

ΗΥ404: ΠΑΡΑΛΛΗΛΟΣ ΚΑΙ ΔΙΚΤΥΑΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

Χειμερινό Εξάμηνο 2014-2015

Προγραμματιστική Εργασία 3:

Υλοποίηση σε MPI περιβάλλον

(Ημερομηνία Παράδοσης: Κυριακή 8/2/2015, (Ωρα: 23:55))

Γενική σημείωση:

Παραδώστε μια φορά την εργασία σας (κείμενο με τις λύσεις/αποτελέσματα και σχόλια σας), μέσω email σε ένα zip (ergasia2_AEM1_AEM2_AEM3.zip) αρχείο.

Σε πολλά πραγματικά προβλήματα οι πίνακες που προκύπτουν είναι αραιοί, δηλαδή έχουν πολλά μηδενικά στοιχεία. Αν η διάσταση του πίνακα είναι N , οι πυκνοί πίνακες έχουν N^2 μη μηδενικά στοιχεία, ενώ οι αραιοί συνήθως έχουν $O(N)$ μη μηδενικά στοιχεία NZ . Για παράδειγμα ο πίνακας A , όπου

$$A = \begin{bmatrix} 11 & 0 & 0 & 14 & 0 & 0 \\ 0 & 22 & 0 & 0 & 0 & 26 \\ 31 & 0 & 33 & 0 & 0 & 0 \\ 41 & 0 & 0 & 0 & 45 & 0 \\ 0 & 0 & 53 & 0 & 55 & 0 \\ 0 & 0 & 0 & 64 & 0 & 66 \end{bmatrix} \text{ είναι αραιός.}$$

Οι πίνακες αυτοί αποθηκεύονται σε συμπιεσμένη μορφή $[1]$ για εξοικονόμηση μνήμης και μείωση υπολογιστικού κόστους. Η πληροφορία που καταγράφεται είναι οι μη μηδενικές τιμές του πίνακα και οι θέσεις τους. Σε κάθε περίπτωση μας δίδεται το N και το NZ .

Αραιοί πίνακες σε Coordinate Format

Ένας δεύτερος τρόπος αποθήκευσης αραιών πινάκων είναι το **Coordinate Format (COO)** κατά γραμμές στο οποίο αποθηκεύονται σε τρία διανύσματα a_values , i_index , j_index οι τιμές των στοιχείων, η γραμμή και η στήλη στην οποία βρίσκονται ως εξής:

το στοιχείο $a_values(k)$ βρίσκεται στην $i_index(k)$ γραμμή και $j_index(k)$ στήλη του πίνακα. Για τον παραπάνω πίνακα τα διανύσματα αυτά είναι τα :

$a_values = [11 \ 14 \ 22 \ 26 \ 31 \ 33 \ 41 \ 45 \ 53 \ 55 \ 64 \ 66]$

$i_index = [1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 4 \ 4 \ 5 \ 5 \ 6 \ 6]$

$j_index = [1 \ 4 \ 2 \ 6 \ 1 \ 3 \ 1 \ 5 \ 3 \ 5 \ 4 \ 6]$

Τα στοιχεία του A , αποθηκεύονται κατά γραμμές μέσα στα 3 διανύσματα. Τα στοιχεία της κάθε γραμμής αποθηκεύονται με αύξουσα σειρά κατά στήλη. Αντίστοιχα υπάρχει το **Coordinate Format** κατά στήλες.

Αραιοί πίνακες σε Compressed Sparse Row Format

Ένας δεύτερος τρόπος αποθήκευσης αραιών πινάκων είναι το **Compressed Sparse Row Format (CSR)** στο οποίο αποθηκεύονται σε τρία διανύσματα a_values , i_ptr , j_index οι τιμές των στοιχείων, η γραμμή και η στήλη στην οποία βρίσκονται ως εξής:

το στοιχείο $a_values(k)$ βρίσκεται στην $j_index(k)$ στήλη του πίνακα. Το διάνυσμα i_ptr περιέχει τις θέσεις στα διανύσματα a_values και j_index που ξεκινά η κάθε γραμμή του A . Έτσι τα στοιχεία της γραμμής i του πίνακα A βρίσκονται στις θέσεις $i_ptr(i)$ έως $i_ptr(i+1)-1$

Για τον παραπάνω πίνακα τα διανύσματα αυτά είναι τα :

$a_values = [11 \ 14 \ 22 \ 26 \ 31 \ 33 \ 41 \ 45 \ 53 \ 55 \ 64 \ 66]$

$i_ptr = [1 \ 3 \ 5 \ 7 \ 9 \ 11 \ 13]$

$j_index = [1 \ 4 \ 2 \ 6 \ 1 \ 3 \ 1 \ 5 \ 3 \ 5 \ 4 \ 6]$

Τα στοιχεία του A , αποθηκεύονται κατά γραμμές μέσα στα 3 διανύσματα. Τα στοιχεία της κάθε γραμμής αποθηκεύονται με αύξουσα σειρά κατά στήλη. Αντίστοιχα υπάρχει το **Compressed Sparse Column Format**.

Υπάρχουν κι άλλα formats για αποθήκευση αραιών πινάκων, που προκύπτουν από τα προβλήματα που δημιουργούν τους πίνακες αλλά και τις πράξεις/συναρτήσεις που πρέπει να εφαρμόσουμε σε αυτούς (π.χ., banded formats, diagonal formats κλπ.)

Για τη αποθήκευση τέτοιων πινάκων στον υπολογιστή υπάρχουν γνωστά file formats [2,3,4] όπως Matrix Market Exchange Format (που υποστηρίζει το COO format), Harwell-Boeing Exchange Format (που υποστηρίζει το CSR format) κλπ., και βιβλιοθήκες που βοηθούν στο I/O μέρος του προγραμματισμού των αλγορίθμων για αραιούς πίνακες.

Άσκηση 1

(40) Υλοποιήστε το σειριακό αλγόριθμο (που μελετήσατε στο σετ ασκήσεων 3, άσκηση 1) για πολλαπλασιασμό αραιού πίνακα (αποθηκευμένο σε **COO**) με διάνυσμα.

Άσκηση 2

α) (70) Υλοποιήστε τον αντίστοιχο παράλληλο αλγόριθμο (που μελετήσατε στο σετ ασκήσεων 3, άσκηση 2α) για N στοιχεία σε p παράλληλες εργασίες. Επιχειρήστε διαχωρισμό του προβλήματος κατά block γραμμών (N/p) ανά επεξεργαστή. Μετρήστε .

β) (30) Φτιάξτε πίνακες όπως τον Πίνακα 1 και σημειώστε τους χρόνους για σειριακή και παράλληλη εκτέλεση και μέσα σε παρένθεση το speedup που είχατε σε κάθε περίπτωση. Κάντε γραφική παράσταση για το «πλήθος των threads vs καλύτερο speedup». Τι παρατηρείτε; Συμφωνούν τα αποτελέσματά σας με τη θεωρητική ανάλυση του σετ ασκήσεων 3;

| NZ vs p | 1 | 2 | 4 | 5 | 8 | 10 | 16 | 20 | 32 |
|---------|-----|-----------|---|---|---|----|----|----|----|
| | 105 | 60 (1,75) | | | | | | | |
| .. | | | | | | | | | |
| | | | | | | | | | |

Πίνακας 1. Χρόνοι εκτέλεσης για διαφορετικό πλήθος στοιχείων και αριθμό επεξεργαστών.

Άσκηση 3

α) (70) Υλοποιήστε τον αντίστοιχο παράλληλο αλγόριθμο (που μελετήσατε στο σετ ασκήσεων 3, άσκηση 3) για N στοιχεία σε p παράλληλες εργασίες. Επιχειρήστε διαχωρισμό του προβλήματος κατά block γραμμών έτσι ώστε να υπάρχουν NZ/p περίπου στοιχεία ανά επεξεργαστή. Υπολογίστε το κόστος του σε συνάρτηση των N , NZ και p .

β) (30) Φτιάξτε πίνακες όπως τον Πίνακα 1 και σημειώστε τους χρόνους για σειριακή και παράλληλη εκτέλεση και μέσα σε παρένθεση το speedup που είχατε σε κάθε περίπτωση. Κάντε γραφική παράσταση για το «πλήθος των threads vs καλύτερο speedup». Τι παρατηρείτε; Συμφωνούν τα αποτελέσματά σας με τη θεωρητική ανάλυση του σετ ασκήσεων 3;

Οδηγίες για MPI στο PC/laptop:

Δουλέψτε στο μηχάνημα σας για compile/debug και πρώτες εκτελέσεις. Χρησιμοποιείτε το cluster για να τρέξετε τα προγράμματα σας και να δείτε πως κλιμακώνουν οι αλγόριθμοι.

Οδηγίες για είσοδο και εργασία στο cluster:

Το cluster βασίζεται στην διανομή Rocks Cluster (<http://www.rocksclusters.org>), διαθέτει τέσσερα computing nodes, ένα Frontend node για την διαχείρισή του και ένα Login node για πρόσβαση.

Κάθε computing node διαθέτει οχτώ επεξεργαστές και 16GB RAM.

Κάθε ομάδα έχει ένα λογαριασμό στην διάθεσή της. Όταν λάβετε το στοιχεία των λογαριασμών θα πρέπει άμεσα να αλλάξετε το password.

Η αποστολή και διαχείριση των εργασιών (jobs) γίνεται με το Sun Grid Engine (SGE).
Οδηγίες για την αποστολή/διαχείριση εργασιών (jobs) με το SGE θα βρείτε στην σελίδα
<http://www.rocksclusters.org/roll-documentation/sge/4.2.1/submitting-batch-jobs.html>

Επισυνάπτεται ενδεικτικό script για την αποστολή εργασίας (testrun.sh)

αποστολή εργασίας με qsub

```
#qsub -pe orte 32 testrun.sh
```

-pe orte -> OpenMPI parallel environment

32 -> αριθμός επεξεργαστών (μεταβάλλοντας τον αριθμό, μεταβάλλεται ο αριθμός των επεξεργαστών που θα χρησιμοποιηθούν)

testrun.sh -> το επισυναπτόμενο script

Η εντολή qstat εμφανίζει τις εργασίες (jobs) και τις ουρές (queues) του SGE.

```
#qstat -f
```

Η εντολή qdel απομακρύνει εργασίες από το SGE.

```
#qdel 15
```

15 -> αριθμός (id) εργασίας

Ενδεικτική εντολή για την μεταγλώττιση του mpi κώδικα

```
# mpicc test.c -> παράγει εκτελέσιμο με όνομα a.out
```

Σημείωση: Στις παρακάτω πηγές θα βρείτε περισσότερες πληροφορίες για τα sparse formats, κώδικες και σχετικούς αλγορίθμους και πίνακες για να τρέξετε τα προγράμματα σας. Τα pdf των αρχείων στα 4. και 5. επισυνάπτονται για πιο εύκολη ανάγνωση.

Πηγές:

1. Sparse Matrix Storage Formats, J. Dongarra, <http://web.eecs.utk.edu/~dongarra/etemplates/node372.html>
2. Matrix Market, <http://math.nist.gov/MatrixMarket/>
3. File formats, <http://math.nist.gov/MatrixMarket/formats.html>
4. ANSI C library for Matrix Market I/O, <http://math.nist.gov/MatrixMarket/mmio-c.html>
5. Sparse Matrix-Vector Multiplication with the Message Passing Interface, A. Edelman, http://web.mit.edu/18.337/WWW/hw5_96.ps
6. Parallel sparse matrix-vector multiplication, F. Tavakoli, <ftp://ftp.nada.kth.se/Num/publications/other/tavakoli-9704.ps.Z>