Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Παράλληλος Προγραμματισμός
2024-25

Ιεραρχίες Μνήμης

(και ο ρόλος τους στην απόδοση της παράλληλης επεξεργασίας)

https://mixstef.github.io/courses/parprog/

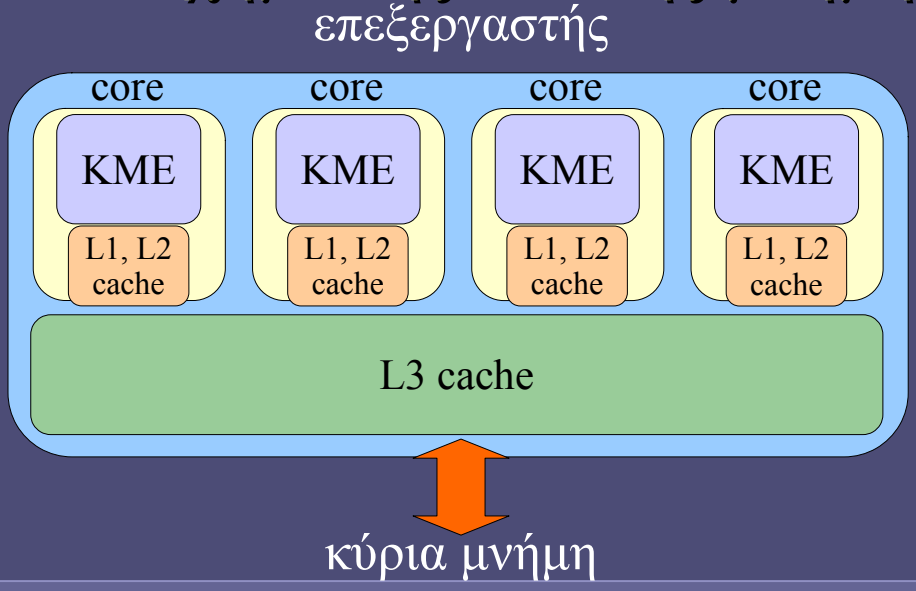
Μ.Στεφανιδάκης

Ιεραρχίες Μνήμης

- **Πολλαπλά επίπεδα κρυφής μνήμης (cache memory)**
  - Μεταξύ ΚΜΕ και κύριας μνήμης
  - Προσπάθεια γεφύρωσης της διαφοράς απόδοσης κατά την προσπέλαση δεδομένων μεταξύ επεξεργαστικών στοιχείων και κύριας μνήμης σε ένα υπολογιστικό σύστημα
  - Η προσπέλαση της κύριας μνήμης είναι τυπικά της τάξης του 100x αργότερη από την προσπέλαση δεδομένων σε καταχωρητές

Κρυφές μνήμες

- Γρηγορότερη προσπέλαση δεδομένων
  - L1: τυπικά 1-4 κύκλοι ρολογιού
  - L2-L3: τυπικά 10-20 κύκλοι ρολογιού
  - Κύρια μνήμη: τυπικά 100 κύκλοι ρολογιού
- Συμμετοχή στο σχήμα της «κοινής μνήμης»



Κρυφές μνήμες: Χωρητικότητα

- Κάθε επίπεδο αποθηκεύει μικρό μέρος του αμέσως κατώτερου
  - L1: μερικές δεκάδες KB
    - Συνήθως χωρισμένα σε κρυφή μνήμη εντολών (L1I) και κρυφή μνήμη δεδομένων (L1D)
  - L2: μερικά MB
  - L3: μερικές δεκάδες MB
- Η ταχύτητα προσπέλασης μειώνεται με την αύξηση του μεγέθους
  - Αλλά είναι σε κάθε περίπτωση γρηγορότερη από την κύρια μνήμη
  - Η μείωση της ταχύτητας δεν επηρεάζει την ΚΜΕ – είναι συνδεδεμένη με την ταχύτερη κρυφή μνήμη (L1)

Γιατί (πότε) επιτυγχάνει η ιεραρχία μνήμης

- Οι κρυφές μνήμες χωρούν πολύ μικρό μέρος των δεδομένων της κύριας μνήμης
  - Μικρότερος χώρος όσο πλησιάζουμε τον επεξεργαστή
  - Διαφορετικά δεδομένα μοιράζονται τις ίδιες θέσεις σε κάθε επίπεδο της κρυφής μνήμης
    - Ανταγωνίζονται και αντικαθιστούν άλλα δεδομένα
  - Κάποια δεδομένα δεν θα είναι διαθέσιμα στις κρυφές μνήμες όταν τα ζητήσει κάποια ΚΜΕ
- Για να είναι επιτυχημένη η ιεραρχία μνήμης θα πρέπει η ΚΜΕ να βρίσκει συχνά τα δεδομένα που χρειάζεται στις κρυφές μνήμες
  - Ποιος το εγγυάται αυτό;

Τοπικότητα κατά την επεξεργασία

- Χρονική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστεί ξανά στο άμεσο μέλλον
  - Ό,τι χρησιμοποιήθηκε θα ζητηθεί ξανά πολύ σύντομα
    - Συνεπώς θα βρίσκεται ακόμα στην κρυφή μνήμη με μεγάλη πιθανότητα
- Χωρική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστούν και οι γειτονικές θέσεις στο άμεσο μέλλον
    - Όταν δεν αρκεί η χρονική τοπικότητα
  - Για να εκμεταλλευτούμε την χωρική τοπικότητα πρέπει να μετακινούμε μπλοκ δεδομένων από/προς την κύρια μνήμη

Μεταφορές σε μπλοκ

- Όλες οι μεταφορές μεταξύ κύριας μνήμης και κρυφών μνημών γίνεται σε μπλοκ λέξεων
  - Τυπικό μέγεθος 64 bytes (cache line)
  - Μόνο οι ΚΜΕ μπορούν να διαβάζουν/γράφουν μεμονωμένες λέξεις ή bytes
- Αντικατάσταση μπλοκ
  - Όταν ένα νέο μπλοκ εισάγεται σε μία κρυφή μνήμη
  - Θα πρέπει να επιλεγεί ποιο υπάρχον μπλοκ θα αντικατασταθεί (victim)
  - Η επιλογή γίνεται ανάλογα με την πολιτική τοποθέτησης της κρυφής μνήμης

Πολιτικές τοποθέτησης

- Direct mapped cache
  - Επιλογή θέσης από ένα μέρος των bits της διεύθυνσης του μπλοκ
    - Γρήγορος υπολογισμός και αναζήτηση
    - «Τυφλή» αντικατάσταση χρήσιμων μπλοκ
- Fully associative cache
  - Ένα μπλοκ μπορεί να τοποθετηθεί παντού
    - Least Recently Used
    - Χρονοβόρα αναζήτηση
- N-way set-associative cache
  - Τοποθέτηση σε μία από N (4, 8, ..) θέσεις (σετ)
    - Συμβιβασμός μεταξύ των δύο προηγούμενων

Λειτουργία κρυφής μνήμης

- Διαφανής λειτουργία – αυτόματη διαχείριση από υλικό
  - Οι ΚΜΕ εκτελούν αναγνώσεις (reads) και εγγραφές (writes) λέξεων δεδομένων από/προς το κοντινότερο επίπεδο κρυφής μνήμης (L1)
  - Αν βρεθεί το μπλοκ που περιέχει τη λέξη (hit)
    - Κατά την ανάγνωση, η λέξη επιστρέφεται στην ΚΜΕ
    - Κατά την εγγραφή συνήθως η λέξη εγγράφεται μόνο στην κρυφή μνήμη (πολιτική write-back)
  - Αν δεν βρεθεί (miss) ζητείται από το χαμηλότερο επίπεδο το μπλοκ που περιέχει τη λέξη
    - Συνήθως αυτό συμβαίνει και κατά την εγγραφή (πολιτική write allocate)
    - Σε αντικατάσταση ενός μπλοκ που έχει υποστεί αλλαγές (νέα δεδομένα), πρέπει πρώτα να γραφτεί το μπλοκ στο χαμηλότερο επίπεδο

Λειτουργία κρυφής μνήμης

- Prefetching - «προληπτική» μεταφορά δεδομένων
  - Μπορεί να εκτελείται αυτόματα από το υλικό
    - Όταν εντοπίσει ένα σχέδιο αλληλουχίας προσπελάσεων
  - Ή από το λογισμικό με ειδικές εντολές prefetch
- Non-temporal stores
  - Σε πολλές εφαρμογές υπολογισμού υψηλής απόδοσης τύπου streaming τα αποτελέσματα (εγγραφή στη μνήμη) δεν θα χρησιμοποιηθούν ξανά
    - Πρέπει να αποφευχθεί η «μόλυνση» της κρυφής μνήμης με μπλοκ δεδομένων που δεν θα χρειαστούν πάλι
  - Υπάρχουν εντολές εγγραφής που παρακάμπτουν την κρυφή μνήμη

Επίδραση στην παράλληλη επεξεργασία

- Πολλά υπολογιστικά προβλήματα εξαρτώνται από τον ρυθμό μεταφοράς δεδομένων από/προς τη μνήμη
  - Memory-bound problems
  - Μικρές ευκαιρίες βελτίωσης μέσω παραλληλισμού
- Όταν τα δεδομένα χωρούν στις κρυφές μνήμες
  - Η προσπέλαση των δεδομένων πρέπει να εκμεταλλεύεται την προσφερόμενη τοπικότητα έτσι ώστε να επιτυγχάνεται αύξηση της απόδοσης
- False sharing
  - Όταν πολλαπλά threads εκτέλεσης ανταγωνίζονται για ένα μπλοκ δεδομένων χωρίς να προσπελαύνουν την ίδια λέξη

Επίσης: η επίδραση της εικονικής μνήμης

- Virtual memory
  - Σχήμα μετάφρασης των λογικών διευθύνσεων μνήμης των εφαρμογών σε φυσικές διευθύνσεις μνήμης
  - Οργάνωση σε σελίδες (pages)
- Συνεργασία υλικού-λογισμικού
  - Πίνακες σελίδων (διαχείριση από λειτουργικό σύστημα)
    - Διαφορετικοί πίνακες για κάθε εφαρμογή
  - Οι ΚΜΕ μεταφράζουν λογικές διευθύνσεις μνήμης σε φυσικές σε κάθε προσπέλαση
    - Δηλαδή... περισσότερες προσπελάσεις (για να πάρει η ΚΜΕ την πληροφορία από τους πίνακες σελίδων) για κάθε μία προσπέλαση δεδομένων;!;!;

Η επίδραση της εικονικής μνήμης

- Πώς αποφεύγεται η συνεχής προσπέλαση των πινάκων σελίδων
  - Μηχανισμός page table walking
- Translation Lookaside Buffer (TLB)
  - Μικρή «κρυφή μνήμη» με πρόσφατες μεταφράσεις (π.χ. 128 θέσεις)
  - Πολύ περιορισμένος πόρος
  - Εάν τα δεδομένα ανήκουν σε πολλαπλές διαφορετικές σελίδες πιθανόν να υπάρχουν αυξημένα TLB misses
    - Το μέγεθος της σελίδας συνήθως είναι μικρό (4KB-2MB)

Βιβλιογραφία

- Michael McCool, James Reinders, and Arch Robison. 2012. *Structured Parallel Programming: Patterns for Efficient Computation* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Georg Hager and Gerhard Wellein. 2010. Introduction to High Performance Computing for Scientists and Engineers (1st ed.). CRC Press, Inc., Boca Raton, FL, USA.