

Κεφάλαιο 6

Συγχρονισμός στα Κ.Σ. και Κατανεμημένος Υπολογισμός

Σύνοψη

Δεδομένου ότι τα Κ.Σ. επικοινωνούν ασύγχρονα μέσω μηνυμάτων εγείρεται η απαίτηση για συγχρονισμό μεταξύ των επικοινωνούντων υποσυστημάτων. Στο κεφάλαιο αυτό, εξετάζεται η έννοια των λογικών ρολογιών και των διανυσματικών ρολογιών, όπου αντίθετα από τα φυσικά ρολόγια, ο χρόνος καθορίζεται με βάση τη σειρά των γεγονότων που συμβαίνουν, και εκφράζεται με τις χρονοσφραγίδες. Αναλύονται οι αλγόριθμοι για τη διάταξη και την αλληλουχία των γεγονότων που συμβαίνουν σε ένα Κ.Σ.. Επιπρόσθετα εξετάζονται οι καθολικές καταστάσεις και η συνέπεια ενός κατανεμημένου υπολογισμού. Μια εκτέλεση ενός Κ.Σ. μπορεί να μην είναι ρεαλιστική, δηλ. μπορεί να μην συμβαίνει, ενώ ένας κατανεμημένος υπολογισμός μπορεί να έχει πολλές ενδεχόμενες εκτελέσεις.

Προαπαιτούμενη γνώση

- 1) Δουλγέρης, Χ., Μητρόπουλος, Σ., 2015. Πληροφοριακά συστήματα στο διαδίκτυο. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/3969>
- 2) Savitch Walter (2015), Java, 7η Έκδοση, Εκδόσεις Α. Τζιόλα & Υιοί Α.Ε.
- 3) Shari Lawrence (2011), Τεχνολογία Λογισμικού: Θεωρία και Πράξη, Έκδοση: 2η Αμερικανική, Εκδόσεις Κλειδάριθμος ΕΠΕ.
- 4) Βακάλη Α., Παπαμήτσιου Ζ. (2012), Πληροφοριακά Συστήματα Παγκόσμιου Ιστού, Έκδοση: 1η, Εκδόσεις Νέων Τεχνολογιών.
- 5) Else Lervik, Vegard B. Havdal (2004), Java Με UML: Αντικειμενοστραφής Σχεδίαση και Προγραμματισμός, Έκδοση: 1η, Εκδόσεις Κλειδάριθμος ΕΠΕ.
- 6) Andrew S. Tanenbaum, Herbert Bos (2018), Σύγχρονα Λειτουργικά Συστήματα, Έκδοση: 4η Αμερικανική, Εκδόσεις Κλειδάριθμος ΕΠΕ

6.1 Εισαγωγή

Είναι συνηθισμένο φαινόμενο κατά τη λειτουργία των κατανεμημένων αλγορίθμων σε ένα κατανεμημένο σύστημα να εμφανίζονται δυσκολίες σε θέματα επεξεργασίας και επικοινωνίας μεταξύ των υποέργων που εκτελούνται κατά τη διάρκεια ενός μεγάλου έργου. Αυτά τα υποέργα περιλαμβάνουν λειτουργίες, όπως η έναρξη της εκτέλεσης ενός συμβάντος σε μία διαδικασία, η καθολική (ολική) μετάδοση πληροφοριών, καθώς και η απαίτηση ενός καθολικού συγχρονισμού μεταξύ διαφορετικών παράλληλα εκτελουμένων διεργασιών. Καθώς οι συγκεκριμένες εργασίες υλοποιούνται με τη μετάδοση μηνυμάτων σύμφωνα με ένα προκαθορισμένο σχήμα αρχιτεκτονικής και επικοινωνίας απαιτείται ο κατάλληλος συγχρονισμός. Για την επίτευξη του συγχρονισμού υλοποιούνται διάφοροι αλγόριθμοι οι οποίοι λειτουργούν με βάση κάποιους συγκεκριμένους τρόπους συγχρονισμού μέσω ρολογιών.

6.1.1 Συγχρονισμός

Με τον όρο “συγχρονισμός” ρολογιού εννοούμε τον μηχανισμό συγχρονισμού της ώρας όλων των υπολογιστών σε κατανεμημένα περιβάλλοντα ή συστήματα. Ας υποθέσουμε ότι υπάρχουν τρία συστήματα σε ένα κατανεμημένο περιβάλλον. Για τη διατήρηση των δεδομένων, δηλαδή για την αποστολή, τη λήψη και τη διαχείριση των δεδομένων μεταξύ των συστημάτων με τον ίδιο χρόνο με συγχρονισμένο τρόπο, χρειάζεται ένα ρολόι που πρέπει να είναι συγχρονισμένο. Αυτή η διαδικασία συγχρονισμού των δεδομένων είναι γνωστή ως συγχρονισμός ρολογιού. Ο συγχρονισμός σε ένα κατανεμημένο σύστημα είναι πιο περίπλοκος από ό,τι σε ένα κεντρικό σύστημα λόγω της χρήσης κατανεμημένων αλγορίθμων.

Οι κατανεμημένοι αλγόριθμοι ακολουθούν κάποιες ιδιότητες για τη διατήρηση του συγχρονισμού ρολογιού:

- Οι πληροφορίες θα είναι διάσπαρτες μεταξύ πολλαπλών μηχανών.
- Οι διεργασίες λαμβάνουν την οποιαδήποτε απόφαση μόνο με βάση τις τοπικές πληροφορίες.
- Πρέπει να αποφεύγεται το να υπάρχει ένα μόνο σημείο αποτυχίας (single point of failure).
- Δεν υπάρχει δυνατότητα ύπαρξης κοινού ρολογιού ή οποιαδήποτε άλλη καθολική πηγή χρόνου.
- Στα καταναμημένα συστήματα, ο χρόνος είναι διφορούμενος.

Καθώς τα καταναμημένα συστήματα έχουν τα δικά τους ρολόγια. Ο χρόνος μεταξύ των ρολογιών μπορεί επίσης να διαφέρει. Έτσι, είναι δυνατόν να συγχρονιστούν όλα τα ρολόγια σε καταναμημένο περιβάλλον.

Οι συγχρονιστές αναπτύχθηκαν από ερευνητές ως απάντηση στην ανάγκη να χρησιμοποιηθούν συνδυαστικά τα καλά χαρακτηριστικά των ασύγχρονων δικτύων και των σύγχρονων αλγορίθμων. Τα πρωτόκολλα με σκοπό τον συγχρονισμό είναι υπεύθυνα για την αναδημιουργία ενός ιδανικού σύγχρονου περιβάλλοντος μέσα σε ένα ασύγχρονο δίκτυο. Τα πρωτόκολλα αυτά εγγυώνται ότι οι παραπάνω ιδιότητες χρησιμοποιούνται με επιτυχία. Οι αλγόριθμοι των συγχρονιστών είναι ικανοί να λειτουργήσουν για οποιονδήποτε σύγχρονο αλγόριθμο και οποιοδήποτε δίκτυο. Είναι ζωτικής σημασίας να καθοριστούν τα χαρακτηριστικά των σύγχρονων και των ασύγχρονων δικτύων προκειμένου να καταστεί σαφές ποια είναι η λειτουργία των συγχρονιστών.

Σύγχρονα δίκτυα.

Η συγχρονισμένη μετάδοση είναι ο τύπος μετάδοσης κατά τον οποίο ένας κοινός παλμός ρολογιού μοιράζεται μεταξύ του πομπού και του δέκτη προκειμένου να επιτραπεί η συγχρονισμένη επικοινωνία. Στη συγχρονισμένη μετάδοση, τα δεδομένα αποστέλλονται με τη μορφή πλαισίων ή μπλοκ, οπότε τα δεδομένα μπορεί να αναγράφονται και ως μπλοκ. Επιπλέον στη συγχρονισμένη μετάδοση το χρονικό διάστημα μεταξύ δύο διαδοχικών μεταδόσεων είναι σταθερό.

Ένα πλεονέκτημα της συγχρονισμένης μετάδοσης είναι ότι μιλάμε για μια γρήγορη διαδικασία. Λόγω της εφαρμογής κοινού παλμού ρολογιού στη συγχρονισμένη μετάδοση, η μετάδοση των δεδομένων είναι ταχύτερη σε σύγκριση με εκείνη στην ασύγχρονη μετάδοση. Βέβαια εδώ υπάρχει το μειονέκτημα ότι αυτή η μορφή μετάδοσης είναι πιο δαπανηρή. Ένα άλλο πλεονέκτημα είναι ότι στη συγχρονισμένη μετάδοση είναι εύκολος ο σχεδιασμός της. Επιπλέον λόγω του κοινού παλμού ρολογιού στη σύγχρονη μετάδοση, δεν υπάρχει χρονικό κενό μεταξύ των δεδομένων.

Σε γενικές γραμμές στα σύγχρονα δίκτυα υπάρχει ένα κεντρικό ρολόι βάσει του οποίου συντονίζονται οι επεξεργαστές ενός Κ.Σ. στη διαχείριση των γεγονότων. Έτσι, όταν αποστέλλεται ένα σήμα καθολικής εκκίνησης, κάθε μεμονωμένος επεξεργαστής ξεκινά την εργασία του ταυτόχρονα. Κάθε επεξεργαστής κάνει ένα βήμα στη διαδικασία υπολογισμού και κοινοποιεί τα αποτελέσματά του σε κάθε τικ του ρολογιού. Θεωρείται δεδομένο ότι όλες οι υπολογιστικές λειτουργίες θα έχουν ολοκληρωθεί πριν από έναν καθορισμένο παλμό του ρολογιού και ότι όλα τα μηνύματα θα έχουν φτάσει με επιτυχία στους προβλεπόμενους παραλήπτες τους.

Ασύγχρονα δίκτυα.

Η ασύγχρονη μετάδοση είναι ο τύπος μετάδοσης κατά τον οποίο ο αποστολέας και ο παραλήπτης έχουν τα δικά τους εσωτερικά ρολόγια- επομένως δεν χρειάζονται εξωτερικό κοινό παλμό ρολογιού. Στην ασύγχρονη μετάδοση, τα δεδομένα μεταδίδονται με τη μορφή bytes ή χαρακτήρων, οπότε η μονάδα δεδομένων είναι byte/χαρακτήρας. Επιπλέον μία άλλη σημαντική διαφορά σε σχέση με τα σύγχρονα δίκτυα είναι ότι το χρονικό διάστημα μεταξύ δύο διαδοχικών μεταδόσεων είναι τυχαίο.

Ένα μειονέκτημα των ασύγχρονων μεταδόσεων είναι ότι χρειάζονται μεγάλο χρονικό διάστημα για να ολοκληρωθούν. Στην ασύγχρονη μετάδοση, κάθε αποστολέας και παραλήπτης έχουν τα δικά τους

ρολόγια εισόδου, επομένως η μετάδοση εξαρτάται από το συγχρονισμό τους. Έτσι, είναι πιο αργή σε σύγκριση με τη σύγχρονη μετάδοση, αλλά ταυτόχρονα, χρειάζονται πολλοί λίγοι πόροι για την ολοκλήρωση όλων των απαραίτητων λειτουργιών.

Η ασύγχρονη μετάδοση είναι πολύπλοκη στη φύση και στο σχεδιασμό σε σύγκριση με τη συγχρονισμένη μετάδοση. Επιπλέον στην ασύγχρονη μετάδοση, υπάρχει χρονικό κενό μεταξύ των δεδομένων (bytes). Ένα επιπλέον χαρακτηριστικό είναι ότι περιέχει bits έναρξης και bits λήξης στη μέση των οποίων υπάρχουν τα πραγματικά δεδομένα (λειτουργούν σαν όρια).

Στα ασύγχρονα δίκτυα δεν υπάρχει ένα κεντρικό ρολόι, ούτε ένα σήμα εκκίνησης, όπως στη προηγούμενη περίπτωση. Ως αποτέλεσμα, δεν υπάρχει τρόπος να προβλέψουμε πόσο χρόνο θα χρειαστεί για να μεταδοθούν τα μηνύματα μεταξύ των επεξεργαστών - υπολογιστικών κόμβων. Με άλλα λόγια, τα μηνύματα φτάνουν στον προορισμό τους σε κάποιο απροσδιόριστο χρονικό διάστημα.

6.2 Φυσικά Ρολόγια

Κάθε υπολογιστής έχει ένα κύκλωμα που είναι υπεύθυνο για την παρακολούθηση του χρόνου. Παρά το γεγονός ότι δεν είναι ένα παραδοσιακό ρολόι, αυτό το στοιχείο αναφέρεται συχνά ως ρολόι ή χρονιστής (clock). Είναι πιθανό η λέξη χρονόμετρο (timer) να είναι πιο ακριβής. Ένας κρύσταλλος χαλαζία, ο οποίος ταλαντώνεται σε μια προκαθορισμένη συχνότητα, και δύο καταχωρητές, γνωστοί ο ένας ως μετρητής (counter) και και ο άλλος ως καταχωρητής συγκράτησης (holding register), είναι τα συστατικά που συνθέτουν ένα ρολόι. Κάθε φορά που ο κρύσταλλος ταλαντώνεται, αφαιρείται μία μονάδα από το σύνολο του μετρητή. Όταν ο μετρητής χτυπήσει το μηδέν, δημιουργείται μια διακοπή και στη συνέχεια στον μετρητή καταχωρείται η τιμή που είναι αποθηκευμένη στον καταχωρητή συγκράτησης. Το κύκλωμα αυτό ονομάζεται φυσικό ρολόι, ή συχνά ρολόι υλικού.

Το ρολόι λογισμικού δημιουργείται από το λειτουργικό σύστημα χρησιμοποιώντας το ρολόι υλικού ως πηγή του. Αυτό έχει ως αποτέλεσμα έναν μηχανισμό χρονομέτρησης, που είναι πολύ κοντά στον πραγματικό για τον συγκεκριμένο υπολογιστή. Δεδομένου ότι όλες οι διεργασίες σε έναν υπολογιστή χρησιμοποιούν το ίδιο ρολόι, το σύστημα θα εξακολουθεί να είναι εσωτερικά συνεπές ανεξάρτητα εάν το ρολόι λογισμικού πηγαίνει πίσω ή μπροστά από το φυσικό ρολόι.

Όταν ερευνούμε ένα σύστημα που έχει έναν αριθμό μηχανών, καθεμία από τις οποίες έχει το δικό της ρολόι, οι συνθήκες αλλάζουν. Αν και η συχνότητα με την οποία ταλαντώνεται ένας συγκεκριμένος κρύσταλλος είναι αρκετά συνεπής, είναι αδύνατο να εγγυηθούμε ότι τα κρυσταλλικά ρολόγια όλων των υπολογιστών του συστήματος θα ταλαντώνονται με την ίδια ακριβώς συχνότητα.

Αυτό συμβαίνει επειδή παρά το ότι η συχνότητα με την οποία ταλαντώνεται ένας συγκεκριμένος κρύσταλλος είναι αρκετά σταθερή, στην πράξη, ο κρύσταλλος μέσα σε καθένα από αυτά τα ρολόγια ταλαντώνεται με μια συχνότητα που είναι κάπως διαφορετική από αυτή των κρυστάλλων μέσα σε όλα τα άλλα ρολόγια. Ως συνέπεια αυτού, οι τιμές των ρολογιών λογισμικού αποκλίνουν τελικά η μία από την άλλη. Ο όρος λόξωση (skew) αναφέρεται σε αυτή τη διαφορά στις τιμές που δείχνουν τα ρολόγια. Ο ρυθμός μετατόπισης ενός ρολογιού μπορεί να οριστεί ως ο ρυθμός μεταβολής της διαφοράς τιμής μεταξύ του εν λόγω ρολογιού και ενός "τέλειου" ρολογιού αναφοράς για ένα ορισμένο χρονικό διάστημα, όπως καθορίζεται από την ακρίβεια του ρολογιού αναφοράς.

Η ταχύτητα ολίσθησης (drift rate) για τα συμβατικά ρολόγια που βασίζονται σε κρυστάλλους χαλαζία είναι περίπου 10^6 δευτερόλεπτα ανά δευτερόλεπτο, κάτι που δείχνει διαφορά ενός δευτερολέπτου κάθε 10^6 δευτερόλεπτα. Η θερμοκρασία ενός κρυστάλλου επιδρά στη συχνότητα ταλάντωσης του κρυστάλλου. Τα ρολόγια χαλαζία με υψηλό επίπεδο ακρίβειας έχουν συχνά ταχύτητα ολίσθησης περίπου 10^7 ή 10^8 .

6.2.1 Συγχρονισμός Φυσικού ρολογιού

Κάθε φορά που ένα πρόγραμμα χρειάζεται να γνωρίζει την τρέχουσα ώρα, εκδίδει μια κλήση συστήματος στον πυρήνα, ο οποίος απαντάει επιστρέφοντας την τρέχουσα ώρα. Αρχικά εάν η διεργασία B ρωτήσει για το χρόνο λίγο αργότερα από τη διαδικασία A, η τιμή που λαμβάνει από τον πυρήνα η διεργασία B θα είναι μεγαλύτερη από (ή το πολύ ίση) από την τιμή που λαμβάνει η Διεργασία A.

Ας δούμε ένα παράδειγμα που καταδεικνύει τη σημασία της διατήρησης του χρονικού συγχρονισμού μεταξύ των κόμβων σε ένα κατανεμημένο σύστημα.

Για παράδειγμα ας υποθέσουμε ότι έχουμε έναν κώδικα σε γλώσσα java, το οποίο το ονομάζουμε «dokimi». Ο κώδικας περιέχεται σε ένα αρχείο dokimi.class. Επιπλέον υπάρχει και το αρχείο dokimi.java που αποτελεί το αρχείο που έχει ήδη μεταγλωττιστεί με σκοπό να εκτελεστεί. Θα εξετάσουμε δύο διαφορετικές περιπτώσεις χρονικού συγχρονισμού με τη βοήθεια της εντολής unix “make”, εκμεταλλευόμενοι την ιδιότητα της “make” να μεταγλωττίζει όσα αρχεία έχουν τροποποιηθεί μετά τη μεταγλώττιση.

1. Ας φανταστούμε ότι ένας χρήστης άλλαξε το dokimi.java και μετά έτρεξε το build με τη βοήθεια της “make”, η οποία ελέγχει τη διαφορά στις ημερομηνίες δημιουργίας μεταξύ του αρχείου πηγής και του αντικειμένου. Εάν το dokimi.java ενημερώθηκε μετά το dokimi.class, τότε η μεταγλώττιση του αρχείου θα προχωρήσει χωρίς προβλήματα.
2. Ας σκεφτούμε την κατάσταση όταν το ρολόι του υπολογιστή του συντάκτη υστερεί σε σχέση με το ρολόι του υπολογιστή του μεταγλωττιστή. Επειδή ο χρόνος ενημέρωσης του HelloWorld.java φαίνεται να είναι μικρότερος από εκείνον του dokimi.class, με την εκτέλεση της “make” μπορεί να θεωρηθεί λανθασμένα ότι το αρχείο προέλευσης δεν χρειάζεται εκ νέου μεταγλώττιση, γεγονός που οδηγεί σε επιπλοκές στη λειτουργία του.

Το παραπάνω αποδεικνύει ότι σε ένα αποκεντρωμένο σύστημα, η ρύθμιση σε μια κοινή χρονική αναφορά δεν είναι εύκολη.

Εδώ πρέπει να αναφερθεί ότι ο συγχρονισμός του ρολογιού μπορεί να επιτευχθεί με 2 τρόπους, με τον εξωτερικό και με τον εσωτερικό .

- Ο εξωτερικός συγχρονισμός ρολογιού είναι αυτός στον οποίο υπάρχει ένα εξωτερικό ρολόι αναφοράς. Το ρολόι αυτό χρησιμοποιείται ως αναφορά και οι κόμβοι του συστήματος μπορούν να ρυθμίσουν και να προσαρμόσουν τον χρόνο τους ανάλογα με αυτό. Παράδειγμα εξωτερικού συγχρονισμού αποτελεί ο αλγόριθμος Cristian που αναλύεται παρακάτω.
- Εσωτερικός συγχρονισμός ρολογιού είναι αυτός στον οποίο κάθε κόμβος μοιράζεται την ώρα του με άλλους κόμβους και όλοι οι κόμβοι ρυθμίζουν και προσαρμόζουν τις ώρες τους αναλόγως. Παράδειγμα εξωτερικού συγχρονισμού αποτελεί ο αλγόριθμος Berkeley που αναλύεται παρακάτω.

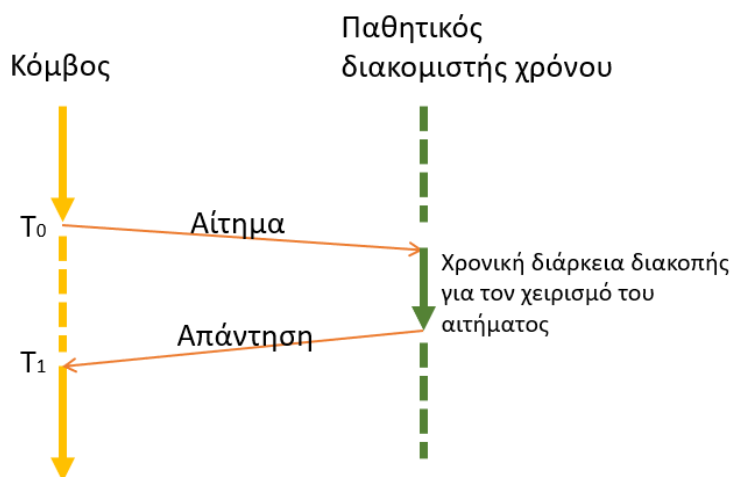
6.2.2 Αλγόριθμοι συγχρονισμού ρολογιών

Στο συγκεκριμένο υποκεφάλαιο θα κατηγοριοποιηθούν και θα μελετηθούν οι αλγόριθμοι που είναι υπεύθυνοι για την υλοποίηση του συγχρονισμού των ρολογιών. Ο συγχρονισμός ενός ρολογιού μπορεί να είναι κεντρικοποιημένος ή κατανεμημένος. Παρακάτω θα αναλυθεί η έννοια των κεντρικοποιημένων και κατανεμημένων συγχρονισμών ρολογιού, καθώς και οι αλγόριθμοι που υλοποιούν τον κάθε έναν από αυτούς τους δύο τύπους συγχρονισμού ρολογιού.

Ο **κεντρικοποιημένος συγχρονισμός ρολογιού** είναι εκείνος στον οποίο χρησιμοποιείται ένας διακομιστής χρόνου ως αναφορά. Ο μοναδικός αυτός διακομιστής χρόνου διαδίδει την ώρα του στους κόμβους και όλοι οι κόμβοι προσαρμόζουν την ώρα ανάλογα. Ο κεντρικοποιημένος συγχρονισμός

ρολογιού εξαρτάται από τον ενιαίο διακομιστή ώρας, οπότε αν ο κόμβος αυτός αποτύχει, ολόκληρο το σύστημα θα χάσει τον συγχρονισμό. Ο στόχος ενός κεντρικοποιημένου αλγορίθμου συγχρονισμού είναι να διατηρεί τα ρολόγια όλων των άλλων κόμβων συγχρονισμένα με τον κόμβο διακομιστή χρόνου. Τα συστήματα που χρησιμοποιούν κεντρικοποιημένο συγχρονισμό ρολογιού έχουν έναν κόμβο με δέκτη πραγματικού χρόνου που ονομάζεται *κόμβος διακομιστή χρόνου*. Ο χρόνος του ρολογιού αυτού του κόμβου θεωρείται σωστός και χρησιμοποιείται ως χρόνος αναφοράς. Παραδείγματα κεντρικοποιημένων αλγορίθμων συγχρονισμού είναι ο Berkeley, και ο αλγόριθμος του Cristian.

Αλγόριθμος του Cristian



Εικόνα 6.1 Παράδειγμα του Αλγορίθμου του Cristian

Πρόκειται για μια προσέγγιση παθητικού διακομιστή χρόνου που σημαίνει ότι ο κάθε κόμβος στέλνει περιοδικά ένα μήνυμα στον διακομιστή χρόνου. Όταν ο διακομιστής χρόνου λάβει το μήνυμα απαντά με ένα μήνυμα T , όπου T είναι η τρέχουσα ώρα του κόμβου του διακομιστή. Ας υποθέσουμε ότι η ώρα του ρολογιού του πελάτη είναι T_0 όταν στέλνει το μήνυμα και T_1 όταν λαμβάνει το μήνυμα από τον εξυπηρετητή. Οι T_0 και T_1 μετρούνται με το ίδιο ρολόι, οπότε η καλύτερη εκτίμηση του χρόνου διάδοσης είναι $(T_1 - T_0)/2$.

Όταν η απάντηση λαμβάνεται στον κόμβο του πελάτη, το ρολόι του αναπροσαρμόζεται σε $T + (T_1 - T_0)/2$. Μπορεί να υπάρχει απρόβλεπτη διακύμανση στο χρόνο διάδοσης του μηνύματος μεταξύ των κόμβων, επομένως το $(T_1 - T_0)/2$ δεν είναι καλό να προστεθεί στο T για τον υπολογισμό του τρέχοντος χρόνου.

Για το σκοπό αυτό, γίνονται αρκετές μετρήσεις του $T_1 - T_0$ και αν αυτές οι μετρήσεις υπερβαίνουν κάποια τιμή κατωφλίου τότε είναι αναξιόπιστες και απορρίπτονται. Για τις υπόλοιπες μετρήσεις υπολογίζεται ο μέσος όρος τους και η μισή από την υπολογιζόμενη τιμή προστίθεται στο T . Επιπλέον η ελάχιστη τιμή θεωρείται ακριβής.

- Πλεονέκτημα: Υποθέτει ότι δεν υπάρχουν διαθέσιμες πρόσθετες πληροφορίες.
- Μειονέκτημα: Περιορίζει τον αριθμό των μετρήσεων για την εκτίμηση της τιμής.

Ο αλγόριθμος του Berkeley

Πρόκειται για μια προσέγγιση ενεργού διακομιστή ώρας που σημαίνει ότι ο διακομιστής ώρας μεταδίδει περιοδικά την ώρα του ρολογιού του και οι άλλοι κόμβοι λαμβάνουν το μήνυμα για να διορθώσουν τα δικά τους ρολόγια.

Σε αυτόν τον αλγόριθμο, ο διακομιστής ώρας στέλνει περιοδικά ένα μήνυμα σε όλους τους κόμβους. Όταν λάβει αυτό το μήνυμα, κάθε κόμβος στέλνει πίσω τη δική του τιμή ρολογιού στον διακομιστή ώρας. Ο διακομιστής χρόνου έχει εκ των προτέρων γνώση του κατά προσέγγιση χρόνου που απαιτείται για τη διάδοση ενός μηνύματος, ο οποίος χρησιμοποιείται για την αναπροσαρμογή των τιμών του ρολογιού. Στη συνέχεια, λαμβάνει έναν μέσο όρο ανοχής σφαλμάτων των τιμών του ρολογιού όλων των κόμβων. Ο υπολογισμένος μέσος όρος είναι η τρέχουσα ώρα στην οποία πρέπει να αναπροσαρμοστούν όλα τα ρολόγια.

Ο διακομιστής χρόνου αναπροσαρμόζει το δικό του ρολόι σε αυτή την τιμή και αντί να στείλει την τρέχουσα ώρα στους άλλους κόμβους στέλνει το χρονικό διάστημα που χρειάζεται κάθε κόμβος για την αναπροσαρμογή. Το διάστημα αυτό υπολογίζεται με βάση τη γνώση που έχει ο διακομιστής ώρας σχετικά με τη διάδοση του μηνύματος και μπορεί να λαμβάνει θετική ή αρνητική τιμή.

Ο **κατανεμημένος συγχρονισμός ρολογιού** είναι αυτός στον οποίο δεν υπάρχει κεντρικός διακομιστής ώρας. Αντ' αυτού οι κόμβοι ρυθμίζουν την ώρα τους χρησιμοποιώντας την τοπική τους ώρα και στη συνέχεια, λαμβάνοντας το μέσο όρο των διαφορών της ώρας με άλλους κόμβους. Οι κατανεμημένοι αλγόριθμοι συγχρονισμού ξεπερνούν τα προβλήματα των κεντρικών αλγορίθμων, όπως η έλλειψη επεκτασιμότητας και η ύπαρξη ενός μονού σημείου αστοχίας. Παραδείγματα κατανεμημένων αλγορίθμων είναι αλγόριθμος του σφαιρικού μέσου όρου, αλγόριθμος του τοπικού μέσου όρου, και το πρωτόκολλο χρόνου δικτύου (NTP -Network Time Protocol).

Κατανεμημένοι αλγόριθμοι καθολικού (global) μέσου όρου

Σε αυτή την προσέγγιση η διαδικασία ρολογιού σε κάθε κόμβο μεταδίδει την τοπική ώρα του ρολογιού της με τη μορφή ενός μηνύματος "επανασυγχρονισμού" στην αρχή κάθε σταθερού μήκους διαστήματος επανασυγχρονισμού. Αυτό γίνεται όταν ο τοπικός χρόνος του είναι ίσος με $T_0 + iR$ για κάποιο ακέραιο i , όπου T_0 είναι ένας σταθερός χρόνος που συμφωνείται από όλους τους κόμβους και R είναι μια παράμετρος του συστήματος που εξαρτάται από το σύνολο των κόμβων σε ένα σύστημα.

Μετά τη μετάδοση της τιμής του ρολογιού, η διαδικασία ρολογιού ενός κόμβου περιμένει για χρόνο T που καθορίζεται από τον αλγόριθμο.

Κατά τη διάρκεια αυτής της αναμονής, η διεργασία ρολογιού συλλέγει τα μηνύματα επανασυγχρονισμού και καταγράφει την ώρα λήψης του μηνύματος, η οποία εκτιμά τη λόξωση (skew) μετά το πέρας της αναμονής. Στη συνέχεια, υπολογίζει τον μέσο όρο ανεκτικότητας σε σφάλματα της εκτιμώμενης στρέβλωσης και τον χρησιμοποιεί για τη διόρθωση των ρολογιών.

Κατανεμημένοι αλγόριθμοι εντοπισμένου μέσου όρου

Οι κατανεμημένοι αλγόριθμοι καθολικού μέσου όρου δεν κλιμακώνονται καθώς χρειάζονται δίκτυο για την υποστήριξη της δυνατότητας της μετάδοσης με αποτέλεσμα να δημιουργείται μεγάλη κίνηση κατά την αποστολή των μηνυμάτων. Οι κατανεμημένοι αλγόριθμοι εντοπισμένου (localized) μέσου έρχονται να δώσουν λύση στο ζήτημα των αλγορίθμων εντοπισμένου μέσου όρου, για τον λόγο ότι οι κόμβοι στην περίπτωση αυτή των κατανεμημένων συστημάτων είναι λογικά διατεταγμένοι σε δακτύλιο ή σε ένα μοτίβο. Κάθε κόμβος ανταλλάσσει το χρόνο ρολογιού του με τους γείτονές του και στη συνέχεια θέτει το δικό του χρόνο ρολογιού στον μέσο όρο του δικού του χρόνου ρολογιού και των γειτόνων του.

Network Time Protocol (NTP)

Το Πρωτόκολλο Χρόνου Δικτύου - Network Time Protocol (NTP) είναι ένα πρωτόκολλο που βοηθάει στον συγχρονισμό των χρόνων του ρολογιού των υπολογιστών σε ένα δίκτυο. Αυτό το πρωτόκολλο είναι ένα πρωτόκολλο εφαρμογής που είναι υπεύθυνο για το συγχρονισμό των υπολογιστών σε ένα δίκτυο TCP/IP. Το NTP αναπτύχθηκε από τον David Mills το 1981 στο Πανεπιστήμιο του Delaware και

απαιτείται σε έναν μηχανισμό επικοινωνίας, ώστε να υπάρχει απρόσκοπτη σύνδεση μεταξύ των υπολογιστών.

Το πρωτόκολλο χρόνου δικτύου έχει κάποια βασικά χαρακτηριστικά που αφορούν τους διακομιστές του NTP, την διαδικασία επικοινωνίας των συστημάτων και την διαχείριση του χρόνου στους διακομιστές αρχείων. Έτσι έχουμε τα παρακάτω χαρακτηριστικά:

- Οι διακομιστές NTP έχουν πρόσβαση σε εξαιρετικά ακριβή ατομικά ρολόγια και ρολόγια GPU.
- Ο διακομιστής NTP χρησιμοποιεί συντονισμένο παγκόσμιο χρόνο (UTC) για τον συγχρονισμό του χρόνου του ρολογιού της CPU.
- Με τη χρήση του διακομιστή NTP αποφεύγονται διάφορες ευπάθειες που προκύπτουν κατά την διαδικασία της επικοινωνίας των συστημάτων που είναι υπεύθυνα για την ανταλλαγή πληροφοριών.
- Ο διακομιστής NTP παρέχει συνεπή και αξιόπιστη διαχείριση του χρόνου για τους διακομιστές αρχείων.

Λειτουργία του NTP

Το NTP είναι ένα πρωτόκολλο που λειτουργεί πάνω από το επίπεδο εφαρμογής, χρησιμοποιεί ένα ιεραρχικό σύστημα χρονικών πόρων και παρέχει συγχρονισμό εντός των διακομιστών στρώματος. Στο ανώτατο επίπεδο, υπάρχουν υψηλής ακρίβειας χρονικοί πόροι" π.χ. ατομικά ρολόγια ή ρολόγια GPS. Αυτοί οι πόροι ρολογιού ονομάζονται εξυπηρετητές στρώματος 0 και συνδέονται με τον παρακάτω εξυπηρετητή NTP που ονομάζεται στρώμα 1,2 ή 3. Αυτοί οι διακομιστές παρέχουν στη συνέχεια την ακριβή ημερομηνία και ώρα, ώστε οι επικοινωνούντες κεντρικοί υπολογιστές να συγχρονίζονται μεταξύ τους.

Εφαρμογές του NTP

Το NTP έχει και διάφορες άλλες εφαρμογές, όπως σε οπτικοακουστικά συστήματα.

- Χρησιμοποιείται σε ένα σύστημα παραγωγής όπου καταγράφεται ο ζωντανός ήχος.
- Χρησιμοποιείται στην ανάπτυξη υποδομών ραδιοτηλεοπτικής μετάδοσης.

Πλεονεκτήματα του NTP

Πέραν των διαφόρων χαρακτηριστικών και εφαρμογών του NTP πρωτοκόλλου, παρακάτω αναφέρονται ορισμένα πλεονεκτήματα στη χρήση του πρωτοκόλλου αυτού.

- Παρέχει διαδικτυακό συγχρονισμό μεταξύ των συσκευών.
- Παρέχει αυξημένη ασφάλεια εντός των εγκαταστάσεων.
- Χρησιμοποιείται στα συστήματα αυθεντικοποίησης όπως το Kerberos.
- Παρέχει επιτάχυνση δικτύου που βοηθά στην αντιμετώπιση προβλημάτων.
- Χρησιμοποιείται σε συστήματα αρχείων που είναι δύσκολα στο συγχρονισμό μέσω δικτύου.

Μειονεκτήματα του NTP

Όπως κάθε πρωτόκολλο και κάθε τεχνολογία πέρα από τα πλεονεκτήματα στη χρήση του πάνω σε συστήματα, υπάρχουν και αντίστοιχα μειονεκτήματα.

- Όταν οι διακομιστές είναι εκτός λειτουργίας επηρεάζεται ο χρόνος συγχρονισμού σε όλη την τρέχουσα επικοινωνία.

- Οι διακομιστές είναι επιρρεπείς σε σφάλματα λόγω των διαφόρων ζωνών ώρας και μπορεί να προκύψουν συγκρούσεις.
- Ελάχιστη μείωση της ακρίβειας του χρόνου.
- Όταν αυξάνονται τα πακέτα NTP ο συγχρονισμός συγκρούεται.
- Μπορεί να γίνει χειραγώγηση στο συγχρονισμό.

6.3 Λογικά Ρολόγια

Μέχρι στιγμής, υποθέσαμε ότι ο συγχρονισμός του ρολογιού σχετίζεται φυσικά με τον πραγματικό χρόνο. Συχνά δεν ξέρουμε ποια έκδοση ενός κομματιού δεδομένων είναι η πιο ενημερωμένη με βάση τη φυσική χρονοσφραγίδα και μόνο, καθώς είναι σχεδόν αδύνατο να διασφαλίσουμε ότι όλες οι οντότητες (με τον όρο αυτό εννοούνται διεργασίες ή κόμβοι) έχουν τέλεια συγχρονισμένα φυσικά ρολόγια. Πάρτε, για παράδειγμα, δύο συγχρονισμένους διακομιστές που γράφουν χρονοσφραγίδες στο ίδιο σύστημα. Αν ο ένας διακομιστής μείνει πίσω έστω και μερικά χιλιοστά του δευτερολέπτου, θα ήταν γρήγορα αδύνατο να γνωρίζουμε την πραγματική σειρά των γεγονότων. Ένα άλλο παράδειγμα είναι το πρόβλημα που αναλύσαμε προηγουμένως με τη συνάρτηση “make”. Για να λύσουμε αυτά τα προβλήματα, μπορούμε να χρησιμοποιήσουμε λογικά ρολόγια που βασίζονται σε γεγονότα και όχι σε χρόνο για να δημιουργήσουμε μερικώς διατεταγμένα σύνολα.

Μπορούμε σχεδόν να σκεφτούμε τα λογικά ρολόγια ως έναν τρόπο έκδοσης των γεγονότων σε ένα σύστημα. Εξετάζοντας αυτά τα διατεταγμένα σύνολα, μπορούμε στη συνέχεια να συγχρονίσουμε τα δεδομένα μας σε ένα σύστημα, επειδή γνωρίζουμε, σε γενικές γραμμές, ποια κομμάτια δεδομένων είναι τα πιο ενημερωμένα και είμαστε σε θέση να εντοπίσουμε ταυτόχρονα γεγονότα.

Σε μια κλασική εργασία, ο Lamport (1978) έδειξε ότι, αν και ο συγχρονισμός ρολογιού είναι δυνατός, δεν χρειάζεται να είναι απόλυτος. Εάν δύο διεργασίες δεν αλληλεπιδρούν, δεν είναι απαραίτητο τα ρολόγια τους να είναι συγχρονισμένα, διότι η έλλειψη συγχρονισμού δεν θα ήταν παρατηρήσιμη και, επομένως, δεν θα μπορούσε να προκαλέσει προβλήματα. Επιπλέον, επεσήμανε ότι αυτό που συνήθως έχει σημασία δεν είναι να συμφωνούν όλες οι διεργασίες για το τι ώρα ακριβώς είναι, αλλά μάλλον να συμφωνούν για τη σειρά με την οποία συμβαίνουν τα γεγονότα. Στο παράδειγμα make, αυτό που μετράει είναι αν η `dokimi.class` είναι παλαιότερη ή νεότερη από την `dokimi.java`, όχι οι απόλυτοι χρόνοι δημιουργίας τους. Στην επόμενη υποενότητα θα συζητήσουμε τον αλγόριθμο του Lamport, ο οποίος συγχρονίζει τα λογικά ρολόγια.

6.3.1 Ρολόγια Lamport

Τα ρολόγια Lamport έχουν κάποια συγκεκριμένα χαρακτηριστικά. Αρχικά ο ρόλος τους είναι να μπορέσουν να συγχρονίσουν τις αλληλεπιδράσεις των διεργασιών μέσα στο χρόνο. Στην περίπτωση που κάποιες διεργασίες δεν αλληλεπιδρούν, τότε προφανώς και δεν είναι απαραίτητο να συγχρονιστούν, διότι δεν δημιουργείται πρόβλημα στην λειτουργία της εκάστοτε εργασίας. Επιπλέον, πρέπει να υπογραμμίσουμε ότι δεν έχει σημασία η ακριβής ώρα εκτέλεσης των γεγονότων, αλλά κυρίως η χρονική σειρά με την οποία τα γεγονότα αυτά συμβαίνουν.

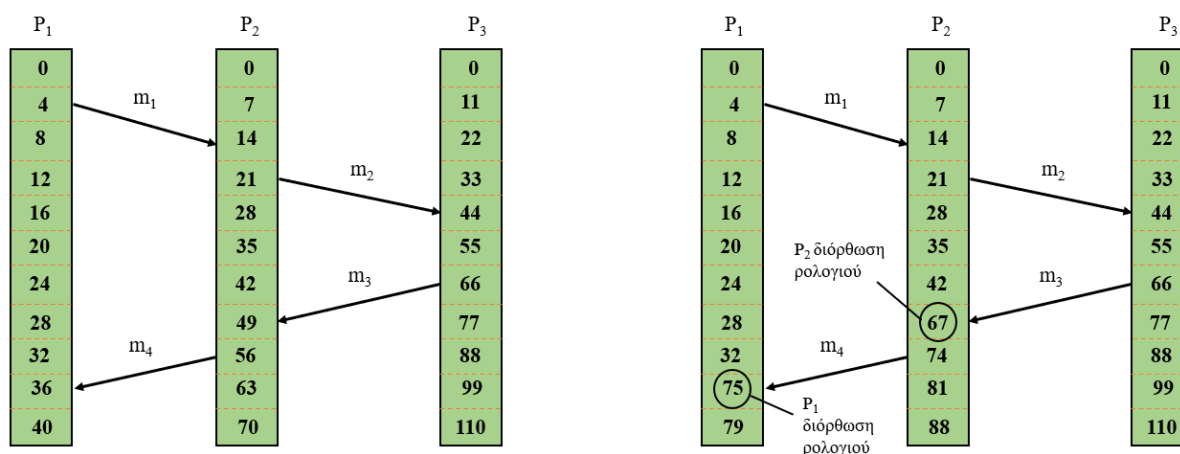
Αυτό που μόλις περιγράψαμε ονομάζεται σχέση «συνέβη-πριν» και ορίζεται με τον εξής τρόπο. Εάν α και β είναι υποδιεργασίες που συμβαίνουν στην ίδια διεργασία και το α συμβαίνει πριν το β , τότε $\alpha \rightarrow \beta$.

Εάν α είναι μία υποδιεργασία αποστολής μηνύματος από μία διεργασία και β είναι η υποδιεργασία της παραλαβής του αντίστοιχου μηνύματος, τότε $\alpha \rightarrow \beta$.

Γενικότερα, η σχέση «συνέβη πριν» είναι μία σχέση που μας δείχνει μετάβαση. Επιπλέον υπάρχουν και τα ταυτόχρονα γεγονότα, όπου μιλάμε για δύο υποδιεργασίες α και β οι οποίες δεν ανήκουν σε διεργασίες που ανταλλάσσουν μηνύματα με κανέναν τρόπο (ούτε μέσω τρίτων). Αυτό σημαίνει ότι δεν ισχύει $\alpha \rightarrow \beta$ ούτε $\beta \rightarrow \alpha$ και ισχύει ότι $\alpha // \beta$. Στην περίπτωση που απαιτείται η ύπαρξη γεγονότων

στο σύστημα που να συμβαίνουν την ίδια χρονική στιγμή, μία λύση είναι η κατάταξη των διεργασιών με γνώμονα τον αύξοντα αριθμό τους σε χαμηλής και υψηλής τάξης.

Η παρακάτω Εικόνα 6.2 αρχικά δείχνει τρεις διαδικασίες η καθεμία με διαφορετικό ρολόι και χρονικό ρυθμό. Παρατηρούμε ότι η διεργασία m_3 που συνδέει την διαδικασία P_3 με τη διαδικασία P_2 δεν εκτελείται χρονικά ορθά. Αυτό φαίνεται γιατί ξεκινάει να εκτελείται στην χρονική στιγμή "66" και τερματίζει στην χρονική στιγμή "49". Ίδιο φαινόμενο παρατηρείται και στην διεργασία m_4 . Η λύση αυτού του προβλήματος σύμφωνα με τον Lamport φαίνεται στο δεύτερο σχήμα όπου στη διαδικασία P_2 η χρονική στιγμή "49" αλλάζει σε "67", έτσι ώστε να συμβαδίζει με την λογική «συνέβη-πριν» της χρονικής στιγμής 66 της διαδικασίας P_3 . Οι επόμενες χρονικές στιγμές από την χρονική στιγμή "67" της διαδικασίας P_2 αλλάζουν σύμφωνα με τον προϋπάρχον ρυθμό που στην περίπτωση αυτή είναι 7. Το ίδιο συμβαίνει αντίστοιχα και για τη διεργασία m_4 αμέσως μετά.



α) Τρεις διαδικασίες, η καθεμία με το δικό της ρολόι. Τα ρολόγια λειτουργούν με διαφορετικούς ρυθμούς.

β) Ο αλγόριθμος του Lamport διορθώνει τα ρολόγια.

Εικόνα 6.2 Παράδειγμα του Αλγόριθμου Lamport

Πολυεκπομπή με πλήρη σειρά

Σε πολλές περιπτώσεις, είναι απαραίτητο να εκτελεστεί μία πολυεκπομπή (multicast) με απόλυτη διάταξη μεταξύ των μηνυμάτων που λαμβάνονται από τον παραλήπτη, τα οποία αποτελούν διπλά αντίγραφα και έχουν παραληφθεί με λάθος σειρά. Με άλλα λόγια, όλα τα μηνύματα πρέπει να μεταβιβάζονται σε κάθε παραλήπτη με την ίδια σειρά. Το λογικό ρολόι του Lamport μπορεί να χρησιμοποιηθεί για την υλοποίηση της πολυεκπομπής με απόλυτη σειρά υπό πλήρως καταναμημένα συστήματα.

Όταν μια διεργασία λαμβάνει ένα συγκεκριμένο μήνυμα, αυτό τοποθετείται στην τοπική ουρά με τη σειρά σύμφωνα με τη χρονοσφραγίδα. Ο παραλήπτης πολυεκπέμπει μια επιβεβαίωση στις άλλες διεργασίες. Αν ακολουθηθεί ο αλγόριθμος του Lamport για να προσαρμοστεί το τοπικό ρολόι, όλες οι διεργασίες θα έχουν στην πραγματικότητα το ίδιο αντίγραφο της τοπικής ουράς. Για να υλοποιηθεί αυτό σύμφωνα με τον αλγόριθμο του Lamport, ο αλγόριθμος μπορεί να περάσει μηνύματα σε μια ουρά σε μια εκτελούμενη διεργασία μόνο εάν το μήνυμα βρίσκεται στην κεφαλή της ουράς και επιβεβαιώνεται (μέσω ack μηνύματος) από όλες τις άλλες διεργασίες. Μία διεργασία για να στείλει

μήνυμα επιβεβαίωσης (ack) πρέπει να μην έχει φτιάξει κάποιο άλλο καινούργιο αίτημα για ενημέρωση της ουράς και η διεργασία αυτή να θεωρείται ισχυρότερη ή διεργασία με μεγαλύτερη προτεραιότητα σε σχέση με τις διεργασίες που θέλει να επικοινωνήσει. Επομένως, όλα τα μηνύματα παραδίδονται με την ίδια σειρά παντού. Με άλλα λόγια, έχει επιτευχθεί η πολυεκπομπή με απόλυτη σειρά.

Παρακάτω παρουσιάζεται ένα παράδειγμα εφαρμογής πολυεκπομπής με πλήρη σειρά σύμφωνα με τον αλγόριθμο του Lamport. Στο παράδειγμα έχουμε ένα μήνυμα (μ_1) και ένα μήνυμα (μ_2), με το μήνυμα (μ_1) ουσιαστικά να προηγείται, το οποίο σημαίνει ότι τα γεγονότα που το περιλαμβάνουν θα πρέπει να εκτελεστούν πριν από το μήνυμα (μ_2). Επίσης το παράδειγμα απεικονίζεται στην εικόνα 6.3.

Πρώτο βήμα παραδείγματος.

- Αρχικά έχουμε την αποστολή του μηνύματος μ_1 μεταξύ διεργασιών σε συνάρτηση με τον χρόνο έκδοσης που είναι 2.1 στη μονάδα χρόνου.
- Έπειτα έχουμε την αποστολή ενός απλού μηνύματος μ_2 για την έναρξη άλλης επικοινωνίας μεταξύ των διεργασιών. Ο χρόνος έκδοσης σε αυτή την περίπτωση είναι 2.2 στη μονάδα του χρόνου.
- Τα μηνύματα μεταδίδονται πολλαπλώς (multicast) σε όλες τις διεργασίες της ομάδας, συμπεριλαμβανομένης της ίδιας.
 - Ας υποθέσουμε ότι ένα μήνυμα που αποστέλλεται από μια διεργασία στον εαυτό της λαμβάνεται από τη διεργασία σχεδόν αμέσως.
 - Για άλλες διεργασίες, μπορεί να υπάρχει καθυστέρηση.

Δεύτερο βήμα παραδείγματος.

- Στο χρονικό σημείο αυτό οι ουρές των διεργασιών θα έχουν την εξής πληροφορία (Με Δ σε αυτό το παράδειγμα συμβολίζουμε την διεργασία):
 - Δ_1 : (t. μ_1 ->2.1), (t. μ_2 ->2.2), όπου t. μ_1 , η χρονοσφραγίδα δημιουργίας του μ_1 και t.m2 χρονοσφραγίδα δημιουργίας του μ_2 . (Παρακάτω το (t. μ_1 ->2.1) και το (t. μ_2 ->2.2) θα το συμβολίζουμε με (μ_1 -2.1) και με (μ_2 -2.2) αντίστοιχα.)
 - Δ_2 : (t. μ_1 ->2.1), (t. μ_2 ->2.2)
- Η Δ_1 θα κάνει multicast μια επιβεβαίωση για το (μ_1 -2.1), αλλά όχι για το (μ_2 -2.2).
 - Αυτό συμβαίνει γιατί το αναγνωριστικό του Δ_1 είναι υψηλότερης σημασίας (προγενέστερο) από το αναγνωριστικό της Δ_2 και η Δ_1 έχει εκδώσει αίτηση, δεδομένου ότι χρονικά ισχύει $2.1 < 2.2$
- Η Δ_2 θα κάνει multicast μια επιβεβαίωση για το (μ_1 -2.1) αλλά και για το (μ_2 -2.2)
 - Αυτό συμβαίνει γιατί το αναγνωριστικό της Δ_2 δεν είναι υψηλότερης σημασίας (προγενέστερο) από το αναγνωριστικό της Δ_1 , δεδομένου ότι χρονικά ισχύει $2.1 < 2.2$

Τρίτο βήμα παραδείγματος.

- Η Δ_1 δεν εκδίδει επιβεβαίωση (ack) για το (μ_2 -2.2), έως ότου ολοκληρωθεί η επεξεργασία της πράξης μ_1 , επειδή εκτελείται πριν την Δ_2 αφού προηγείται σαν προγενέστερη χρονικά διεργασία.
- Η πραγματική παραλαβή από την Δ_1 του μηνύματος (μ_2 -2.2) λαμβάνει χρονοσφραγίδα 6.1.
- Η πραγματική παραλαβή από την Δ_2 του μηνύματος (μ_1 -2.1) λαμβάνει χρονοσφραγίδα 6.2.

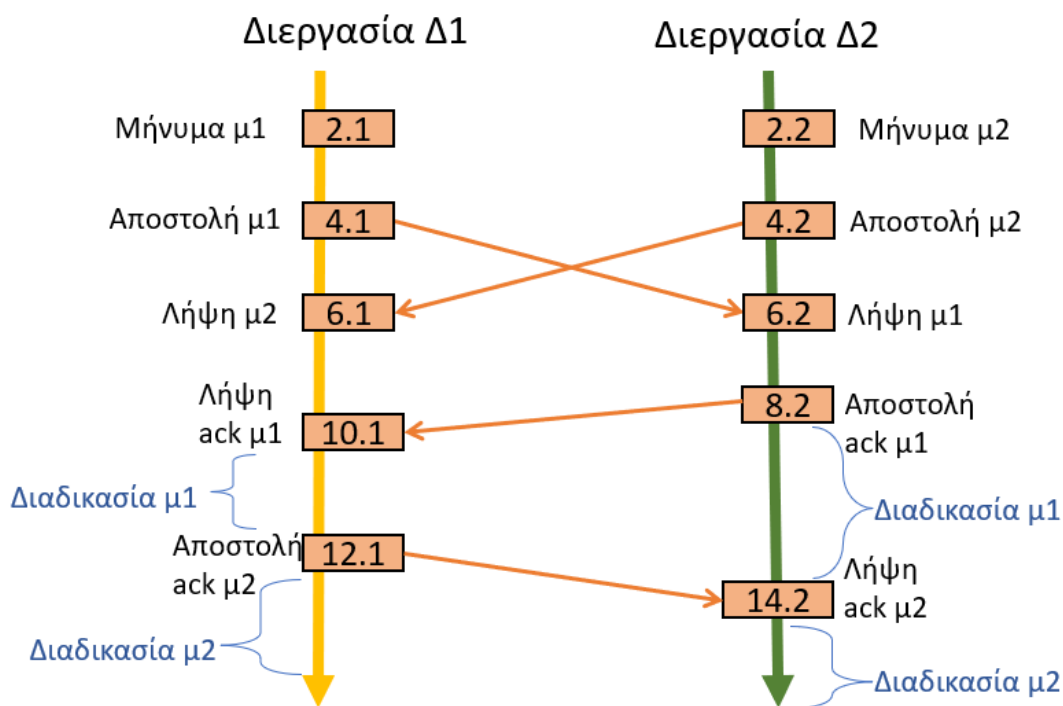
Τέταρτο βήμα παραδείγματος.

- Εάν η Δ_2 λάβει το (μ_2 -2.2) πριν από το (μ_1 -2.1), εξακολουθεί να αποστέλλει μια επιβεβαίωση για το (μ_2 -2.2). Αυτό συμβαίνει γιατί έχει οριστεί η σειρά από την αρχή και δεν αλλάζει ούτε σε περίπτωση κάποιας χρονικής καθυστέρησης στην Δ_1 .
- Σε αυτό το σημείο, πώς γνωρίζει η Δ_2 ότι υπάρχουν άλλες ενημερώσεις που πρέπει να γίνουν πριν από αυτή που εξέδωσε;
 - Δεν το γνωρίζει,

- Δεν προχωράει στην εκτέλεση της ενημέρωσης που καθορίζεται στο (μ2-2.2) μέχρι να λάβει επιβεβαίωση από όλες τις άλλες διεργασίες, που στην προκειμένη περίπτωση σημαίνει την Δ1.
- Η Δ2 στέλνει μια επιβεβαίωση για το (μ1-2.1) όταν τη λάβει;
 - Ναι, αφού $1 < 2$

Συνοψίζοντας, έχουν σταλεί τα ακόλουθα μηνύματα:

- Οι Δ1 και Δ2 έχουν εκδώσει μηνύματα ενημέρωσης.
- Η Δ1 έχει αποστείλει multicasting ένα μήνυμα επιβεβαίωσης για το (μ1-2.1).
- Η Δ2 έχει αποστείλει πολλαπλά μηνύματα επιβεβαίωσης (ack) για τα (μ1-2.1), (μ2-2.2).
- Οι Δ1 και Δ2 έχουν λάβει μήνυμα επιβεβαίωσης από όλες τις διεργασίες για το (μ1-2.1).
- Συνεπώς, η επιβεβαίωση(ack) που εκτελείται από το μ1(Διαδικασία μ1 στο σχήμα 6.3) μπορεί να προχωρήσει τόσο στην Δ1 όσο και στην Δ2.



Εικόνα 6.3 Παράδειγμα Πολυεκπομπής με πλήρη σειρά

Όταν η Δ1 έχει τελειώσει με το μ1 μπορεί στη συνέχεια να προχωρήσει στην πολυεκπομπή μιας επιβεβαίωσης για το (μ2-2.2).

Όταν τόσο η Δ1 όσο και η Δ2 έχουν λάβει αυτή την επιβεβαίωση, τότε ισχύει ότι έχουν ληφθεί επιβεβαιώσεις από όλες τις διεργασίες για το (μ2-2.2).

Σε αυτό το σημείο, είναι γνωστό ότι η ενημέρωση που αντιπροσωπεύεται από το μ2 μπορεί να προχωρήσει τόσο στην Δ1 όσο και στην Δ2. Όπως φαίνεται και στην εικόνα 6.3 παραπάνω.

6.4 Διανυσματικά Ρολόγια

Το διανυσματικό ρολόι είναι ένας αλγόριθμος που παράγει μια μερική σειρά των γεγονότων και ανιχνεύει παραβιάσεις αιτιότητας σε ένα κατανεμημένο σύστημα. Αυτά τα ρολόγια επεκτείνονται στον κλιμακωτό χρόνο για να διευκολύνουν μια αιτιωδώς συνεπή άποψη του κατανεμημένου συστήματος, και ανιχνεύουν αν ένα συμβάν - γεγονός που συνέβηλε έχει προκαλέσει ένα άλλο συμβάν στο κατανεμημένο σύστημα. Ουσιαστικά αποτυπώνει όλες τις αιτιώδεις σχέσεις. Αυτός ο αλγόριθμος μας βοηθά να επισημάνουμε κάθε διεργασία με ένα διάνυσμα (μια λίστα ακεραίων αριθμών) με έναν ακέραιο αριθμό για κάθε τοπικό ρολόι κάθε διεργασίας εντός του συστήματος. Έτσι, για N δεδομένες διεργασίες, θα υπάρχει διάνυσμα/πίνακας μεγέθους N .

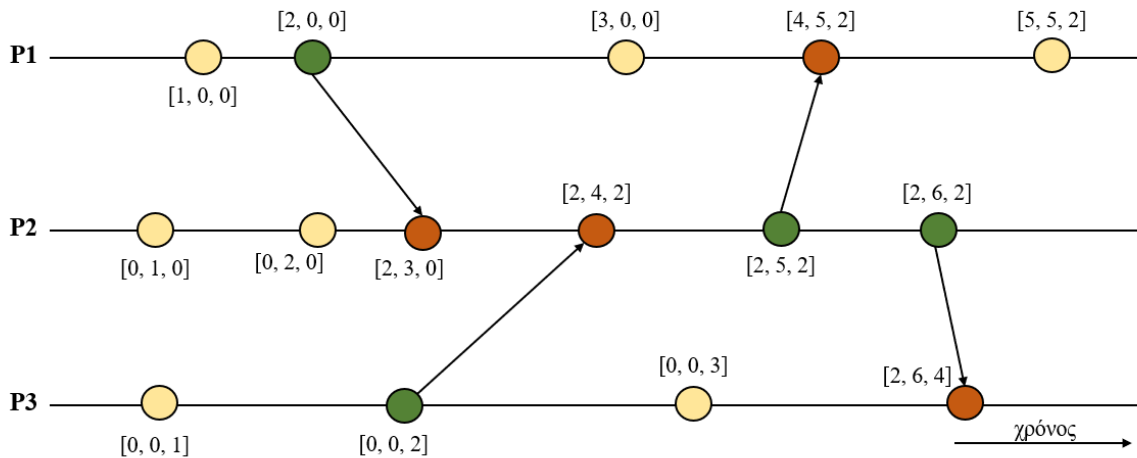
Αρχικά θα πρέπει να θέσουμε ότι κάποιο διανυσματικό ρολόι ενός συστήματος με n διεργασίες συμβολίζεται με VT και αντικατοπτρίζει ένα πίνακα με μέγεθος n . Όλες οι διεργασίες δ_j έχουν το δικό τους διανυσματικό ρολόι VT_j . Οι διεργασίες χρησιμοποιούν το διανυσματικό ρολόι για την δημιουργία χρονοσφραγίδων TT των τοπικών γεγονότων γ . Παρακάτω εξηγείται πώς λειτουργεί ο αλγόριθμος του διανυσματικού ρολογιού.

- Αρχικά, όλα τα ρολόγια είναι ρυθμισμένα στο μηδέν που πρακτικά σημαίνει $VT_j[j]=0$ για κάθε διεργασία δ_j και για κάθε j
- Κάθε φορά που συμβαίνει ένα εσωτερικό συμβάν όπως η αποστολή ενός μηνύματος ack - $send(ack)$ σε μια διεργασία δ_j , η τιμή του λογικού ρολογιού της διεργασίας στο διάνυσμα αυξάνεται κατά 1 που πρακτικά μεταφράζεται σε $VT_j[i]=VT_j[i]+1$
- Μαζί με την αποστολή ενός μηνύματος ack που στέλνει η διεργασία δ_j μεταφέρεται και η χρονοσφραγίδα $TT(ack)$ του γεγονότος - συμβάντος $send(ack)$.
- Επίσης, κάθε φορά που μια διεργασία στέλνει ένα μήνυμα, η τιμή του λογικού ρολογιού της διεργασίας στο διάνυσμα αυξάνεται κατά 1 που σημαίνει $VT_j[j]=\max\{VT_j[j], TT(ack)[j]\}$, $\forall j$, και $VT_j[i]=VT_j[i]+1$

Τη διανυσματική χρονοσφραγίδα τη συμβολίζουμε με $VT(\gamma)$. Όπως φαίνεται σε ένα γεγονός γ , έτσι έχουμε $TT(\gamma)=VT(\gamma)$. Μπορεί να αποδειχθεί ότι τα διανυσματικά ρολόγια ικανοποιούν τη λεγόμενη ισχυρή συνθήκη του ρολογιού (strong clock condition).

Η ισχυρή συνθήκη του ρολογιού είναι η : $\gamma \rightarrow \gamma' \Leftrightarrow VT(\gamma) \leq VT(\gamma')$

Κάθε φορά που μια διεργασία λαμβάνει ένα μήνυμα, η τιμή του λογικού ρολογιού των διεργασιών στο διάνυσμα αυξάνεται κατά 1, και, επιπλέον, κάθε στοιχείο ενημερώνεται λαμβάνοντας το μέγιστο της τιμής στο δικό του διανυσματικό ρολόι και της τιμής στο διάνυσμα στο μήνυμα που έχει ληφθεί (για κάθε στοιχείο).



ΣΚΟΥΡΟΧΡΩΜΟ ΚΥΚΛΑΚΙ ΕΙΝΑΙ ΓΕΓΟΝΟΣ ΤΥΠΟΥ ΛΗΨΗΣ (RECEIVE) ΜΗΝΥΜΑΤΟΣ
 ΜΠΕΖ ΚΥΚΛΑΚΙ ΕΙΝΑΙ ΕΣΩΤΕΡΙΚΟ ΓΕΓΟΝΟΣ
 ΠΡΑΣΙΝΟ ΚΥΚΛΑΚΙ ΕΙΝΑΙ ΓΕΓΟΝΟΣ ΤΥΠΟΥ ΑΠΟΣΤΟΛΗΣ (SEND) ΜΗΝΥΜΑΤΟΣ

Εικόνα 6.4 Παράδειγμα λειτουργίας διανυσματικού ρολογιού

Θεωρήστε μια διεργασία (P) με ένα διάνυσμα μεγέθους N για κάθε διεργασία: το παραπάνω σύνολο κανόνων που αναφέρθηκε πρέπει να εκτελεστεί από το διανυσματικό ρολόι:

Το παραπάνω παράδειγμα απεικονίζει τον μηχανισμό των διανυσματικών ρολογιών στον οποίο τα διανυσματικά ρολόγια ενημερώνονται μετά την εκτέλεση εσωτερικών γεγονότων, τα βέλη δείχνουν πώς οι τιμές των διανυσμάτων αποστέλλονται μεταξύ των διεργασιών (P₁, P₂, P₃).

Συνοψίζοντας, οι αλγόριθμοι διανυσματικών ρολογιών χρησιμοποιούνται στα κατακεμημένα συστήματα για να παρέχουν μια αιτιωδώς συνεπή διάταξη των γεγονότων, έτσι ολόκληρο το διάνυσμα αποστέλλεται σε κάθε διεργασία για κάθε μήνυμα που αποστέλλεται, προκειμένου να διατηρούνται τα διανυσματικά ρολόγια συγχρονισμένα.

6.5 Κατακεμημένος Υπολογισμός

Ένας κατακεμημένος υπολογισμός περιγράφει την εκτέλεση ενός κατακεμημένου προγράμματος μέσα σε ένα κατακεμημένο σύστημα από μια συλλογή διεργασιών. Η δραστηριότητα των διεργασιών [δ] μοντελοποιείται ως εκτέλεση μιας ακολουθίας γεγονότων [γ]. Ένα συμβάν - γεγονός μπορεί να είναι είτε εσωτερικό σε μια διεργασία και να προκαλεί μόνο μια τοπική αλλαγή κατάστασης, είτε να περιλαμβάνει επικοινωνία με μια άλλη διεργασία, δηλαδή επικοινωνία διαφορετικών μονάδων μέσω διεργασιών που εξυπηρετούν το ίδιο κατακεμημένο σύστημα.

Ας υποθέσουμε ότι η ακολουθία των γεγονότων αποστέλλω(μηνυμα m) και λαμβάνω(m) που αναφέρθηκε προηγουμένως ουσιαστικά πραγματοποιεί την επικοινωνία σε ένα κατακεμημένο σύστημα και ότι η αντιστοίχιση των γεγονότων βασίζεται στο αναγνωριστικό του μηνύματος που αποστέλλεται - γεγονός αποστέλλω(m). Με άλλα λόγια αυτό σημαίνει ότι, ακόμη και αν πολλές

διεργασίες στέλνουν την ίδια τιμή δεδομένων στην ίδια διεργασία, τα ίδια τα μηνύματα θα είναι μοναδικά. Εδώ μπορούμε να σκεφτούμε ότι, το γεγονός *αποστέλλω(m)* εγγράφει το μήνυμα σε ένα εξερχόμενο κανάλι για να γίνει η μετάδοση στη διεργασία προορισμού. Το συμβάν *λαμβάνω(m)*, από την άλλη πλευρά, αντιστοιχεί στην πράξη της αφαίρεσης - απόσυρσης (dequeuing) του μηνύματος από ένα εισερχόμενο κανάλι στη διεργασία προορισμού.

Σαφώς, για να συμβεί το γεγονός *λαμβάνω(m)* στη διεργασία $[δ]$, το μήνυμα $[m]$ πρέπει να έχει φτάσει στη διεργασία $[δ]$ και η $[δ]$ πρέπει να έχει δηλώσει την προθυμία της να λάβει ένα μήνυμα. Διαφορετικά, είτε το μήνυμα καθυστερεί (επειδή η διεργασία δεν είναι έτοιμη) είτε η διεργασία καθυστερεί (επειδή το μήνυμα δεν έχει φτάσει).

Αυτός ο τύπος επικοινωνίας στο επίπεδο συμβάντων αναφέρεται ως "message passing" και μπορεί να είναι αρκετά διαφορετική από εκείνη των ανώτερων επιπέδων του συστήματος. Η απομακρυσμένη επικοινωνία στο επίπεδο της γλώσσας προγραμματισμού μπορεί να επιτευχθεί μέσω οποιουδήποτε αριθμού παραδειγμάτων, συμπεριλαμβανομένων των απομακρυσμένων κλήσεων διαδικασιών, των μεταδόσεων (broadcasts), των κατανεμημένων συναλλαγών, των κατανεμημένων αντικειμένων ή της κατανεμημένης κοινής χρήσης της μνήμης. Στο επίπεδο που παρατηρούνται κατανεμημένοι υπολογισμοί, ωστόσο, όλες αυτές οι επικοινωνίες υψηλού επιπέδου καταλήγουν στη δημιουργία ταιριαστών γεγονότων αποστολής και λήψης σε ζεύγη διεργασιών.

6.5.1 Κατανεμημένες καταστάσεις

Κάθε κατανεμημένο σύστημα διαθέτει έναν αριθμό διεργασιών που εκτελούνται σε έναν αριθμό διαφορετικών φυσικών διακομιστών. Αυτές οι διεργασίες επικοινωνούν μεταξύ τους μέσω καναλιών επικοινωνίας με τη χρήση μηνυμάτων κειμένου. Οι διεργασίες αυτές δεν έχουν ούτε κοινή μνήμη ούτε κοινό φυσικό ρολόι, γεγονός που καθιστά δύσκολη τη διαδικασία προσδιορισμού της στιγμιαίας συνολικής κατάστασης.

Βέβαια για να γίνει η καταγραφή μιας κατάστασης ενός κατανεμημένου συστήματος πρέπει να γνωρίζουμε ότι :

- Δεν γίνεται να γίνει ταυτόχρονα προσωρινή παύση των διεργασιών.
- Εφόσον μιλάμε για κατανεμημένα συστήματα, οι διεργασίες μπορεί να εκτελούνται σε μονάδες (κόμβους) διαφορετικές.
- Δεν υπάρχει ταυτόχρονη παράδοση των μηνυμάτων ελέγχου, που στέλνει η κάθε μονάδα (κόμβος).
- Υπάρχει και το ζήτημα του συγχρονισμού που αναλύθηκε στις αρχές του κεφαλαίου, όπου αναφερόμαστε στην δυσκολία συντονισμού των διεργασιών σε μία προκαθορισμένη στιγμή.

Εδώ αξίζει να αναφέρουμε ότι σε πραγματικές συνθήκες είναι μεγάλης σημασίας να μην γίνεται παύση της εκτέλεσης, ούτε καν για προσωρινό χρονικό διάστημα. Επιπλέον η καταγραφή της κατάστασης σε ένα κατανεμημένο σύστημα έχει διάφορες εφαρμογές σε διάφορα πραγματικά σενάρια. Ένα σενάριο είναι η χρήση σε συστήματα που έχουν ως σκοπό την μέτρηση ποσοτήτων που αλλάζουν δυναμικά. Ένα τέτοιο παράδειγμα είναι τα χρήματα που υπάρχουν σε έναν τραπεζικό λογαριασμό. Ένα άλλο παράδειγμα είναι τα αντίγραφα ασφαλείας που μπορούν να ενημερώνονται σε διαφορετικές χρονικές στιγμές από διαφορετικές μονάδες. Επιπλέον η καταγραφή κατάστασης χρησιμοποιείται σε συστήματα ανίχνευσης τερματισμού αλλά και στους ίδιους τους κατανεμημένους αλγόριθμους κατά τη διάρκεια της αποσφαλμάτωσης για τον έλεγχο, εάν υπάρχει παραβίαση κάποια συνθήκης.

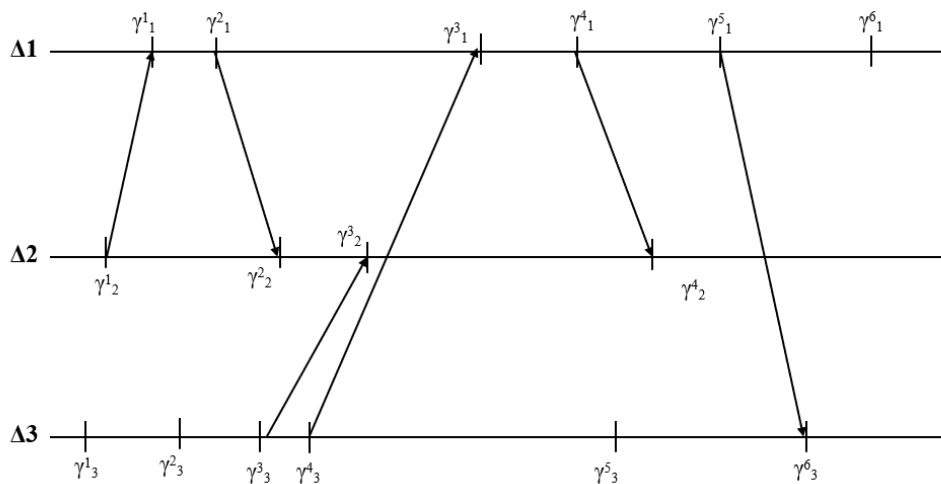
Ανάγκη λήψης στιγμιότυπου ή καταγραφής της συνολικής κατάστασης του συστήματος αφορά κυρίως :

- Σημεία ελέγχου: βοηθά στη δημιουργία σημείου ελέγχου. Εάν η εφαρμογή αποτύχει κατά κάποιο τρόπο, αυτό το σημείο ελέγχου μπορεί να επαναχρησιμοποιηθεί.
- Συλλογή σκουπιδιών: χρησιμοποιείται για την αφαίρεση/διαγραφή αντικειμένων που δεν έχουν αναφορές.
- Μπορεί να χρησιμοποιηθεί στην ανίχνευση αδιεξόδων και τερματισμού.
- Είναι επίσης χρήσιμη σε άλλες εργασίες εντοπισμού σφαλμάτων.

Καθολικές καταστάσεις

Για την καλύτερη κατανόηση των καθολικών καταστάσεων θα πρέπει αρχικά να παρουσιαστούν κάποιες έννοιες. Οι έννοιες αυτές ουσιαστικά συγκροτούν την έννοια των καθολικών καταστάσεων (global states). Πιο συγκεκριμένα παρουσιάζονται οι ορισμοί της τοπικής ιστορίας (local history), της καθολικής ιστορίας (global history), της τοπικής κατάστασης (local state) και της καθολικής κατάστασης.

Τοπική ιστορία μιας διεργασίας (Δ) ορίζεται ως μία ακολουθία γεγονότων (γ) που εκτελούνται κατά χρονική σειρά στα πλαίσια αυτής της διεργασίας. Για τις ανάγκες του παραδείγματος ορίζουμε την τοπική ιστορία με (TI). Η τοπική ιστορία κάθε διεργασίας είναι διατεταγμένη πλήρως με βάση τον χρόνο. Παρακάτω στην εικόνα 6.5 παρουσιάζεται παράδειγμα τοπικής ιστορίας μέσα από ένα διάγραμμα χώρου-χρόνου κατανεμημένου υπολογισμού.



Εικόνα 6.5 Παράδειγμα διάγραμματος χώρου-χρόνου κατανεμημένου υπολογισμού

$$TI(1) = \gamma^1_1, \gamma^2_1, \gamma^3_1, \gamma^4_1, \gamma^5_1, \gamma^6_1$$

$$TI(2) = \gamma^1_2, \gamma^2_2, \gamma^3_2, \gamma^4_2$$

$$TI(3) = \gamma^1_3, \gamma^2_3, \gamma^3_3, \gamma^4_3, \gamma^5_3, \gamma^6_3$$

Να διευκρινίσουμε πως ο δείκτης του γεγονότος (γ) συμβολίζει τον αριθμό της διεργασίας που εξυπηρετεί και ο εκθέτης την διατεταγμένη σειρά του με βάση τον χρόνο σε σχέση με τα άλλα γεγονότα της ίδια διεργασίας.

Τοπική κατάσταση μιας διεργασίας Δ_i ορίζεται ως η κατάσταση αμέσως μετά την εκτέλεση ενός γεγονότος γ^i_j και την συμβολίζουμε με s^i_j . Η αρχική κατάσταση της διεργασίας πριν την εκτέλεση κάποιου γεγονότος συμβολίζεται με γ^0_j .

Η τοπική κατάσταση περιέχει τις τιμές των τοπικών μεταβλητών της διεργασίας, όπως και υπό συνθήκες, όπως για παράδειγμα ζητήματα που αφορούν την μνήμη, μπορεί να περιλαμβάνει και αρχεία που επηρεάζουν κάποιες εσωτερικές λειτουργίες του τοπικού λειτουργικού συστήματος.

Μια διεργασία θα μπορούσε να καταγράψει τη δική της τοπική κατάσταση σε μια δεδομένη χρονική στιγμή, αλλά τα μηνύματα που βρίσκονται καθ' οδόν για να παραδοθούν δεν θα συμπεριλαμβάνονταν στην καταγεγραμμένη κατάσταση και ως εκ τούτου η πραγματική κατάσταση του συστήματος θα ήταν εσφαλμένη μετά την παράδοση του μηνύματος που βρίσκεται καθ' οδόν για να παραδοθεί.

Καθολική ιστορία (KI) ως όρος χρησιμοποιείται για την περιγραφή της ένωσης των τοπικών ιστοριών των επιμέρους διεργασιών. $KI = Tl(1) \cup Tl(2) \cup \dots \cup Tl(N)$

Καθολική κατάσταση (KK) είναι η κατάσταση ενός κατανεμημένου συστήματος που περιλαμβάνει την ένωση τοπικών καταστάσεων, όλων των διεργασιών που εκτελούνται στα πλαίσια ενός κατανεμημένου συστήματος. Αποτελεί ένα n-διάστατο διάνυσμα των τοπικών καταστάσεων $KK = (k_1, \dots, k_n)$, έχοντας μία τοπική κατάσταση για κάθε διεργασία. Κοιτάζοντας την εικόνα 6.5 παραπάνω προκύπτουν κάποιες καθολικές καταστάσεις.

$$KK1 = \{k_{1,1}^3, k_{2,2}^3, k_{3,3}^4\}$$

$$KK2 = \{Y_{1,1}^5, Y_{2,2}^4, Y_{3,3}^6\}$$

Είναι φανερό ότι υπάρχουν και άλλες δυνατές καθολικές καταστάσεις.

Στιγμιότυπο κατανεμημένου συστήματος με χρήση τομής

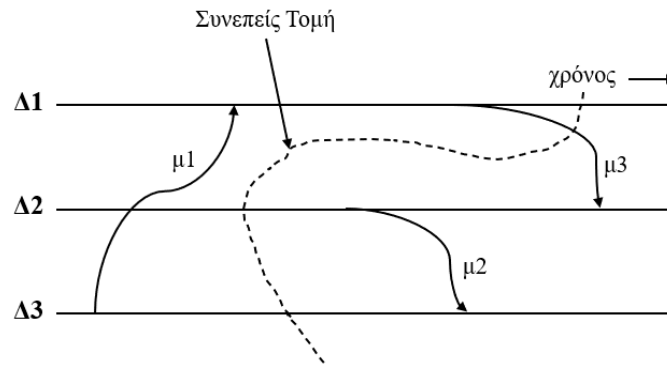
Για να πάρουμε ένα στιγμιότυπο όλων των διεργασιών χρησιμοποιείται μια τεχνική που ονομάζεται *τομή (cut)[C]*. Μια τομή προσδιορίζεται από ένα σύνολο στοιχείων της διεργασίας, της οποίας τα στοιχεία αυτά υποδηλώνουν τα τελευταία γεγονότα κάθε διεργασίας (ονομάζεται *σύνορο -frontier*).

Άρα τομή ονομάζουμε ένα υποσύνολο της καθολικής ιστορίας (KI) που αποτελείται από $\gamma_i \geq 0$ (με i τον χαρακτηριστικό αριθμό της διεργασίας Δ) αρχικά γεγονότα από κάθε διεργασία Δ_i .

Η μαθηματική απεικόνιση είναι T δηλ. $T = Tl_1^{\gamma_1} \cup Tl_2^{\gamma_2} \cup \dots \cup Tl_i^{\gamma_i}$. Επομένως, μια τομή προσδιορίζεται μέσω του διανύσματος $\{\gamma(1), \dots, \gamma(i)\}$.

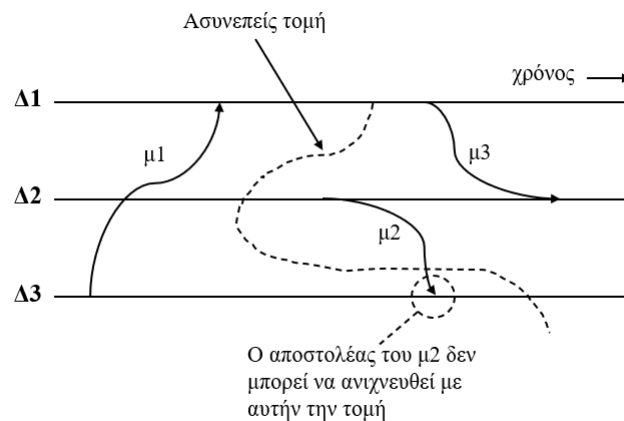
Το μέγιστο αυτών των τιμών αποτελεί το σύνορο $\{\max[\gamma(1)], \dots, \max[\gamma(i)]\}$.

Υπάρχουν δύο τύποι τομών. Οι συνεπείς (consistent) και οι ασυνεπείς (inconsistent) τομές. Μια συνεπής καθολική κατάσταση αντιστοιχεί σε μια τομή στην οποία κάθε μήνυμα που ελήφθη στο παρελθόν της τομής, εστάλη στο παρελθόν της εν λόγω τομής. Με τον όρο παρελθόν εννοούμε οτιδήποτε συμβαίνει αριστερά της τομής. Μια τέτοια τομή είναι γνωστή ως συνεπής τομή. Το αντίστοιχο παράδειγμα φαίνεται στην εικόνα 6.6.



Εικόνα 6.6 Παράδειγμα συνεπής τομής [6]

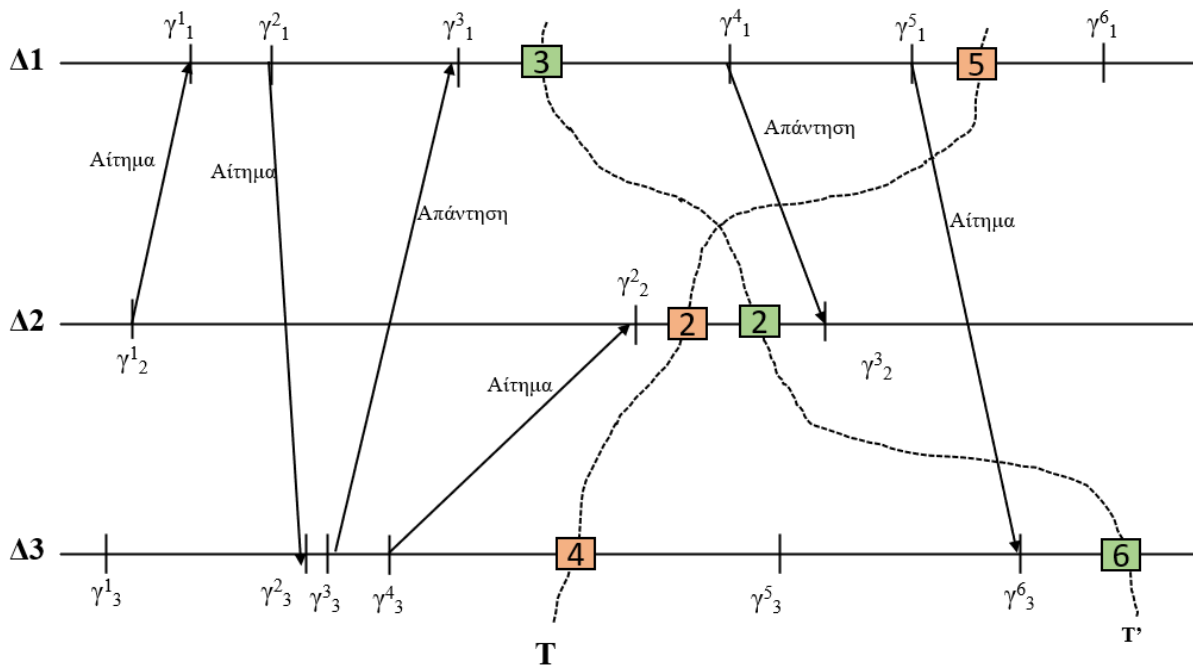
Μια ασυνεπής καθολική κατάσταση αντιστοιχεί σε μια τομή στην οποία κάθε μήνυμα που ελήφθη στο ΠΑΡΕΛΘΟΝ της τομής μπορεί να εστάλη και χρονικά αργότερα του συνόρου της τομής. Έτσι ο αποστολέας του μηνύματος δεν μπορεί να αναγνωρισθεί. Το αντίστοιχο παράδειγμα φαίνεται στην εικόνα 6.7.



Εικόνα 6.7 Παράδειγμα ασυνεπής τομής [6]

Στην παρακάτω εικόνα 6.8 υπάρχουν δύο τομές. Η πρώτη τομή είναι η $T(5, 2, 4)$ με σύνορο τομής τα γεγονότα $\{\gamma^5_1, \gamma^2_2, \gamma^4_3\}$ ενώ η δεύτερη τομή είναι η $T'(3, 2, 6)$ με σύνορο τομής τα γεγονότα $\{\gamma^3_1, \gamma^2_2, \gamma^6_3\}$. Για την πρώτη τομή, συμπεριλαμβάνουμε τα γεγονότα (γ) του $\Delta 1$ μέχρι (και συμπεριλαμβανομένου) του γεγονότος $\gamma 5$. Για την ίδια τομή, συμπεριλαμβάνουμε τα γεγονότα του $\Delta 2$ μέχρι (και συμπεριλαμβανομένου) του γεγονότος $\gamma 2$ και του $\Delta 3$ μέχρι το $\gamma 4$.

Η $T'(3, 2, 6)$ λόγω του $\gamma 6$ της $\Delta 3$ είναι ασυνεπής άρα δεν αποτελεί μέρος καθολικής κατάστασης και ούτε μπορεί να είναι μέσα στα γεγονότα μιας καθολικής εκτέλεσης. Για την εκτέλεση καθολικών καταστάσεων κάνουμε ανάλυση της ακριβώς παρακάτω.



Εικόνα 6.8 Τομές ενός καταναμημένου υπολογισμού

Εκτέλεση και πλέγμα

Μια εκτέλεση (run) ενός καταναμημένου υπολογισμού είναι η συνολική διάταξη (R) που περιλαμβάνει όλα τα γεγονότα του συνολικού ιστορικού και είναι συνεπής με κάθε τοπικό ιστορικό. Με άλλα λόγια, για κάθε διεργασία (Δ), τα γεγονότα της (γ) εμφανίζονται με την ίδια σειρά που εμφανίζονται στο (R). Σημειώστε ότι μια εκτέλεση δεν είναι απαραίτητο να αντιστοιχεί σε οποιαδήποτε πιθανή εκτέλεση και ένας καταναμημένος υπολογισμός μπορεί να έχει πολλές εκτελέσεις, καθεμία από τις οποίες αντιστοιχεί σε διαφορετική εκτέλεση. Παρακάτω παρουσιάζεται παράδειγμα εκτέλεσης καταναμημένου υπολογισμού που αφορά τις διεργασίες της εικόνας 6.5.

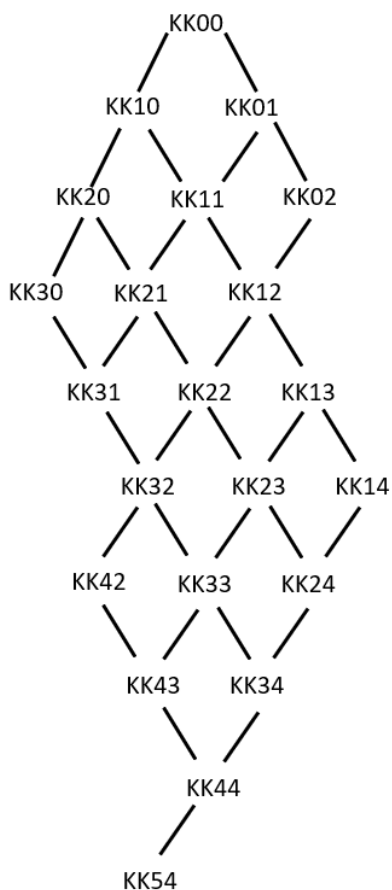
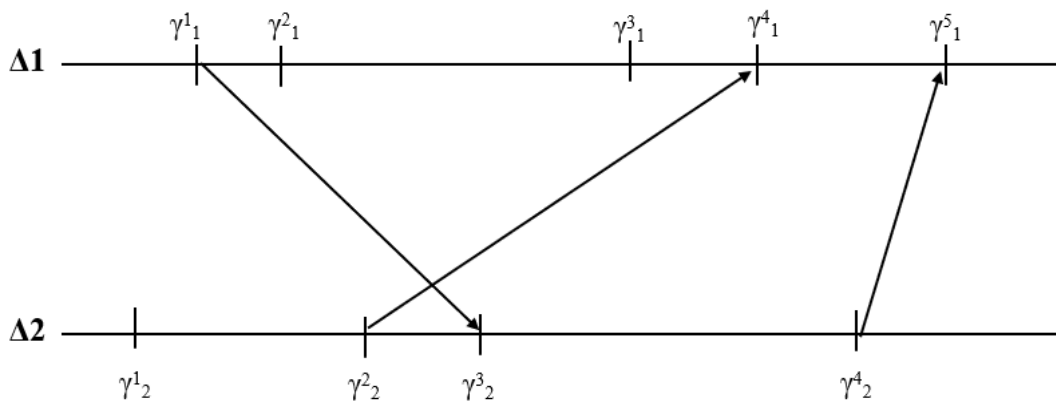
$$R = \{\gamma^1_3, \gamma^1_2, \gamma^1_1, \gamma^2_3, \gamma^2_1, \gamma^2_2, \gamma^3_3, \gamma^3_2, \gamma^3_1, \gamma^4_3, \gamma^4_1, \gamma^4_2, \gamma^5_3, \gamma^5_1, \gamma^6_3, \gamma^6_1\}$$

Στο συγκεκριμένο παράδειγμα εκτέλεσης έχουμε μία ακολουθία συνεπών καθολικών καταστάσεων που σημαίνει ταυτόχρονα ότι έχουμε μία συνεπή εκτέλεση.

Το πλέγμα (lattice) αποτελείται από το σύνολο όλων των δυνατών συνεπών καθολικών καταστάσεων σε σχέση πάντα με τις συνεπείς εκτελέσεις που προκύπτουν. Το πλέγμα σαν σχηματισμός έχει κάποια χαρακτηριστικά.

- Κάθε ακμή του είναι μία καθολική κατάσταση (ΚΚ)
- Κάθε ακμή του έχει έναν συμβολισμό $ΚΚ^{xy}$ όπου με x αναφερόμαστε στον αριθμό των γεγονότων της μιας διεργασίας και με γ στην άλλη διεργασία.
- Εάν είχαμε 5 διεργασίες θα είχαμε $ΚΚ^{xyvwz}$
- Η μέτρηση του αριθμού των γεγονότων κάθε διεργασίας ξεκινάει από το "0"

- Το άθροισμα των αριθμων “xy” του στοιχείου KK^{xy} θα πρέπει να είναι ίδιο με το αντίστοιχο άθροισμα των στοιχείων που βρίσκονται στην ίδια σειρά.



Εικόνα 6.8 Παράδειγμα διάγραμματος χώρου-χρόνου και αντίστοιχου πλέγματος

Αλγόριθμος των Chandy και Lamport καταγραφής συνεπούς καθολικής κατάστασης

Οι Chandy και Lamport ήταν οι πρώτοι που πρότειναν έναν αλγόριθμο για την καταγραφή της συνεπούς καθολικής κατάστασης ενός κατανεμημένου συστήματος. Η βασική ιδέα πίσω από τον προτεινόμενο αλγόριθμο είναι ότι αν γνωρίζουμε ότι όλα τα μηνύματα που έχουν σταλεί από μια διεργασία έχουν ληφθεί από μια άλλη, τότε μπορούμε να καταγράψουμε την καθολική κατάσταση του συστήματος.

Εδώ πρέπει να αναφέρουμε ότι η καθολική κατάσταση (global state) του συστήματος αποτελείται από την ένωση των τοπικών καταστάσεων (local state) των επιμέρους διεργασιών. Η τοπική κατάσταση μιας διεργασίας αποτελείται από τις τιμές όλων των τοπικών μεταβλητών της. Επιπλέον, μπορεί να περιλαμβάνει πράγματα, όπως αρχεία που θα μπορούσαν να έχουν αντίκτυπο στο τοπικό λειτουργικό της σύστημα.

Οποιαδήποτε διεργασία στο κατανεμημένο σύστημα μπορεί να ξεκινήσει αυτόν τον αλγόριθμο καταγραφής της συνολικής κατάστασης χρησιμοποιώντας ένα ειδικό μήνυμα που ονομάζεται MARKER. Αυτός ο δείκτης διασχίζει το κατανεμημένο σύστημα σε όλα τα κανάλια επικοινωνίας και προκαλεί κάθε διεργασία να καταγράψει τη δική της κατάσταση. Στο τέλος, καταγράφεται η κατάσταση ολόκληρου του συστήματος (καθολική κατάσταση). Αυτός ο αλγόριθμος δεν παρεμβαίνει στην κανονική εκτέλεση των διεργασιών.

Παραδοχές του αλγορίθμου:

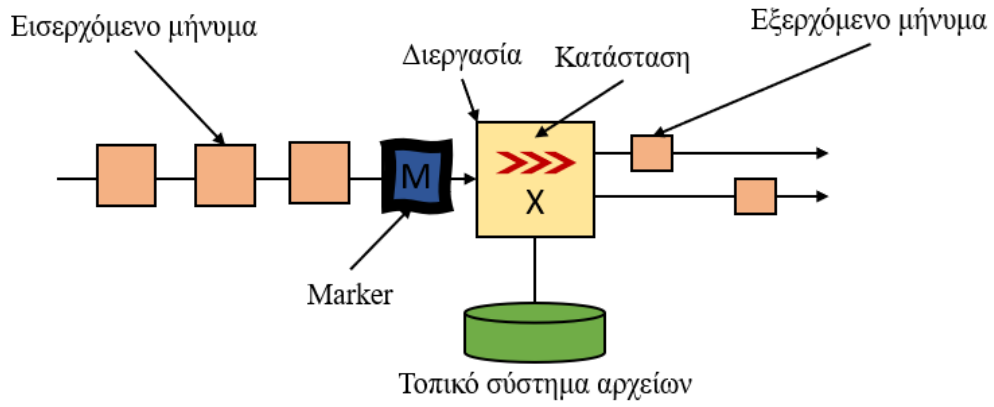
- Υπάρχει πεπερασμένος αριθμός διεργασιών στο κατανεμημένο σύστημα και οι διεργασίες αυτές δεν μοιράζονται μνήμη και ρολόγια.
- Υπάρχει πεπερασμένος αριθμός καναλιών επικοινωνίας και είναι μονής κατεύθυνσης και FIFO διατεταγμένος.
- Υπάρχει ένα μονοπάτι επικοινωνίας μεταξύ δύο οποιωνδήποτε διεργασιών στο σύστημα.
- Σε ένα κανάλι, τα μηνύματα λαμβάνονται με την ίδια σειρά που αποστέλλονται.

Αλγόριθμος

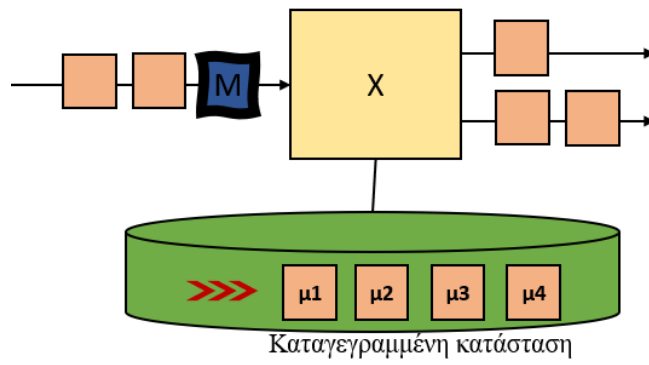
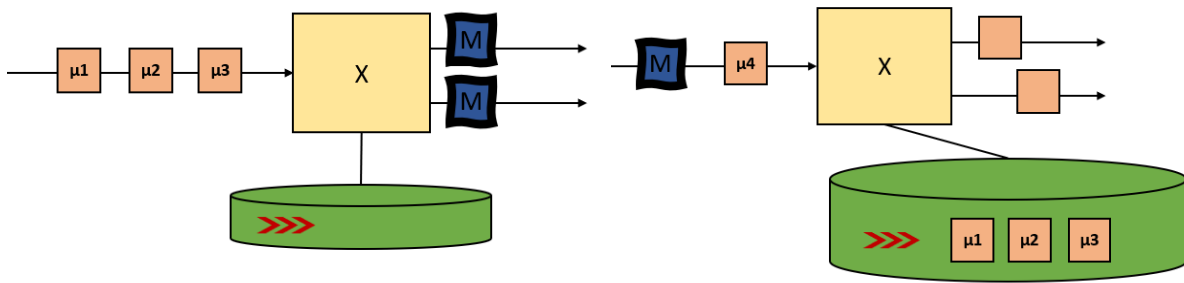
Σε αυτό σημείο περιγράφεται το πως εκτελείται ο αλγόριθμος (εικόνα 6.10)

- Η διεργασία X ξεκινά καταγράφοντας τη δική της τοπική κατάσταση
- Η X στη συνέχεια στέλνει έναν δείκτη MARKER κατά μήκος κάθε εξερχόμενου καναλιού της
- Όταν η διεργασία X λαμβάνει έναν δείκτη MARKER μέσω του καναλιού K, η ενέργειά της εξαρτάται από το αν είχε ήδη καταγράψει την τοπική της κατάσταση:
 - ο Εάν δεν έχει ακόμη καταγραφεί, τότε καταγράφει την τοπική της κατάσταση (η κατάσταση αμέσως μετά την εκτέλεση ενός γεγονότος), στέλνει τον δείκτη MARKER κατά μήκος κάθε εξερχόμενου καναλιού της.
 - ο Εάν έχει ήδη καταγραφεί, τότε ο δείκτης MARKER στο κανάλι K υποδεικνύει ότι η κατάσταση του καναλιού πρέπει να καταγραφεί, δηλαδή όλα τα μηνύματα που έχουν ληφθεί πριν από αυτόν τον δείκτη MARKER και τη στιγμή που η X κατέγραψε τη δική της κατάσταση.
- Η X έχει τελειώσει, όταν έχει λάβει έναν δείκτη MARKER κατά μήκος του καθενός από τα εισερχόμενα κανάλια της.

Οπότε έχουμε μια συνεπή καθολική κατάσταση διότι γνωρίζουμε ότι όλα τα μηνύματα που έχουν σταλεί από μια διεργασία έχουν ληφθεί από μια άλλη.



(α)



(δ)

Εικόνα 6.10 Αλγόριθμος των Chandy και Lamport [<https://www.slideshare.net/sandpooonia/6-26928964>]

Βιβλιογραφικές αναφορές

- [1] Andrew S. Tanenbaum, Maarten Van Steen (2006), Κατανεμημένα Συστήματα: Αρχές και Υποδείγματα, Έκδοση: 1η/2006, Εκδόσεις Κλειδάριθμος.
- [2] Coulouris, J. Dollimore, T. Kindberg, G. Blair (2020), Κατανεμημένα Συστήματα, Έκδοση: 2η, Da Vinci M.E.Π.Ε.
- [3] Κάβουρας Ι.Κ., Μήλης Ι.Ζ., Ρουκουνάκη Α.Α., Ξηλωμένος Γ.Β. (2011), Κατανεμημένα συστήματα σε Java, 3η έκδοση, Εκδόσεις Κλειδάριθμος ΕΠΕ.
- [4] Dona, S. (1993). Consistent Global States of Distributed Systems : Fundamental Concepts and Mechanisms .
- [5] Chandy, K. M., & Lamport, L. (1985). Distributed Snapshots: Determining Global States of Distributed Systems. ACM Transactions on Computer Systems (TOCS), 3(1), 63–75. doi:10.1145/214451.214456
- [6] Systems, O. (2013). Lecture 14 : March 11 Election Algorithms. 1–5.

Κριτήρια αξιολόγησης

Ερώτηση 1

Ποιές από τις παρακάτω αποτελούν ιδιότητες για την διατήρηση του συγχρονισμού ρολογιού; (παραπάνω από μία σωστές απαντήσεις)

α. Οι πληροφορίες θα είναι διάσπαρτες μεταξύ πολλαπλών μηχανών.

β. Δεν υπάρχει δυνατότητα ύπαρξης κοινού ρολογιού ή οποιαδήποτε άλλη καθολική πηγή χρόνου

γ. Υπάρχει δυνατότητα ύπαρξης κοινού ρολογιού ή οποιαδήποτε άλλη καθολική πηγή χρόνου

δ. Οι διεργασίες λαμβάνουν την οποιαδήποτε απόφαση μόνο με βάση τις τοπικές πληροφορίες

ε. Πρέπει να αποφεύγεται το να υπάρχει ένα μόνο σημείο αποτυχίας (single point of failure)

στ. Οι διεργασίες λαμβάνουν την οποιαδήποτε απόφαση με βάση τοπικές και καθολικές πληροφορίες

Ερώτηση 2

Ποιές από τις παρακάτω προτάσεις είναι σωστή;

α. Ένα πλεονέκτημα της συγχρονισμένης μετάδοσης είναι ότι μιλάμε για μια διαδικασία με μηδέν υπολογιστικό κόστος.

β. Ένα πλεονέκτημα της συγχρονισμένης μετάδοσης είναι ότι μιλάμε για μια γρήγορη διαδικασία.

γ. Ένα πλεονέκτημα της μη συγχρονισμένης μετάδοσης είναι ότι μιλάμε για μια γρήγορη διαδικασία.

δ. Ένα μειονέκτημα της συγχρονισμένης μετάδοσης είναι ότι μιλάμε για μια χρονοβόρα διαδικασία

Ερώτηση 3

Η ασύγχρονη μετάδοση είναι ο τύπος μετάδοσης κατά τον οποίο:

α. ο αποστολέας και ο παραλήπτης έχουν τα δικά τους εσωτερικά ρολόγια- επομένως δεν χρειάζονται εξωτερικό κοινό παλμό ρολογιού.

β. μόνο ο αποστολέας έχει δικό του εσωτερικό ρολόι και με τον εξωτερικό παλμό διαχειρίζεται το ρολόι του παραλήπτη.

γ. μόνο ο παραλήπτης έχει δικό του εσωτερικό ρολόι και με τον εξωτερικό παλμό διαχειρίζεται το ρολόι του αποστολέα.

δ. ο αποστολέας και ο παραλήπτης δεν έχουν τα δικά τους εσωτερικά ρολόγια - επομένως χρειάζονται εξωτερικό κοινό παλμό ρολογιού.

Ερώτηση 4

Ποιές από τις παρακάτω προτάσεις είναι σωστή;

α. Ένα μειονέκτημα των ασύγχρονων μεταδόσεων είναι ότι χρειάζονται μεγάλο υπολογιστικό κόστος.

β. Ένα πλεονέκτημα της συγχρονισμένης μετάδοσης είναι ότι μιλάμε για μια χρονοβόρα διαδικασία.

γ. Ένα πλεονέκτημα των ασύγχρονων μεταδόσεων είναι ότι χρειάζονται μικρό χρονικό διάστημα για να ολοκληρωθούν.

δ. Ένα μειονέκτημα των ασύγχρονων μεταδόσεων είναι ότι χρειάζονται μεγάλο χρονικό διάστημα για να ολοκληρωθούν.

Ερώτηση 5

Εσωτερικός συγχρονισμός ρολογιού είναι αυτός στον οποίο:

α. κάθε κόμβος μοιράζεται την ώρα του με άλλους κόμβους και όλοι οι κόμβοι ρυθμίζουν και προσαρμόζουν τις ώρες τους αναλόγως.

β. μόνο ο κεντρικός κόμβος μοιράζεται την ώρα του με άλλους κόμβους και όλοι οι κόμβοι ρυθμίζουν και προσαρμόζουν τις ώρες τους αναλόγως

γ. μόνο ο κεντρικός κόμβος μοιράζεται την ώρα του με άλλους κόμβους και αυτοί ενημερώνουν το εξωτερικό ρολόι αναφοράς

δ. κάθε κόμβος μοιράζεται την ώρα του με άλλους κόμβους και όλοι οι κόμβοι ενημερώνουν το εξωτερικό ρολόι αναφοράς

Ερώτηση 6

Εξωτερικός συγχρονισμός ρολογιού είναι αυτός στον οποίο:

α. υπάρχει ένα εξωτερικό ρολόι αναφοράς. Το ρολόι αυτό χρησιμοποιείται ως αναφορά και οι κόμβοι του συστήματος ρυθμίζουν και προσαρμόζουν τις ώρες τους αναλόγως.

β. μόνο ο κεντρικός κόμβος μοιράζεται την ώρα του με άλλους κόμβους και όλοι οι κόμβοι ρυθμίζουν και προσαρμόζουν τις ώρες τους αναλόγως

γ. μόνο ο κεντρικός κόμβος μοιράζεται την ώρα του με άλλους κόμβους και αυτοί ενημερώνουν το λεγόμενο “εσωτερικό ρολόι αναφοράς”

δ. κάθε κόμβος μοιράζεται την ώρα του με άλλους κόμβους και όλοι οι κόμβοι ρυθμίζουν και προσαρμόζουν τις ώρες τους αναλόγως.

Ερώτηση 7

Ο κεντροποιημένος συγχρονισμός ρολογιού εξαρτάται από τον ενιαίο διακομιστή ώρα, οπότε αν ο κόμβος αυτός αποτύχει, ολόκληρο το σύστημα θα χάσει τον συγχρονισμό.

α. Σωστό

β. Λάθος

Ερώτηση 8

Ο αλγόριθμος του Cristian: (πάνω από μία σωστές απαντήσεις)

α. Υποθέτει ότι υπάρχουν διαθέσιμες πρόσθετες πληροφορίες.

β. Δεν περιορίζει τον αριθμό των μετρήσεων για την εκτίμηση της τιμής.

γ. Υποθέτει ότι δεν υπάρχουν διαθέσιμες πρόσθετες πληροφορίες.

δ. Περιορίζει τον αριθμό των μετρήσεων για την εκτίμηση της τιμής.

Ερώτηση 9

Στον αλγόριθμο του Berkeley:

- α. οι κόμβοι πολυ-εκπέμπουν τα μηνύματα στο διακομιστή ώρας.
- β. ο διακομιστής ώρας στέλνει περιοδικά πολλαπλά μηνύματα στον κεντρικό κόμβο.
- γ. ο διακομιστής ώρας στέλνει περιοδικά ένα μήνυμα σε όλους τους κόμβους.**
- δ. Δεν υπάρχει διακομιστής ώρας

Ερώτηση 10

Ποιές από τις παρακάτω προτάσεις είναι σωστή για τον NTP: (πάνω από μία σωστές απαντήσεις)

- α. Δεν χρησιμοποιείται σε συστήματα αυθεντικοποίησης.
- β. Δεν παρέχει επιτάχυνση δικτύου.
- γ. Χρησιμοποιείται σε συστήματα αρχείων που είναι εύκολα στο συγχρονισμό μέσω δικτύου.
- δ. Όταν οι διακομιστές είναι εκτός λειτουργίας επηρεάζεται ο χρόνος συγχρονισμού σε όλη την τρέχουσα επικοινωνία.**
- ε. Οι διακομιστές είναι επιρρεπείς σε σφάλματα λόγω των διαφόρων ζωνών ώρας και μπορεί να προκύψουν συγκρούσεις**

Ερώτηση 11

Ποιές από τις παρακάτω προτάσεις είναι σωστή για τον NTP (πάνω από μία σωστές απαντήσεις)

- α. Ελάχιστη μείωση της ακρίβειας του χρόνου.**
- β. Όταν μειώνονται τα πακέτα NTP ο συγχρονισμός συγκρούεται.
- γ. Μπορεί να γίνει χειραγώγηση στο συγχρονισμό.**
- δ. Δεν παρέχει διαδικτυακό συγχρονισμό μεταξύ των συσκευών.
- ε. Δεν παρέχει αυξημένη ασφάλεια εντός των εγκαταστάσεων.

Ερώτηση 12

Στον αλγόριθμο Lamport:

- α. δεν έχει σημασία η ακριβής ώρα εκτέλεσης των γεγονότων, ούτε η χρονική σειρά με την οποία τα γεγονότα αυτά συμβαίνουν.
- β. δεν έχει σημασία η ακριβής ώρα εκτέλεσης των γεγονότων, αλλά κυρίως η χρονική σειρά με την οποία τα γεγονότα αυτά συμβαίνουν.**
- γ. έχει σημασία η ακριβής ώρα εκτέλεσης των γεγονότων, και όχι η χρονική σειρά με την οποία τα γεγονότα αυτά συμβαίνουν.
- δ. έχει σημασία η ακριβής ώρα εκτέλεσης των γεγονότων, και η χρονική σειρά με την οποία τα γεγονότα αυτά συμβαίνουν.

Ερώτηση 13

Στα διανυσματικά ρολόγια κάθε φορά που συμβαίνει ένα εσωτερικό συμβάν, όπως η αποστολή ενός μηνύματος σε μια διεργασία η τιμή του λογικού ρολογιού της διεργασίας στο διάνυσμα:

- α. αποθηκεύεται στον κεντρικό κόμβο.
- β. ξεκινάει πάλι από το 0 (reset)
- γ. αυξάνεται κατά 1**
- δ. μειώνεται κατά 1

Ερώτηση 14

Ποιες από τις παρακάτω αποτελούν παραδοχές του αλγορίθμου των Chandy και Lamport; (πάνω από μία σωστές απαντήσεις)

- α. Υπάρχει πεπερασμένος αριθμός διεργασιών στο κατανεμημένο σύστημα και οι διεργασίες αυτές δεν μοιράζονται μνήμη και ρολόγια.**
- β. Υπάρχει πεπερασμένος αριθμός καναλιών επικοινωνίας και είναι μονής κατεύθυνσης και LIFO διατεταγμένος.
- γ. Υπάρχει ένα μονοπάτι επικοινωνίας μεταξύ δύο οποιωνδήποτε διεργασιών στο σύστημα.**
- δ. Σε ένα κανάλι, τα μηνύματα λαμβάνονται με διαφορετική σειρά από αυτήν που αποστέλλονται.

Ερώτηση 15

Το πλέγμα (lattice) αποτελείται από το υποσύνολο ενός δείγματος των δυνατών συνεπών καθολικών καταστάσεων σε σχέση πάντα με τις συνεπίεις εκτελέσεις που προκύπτουν.

α. Σωστό

β. Λάθος

Ερώτηση 16

Αντιστοιχίστε τις έννοιες της πρώτης στήλης με τους ορισμούς της δεύτερης.

A ΣΤΗΛΗ

- α. Τοπική ιστορία
- β. Τοπική κατάσταση
- γ. Καθολική ιστορία
- δ. Καθολική κατάσταση
- ε. Συνεπής τομή
- στ. Ασυνεπής τομή

B ΣΤΗΛΗ

1. είναι ο όρος που χρησιμοποιείται για την περιγραφή της ένωσης των τοπικών ιστοριών των επιμέρους διεργασιών.
2. ορίζεται ως μία ακολουθία γεγονότων που εκτελούνται κατα χρονική σειρά στα πλαίσια μίας διεργασίας.
3. είναι η κατάσταση ενός κατανεμημένου συστήματος που περιλαμβάνει την ένωση τοπικών καταστάσεων, όλων των διεργασιών που εκτελούνται στα πλαίσια ενός κατανεμημένου συστήματος.
4. ορίζεται ως η κατάσταση αμέσως μετά την εκτέλεση ενός γεγονότος.
5. μια τομή στην οποία κάθε μήνυμα που ελήφθη στο ΠΑΡΕΛΘΟΝ της τομής, εστάλη στο ΠΑΡΕΛΘΟΝ της εν λόγω τομής.
6. μια τομή στην οποία κάθε μήνυμα που ελήφθη στο ΠΑΡΕΛΘΟΝ της τομής μπορεί να εστάλη και χρονικά αργότερα του συνόρου της τομής

ΛΥΣΕΙΣ

α->2, β->4, γ->1, δ->3, ε->5, στ->6