

# Λογισμικό και Προγραμματισμός Συστημάτων Υψηλής Επίδοσης

## 1<sup>ο</sup> Σετ Ασκήσεων

Χειμερινό Εξάμηνο 2023-2024

### Στοιχεία Ομάδας

Παπανικολάου Αικατερίνη	1064041	st1064041@ceid.upatras.gr
Πρέτσιος Κωνσταντίνος	1084666	st1084666@ceid.upatras.gr

## Πληροφορίες

Τα πειράματά μας έγιναν σε υπολογιστή με τα παρακάτω χαρακτηριστικά:

OS: Windows 11 Home

RAM: 8GB

Processor: 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 2.90 GHz

---

---

### Question 1: Quadratic Form

Ο κώδικας για τη συγκεκριμένη άσκηση βρίσκεται στο αρχείο `qf_hybrid.c`.

Στην υλοποίηση αυτή μοιράσαμε τον αριθμό των επαναλήψεων ανάλογα με το πλήθος των διεργασιών κάθε φορά και όσες επαναλήψεις περισσεύουν τις παίρνει η τελευταία διεργασία. Έπειτα με τα `local_start` και `local_end` ορίσαμε σε ποιο εύρος θα γράψει η κάθε διεργασία και έπειτα βάλαμε τις αρχικοποιήσεις που πρέπει να γίνουν στους πίνακες σε sections. Με το `barrier` βάζουμε τις διεργασίες να συγχρονιστούν όταν τελειώσουν τους υπολογισμούς. Έπειτα καθορίζουμε με τα `sendcounts` και `displs` αντίστοιχα τον αριθμό στοιχείων που θα στείλει η κάθε διεργασία με το `Gatherv` και πού αυτά θα αποθηκευτούν στον πίνακα. Όλα τα δεδομένα από τις διεργασίες τα συγκεντρώνει αυτή που έχει `rank = 0` για να κάνει το reduction με OpenMP για το τελικό result.

Οι μετρήσεις που πήραμε σε (s) για διάφορα πλήθη διεργασιών και threads είναι:

`qf_seq.c` : 12.915s

P\T	1	2	3	4
1	7.749	3.212	1.725	1.684
2	8.641	6.748	4.829	2.490
4	10.299	7.775	11.517	5.838

Να σημειώσουμε σε αυτό το σημείο ότι με την υλοποίηση αυτή για 3 διεργασίες πήραμε πίσω -nan ως result ενώ για 5 διεργασίες και πάνω segmentation fault. Ακόμα επειδή χρησιμοποιήσαμε το περιβάλλον WSL δεν ήταν διαθέσιμη η εντολή `perf` για να πάρουμε τιμές για τα FLOPs.

## Question 2: 2D Diffusion

b. Οι χρόνοι που πήραμε τρέχοντας τον κώδικα με non-blocking επικοινωνία υποδεικνύουν ότι δεν επιτυγχάνεται τέλειο overlap επικοινωνίας και υπολογισμών. Ναι μεν οι χρόνοι μειώνονται όταν αυξάνεται το πλήθος των επεξεργαστών αλλά η μείωση αυτή δεν είναι ανάλογη του πλήθους των επεξεργαστών που βάζουμε κάθε φορά. Τα αποτελέσματα των μετρήσεων φαίνονται παρακάτω.

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7o Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc
_ex01_code/mpiio$ time mpirun -n 1 ./diffusion2d_mpi_nb 1 1 2048 1000 1e-9
timestep from stability condition is 5.966290e-08
Timing: 2048 38.762307

real    0m39.054s
user    0m38.800s
sys     0m0.050s
```

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7o Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc
_ex01_code/mpiio$ time mpirun -n 2 ./diffusion2d_mpi_nb 1 1 2048 1000 1e-9
timestep from stability condition is 5.966290e-08
Timing: 2048 23.774442

real    0m24.069s
user    0m47.599s
sys     0m0.107s
```

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7o Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc
_ex01_code/mpiio$ time mpirun -n 3 ./diffusion2d_mpi_nb 1 1 2048 1000 1e-9
timestep from stability condition is 5.966290e-08
Timing: 2048 18.805448

real    0m19.098s
user    0m56.531s
sys     0m0.068s
```

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7o Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc
_ex01_code/mpiio$ time mpirun -n 4 ./diffusion2d_mpi_nb 1 1 2048 1000 1e-9
timestep from stability condition is 5.966290e-08
Timing: 2048 16.317951

real    0m16.612s
user    1m5.422s
sys     0m0.120s
```

c. Ο κώδικας για το συγκεκριμένο ερώτημα βρίσκεται στο αρχείο diffusion2d\_mpi\_nb\_io.c στη συνάρτηση write\_density\_mpi.

Αρχικά τα δεδομένα που πρόκειται να γραφτούν στο density\_mpi.bin τα κρατάμε στον πίνακα all\_data χωρίς τα ghost cells. Στη συνέχεια ορίζουμε το offset και το len που ουσιαστικά λένε σε κάθε διεργασία πού και πόσο θα γράψει μέσα στο αρχείο. Έπειτα με τις κατάλληλες εντολές MPI βάλαμε τις διεργασίες να γράψουν τα κομμάτια τους στο binary αρχείο.

Ακόμα υλοποιήσαμε το αρχείο sum\_of\_bin.c στο οποίο δίνουμε ως ορίσματα binary αρχεία και αυτό στη συνέχεια υπολογίζει για το καθένα το άθροισμα των στοιχείων που αυτό περιέχει.

Τρέχοντας ακολουθιακά το diffusion2d\_mpi\_nb\_io.c για να δημιουργηθεί το density\_seq.bin :

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7o Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc
_ex01_code/mpiio$ time mpirun -n 1 ./diffusion2d_mpi_nb_io 1 1 128 50000 0.00001
timestep from stability condition is 1.550003e-05
Timing: 128 6.665717

real    0m7.271s
user    0m6.765s
sys     0m0.071s
```

Τρέχοντας με 2 επεξεργαστές το diffusion2d\_mpi\_nb\_io.c για να δημιουργηθεί το density\_mpi.bin :

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7o Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc
_ex01_code/mpiio$ time mpirun -n 2 ./diffusion2d_mpi_nb_io 1 1 128 50000 0.00001
timestep from stability condition is 1.550003e-05
Timing: 128 4.174669

real    0m4.675s
user    0m8.730s
sys     0m0.033s
```

Υπολογισμός των αθροισμάτων των 2 bin files με το sum\_of\_bin.c :

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7ο Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc_ex01_code/mpiio$ ./sum_of_bin density_seq.bin density_mpi.bin
Sum of values in density_seq.bin: 0.378233
Sum of values in density_mpi.bin: 0.378233
```

Αφού τα αθροίσματα είναι ίσα μπορούμε να συμπεράνουμε ότι η συνάρτηση write\_density\_mpi γράφει τις σωστές τιμές στο density\_mpi.bin.

Επιπλέον, η υλοποίηση είναι σωστή και για μεγαλύτερο αριθμό διεργασιών, 4 εδώ :

```
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7ο Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc_ex01_code/mpiio$ time mpirun -n 4 ./diffusion2d_mpi_nb_io 1 1 128 50000 0.00001
timestep from stability condition is 1.550003e-05
Timing: 128 3.269529

real    0m3.771s
user    0m13.783s
sys      0m0.130s
katherine@LAPTOP-6P6F3R1F:/mnt/c/Users/Katherine/Desktop/CEID/7ο Εξάμηνο/Λογισμικό και Προγραμματισμός Συστ/Ασκήσεις/hpc_ex01_code/mpiio$ ./sum_of_bin density_seq.bin density_mpi.bin
Sum of values in density_seq.bin: 0.378233
Sum of values in density_mpi.bin: 0.378233
```

d. Ο κώδικας για το συγκεκριμένο ερώτημα βρίσκεται στο αρχείο diffusion2d\_mpi\_nb\_io.c στη συνάρτηση write\_density\_mpi\_compressed σε σχόλια. Εδώ παραθέτουμε μια προσπάθεια να κάνουμε πρώτα συμπίεση στα δεδομένα μέσω της zlib βιβλιοθήκης αλλά τελικά δεν ήταν επιτυχής. Η σκέψη μας εδώ ήταν να κάνουμε στη συνέχεια decompress τα δεδομένα, να υπολογίσουμε ξανά τα αθροίσματα με τη βοήθεια του sum\_of\_bin.c και να βρούμε εάν το σφάλμα ήταν λογικό με βάση τις παραμέτρους που βάλαμε στο precision και το rate.

### Question 3: MPI bug hunting and asynchronous communication

Ο κώδικας είναι στο αρχείο ex01q3\_sol.c.

Η σύγχρονη ανταλλαγή μηνυμάτων μεταξύ των διεργασιών με τις εντολές MPI\_Send, MPI\_Receive έχει ως αποτέλεσμα να χάνεται χρόνος κατά την αναμονή για τη λήψη των δεδομένων από τις διεργασίες. Δοκιμάσαμε την εντολή MPI\_Scatter για να πετύχουμε ασύγχρονη επικοινωνία μεταξύ των διεργασιών. Παρακάτω παραθέτουμε τους χρόνους που πήραμε όταν τρέξαμε τους κώδικες για διάφορα πλήθη διεργασιών.

- Για 5 διεργασίες:      ex01q3.c      ex01q3\_sol.c  

real	0m10.646s	real	0m10.577s
user	0m1.354s	user	0m1.210s
sys	0m0.174s	sys	0m0.151s
- Για 25 διεργασίες: :      ex01q3.c      ex01q3\_sol.c  

real	0m12.391s	real	0m12.189s
user	0m7.935s	user	0m7.419s
sys	0m0.581s	sys	0m0.471s
- Για 100 διεργασίες: :      ex01q3.c      ex01q3\_sol.c  

real	0m23.748s	real	0m18.812s
user	0m57.142s	user	0m33.095s
sys	0m1.882s	sys	0m1.884s
- Για 200 διεργασίες: :      ex01q3.c      ex01q3\_sol.c  

real	0m33.915s	real	0m29.425s
user	1m37.214s	user	1m15.622s
sys	0m5.034s	sys	0m3.327s