Πειραματικά Αποτελέσματα

Τεχνικά Χαρακτηριστικά Συστήματος

Για τις μετρήσεις χρησιμοποιήθηκε το ακόλουθο σύστημα:

Λογισμικό	
Operating System	Linux Ubuntu
Version	16.0.4
Kernel	4.17.11
Julia	0.6.4
PAPI	5.6.0

Υλικό	
CPU	Intel i7 2600K
Πυρήνες	4
Νήματα	8
Συχνότητα	3.4GHz (Turbo Boost enabled)
Instruction Sets	SSE4.2, AVX
L1 Data Cache	8 way 32KB
L1 Instruction Cache	8 way 32KB
L2 Cache	8 way 256KB
L3 Cache	16 way 8MB
Cache line	64 Bytes
RAM	12GB DDR3
Συχνότητα RAM	1600MHz

Στόχος πειράματος

Η πειραματική μέθοδος που ακολουθήθηκε, αποσκοπεί στη συγκέντρωση δεδομένων επίδοσης του επεξεργαστή, ώστε να παρουσιαστεί η δυνατότητα της χρήσης αυτής της μεθόδου ως benchmark.

Ο κύριος σκοπός είναι να συγκριθεί η συγκεκριμένη μέθοδος, με υπάρχουσες μεθόδους benchmarking που χρησιμοποιούν διαφορετικούς αλγορίθμους. Θα παρουσιαστούν τα πλεονεκτήματα και μειονεκτήματα αυτής της μεθόδου, σε σχέση με τις υπάρχουσες, και θα αναλυθεί η καταλληλόλητά της ως ένα αντικειμενικότερο κριτήριο επίδοσης.

Επιπροσθέτως, είναι θεμιτή η ανάλυση της συμπεριφοράς του συστήματος, σχετικά με τα διαφορετικά είδη και μεγέθη μητρώων. Συνεπώς, θα παρουσιαστεί μια συγκριτική ανάλυση επιδόσεων, αναλογικά με το είδος του μητρώου, το μέγεθός του, καθώς και το πόσο πυκνό είναι (το ποσοστό μη μηδενικών στοιχείων του).

Ένας ακόμη στόχος, είναι η μελέτη της αύξησης των επιδόσεων με την εκμετάλλευση παραλληλίας. Η παραλληλία είναι ένα βασικό χαρακτηριστικό της προόδου των σύγχρονων επεξεργαστών, συνεπώς είναι κρίσιμο το να συμπεριληφθεί σε ένα benchmark επιδόσεων. Όπως θα αναλυθεί, το πρόβλημα το οποίο θα χρησιμοποιήσουμε για το πείραμα, προσφέρεται για παραλληλία. Τέλος, έχει αξία να συγκριθούν διαφορετικές υλοποιήσεις παραλληλίας.

Διαδικασία μετρήσεων

Για τη λήψη των μετρήσεων, ακολουθήθηκε η εξής μέθοδος, όπως θα επαληθευθεί στον κώδικα που ακολουθεί:

1. Ορίστηκαν και φορτώθηκαν στη RAM όλα τα μητρώα που θα χρειαστούν στις μετρήσεις.
2. Εκκινήθηκαν οι απαραίτητοι μετρητές πράξεων της βιβλιοθήκης PAPI.
3. Ο κώδικας προς μέτρηση, εκτελέστηκε μία φορά ώστε να προθερμανθούν όλα τα σχετικά υποσυστήματα σε επίπεδο λογισμικού και υλικού (μεταγλώττιση των συναρτήσεων και cache αντίστοιχα)
4. Εκτελέστηκε η μέτρηση των πράξεων, με δέκα επαναλήψεις
5. Έπειτα, για δέκα άλλες επαναλήψεις χρονομετρήθηκε η διάρκεια ολοκλήρωσής τους
6. Οι χρόνοι αυτοί καταγράφηκαν, και υπολογίστηκε ο μέσος όρος τους.
7. Από τους δύο αυτούς μέσους όρους (πλήθος πράξεων και χρόνου εκτέλεσης), παράχθηκε ο αριθμός των FLOP/s.

Αιτιολόγηση

Ο κύριος στόχος ήταν η ελαχιστοποίηση των εξωτερικών παραγόντων που μπορεί να επηρεάσουν τις μετρήσεις. Γι' αυτό το λόγο, περιορίστηκαν / τερματίστηκαν, στο μέγιστο δυνατόν, όλες οι υπόλοιπες διεργασίες στο σύστημα. Επιπλέον, όλα τα δεδομένα που θα χρειάζονταν για τις μετρήσεις, φορτώθηκαν στην κύρια μνήμη εκ των προτέρων.

Ο κώδικας Julia είναι JIT (Just In Time) compiled, δηλαδή μεταγλωττίζεται τη στιγμή που χρειάζεται να εκτελεστεί. Επειδή η κλήση και εκτέλεση του μεταγλωττιστή αποτελεί μεγάλο κόστος, το οποίο δεν συσχετίζεται με τη μέτρηση εκτέλεσης, αλλά την ίδια τη γλώσσα, φροντίζουμε να μεταγλωττίσουμε τον κώδικα προς εξέταση πριν τη διαδικασία μέτρησης. Αυτό επιτυγχάνεται, κάνοντας χρήση τόσο της συνάρτησης `precompile()` της Julia, καθώς και με το να κληθεί κανονικά η συνάρτηση που θέλουμε να μετρήσουμε, με τα κανονικά της ορίσματα, απλά χωρίς να μετρήσουμε τίποτα σχετικό με την εκτέλεση αυτή. Αυτή η διαδικασία, είναι

συνήθως γνωστή ως **warmup** (προθέρμανση) στη βιβλιογραφία της μέτρησης απόδοσης.

Ένα συνακόλουθο της κλήσης της συνάρτησης πριν χρονομετρηθεί, είναι πως είναι εξαιρετικά πιθανό, να βρίσκονται σε κάποιο επίπεδο της cache τα δεδομένα που θα χρειαστούν στον υπολογισμό. Τα προαναφερθέντα, προϋποθέτουν ότι το μητρώο/τα δεδομένα μας, χωράνε εξ ολοκλήρου στην κρυφή μνήμη. Σε περίπτωση που είναι μεγαλύτερα από το μέγιστο επίπεδο κρυφής, το τελευταίο τμήμα των δεδομένων θα γραφτεί στη θέση του πρώτου, άρα κατά τη διάρκεια των μετρήσεων ξεκινάμε με άδεια cache (από χρήσιμα δεδομένα). Επιπροσθέτως, το υποσύστημα branch prediction (πρόβλεψης διακλάδωσης) της κεντρικής μονάδας επεξεργασίας, μιας και έχει συναντήσει τον κώδικα που πρόκειται να εκτελεστεί, έχει ορθά προετοιμάσει τον αλγόριθμο που θα χρησιμοποιήσει για την πρόβλεψη. Όλα αυτά, δημιουργούν τις πλέον ιδανικές συνθήκες μέτρησης χρόνου εκτέλεσης, κάτι το οποίο είναι θεμιτό ώστε να αντιπροσωπεύει μετρήσεις στις οποίες ο χρόνος εκτέλεσής τους, είναι τάξης μεγέθους μεγαλύτερος από αυτόν της αρχικοποίησης και των σταθερών (O(1)) διαδικασιών.

Δεδομένα

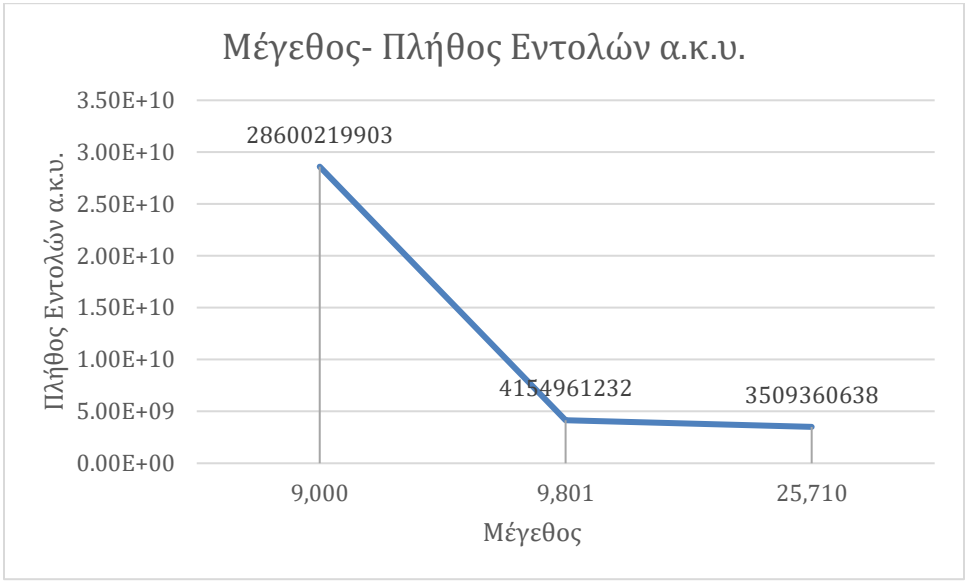
Τα μητρώα που θα χρησιμοποιηθούν στα πειράματα, προέρχονται από τη Suite Sparse συλλογή μητρώων (γνωστή και ως συλλογή αραιών μητρώων του πανεπιστημίου της Florida). Για τη διεπαφή με τον ιστότοπο της Suite Sparse, χρησιμοποιήθηκε το πακέτο MatrixDepot της Julia, το οποίο επιτρέπει το κατέβασμα και χρήση στη Julia οποιουδήποτε μητρώου στην βιβλιοθήκη. Η χρήση αυτού του πακέτου δεν είναι απαραίτητη, μιας και τα μητρώα παρέχονται στην ιστοσελίδα και σε μορφή `.mat` συμβατή τόσο με Matlab όσο και με Julia, ωστόσο είναι ευκολότερο να εκμεταλλευτούμε αυτή τη διεπαφή.

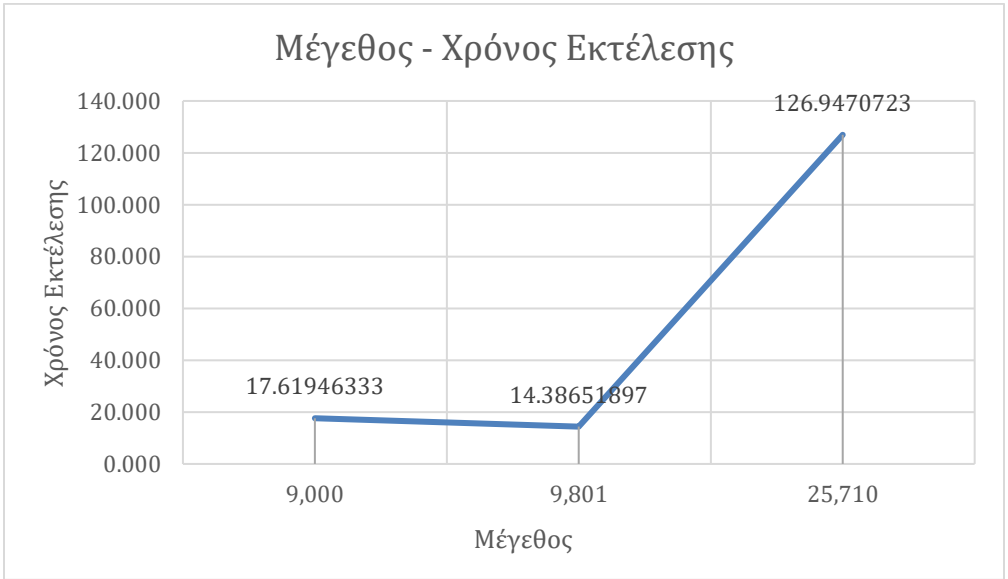
Τα μητρώα που θα χρησιμοποιηθούν για μετρήσεις, θα είναι όλα συμμετρικά θετικά ορισμένα (Symmmetric Positive Definite). Τα χαρακτηριστικά τα οποία θέλουμε να μελετήσουμε ως μεταβλητές είναι τα ακόλουθα:

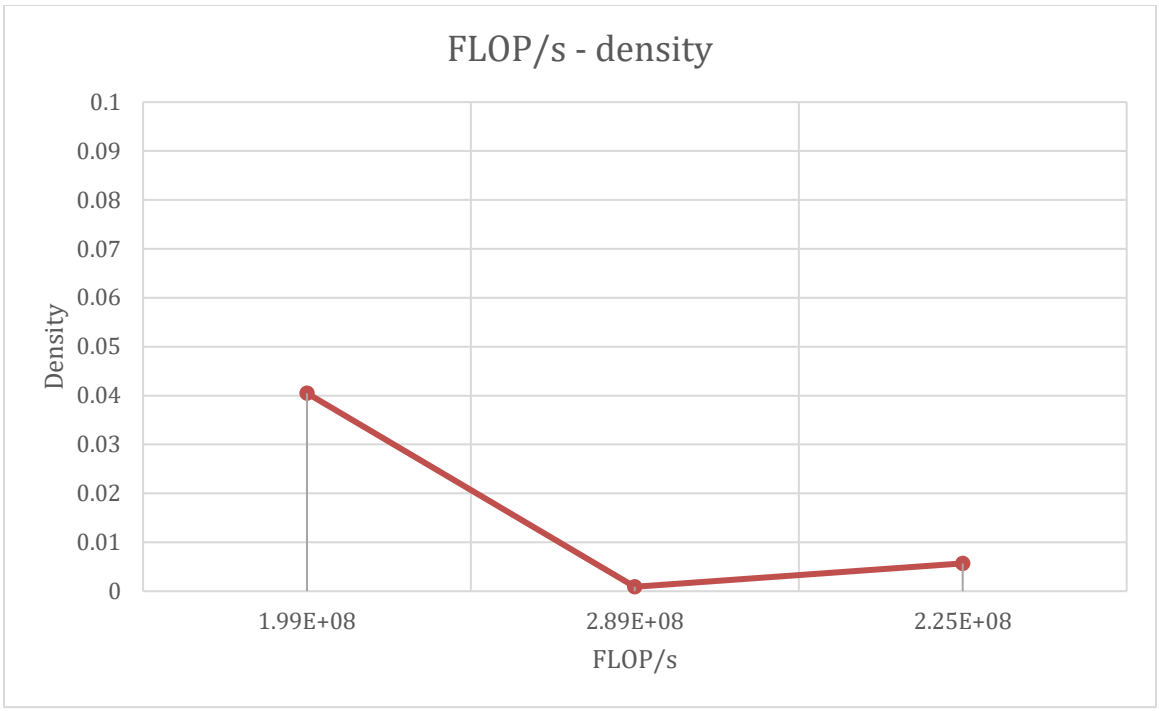
- Το μέγεθος του μητρώου, με κατηγορίες μικρά, μεσαία και μεγάλα μητρώα, που εξαρτάται απ' το μέγεθος των διαστάσεων του μητρώου.
- Η πυκνότητα του μητρώου, δηλαδή η αναλογία των μη μηδενικών στοιχείων, προς το σύνολο των στοιχείων του μητρώου
- Η συμμετρία των στοιχείων, μιας και μπορούμε να εξετάσουμε ψευδο-συμμετρικά μητρώα, δηλαδή αυτά που οι τιμές τους δεν είναι συμμετρικές, αλλά είναι πολύ κοντά σε συμμετρικές

Τα μητρώα που επιλέχθηκαν είναι τα ακόλουθα:

Όνομα	Μέγεθος	Μη Μηδενικά Στοιχεία	URL
smt	25,710x25,710	3,749,582	https://sparse.tamu.edu/TKK/smt
fv3	9,801x 9,801	87,025	https://sparse.tamu.edu/Norris/fv3
nd3k	9,000x9,000	3,279,690	https://sparse.tamu.edu/ND/nd3k






FLOP/s - density
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Density
1.99E+08
2.89E+08
2.25E+08
FLOP/s