσο, η χρήση 8 και 16 διεργασιών οδήγησε σε αύξηση του χρόνου εκτέλεσης (22.11091 s και 16.76202 s αντίστοιχα), γεγονός που υποδηλώνει ότι το κόστος επικοινωνίας αρχίζει να γίνεται σημαντικό για μεγαλύτερο αριθμό διεργασιών.

## 3. Για μεγάλα μεγέθη πλέγματος (4096x4096):

- Η χρήση περισσότερων διεργασιών οδήγησε σε σημαντική μείωση του χρόνου εκτέλεσης. Για παράδειγμα, η χρήση 16 διεργασιών οδήγησε σε μέσο χρόνο εκτέλεσης 75.97231 s, σε σύγκριση με 425.03920 s για 2 διεργασίες.
- Αυτό δείχνει ότι για πολύ μεγάλα πλέγματα, η παράλληλη εκτέλεση είναι πολύ αποτελεσματική, καθώς ο χρόνος υπολογισμού κυριαρχεί έναντι του χρόνου επικοινωνίας.

## Γραφήματα



Γράφημα 1: Χρόνοι εκτέλεσης για το Παιχνίδι της Ζωής με MPI

## Συμπεράσματα

Από τα αποτελέσματα και τα γραφήματα, μπορούμε να εξάγουμε τα εξής συμπεράσματα:

### 1. Επίδραση του μεγέθους πλέγματος:

- Για μικρά μεγέθη πλέγματος, η παράλληλη εκτέλεση δεν είναι αποτελεσματική λόγω του υψηλού κόστους επικοινωνίας σε σχέση με τον χρόνο υπολογισμού.
- Για μεσαία και μεγάλα μεγέθη πλέγματος, η παράλληλη εκτέλεση γίνεται ολοένα και πιο αποτελεσματική, καθώς ο χρόνος υπολογισμού κυριαρχεί έναντι του χρόνου επικοινωνίας.

### 2. Επίδραση του αριθμού των διεργασιών:

- Για μικρά πλέγματα, η αύξηση του αριθμού των διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης λόγω του κόστους επικοινωνίας.
- Για μεγάλα πλέγματα, η αύξηση του αριθμού των διεργασιών οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης, καθώς ο χρόνος υπολογισμού είναι πολύ μεγαλύτερος από το κόστος επικοινωνίας.

### 3. Βέλτιστος αριθμός διεργασιών:

- Για κάθε μέγεθος πλέγματος, υπάρχει ένας βέλτιστος αριθμός διεργασιών που εξαρτάται από το μέγεθος του πλέγματος και το κόστος επικοινωνίας. Για παράδειγμα, για πλέγμα  $1024 \times 1024$ , ο βέλτιστος αριθμός διεργασιών φαίνεται να είναι 4, ενώ για πλέγμα  $4096 \times 4096$ , ο βέλτιστος αριθμός διεργασιών φαίνεται να είναι 16.

Συνολικά, η παράλληλη υλοποίηση του Παιχνιδιού της Ζωής με MPI αποδεικνύεται πολύ αποτελεσματική για μεγάλα προβλήματα, ενώ για μικρά προβλήματα η σειριακή εκτέλεση μπορεί να είναι επαρκής. Η επιλογή του βέλτιστου αριθμού διεργασιών εξαρτάται από το μέγεθος του πλέγματος και το κόστος επικοινωνίας μεταξύ των διεργασιών.

## Άσκηση 3.2

### Εισαγωγή

Ο πολλαπλασιασμός πίνακα-διανύσματος είναι μια θεμελιώδης πράξη στη γραμμική άλγεβρα, η οποία βρίσκει εφαρμογές σε πολλούς τομείς όπως η επιστήμη των υπολογιστών, η φυσική και η μηχανική. Σε αυτή την άσκηση, υλοποιήθηκε ένα παράλληλο πρόγραμμα χρησιμοποιώντας την MPI (Message Passing Interface) για τον υπολογισμό του πολλαπλασιασμού πίνακα-διανύσματος με κατανομή του πίνακα κατά μπλοκ-στήλες. Η υλοποίηση βασίζεται στη σειριακή έκδοση του αλγορίθμου, όπου η διεργασία 0 δημιουργεί τον πίνακα και το διάνυσμα και κατανέμει τα δεδομένα στις υπόλοιπες διεργασίες για παράλληλο υπολογισμό.

### Συγχρονισμός

Ο συγχρονισμός μεταξύ των διεργασιών είναι ένα κρίσιμο στοιχείο στην παράλληλη υλοποίηση του πολλαπλασιασμού πίνακα-διανύσματος. Στον κώδικα, ο συγχρονισμός επιτυγχάνεται με τις εξής λειτουργίες:

1. `MPI_Scatter`: Η διεργασία 0 κατανέμει τις στήλες του πίνακα στις υπόλοιπες διεργασίες.
2. `MPI_Bcast`: Η διεργασία 0 μεταδίδει το διάνυσμα σε όλες τις διεργασίες.
3. `MPI_Reduce`: Οι διεργασίες στέλνουν τα τοπικά αποτελέσματα στη διεργασία 0, όπου συγκεντρώνονται για τον τελικό υπολογισμό.

Αυτές οι λειτουργίες εξασφαλίζουν ότι οι διεργασίες συνεργάζονται αποτελεσματικά και ότι τα δεδομένα παραμένουν συνεπή κατά τη διάρκεια της εκτέλεσης.

## Πειραματική Διαδικασία

- **Παραμετροποίηση:**

- Μέγεθος πίνακα:  $100 \times 100$ ,  $1000 \times 1000$ ,  $5000 \times 5000$ ,  $10000 \times 10000$ .
- Αριθμός διεργασιών: 2, 4, 5, 10.

- **Εκτέλεση:**

- Τα πειράματα εκτελέστηκαν 5 φορές για κάθε συνδυασμό παραμέτρων.
- Καταγράφηκε ο χρόνος εκτέλεσης για κάθε πείραμα.
- Τα δεδομένα αποθηκεύτηκαν σε CSV αρχείο.

- **Αυτοματοποίηση:**

- Αναπτύχθηκαν Python scripts για την εκτέλεση των πειραμάτων και την καταγραφή των δεδομένων.
- Χρησιμοποιήθηκαν Python scripts για την επεξεργασία των αποτελεσμάτων και τη δημιουργία γραφημάτων.

## Αποτελέσματα

Σε αυτή την ενότητα, παρουσιάζονται τα αποτελέσματα των χρόνων εκτέλεσης για τον πολλαπλασιασμό πίνακα-διανύσματος με βάση τα δεδομένα που παρέχονται. Οι χρόνοι εκτέλεσης καταγράφηκαν για διαφορετικούς αριθμούς διεργασιών (2, 4, 5, 10) και διαφορετικά μεγέθη πινάκων ( $100 \times 100$ ,  $1000 \times 1000$ ,  $5000 \times 5000$ ,  $10000 \times 10000$ ). Τα αποτελέσματα παρουσιάζονται σε μορφή πίνακα και αναλύονται παρακάτω.

### Πίνακας Χρόνων Εκτέλεσης

Διεργασίες	Μέγεθος Πίνακα	Χρόνος (s)	Παρατηρήσεις
2	100x100	1.43	-
2	1000x1000	3.02	-
2	5000x5000	38.35	-
2	10000x10000	150.54	-
4	100x100	1.49	Χαμηλή βελτίωση σε σχέση με 2 διεργασίες
4	1000x1000	3.79	Αύξηση του χρόνου λόγω επικοινωνίας
4	5000x5000	59.94	Αύξηση του χρόνου σε σχέση με 2 διεργασίες
4	10000x10000	230.98	Αύξηση του χρόνου σε σχέση με 2 διεργασίες
5	100x100	1.55	Χαμηλή βελτίωση σε σχέση με 2 διεργασίες
5	1000x1000	4.01	Αύξηση του χρόνου λόγω επικοινωνίας
5	5000x5000	58.86	Μικρή βελτίωση σε σχέση με 4 διεργασίες
5	10000x10000	236.74	Μικρή βελτίωση σε σχέση με 4 διεργασίες
10	100x100	1.58	Χαμηλή βελτίωση σε σχέση με 2 διεργασίες
10	1000x1000	3.94	Αύξηση του χρόνου λόγω επικοινωνίας
10	5000x5000	60.55	Αύξηση του χρόνου σε σχέση με 2 διεργασίες
10	10000x10000	234.36	Μικρή βελτίωση σε σχέση με 4 διεργασίες

Table 1: Σύγκριση χρόνων εκτέλεσης για διαφορετικούς αριθμούς διεργασιών και μεγέθη πινάκων

## Ανάλυση Αποτελεσμάτων

### 1. Για μικρούς πίνακες (100x100):

- Η χρήση περισσότερων διεργασιών δεν οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης. Για παράδειγμα, η χρήση 10 διεργασιών οδήγησε σε μέσο χρόνο εκτέλεσης 1.582433 s, σε σύγκριση με 1.429126 s για 2 διεργασίες. Αυτό οφείλεται στο γεγονός ότι το κόστος επικοινωνίας μεταξύ των διεργασιών είναι υψηλό σε σχέση με τον χρόνο υπολογισμού για τόσο μικρούς πίνακες.

### 2. Για μεσαίους πίνακες (1000x1000):

- Η χρήση περισσότερων διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης. Για παράδειγμα, η χρήση 10 διεργασιών οδήγησε σε μέσο χρόνο εκτέλεσης 3.938691 s, σε σύγκριση με 3.018212 s για 2 διεργασίες. Αυτό υποδηλώνει ότι το κόστος επικοινωνίας αρχίζει να γίνεται σημαντικό για μεγαλύτερο αριθμό διεργασιών.

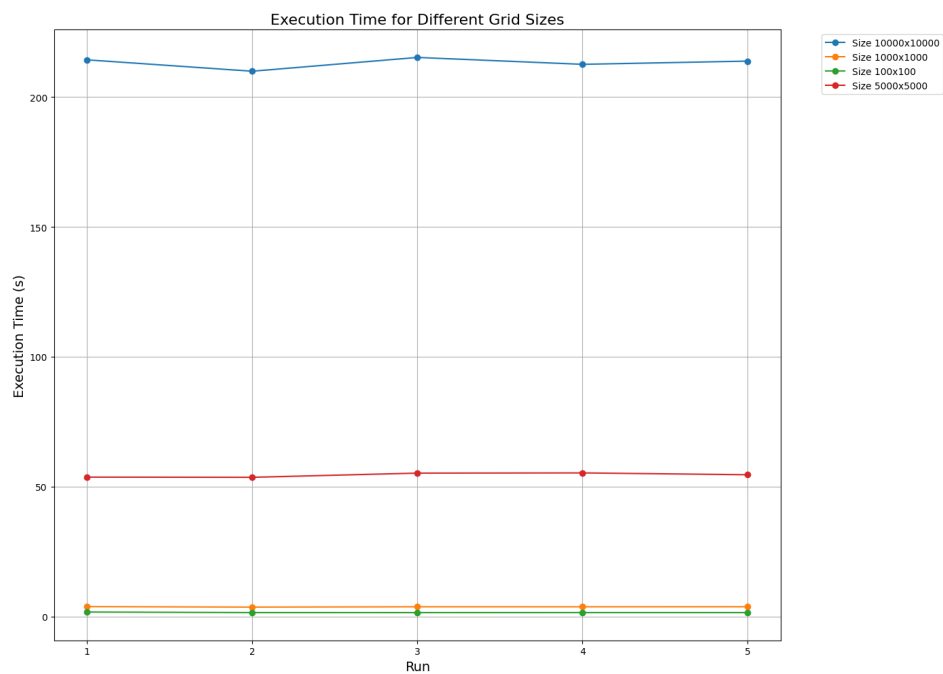
### 3. Για μεγάλους πίνακες (5000x5000 και 10000x10000):

- Η χρήση περισσότερων διεργασιών δεν οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης. Για παράδειγμα, για πίνακα 5000x5000, η χρήση 10 διεργασιών οδήγησε σε μέσο χρόνο εκτέλεσης 60.547491 s, σε σύγκριση με 38.354262 s για 2 διεργασίες. Αυτό δείχνει ότι για πολύ μεγάλους πίνακες, το κόστος επικοινωνίας είναι υψηλό και η παράλληλη εκτέλεση δεν είναι πάντα αποτελεσματική.

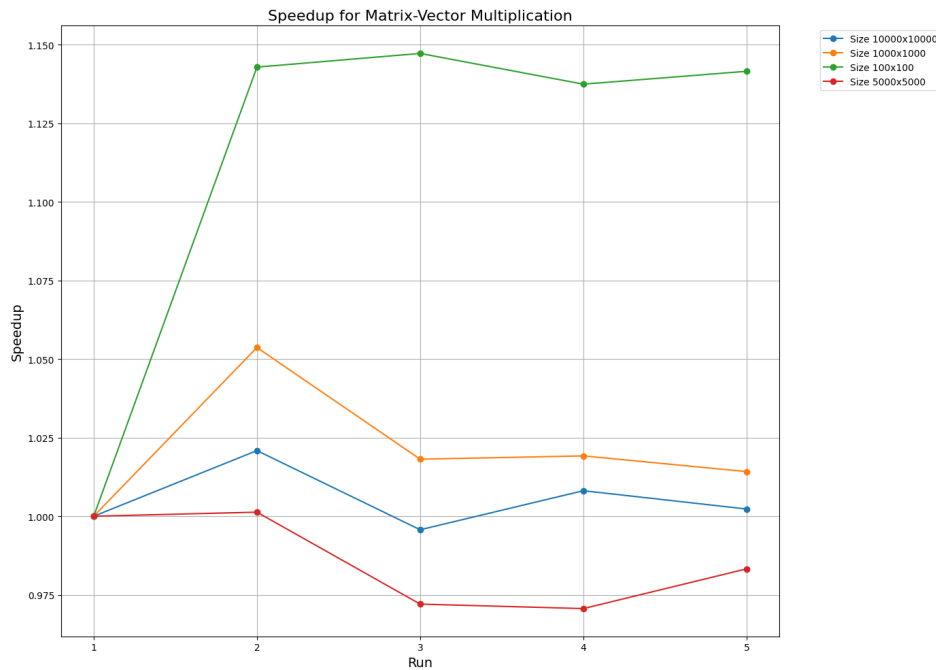
### 4. Βέλτιστος αριθμός διεργασιών:

- Για μικρούς πίνακες, ο βέλτιστος αριθμός διεργασιών φαίνεται να είναι 2, καθώς η χρήση περισσότερων διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης λόγω του κόστους επικοινωνίας.
- Για μεσαίους πίνακες, ο βέλτιστος αριθμός διεργασιών φαίνεται να είναι επίσης 2, καθώς η χρήση περισσότερων διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης.
- Για μεγάλους πίνακες, ο βέλτιστος αριθμός διεργασιών εξαρτάται από το κόστος επικοινωνίας και τον χρόνο υπολογισμού. Ωστόσο, από τα δεδομένα, φαίνεται ότι η χρήση 2 διεργασιών είναι η πιο αποτελεσματική.

## Γραφήματα



Γράφημα 2: Χρόνοι εκτέλεσης για τον Πολλαπλασιασμό Πίνακα-Διανύσματος με MPI



Γράφημα 3: Επιτάχυνση για τον Πολλαπλασιασμό Πίνακα-Διανύσματος με MPI

## Συμπεράσματα

Από τα αποτελέσματα και τα γραφήματα, μπορούμε να εξάγουμε τα εξής συμπεράσματα:

### 1. Για μικρούς πίνακες (100x100):

- Η χρήση περισσότερων διεργασιών δεν οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης λόγω του υψηλού κόστους επικοινωνίας σε σχέση με τον χρόνο υπολογισμού.

### 2. Για μεσαίους πίνακες (1000x1000):

- Η χρήση περισσότερων διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης λόγω του κόστους επικοινωνίας.

### 3. Για μεγάλους πίνακες (5000x5000 και 10000x10000):

- Η χρήση περισσότερων διεργασιών δεν οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης, καθώς το κόστος επικοινωνίας είναι υψηλό.

### 4. Βέλτιστος αριθμός διεργασιών:

- Για όλα τα μεγέθη πινάκων, η χρήση 2 διεργασιών φαίνεται να είναι η πιο αποτελεσματική, καθώς η αύξηση του αριθμού των διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης λόγω του κόστους επικοινωνίας.

Συνολικά, η παράλληλη υλοποίηση του πολλαπλασιασμού πίνακα-διανύσματος με MPI αποδεικνύεται πιο αποτελεσματική για μικρούς και μεσαίους πίνακες όταν χρησιμοποιούνται 2 διεργασίες. Για μεγάλους πίνακες, η αύξηση του αριθμού των διεργασιών δεν οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης λόγω του υψηλού κόστους επικοινωνίας.



### Άσκηση 3.3

Εισαγωγή

Πειραματική Διαδικασία

Αποτελέσματα

Γραφήματα

Συμπεράσματα

### Άσκηση 3.4

Εισαγωγή

Πειραματική Διαδικασία

Αποτελέσματα

Γραφήματα

Συμπεράσματα