

## Ανάλυση Work-Span

(και οι περιπτώσεις των λειτουργιών map και reduce)

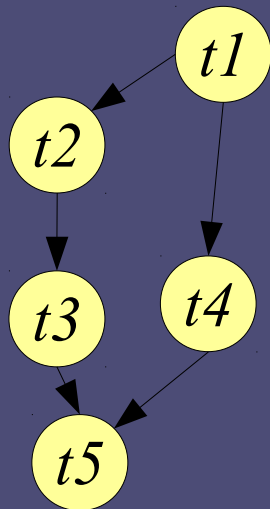
<http://mixstef.github.io/courses/pms-parcomp/>

Μ.Στεφανιδάκης



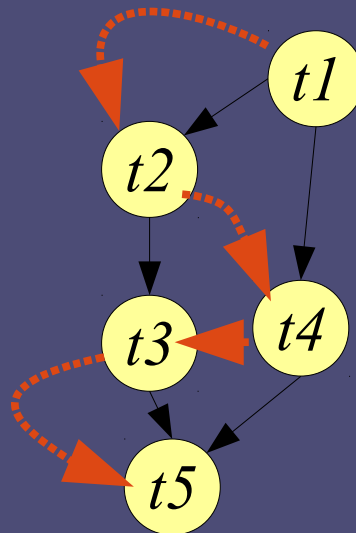
# Ανάλυση Work – Span

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



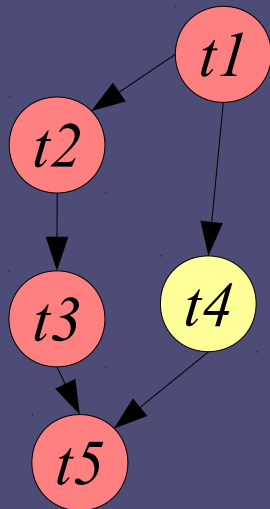
# Ανάλυση Work – Span

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



# Ανάλυση Work – Span

- $T_{\infty}$  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

- $T_p$  είναι ο χρόνος σε σύστημα με  $P$  επεξεργαστικά στοιχεία

$$T_p \leq \frac{T_1}{P} \qquad T_p \leq T_\infty$$

$$T_p \leq (T_1 - T_\infty) / P + T_\infty \quad (\text{Brent's Lemma})$$

- Ασυμπτωτικά

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

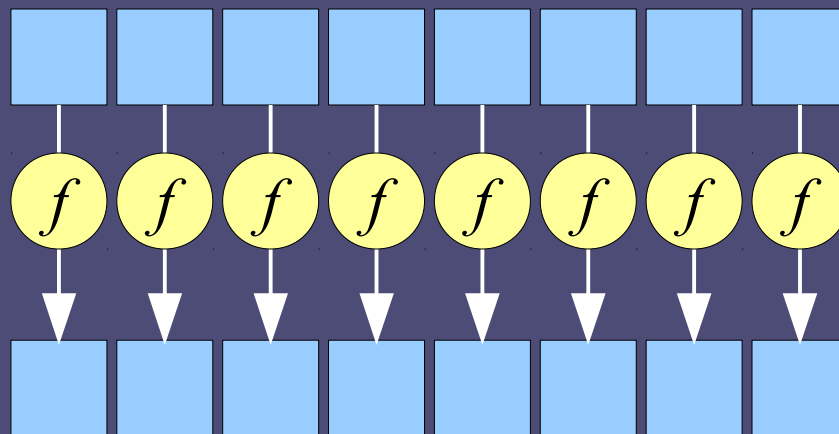
# Ανάλυση Work – Span

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

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

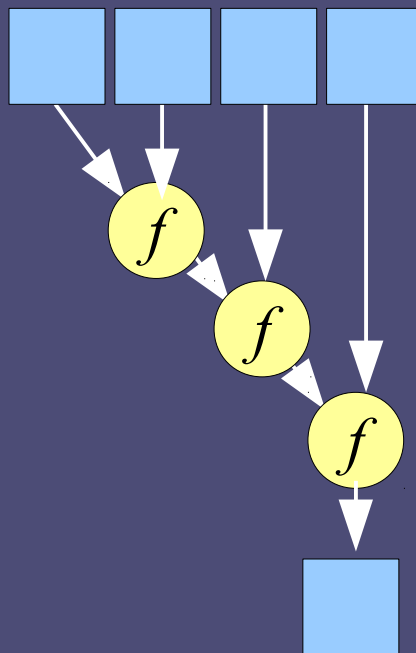
# Η λειτουργία map ξανά

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



# Η λειτουργία reduce

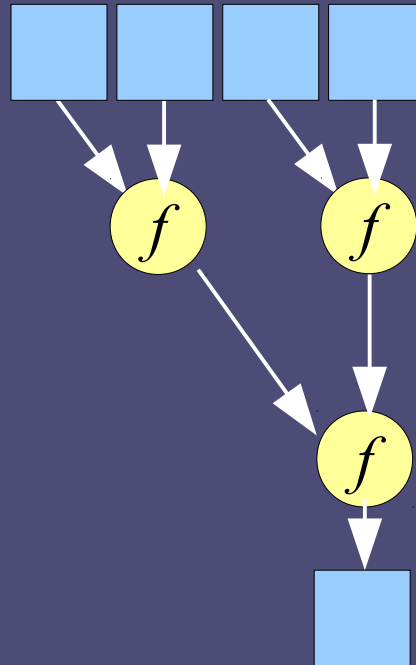
- Συνδυάζει όλα τα στοιχεία μιας συλλογής (collection) σε ένα μοναδικό στοιχείο μέσω τελεστή  $f$





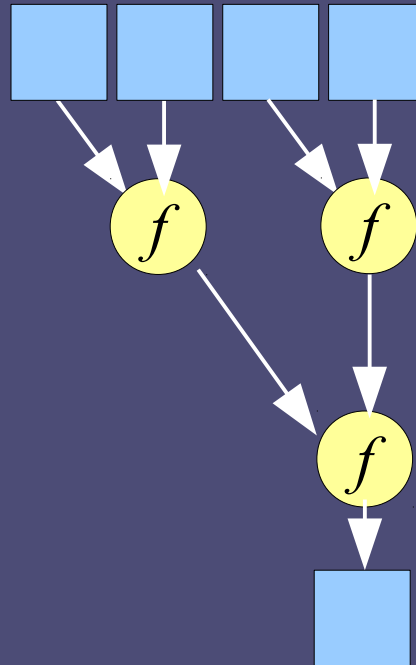
# Παραλληλοποίηση της reduce

- 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
  - $Span = O(\log n)$



# 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.