# Προγραμματισμός Συστημάτων Υψηλών Επιδόσεων (ΗΥ 421/ΜΔΕ 646)
# Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
# Πανεπιστήμιο Θεσσαλίας


## Διδάσκων: Χρήστος Δ. Αντωνόπουλος


## 3η Εργαστηριακή Άσκηση

**Στόχος**: Μικρές αλγοριθμικές αλλαγές, εκμετάλλευση ιεραρχίας μνήμης GPU, αξιολόγηση επίδοσης.

**Βήματα**:
Στην προηγούμενη άσκηση κατασκευάσατε και αξιολογήσατε πειραματικά κώδικα ο οποίος μπορεί να πραγματοποιήσει συνέλιξη 2 διαστάσεων με χρήση γραμμικού φίλτρου, πρώτα κατά γραμμές και μετά κατά στήλες πάνω σε "οποιουδήποτε" μεγέθους τετραγωνικές εικόνες, εντός των περιορισμών χωρητικότητας της global memory της GPU. Σε αυτήν την άσκηση θα επαναχρησιμοποιήσετε αυτόν τον κώδικα και πιο συγκεκριμένα τον κώδικα που πραγματοποιεί συνέλιξη σε floats. Επειδή σε αυτή την άσκηση δεν συγκρίνουμε επίδοση CPU με GPU, στις μετρήσεις χρόνου δε χρειάζεται να συμπεριλαμβάνετε τις μεταφορές μνήμης και το χρόνο δέσμευσης / αποδέσμευσης μνήμης. Αν δεν καταφέρατε να υλοποιήσετε τον κώδικα με την τεχνική "padding" του προηγούμενου lab, μπορείτε να εκτελέσετε όλα τα βήματα που ακολουθούν στον κώδικα με τα divergences. Σε όλα τα ερωτήματα χρησιμοποιήστε φίλτρο με ακτίνα 16 (διάμετρο 33).

**0)** Τρέξτε το deviceQuery() και καταγράψτε τα αποτελέσματα.

**1)** Μεταγλωττίστε τον κώδικα, προσθέτοντας αυτή τη φορά ως παράμετρο και το:

--ptxas-options -dlcm=cg

κατά τη μεταγλώττιση. Η παράμετρος αυτή δίνει την εντολή στον μεταγλωττιστή να παράγει κώδικα ο οποίος δε χρησιμοποιεί την L1 cache. Μετρήστε και καταγράψτε σε διάγραμμα την επίδοση για διαφορετικά μεγέθη εικόνας (δυνάμεις του 2), όταν είναι ενεργοποιημένα όλα τα επίπεδα της cache και όταν είναι απενεργοποιημένη η L1.

Ακολούθως, χρησιμοποιήστε τις παραμέτρους:

--ptxas-options -dlcm=cv –ptxas-options -dscm=wt

Η 1η παράμετρος απενεργοποιεί τη χρήση της L2 για reads, ενώ η δεύτερη ορίζει την πολιτική για της εγγραφές write through (άρα ουσιαστικά εξαφανίζει τα οφέλη της L2 για τα writes). Επαναλάβετε τις μετρήσεις και καταγράψτε τις στο ίδιο διάγραμμα. Εάν επαληθεύσετε μία φορά ότι ο κώδικάς σας λειτουργεί σωστά, δεν είναι ανάγκη να εκτελείτε το τμήμα του κώδικα το οποίο κάνει την συνέλιξη στη CPU και την επαλήθευση.

**2)** Υλοποιήστε 2 νέες συναρτήσεις __global__, οι οποίες πραγματοποιούν blocked (tiled) convolution κατά γραμμές και στήλες αντίστοιχα, αξιοποιώντας τη shared memory των streaming multiprocessors ως cache. Θεωρήστε ότι το μέγεθος του thread block είναι ίδιο με το μέγεθος του tile. Χρησιμοποιήστε την κατάλληλη γεωμετρία block/grid για το πρόβλημά σας. Η shared memory θα χρησιμοποιηθεί για τα δεδομένα εισόδου, ενώ για την αποθήκευση των συντελεστών του γραμμικού φίλτρου συνέλιξης χρησιμοποιήστε την constant memory.

**3)** Εκτελέστε τη μεταγλώττιση με παράμετρο:

`-ptxas-options=-v`

Η παράμετρος αυτή οδηγεί τον μεταγλωττιστή να σας ενημερώσει για την ποσότητα των πόρων (shared memory, constant memory, registers) που χρησιμοποιεί κάθε thread στον kernel σας. Καταγράψτε αυτές τις ποσότητες. Με βάση τα παραπάνω και τους περιορισμούς του device σας, πόσο είναι το μέγιστο μέγεθος γραμμικού thread block (και άρα και tile) που μπορείτε να υποστηρίξετε;

**4)** Εκτελέστε τον κώδικα για διαφορετικά μεγέθη εικόνας (δυνάμεις του 2). Πειραματιστείτε με διαφορετικά μεγέθη tiles. Σκεφτείτε εναλλακτικούς τρόπους για να χειριστείτε (όσον αφορά την ιεραρχία μνήμης) τα pixels στα άκρα της περιοχής που υπολογίζει κάθε thread block, κάνοντας το μέγεθος του tile να μην αντιστοιχεί στο μέγεθος του block αν αυτό σας βοηθάει να έχετε καλύτερη αξιοποίηση της shared memory. Σημειώστε τους χρόνους που παρατηρείτε σε κάθε υλοποίηση που δοκιμάζετε για ένα μεγάλο μέγεθος πίνακα.

Για την υλοποίηση με την υψηλότερη επίδοση, καταγράψτε τον χρόνο εκτέλεσης και συγκρίνετέ τον (σε ένα διάγραμμα) με τον αντίστοιχο της μη tiled υλοποίησης. Τι παρατηρείτε; Φροντίστε να ελέγξετε την ορθότητα των νέων kernel σας αντιπαραβάλλοντας τα αποτελέσματά τους με αυτά της CPU. Αφού βεβαιωθείτε για την ορθότητα δεν είναι ανάγκη να επανεκτελείτε τον κώδικα της CPU κάθε φορά.

*Σημείωση*: Αν χρειαστεί μπορείτε να εκτελέσετε τη συνάρτηση

`cudaFuncSetCacheConfig(<όνομα kernel>, cudaFuncCachePreferShared)`

ώστε να ορίσετε ότι ο kernel σας προτιμάει να χρησιμοποιοεί 48KB ως shared memory και 16KB ως L1 cache. Αν χρησιμοποιήσετε την παράμετρο `cudaFuncCachePreferL1` επιλέγετε το αντίστροφο (16KB ως shared memory και 48KB ως L1 cache).

**5)** Απαντήστε στις ακόλουθες ερωτήσεις για την blocked διδιάστατη συνέλιξη (και για τους 2 kernels):

- Με χρήση του βέλτιστου tile που καταφέρατε να χρησιμοποιήσετε, πόσες φορές διαβάζεται κάθε στοιχείο της εικόνας εισόδου και του φίλτρου κατά την εκτέλεση του kernel (π.χ. Για τον μεγαλύτερο πίνακα που καταφέρατε να εκτελέσετε); Πόσες από αυτές τις αναγνώσεις είναι από global memory. Πόσες από shared memory; Πόσες αναγνώσεις γίνονται από constant memory; Πώς επηρεάζει το μέγεθος του tile το λόγο αναγνώσεων από shared προς τις αναγνώσεις από global memory;
- Για το βέλτιστο tile, ποιός είναι ο λόγος προσπελάσεων μνήμης προς πράξεις κινητής υποδιαστολής; Θεωρήστε τους πολλαπλασιασμούς και τις προσθέσεις ως ξεχωριστές πράξεις. Μετρήστε μόνο τις προσπελάσεις στη global memory ως προσπελάσεις μνήμης. Αγνοήστε αριθμητικές πράξεις που αφορούν τον υπολογισμό διευθύνσεων στη μνήμη.
- Συγκρίνετε τις εκτιμήσεις σας με αυτές της 1ης άσκησης για την απλή συνέλιξη 2 διαστάσεων.

**6)** Επαναλάβετε το βήμα (1), αλλά αυτή τη φορά για τον tiled κώδικα. Τι παρατηρείτε για την επίδραση της cache σε σχέση με τον non-tiled κώδικα; Γιατί;

**7)** Χρησιμοποιήστε doubles αντί για floats. Τι μέγεθος γραμμικού block / tile μπορείτε πλέον να χρησιμοποιήσετε; Γιατί; Επαναλάβετε τις μετρήσεις σας για διαφορετικά μεγέθη εικόνας και φίλτρου και συγκρίνετέ τις με τις αντίστοιχες μετρήσεις για floats. Δεν χρειάζεται να πειραματιστείτε με την απενεργοποίηση της cache.

**Παράδοση**:
Πρέπει να παραδώσετε:

- Τον τελικό κώδικα, με τον πυρήνα της απλής συνέλιξης 2 διαστάσεων, τον πυρήνα της blocked συνέλιξης 2 διαστάσεων με floats και doubles.
- Αναφορά με τις απαντήσεις στα ερωτήματα (και το αποτέλεσμα του deviceQuery). Δημιουργήστε ένα αρχείο .tar.gz με τα παραπάνω περιεχόμενα και όνομα <όνομα>_<AEM>_lab3.tar.gz. Στείλτε το με mail στο CE421lab@gmail.com έως τα μεσάνυχτα της Κυριακής 17/4/2016 23:59.