

Επιστημονικός Υπολογισμός Ι

ΗΥ 343

Ε. Γαλλόπουλος

Τμήμα Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών



Πανεπιστήμιο Πατρών



Παρατηρήσεις

- Υπολογιστικοί πυρήνες
- Επικρατεί γραμμική άλγεβρα
- Ακόμα και στο μετασχηματισμό Fourier!

*Η Υπολογιστική Γραμμική Άλγεβρα θα αποτελέσει το
όχημα για την παρουσίαση των τεχνικών του ΕΥ*



Πανεπιστήμιο Πατρών



- 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

Ακόμα και «μέτρια» επιτάχυνση συχνά καλούμενων υποπρογραμμάτων / υπολογιστικών πυρήνων συνήθως έχει πολλαπλασιαστικό αποτέλεσμα και επιφέρει μεγάλη συνολική επιτάχυνση



Θεμελιώδη προβλήματα ΑΓΑ (1/3)

- **ΑΓΑ0: Βασικές πράξεις – (άμεσα προβλήματα)**

$$C \leftarrow C + AB, A \in \mathbb{R}^{n_1 \times n_2}, B \in \mathbb{R}^{n_1 \times n_3}, C \in \mathbb{R}^{n_2 \times n_3}$$
- **ΑΓΑ1: Επίλυση γραμμικών συστημάτων**

$$Ax = b, A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$$
- **ΑΓΑ2: Πρόβλημα ελαχίστων τετραγώνων**

$$\arg \min_{x \in \mathbb{R}^n} \|b - Ax\|_2, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$



Θεμελιώδη προβλήματα ΑΓΑ (2/3)

- ΑΓΑ3, ΑΓΑ4: Πρόβλημα ιδιοτιμών και γενικευμένων ιδιοτιμών

$$Ax = \lambda x, A \in \mathbb{C}^{n \times n}, x \in \mathbb{C}^n, \lambda \in \mathbb{C}$$

$$Ax = \lambda Bx, A \in \mathbb{C}^{n \times n}, x \in \mathbb{C}^n, \lambda \in \mathbb{C}, B \in \mathbb{C}^{n \times n}$$

- ΑΓΑ5: Διάσπαση ιδιαζουσών τιμών

$$A = U \Sigma V^T, U \in \mathbb{C}^{m \times m}, V \in \mathbb{C}^{n \times n}, \Sigma \in \mathbb{R}^{m \times n},$$

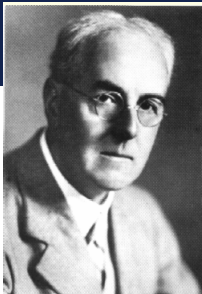
$$U^*U = I, V^*V = I, \Sigma = \text{diag}[\sigma_1, \dots, \sigma_r], \sigma_j \geq 0$$

- ΑΓΑ6: Υπολογισμός συναρτήσεων μητρώων

$$\phi(A), \phi(A)b, A \in \mathbb{C}^{n \times n}, b \in \mathbb{C}^n$$



Πανεπιστήμιο Πατρών



Lewis Fry Richardson (1881 - 1953)

«*Weather Prediction by Numerical Process*» (1922)

Richardson's interest in meteorology led him to propose a scheme for weather forecasting by solution of differential equations, the method used today, though, when he published *Weather Prediction by Numerical Process* in 1922, suitable fast computing was unavailable. He described his ideas thus :-

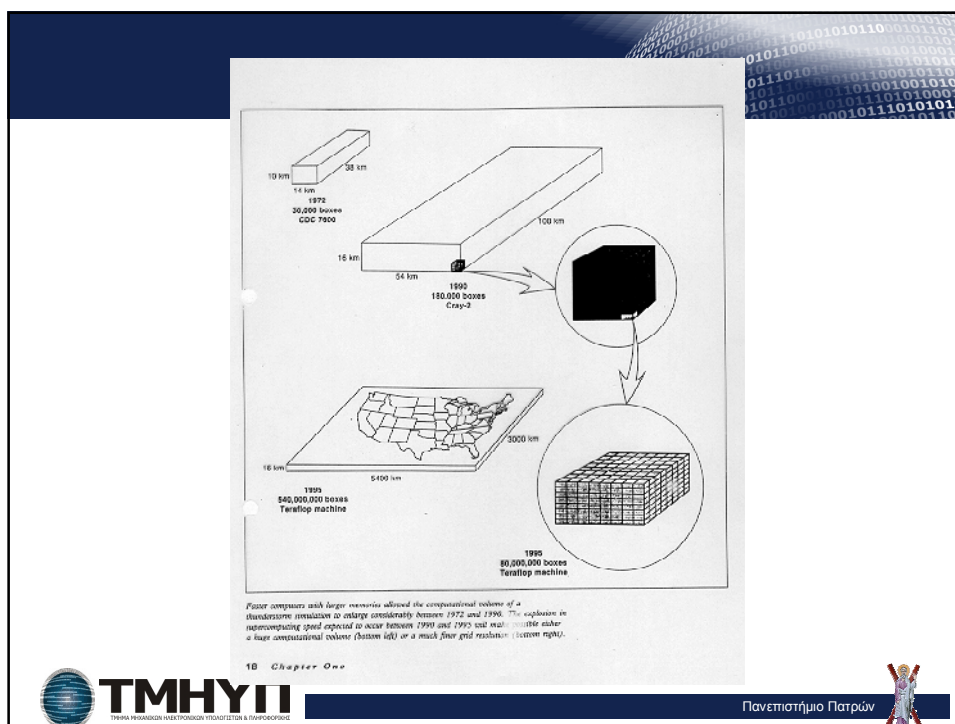
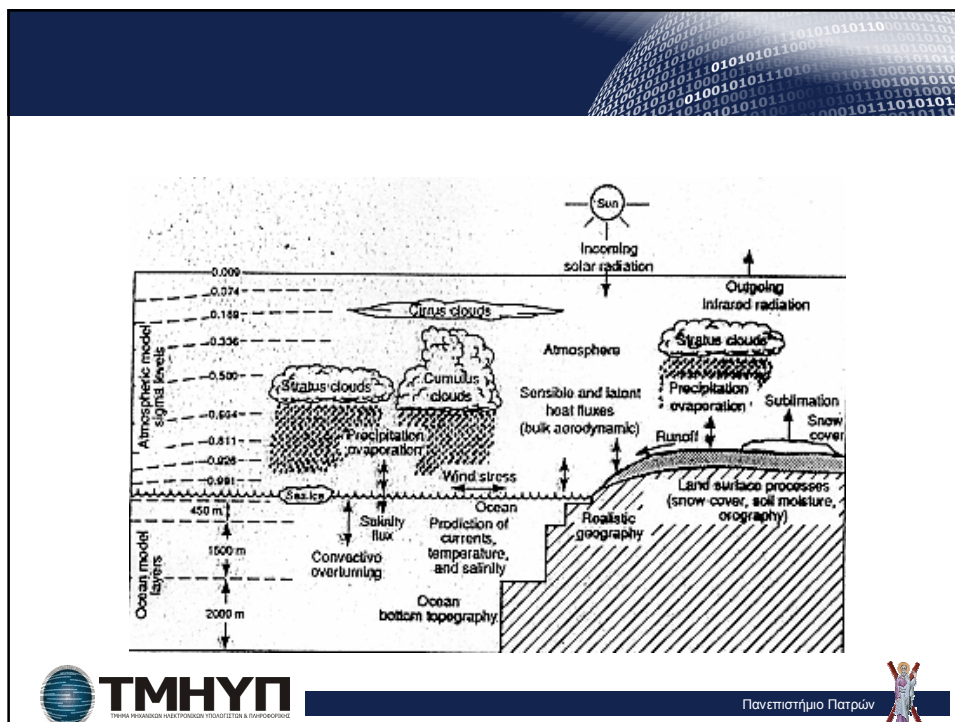
"After so much hard reasoning, may one play with a fantasy? Imagine a large hall like a theatre, except that the circles and galleries go right round through the space usually occupied by the stage. The walls of this chamber are painted to form a map of the globe. The ceiling represents the north polar regions, England is in the gallery, the tropics in the upper circle, Australia on the dress circle and the antarctic in the pit.

A myriad computers are at work upon the weather of the part of the map where each sits, but each computer attends only to one equation or part of an equation. The work of each region is coordinated by an official of higher rank. Numerous little "night signs" display the instantaneous values so that neighbouring computers can read them. Each number is thus displayed in three adjacent zones so as to maintain communication to the North and South on the map.

From the floor of the pit a tall pillar rises to half the height of the hall. It carries a large pulpit on its top. In this sits the man in charge of the whole theatre; he is surrounded by several assistants and messengers. One of his duties is to maintain a uniform speed of progress in all parts of the globe. In this respect he is like the conductor of an orchestra in which the instruments are slide-rules and calculating machines. But instead of waving a baton he turns a beam of rosy light upon any region that is running ahead of the rest, and a beam of blue light upon those who are behindhand.

Four senior clerks in the central pulpit are collecting the future weather as fast as it is being computed, and despatching it by pneumatic carrier to a quiet room. There it will be coded and telephoned to the radio transmitting station. Messengers carry piles of used computing forms down to a storehouse in the cellar.

In a neighbouring building there is a research department, where they invent improvements. But these is much experimenting on a small scale before any change is made in the complex routine of the computing theatre. In a basement an enthusiast is observing eddies in the liquid lining of a huge spinning bowl, but so far the arithmetic proves the better way. In another building are all the usual financial, correspondence and administrative offices. Outside are playing fields, houses, mountains and lakes, for it was thought that those who compute the weather should breathe of it freely." (Richardson 1922)



Κριτήρια αξιολόγησης

- Να φθάσουμε γρήγορα ... **Ταχύτητα**
 - χωρίς να τρακάρουμε ... **Ακρίβεια**
 - με **VW** αντί **BMW** ... **Κόστος**
- Πόσο γρήγορα επιτυγχάνεται η λύση
 - ❖ Υπολογιστική πολυπλοκότητα
 - ❖ Καλή χρήση των υπολογιστικών πόρων
 - Πόσο ακριβή είναι τα αποτελέσματα
 - ❖ Συγκριτικά με τη «θεωρητική» λύση
 - ❖ Σχετικά με τα δεδομένα
 - Τι πόροι (σε υλικό, λογισμικό, ανθρωποώρες) χρειάζονται για την υλοποίηση



ΤΜΗΥΠ
ΤΕΧΝΗ ΠΡΟΒΛΕΨΕΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΔΕΙΞΤΕΩΝ & ΠΑΝΕΠΙΣΤΗΜΙΟ

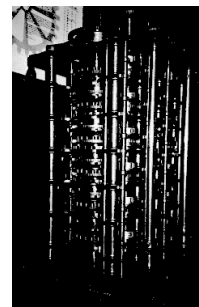
Πανεπιστήμιο Πατρών



Η σημασία της επίδοσης



• As soon as an Analytical Engine exists, it will necessarily guide the future course of science. Whenever any result is sought by its aid, the question will then arise - By what course of calculation can these results be arrived at by the machine in the shortest time? *Charles Babbage, [Passages from the Life of a Philosopher, 1864]*



ΤΜΗΥΠ
ΤΕΧΝΗ ΠΡΟΒΛΕΨΕΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΔΕΙΞΤΕΩΝ & ΠΑΝΕΠΙΣΤΗΜΙΟ

Πανεπιστήμιο Πατρών



Σχεδόν πάντα υπάρχουν, παράγονται και διαδίδονται αριθμητικά σφάλματα!

- Οι παρακάτω πράξεις είναι ισοδύναμες και το αποτέλεσμα ίσο με **10**, αλλά αν τις εκτελέσετε σε ένα τυπικό επεξεργαστή (π.χ. σε **MATLAB**)

- $10^{20} - 10 - 10^{20} + 20 = 20$

- $10^{20} + 20 - 10^{20} - 10 = -10$

- $10^{20} + 20 - 10 - 10^{20} = 0$

- ΚΑΝΕΝΑ αποτέλεσμα δεν είναι σωστό;!! Όμως είναι το

- $10^{20} - 10^{20} + 20 - 10 = 10$

- Ερώτηση: Από τους **24 (=4!)** τρόπους υπολογισμού παραπάνω, ποιοί επιστρέφουν σωστό αποτέλεσμα;

- ΓΙΑΤΙ;



TMHYP
ΤΕΧΝΙΚΗ ΜΕΤΑΦΡΑΣΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ

Πανεπιστήμιο Πατρών



Ταχύτητα

- Θέλουμε ταχύτερη επίλυση
- Τι βοηθά στην ταχύτερη επίλυση;
- Τι μετράμε;
 - χρόνο αναμονής για αποτέλεσμα
 - χρόνο χρήσης των πόρων του συστήματος
 - χρόνο χρήσης του επεξεργαστή
- Αξιολόγηση επίδοσης (performance evaluation)

Πως μπορούμε να προβλέψουμε την ταχύτητα;



TMHYP
ΤΕΧΝΙΚΗ ΜΕΤΑΦΡΑΣΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ

Πανεπιστήμιο Πατρών



Ακρίβεια

- Θέλουμε «σωστές» απαντήσεις
 - θεωρούμε ότι δεν υπάρχουν σφάλματα λογικής στο πρόγραμμα
 - Συνήθως τα αποτελέσματα περιέχουν “λάθη”!
 - ❖ Σφάλματα εισόδου
 - ❖ Σφάλματα διακριτοποίησης
 - ❖ Σφάλματα στρογγύλευσης
- Πως μπορούμε να προβλέψουμε το σφάλμα;***



Πανεπιστήμιο Πατρών



Προκλήσεις

- Βελτιωτικές αλλαγές σε υπολογισμούς πυρήνα έχουν πολλαπλασιαστικό αποτέλεσμα.
- Η βελτιστοποίηση της ακρίβειας και της ταχύτητας συνήθως είναι αντικρουόμενοι στόχοι
 - Σχηματικά
$$\text{ΧΡΟΝΟΣ ΥΠΟΛΟΓΙΣΜΟΥ} = O(1/\text{ΑΚΡΙΒΕΙΑ})$$
Τίθεται το θέμα αντιστάθμισης ώστε να επιτευχθεί ικανοποιητικός συνδυασμός ταχύτητας και ακριβειας.



Πανεπιστήμιο Πατρών



Έλεγχος ακρίβειας

- Τουλάχιστον
 - να γνωρίζουμε την επίδραση των σφαλμάτων στα αποτελέσματα
 - Να εξασφαλίζουμε αποδεκτό σφάλμα
 - Να μπορούμε να ποσοτικοποιούμε το σφάλμα στο υπολογισμένο αποτέλεσμα
- Τα εργαλεία κρίνονται από το αν μας ενημερώνουν σχετικά με την ποιότητα των απαντήσεων



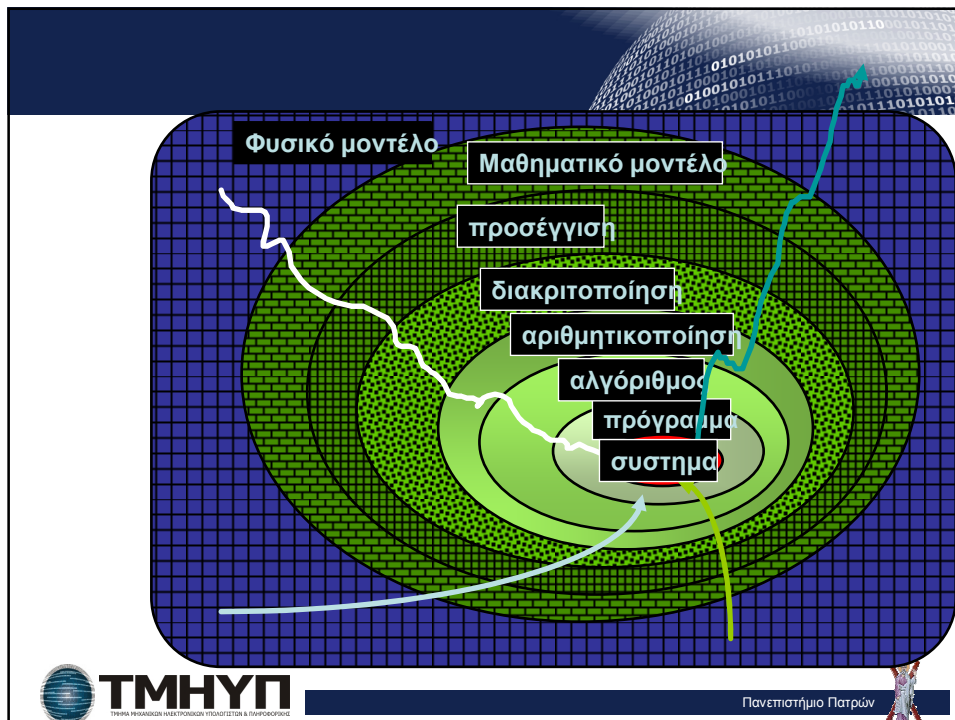
Μοντέλα@ επιστήμη& τεχνολογία

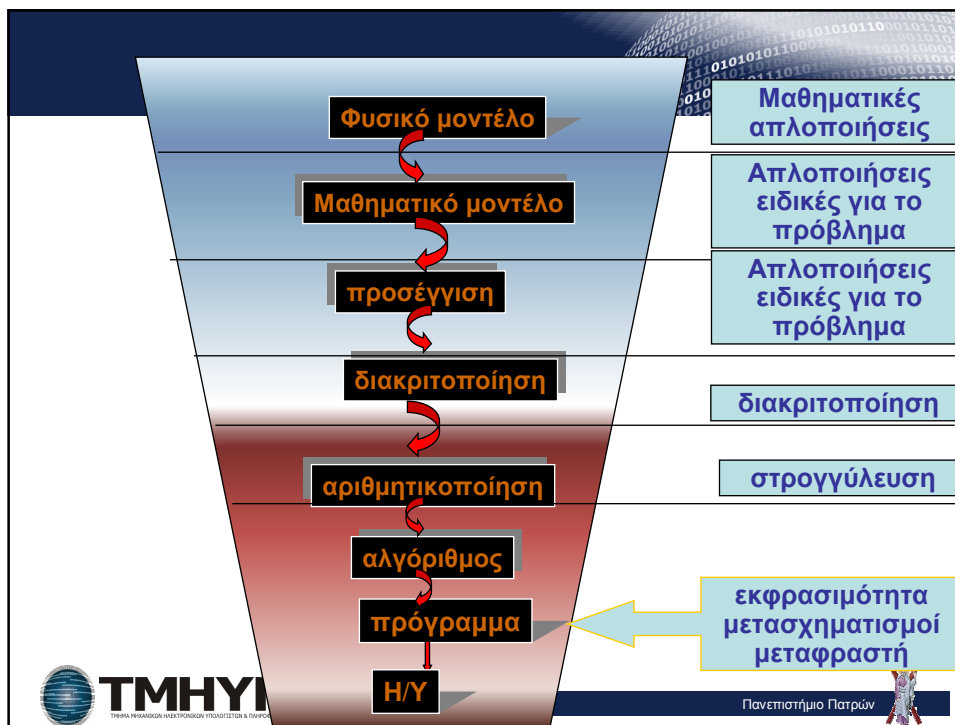
- **Model = πρότυπο**
 - ... 6. Η τρισδιάστατη αναπαράσταση σε μικρογραφία ή μεγέθυνση ... [Λεξικό Μπαμπινιώτη]
 - π.χ. Φυσικό ομοίωμα πλοίου, αεροπλάνου, ...
 - πραγματικά φυσικά πρότυπα / τεχνουργήματα που τα συστατικά τους μέρη οι σχέσεις τους μεταξύ τους και η λειτουργία εμφανίζει αξιοσημείωτη αναλογία με κάποιο άλλο σύστημα, ώστε παρατηρώντας το τεχνουργημα, επιφέροντας σ' αυτό αλλαγές και βλέποντας τις συνεπειές τους, να μπορεί κανείς να συναγάγει συμπεράσματα για το σύστημα του οποίου το τεχνουργημα αποτελεί πρότυπο. [Λεξ. *Unesco*]



- **Model**

- πρότυπο
- The term “mathematical model” will be used for any complete and consistent set of mathematical equations which is thought to correspond to some other entity, its prototype... being derived from “modus” (a measure) the word “model” implies a **change of scale in its representation** ... certain properties have been removed and simplifications made ... [Rutherford Aris]





Μαθηματικά μοντέλα

$$\nabla \cdot \mathbf{D} = 4\pi\rho, \nabla \times \mathbf{H} = \frac{4\pi}{c} \mathbf{J}, \nabla \cdot \mathbf{B} = 0,$$

$$\nabla \times \mathbf{E} + \frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{\nabla P}{\rho} + \nu \nabla^2 \mathbf{u}$$

$$\frac{\partial C}{\partial t} + \frac{1}{2} \frac{\partial^2 C}{\partial s^2} \sigma^2 S^2 + rS \frac{\partial C}{\partial S} - rC = 0$$

TMHYI

Πανεπιστήμιο Πατρών

THE GRAND CHALLENGE EQUATIONS

$$B_i A_i = E_i A_i + \rho_i \sum_j B_j A_j F_{ji} \quad \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad \vec{F} = m \vec{a} + \frac{dm}{dt} \vec{v}$$

$$dU = \left(\frac{\partial U}{\partial S} \right)_V dS + \left(\frac{\partial U}{\partial V} \right)_S dV \quad \nabla \cdot \vec{D} = \rho \quad Z = \sum_j g_j e^{-E_j/kT}$$

$$F_j = \sum_{k=0}^{N-1} f_k e^{2\pi i j k / N} \quad \nabla^2 u = \frac{\partial u}{\partial t} \quad \nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J} \quad P(t) = \frac{\sum_i W_i B_i(t) P_i}{\sum_i W_i B_i(t)}$$

$$p_{n+1} = r p_n (1 - p_n) \quad \nabla \cdot \vec{B} = 0$$


$$-\frac{\hbar^2}{8\pi^2 m} \nabla^2 \Psi(r, t) + V \Psi(r, t) = -\frac{\hbar}{2\pi i} \frac{\partial \Psi(r, t)}{\partial t} \quad -\nabla^2 u + \lambda u = f$$

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{1}{\rho} \nabla p + \gamma \nabla^2 \vec{u} + \frac{1}{\rho} \vec{F} \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f$$

• NEWTON'S EQUATIONS • SCHROEDINGER EQUATION (TIME DEPENDENT) • NAVIER-STOKES EQUATION •
 • POISSON EQUATION • HEAT EQUATION • HELMHOLTZ EQUATION • DISCRETE FOURIER TRANSFORM •
 • MAXWELL'S EQUATIONS • PARTITION FUNCTION • POPULATION DYNAMICS •
 • COMBINED 1ST AND 2ND LAWS OF THERMODYNAMICS • RADIOSITY • RATIONAL B-SPLINE •

Μοντέλα στον ΕΥ

- Στον ΕΥ πλοηγούμε αδιάκοπα μεταξύ διαφορετικών μοντέλων
- Τα βασικά μοντέλα στο HY343 είναι τα
 - υπολογιστικό μοντέλο
 - αριθμητικό μοντέλο
 - μοντέλο διακριτοποίησης
 - χρησιμοποιούνται για να επιτρέπουν στο σχεδιαστή και στο χρήστη να κάνει προβλέψεις σχετικές με την επίδοση των μεθόδων και την ακρίβεια των λύσεων


TMHYP
Πανεπιστήμιο Πατρών

Υπολογιστικό μοντέλο

- Ένα υπολογιστικό μοντέλο περιγράφει μια “ιδεατή μηχανή” για την οποία μπορούμε να γράψουμε λογισμικό.
 - Turing machine
 - Random Access Machine
 - Random Access Stored Program (RASP)



Υπολογιστικό μοντέλο

- Ένα υπολογιστικό μοντέλο περιγράφει μια “ιδεατή μηχανή” για την οποία μπορούμε να γράψουμε λογισμικό.
- 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.



Εξελίξεις

- Υ> κύκλος ρολογιού
- Υ> *RISC*: μικρό *CPI*, *Load/Store*
- Υ> **Superscalar**: $CPI < 1$
- Υ,Λ> **pipelining**
- Υ,Λ,Θ> παραλληλία
- Υ,Λ> **NOW**
- Υ,Λ> **Internet** και **Grid**
- Λ> **Multithreading**
- Υ> **Multicore**
- Υ> **GPU**
- Υ,Λ,Θ> **Ιεραρχία μνήμης**
- Υ,Λ> **interleaving**
- Λ,Υ,Θ> **μεταφραστές**
- Λ> γλώσσες
- Λ> **λειτουργικά**
- Λ> **βιβλιοθήκες**
- Λ> **Περιβάλλοντα επίλυσης προβλημάτων**
- Θ,Λ> **Νέοι αλγόριθμοι**



TMHYP
ΤΕΧΝΗ ΜΕΤΑΦΡΑΣΤΕΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΔΙΟΤΥΠΩΝ & ΣΥΝΔΕΣΜΟΓΡΑΦΩΝ

Πανεπιστήμιο Πατρών



Χαρακτηριστικά RISC

- **Reduced Instruction Set Computers:**
 - Μικρό **clocks per cycle**
 - Χωρίς μικροκώδικα
 - Μικρός αριθμός εντολών
 - Ίδιου μήκους η κάθε μια
 - Πρόσβαση στη μνήμη μόνο με **LOAD, STORE**
 - **register files** αναγκαία
 - Έντονη χρήση **pipelining**
 - Το υπολογιστικό μοντέλο που θα χρησιμοποιήσουμε θα είναι αρχιτεκτονική **LOAD/STORE**



TMHYP
ΤΕΧΝΗ ΜΕΤΑΦΡΑΣΤΕΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΔΙΟΤΥΠΩΝ & ΣΥΝΔΕΣΜΟΓΡΑΦΩΝ

Πανεπιστήμιο Πατρών



Γλώσσες και Περιβάλλοντα Επίλυσης Προβλημάτων

- Fortran, C
- C++, Java
- Άλλες? (imperative, functional, ...)
- Perl, Python, (scripting – σεναριογλώσσες!)
- MATLAB, SciLab
- Mathematica, Maple, Axiom

I don't know what the technical characteristics of the standard language for scientific and engineering computation in the year 2000 will be ... but I know it will be called Fortran

Αποδίδεται στον John Backus (τηςBNF)

I have not written a single statement in Fortran (or C) since 1988

Cleve Moler (ιδρυτής Mathworks)



Πανεπιστήμιο Πατρών



Υπολογισμός -επικοινωνία

- Μεγάλη διαφοροποίηση στην ταχύτητα επεξεργασίας σε σχέση με την ταχύτητα επικοινωνίας επεξεργαστή/μνήμης
- Δεδομένο: Η ταχύτητας επεξεργασίας έχει μεγαλώσει πολύ σε σχέση με την ταχύτητα μεταφορών από/προς τη μνήμη
- Πρόβλημα: Πώς μπορεί να μειωθεί η επιβάρυνση του κόστους;
- Πρόταση: Τεχνικές απόκρυψης (κόστους) μεταφορών;



Πανεπιστήμιο Πατρών



Θέματα ιεραρχίας μνήμης

- Interleaving (διαφύλλωση) των memory banks
 - Κύρια μνήμη οργανωμένη σε τμήματα
 - Pipelined πρόσβαση μεταξύ τμημάτων
- Ιεραρχία μνήμης
 - Η αποτελεσματική χρήση της ιεραρχίας εξαρτάται από την εγγενή χρονική και χωρική τοπικότητα του προγράμματος

επίπεδο	0	1 (α, β)	2	3, 4
όνομα	καταχωρητές	κρυφή μνήμη	κύρια μνήμη	δίσκος, ...
Πρόσβαση (κύκλοι)	0.5-1	2-5	10-50	10 ⁶
εύρος MB/s	800	200	133	4
διαχείριση	μεταφραστής	υλικό	Λ/Σ	Λ/Σ - χρήστης
τυπικό μέγεθος	KB	MB, GB	GB	



Χαρακτηριστικό στο υλικό της κρυφής μνήμης

- Φόρτωση σε cache lines: Όταν ζητηθεί ένα στοιχείο από τη μνήμη δεν μεταφέρεται μόνον αυτό αλλά το cache line που το περιέχει (32 – 128B)
- Προφόρτωση δεδομένων στην ταχύτερη μνήμη πριν χρειαστούν
 - Εκμετάλλευση μέσω υλικού ή/κ λογισμικού της δυνατότητας ταυτόχρονης εκτέλεσης αριθμητικών πράξεων με το **LOAD** από τη μνήμη



Είδη τοπικότητας

- Χρόνου
 - Αν τώρα ζητηθεί το στοιχείο από τη θέση s της μνήμης, σε λίγο θα ζητηθεί πάλι το ίδιο στοιχείο
- Χώρου
 - Αν τώρα ζητηθεί το στοιχείο από τη θέση s της μνήμης τώρα, σε λίγο θα ζητηθούν και στοιχεία που βρίσκονται σε παραπλήσια θέση.

Είναι το στοιχείο που ζητάμε στη μνήμη όπου το αναζητάμε;
Ναι \Rightarrow *hit*
Όχι \Rightarrow *miss*(αστοχία)

Στόχος: να σχεδιάσουμε προγράμματα που εκμεταλλεύονται την ιεραρχία και μεγιστοποιούν τα hits στις ταχύτερες μνήμες

Σημαντικό: Η πρόσβαση ανά cache lines εκμεταλλεύεται την χωρική τοπικότητα και την προφόρτωση



TMHYP
ΤΕΧΝΗ ΠΡΟΒΛΕΨΕΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ & ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ

Πανεπιστήμιο Πατρών



Μοντέλο RAM/RASP

- **CPU + Mem**
- **κόστος εκτέλεσης**
 $T = O(\text{αριθμ. πράξεις ακυ})$
- «ικανοποιητικό» αν το βασικό κόστος είναι η αριθμητική.
- ... 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]

```
S = a(n);  
for I=n-1:-1:0  
    s = s. x + a(I);  
end;
```

- κόστος $\Omega = 2n$ πράξεις α.κ.υ.

Δεν λαμβάνεται υπόψη η επικοινωνία
Λίγες φορές αυτό δεν αποτελεί πρόβλημα

Συχνά όμως δίδονται λάθος εκτιμήσεις
υπάρχουν αλγόριθμοι για το ίδιο πρόβλημα με το ίδιο κόστος σε αριθμητικές πράξεις που έχουν πολύ διαφορετική επίδοση.

ΔΟΚΙΜΑΣΤΕ ΤΟΥΣ mulr.m και mulc.m
(ημερολόγιο).

Χρειαζόμαστε μοντέλο που να επιτρέπει
σχεδιασμό που αναδεικνύει και **διακρίνει τις**
διαφορετικές επιδόσεις.

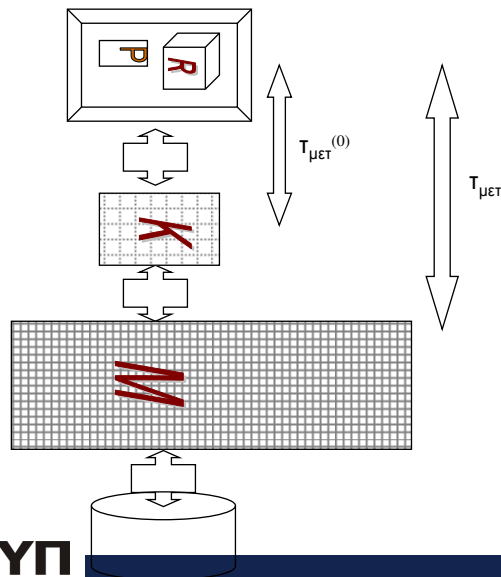


TMHYP
ΤΕΧΝΗ ΠΡΟΒΛΕΨΕΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ & ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ

Πανεπιστήμιο Πατρών



Απλό μοντέλο



Απλό μοντέλο

- Επεξεργαστής και αρχιτεκτονική **Load/Store**
- αρχείο καταχωρητών
- κρυφή μνήμη **K** θέσεων με **write back**
- κύρια μνήμη **M** θέσεων
- κόστος: **Load, Store**, πράξεις α.κ.υ.
- Κάθε πράξη αριθμ.κ.υ. στοιχίζει $\tau_{αρθ}$
- **load** από μνήμη στον επεξεργαστή σε χρόνο $\tau_{μετ}$
- **load** από κρυφή μνήμη στον επεξεργαστή σε $\tau_{μετ}^{(0)}$
- **store** από κρυφή μνήμη η επεξεργαστή σε $\tau_{μετ}$
- $\tau_{μετ}^{(0)} \approx 0$

Όταν αναφερόμαστε στο «γνωστό μοντέλο» θα εννοούμε το παραπάνω!

Μετρητές για το απλό μοντέλο

- Ω : αριθμός πράξεων α.κ.υ.
- Φ : αριθμός μεταφορών μεταξύ κύριας μνήμης και καταχωρητών ή κρυφής μνήμης
- Φ_{\min} : ελάχιστος αριθμός μεταφορών αν διαθέταμε απεριόριστη μνήμη σε όλα τα επίπεδα
- Εκτίμηση και Μείωση του Ω
 - αλλά πολλές φορές οι διαφορές είναι πολύ μικρές
 - ή έχουμε νέο αλγόριθμο
 - πολλές φορές το Ω είναι σταθερό!
- Εκτίμηση και Μείωση του $\Phi \rightarrow$ κύριος στόχος



TMHYP
ΤΕΧΝΗ ΜΕΤΑΦΟΡΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΔΙΟΤΥΠΩΝ & ΠΑΡΑΒΟΛΟΓΩΝ

Πανεπιστήμιο Πατρών



Ανάλυση κόστους

Το κόστος T εκτέλεσης μια υλοποίησης εκτιμάται ως

$$\begin{aligned} T &= T_{\alpha\rho\theta} + T_{\mu\epsilon\tau} \\ &= \tau_{\alpha\rho\theta}\Omega + \tau_{\mu\epsilon\tau}\Phi, \end{aligned}$$

$$T = T_{\alpha\rho\theta} \left(1 + \mu \frac{\tau_{\mu\epsilon\tau}}{\tau_{\alpha\rho\theta}} \right),$$

όπου $\mu := \Phi/\Omega$ είναι ο αριθμός μεταφορών ανά αριθμ. πράξη για την συγκεκριμένη υλοποίηση, και $\mu_{\min} = \Phi_{\min}/\Omega$ είναι ο ελάχιστος αριθμός μεταφορών ανά πράξη α.κ.υ. Έτσι

$$T = T_{\alpha\rho\theta} \left(1 + \mu \frac{\tau_{\mu\epsilon\tau}}{\tau_{\alpha\rho\theta}} \right) \geq T_{\alpha\rho\theta} \left(1 + \mu_{\min} \frac{\tau_{\mu\epsilon\tau}}{\tau_{\alpha\rho\theta}} \right)$$



TMHYP
ΤΕΧΝΗ ΜΕΤΑΦΟΡΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΡΟΔΙΟΤΥΠΩΝ & ΠΑΡΑΒΟΛΟΓΩΝ

Πανεπιστήμιο Πατρών



Κριτική υπολογιστικού μοντέλου

- Όλες οι πράξεις δεν έχουν ίδιο κόστος
«σχετικά flops»:
 - $+, -, *$ για 1 flop
 - $/, \text{sqrt}$ για 10-30 flops
 - \exp, \sin για 50+ flops
 - $+ \text{integer}$ δεν υπολογίζεται ($< 1 \text{ flop}$)
- Το κόστος μεταφορών πολύπλοκο εξαρτάται από θέση των στοιχείων
 - καταχωρητές 2 ns
 - L1 on-chip 4 ns
 - L2 on-chip 5 ns
 - L3 off-chip 30 ns
 - κύρια μνήμη 220 ns
 - το κόστος φόρτωσης (LOAD) δεν είναι το ίδιο με το κόστος εγγραφής (STORE)
- Επιτρέπεται επικάλυψη μεταφορών και πράξεων στον ίδιο κύκλο
- Η διαχείριση δεδομένων της cache γίνεται από το υλικό
- Τα στοιχεία μεταφέρονται ανά cache line



Πανεπιστήμιο Πατρών



Ιεραρχικό μοντέλο (ενός επιπέδου)

- CPU + Mem
 - κόστος εκτέλεσης
- $T \approx$ αριθμητικές πράξεις ακυ.και
πράξεις LOAD/STORE

```
LOAD a(0:n), x
s = a(n);
for I=n-1:-1:0
    s = s * x + a(I);
end;
STORE s
```

- κόστος $\Phi = \Phi_{\min} = n + 3 L/S$
- χρειάζεται $O(n)$ cache



Πανεπιστήμιο Πατρών



Ιεραρχικό μοντέλο (ενός επιπέδου)

- CPU + Mem
 - κόστος εκτέλεσης
- T = αριθμητικές πράξεις ακυ. και πράξεις LOAD/STORE

```
LOAD a(n), x
s = a(n);
for I=n-1:-1:0
    LOAD a(I)
    s = s . x + a(I);
end;
STORE s
```

- κόστος $\Phi = n+3 \text{ L/S}$
- χρειάζεται $O(1)$ cache



Πανεπιστήμιο Πατρών



Υλοποίηση για Φ_{min}

- Αν $\tau_{\text{MET}}^{(0)} \neq 0$ υποθέτουμε ότι διατίθεται όσος χώρος καταχωρητών χρειάζεται.
- Αν $\tau_{\text{MET}}^{(0)} = 0$ υποθέτουμε ότι διατίθεται όσος χώρος κρυφής μνήμης & καταχωρητών χρειάζεται.
- Το Φ_{min} θα ήταν εφικτό αν μπορούσαμε να υλοποιήσουμε το εξής:

Program PhiMin file

```
% φόρτωση των δεδομένων που αξιοποιούνται στον υπολογισμό
Load ***
statement_1; statement_2; ... statement_s
% ενδιάμεσα αποτελέσματα διατηρούνται σε καταχωρητές
% εγγραφή των αποτελεσμάτων στη μνήμη
Store ***
```



Πανεπιστήμιο Πατρών

