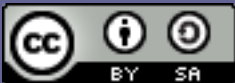


Work – Span

(και οι περιπτώσεις των λειτουργιών map και reduce)

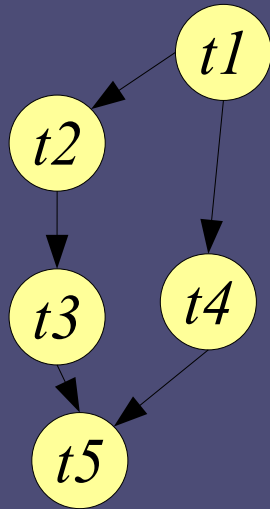
<https://mixstef.github.io/courses/parprog/>

Μ.Στεφανιδάκης



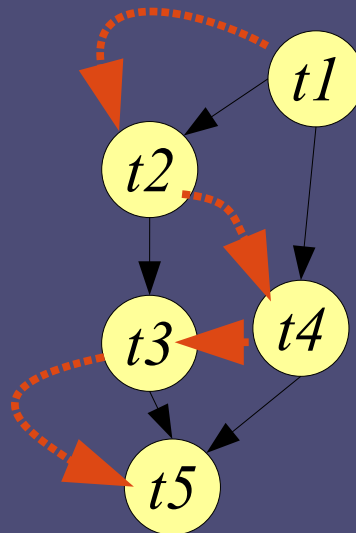
Ορισμός Work και Span

- Η παράλληλη εκτέλεση γίνεται με την ολοκλήρωση tasks
 - Ακολουθώντας τη ροή των αλληλεξαρτήσεων των δεδομένων
 - Ένας κατευθυνόμενος μη κυκλικός γράφος (DAG)



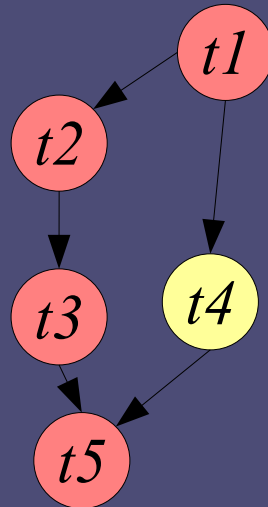
Ορισμός Work και Span

- T_1 είναι ο χρόνος σειριακής εκτέλεσης (work)
 - Μια οποιαδήποτε έγκυρη σειριοποίηση της δουλειάς που πρέπει να γίνει



Ορισμός Work και Span

- T_{∞} είναι ο χρόνος σε ένα ιδανικά παράλληλο σύστημα (span)
 - Διαθέσιμα άπειρα επεξεργαστικά στοιχεία
 - Η καλύτερη περίπτωση παραλληλίας
 - Το όριο είναι το κρίσιμο μονοπάτι tasks (critical path)



Ανάλυση Work – Span

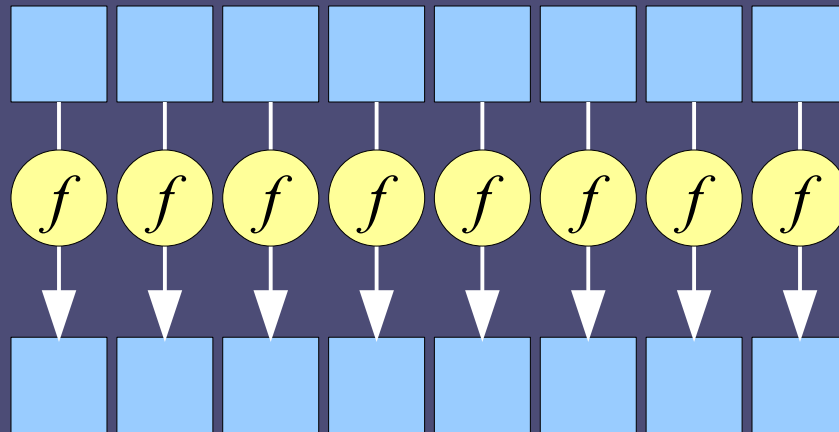
Αν T_p είναι ο χρόνος σε σύστημα με P επεξεργαστικά στοιχεία, τότε

$$T_p = O(T_1 / P + T_\infty)$$

- Το T_∞ εμποδίζει την επεκτασιμότητα
- Η αύξηση του T_1 επιβαρύνει την απόδοση
- Συνεπώς η σχεδίαση των παράλληλων αλγορίθμων θα πρέπει να αποσκοπεί στην μείωση του T_∞ (span)
 - Αποφεύγοντας την υπέρμετρη αύξηση του T_1 (work), εκτός κι αν αυτό μειώνει δραστικά το span

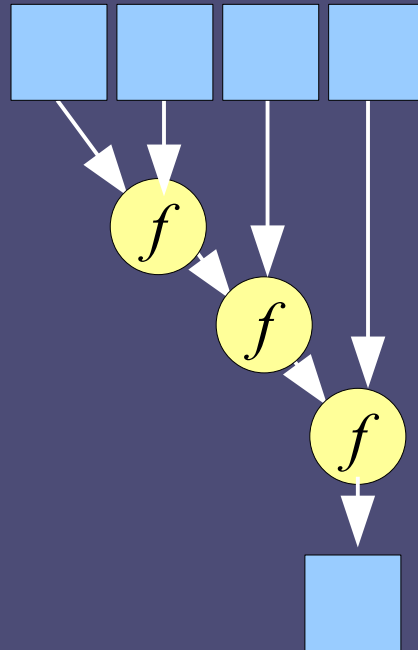
Η λειτουργία map ξανά

- Εφαρμογή μιας συνάρτησης σε κάθε στοιχείο μιας ακολουθίας δεδομένων
 - $Work = O(n)$
 - $Span = O(1)$ (αν η f έχει σταθερό κόστος)



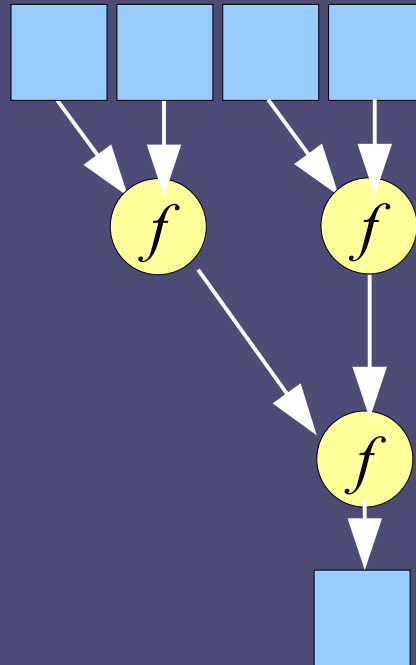
Η λειτουργία reduce

- Συνδυάζει όλα τα στοιχεία μιας συλλογής (collection) σε ένα μοναδικό στοιχείο μέσω τελεστή f



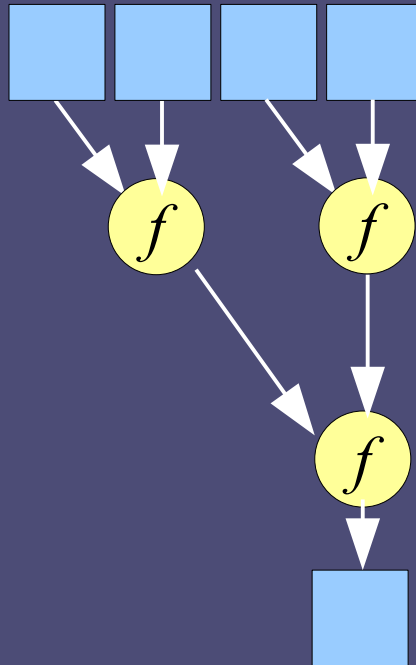
Παραλληλοποίηση της reduce

- Δεν είναι πάντα δυνατή
 - Θα πρέπει ο τελεστής f να είναι προσεταιριστικός
 - Όταν $((x1 \circ x2) \circ x3) \circ x4 = (x1 \circ x2) \circ (x3 \circ x4)$



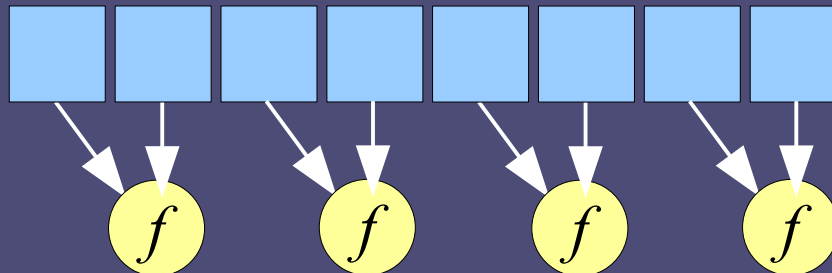
Reduce: Work και Span

- Στην ιδανική περίπτωση
 - $Work = O(n)$ – όσο και η σειριακή εκδοχή
 - $Span = O(\log n)$

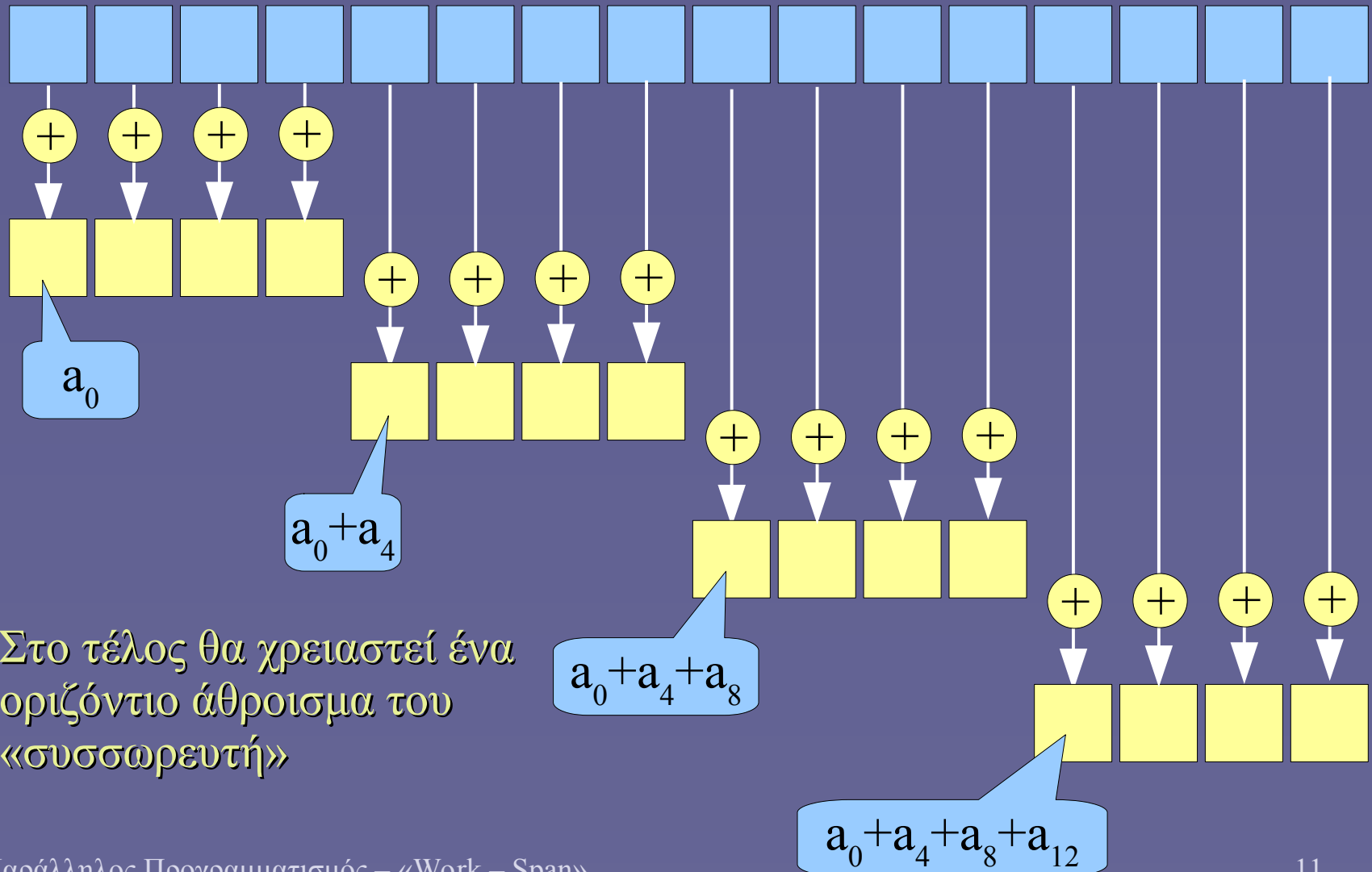


Reduce και εντολές SSE/AVX

- Για την άμεση υλοποίηση θα ήταν απαραίτητες εντολές με μη-κάθετες πράξεις
 - «Οριζόντιες» λειτουργίες
 - Διάσχιση κάθετων λωρίδων (cross-lane operations)
 - Μη αποδοτικές πράξεις σε σχέση με τις «κάθετες»
 - Πολλές εντολές έχουν περιορισμούς στη διάσχιση λωρίδων
 - Χρειάζεται υλοποίηση με όσο το δυνατόν περισσότερες «κάθετες» πράξεις
 - Ιδίως στα κρίσιμα για την απόδοση σημεία του κώδικα



Παράδειγμα reduction: Άθροισμα πίνακα



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.
- Intel® Intrinsic Guide (<https://software.intel.com/sites/landingpage/IntrinsicGuide/>)