

Ηλίας Λεοντιάδης
Στέφανος Πέτσιος

Κατανεμημένα συστήματα
6^ο σετ ασκήσεων

Πανεπιστήμιο Ιωαννίνων
19 Μαΐου 2003

Άσκηση 1

Ο χρόνος σε κάποιο σύστημα χρησιμοποιείται ως επί των πλείστον για να μπορούμε να αναπαριστούμε λογικά μία σειρά γεγονότων. Σε πολλές εφαρμογές υπάρχει το ενδεχόμενο μία διόρθωση του ρολογιού προς τα πίσω να μας οδηγήσει σε πλασματικές – εσφαλμένες καταστάσεις αφού υποθέτουμε ότι ο χρόνος προχωράει πάντα μπροστά. Για παράδειγμα εάν κάποιος κάνει compile μιας εφαρμογής και μετά το ρολόι γυρίσει πίσω τότε αν αλλάξει κάτι και κάνει πάλι makefile το εκτελέσιμο δεν θα ανανεωθεί (με αποτέλεσμα να χαθεί το update).

Μπορούμε να μειώσουμε την ταχύτητα του ρολογιού στο μισό για τα επόμενα 8 «κανονικά» δευτερόλεπτα. Έτσι ο χρόνος που θα περάσει για το σύστημα είναι 4 δευτερόλεπτα με αποτέλεσμα να «χάσει» τα τέσσερα δευτερόλεπτα που πήγαινε μπροστά.

Άσκηση 2

Θα επιλέξουμε τον χρόνο με το μικρότερο *round trip*. Και αυτό είναι μικρότερη η διακύμανση που μπορούμε να έχουμε. Για παράδειγμα αν το round-trip είναι 20ms το μήνυμα από τον time server μπορεί να έφυγε από 1 έως 20 ms πριν.

Στην συγκεκριμένη περίπτωση διαλέγουμε το ζεύγος 20 10:54:28.342 και θα προσθέσουμε το μισό. Δηλαδή υποθέτουμε ότι το μήνυμα στον μισό χρόνο ταξίδεψε ως τον time server και στον υπόλοιπο μισό επέστρεψε.

Δηλαδή θέτουμε την τιμή 10:54:28.352 και έχουμε απόκλιση $\pm 10ms$

Στην περίπτωση που είναι γνωστό ότι έχουμε τουλάχιστον 8ms καθυστέρηση το μόνο που αλλάζει είναι η απόκλιση. Γιατί ξέρουμε ότι το λιγότερο που χρειάστηκε το μήνυμα για να έρθει από τον server είναι 8ms και το περισσότερο $20 - 8 = 12$.

Άρα συνολικά θέτουμε την τιμή 10:54:28.352 και έχουμε απόκλιση $\pm 2ms$

Άσκηση 3

Έστω ότι $e \rightarrow e' \rightarrow L(e) > L(e')$.

Έστω ότι $e \rightarrow a \rightarrow e'$ (για να καλύψουμε το ενδεχόμενο αλυσιδωτών μηνυμάτων). Με βάση τον κανόνα του Lamport το νέο time stamp του a θα είναι ή $L(e)+1$ ή $L(a) > L(e)$.

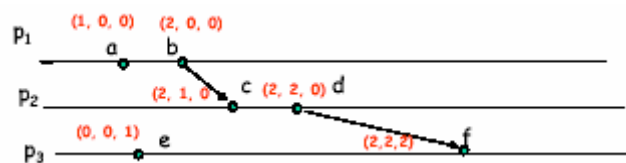
Αντίστοιχα το $L(e') > L(a)$ ή $L(e') = L(a)+1$.

Συνολικά $L(a) > L(e)$ και $L(e') > L(a)$.

Άρα $L(e) < L(e')$.

Άσκηση 4

A)



Παρατηρούμε ότι η C έχει $V_2[2] = 1$ και $V_2[1] = 2$. Άρα δεν ισχύει $V_2[1] \leq V_2[2]$.

Το πιο λογικό είναι η υπόθεση

$V_j[i] \leq V_i[i]$.

B) το B υποθέτει την απόδειξη της ορθότητας του A...

Άσκηση 5

Μόνο εάν χρησιμοποιήσουμε multicast acknowledgement μπορούμε να εξασφαλίσουμε ότι τελικά όλες οι διεργασίες θα έχουν το ίδιο αντίγραφο της τοπικής ουράς ταξινομημένο ως προς τα timestamps. Ο αλγόριθμος Lamport και η εγγύηση του ότι δύο διαδοχικά μηνύματα από τον ίδιο αποστολέα φτάνουν με την ίδια σειρά μας εγγυούνται την totally order multicast. Χωρίς τα acknowledgements δεν είναι δυνατόν να διατάξουμε τα μηνύματα (αφού θα επεξεργάζονται με την σειρά που έρχονται).

Άσκηση 6

Θα μπορούσε κάποιος που ξεκίνησε διαδικασία ψηφοφορίας όταν λάβει μήνυμα εκλογής από άλλον κόμβο (δεύτερη ταυτόχρονη ψηφοφορία) να κάνει έναν έλεγχο αν το id του είναι για παράδειγμα μικρότερο από το id του άλλου κόμβου που ξεκίνησε την ψηφοφορία και αν είναι να μην προωθεί το μήνυμα.

Άσκηση 7

Ο συνολικός χρόνος $t_{total} = synchronization\ delay + ελάχιστος\ χρόνος\ που\ μένει\ στην\ κρίσιμη\ περιοχή\ κάποια\ διεργασία$. Όσο μεγαλύτερος είναι αυτός ο χρόνος τόσο μικρότερο είναι το throughput.

Άρα $throughput = 1 / t_{total}$

Άσκηση 8

Σε ένα καταναμημένο σύστημα το two phase locking σημαίνει ότι όταν κάποια διεργασία αφήσει ένα lock σε κάποιον server καμιά άλλη διεργασία σε οποιαδήποτε server δεν μπορεί να πάρει κάποιο lock.

Ο πελάτης ζητάει συνεχώς να πάρει locks. Μόλις τα λάβει κάνει commit έτσι ώστε οι εμπλεκόμενοι servers να τα απελευθερώσουν. Με αυτό τον τρόπο πρώτα λαμβάνονται όλες οι κλειδαριές και μετά απελευθερώνονται.