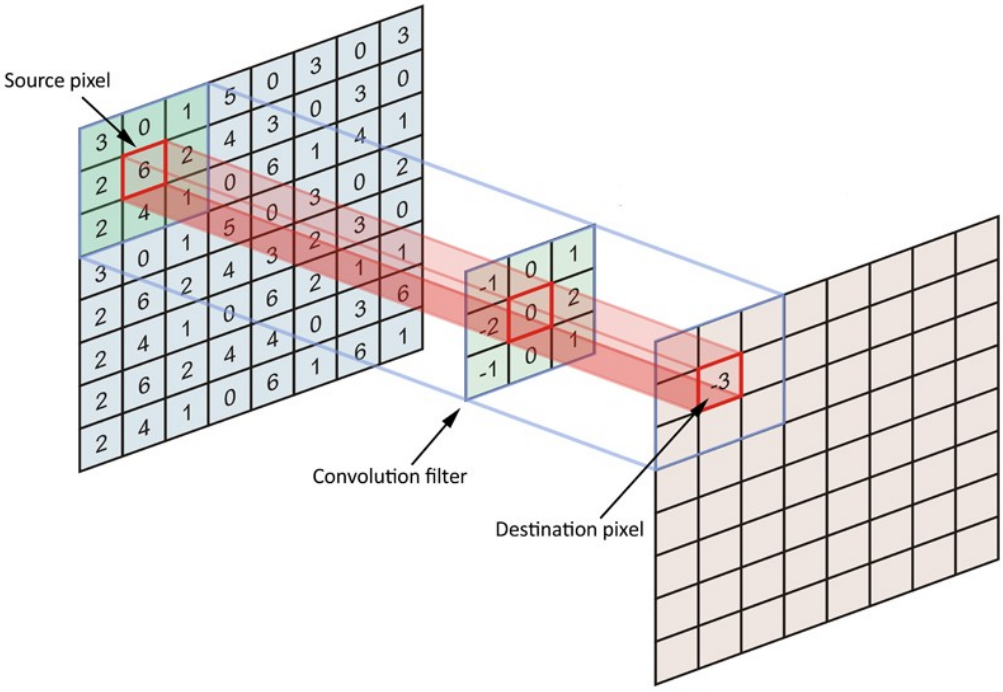
ΣΥΝΕΛΙΞΗ ΕΙΚΟΝΩΝ ΜΕ MPI ΚΑΙ OpenMP

ΠΑΡΑΛΛΗΛΑ ΣΥΣΤΗΜΑΤΑ – Project

Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών
2017-2018

Μαρκέλλα Γκιόκα 1115201400269
Παναγιώτα Ψυχάρη 1115201500184



MPI

ΜΕΤΑΓΛΩΤΙΣΗ ΚΑΙ ΕΚΤΕΛΕΣΗ
COMPILE: mpicc mpi_convolution -o mpi_convolution
EXECUTE: mpiexec -n <num_of_processes> ./mpi_convolution <image_path> <image_type> <image_width> <image_height> <conv_reps>

image_path: μονοπάτι εικόνας στο file system
image_type: rgb ή grey ανάλογα την εικόνα
image_width: μήκος εικόνας
image_height: ύψος εικόνας
conv_reps: πλήθος επαναλήψεων για την εφαρμογή συνέλιξης
* Η σειρά και το είδος των arguments είναι ίδια και στην εκτέλεση του κώδικα του MPI + OpenMP.

ΑΝΑΠΤΥΞΗ ΚΩΔΙΚΑ:

- Η διεργασία root ελέγχει την εγκυρότητα των arguments της εκτέλεσης και έπειτα κάνει broadcast τις τιμές τους στις υπόλοιπες διεργασίες.
- Σε περίπτωση που η εικόνα δεν μπορεί να χωριστεί σε ίσα κομμάτια με βάση το πλήθος των διεργασιών, τότε συγκεκριμένες διεργασίες αναλαμβάνουν λίγο μεγαλύτερο μέρος του προβλήματος από άλλες. Για παράδειγμα κάποιες μπορεί να έχουν μία παραπάνω στήλη ή/και σειρά από pixels
- Επικοινωνία διεργασιών. Δημιουργείται νέος communicator με καρτεσιανή τοπολογία δύο διαστάσεων η οποία χρησιμοποιείται έτσι ώστε διεργασίες που χρειάζεται να επικοινωνούν να είναι σχετικά κοντά, μειώνοντας το χρονικό κόστος επικοινωνίας.
- Χρησιμοποιείται MPI_Datatype PIXEL το οποίο ορίζεται να έχει μέγεθος 1 ή 3 bytes ανάλογα με τον τύπο χρώματος, και χρησιμοποιείται για το διάβασμα του αρχείου
- Παράλληλο I/O. Με τη χρήση της συνάρτησης MPI_Type_create_darray γίνεται χωρισμός του αρχείου σε τμήματα ανάλογα με το πλήθος και τη τοπολογία των διεργασιών. Στη συνέχεια μπορεί η κάθε διεργασία να διαβάσει και να γράψει το κομμάτι που της αναλογεί παράλληλα με τις άλλες.
- Ο πίνακας (local_buf) της κάθε διεργασίας στον οποίον αποθηκεύονται οι τιμές των pixels που αναλαμβάνει είναι σειριακός. Σε σχεδιαστικό επίπεδο, ο πίνακας περιστοιχίζεται από hollow points. Σειριακός είναι επίσης και ο πίνακας temp_buf στον οποίο αποθηκεύονται τα αποτελέσματα της συνέλιξης. Οι δύο πίνακες έχουν ίδιο μέγεθος.
- Πριν την διαδικασία της συνέλιξης υπολογίζονται οι γειτονικές διεργασίες μέσα στη καρτεσιανή τοπολογία της κάθε process και αποθηκεύονται σε σειριακό πίνακα.
- Το default φίλτρο που χρησιμοποιείται είναι GAUSSIAN_BLUR και είναι κανονικοποιημένο.
- Χρησιμοποιείται non-blocking επικοινωνία μεταξύ των διεργασιών.
- Τα hollow points κάθε διεργασίας αρχικοποιούνται σε μια default τιμή. Έτσι ακόμη και για τα pixels των processes που δεν δέχονται hollow points μπορεί να πραγματοποιηθεί συνέλιξη.
- Στη περίπτωση που μια διεργασία είναι ακριανή και δεν έχει γείτονες για να στείλει και να λάβει δεδομένα, οι τιμές τους θέτονται MPI_PROC_NULL έτσι ώστε οι συναρτήσεις Isend και Irecv να ολοκληρώνονται χωρίς να συμβαίνει τίποτα.
- DATATYPES:

→ PIXEL: Ανάλογα με το είδος της εικόνας (rgb ή grey) μετρά 3 bytes ή 1 byte. Η χρήση του αυτοματοποιεί και γενικεύει την εκτέλεση του προγράμματος αφού δεν χρειάζεται να γίνεται σε μετέπειτα στάδια έλεγχος του είδους της εικόνας για την πραγματοποίηση ανάλογων ενεργειών.

→ PIXELS_BLOCK: Χρησιμοπείται στο διάβασμα των pixels από την εικόνα στον local_buf. Αποφεύγεται η επαναληψημότητα και η συνεχόμενη αναζήτηση εντός του αρχείου για λήψη των κατάλληλων bytes.
→ row_type: Χρησιμοποιείται στη λήψη/αποστολή σειρών από pixels από/σε γείτονες διεργασίες.
→ column_type: Χρησιμοποιείται στη λήψη/αποστολή σειρών από pixels από/σε γείτονες διεργασίες.

ΒΗΜΑΤΑ_ΣΥΝΕΛΙΞΗΣ:

1. Αποστολή outer pixels σε όλους τους γείτονες.
2. Non-blocking λήψη hollow point pixels από όλους του γείτονες.
3. Συνέλιξη inner pixels.
4. Με waitany και testany αναμονή για λήψη όλων των πλευρικών hollow points. Συνέλιξη κάθε πλευράς outer pixels για την οποία έχουν επιστραφεί τα ανάλογα hollow points.
5. Με waitany και testany αναμονή για λήψη όλων των γωνιακών hollow points. Συνέλιξη κάθε γωνιακού outer pixel για το οποίο έχει επιστραφεί το ανάλογο hollow point. Η συνέλιξη σαυτήν την περίπτωση μπορεί να πραγματοποιηθεί επιτυχώς αφού είναι διαθέσιμα όλα τα hollow points (2 πλευρές και 1 γωνία).
6. Αναμονή me waitall για ολοκλήρωση αποστολής outer pixels στις γειτονικές διεργασίες.
7. Ανταλλαγή pointers του local_buf με το temp_buf.
8. Έλεγχος σύγκλισης αποτελέσματος τρέχουσας σύγκλισης με τις προηγούμενες τιμές της. Ο έλεγχος πραγματοποιείται ανά συγκεκριμένες επαναλήψεις.

MPI + OpenMP

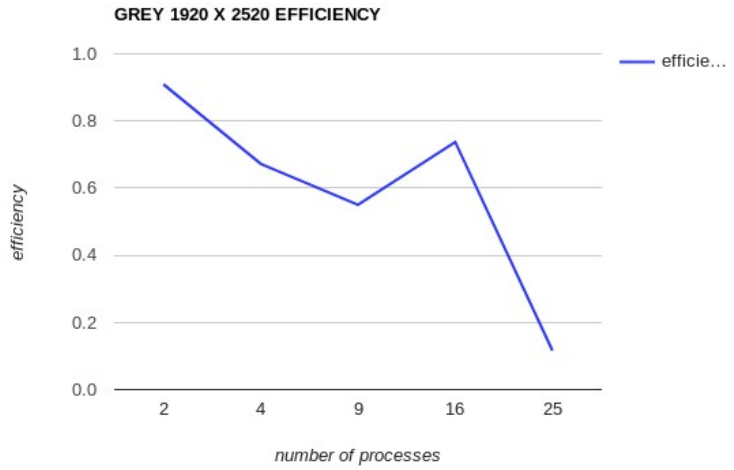
ΜΕΤΑΓΛΩΤΙΣΗ ΚΑΙ ΕΚΤΕΛΕΣΗ
COMPILE: mpicc -openmp mpi_convolution -o mpi_convolution
EXECUTE: mpiexec -n <num_of_processes> ./mpi_convolution <image_path> <image_type> <image_width> <image_height> <conv_reps>

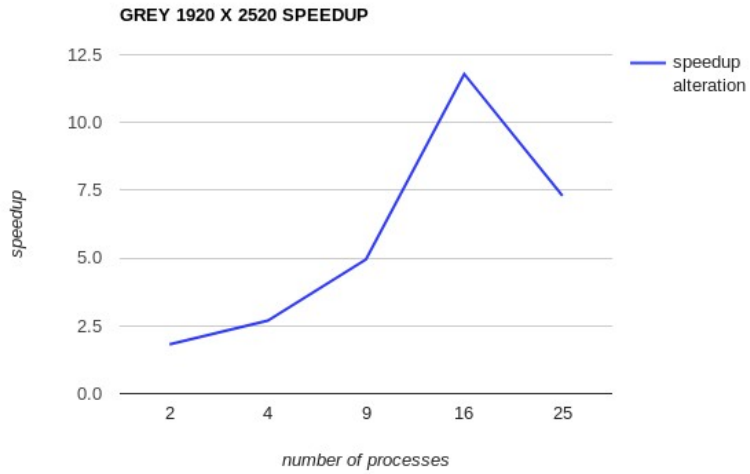
Ο κώδικας και η λογική που ακολουθείται είναι ακριβώς η ίδια με αυτή του MPI, με την διαφορά ότι σε σειριακές ενέργειες της κάθε διεργασίας (συνέλιξη εσωτερικών pixels, αλλά και γενικά ενός εκτενούς block από pixels) δημιουργούνται νήματα για την παραλληλοποίηση και εξισορρόπηση τους φόρτου εργασίας της διεργασίας. Ιδιάιτερα σε μεγάλο πλήθος pixels, η χρήση τους επηρεάζει άμεσα το χρονικό αποτέλεσμα.

ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΣΧΟΛΙΑΣΜΟΙ

MPI

Size/Processes	1	2	4	9	12	25
Grey ¼	1.09	0.51	0.26	0.15	0.12	0.09
Grey ½	2.12	1.05	0.55	0.30	0.50	0.17
Grey	4.60	2.53	1.71	0.93	0.39	0.63
Grey x2	7.82	4.72	2.66	0.80	0.97	0.42
Rgb ¼	3.08	1.74	0.62	0.37	0.73	1.39
Rgb ½	6.65	3.44	3.06	1.19	0.66	0.95
Rgb	12.60	6.72	3.43	1.85	0.80	1.30
Rgb x2	25.33	12.67	6.73	3.51	0.95	1.10



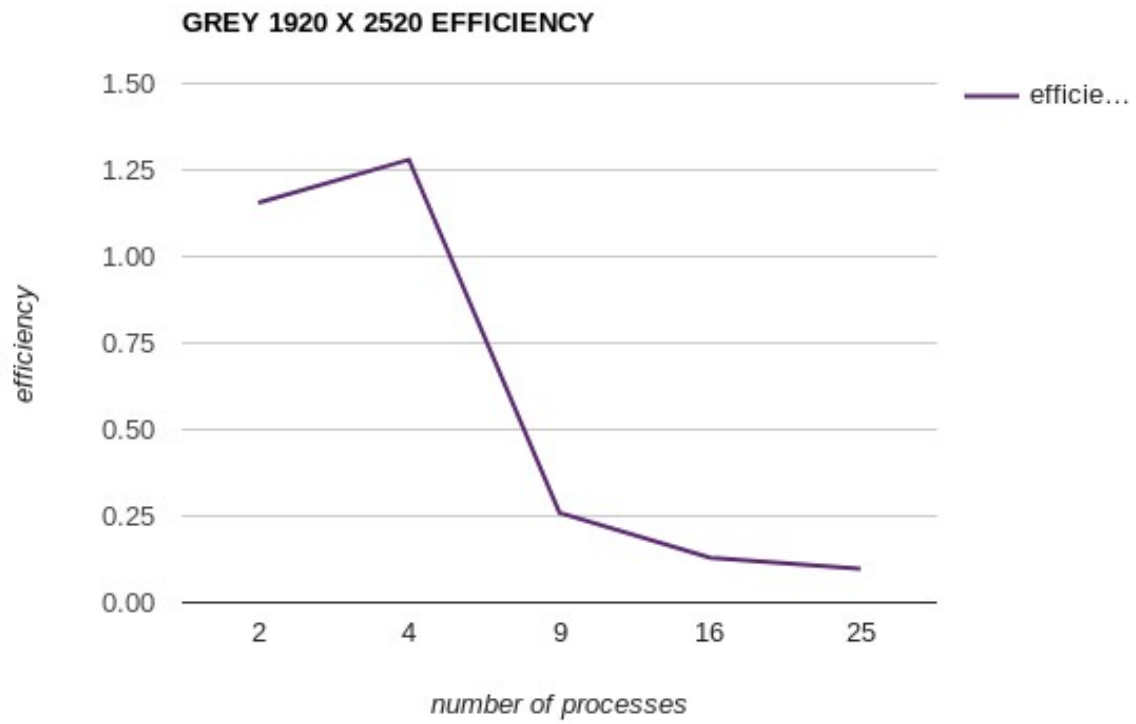


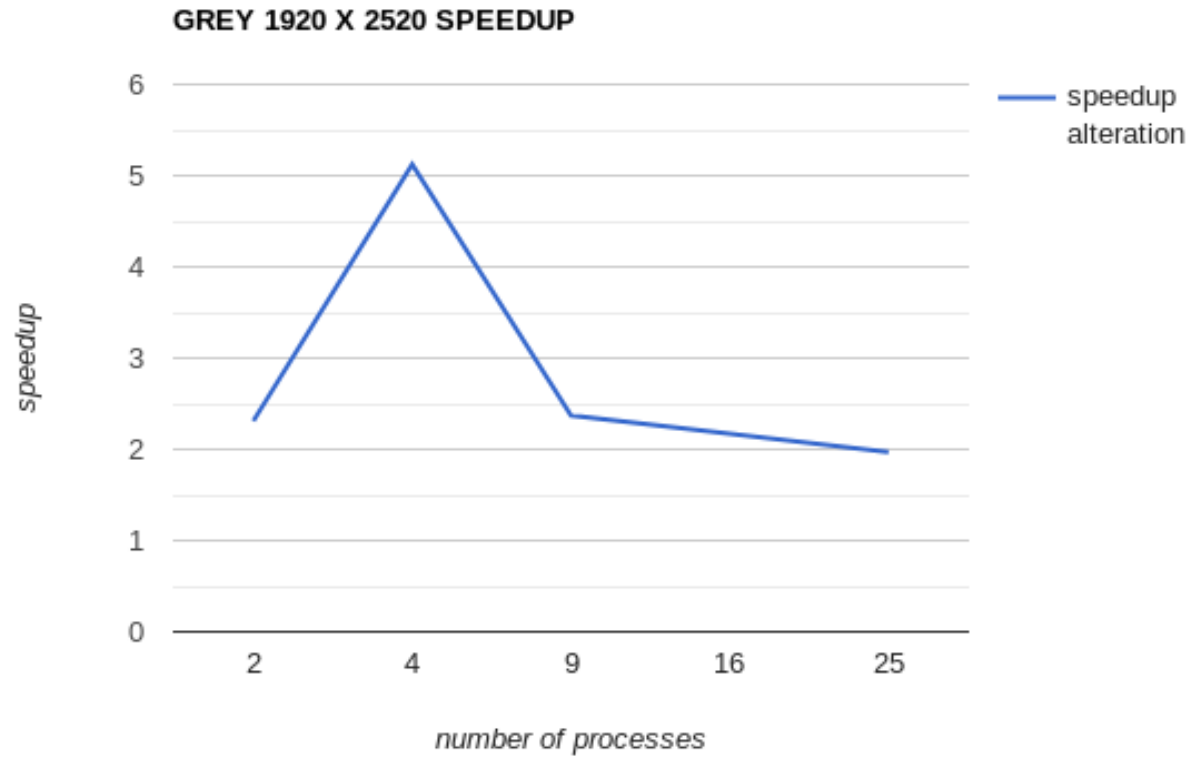

RGB 1920 X 2520 EFFICIENCY
1.0
0.8
0.6
0.4
0.2
0.0
efficiency
2
4
9
16
25
number of processes
efficie…

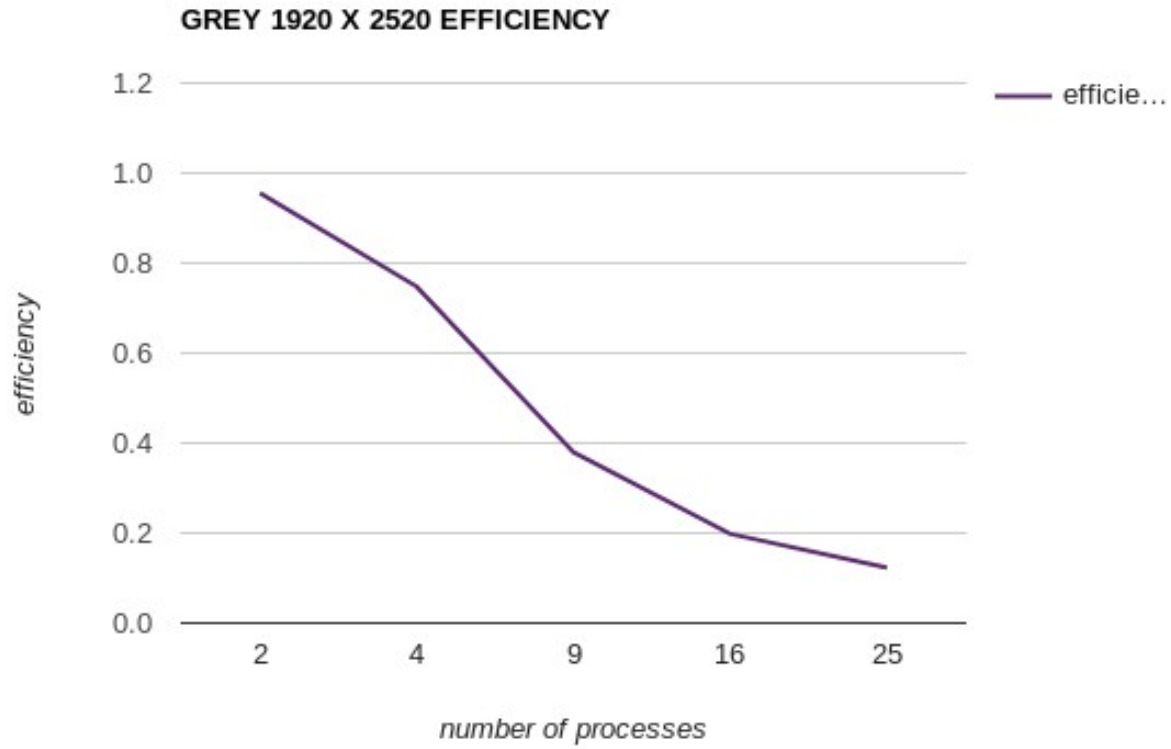

RGB 1920 X 2520 SPEEDUP
15
10
5
0
speedup
2
4
9
16
25
number of processes
speedup
alteration

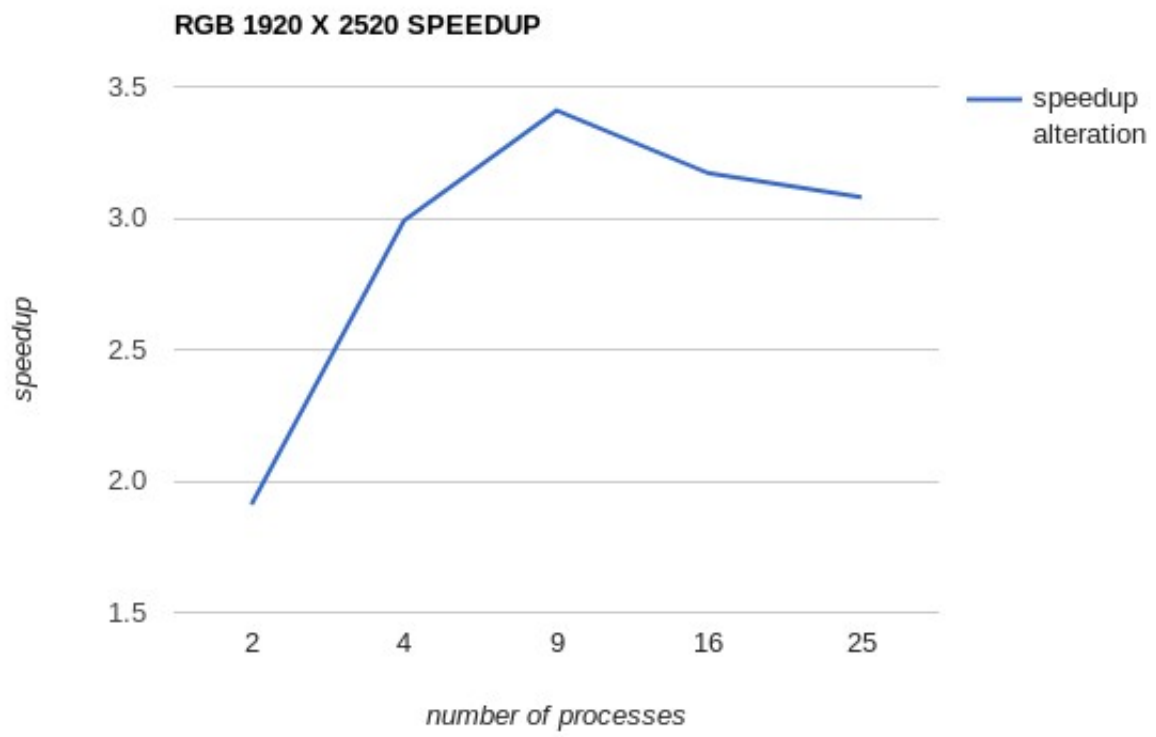
OpenMP+MPI

Size/Processes	1	2	4	9	16	25
Grey ¼	1.10	0.51	0.53	0.18	0.11	0.55
Grey ½	2.20	1.03	0.86	0.56	0.49	0.61
Grey	5.18	2.02	1.07	0.55	0.39	0.73
Grey x2	8.48	3.93	2.09	1	0.25	0.61
Rgb ¼	3.09	1.67	0.64	0.83	0.51	0.48
Rgb ½	6.33	3.38	1.18	0.64	0.42	0.70
Rgb	12.53	6.75	3.42	1.25	1.16	0.74
Rgb x2	24.06	13.92	11.29	2.20	1.45	1.13




GREY 1920 X 2520 SPEEDUP
6
5
4
3
2
1
0
speedup
2
4
9
16
25
number of processes
speedup
alteration


GREY 1920 X 2520 EFFICIENCY
1.2
1.0
0.8
0.6
0.4
0.2
0.0
efficiency
2
4
9
16
25
number of processes
efficie…



Παρατηρήσεις

Όταν αυξάνονται τα δεδομένα και το πλήθος των διεργασιών τότε χρόνος εκτέλεσης θα πρέπει να παραμένει όσο πιο σταθερός γίνεται. Αν τα δεδομένα παραμένουν σταθερά αλλά αυξάνεται το πλήθος των διεργασιών, ο χρόνος πρέπει να μειώνεται ανάλογα με τον ρυθμό αύξησης των διεργασιών. Στην περίπτωση του mpi και του mpi+openmp, για εικόνα συγκεκριμένου σταθερού μεγέθους, όσο αυξάνεται το πλήθος των διεργασιών μειώνεται ο elapsed time και παρατηρείται επιτάχυνση. Επίσης παρατηρούμε ότι καθώς αυξάνονται το μέγεθος της εικόνας και το πλήθος των διεργασιών, οι χρόνοι είναι παραπλήσιοι. Επομένως και στην περίπτωση του MPI και του MPI+OpenMP παρατηρείται κλιμάκωση.