

# ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ 3 ΠΑΡΑΛΛΗΛΑ ΣΥΣΤΗΜΑΤΑ

ΜΕΛΗ ΕΡΓΑΣΙΑΣ: ΠΑΡΑΣΚΕΥΑΣ ΤΣΙΛΗΣ ΑΜ: 1115202200195

ΜΠΑΛΩΜΕΝΟΣ ΙΑΣΟΝΑΣ ΑΜ: 1115202200104

## ΑΣΚΗΣΗ 1:

ΤΟ ΠΡΟΒΛΗΜΑ ΗΤΑΝ Ο ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΠΙΝΑΚΩΝ, ΑΝΤΙΣΤΟΙΧΑ ΜΕ ΤΙΣ ΠΡΟΗΓΟΥΜΕΝΕΣ ΕΡΓΑΣΙΕΣ. Η ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗ ΓΙΝΕΤΑΙ ΩΣ ΕΞΗΣ:

-ΔΙΑΙΡΕΣΗ ΤΟΥ ΠΡΩΤΟΥ ΠΟΛΥΩΝΥΜΟΥ ΣΕ ΤΜΗΜΑΤΑ ΟΣΕΣ ΟΙ ΔΙΕΡΓΑΣΙΕΣ

-ΔΙΑΝΟΜΗ ΤΟΥ ΔΕΥΤΕΡΟΥ ΠΟΛΥΩΝΥΜΟΥ ΣΕ ΟΛΕΣ ΤΙΣ ΔΙΕΡΓΑΣΙΕΣ(MPI\_BCAST)

-ΑΠΟΣΤΟΛΗ ΤΩΝ ΤΜΗΜΑΤΩΝ ΤΟΥ 1ο ΠΟΛΥΩΝΥΜΟΥ(Η 1 ΔΙΕΡΓΑΣΙΑ ΚΡΑΤΑ ΤΟ ΔΙΚΟ ΤΗΣ ΚΟΜΜΑΤΙ, ΤΑ ΥΠΟΛΟΙΠΑ ΚΟΜΜΑΤΙΑ ΑΠΟΣΤΕΛΟΝΤΑΙ ΣΤΙΣ ΥΠΟΛΟΙΠΕΣ ΔΙΕΡΓΑΣΙΕΣ (MPI\_SEND) ΚΑΙ ΔΙΕΡΓΑΣΙΑ ΛΑΜΒΑΝΕΙ ΤΟ ΤΜΗΜΑ ΤΗΣ ΜΕ MPI\_RECV.

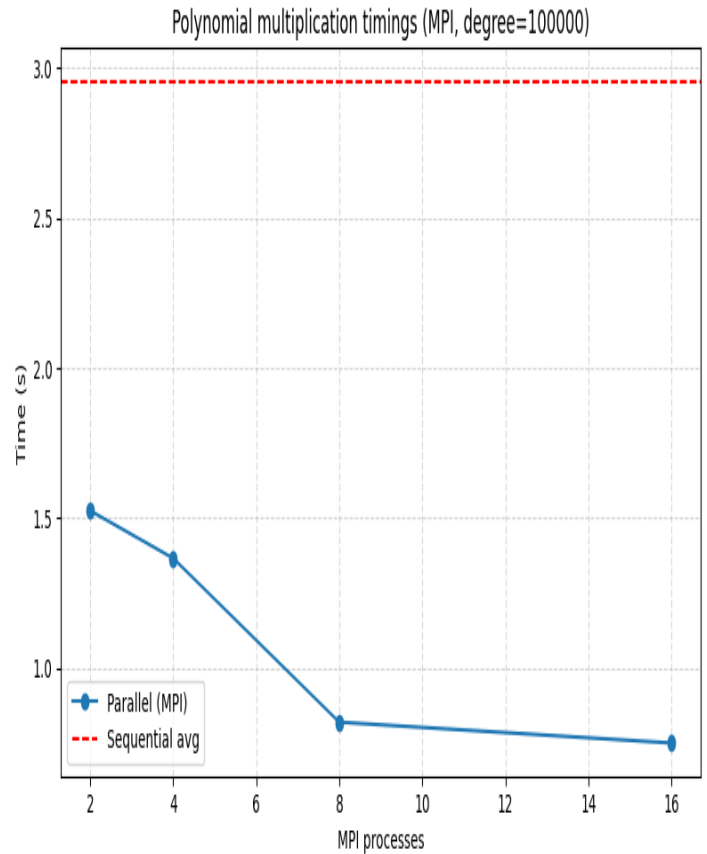
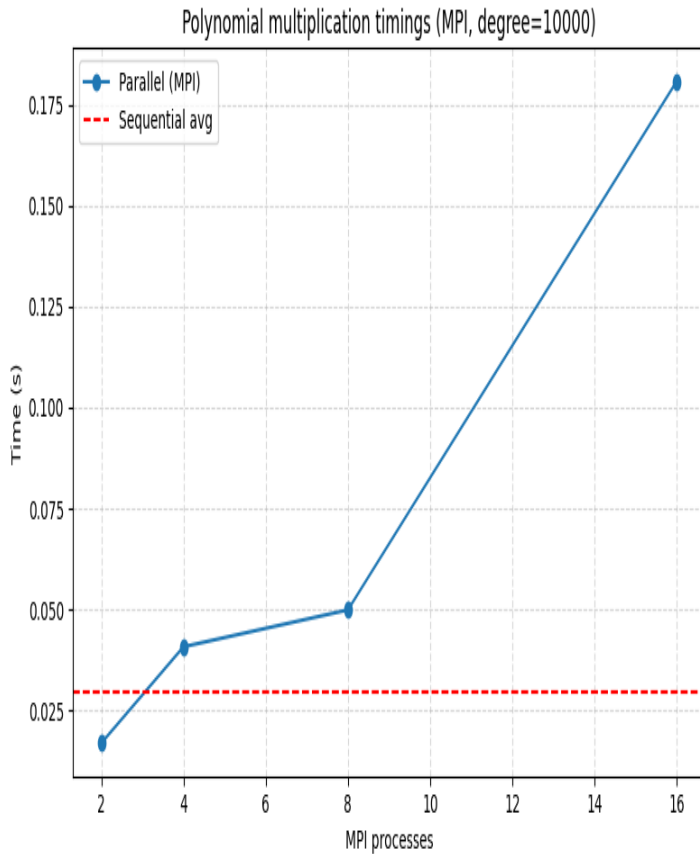
-ΤΟΠΙΚΟΣ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΚΑΘΕ ΔΙΕΡΓΑΣΙΑΣ ΜΕ ΤΟ ΔΕΥΤΕΡΟ ΠΟΛΥΩΝΥΜΟ.

-ΣΥΝΕΝΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΑΠΟ ΤΗΝ RANK 0

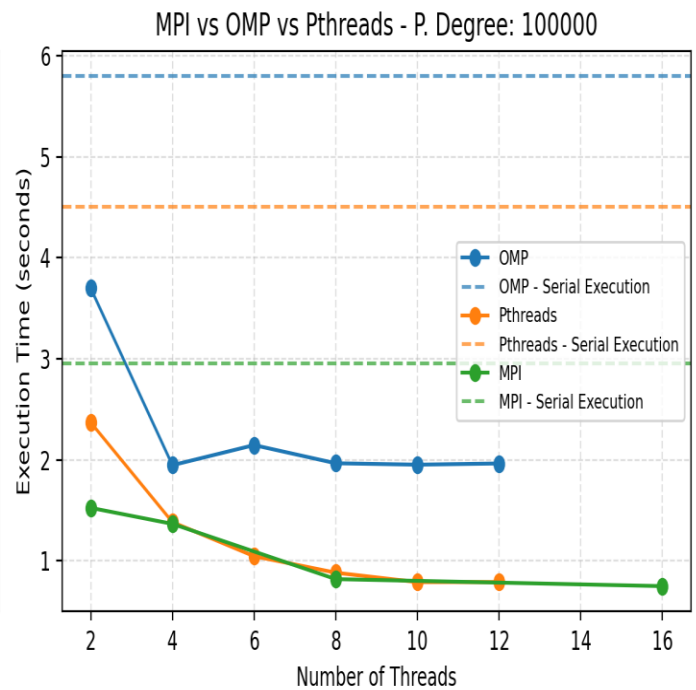
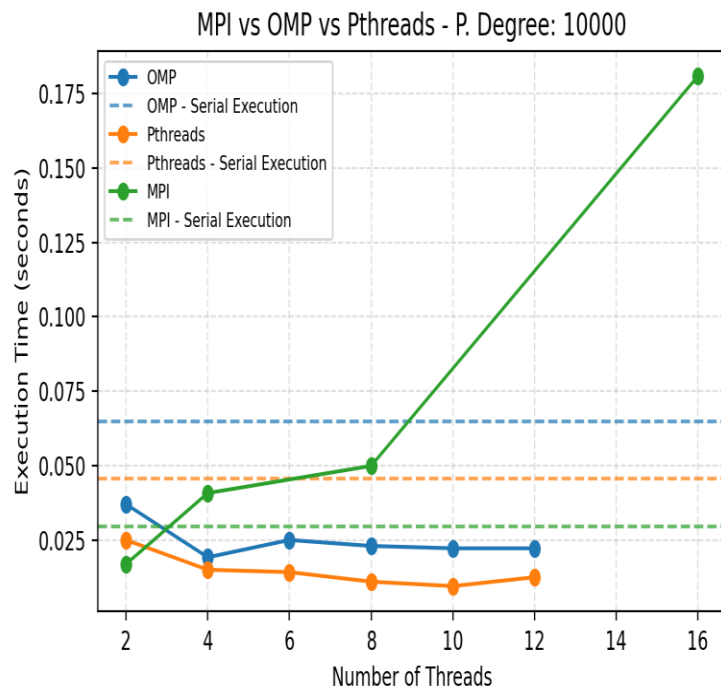
ΠΑΡΑΚΑΤΩ ΕΙΝΑΙ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ:

Parallel multiplication (MPI) — averages (s)

	2	4	8	16	Sequential
POLY. DEGREE: 10000	0.016931	0.040766	0.049888	0.180775	0.029546
POLY. DEGREE: 100000	1.524241	1.365687	0.818579	0.749095	2.956610



Η ΣΥΓΚΡΙΣΗ ΜΕ ΤΑ ΠΡΟΗΓΟΥΜΕΝΑ ΕΡΩΤΗΜΑΤΑ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΡΓΑΣΙΩΝ:



ΠΑΡΑΤΗΡΟΥΜΕ ΟΤΙ ΓΙΑ ΜΙΚΡΟΤΕΡΟ ΒΑΘΜΟ ΠΟΛΥΩΝΥΜΟΥ ΚΑΙ ΠΟΛΛΕΣ ΔΙΕΡΓΑΣΙΕΣ Η ΜΡΙ ΕΙΝΑΙ ΠΙΟ ΑΡΓΗ ΠΙΘΑΝΟΤΑ ΕΞΑΙΤΙΑΣ ΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕΤΑΞΥ ΤΩΝ ΔΙΕΡΓΑΣΙΩΝ (SEND/RECIEV) ΓΙΑ ΜΙΚΡΟ ΑΡΙΘΜΟ ΔΕΔΟΜΕΝΩΝ. ΑΝΤΙΘΕΤΩΣ ΓΙΑ ΜΕΓΑΛΥΤΕΡΟ ΒΑΘΜΟ ΠΟΛΥΩΝΥΜΩΝ ΚΑΙ ΠΟΛΛΕΣ ΔΙΕΡΓΑΣΙΕΣ ΟΠΟΥ ΤΟ

COMPUTATION ΕΙΝΑΙ ΜΕΓΑΛΥΤΕΡΟ Η ΜΡΙ ΕΜΦΑΝΙΖΕΤΑΙ ΝΑ ΕΙΝΑΙ Η ΓΡΗΓΟΡΟΤΕΡΗ ΜΕΘΟΔΟΣ ΑΠΟ ΤΙΣ ΑΛΛΕΣ 2 ΚΑΙ ΛΕΙΤΟΥΡΓΕΙ ΑΠΟΤΕΛΕΣΜΑΤΙΚΑ.

## ΑΣΚΗΣΗ 2:

ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΑΣΚΗΣΗΣ ΗΤΑΝ Η ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΥ ΑΡΑΙΟΥ ΠΙΝΑΚΑ ΜΕ ΔΙΑΝΥΣΜΑ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ ΜΡΙ.

ΕΓΙΝΕ ΣΥΓΚΡΙΣΗ 2 ΔΙΑΦΟΡΕΤΙΚΩΝ ΑΝΑΠΑΡΑΣΤΑΣΕΩΝ

- COMPRESSED SPARSE ROW (CSR) ΑΝΑΠΑΡΑΣΤΑΣΗ
  - ο ΜΕΤΑΤΡΟΠΗ ΠΥΚΝΟΥ ΠΙΝΑΚΑ ΣΕ ΑΝΑΠΑΡΑΣΤΑΣΗ CSR.
  - ο ΔΙΑΜΟΙΡΑΣΜΟΣ ΓΡΑΜΜΩΝ ΤΟΥ ΠΙΝΑΚΑ ΙΣΟΜΕΡΩΣ ΣΤΙΣ ΔΙΕΡΓΑΣΙΕΣ
  - ο ΣΥΛΛΟΓΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΑΠΟ ΟΛΕΣ ΤΙΣ ΔΙΕΡΓΑΣΙΕΣ ΚΑΙ ΑΠΟΣΤΟΛΗ ΤΟΥ ΤΕΛΙΚΟΥ ΑΠΟΤΕΛΕΣΜΑΤΟΣ ΣΕ ΟΛΕΣ ΤΙΣ ΔΙΕΡΓΑΣΙΕΣ ΩΣΤΕ ΝΑ ΓΙΝΕΙ ΕΚ ΝΕΟΥ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ (ΤΟ ΑΠΟΤΕΛΕΣΜΑ  $y$  ΕΙΝΑΙ Η ΕΙΣΟΔΟΣ  $x$  ΣΤΗΝ ΕΠΟΜΕΝΗ ΕΠΑΝΑΛΗΨΗ) ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΗΣ `MPI_Allgatherv()`.
- ΠΥΚΝΗ ΑΝΑΠΑΡΑΣΤΑΣΗ

ΓΙΑ ΠΟΙΚΙΛΕΣ ΤΙΜΕΣ ΠΑΡΑΛΛΗΛΩΝ ΔΙΕΡΓΑΣΙΩΝ, ΚΑΙ ΤΩΝ ΥΠΕΡΠΑΡΑΜΕΤΡΩΝ: ΜΕΓΕΘΟΣ ΠΙΝΑΚΑ, ΠΥΚΝΟΤΗΤΑ ΚΑΙ ΕΠΑΝΑΛΗΨΕΩΝ ΤΟΥ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΥ.

ΠΑΡΟΥΣΙΑΖΟΝΤΑΙ ΜΕΡΙΚΑ ΑΠΟΤΕΛΕΣΜΑ ΑΠΟ ΤΙΣ ΕΚΤΕΛΕΣΕΙΣ ΚΑΙ ΤΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΟΥ ΠΡΟΚΥΠΤΟΥΝ ΑΠΟ ΑΥΤΑ:

TIMING   N = 10.000   Sparsity = 0,2   Iterations = 10				
PROCESSES	CSR_CALC	DENSE_CALC	CSR_TOTAL	DENSE_TOTAL
1	1,582	1,292	3,121	1,292
2	0,967	0,650	10,904	1,833
4	0,752	0,565	6,658	3,048
8	0,527	0,434	13,391	5,249

Σε πυκνούς πίνακες (χαμηλό sparsity), ο υπολογισμός στον CSR αλγόριθμος είναι πιο αργός από τον κανονικό.

TIMING   N = 1.000   Sparsity = 0,5   Iterations = 10				
PROCESSES	CSR_CALC	DENSE_CALC	CSR_TOTAL	DENSE_TOTAL
1	0,017	0,006	0,046	0,006
2	0,002	0,012	0,134	0,034
4	0,173	0,213	0,274	0,347

Για μικρό μέγεθος πίνακα, ο χρόνος υπολογισμού εκτοξεύεται όσο αυξάνονται και τα processes. Η χρήση MPI δεν αποδίδει λόγω του υψηλού overhead επικοινωνίας, η σειριακή εκτέλεση είναι αποδοτικότερη.

TIMING   N = 10.000   Sparsity = 0,5   Iterations = 20				
PROCESSES	CSR_CALC	DENSE_CALC	CSR_TOTAL	DENSE_TOTAL
1	1,494	1,889	2,753	1,889
2	1,048	1,183	3,740	2,154
4	1,015	1,028	5,444	2,953
8	0,706	0,740	7,056	6,038

Ο χρόνος υπολογισμού μειώνεται όσο αυξάνονται τα processes και στους 2 αλγορίθμους. Ωστόσο, ο συνολικός χρόνος αυξάνεται λόγω του υψηλού communication time.

TIMING   N = 10.000   Sparsity = 0,8   Iterations = 20				
PROCESSES	CSR_CALC	DENSE_CALC	CSR_TOTAL	DENSE_TOTAL
1	0,643	1,842	1,363	1,842
2	0,422	1,009	1,622	1,848
4	0,476	0,858	2,280	2,765
8	0,453	0,557	4,391	4,926

CSR αναπαράσταση είναι ταχύτερος από τον αντίστοιχο της πυκνής αναπαράστασης. Τόσο ο χρόνος υπολογισμού όσο και ο συνολικό χρόνος με CSR είναι καλύτερος.

TIMING   N = 10.000   Processes = 8   Iterations = 20			
Sparsity	0,2	0,5	0,8
CSR_construction	1,804	1,990	0,978
CSR_communication	7,227	4,360	2,960
DENSE_communication	3,320	5,299	4,369
CSR_calculation	0,940	0,706	0,453
DENSE_calculation	0,800	0,740	0,557
CSR_total	9,971	7,056	4,391
DENSE_total	4,120	6,039	4,926
CSR_Serial	4,530	2,753	1,363
DENSE_Serial	2,690	1,889	1,842

Ο CSR αλγόριθμος εκτελεί ταχύτερα το κομμάτι υπολογισμού από τον κανονικό. Το υψηλότερο κόστος επικοινωνίας όμως οδηγεί σε πιο χρονοβόρα συνολική εκτέλεση.

Για αραιούς πίνακες (0.8 sparsity), ο υπολογισμός με CSR αναπαράσταση είναι ταχύτερος από τον αντίστοιχο της πυκνής αναπαράστασης. Τόσο ο χρόνος υπολογισμού όσο και ο συνολικό χρόνος με CSR είναι καλύτερος.

Σε κάθε περίπτωση, η σειριακή εκτέλεση είναι ταχύτερη από την αντίστοιχη παράλληλη με χρήση της MPI. Το overhead της επικοινωνίας καταστρέφει τα οφέλη της παράλληλης εκτέλεσης.

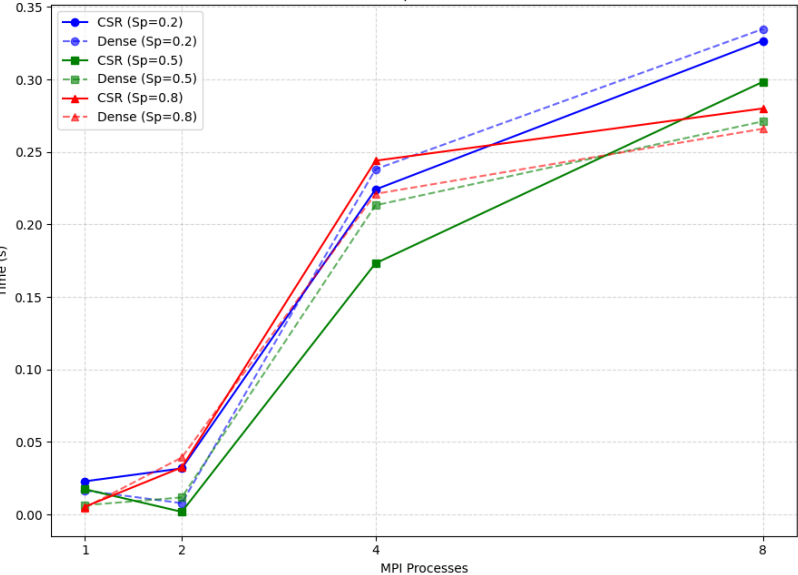
Η CSR αναπαράσταση συμπεριφέρεται καλύτερα από την πυκνή(κλασσική) αναπαράσταση για αρκετά αραιούς πίνακες (80% αραιούς και πάνω).

ΤΑ ΣΥΣΤΗΜΑΤΑ ΣΤΑ ΟΠΟΙΟ ΕΚΤΕΛΕΣΤΗΚΑΝ ΤΑ ΠΕΙΡΑΜΑΤΑ:

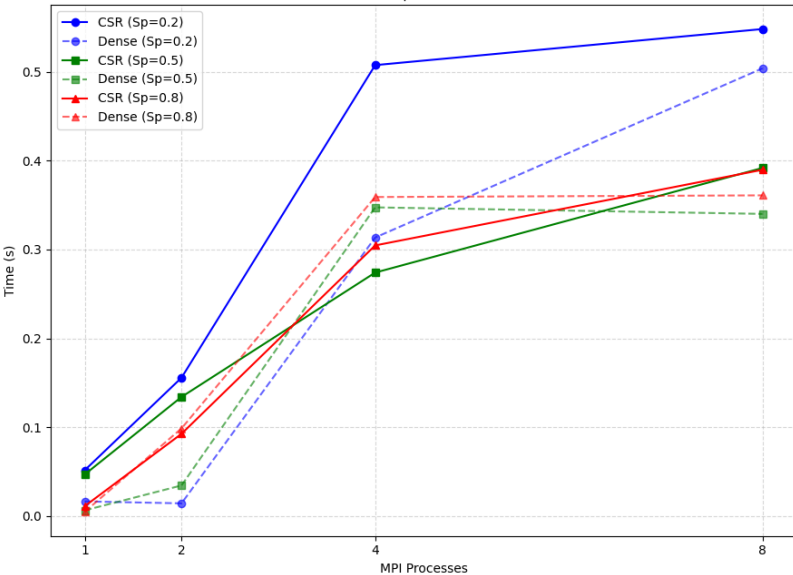
(DESKTOP-OQLOAOM)

Architecture: x86\_64 CPU op-mode(s): 32-bit, 64-bit Address sizes: 48 bits physical, 48 bits virtual Byte Order: Little Endian CPU(s): 12 On-line CPU(s) list: 0-11 Vendor ID: AuthenticAMD Model name: AMD

Pure Calculation Time (N=1000, Iter=10)  
Solid = CSR | Dashed = Dense



Total Execution Time (N=1000, Iter=10)  
Solid = CSR | Dashed = Dense



Ryzen 5 3600 6-Core Processor CPU family: 23 Model: 113 Thread(s) per core: 2 Core(s) per socket: 6  
Socket(s): 1 Stepping: 0 BogoMIPS: 7186.51 Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr\_opt pdpe1gb rdtscp lm  
rep\_good nopl cpuid extd\_apicid tsc\_known\_freq pni pclm ulqdq ssse3 fma cx16 sse4\_1 sse4\_2 movbe  
popcnt aes xsave avx f16c rdrand hypervisor lahf\_lm cmp\_legacy cr8\_legacy abm sse4a misalignsse  
3dnowprefetch osvw topoext ssbd ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 rdseed adx smap  
clflushopt clwb sha\_ni xsaveopt xsavec xgetbv1 clzero xsaveerptr virt\_ssbd arat umip rdpid

ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ: Linux DESKTOP-OQLOAOM 6.6.87.2-microsoft-standard-WSL2 #1 SMP  
PREEMPT\_DYNAMIC Thu Jun 5 18:30:46 UTC 2025 x86\_64 x86\_64 x86\_64 GNU/Linux

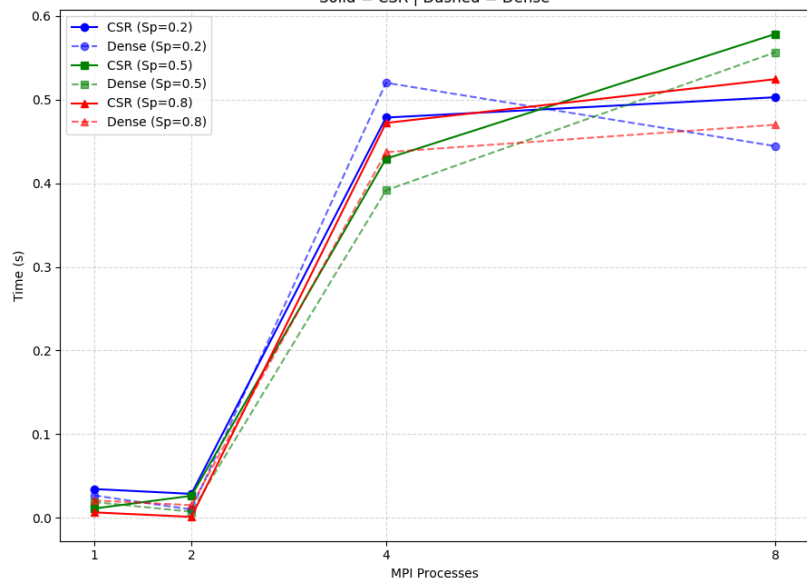
MΕΤΑΓΛΩΤΙΣΤΗΣ: gcc (Ubuntu 11.4.0-2ubuntu1~20.04) 11.4.0

LINUX10(ΣΧΟΛΗΣ):

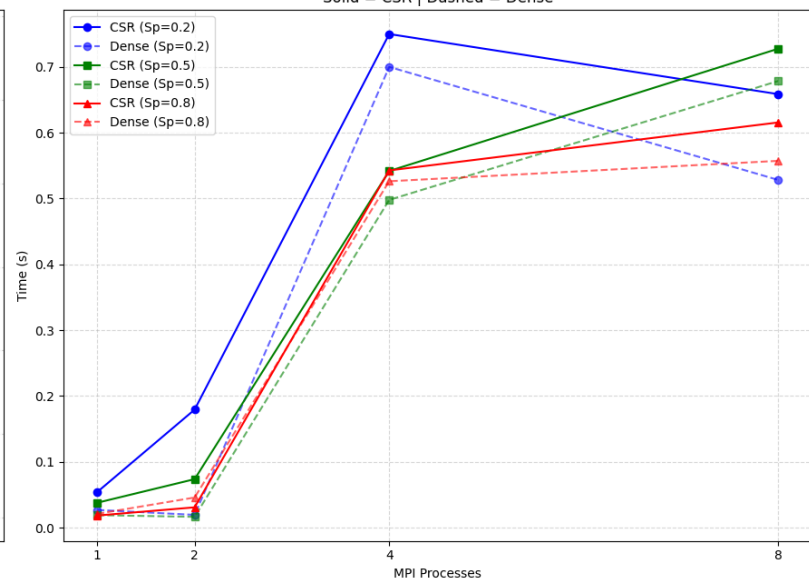
Architecture: x86\_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian Address sizes:  
39bits physical, 48bits virtual CPU(s): 4 On-line CPU(s) list: 0-3 Thread(s) per core: 1 Core(s) per socket:  
4 Socket(s): 1 NUMA node(s): 1 Vendor ID: GenuineIntel CPU family: 6 Model: 94 Model  
name: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz Stepping: 3 CPU MHz: 3393.126 CPU max MHz:  
3600,0000 CPU min MHz: 800,0000

ΟΛΑ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ/ΓΡΑΦΗΜΑΤΑ ΕΙΝΑΙ ΑΝΑΛΥΤΙΚΑ ΣΤΟΥΣ ΦΑΚΕΛΟΥΣ ΤΗΣ ΕΡΓΑΣΙΑΣ RESULTS/PLOTS  
ΑΝΤΙΣΤΟΙΧΑ

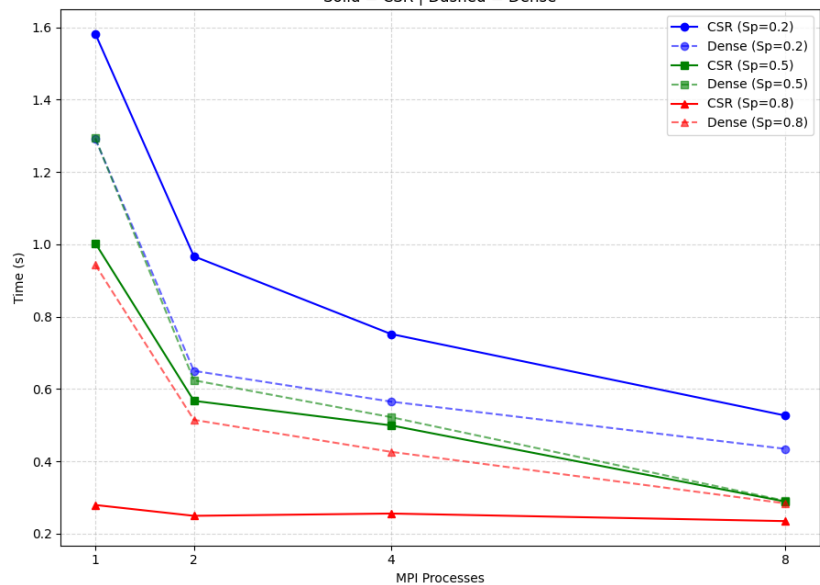
Pure Calculation Time (N=1000, Iter=20)  
Solid = CSR | Dashed = Dense



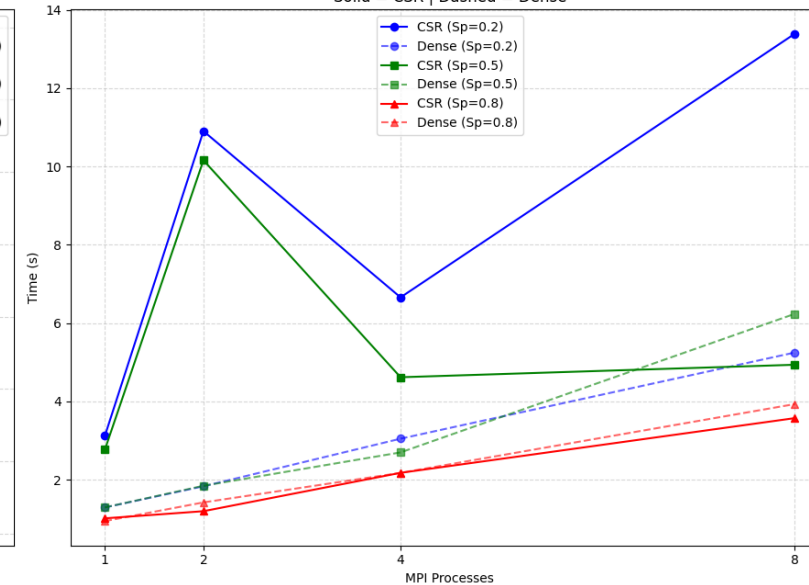
Total Execution Time (N=1000, Iter=20)  
Solid = CSR | Dashed = Dense



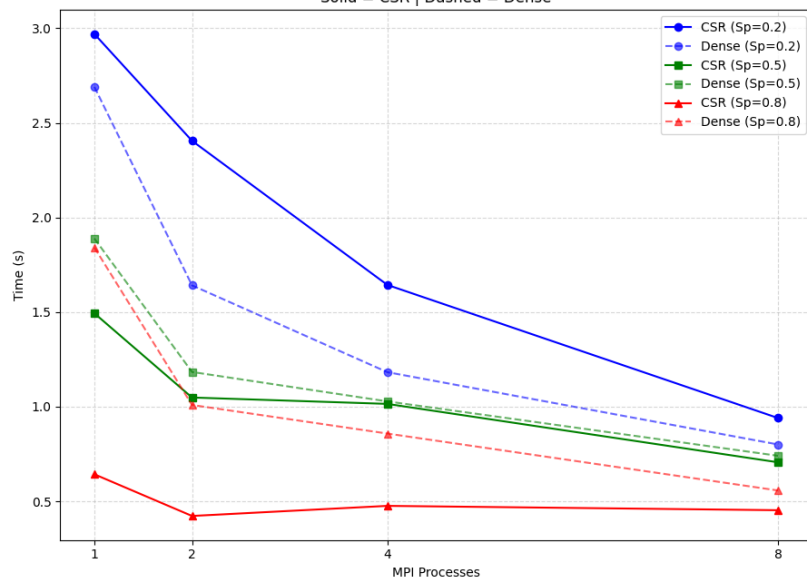
Pure Calculation Time (N=10000, Iter=10)  
Solid = CSR | Dashed = Dense



Total Execution Time (N=10000, Iter=10)  
Solid = CSR | Dashed = Dense



Pure Calculation Time (N=10000, Iter=20)  
Solid = CSR | Dashed = Dense



Total Execution Time (N=10000, Iter=20)  
Solid = CSR | Dashed = Dense

