

Εργασία Παράλληλων Συστημάτων 2017-2018

Συνέλιξη Εικόνων

Η εργασία αυτή εκπονήθηκε από τους:
Πασχόπουλος Κωνσταντίνος (1115201500127)
Ταβουλάρης Φώτης Έκτορας(1115201500154)

Οκτώβριος 2018

Παράλληλα Υπολογιστικά Συστήματα

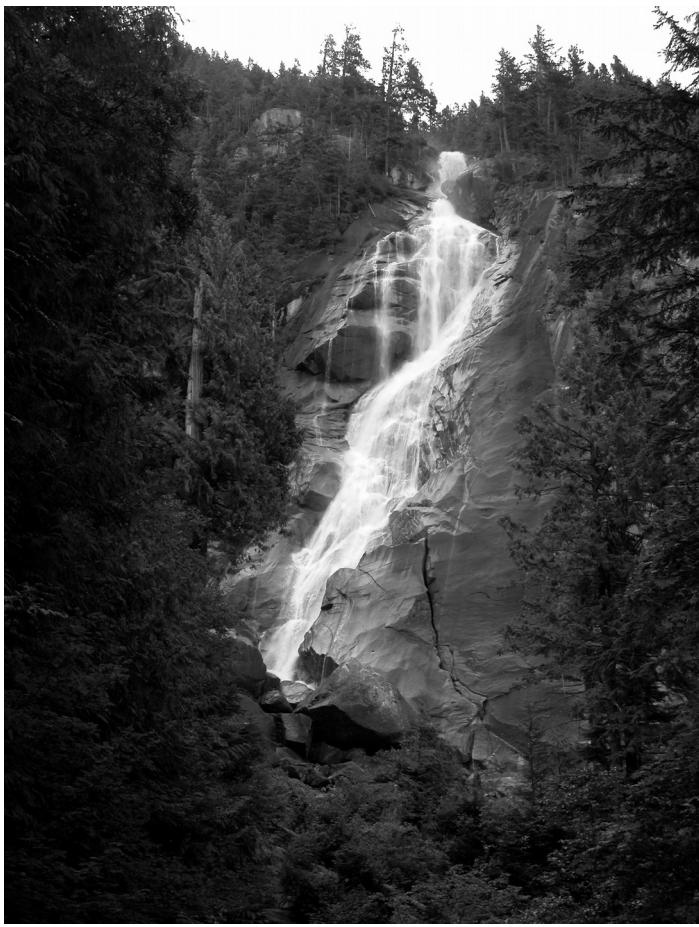
ΠΕΡΙΕΧΟΜΕΝΑ

1. ΠΡΟΛΟΓΟΣ - ΣΥΝΟΨΗ.....	3
1.1 ΘΕΩΡΗΤΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....	3
1.2 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	4
2. ΣΧΟΛΙΑ ΠΑΝΩ ΣΤΟΝ ΚΩΔΙΚΑ.....	10
2.1 Σχολιασμός του MPI κώδικα.....	10
2.2 Σχολιασμός του MPI+OpenMP κώδικα.....	11
2.3 Σχολιασμός του CUDA κώδικα.....	11
2.4 Hardware Constraints for CUDA.....	12
2.5 Optimization for CUDA.....	12
3. ΧΡΟΝΙΚΕΣ ΜΕΤΡΗΣΕΙΣ ΜΕ ΧΡΗΣΗ MPI.....	14
3.1 ΕΠΙΠΛΕΟΝ ΧΡΟΝΙΚΕΣ ΜΕΤΡΗΣΕΙΣ ΓΙΑ ΤΟ MPI.....	20
4. ΧΡΟΝΙΚΕΣ ΜΕΤΡΗΣΕΙΣ ΜΕ ΧΡΗΣΗ CUDA.....	22
5. ΠΕΡΙΕΧΟΜΕΝΑ ΕΡΓΑΣΙΑΣ.....	24
6. ΕΠΙΛΟΓΟΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ.....	24
7. ΠΗΓΕΣ - ΒΙΒΛΙΟΓΡΑΦΙΑ.....	25

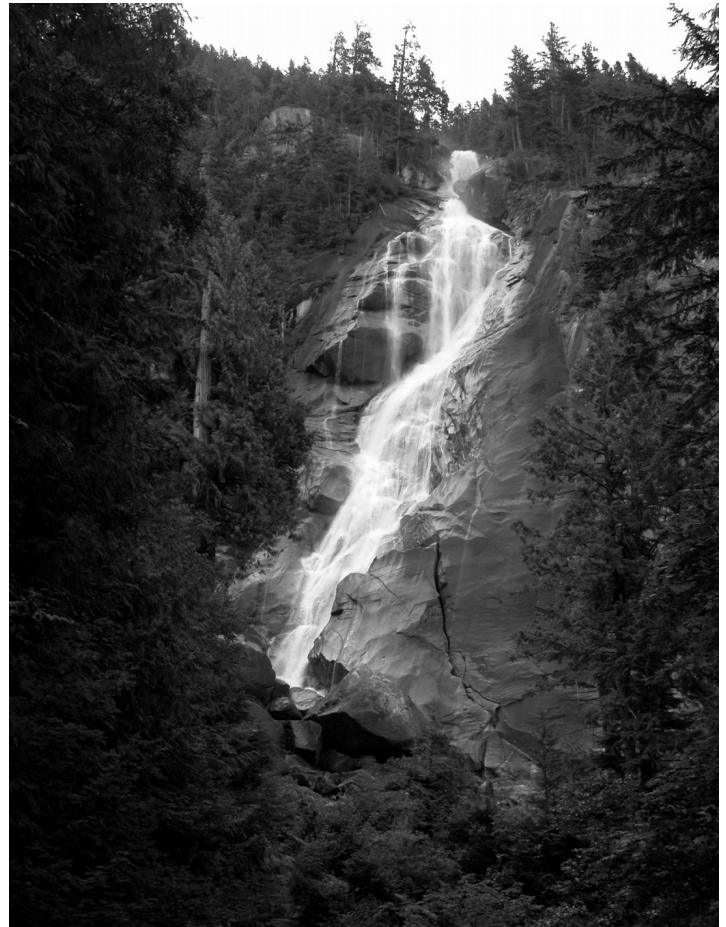
Παράλληλα Υπολογιστικά Συστήματα

Αναφορικά με το CUDA κομμάτι, καταλήξαμε ότι είναι καλύτερα να χρησιμοποιήσουμε τρισδιάστατες δομές, καθώς μας προσφέρουν μεγαλύτερη ελαστικότητα στις δοκιμές με διάφορες μορφές και συνδυασμούς από threads. Σε αυτό το σημείο βλέπουμε και τη χρήση των παραμέτρων `threadX`, `threadY`, `threadZ`. Στις παραπάνω παραδοχές καταλήξαμε με την βοήθεια των σημειώσεων και των σχετικών άρθρων της nvidia.

1.2 ΑΠΟΤΕΛΕΣΜΑΤΑ



Η εικόνα "waterfall_grey_1920_2520.raw" χωρίς το φίλτρο



Η εικόνα "waterfall_grey_1920_2520.raw" εφαρμόζοντας το φίλτρο μία φορά

Παράλληλα Υπολογιστικά Συστήματα



10 Επαναλήψεις



20 Επαναλήψεις

Παράλληλα Υπολογιστικά Συστήματα



40 Επαναλήψεις

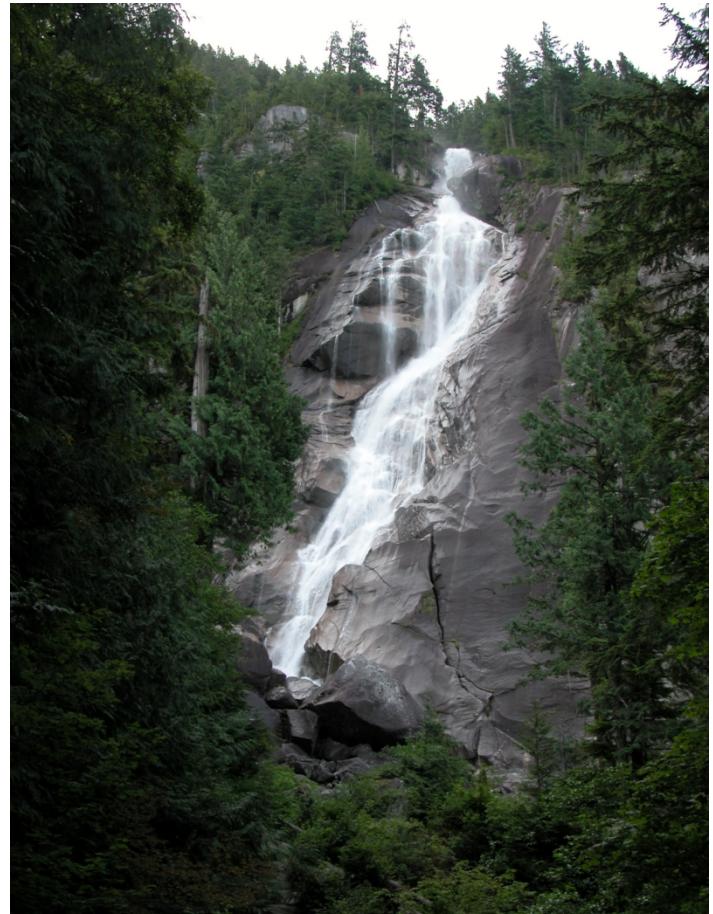


80 Επαναλήψεις

Παράλληλα Υπολογιστικά Συστήματα



Η εικόνα "waterfall_rgb_1920_2520.raw" χωρίς το φίλτρο

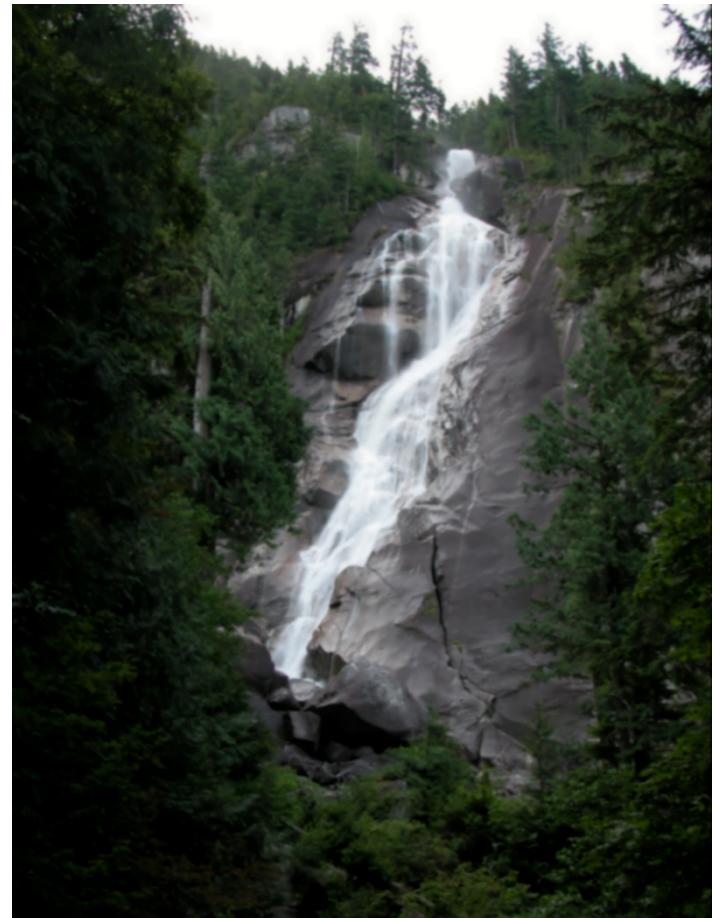


Η εικόνα "waterfall_rgb_1920_2520.raw" εφαρμόζοντας το φίλτρο μία φορά

Παράλληλα Υπολογιστικά Συστήματα



10 Επαναλήψεις

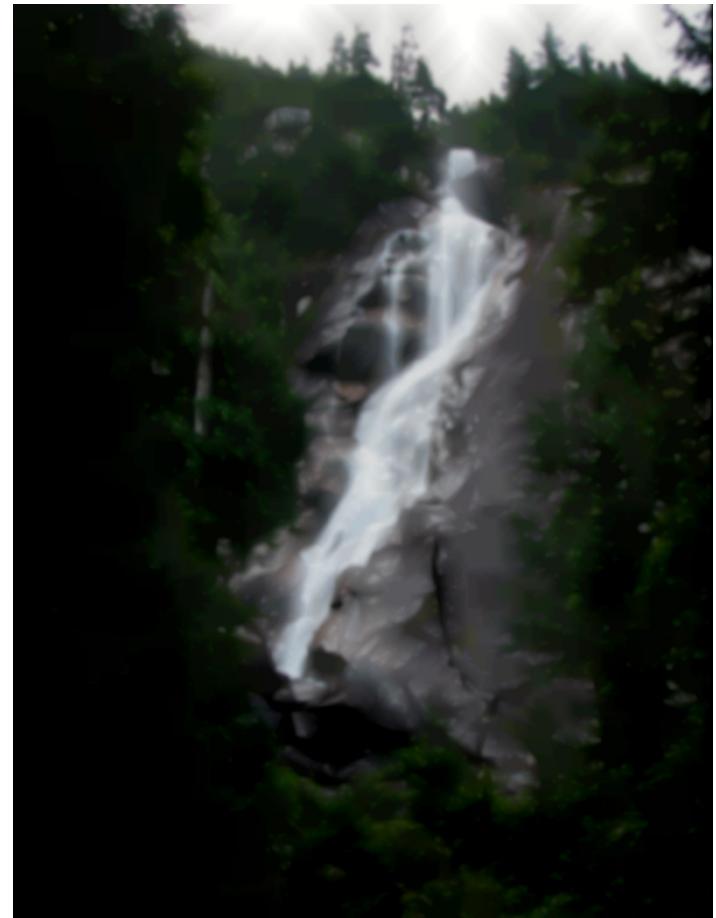


20 Επαναλήψεις

Παράλληλα Υπολογιστικά Συστήματα



40 Επαναλήψεις



80 Επαναλήψεις

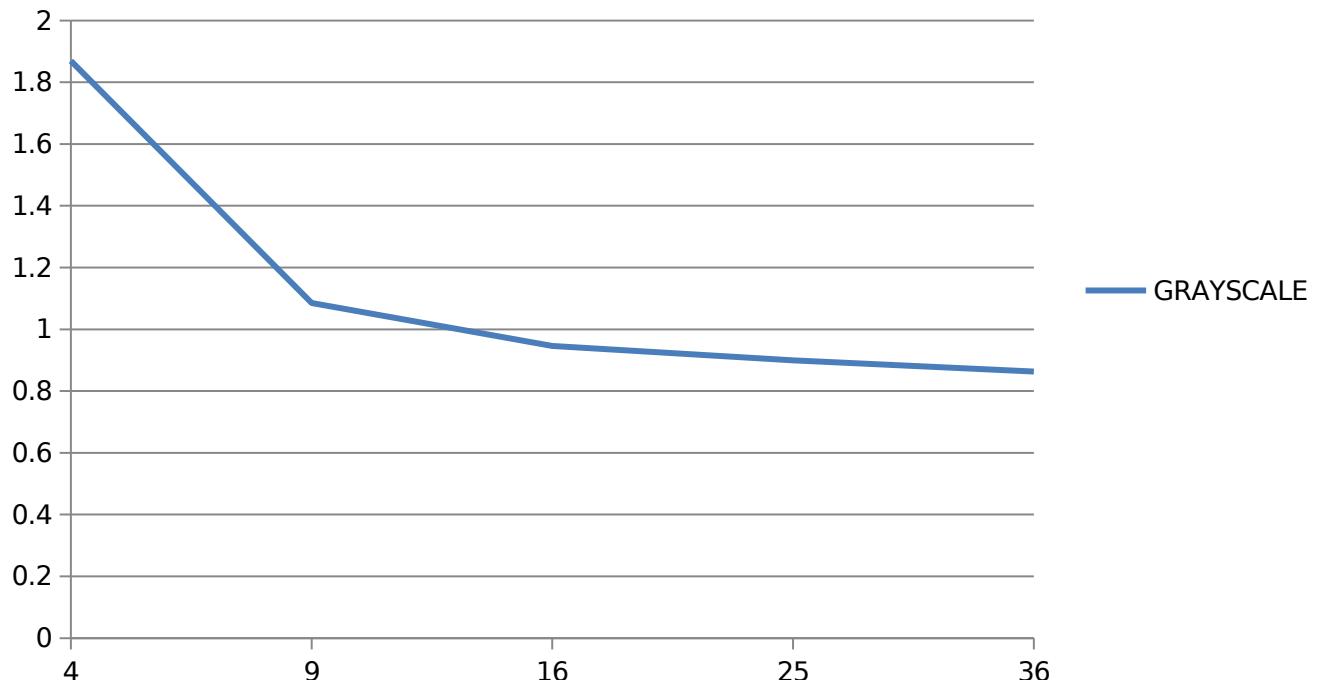
Παράλληλα Υπολογιστικά Συστήματα

(Στην περίπτωση όμως που δεν ισχύει αυτό, δεν δημιουργούνται προβλήματα καθώς κάθε warp λειτουργεί κανονικά, και για αυτό δεν το έχουμε ως περιορισμό.)

Εξίσου σημαντικός παράγοντας στην απόδοση είναι το να έχουμε σε λειτουργία επαρκές πλήθος από warps και κατά συνέπεια να πετύχουμε optimal hardware occupancy, δηλαδή να χρησιμοποιούνται όλοι οι πόροι της κάρτας γραφικών.

Παράλληλα Υπολογιστικά Συστήματα

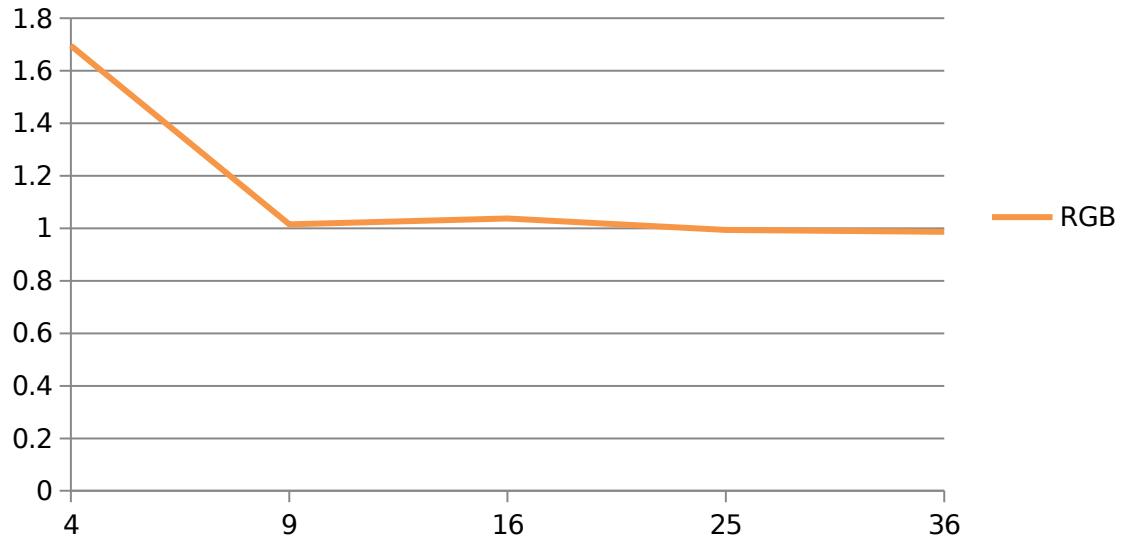
GRAYSCALE



RGB				
4	9	16	25	36
1.696	1.015	1.037	0.994	0.987

Παράλληλα Υπολογιστικά Συστήματα

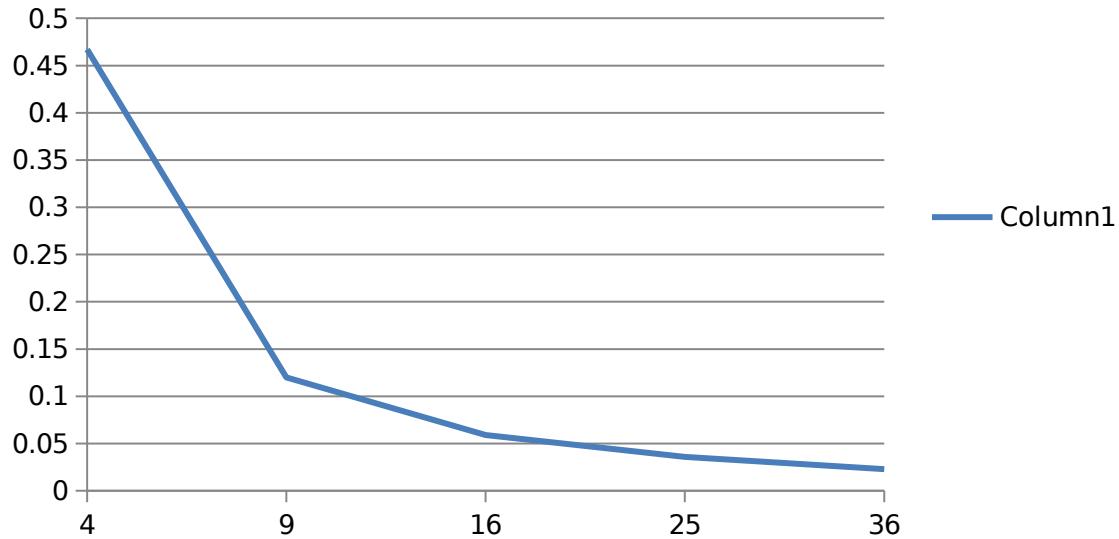
RGB



EFFICIENCY - E = S(n)/n

Grayscale				
4	9	16	25	36
0.467	0.12	0.059	0.036	0.023

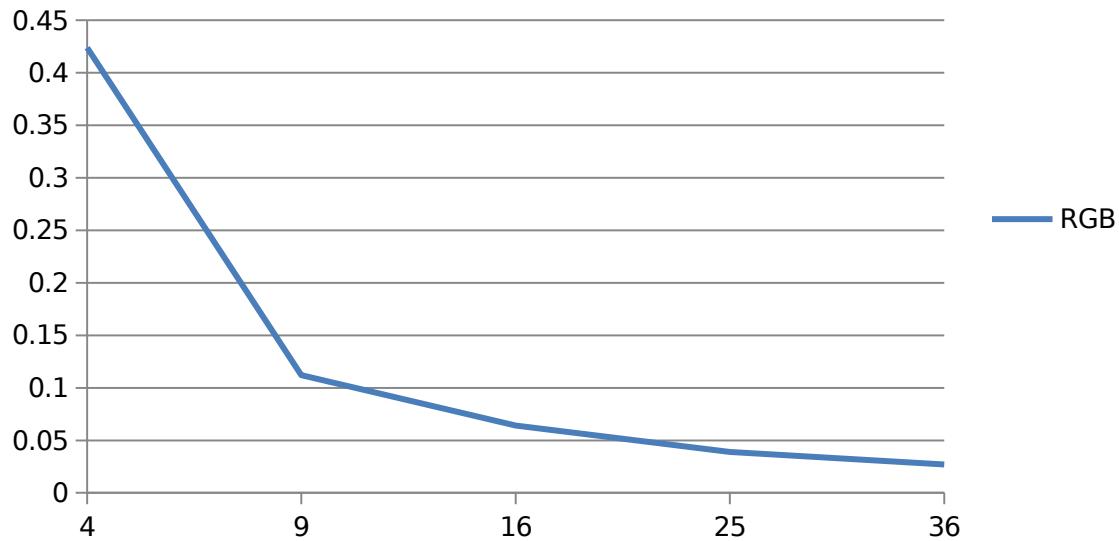
GRAYSCALE



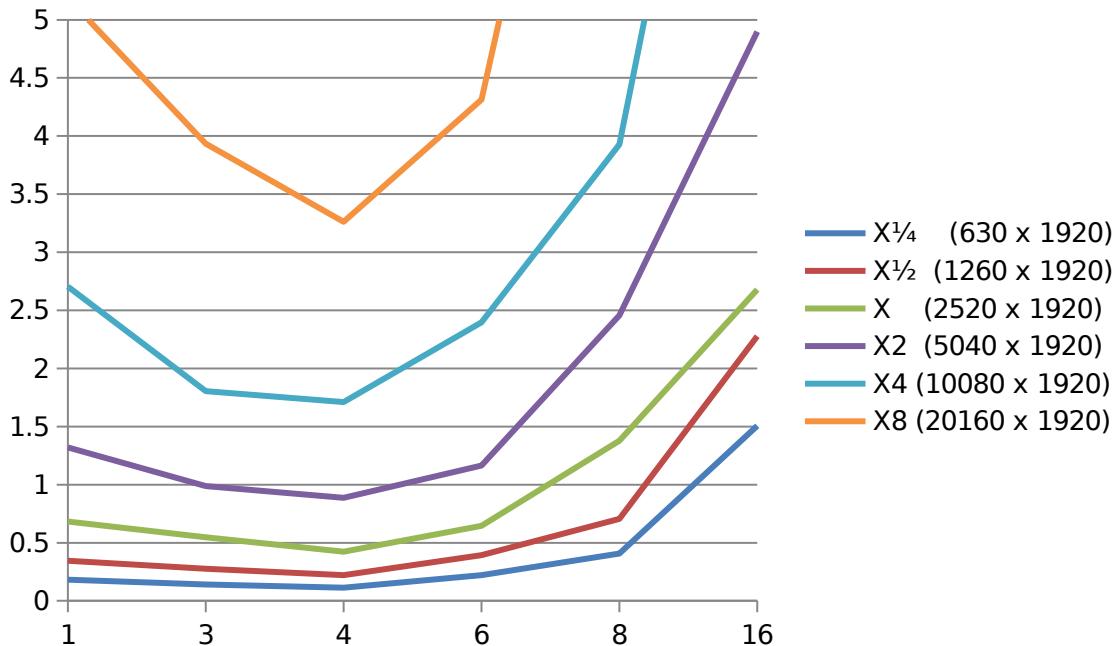
RGB				
4	9	16	25	36
0.424	0.112	0.064	0.039	0.027

Παράλληλα Υπολογιστικά Συστήματα

RGB

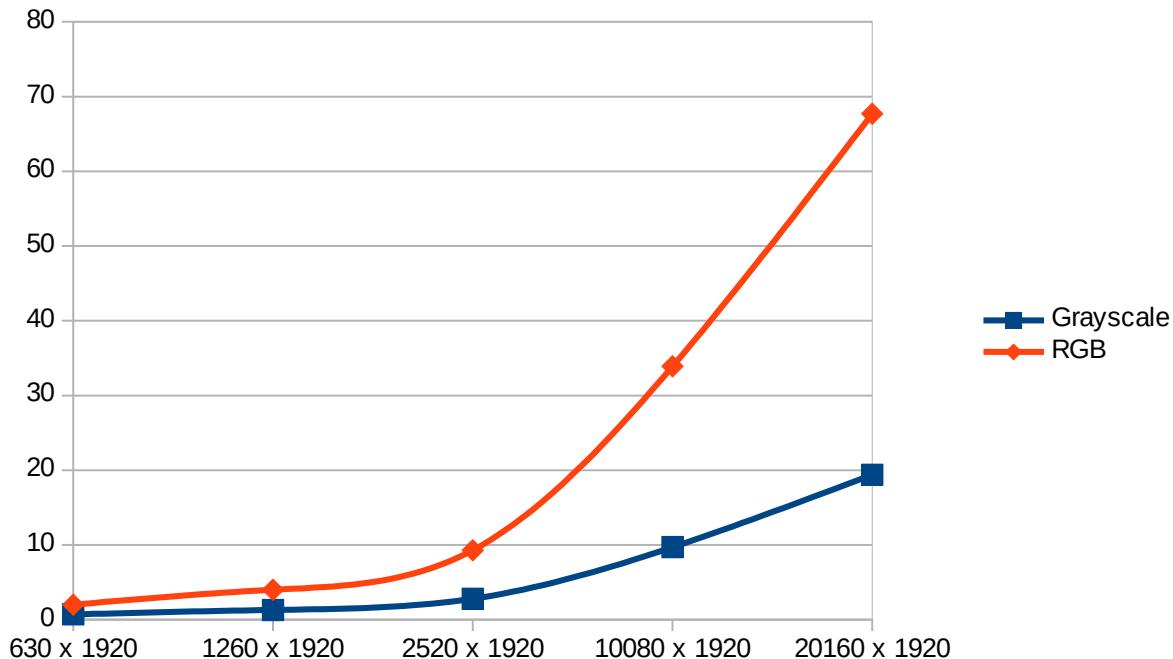


Παράλληλα Υπολογιστικά Συστήματα



Σε αυτή την περίπτωση παρατηρούμε ότι μέχρι τις 4 διεργασίες ο συνολικός χρόνος μειώνεται. Στη συνέχεια όμως όταν χρησιμοποιούμε παραπάνω από 4 διεργασίες, ο συνολικός χρόνος αρχίζει να αυξάνεται και να ξεπερνάει τον σειριακό χρόνο κατά πολύ. Αυτό είναι ένα αποτέλεσμα το οποίο περιμέναμε, καθώς ο συγκεκριμένος επεξεργαστής έχει μόνο 2 cores και 4 threads. Αυτό που συμβαίνει ονομάζεται oversubscribing και είναι το φαινόμενο κατά το οποίο δημιουργούμε περισσότερες διεργασίες από τους διαθέσιμους processors (4 σε αυτή την περίπτωση).

Παράλληλα Υπολογιστικά Συστήματα



Όπως βλέπουμε, όσο αυξάνεται το πρόβλημα έχουμε ανάγκη για μεγαλύτερα blocks από threads που στην αρχική περίπτωση που το πρόβλημα ήταν μικρότερο θα μας καθυστερούσαν (καθώς θα είχαμε κάποια πιο αργά threads τα οποία θα περιμέναμε να τελειώσουν πριν εμφανίσουμε το αποτέλεσμα).

Πιο αναλυτικά η σχέση μεγέθους προβλήματος και αριθμού - μορφής των threads, φαίνεται στα αρχεία: [cuda_file_rgb.csv](#) και [cuda_file_grayscale.csv](#) που έχουμε συμπεριλάβει στο φάκελο.

5. ΠΕΡΙΕΧΟΜΕΝΑ ΕΡΓΑΣΙΑΣ

Ο φάκελος της εργασίας περιέχει:

- Τα αρχεία του κώδικα.
 1. mpi/mpifilter.c, mpi/mpifunctions.c, mpi/mpifunctions.h
 2. openmp/openmpfilter.c, openmp/openmpfunctions.c, openmp/openmpfunctions.h
 3. cuda/cudafilter.cu, cuda/cudafunctions.cu, cuda/cudafunctions.h
- Ένα Makefile σε κάθε φάκελο.
- Τα προαναφερόμενα `cuda_file_rgb.csv` και `cuda_file_grayscale.csv`.

6. ΕΠΙΛΟΓΟΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ

Με τη χρήση παράλληλων δομών και με τις πολυάριθμες δοκιμές που εκτελέσαμε πάνω σε αυτές, μπορούμε να κατανοήσουμε τη χρησιμότητά τους σε προβλήματα μεγαλύτερου μεγέθους. Έτσι καταλήγουμε στο συμπέρασμα ότι, η παραλληλία είναι ένα σημαντικό εργαλείο για να επιτύχουμε ταχύτητα και εγκυρότητα στα προγραμματιστικά προβλήματα που συναντάμε.

7. ΠΗΓΕΣ - ΒΙΒΛΙΟΓΡΑΦΙΑ

- Σημειώσεις μαθήματος “Παράλληλα Υπολογιστικά Συστήματα” - Γιάννης Κοτρώνης
- Παρουσιάσεις Nvidia