ΛΟΓΙΣΜΙΚΟ & ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ ΥΨΗΛΗΣ ΕΠΙΔΟΣΗΣ

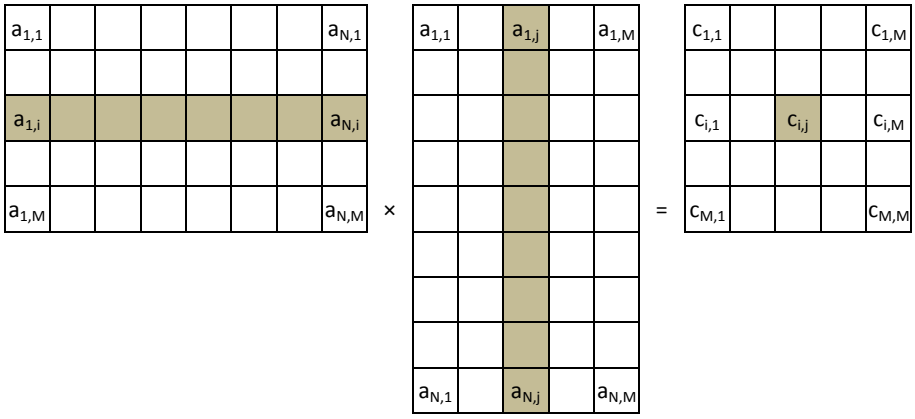
ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2016/2017

Εισαγωγή

Σε πολλές εφαρμογές είναι απαραίτητος ο υπολογισμός του γινομένου του ανάστροφου ενός μητρώου με το αρχικό μητρώο. Δηλαδή αν Α είναι το μητρώο, τότε πρέπει να υπολογιστεί το γινόμενο $A^T \cdot A$. Φυσικά ο υπολογισμός αυτός συνήθως αποτελεί τμήμα μόνο του συνολικού προγράμματος. Ωστόσο, μπορεί να απαιτεί αρκετά σημαντικό ποσοστό από τον συνολικό χρόνο εκτέλεσης της εφαρμογής. Κατά συνέπεια μια αποδοτική υλοποίηση είναι απαραίτητη. Στα πλαίσια της εργασίας αυτής ζητείται να υλοποιήσετε και να μετρήσετε την απόδοση της προαναφερθείσας πράξης στο προγραμματιστικό μοντέλο CUDA. **Για λόγους εξοικονόμησης μνήμης, θα πρέπει να υπάρχει αποθηκευμένο στην μνήμη μόνο το μητρώο Α.**

Ο πολλαπλασιασμός $A^T \cdot A$

Ο αλγόριθμος για τον πολλαπλασιασμό του ανάστροφου ενός μητρώου με το αρχικό μητρώο είναι απλός και αποτελεί ειδική περίπτωση του πολλαπλασιασμού δύο μητρώων. Θεωρώντας ένα μητρώο A διαστάσεων $N \times M$, το μητρώο A^T θα έχει διαστάσεις $M \times N$ και το αποτέλεσμα του πολλαπλασιασμού θα είναι ένα διάνυσμα C διαστάσεων $M \times M$. Σχηματικά αυτό φαίνεται παρακάτω:



Το στοιχείο $C_{i,j}$ υπολογίζεται από τον πολλαπλασιασμό της γραμμής i του μητρώου A^T (που ταυτόχρονα είναι η στήλη i του μητρώου A) με την στήλη j του μητρώου A:

$$C_{i,j} = \sum_{k=1}^{N} a_{i,k}^T \cdot a_{k,j} = \sum_{k=1}^{N} a_{k,i} \cdot a_{k,j}$$

Η διαδικασία αυτή επαναλαμβάνεται για όλες τις γραμμές του μητρώου A^T, δηλαδή για $1 \leq i \leq M$ και όλες τις στήλες του μητρώου A, δηλαδή για $1 \leq j \leq M$.

Η βιβλιοθήκη cuBLAS

Η χρήση πράξεων μεταξύ διανυσμάτων, μεταξύ μητρώων και διανυσμάτων και μεταξύ μητρώων είναι συνήθεις σε πολλά προγράμματα. Για τον λόγο αυτό η βιβλιοθήκη BLAS ορίζει ένα σύνολο πράξεων σε μορφή συναρτήσεων, οι οποίες υλοποιούνται στην συνέχεια με αποδοτικό τρόπο για κάθε σύστημα στο οποίο θέλουμε να χρησιμοποιήσουμε αντίστοιχες πράξεις. Περισσότερες πληροφορίες μπορείτε να βρείτε στην ιστοσελίδα http://www.netlib.org/blas. Σε πολλά συστήματα υπάρχουν εξαιρετικά βελτιστοποιημένες υλοποιήσεις των συναρτήσεων αυτών. Για παράδειγμα, η Intel προσφέρει την "Math Kernel Library (MKL)" (https://software.intel.com/en-us/intel-mkl) που περιλαμβάνει βελτιστοποιημένες υλοποιήσεις των συναρτήσεων BLAS για τους επεξεργαστές της.

Η ονοματολογία που ακολουθείται στην παραπάνω βιβλιοθήκη για το είδος των πράξεων αλλά και τις ίδιες τις πράξεις έχει συνοπτικά ως εξής. Πράξεις "Level-1" είναι αυτές μεταξύ διανυσμάτων. Πράξεις "Level-2" είναι μεταξύ μητρώων και διανυσμάτων. Τέλος, πράξεις "Level-3" είναι μεταξύ μητρώων. Το πρώτο γράμμα της συνάρτησης δηλώνει τον τύπο δεδομένων των διανυσμάτων και μητρώων. Για single precision χρησιμοποιείται το "s", για double precision το "d", για complex το "c" και για double complex το "z". Όταν εμπλέκονται μητρώα, τότε χρησιμοποιούνται επιπλέον δύο γράμματα για να υποδηλώσουν το είδος του μητρώου. Για γενικά, πυκνά μητρώα χρησιμοποιούνται τα "ge" (general), για συμμετρικά μητρώα τα "sy" (symmetric), για Hermitian μητρώα τα "he" (hermitian), για τριγωνικά μητρώα τα "tr" (triangular), κλπ. Τέλος, χρησιμοποιούνται ένα ή δύο ακόμα γράμματα και αριθμοί για να δηλώσουν το είδος της πράξης. Για παράδειγμα το "mm" υποδηλώνει πράξεις μεταξύ μητρώων (matrix-matrix), το "mv" μεταξύ μητρώου και διανύσματος (matrix-vector), κλπ. Έτσι η πράξη "CSYRK" υποδηλώνει ένα "rank-k update" σε ένα συμμετρικό μητρώο από μιγαδικούς αριθμούς.

Η υλοποίηση των πράξεων που ορίζονται στην BLAS για το προγραμματιστικό μοντέλο CUDA οδήγησε στην δημιουργία της βιβλιοθήκης cuBLAS. Μπορείτε να βρείτε μια περιγραφή της βιβλιοθήκης και όλων των συναρτήσεων που υποστηρίζει στην ιστοσελίδα http://docs.nvidia.com/cuda/cublas. Η βιβλιοθήκη αυτή εγκαθίσταται κατά την εγκατάσταση του πακέτου της CUDA. Μεταξύ των συναρτήσεων αυτών υπάρχει και η "cublasDgemm", η οποία μας αφορά στα πλαίσια της εργασίας. Πιο συγκεκριμένα, υποδηλώνει την γενική περίπτωση του πολλαπλασιασμού δύο γενικών, πυκνών μητρώων, όπου όλα τα στοιχεία των παραπάνω είναι τύπου "double". Η πράξη που υλοποιείται είναι η:

$$C = \alpha \cdot op(A) \cdot op(B) + \beta \cdot C$$

Στην πραγματικότητα λοιπόν πρόκειται για ανανέωση του μητρώου C με το αποτέλεσμα του πολλαπλασιασμού του μητρώου A με το μητρώο B. Τα α, β είναι βαθμωτοί και η πράξη $op()$ υποδηλώνει αν το μητρώο θα χρησιμοποιηθεί ως έχει ή αν θα χρησιμοποιηθεί ο ανάστροφος του. Για να επιτύχουμε λοιπόν την πράξη που ορίσαμε νωρίτερα θα πρέπει να θέσουμε $\alpha = 1,\ \beta = 0,\ op(A) = \text{CUBLAS\_OP\_N}$ και $op(B) = \text{CUBLAS\_OP\_T}$. Διαβάστε προσεκτικά πως θεωρεί η βιβλιοθήκη cuBLAS ότι είναι αποθηκευμένα τα μητρώα στην μνήμη για να καταλάβετε γιατί ορίστηκαν με τον παραπάνω τρόπο οι πράξεις $op()$!

Ζητούμενα της άσκησης

Στα πλαίσια της άσκησης σας ζητείται να υλοποιήσετε και να πάρετε μετρήσεις απόδοσης για 3 διαφορετικές υλοποιήσεις της πράξης του πολλαπλασιασμού ανάστροφου μητρώου με το αρχικό μητρώο. **Οι υλοποιήσεις θα πρέπει να είναι γενικές, υπό την έννοια ότι θα πρέπει να υποστηρίζεται οποιοδήποτε μέγεθος μητρώου (και όχι, π.χ., μόνο τετραγωνικά μητρώα ή μεγέθη που είναι δυνάμεις του 2). Επιπλέον, για λόγους εξοικονόμησης μνήμης, θα πρέπει να υπάρχει αποθηκευμένο στην μνήμη μόνο το μητρώο Α.**Τα στοιχεία των μητρώων θα είναι τύπου "double". Πιο συγκεκριμένα:

1) Χρησιμοποιείστε την συνάρτηση "cublasDgemm" της βιβλιοθήκης cuBLAS και μετρήστε για διάφορα μεγέθη μητρώων τον χρόνο εκτέλεσης της πράξης. Προσπαθήστε να φτάσετε σε όσο μεγαλύτερα μεγέθη μητρώων μπορείτε. **Κατά την χρονομέτρηση μην λαμβάνετε υπ' όψη σας τους χρόνους μεταφοράς δεδομένων από και προς την κάρτα γραφικών! Λάβετε υπ' όψη μόνο τον χρόνο υπολογισμού.**
2) Υλοποιήστε σε CUDA τον πιο απλό τρόπο πραγματοποίησης της πράξης στην κάρτα γραφικών: Δεσμεύστε καθολική μνήμη (global memory) για το μητρώο, μεταφέρετε τα δεδομένα και κάνετε τις πράξεις. Χρονομετρήστε όπως και πριν τον χρόνο υπολογισμού για τα ίδια μεγέθη μητρώων.
3) Προσπαθήστε να βελτιστοποιήσετε την απόδοση της δικής σας συνάρτησης, αξιοποιώντας την ιεραρχία μνήμης της CUDA. Χρησιμοποιείστε καταχωρητές, κοινή μνήμη (shared memory), streams, εσωτερική αναδιοργάνωση των δεδομένων στην κάρτα γραφικών και ότι άλλο θεωρείτε ότι μπορεί να σας δώσει καλύτερη απόδοση! **Θα πρέπει οπωσδήποτε να λάβετε υπόψη σας το γεγονός πως προσπελαύνετε στοιχεία του ίδιου μητρώου, τόσο κατά στήλες (A^T) αλλά ταυτόχρονα και κατά γραμμές (A). Το γεγονός αυτό θα πρέπει να ληφθεί υπόψη τόσο κατά την εκμετάλλευση της κοινής μνήμης, όσο και για την αξιοποίηση της προσπέλασης συνεχόμενων στοιχείων στην καθολική μνήμη (coalesced accesses in main memory).** Οποιοδήποτε υλικό βρείτε (βιβλία, GPU Gems, δημοσιεύσεις σε επιστημονικά περιοδικά ή συνέδρια, κλπ) μπορείτε φυσικά να το χρησιμοποιήσετε.

Είναι προφανές πως το σημαντικότερο κομμάτι της άσκησης είναι το τελευταίο στάδιο. Αυτό που ζητείται είναι να δείξετε ότι έχετε κατανοήσει την αρχιτεκτονική των καρτών γραφικών και πως μπορείτε να απεικονίσετε αποδοτικά έναν αλγόριθμο στην αρχιτεκτονική αυτή, λαμβάνοντας υπ' όψη σας όλες τις παραμέτρους του αλγόριθμου και της αρχιτεκτονικής. Μας ενδιαφέρει η απόδοση της λύσης σας!

Διαδικαστικά

Η εργασία θα πρέπει να γίνει σε ομάδες των 2 ή 3 ατόμων. Η διαχείριση των ομάδων θα γίνει μέσω της ηλεκτρονικής πλατφόρμας "Open eClass" του Πανεπιστημίου Πατρών (http://eclass.upatras.gr). Για τον σκοπό αυτό θα πρέπει όλοι οι φοιτητές που επιθυμούν να παραδώσουν εργασία να εγγραφούν πρώτα στην παραπάνω πλατφόρμα. Στην συνέχεια, ένα άτομο από κάθε ομάδα θα αναλάβει να δηλώσει την ομάδα του μέχρι την **Τρίτη, 01/11/2016 και ώρα 23:59:59**. Το άτομο αυτό θα είναι επίσης υπεύθυνο για όλη την επικοινωνία της ομάδας μαζί μας, καθ' όλη την διάρκεια του εξαμήνου και μέχρι την παράδοση της άσκησης. Η ομάδα θα δηλωθεί μέσω e-mail στην διεύθυνση venetis@ceid.upatras.gr. Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, το e-mail θα πρέπει να έχει τον εξής τίτλο (subject):

`[HPC16-17] Δήλωση ομάδας`

Το περιεχόμενο του e-mail θα πρέπει να είναι ο Α.Μ. και το ονοματεπώνυμο του φοιτητή που κάνει την δήλωση της ομάδας. Στην συνέχεια θα αναλάβουμε να φτιάξουμε μια ομάδα στο "Open eClass" και θα σας ενημερώσουμε για τον αριθμό της ομάδας σας.

Σε περίπτωση που χρειαστεί επιπλέον επικοινωνία μαζί μας μέσω e-mail, αυτή θα πρέπει να γίνει με τον κ. Ιωάννη Βενέτη (venetis@ceid.upatras.gr). **Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, ο τίτλος (subject) κάθε e-mail θα πρέπει να ξεκινάει με [HPC16-17].**

Παραδοτέα

Τα παραδοτέα για την εργασία σας είναι μια γραπτή αναφορά και ο κώδικας της άσκησης που θα αναπτύξετε. **Η προθεσμία παράδοσης της εργασίας ορίζεται η Κυριακή 15/01/2017 και ώρα 23:59:59. Η εργασία θα πρέπει να παραδωθεί αποκλειστικά μέσω της ηλεκτρονικής πλατφόρμας "Open eClass" (εργασίες που θα αποσταλούν μέσω e-mail δεν θα βαθμολογηθούν).** Μετά την είσοδο σας στο σύστημα θα πρέπει να μεταβείτε στο μάθημα "Λογισμικό & Προγραμματισμός Συστημάτων Υψηλής Επίδοσης " και στο μενού αριστερά να μεταβείτε στο "Εργασίες". **Κάθε ομάδα θα παραδώσει μια φορά μόνο την εργασία (όχι κάθε φοιτητής ξεχωριστά). Βεβαιωθείτε πως στο εξώφυλλο της γραπτής αναφοράς αναφέρονται τα ονόματα και οι ΑΜ όλων των συμμετεχόντων της ομάδας.**

Στην αναφορά επικεντρωθείτε στην επεξήγηση της παραλληλοποίησης που κάνατε, πως αξιοποιήσατε τις δυνατότητες της CUDA, πως απεικονίσατε τον αλγόριθμο στην αρχιτεκτονική σας, στις μετρήσεις σας και στα διαγράμματα που θα προσθέσετε.

Παράρτημα Α

Για να εγκαταστήσετε την CUDA στο σύστημα σας μεταβείτε στην ιστοσελίδα https://developer.nvidia.com/cuda-zone και επιλέξτε τον σύνδεσμο “Downloads”. Επιλέξτε το πακέτο που θα κατεβάσετε ανάλογα με το σύστημα σας.

Σε περιβάλλον Windows, αν και δεν είναι απαραίτητο, η ύπαρξη του Visual Studio βοηθάει ιδιαίτερα στην ανάπτυξη εφαρμογών. Κατά την εγκατάσταση της CUDA εγκαθίσταται μια επέκταση για το Visual Studio ειδικά για την ανάπτυξη εφαρμογών CUDA. Αν δεν υπάρχει το Visual Studio η ανάπτυξη προγραμμάτων μπορεί να γίνει σε οποιονδήποτε κειμενογράφο (editor) και να χρησιμοποιείται απευθείας ο μεταγλωττιστής της CUDA (nvcc) από την “Γραμμή Εντολών” (“Command Prompt”). Αντίστοιχα, σε περιβάλλον Linux ο μεταγλωττιστής καλείται από το κέλυφος (shell).

Αν η ομάδα σας δεν διαθέτει σύστημα με κάρτα γραφικών της NVidia (ή αυτή δεν υποστηρίζεται από την CUDA) επικοινωνήστε μαζί μας για να σας δώσουμε πρόσβαση σε δικό μας σύστημα. Θα μπορείτε να συνδέεστε με απομακρυσμένη πρόσβαση σε αυτό (ssh).

Σημαντική παρατήρηση: Το σύστημα στο οποίο θα σας δωθεί πρόσβαση έχει δύο κάρτες γραφικών που υποστηρίζουν CUDA. Μόνο η κάρτα με Device ID 1 έχει δυνατότητα επεξεργασίας αριθμών κινητής υποδιαστολής διπλής ακρίβειας (double). Δείτε την συνάρτηση cudaSetDevice() για να θέσετε ποια κάρτα γραφικών θα χρησιμοποιηθεί για την εκτέλεση των υπολογιστικών πυρήνων της εφαρμογής σας.