



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Επιστημονικός Υπολογισμός I

Ενότητα 2 : Μοντέλα Υπολογισμού

Ευστράτιος Γαλλόπουλος

Τμήμα Μηχανικών Η/Υ & Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



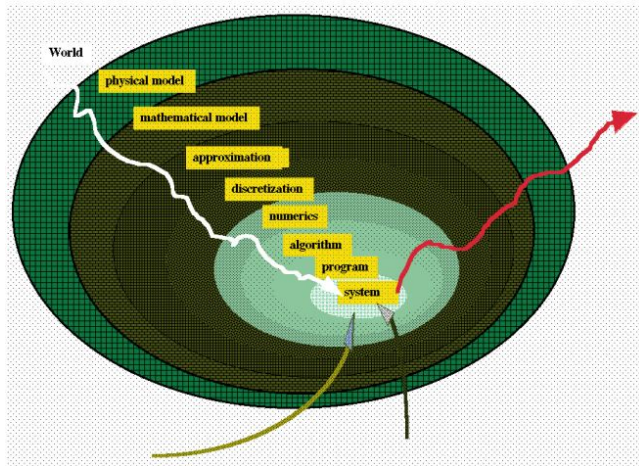
ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

- Μοντέλα υπολογισμού.
- Μοντέλο ιεραρχικής μνήμης.
- Μετρικές επίδοσης και απόδοσης προγραμμάτων στο μοντέλο ιεραρχικής μνήμης.

1 Υπενθύμιση

2 Υπολογιστικοί πυρήνες

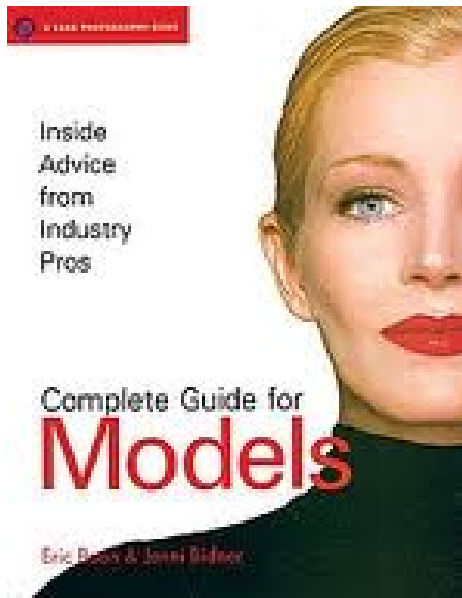
- Υπολογιστικό μοντέλο
- Αδυναμίες του μοντέλου RAM
- Μετρικές επίδοσης



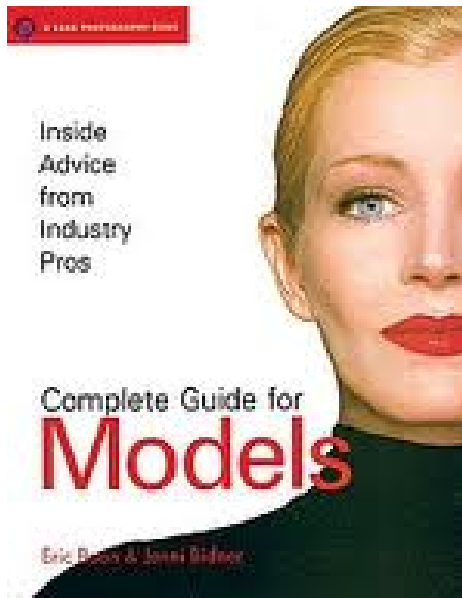
Σχήμα: CSE και μοντέλα: Από το άρθρο EG+Sameh'97, "CSE: Content and Product", IEEE CSE Magazine, 1997.

Τι μπορούν να πουν τα μοντέλα;

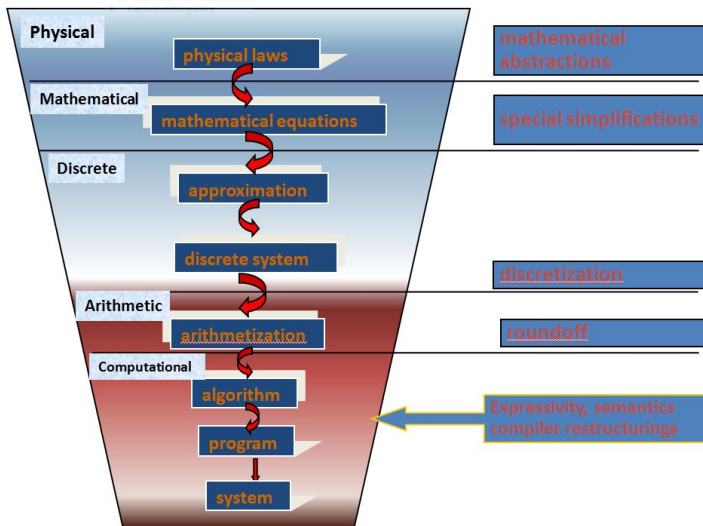
Ω Μένανδρε και βίε πότερος αρ' υμών
πότερος απεμιμήσατο



- Φυσικό
- Μαθηματικό
- Διακριτό
- Αριθμητικό
- Υπολογιστικό



Μοντέλα: Διαδρομές και Απώλεια Πληροφορίας



- Στον ΕΥ συνήθως πλοηγούμε μεταξύ Διακριτού, Αριθμητικού και Υπολογιστικού μοντέλου.
- Τα μοντέλα **βοηθούν** στο να προβαίνουμε σε **προβλέψεις** σχετικά με την **ταχύτητα**, την **ακρίβεια** και τα **κόστη** των υπολογισμών,
- ... να προσαρμόζουμε μεθόδους και εργαλεία
- ... και εντέλει να βελτιώνουμε τη **συνολική επίδοση**.

Η διαδρομή και ο στόχος

- Στον ΕΥ συνήθως πλοηγούμε μεταξύ Διακριτού, Αριθμητικού και Υπολογιστικού μοντέλου.
- Τα μοντέλα **βοηθούν** στο να προβαίνουμε σε **προβλέψεις** σχετικά με την **ταχύτητα**, την **ακρίβεια** και τα **κόστη** των υπολογισμών,
- ... να προσαρμόζουμε μεθόδους και εργαλεία
- ... και εντέλει να βελτιώνουμε τη **συνολική επίδοση**.

Κριτήρια

- **ταχύτητα** ... να παράγουμε γρήγορα τα αποτελέσματα (time to solution)
- **ακρίβεια** ... τα αποτελέσματα να είναι αξιόπιστα
- **κόστος** ... οι πόροι να είναι εντός προϋπολογισμού
- **ενέργεια** ... οι υπολογισμοί να είναι ενεργειακά αποτελεσματικοί
- **προϊόν** ... να εξυπηρετούμε ανάγκες των χρηστών, να παρέχονται επαναχρησιμοποιήσιμες υλοποιήσεις π.χ. σε μορφή λογισμικού, κ.λπ.

- πόσο γρήγορα επιτυγχάνεται η λύση ;
 - υπολογιστική πολυπλοκότητα και μεταφορές
 - αποτελεσματική χρήση των πόρων
- πόσο ακριβή είναι τα αποτελέσματα ;
 - → συγκριτικά με τη θεωρητική λύση ; σε σχέση με εναλλακτικές μεθόδους ; σε σχέση με τα δεδομένα εισόδου ;
- Πόσο στοιχίζει η υλοποίηση ;
- Ποιές είναι οι ενεργειακές ανάγκες ;
- Πώς αξιολογούνται οι πόροι που χρησιμοποιήθηκαν ;

- **Προσομοίωση in silico:** Έχει καταστεί ομότιμος τρόπος επιστημονικής έρευνας καθώς σε πάρα πολλές περιπτώσεις, τα πειράματα δεν αποτελούν επιλογή π.χ. λόγω κόστους, επικινδυνότητας, καταστροφικών επιπτώσεων, κ.λπ.
 - Ανάλυση δεδομένων
 - Γραφικά
 - Εφαρμογές πραγματικού χρόνου
-
- Σχεδιασμός και υλοποίηση
 - «Πακετάρισμα» σε βιβλιοθήκες
 - Πακέτα ως προϊόντα

Θέση 1 (GS97)

Ο Επιστημονικός Υπολογισμός χαρακτηρίζεται από διαδρομές μεταξύ των μοντέλων ιδιαίτερα μεταξύ του υπολογιστικού, του αριθμητικού και του διακριτού.

Θέση 2 (GS97)

“If there is no end to CSE that will be the end of CSE.” Προτείνουμε ως προϊόν τα Περιβάλλοντα Επίλυσης Προβλημάτων (Problem Solving Environments) π.χ. συστήματα όπως η MATLAB.

Εφαρμογές και Υπολογιστικοί Πυρήνες (DKLS86)

	1	2	3	4	5	6	7	8	9
<i>Lattice gauge (QCD)</i>	*	.	.	*	*
<i>quantum mechanics</i>	.	.	.	*	.	.	*	*	*
<i>meteorology</i>	*	*	.	.	.
<i>CFD</i>	*	.	*	.	*	*	.	.	.
<i>geodesy</i>	*	*
<i>inverse problems</i>	.	*	.	.	*
<i>structures</i>	*	.	*	*
<i>circuit simulation</i>	*	.	*	.	.	.	*	.	.
<i>electromagnetics</i>	*	*	*	*	*	*	.	.	.
<i>financial</i>	*	*	*	.	.	.	*	*	*
<i>IR, Machine Learning</i>	*	*	.	*	.	.	.	*	.
<i>DS& Image P.</i>	*	*	*	*	*	*	.	.	*
<i>Internet Algorithmics</i>	*	.	.	*	.	.	.	*	.

1. γραμμικά συστ.

2. ελάχιστα τετράγωνα

3. μη γραμμικά συστ.

4. ιδιοδιασπάσεις/SVD

5. ταχείς μετασχηματισμοί

6. ταχείς ελλειπτικοί επιλυτές

7. άκαμπτες ΔΕ

8. Μόντε Κάρλο

9. ολοκληρωματικοί μετασχ.

μετασχηματισμοί «κρυμμένοι» ως Μητρώα/ συλλογές δεδομένων «κρυμμένοι»
ως μητρώα

Υπολογιστικός πυρήνας

Τμήμα κώδικα που μπορεί να αντιστοιχεί σε κάποια μαθηματική πράξη (π.χ. πολλαπλασιασμός μητρώων, λύση συστήματος, υπολογισμός ιδιοδιανυσμάτων), όπου αναλώνεται σχετικά μεγάλος χρόνος εκτέλεσης σε σύγκριση με άλλα τμήματα του κώδικα.

Εύρημα - αφορά προγράμματα με υπολογισμούς μεγάλης κλίμακας

- 1 Ο μεγαλύτερος χρόνος υπολογισμών αναλώνεται σε εκτέλεση βρόχων.
- 2 Οι υπολογισμοί με μητρώα είναι **υπολογιστικοί πυρήνες** των περισσότερων σημαντικών εφαρμογών.



Η **Υπολογιστική Γραμμική Άλγεβρα** είναι όχημα για την παρουσίαση των τεχνικών του ΕΥ

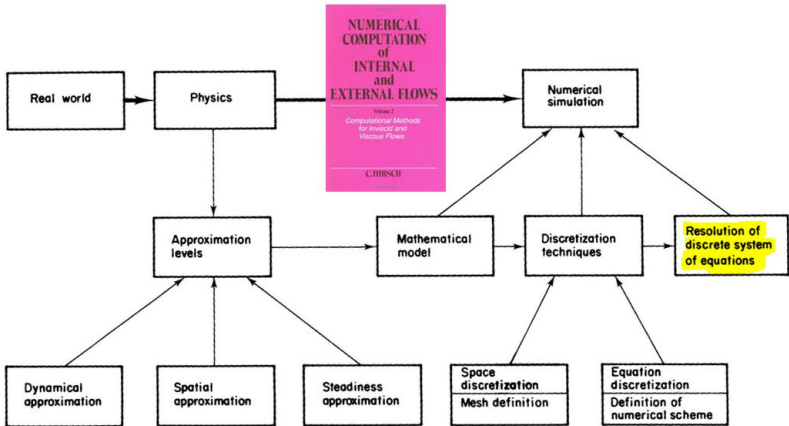


Figure 1.1 Computational models

Intel' RMS: Recognition, Mining, Synthesis (Dub05)

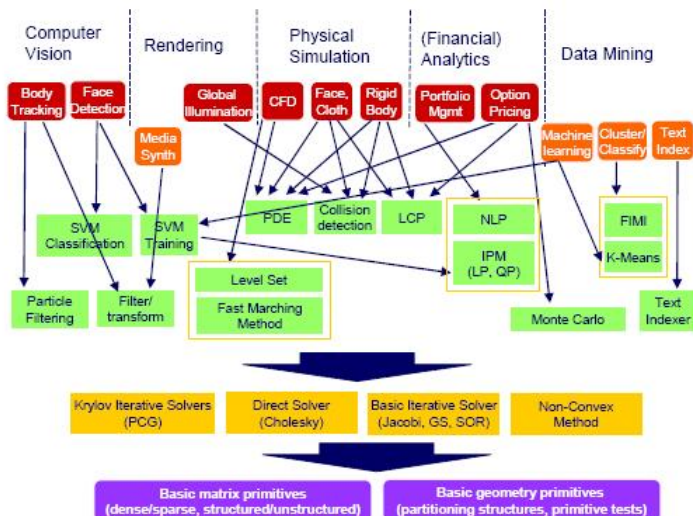
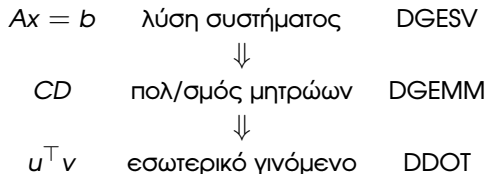


Figure 5. Intel's RMS and how it maps down to functions that are more primitive. Of the five categories at the top of the figure, Computer Vision is classified as Recognition, Data Mining is Mining, and Rendering, Physical Simulation, and Financial Analytics are Synthesis. [Chen 2006]

Ο πυρήνας ... του πυρήνα ... του πυρήνα

Προσοχή Οι υπολογισμοί με μητρώα έχουν συνήθως **ιεραρχική** δομή: π.χ.



Γιατί να ασχοληθούμε με υπολογιστικούς πυρήνες ;

- Συνήθως οι εφαρμογές είναι μεγάλης κλίμακας,
- Πολλές γραμμές πολύπλοκου κώδικα
- Συχνά αποτελούμενες από κώδικα σε περισσότερες από μια γλώσσες
- Μεγάλο μέγεθος δεδομένων
- Υπολογισμοί υψηλής πολυπλοκότητας

Εύρημα

Ακόμα και μέτρια επιτάχυνση των υπολογιστικών πυρήνων επιφέρει πολλαπλασιαστικό αποτέλεσμα και επιφέρει σημαντική μείωση του συνολικού χρόνου εκτέλεσης (χρονοβελτίωση)

Υπολογισμοί αντί για Επικοινωνία/Μεταφορές

Οι μεταφορές δεδομένων εκτελούνται με πολύ αργότερο ρυθμό από τους υπολογισμούς.

Παραλληλία

Μόνο με χρήση παράλληλης επεξεργασίας μπορούμε να αξιοποιήσουμε το ρυθμό αύξησης της υπολογιστικής ισχύος.

Ενεργειακά φράγματα

Οι υπολογισμοί και ακόμα περισσότερο οι μεταφορές είναι ενεργοβόροι και επιβάλλουν επιπλέον περιορισμούς.

Υπολογισμοί αντί για Επικοινωνία/Μεταφορές

Οι μεταφορές δεδομένων εκτελούνται με πολύ αργότερο ρυθμό από τους υπολογισμούς.

Παραλληλία

Μόνο με χρήση παράλληλης επεξεργασίας μπορούμε να αξιοποιήσουμε το ρυθμό αύξησης της υπολογιστικής ισχύος.

Ενεργειακά φράγματα

Οι υπολογισμοί και ακόμα περισσότερο οι μεταφορές είναι ενεργοβόροι και επιβάλλουν επιπλέον περιορισμούς.

Τα παραπάνω χαρακτηριστικά να λαμβάνονται υπόψη στην ανάπτυξη εργαλείων του Επιστημονικού Υπολογισμού. **Η έμφαση στο παρόν μάθημα είναι στο 1ο θέμα**

Στον πυρήνα της ιεραρχίας! Ένα υπολογιστικό μοντέλο περιγράφει μίαν *ιδεατή μηχανή* για την οποία μπορούμε να γράψουμε λογισμικό.

Παραδείγματα Turing machine, Random Access Machine (RAM), Random Access Stored Program (RASP) machine

Τι ζητάμε Μοντέλο που βοηθά στην πρόβλεψη της επίδοσης και παρέχει πληροφορίες για την βελτίωσή της.

Πρόκληση Δεν είναι εύκολο!

- Η μηχανή κρύβει αρχιτεκτονικές λεπτομέρειες και αλλαγές που οφείλονται σε καθαρά τεχνολογικές εξελίξεις (π.χ. μικρότερο κύκλο)
- Η ιδεατή μηχανή πρέπει να είναι αρκετά συγκεκριμένη ώστε να επιτρέπει την εξαγωγή συμπερασμάτων σχετικών με την επίδοση των προγραμμάτων που γράφονται γι' αυτήν.

- Κλασικά μοντέλα (όπως το RAM) επικεντρώνονταν στο πλήθος των αριθμητικών πράξεων (αριθμητικό κόστος ή αριθμητική πολυπλοκότητα)
- Για πολλά χρόνια η προσπάθεια αφορούσε στην κατασκευή αλγορίθμων που ελαχιστοποιούσαν την αριθμητική πολυπλοκότητα ή για εύρεση κάτω φραγμάτων γι' αυτήν.

Παράδειγμα: Πολλαπλασιασμός μητρώων

Αναφερόμαστε στην πράξη

$$C \leftarrow AB, \text{ όπου } A, B \in \mathbb{R}^{n \times n} \text{ ή και } C \leftarrow C + AB$$

- Εν συντομία MM για Matrix-Matrix multiplication (και για τις δύο εκδοχές)
- Ο κλασικός αλγόριθμος στοιχίζει περί τις $2n^3$ πράξεις
- Αλγόριθμος με υποκυβική πολυπλοκότητα, $\mathcal{O}(n^{\log_2 7})$, οφείλεται στον $(\log_2 7 = 2.81)$: Σημαντική ανακάλυψη (breakthrough) του **Strassen (Str69)**
- Ταχύτερος γνωστός αλγόριθμος χρειάζεται $\mathcal{O}(n^{2.376})$ πράξεις (Coppersmith, Winograd'90) (**CW90**)

Ανοικτό ερευνητικό ερώτημα:

Ποιά είναι η ελάχιστη τιμή ω ώστε να αρκούν $\mathcal{O}(n^{\omega+\epsilon})$ πολλαπλασιασμοί βαθμωτών για να εκτελεστεί η MM για οποιοδήποτε $\epsilon > 0$. (προφανώς $\omega \geq 2$).

An overview of the recent progress on matrix multiplication

Virginia Vassilevska Williams

November 3, 2012

Abstract

The exponent ω of matrix multiplication is the infimum over all real numbers c such that for all $\varepsilon > 0$ there is an algorithm that multiplies $n \times n$ matrices using at most $O(n^{c+\varepsilon})$ arithmetic operations over an arbitrary field. A trivial lower bound on ω is 2, and the best known upper bound until recently was $\omega < 2.376$ achieved by Coppersmith and Winograd in 1987. There were two independent improvements on ω , one by Stothers in 2010 who showed that $\omega < 2.374$, and one by myself that ultimately resulted in $\omega < 2.373$. Here I discuss the road to these improvements and conclude with some open questions.

Standard ή Superfast; 'Ό,τι λάμπει δεν είναι χρυσός!

Πρόβλημα υλοποίησης/κόστους/ταχύτητας

Παρά την αδιαμφισβήτητα χαμηλότερη πολυπλοκότητα των αλγορίθμων τύπου Strassen, Coppersmith και Winograd, δεν είχε καταστεί δυνατή μέχρι πρόσφατα η αξιοποίησή τους στις υπάρχουσες αρχιτεκτονικές.

Πρόβλημα ακρίβειας

Εγγενείς λόγοι φαίνεται να οδηγούν τις μεθόδους αυτές σε χειρότερη συμπεριφορά ως προς το μέγιστο αριθμητικό σφάλμα σε αριθμητική κινητής υποδιαστολής σε σύγκριση με τις κλασικές μεθόδους (Mil75).

Η κατάσταση ως σήμερα

Η συντριπτική πλειοψηφία των υλοποιήσεων για σύγχρονα συστήματα Η/Υ αφορούν σε παραλλαγές αλγορίθμων κυβικής αριθμητικής πολυπλοκότητας. Η ανάπτυξη πρακτικών μεθόδων μειωμένης πολυπλοκότητας είναι σε εξέλιξη και έχει πολύ μεγάλο ενδιαφέρον (βλ. (BDHS14, BB14) και αναφορές)!

Στις σύγχρονες αρχιτεκτονικές: **It's the memory stupid!** - R. Sites, 1996

- Το μοντέλο RAM δεν αρκεί για να προβλέψει την επίδοση
- Υπάρχει χάσμα μεταξύ της διάρκειας ενός κύκλου της ΚΜΕ και της ταχύτητας επικοινωνίας επεξεργαστή με τη μνήμη
- Παράδειγμα: Τυπικά, ο χρόνος που απαιτείται για να μεταφέρουμε δεδομένα από την κύρια μνήμη είναι 100 φορές περισσότερο από το χρόνο ενός κύκλου σε έναν πυρήνα επεξεργαστή.
- Πρόβλημα: Πώς μπορεί να μειωθεί η επιβάρυνση του κόστους;
- Πρόταση: Αναπτύσσοντας τεχνικές απόκρυψης (κόστους) μεταφορών.
- Πώς: Αποκαλύπτοντας και αξιοποιώντας την **τοπικότητα** των προγραμμάτων με την υποστήριξη υλικού, ιδίως της **κρυφής μνήμης**

Πόσο γρήγορα λύνεται το πρόβλημα; Συνήθως εννοούμε ποιός είναι ο χρόνος επίλυσης.

Ποιά είναι η ταχύτητα εκτέλεσης του προγράμματος από το υπολογιστικό σύστημα; Συνήθως εννοούμε τον ρυθμό εκτέλεσης, δηλ. το πλήθος των πράξεων που εκτελούνται ανά μονάδα χρόνου. Η μέτρηση αυτή συνήθως αφορά στο μέσο ρυθμό κατά την εκτέλεση, δηλαδή στο λόγο (πράξεις/χρόνος). Ενίοτε ενδιαφέρει η μεταβολή του ρυθμού καθώς εκτελούνται διαφορετικά τμήματα του κώδικα.

Μονάδα μέτρησης ταχύτητας

Mflop/s = Million floating-point operations per second (προφ. μέγκαφλοπς)

- Λόγω των εξελίξεων στους επεξεργαστές, συνηθίζεται πλέον να αναφερόμαστε σε Gflop/s (δισεκατομμύρια πράξεις α.κ.υ. το δευτερόλεπτο) ... ενώ βαδίζουμε (στο απώτερο μέλλον) προς Petaflop/s, μεσοπρόθεσμα προς (h)Exaflops
- Η μετρική αυτή θεωρεί ότι το πλήθος πράξεων στον ορισμό του ρυθμού εκτέλεσης είναι οι πράξεις α.κ.υ. δηλ. στο Ω .
- Όλοι οι επεξεργαστές σήμερα εκτελούν τις αριθμητικές πράξεις με πολύ υψηλούς ρυθμούς.
- Αυτό δεν σημαίνει ότι εκτελούν κάθε πρόγραμμα γρήγορα καθώς πρέπει να εκτελεστούν και άλλες πράξεις που δεν συνυπολογίζονται στο Ω π.χ. μεταφορές. Αυτές επηρεάζουν και μειώνουν το μέσο ρυθμό εκτέλεσης.

Πώς μετράμε την επίδοση (Mflor/s);

Θεωρητικά Εκτίμηση από το Ω και το Φ (πλήθος μεταφορών - **περισσότερα σε λίγο**)

Πρακτικά Προσεκτική μέτρηση του «χρόνου επίλυσης» και του «πλήθους πράξεων».

Ζητήματα:

- Χρειάζεται υποστήριξη (εργαλεία υλικού και λογισμικού),
- Χρειάζονται πολλές μετρήσεις και εξαγωγή στατιστικών στοιχείων από αυτές π.χ. μέσος όρος και απόκλιση
- συνήθως απορρίπτονται οι εξωφρενικές τιμές (outliers) ή διερευνάται γιατί εμφανίζονται.



... we combine LOAD and STORE into the arithmetic operations by replacing sequences such as LOAD a ; ADD b ; STORE c by $c \leftarrow a+b$ (Aho et al. '74)

Παράδειγμα

Κώδικας 1: MATLAB Horner, $\Omega = 2n$

```
1 s = a(n);  
2 for j=n-1:-1:1  
3     s = s*x + a(j);  
4 end;
```

- Αμελείται το κόστος επικοινωνίας
- \Rightarrow συχνά γίνονται λάθος προβλέψεις
- π.χ. αλγόριθμοι για το ίδιο πρόβλημα με το ίδιο πλήθος αριθμ. πράξεων που έχουν πολύ διαφορετική επίδοση!

Abstraction should not be an obstruction

(Joseph Gorse, 2013)

Από συζήτηση στο δίκτυο *ResearchGate* επί της ερώτησης
*Should we hide the complexity of computer systems behind
layers of abstraction?*

Παράδειγμα

MV: Πολλαπλασιασμός μητρώου με διάνυσμα

Κώδικας 2: MV ανά γραμμές

```
1 function [y] = ...
    mulrLOOPS(A, x);
2 [m,n]=size(A);
3 for i=1:m, y(i) = 0; ...
    end; % y <- 0
4 for i = 1:m
5     for j = 1:n
6         y(i) = y(i) ...
            + ...
            A(i, j)*x(j);
7     end
8 end
```

Κώδικας 3: MV ανά στήλες

```
1 function [y] = ...
    mulcLOOPS(A, x);
2 [m,n]=size(A);
3 for i=1:m, y(i) = 0; ...
    end; % y <- 0
4 for j = 1:n
5     for i = 1:m
6         y(i) = y(i) ...
            + ...
            A(i, j)*x(j);
7     end
8 end
```

Κώδικας 4: MV ανά γραμμές

```
1 function [y] = ...  
    mulr(A,x);  
2 [m,n]=size(A); y = ...  
    zeros(m,1);  
3 for i=1:m  
4     y(i) = A(i,:)*x;  
5 end
```

Κώδικας 5: MV ανά στήλες

```
1 function [y] = ...  
    mulc(A,x);  
2 [m,n]=size(A);  
3 y = A(:,1)*x(1);  
4 for j=2:n  
5     y = y + A(:,j)*x(j);  
6 end
```

Κώδικας 6: Ενδογενής

```
1 function [y] = mulmv(A,x);  
2 y = A*x; % y = mtimes(A,x);
```

Σχετικά με τις μετρήσεις

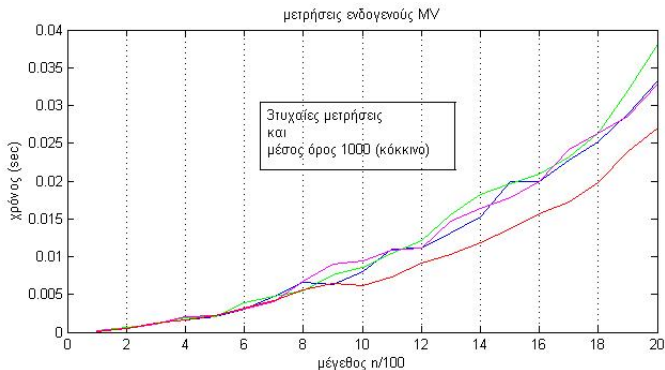
Σύστημα Intel Atom N270 @ 1.6 GHz, RAM 1.48 GB, caches: level 1, 32KB; level 2, 512KB cache (write-back). Οι πληροφορίες μέσω του προγράμματος `winaudit.exe`.

Λογισμικό Windows XP Pro, MATLAB 7.11 (R2010b)

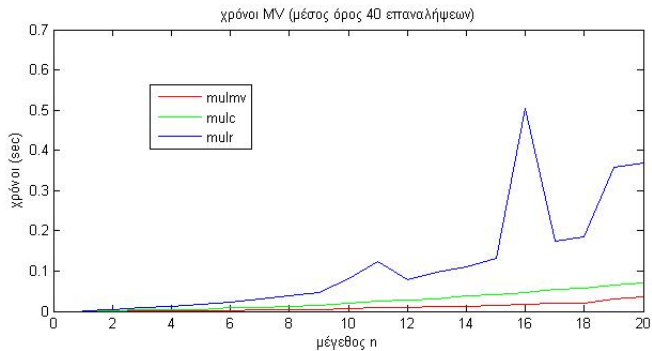
Μετρήσεις Λήφθηκαν με τις εντολές `tic`, `toc`.

Κώδικας 7: script χρονομέτρησης για `mtimes`

```
1 A = rand(2000,2000);
2 x = rand(2000,1);
3 total=40; jstart=100;
4 jend = 2000; jstep = 100;
5 for j= jstart:jstep:jend
6     Aj=A(1:j,1:j); xj = x(1:j);
7     tsum =0;
8     for itimes=1:total
9         tic; y = mtimes(Aj,xj); %y=mulr(Aj,xj);
10        tsum = tsum + toc;
11    end
12    tmv(j/100,3)=tsum/total;
13 end
```



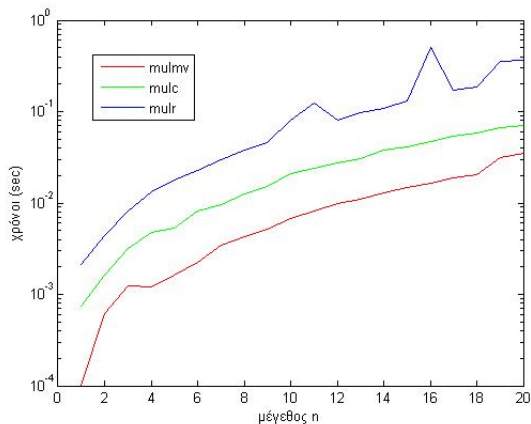
Σχήμα: Χρόνοι εκτέλεσης ενδογενούς `mtimes` (εμείς την έχουμε πακετάρει εντός της `pcodemulmv`) για $n=100:10:2000$. Παρουσιάζονται 3 τυχαίες μετρήσεις και ο μέσος όρος 1000 μετρήσεων



Σχήμα: Μέσοι χρόνοι εκτέλεσης των mulmv (mtimes), mult, mulc για $n=100:10:2000$.

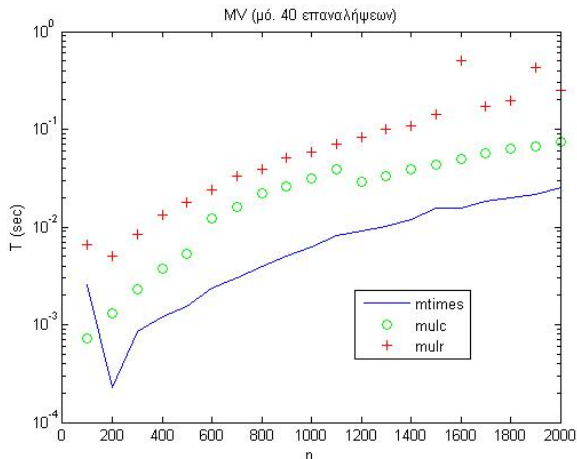
Ενδεικτικές μετρήσεις

Ίδια δεδομένα σε λογαριθμική κλίμακα



Σχήμα: Μέσοι χρόνοι εκτέλεσης των `mulmv` (`mtimes`), `mult`, `mulc` για $n=100:10:2000$. Προσέξτε ότι η λογαριθμική κλίμακα για το χρόνο αναδεικνύει πολύ καλύτερα τις διαφορές στην επίδοση για όλα τα μεγέθη n .

Επανάληψη μετρήσεων (μέσος όρος διάρκειας)



Σχήμα: Επανάληψη με λιγότερες μετρήσεις $n=100 : 100 : 2000$. Οι μετρήσεις των `mulc`, `mulr` σημειώνονται με στίγμα αντί συνεχή γραμμή. Έχουμε μεγαλύτερη πιστότητα ίσως όμως λιγότερη ευκρίνεια.



A.R. Benson and G. Ballard.

A framework for practical parallel fast matrix multiplication.

arXiv preprint arXiv:1409.2908, 2014.



G. Ballard, J. Demmel, O. Holtz, and O. Schwartz.

Communication costs of strassen's matrix multiplication.

Comm. ACM, 57(2):107–114, 2014.



D. Coppersmith and S. Winograd.

Matrix multiplication via arithmetic progressions.

J. Symb. Comput., 9(3):251–280, 1990.



E. Davidson, D. Kuck, D. Lawrie, and A. Sameh.

Supercomputing tradeoffs and the Cedar system.

Technical Report 577, Center for Supercomputing Research and Development, Urbana, May 1986.



P. Dubey.

A Platform 2015 Workload Model Recognition, Mining and Synthesis Moves Computers to the Era of Tera: Executive Summary.

White paper, Microprocessor Technology Lab, Intel Corporation, 2005.



E. Gallopoulos and A.H. Sameh.

CSE: Content and product.

IEEE Computational Science & Engineering Mag., 4(2):39–43, 1997.



W. Miller.

Computational complexity and numerical stability.

SIAM J. Comput., 4(2):97–107, June 1975.



V. Strassen.

Gaussian elimination is not optimal.

Numer. Math., 13:354–356, 1969.



Ε. Γαλλόπουλος.

Επιστημονικός Υπολογισμός I.

Πανεπιστήμιο Πατρών, 2008.

- 1 http://ecx.images-amazon.com/images/I/41QaefwoZqL._SY344_BO1,204,203,200_.jpg (βλ. σελ 6)
- 2 <http://theory.stanford.edu/~virgi/sigactcolumn.pdf> (βλ. σελ 22)
- 3 http://www.gobosoft.com/eiffel/gobo/image/book/data_structures_and_algorithms.gif (βλ. σελ 27)

Copyright Πανεπιστήμιο Πατρών - Ευστράτιος Γαλλόπουλος 2015

“Επιστημονικός Υπολογισμός Ι”, Έκδοση: 1.0, Πάτρα 2013-2014.

Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.upatras.gr/courses/CEID1096/>

Τέλος Ενότητας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης