



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Ακ. έτος 2017-2018, 9ο Εξάμηνο ΗΜ&ΜΥ

*N. Κοζύρης, Κ. Δόκα*

Χειμερινό Εξάμηνο

**Χρόνος εξέτασης: 2.5 ώρες**

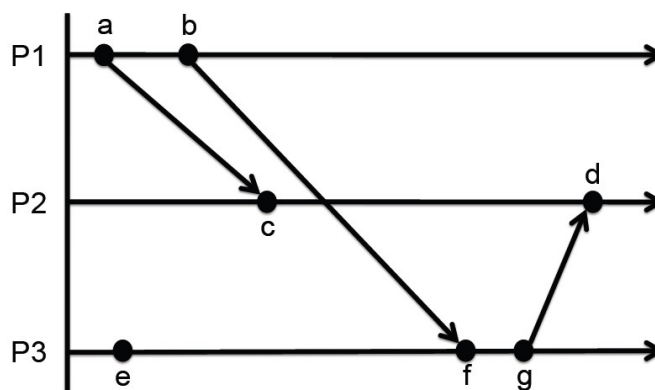
### Θέμα 1 (20%)

Θεωρήστε τις διεργασίες P1, P2 και P3 οι οποίες παράγουν τα παρακάτω γεγονότα {a..g}. Τα βέλη αντιπροσωπεύουν την ανταλλαγή μηνυμάτων ανάμεσα στις διεργασίες.

(α) Καταγράψτε τις χρονοσφραγίδες Lamport για όλα τα γεγονότα. Θεωρήστε ότι όλες οι διεργασίες ξεκινούν με μηδενικές χρονοσφραγίδες. (5%)

(β) Παρατηρήστε τα γεγονότα c και g. Μπορείτε να αποφανθείτε αν συμβαίνουν ταυτόχρονα ή αν το ένα συμβαίνει πριν από το άλλο (happened-before) χρησιμοποιώντας μόνο τις χρονοσφραγίδες Lamport; Δώστε σύντομη εξήγηση. (5%)

(γ) Τώρα χρησιμοποιήστε διανυσματικά ρολόγια και καταγράψτε τις χρονοσφραγίδες για όλα τα γεγονότα. Θεωρήστε όλα τα ρολόγια αρχικοποιημένα στο (0, 0, 0). Μπορούμε τώρα να απαντήσουμε με σιγουριά αν τα γεγονότα c και g συμβαίνουν ταυτόχρονα; (10%)



## Θέμα 2 (30%)

Έχετε ένα dataset του Netflix που καταγράφει τις ταινίες που παρακολουθούν οι χρήστες του. Κάθε εγγραφή έχει τα εξής στοιχεία: {userID, movieID, timestamp}: Ο χρήστης userID είδε την ταινία movieID τη χρονική στιγμή timestamp. Θέλετε να φτιάξετε μια υπηρεσία που καθώς ο χρήστης ψάχνει ταινίες θα του προτείνει κι άλλες παρόμοιες ("Οι χρήστες που έχουν δει αυτήν την ταινία έχουν δει επίσης..."). Για να γίνει αυτό θα πρέπει να υπολογιστεί η ομοιότητα κάθε ταινίας με κάθε άλλη διαθέσιμη ταινία. Ως μετρική ομοιότητας χρησιμοποιήστε το Jaccard similarity

$$Jaccard(X, Y) = \frac{n}{n_x + n_y - n}$$

όπου  $n$  το πλήθος των χρηστών που έχουν παρακολουθήσει και την ταινία  $X$  και την ταινία  $Y$

$n_x$  το πλήθος των χρηστών που έχουν παρακολουθήσει την ταινία  $X$  και

$n_y$  το πλήθος των χρηστών που έχουν παρακολουθήσει την ταινία  $Y$ .

Χρησιμοποιήστε MapReduce για να υπολογίσετε το Jaccard similarity για κάθε ζεύγος ταινιών. Θα χρειαστεί να χρησιμοποιήσετε περισσότερες από μια MapReduce εργασίες όπου η έξοδος της μιας θα είναι η είσοδος της επόμενης.

Εξηγήστε όλες τις λειτουργίες λεκτικά και δώστε ψευδοκώδικα. Ο ψευδοκώδικας θα πρέπει να περιλαμβάνει για κάθε εργασία MapReduce τις συναρτήσεις map και reduce στην παρακάτω μορφή:

MAP(key1, value1):	REDUCE(key2, list(value2)):
//επεξεργασία για κάθε	//επεξεργασία για
<key1, value1>	κάθε //<key2, list(value2)>
emit(key2, value2)	emit(key3, value3)

## Θέμα 3 (30%)

(α) Στην πιο απλή μορφή των συστημάτων DHT ο κάθε κόμβος συνδέεται απλώς με τον προηγούμενο και τον επόμενο του (στον λογικό δακτύλιο των ids). Τι κόστος έχει μια αναζήτηση (lookup) στην χειρότερη περίπτωση; Τι πρόβλημα μπορεί να δημιουργηθεί όσο το δίκτυο μεγαλώνει; (5%)

(β) Το Chord λύνει το παραπάνω πρόβλημα με τη χρήση των finger tables. Ποιο είναι το μέσο κόστος αναζήτησης στην περίπτωση αυτή; (5%)

(γ) Θεωρήστε ένα P2P δίκτυο όπου κάθε κόμβος συνδέεται μόνο με τον κόμβο με id 0. Πώς θα γινόταν σε αυτήν την περίπτωση η δρομολόγηση των ερωτημάτων; Ποιο θα ήταν το κόστος αναζήτησης; (10%)

(δ) Θεωρήστε ένα DHT που κατασκευάζεται με τον ακόλουθο τρόπο: Αν  $n$  κόμβοι συμμετέχουν στο δίκτυο αυτήν τη στιγμή, αυτοί είναι αριθμημένοι από το 0 έως το  $n-1$  και κάθε κόμβος γνωρίζει το αναγνωριστικό (id) του. Τα αναγνωριστικά των αντικειμένων που εισάγονται στους κόμβους είναι επίσης φυσικοί αριθμοί. Ως συνάρτηση κατακερματισμού (hash function) χρησιμοποιούμε την

$$h(k) = k \bmod n$$

έτσι ώστε το αντικείμενο με id  $k$  να ανατίθεται στον κόμβο με id  $h(k)$ .

Είναι αυτή η συνάρτηση καλή βάση για ένα DHT; Δώστε (τουλάχιστον) 2 λόγους. (10%)

## Θέμα 4 (20%)

A. Παραδοχές και συμβολισμοί: Στα παρακάτω  $R_i(X)$  σημαίνει ότι η δοσοληψία  $T_i$  διαβάζει το δεδομένο  $X$ , ενώ  $W_i(X)$  σημαίνει ότι η δοσοληψία  $T_i$  γράφει το δεδομένο  $X$ . Δίνονται τα παρακάτω χρονοπρογράμματα (schedules):

S1:  $R_2(Z), R_2(Y), W_2(Y), R_3(Y), R_3(Z), R_1(X), W_1(X), W_3(Y), W_3(Z), R_2(X), R_1(Y), W_1(Y), W_2(X)$

S2:  $R_3(Y), R_3(Z), R_1(X), W_1(X), W_3(Y), W_3(Z), R_2(Z), R_1(Y), W_1(Y), R_2(Y), W_2(Y), R_2(X), W_2(X)$

Για καθένα από τα χρονοπρογράμματα αποφανθείτε αν είναι σειριοποιήσιμο σε σχέση με τις συγκρούσεις (conflict-serializable) και δικαιολογήστε την απάντησή σας.

B. Ένας server διαχειρίζεται τα αντικείμενα  $i, j, k$  και παρέχει 2 λειτουργίες στους clients του:

$read(i)$ , που επιστρέφει την τιμή του  $i$

$write(i, value)$ , που δίνει στο  $i$  την τιμή  $value$

Αν έχουμε τα transactions  $T$  και  $U$  για τα οποία ισχύει:

$T: x=read(j); y=read(i); write(j, 44); write(i,33);$

$U: x=read(k); write(i, 55); y=read(j); write(k,66);$

Δώστε ένα interleaving των δύο transactions ώστε το χρονοπρόγραμμα που προκύπτει:

(i) να είναι ισοδύναμο με ένα σειριακό,

(ii) να ακολουθεί αυστηρή εκτέλεση των transactions,

(iii) να μην ακολουθεί αυστηρή εκτέλεση των transactions αλλά να μην κινδυνεύει από cascading aborts,

(iv) να κινδυνεύει από cascading aborts.