

## Επεξεργασία Δοσοληψιών Ανακεφαλαίωση

### Πρόβλημα

«Σωστή» εκτέλεση προγραμμάτων όταν επιτρέψουμε *ταυτοχρονισμό* και ακόμα και αν υπάρχουν *αποτυχίες*

### Δοσοληψία (transaction)

εκτέλεση ενός προγράμματος που προσπελαίνει ή τροποποιεί το περιεχόμενο της βάσης δεδομένων

το πώς βλέπει το ΣΔΒΔ τα προγράμματα των χρηστών

- Η **συνδρομικότητα** (concurrency) επιτυγχάνεται από το ΣΔΒΔ που διαπλέκει τις πράξεις (αναγνώσεις/εγγραφές) των διαφόρων συναλλαγών

### Δοσοληψία (transaction)

Ένα πρόγραμμα χρήστη μπορεί να εκτελεί πολλές λειτουργίες στα δεδομένα που ανακτεί από τη ΒΔ, αλλά το ΣΔΒΔ ενδιαφέρεται μόνο για τα δεδομένα που *διαβάζονται/γράφονται* στη ΒΔ

- Ανάγνωση(X) - R(X)
- Εγγραφή(X) - W(X)

### Πράξεις Δοσοληψιών

- **BEGIN**
- **R(X) W(X)**
- **END**
- **COMMIT** (επικύρωση) - επιτυχία - όλες οι τροποποιήσεις επικυρώνονται και δεν μπορούν να αναιρεθούν
- **ABORT** (ακύρωση ή ανάκληση) - αποτυχία - όλες οι τροποποιήσεις πρέπει να αναιρεθούν
- Μια δοσοληψία μπορεί να ανακυρωθεί (commit) αφού ολοκληρώσει όλες τις πράξεις της ενώ μπορεί να ακυρωθεί (abort) αφού εκτελέσει κάποιες από τις πράξεις της

### Ιδιότητες Δοσοληψιών

- **Atomicity** (ατομικότητα) - είτε όλες οι πράξεις είτε καμία
- **Consistency** (συνέπεια) - διατήρηση συνέπειας της ΒΔ
- **Isolation** (απομόνωση) - δεν αποκαλύπτει ενδιάμεσα αποτελέσματα
- **Durability** (μονιμότητα ή διάρκεια) - μετά την επικύρωση μιας δοσοληψίας οι αλλαγές δεν είναι δυνατόν να χαθούν

Επανάληψη: Επιθυμητές Ιδιότητες μιας Δοσοληψίας

- Atomicity (ατομικότητα) → ΤΕΧΝΙΚΕΣ ΑΝΑΚΑΜΨΕΙΣ
- Consistency (συνέπεια) → ΥΠΕΥΘΥΝΟΤΗΤΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ
- Isolation (απομόνωση) → ΕΛΕΓΧΟΣ ΣΥΝΔΡΟΜΙΚΟΤΗΤΑΣ
- Durability (μονιμότητα ή διάρκεια) → ΤΕΧΝΙΚΕΣ ΑΝΑΚΑΜΨΕΙΣ

Επανάληψη: Ορισμός Δοσοληψίας

Μια **δοσοληψία** είναι μια ακολουθία από πράξεις εγγραφής και ανάγνωσης που τελειώνει με μια πράξη επικύρωσης (commit) ή με μια πράξη ακύρωσης (abort)

☞ Παράδειγμα:: Θεωρείστε τις δύο συναλλαγές (Xacts):

```
T1: BEGIN R(X), X=X-N, W(X), R(Y), Y=Y+N, W(Y), END
T2: BEGIN R(X) X=X+M, W(X), END
```

- ❖ T1: R<sub>1</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>1</sub>(Y) C<sub>1</sub>
- ❖ T2: R<sub>2</sub>(X) W<sub>2</sub>(X) C<sub>2</sub>

Επανάληψη: Ορισμός Χρονοπρογράμματος

- Εκφράζει μια συγκεκριμένη εκτέλεση ενός συνόλου δοσοληψιών
- Οι πράξεις των δοσοληψιών εμφανίζονται στο χρονοπρόγραμμα με τη σειρά που εκτελούνται

Συγκεκριμένα

Ένα **χρονοπρόγραμμα (schedule) S** των δοσοληψιών T<sub>1</sub>, T<sub>2</sub>, ..., T<sub>n</sub> είναι μια διάταξη των πράξεων τους με τον περιορισμό ότι για κάθε δοσοληψία T<sub>i</sub> που συμμετέχει στο S οι πράξεις της T<sub>i</sub> στο S πρέπει να εμφανίζονται με την ίδια σειρά που εμφανίζονται στην T<sub>i</sub>

Επανάληψη: Ορισμός Χρονοπρογράμματος

T1 T2

R<sub>1</sub>(X)

R<sub>2</sub>(X)

W<sub>1</sub>(X)  
R<sub>1</sub>(Y)

W<sub>2</sub>(X)  
C<sub>2</sub>

W<sub>1</sub>(Y)  
C<sub>1</sub>

S: R<sub>1</sub>(X) R<sub>2</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>2</sub>(X) C<sub>2</sub> W<sub>1</sub>(Y) C<sub>1</sub>

Τόσα διαφορετικά χρονοπρογράμματα όσες και πιθανές εκτελέσεις

Επανάληψη: Σειριοποιησιμότητα

- **Σειριακά Χρονοπρογράμματα:** χρονοπρογράμματα που δεν διαπλέκουν πράξεις διαφορετικών δοσοληψιών (οι πράξεις κάθε δοσοληψίας εκτελούνται διαδοχικά, χωρίς παρεμβολή πράξεων από άλλη δοσοληψία)

Παρατήρηση: Αν κάθε δοσοληψία διατηρεί τη συνέπεια, τότε κάθε σειριακό χρονοπρόγραμμα διατηρεί τη συνέπεια

Ένα σειριακό χρονοπρόγραμμα είναι σωστό

S: R<sub>1</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>1</sub>(Y) C<sub>1</sub> R<sub>2</sub>(X) W<sub>2</sub>(X) C<sub>2</sub>

Επανάληψη: Σειριοποιησιμότητα

- **Ισοδύναμα Χρονοπρογράμματα :**

Για κάθε κατάσταση της ΒΔ, το αποτέλεσμα της εκτέλεσης του πρώτου χρονοπρογράμματος είναι το ίδιο με το αποτέλεσμα του δεύτερου χρονοπρογράμματος

Ένα χρονοπρόγραμμα ισοδύναμο με ένα σειριακό είναι σωστό

Σύγκρουση πράξεων σε χρονοπρόγραμμα

Δύο πράξεις σε ένα χρονοπρόγραμμα **συγκρούονται** αν (α) ανήκουν σε διαφορετικές δοσοληψίες, (β) προσπελαίνουν το ίδιο στοιχείο, και (γ) μια από αυτές είναι πράξη εγγραφής (W) Ποια είναι η σχέση των χρονοπρογραμμάτων S1 και S2;

S1: R<sub>1</sub>(X) R<sub>2</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>2</sub>(X) C<sub>2</sub> W<sub>1</sub>(Y) C<sub>1</sub>

S2: R<sub>1</sub>(X) R<sub>2</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>2</sub>(X) C<sub>2</sub> W<sub>1</sub>(Y) C<sub>1</sub>

• Ισοδύναμα Χρονοπρογράμματα βάσει Συγκρούσεων:

Δυο χρονοπρογράμματα είναι **ισοδύναμα** βάσει συγκρούσεων αν η διάταξη κάθε ζεύγους συγκρουόμενων πράξεων είναι ίδια και στα δυο χρονοπρογράμματα.

• Σειριοποιήσιμο Χρονοπρόγραμμα :

Ένα χρονοπρόγραμμα που είναι **ισοδύναμο** με κάποιο σειριακό

• Σειριοποιησιμότητα βάσει Συγκρούσεων:

Ένα χρονοπρόγραμμα S είναι **σειριοποιήσιμο** βάσει συγκρούσεων αν είναι **ισοδύναμο** βάσει συγκρούσεων με κάποιο **σειριακό** χρονοπρόγραμμα S'.

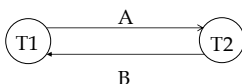
• Σε αυτήν την περίπτωση μπορούμε να αναδιατάξουμε τις μη συγκρουόμενες πράξεις στο S μέχρι να σχηματίσουμε ένα **ισοδύναμο** **σειριακό** χρονοπρόγραμμα.

Γράφος προήγησης (precedence graph) ή γράφος σειριοποιησιμότητας (serialization graph)

Κόμβος :: Δοσοληψία

Ακμή T<sub>i</sub> → T<sub>j</sub> αν μια πράξη της T<sub>i</sub> προηγείται μιας συγκρουόμενης πράξης της T<sub>j</sub>

T1: R<sub>1</sub>(A) W<sub>1</sub>(A), R<sub>1</sub>(B) W<sub>1</sub>(B)  
T2: R<sub>2</sub>(A) W<sub>2</sub>(A) R<sub>2</sub>(B) W<sub>2</sub>(B)



Η ετικέτα στην ακμή δείχνει σε πιο δεδομένο συγκρούονται (απλώς διακοσμητική!)

Θεώρημα

Ένα χρονοπρόγραμμα είναι **σειριοποιήσιμο** (βάσει συγκρούσεων) αν και μόνο αν ο γράφος προήγησης του είναι **ακυκλικός**.

## Επανάληψη: Χρονοπρογράμματα και Δυνατότητα Ανάκαμψης

### • Χρονοπρογράμματα με δυνατότητα ανάκαμψης (recoverable)

αν καμιά δοσοληψία  $T$  στο  $S$  δεν επικυρώνεται έως ότου επικυρωθούν όλες οι δοσοληψίες οι οποίες τροποποίησαν ένα δεδομένο που διαβάζει η  $T$

### • Χρονοπρογράμματα χωρίς διάδοση ανακλήσεων (avoids cascading aborts)

αν κάθε δοσοληψία  $T$  στο  $S$  διαβάζει μόνο στοιχεία που έχουν γραφεί από επικυρωμένες δοσοληψίες

### • Αυστηρά Χρονοπρογράμματα (strict)

οι δοσοληψίες δεν μπορούν ούτε να διαβάσουν ούτε να γράψουν ένα στοιχείο  $X$  έως ότου επικυρωθεί η δοσοληψία που έγραψε το  $X$

## Επεξεργασία Δοσοληψιών

### Παράδειγμα

Χαρακτηρίστε καθένα από τα παρακάτω χρονοπρογράμματα:

- σειριοποιήσιμα βάσει συγκρούσεων, σειριοποιήσιμα βάσει όψων
- με δυνατότητα ανάκαμψης, χωρίς διάδοση ανακλήσεων, αυστηρά

$R_1(X) W_2(X) W_1(X) A_2 C_1$

$R_1(X) W_2(X) C_2 W_1(X) C_1 R_3(X) C_3$

Σημείωση: σειριοποιήσιμα βάσει συγκρούσεων (χρήση του θεωρήματος)  
σειριοποιήσιμα βάσει όψων (έλεγχος με πιθανά σειριακά)

## Επεξεργασία Δοσοληψιών

### Παράδειγμα

Έστω το χρονοπρόγραμμα

$R_1(X) R_1(Y) W_1(X) R_2(Y) W_3(Y) W_1(X) R_2(Y)$

- (α) Υποθέτοντας ότι και οι τρεις δοσοληψίες τελικά επικυρώνονται, δώστε το γράφο σειριοποιησιμότητας
- (β) Τροποποιήστε το χρονοπρόγραμμα ώστε ένα πλήρες χρονοπρόγραμμα με τις παρακάτω ιδιότητες
- δυνατότητα ανάκαμψης όχι χωρίς διάδοση ανακλήσεων
  - χωρίς διάδοση ανακλήσεων
  - σειριοποιήσιμο βάσει συγκρούσεων

## Τέλος Επανάληψης

## Τεχνικές Ελέγχου Συνδρομικότητας

- Ο χρήστης δεν ασχολείται με τη συνδρομικότητα
- Το  $\Sigma\Delta\Theta\Delta$  εξασφαλίζει «σωστή συνδρομικότητα», γενικά *δρομολογεί* τις πράξεις των δοσοληψιών ώστε να προκύπτουν χρονοπρογράμματα σειριοποιήσιμα βάσει συγκρούσεων
- Μέσω τεχνικών ελέγχου συνδρομικότητας

Δηλαδή, ψάχνουμε αλγόριθμους (πρωτόκολλα) που θα δρομολογούν τις πράξεις των δοσοληψιών

## Τεχνικές Ελέγχου Συνδρομικότητας

- Ο χρήστης δεν ασχολείται με τη συνδρομικότητα
- Το  $\Sigma\Delta\Theta\Delta$  εξασφαλίζει «σωστή συνδρομικότητα», γενικά *δρομολογεί* τις πράξεις των δοσοληψιών ώστε να προκύπτουν χρονοπρογράμματα σειριοποιήσιμα βάσει συγκρούσεων
- Μέσω τεχνικών ελέγχου συνδρομικότητας

Δηλαδή, ψάχνουμε αλγόριθμους (πρωτόκολλα) που θα δρομολογούν τις πράξεις των δοσοληψιών

## Τεχνικές

1. **Κλειδώματος (locking)** για να αποτρέψουν τη συνδρομική (ταυτόχρονη) προσπέλαση των δεδομένων από πολλές δοσοληψίες
2. **Διάταξης χρονοσημάτων (timestamps)**
3. **Πιστοποίησης (validation)** μιας δοσοληψίας (αισιόδοξα πρωτόκολλα)

Η πιο απλή εκδοχή: ένα κλειδί ανά δεδομένο

- μια δοσοληψία πριν προσπελάσει ένα δεδομένο X ζητά ένα κλειδί -- αίτηση **lock(X)**
- μπορεί να προσπελάσει το δεδομένο, μόνο αφού της δοθεί το κλειδί -- *πότε παίρνει το κλειδί;*
- μια δοσοληψία μπορεί να άρει το κλειδί στο δεδομένο -- αίτηση **unlock (X)**

## ένα απλό κλειδώμα:

στην πιο απλή περίπτωση, ένα **μόνο είδος κλειδιού**

- **lock(X)** :: πραγματοποιείται αν το δεδομένο δεν είναι ήδη κλειδωμένο, αλλιώς η δοσοληψία περιμένει μέχρι να ελευθερωθεί το δεδομένο
- μια δομή (πίνακας) (δεδομένο, μια ένδειξη (κλειδωμένο - μη-κλειδωμένο), ουρά με δοσοληψίες που περιμένουν)

• **unlock (X)**

Δυο προβλήματα:

1. Δεν επιτρέπει ταυτόχρονες αναγνώσεις.
  2. Δε δουλεύει! (δηλαδή, δεν αρκεί για να δώσει σειριοποιησιμα χρονοπρογράμματα)
- Χρειάζεται (όπως θα δούμε) διάταξη των πράξεων lock-unlock κάθε δοσοληψίας

Ας διορθώσουμε το πρώτο

Στόχος: πολλές δοσοληψίες να μπορούν να διαβάσουν ένα δεδομένο ταυτόχρονα

Δύο ειδών κλειδιά:

- **διαμοιραζόμενο (shared)** κλειδί ή κλειδί ανάγνωσης
- **αποκλειστικό (exclusive)** κλειδί ή κλειδί εγγραφής

• μια δοσοληψία πριν **διαβάσει** ένα δεδομένο X ζητά ένα διαμοιραζόμενο κλειδί -- αίτηση **S-lock(X)**

• μια δοσοληψία πριν **γράψει** ένα δεδομένο X ζητά ένα αποκλειστικό κλειδί -- αίτηση **X-lock(X)**

• η αίτηση για κλειδί δίνεται αν δεν υπάρχει «**συγκρούμενο κλειδί**»

• (πάλι) μια δοσοληψία μπορεί να άρει το κλειδί στο δεδομένο -- αίτηση **unlock (X)**

Πίνακας συμβατότητας κλειδιών

	S-Lock(X)	X-Lock(X)
S-Lock(X)	✓	
X-Lock(X)		

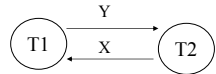
T1  
S-Lock(Y)  
R<sub>1</sub>(Y)  
Unlock(Y)

T2

S-Lock(X)  
R<sub>2</sub>(X)  
Unlock(X)  
X-Lock(Y)  
W<sub>2</sub>(Y)  
Unlock(Y)  
C<sub>2</sub>

Δεν αρκεί για  
σειριοποιησιμότητα

S: R<sub>1</sub>(Y) R<sub>2</sub>(X) W<sub>2</sub>(Y) C<sub>2</sub> W<sub>1</sub>(X) C<sub>1</sub>



S-Lock(X)  
W<sub>1</sub>(X)  
Unlock(X)  
C<sub>1</sub>

Λύση: Κλειδωμα Δύο Φάσεων

- Πρωτόκολλο κλειδώματος δυο φάσεων (Two-Phase Locking 2PL)

Όλες οι πράξεις (αιτήσεις) κλειδώματος μιας δοσοληψίας προηγούνται της πρώτης πράξης (αίτησης) άρσης κλειδώματος της διαδικασίας

Δηλαδή, μόλις μια δοσοληψία αφήσει (unlock) ένα κλειδί δεν μπορεί να ζητήσει ξανά κλειδί

Κάθε δοσοληψία δυο φάσεις

- μια φάση επέκτασης ή εξάπλωσης
- μια φάση συρρίκνωσης

Αποδεικνύεται ότι είναι σωστό

Τροποποίηση ώστε να είναι αυστηρό;

Παρατηρήσεις

- Οι αιτήσεις lock και unlock πρέπει να είναι ατομικές πράξεις
- **Αναβάθμιση κλειδιού:** μια δοσοληψία που κατέχει ένα διαμοιραζόμενο κλειδί μπορεί να αναβαθμιστεί ώστε να κατέχει ένα αποκλειστικό κλειδί

Οι τεχνικές κλειδώματος μπορεί να προκαλέσουν αδιέξοδα (deadlocks)

T1  
S-Lock(Y)  
R<sub>1</sub>(Y)

T2

S-Lock(X)  
R<sub>2</sub>(X)

X-Lock(X)

X-Lock(Y)

Η T1 περιμένει την T2 να ελευθερώσει το X, και η T2 περιμένει την T1 να ελευθερώσει το Y

Δυο τεχνικές:

- Πρωτόκολλα Πρόληψης Αδιεξόδων (Deadlock Prevention): Αποφυγή δημιουργίας αδιεξόδου
- Πρωτόκολλα Ανίχνευσης Αδιεξόδου (Deadlock Detection): Ελέγχουμε περιοδικά αν το σύστημα βρίσκεται σε κατάσταση αδιεξόδου

- Κάθε δοσοληψία πρέπει να κλειδώσει *όλα* τα δεδομένα που χρειάζεται πριν ξεκινήσει
- Αν δε μπορεί να κλειδώσει έστω και ένα, δε κλειδώνει κανένα και προσπαθεί ξανά

«σπάσιμο» του κύκλου  $\Rightarrow$  κάποια διάταξη μεταξύ των δοσοληψιών

Κάθε δοσοληψία  $T$  έχει ένα χρονόσημο  $TS(T)$ :

Μια διαδικασία παίρνει ένα χρονόσημο κατά την εκκίνησή της.

$TS(T_1) < TS(T_2)$ , σημαίνει ότι η  $T_1$  ξεκίνησε πριν την  $T_2$

ιδέα: μια δοσοληψία **περιμένει** μόνο αν το κλειδί το έχει μια δοσοληψία με **μικρότερο** (μεγαλύτερο) χρονόσημο, αλλιώς **ακυρώνεται**

Δύο σχήματα

- αναμονής-θανάτωσης
- τραυματισμού-αναμονής

Και στα δύο σχήματα οι παλιές δοσοληψίες «εκτοπιζούν» τις νεώτερες

Έστω ότι η  $T_i$  ζητά να κλειδώσει το  $X$  που είναι κλειδωμένο από την  $T_j$

- αναμονής-θανάτωσης

Αν  $TS(T_i) < TS(T_j)$

$T_i$  περιμένει (\* περιμένουμε αν το κλειδί το έχει νεώτερη \*)

Αλλιώς (\* το κλειδί το έχει παλαιότερη \*)

ακυρώνεται (πεθαίνει!) η  $T_i$  και επανεκκινείται με το ίδιο (γιατί;) χρονόσημο

- τραυματισμός -αναμονή

Αν  $TS(T_i) > TS(T_j)$

$T_i$  περιμένει (\* περιμένουμε αν το κλειδί το έχει παλαιότερη \*)

Αλλιώς (\* το κλειδί το έχει νεώτερη \*)

ακυρώνεται (τραυματίζεται) η  $T_j$  και επανεκκινείται με το ίδιο χρονόσημο

Κατασκευή **γράφου αναμονής** (wait-for graph)

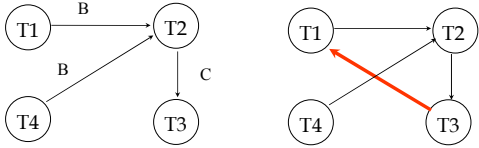
κόμβοι : δοσοληψίες

ακμή από τον κόμβο  $T_i$  στον  $T_j$ , αν η  $T_i$  περιμένει την  $T_j$  να αφήσει ένα κλειδί

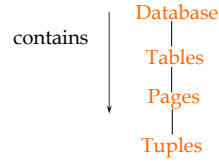
- Περιοδικά έλεγχος για κύκλους στο γράφο αναμονής

Παράδειγμα:

T1: S-Lock(A), R(A), S-Lock(B)  
 T2: X-Lock(B), W(B)  
 T3: S-Lock(C), R(C)  
 T4: X-Lock(C), X-Lock(A)



⌘ Δύσκολο να αποφασίσουμε τη διακρίτοτητα (granularity) για το κλειδωμα



Τεχνικές

- ✓ 1. Κλειδώματος (locking)
- 2. Διάταξης χρονοσημάτων (timestamps)
- 3. Πιστοποίησης (validation)

Το χρονόσημα δημιουργείται από το ΣΔΒΔ και προσδιορίζει μοναδικά μια δοσοληψία

Ιδέα: διάταξη των δοσοληψιών με βάση το χρονόσημα τους (δηλαδή, χρονοπρόγραμμα ισοδύναμο με σειριακό στο οποίο οι δοσοληψίες εμφανίζονται διατεταγμένες με βάση τις τιμές των χρονοσημάτων)

⇒ άρα η σειρά προσπέλασης στα δεδομένα πρέπει να μη παραβιάζει τη σειριοποιησιμότητα

Δηλαδή: αν μια πράξη  $a_i$  μιας δοσοληψίας  $T_i$  συγκρούεται με μια πράξη  $a_j$  μιας δοσοληψίας  $T_j$  και  $TS(T_i) < TS(T_j)$ , τότε η  $a_i$  πρέπει να προηγείται της  $a_j$ . Αλλιώς, restart τη δοσοληψία.

Κάθε δεδομένο  $X$  έχει δύο τιμές χρονοσημάτων:

$ΧΣΑ(X)$  (χρονόσημα ανάγνωσης) το μεγαλύτερο μεταξύ όλων των χρονοσημάτων των δοσοληψιών που διάβασαν το  $X$  (διαισθητικά, η πιο πρόσφατη που το διάβασε)

$ΧΣΕ(X)$  (χρονόσημα εγγραφής) το μεγαλύτερο μεταξύ όλων των χρονοσημάτων των δοσοληψιών που έγραψαν το  $X$



### Διάταξη Χρονοσημάτων

Η δοσοληψία T με ΧΣ(T) εκτελεί μια πράξη **ανάγνωσης R(X)**

Αν  $ΧΣ(T) < ΧΣΕ(X)$  (αυτό παραβιάζει τη διάταξη)

η T ακυρώνεται,

μπορεί να ξαναρχίσει αλλά με μεγαλύτερο χρονόσημα (γιατί:)

Αν  $ΧΣ(T) > ΧΣΕ(X)$

η ανάγνωση είναι επιτρεπτή

Θέσε το  $ΧΣΑ(X) = \max\{ΧΣΑ(T), ΧΣ(ΤΑ)\}$

• Οι αλλαγές στο  $ΧΣΑ(X)$  πρέπει να γράφονται στο δίσκο! Αυτό και το ότι η δοσοληψίες ξαναρχίζουν προκαλεί overhead

### Διάταξη Χρονοσημάτων

Η δοσοληψία T με ΧΣ(T) εκτελεί μια **πράξη εγγραφής W(X)**

Αν  $ΧΣΑ(X) > ΧΣ(T)$  ή  $ΧΣΕ(X) > ΧΣ(T)$

η T ακυρώνεται (γιατί:)

Βελτιστοποίηση: τι σημαίνει  $ΧΣΕ(X) > ΧΣ(T)$

### Διάταξη Χρονοσημάτων

Ο **κανόνας του Thomas για εγγραφές** (Thomas Write Rule) Μπορούμε να αγνοήσουμε μερικές «ξεπερασμένες» ή **τυφλές** εγγραφές, δε χρειάζεται επανεκκίνηση της T (η εγγραφή της T ακολουθείται από άλλη εγγραφή, χωρίς ενδιάμεση ανάγνωση)

T1	T2
R(A)	W(A)
	Commit
W(A)	
Commit	

Επιτρέπει σειριοποίηση - αλλά όχι σειριοποίησιμα βάσει συγκρούσεων

### Διάταξη Χρονοσημάτων

• Δυστυχώς, παράγει και χρονοπρογράμματα χωρίς δυνατότητα ανάκαμψης

T1	T2
W(A)	R(A)
	W(B)
	Commit

⌘ Τροποποίηση

☒ Buffer all writes μέχρι την επικύρωση του writer (αλλά το  $ΧΣΕ(X)$  τροποποιείται κανονικά)

☒ Block readers T (όπου  $ΧΣ(T) > ΧΣΕ(X)$ ) μέχρι να επικυρωθεί ο writer του X

⌘ Όμοια με το να κρατούν οι writers X-Locks μέχρι την επικύρωση τους αλλά όχι ακριβώς 2PL

### Τεχνικές Ελέγχου Συνδρομικότητας

Τεχνικές

✓ 1. Κλειδώματος (locking)

✓ 2. Διάταξης χρονοσημάτων (timestamps)

→ 3. Πιστοποίησης (validation)

### Αισιόδοξες Τεχνικές

Οι τεχνικές κλειδώματος είναι συντηρητικές (αποφεύγονται οι συγκρούσεις)

**Μειονεκτήματα**

• επιβάρυνση (overhead) χειρισμού κλειδώματος

• αποφυγή/ανίχνευση αδιεξόδων

• lock contention για τα δεδομένα που χρησιμοποιούνται συχνά

Αν οι συγκρούσεις είναι σπάνιες, μεγαλύτερη συγχρονικότητα, αν αντί για κλειδωμα, έλεγχος για συγκρούσεις όταν μια δοσοληψία επικυρώνεται (commits)

Κάθε δοσοληψία έχει τρεις φάσεις

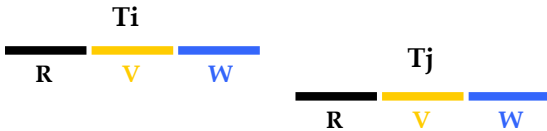
- **ΑΝΑΓΝΩΣΗ**: η δοσοληψία διαβάζει από τη βδ, αλλά τροποποιεί προσωπικά αντίγραφα των δεδομένων
- **ΠΙΣΤΟΠΟΙΗΣΗ**: έλεγχος για συγκρούσεις
- **ΕΓΓΡΑΦΗ**: γράφει τα τοπικά αντίγραφα στη βδ

Έλεγχος συνθηκών που είναι ικανές για να εξασφαλίσουν ότι δεν υπήρχαν συγκρούσεις

- Κάθε δοσοληψία T έχει ένα μοναδικό αριθμό TID (χρονόσημα)
  - Το TID ανατίθεται στο τέλος της φάσης ΑΝΑΓΝΩΣΗΣ (ακριβώς πριν αρχίσει η πιστοποίηση)
  - Με κάθε δοσοληψία
- ReadSet(T)**: το σύνολο των δεδομένων που διάβασε η T
- WriteSet(T)**: το σύνολο των δεδομένων που διάβασε η T

ΕΛΕΓΧΟΣ Περίπτωση 1

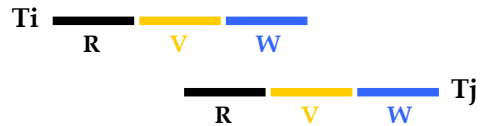
⌘ Για όλα τα i και j τέτοια ώστε  $T_i < T_j$ , η  $T_i$  τελειώνει πριν αρχίσει η  $T_j$ .



ΕΛΕΓΧΟΣ Περίπτωση 2

⌘ Για όλα τα i και j τέτοια ώστε  $T_i < T_j$ :

- η  $T_i$  τελειώνει πριν αρχίσει η φάση εγγραφής της  $T_j$
- $WriteSet(T_i) \cap ReadSet(T_j) = \emptyset$



ΕΛΕΓΧΟΣ Περίπτωση 3

⌘ Για όλα τα i και j τέτοια ώστε  $T_i < T_j$ :

- η  $T_i$  τελειώνει τη φάση ανάγνωσης πριν αρχίσει η φάση ανάγνωσης της  $T_j$
- $WriteSet(T_i) \cap ReadSet(T_j) = \emptyset$
- $WriteSet(T_i) \cap WriteSet(T_j) = \emptyset$



⌘ Πιστοποίηση της T (έλεγχος 1 & 2):

```

valid = true;
// S = set of Xacts that committed after Begin(T)
< foreach Ts in S do {
  if ReadSet(Ts) does intersect WriteSet(T)
    then valid = false;
}
if valid then { install updates; // Write phase
  Commit T } >
else Restart T
    
```

τέλος κρίσιμης περιοχής

- Υπάρχουν πολλά πρωτόκολλα ελέγχου συνδρομικότητας (concurrency control protocols)
- Τρεις βασικές κατηγορίες: κλείδωμα, χρονοσήματα, αισιόδοξα

Επόμενο Μάθημα:

Τι γίνεται με την SQL (μπορεί ο προγραμματιστής να καθορίσει τι θέλει);

Ανάκτηση από σφάλματα (ποιες ιδιότητες των δοσοληψιών;)