

Κατανεμημένα και Πολυεπεξεργαστικά Υπολογιστικά Συστήματα

Εισαγωγή

Μιχάλης Ψαράκης

Ιστοσελίδα

- Ιστοσελίδα μαθήματος στη νέα πλατφόρμα τηλεκπαίδευσης:
<http://gunet2.cs.unipi.gr/eclass>
 - Εγγραφείτε ως χρήστες
- Διαφάνειες
 - Μπορείτε να τις βρείτε στην πλατφόρμα gunet
- Επιπλέον βιβλιογραφία
 - Στη βιβλιοθήκη του Πανεπιστημίου Πειραιώς

Βαθμολογία μαθήματος

- Δεν θα υπάρχει γραπτή εξέταση στο μάθημα, αλλά 2 εργασίες
 - Μία εργασία στην ενότητα των πολυεπεξεργαστικών συστημάτων και μία εργασία στην ενότητα των κατανεμημένων συστημάτων
 - Η μία εργασία είναι υποχρεωτική και η άλλη προαιρετική
 - Οι φοιτητές μπορούν να επιλέξουν ποια εργασία θα κάνουν ως υποχρεωτική και ποια ως προαιρετική
 - Η υποχρεωτική εργασία θα μετρηθεί στο 70% του τελικού βαθμού και η προαιρετική στο 30%
- Εργασίες στην ενότητα των πολυεπεξεργαστικών συστημάτων
 - Προγραμματισμός embedded dual core processor – Zybo board
 - 1 ή 2 ατομικές εργασίες (προαιρετικές): 30%
 - Μπορείτε να τις παραδώσετε μόνο κατά την διάρκεια του εξαμήνου
 - 1 ομαδική εργασία: 40%
 - Ομάδες μέχρι 3 άτομα
 - Μπορείτε να την παραδώσετε και τον Σεπτέμβρη

Τι είναι τα κατανεμημένα και πολυεπεξεργαστικά συστήματα

- Multiprocessor system:
 - ένα υπολογιστικό σύστημα που περιέχει πολλαπλές υπολογιστικές μονάδες (επεξεργαστές), που συνεργάζονται και επικοινωνούν μεταξύ τους για να εκτελέσουν προγράμματα ταχύτερα
- Distributed system:
 - ένα υπολογιστικό σύστημα που περιέχει ανεξάρτητους υπολογιστές, που δίνουν την εντύπωση στον χρήστη ότι πρόκειται για ένα ενιαίο σύστημα

Γιατί τα εξετάζουμε μαζί

- Κοινά χαρακτηριστικά:
 - Πολλαπλές υπολογιστικές μονάδες
 - Συνεργάζονται και επικοινωνούν μεταξύ τους
 - Λειτουργούν ως ένα ενιαίο υπολογιστικό σύστημα
- Απαιτούν παράλληλο προγραμματισμό (parallel programming)

Σε τι διαφέρουν

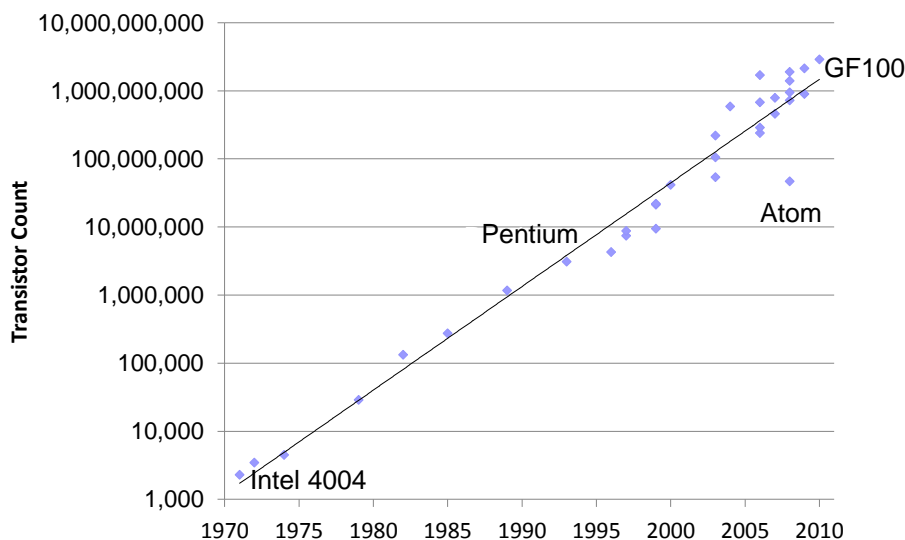
- Σε τι διαφέρουν τα κατανεμημένα από τα παράλληλα (ή πολυεπεξεργαστικά) συστήματα?
- Συχνά συναντάμε τους όρους "concurrent computing", "parallel computing", και "distributed computing"
 - Δεν υπάρχει ξεκάθαρη διάκριση μεταξύ τους
 - Το ίδιο σύστημα μπορεί να χαρακτηριστεί ως «παράλληλο» ή «κατανεμημένο»
- Μπορούμε να τα διακρίνουμε με κριτήριο την μνήμη τους
 - Στα παράλληλα συστήματα, όλοι οι επεξεργαστές έχουν πρόσβαση σε μια κοινόχρηστη μνήμη (shared memory)
 - Ανταλλάσσουν πληροφορίες μέσω της κοινόχρηστης μνήμης
 - Στα κατανεμημένα συστήματα, κάθε επεξεργαστής έχει την δική του ιδιωτική (private) ή κατανεμημένη μνήμη (distributed memory)
 - Ανταλλάσσουν πληροφορίες με πέρασμα μηνυμάτων (message passing) μεταξύ των επεξεργαστών

Προγραμματισμός κατανεμημένων και πολυεπεξεργαστικών συστημάτων

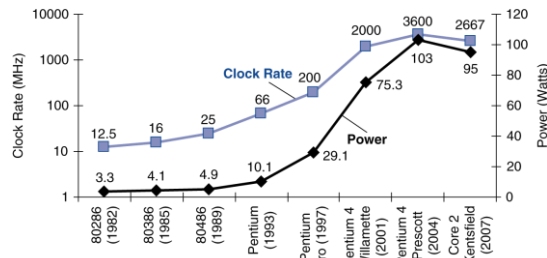
■ Κοινά προγραμματιστικά ζητήματα

- Αποθήκευση και προσπέλαση κοινόχρηστων δεδομένων
- Συγχρονισμός των διεργασιών
- Επιβάρυνση επικοινωνίας
- Αξιοπιστία και ανοχή ελαττωμάτων
- Κλιμάκωση

Νόμος του Moore



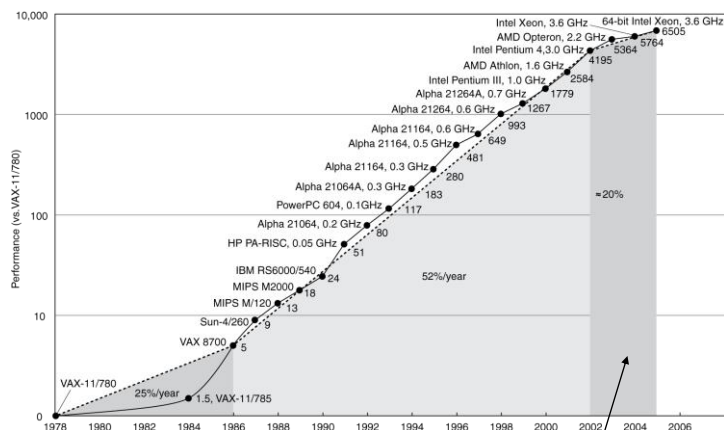
Γιατί πολυεπεξεργαστικά συστήματα



■ Κατανάλωση στην τεχνολογία CMOS

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

Απόδοση μονού επεξεργαστή



Περιορισμοί από κατανάλωση, παραλληλία σε επίπεδο εντολής, καθυστέρηση μνήμης

Η αύξηση της σειριακής απόδοσης έφτασε στο τέλος...

- **Δεν** μπορούμε να συνεχίσουμε να αυξάνουμε την συχνότητα των επεξεργαστών
 - Δεν θα φτιάξουμε 10 GHz chips...
- **Δεν** μπορούμε να συνεχίσουμε να αυξάνουμε την κατανάλωση ισχύος
 - Θα λιώσουν τα τσιπάκια...
- Μπορούμε όμως να συνεχίσουμε να αυξάνουμε την πυκνότητα των transistor
 - Συνέχεια του Moore's Law ...

Πως θα χρησιμοποιήσουμε τα παραπάνω τρανζίστορ;

- Instruction-level parallelism
 - out-of-order execution, speculation, ...
 - **Αύξηση της απόδοσης** όσο μας επιτρέπουν οι περιορισμοί κατανάλωσης
- Data-level parallelism
 - vector units, SIMD execution, ...
 - increasing ... SSE, AVX, Cell SPE, GPU
- Thread-level parallelism
 - increasing ... multithreading, multicore, manycore
 - Intel Core2, AMD Phenom, Sun Niagara, STI Cell, NVIDIA Fermi, ...

Νέος νόμος του Moore

- Οι υπολογιστές δεν θα γίνουν στο μέλλον γρηγορότεροι, απλά περισσότεροι...
- Πρέπει να ξανα-σκεφτούμε τους αλγορίθμους μας εάν μπορούν να παραλληλοποιηθούν
- Η παραλληλοποίηση των δεδομένων (data-parallel computing) είναι η καλύτερη λύση
 - Συνήθως θα έχεις περισσότερα δεδομένα από ότι πυρήνες
 - Σκέψου την υλοποίηση του προγράμματος με βάση τα δεδομένα

Πολυεπεξεργαστές

- Πολυπύρηννοι επεξεργαστές
 - Περισσότεροι πυρήνες σε κάθε τσιπ
- Απαιτείται παράλληλος προγραμματισμός
 - Σε σύγκριση με την παραλληλία σε επίπεδο εντολής (instruction level parallelism)
 - Το υλικό εκτελεί πολλαπλές εντολές ταυτόχρονα
 - ...χωρίς να επεμβαίνει ο προγραμματιστής
 - Δύσκολη δουλειά
 - Προγραμματισμός για βέλτιστη απόδοση
 - Μέχρι τώρα αυτό ήταν ευθύνη των σχεδιαστών του επεξεργαστή
 - Εξισορρόπηση φορτίου
 - Βελτιστοποίηση της επικοινωνίας και του συγχρονισμού

Ενότητα: πολυεπεξεργαστικά υπολογιστικά συστήμα

- Θεωρητικό μέρος:
 - Πώς επιτυγχάνεται στους μονούς επεξεργαστές η παραλληλία στην εκτέλεση (instruction level parallelism)
 - π.χ. dynamic scheduling, superscalar, multithreading
 - Ποια είναι τα βασικά σχεδιαστικά ζητήματα στις πολυπύρηνες αρχιτεκτονικές
 - π.χ. data consistency
 - Ποιες είναι οι τελευταίες εξελίξεις στην περιοχή των πολυπύρηνων επεξεργαστών
 - αρχιτεκτονική GPUs (graphics processing units)
- Πρακτική εξάσκηση:
 - Πολυπύρηνος (dual core) ενσωματωμένος επεξεργαστής
 - Ανάπτυξη λογισμικού για ενσωματωμένη πλατφόρμα

Περιεχόμενο ενότητας

- Πολυνημάτωση (multithreading), Πολυεπεξεργαστές (multiprocessors) και Graphics Processing Units (GPUs)
 - Παραλληλία επιπέδου νήματος (thread level parallelism, TLP)
 - Πολυεπεξεργαστές κοινόχρηστης μνήμης (shared-memory, SMPs)
 - Πολυεπεξεργαστές με μεταβίβαση μηνυμάτων (message passing)
 - Πρωτόκολλα συνοχής (coherence protocols)
 - Μονάδες Επεξεργασίας Γραφικών (graphics processing units, GPUs)
 - Αρχιτεκτονικές GPU
 - Βιβλιογραφία:
 - «Αρχιτεκτονική Υπολογιστών», 4^η έκδοση, Κεφ. 3, 4
 - «Οργάνωση και Σχεδίαση Υπολογιστών», 4^η έκδοση, Κεφ. 7
- Ανάπτυξη εφαρμογής σε dual core System-on-Chip (SoC)
 - Αναπτυξιακή πλατφόρμα Xilinx Zybo board
 - Εκμάθηση πλατφόρμας (εργαλεία Vivado & SDK)
 - Xilinx workshops
 - Embedded System Design Flow on Zynq using Vivado