Software Transactional Memory

Η STM υποστηρίζει την εκτέλεση δοσοληψιών από τις διεργασίες, οι οποίες περιέχουν λειτουργίες που ο χρήστης θέλει να εκτελέσει στα διαμοιραζόμενα αντικείμενα.

Η STM εγγυάται την ατομική εκτέλεση των λειτουργιών μιας δοσοληψίας -> είτε όλες οι λειτουργίες της δοσοληψίας θα εκτελεστούν ατομικά ή καμία.

Σκοπός της STM

- Απλούστευση της συγγραφής παράλληλου κώδικα
- Υποστηρίζεται η σύνθεση μιας ή περισσότερων λειτουργιών ατομικών αντικειμένων σε μια που εκτελείται με ατομικό τρόπο.
    - π.χ., InsertDual() σε λίστα (εισαγωγή δυο στοιχείων στη λίστα)

STM: Πως λειτουργεί;

Αντιστοιχίζονται κάποιου είδους επιπλέον πληροφορίες (metadata):

1. σε κάθε δοσοληψία
   - π.χ., κατάσταση δοσοληψίας (ενεργή, ολοκληρωμένη επιτυχώς, ολοκληρωμένη μη-επιτυχώς), κλπ.
2. σε κάθε διαμοιραζόμενο αντικείμενο που προσπελάζεται μέσω δοσοληψιών
   - π.χ., περιγραφή δοσοληψίας που κατέχει δικαίωμα προσπέλασης στο αντικείμενο

Ιδιοκτησία Αντικειμένου

Αποκτάται από μια δοσοληψία πριν την ενημέρωση του αντικειμένου

- Αντιστρέψιμη (π.χ., βίαιος τερματισμός δοσοληψίας, μηχανισμός βοήθειας, κ.α.)
- Μη-αντιστρέψιμη (κλείδωμα, κ.α.)

Ορατές – Μη ορατές Αναγνώσεις

Η ανάγνωση ενός αντικειμένου από μια δοσοληψία πρέπει να εξασφαλίζει ότι τα δεδομένα δεν ενημερώνονται ταυτόχρονα από άλλες δοσοληψίες.

- Απόκτηση ιδιοκτησίας ανάγνωσης (ορατή ανάγνωση)
- Μη ορατή ανάγνωση (αναγκαιότητα μηχανισμού ελέγχου συνέπειας)

Ορισμός

t-μεταβλητές: κελιά μνήμης που προσπελάζονται μόνο μέσω δοσοληψιών

Αν οι t-μεταβλητές είναι γνωστές εξ αρχής (κατά την εκκίνηση κάθε διεργασίας) η STM λέγεται **στατική**. Αν αυτό δεν ισχύει, η STM λέγεται **δυναμική**.

- Πίνακας από μεταβλητές;
- Λίστα; Ουρά;

Η STM επιτρέπει στις διεργασίες να επικοινωνούν διαβάζοντας και ενημερώνοντας t-μεταβλητές μέσω δοσοληψιών.

STM ως shared object

Η STM υποστηρίζει τις εξής λειτουργίες:

1. pointer BeginTransaction(): εκκίνηση νέας δοσοληψίας -> επιστρέφεται δείκτης στα metadata που περιγράφουν τη δοσοληψία. Ο δείκτης αυτός χρησιμοποιείται ως παράμετρος σε όλες τις υπόλοιπες λειτουργίες
2. pointer CreateNewTMVar(pointer t, pointer d): δημιουργία νέας t-μετβλητής -> t: δείκτης στα metadata της δοσοληψίας, d: δείκτης στη θέση μνήμης με την οποία συσχετίζεται η νέα t-μεταβλητή.
3. (Boolean, d) ReadTMVar(pointer t, pointer tVar): ανάγνωση της t-μεταβλητής στην οποία δείχνει ο δείκτης tVar. Επιστρέφεται TRUE ή FALSE, ανάλογα με το αν η ανάγνωση ολοκληρώνεται επιτυχώς ή όχι και επιστρέφεται και τα δεδομένα της tVar σε περίπτωση επιτυχούς ανάγνωσης.
4. Boolean WriteTMVar(pointer t, pointer tVar, pointer d): εγγραφή στην tvar των δεδομένων που δείχνει ο d
5. Boolean CommitTransaction(pointer t): προσπάθεια επιτυχούς ολοκλήρωσης της δοσοληψίας t
6. AbortTransaction(pointer t): τερματισμός της t ως μη-επιτυχημένης.

STM: Χρήσιμοι Ορισμοί

Μια δοσοληψία είναι μια ακολουθία από λειτουργίες που εκτελούνται ατομικά από μια συγκεκριμένη διεργασία.

Μια δοσοληψία είναι της μορφής:

(BeginTransaction (CreateTMVar | ReadTMVar | WriteTMVar)* (CommitTransaction | AbortTransaction))

- δοσοληψία ανάγνωσης
- δοσοληψία ενημέρωσης

Αν μια δοσοληψία Τ καλέσει την CommitTransaction και αυτή επιστρέψει TRUE, τότε η Τ έχει ολοκληρωθεί επιτυχώς.

Αν μια δοσοληψία Τ καλέσει την AbortTransaction λέμε ότι η Τ ολοκληρώνεται μη-επιτυχώς.

Λέμε ότι η Τ τερματίζεται βίαια ως μη-επιτυχημένη σε μια εκτέλεση, αν η Τ ολοκληρωθεί ως μη-επιτυχημένη χωρίς να καλέσει την AbortTransaction.

STM: Ορθότητα και Απόδοση

Ορθότητα

Σειριοποιησιμότητα για δοσοληψίες;

Μετρικά Πολυπλοκότητας

- Χρονική πολυπλοκότητα μιας λειτουργίας
- Χωρική πολυπλοκότητα

- **Ικανότητα Διεκπεραίωσης (throughput):** πλήθος δοσοληψιών που ολοκληρώνονται επιτυχώς στη μονάδα του χρόνου

- **Μέσος χρόνος διεκπεραίωσης:** μέσος χρόνος διεκπεραίωσης μιας δοσοληψίας ως επιτυχημένης (δεδομένου ότι κάθε φορά που αυτή αποτυγχάνει επαν-εκτελείται μέχρι να επιτευχθεί επιτυχία).

STM: Χρήσιμοι Ορισμοί

Σύγκρουση μεταξύ δυο δοσοληψιών t_1 και t_2: παρουσιάζεται αν οι t_1, t_2 προσπελάζουν την ίδια t-μεταβλητή tVAr και τουλάχιστον μια από τις δυο αποσκοπεί στην ενημέρωση της tVar.

- Εγγραφής-ανάγνωσης (W-R)
- Ανάγνωσης-εγγραφής (R-W)
- Εγγραφής-εγγραφής (W-W)

Απαραίτητες Συμβάσεις που πρέπει να ακολουθούνται από τον χρήστη

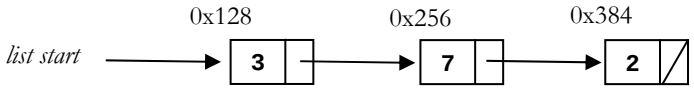
- Μια δοσοληψία εκκινείται με την εκτέλεση της BeginTransaction().
- δείκτης που επιστρέφεται από την BeginTransaction() χρησιμοποιείται ως παράμετρος σε όλες τις λειτουργίες που καλούνται μέσω αυτής της δοσοληψίας.
- Ο χρήστης μπορεί να έχει το πολύ μια εκκρεμή δοσοληψία κάθε χρονική στιγμή.
- Μια δοσοληψία μπορεί να προσπελάζει μόνο t-μεταβλητές καλώντας τις ReadTMVar() και WriteTMVar().
- t-μεταβλητές μπορούν να δημιουργηθούν μόνο καλώντας την CreateNewTMVar().

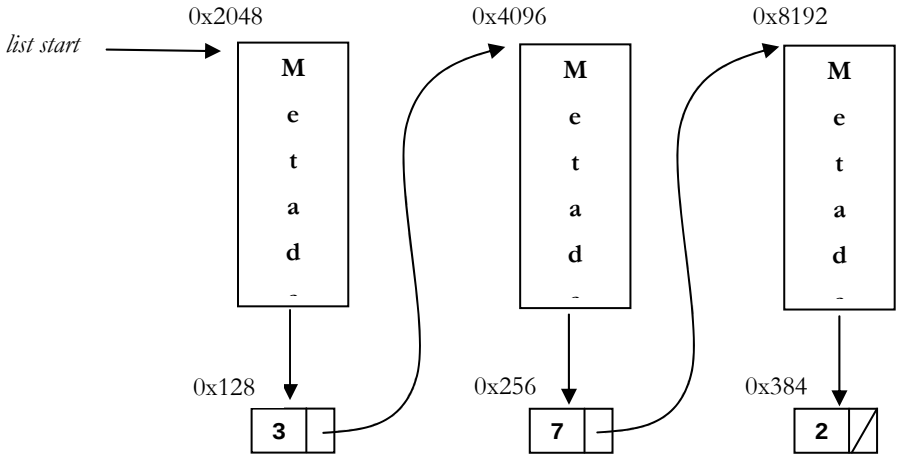
Παράδειγμα Χρήσης της STM: Διαμοιραζόμενος Μετρητής

```
1.  void AtomicCounterIncrement (TmVar counter) {
2.    int tmp;
3.    TmVar newTmVar;
4.    boolean bool;
5.    void *t;
6.    while (1) {
7.      t = BeginTransaction ();                    // δημιουργία μιας νέας δοσοληψίας
8.    (bool, tmp) = ReadTmVar (t, counter);         // ανάγνωση της t-μεταβλητής counter
                                                    // tmp = τιμή της t-μετ/τής counter
9.      tmp ++;
10.     bool = WriteTmVar (t, counter, tmp);        // εγγραφή στην t-μετ/τής counter
                                                    // counter = tmp
11.     if (bool == false) {                  // αν η αυξημένη τιμή είναι απαρχαιωμένη
                                              // η WriteTmVar αποτυγχάνει
                                              // επιστρέφοντας FALSE
12.       AbortTransaction(t);
13.       continue;
14.    }
15.   if (CommitTransaction(t) == TRUE)
16.                   break;                  // η δοσοληψία τερματίζει επιτυχώς
17.    }
18. }
```

Παράδειγμα Χρήσης της STM: Shared Ordered List

struct node{
  int num;
  struct node *next;
}
	typedef void * TmVar;
	struct node {
	  int num;
	  TmVar next;
	}





Παράδειγμα Χρήσης της STM: Shared Ordered List

```
boolean InsertS(struct node **L, int x) {
    struct node *P, *prevP = NULL,
                    *newnode;
    P = *L;
    while (P != NULL &&
                              P->num < x) {
        prevP = P;
        P = P->next
    }
    if (P != NULL &&
                          P->num == x)
        return(FALSE);
    else {
        newnode = malloc(
                        sizeof( struct node));
        newnode->next = P;
        if (prevP == NULL) *L = P;
        else prevP -> next = newNode;
        return (TRUE);
    }
}
```

Παράδειγμα Χρήσης της STM: Shared Ordered List

```
void Insert (int num, TmVar listStart) {
    Object *tmpNode=null, *newNode;
    TmVar tmpNodeTmVar, newTmVar;
    boolean bool;
    void *t;

    while (1) {
      t = BeginTransaction ();
      tmpNodeTmVar = listStart;
      do {
        if (tmpNode != null)
          tmpNodeTmVar = tmpNode->next;
        (bool, tmpNode) = ReadTmVar (t, tmpNodeTmVar);
        if (bool == false) break;
      } while (tmpNode->next == null);

      if (bool == false)
        AbortTransaction(t);
      else {
        newNode = (Node *)malloc (sizeof(Node));
        newNode->num = num;
        newNode->next = null;
        newTmVar = CreateNewTmVar (t, newNode);
        tmpNode->next = newTmVar;
        bool = WriteTmVar (t, tmpNodeTmVar, tmpNode);

        if (bool == false)
            continue;
        if (CommitTransaction(t) == TRUE)
            break;
       }  // else
     }    // while
}  // Insert
```

Κατηγορίες Σχεδιαστικών Επιλογών

Οι αλγόριθμοι STM διαφέρουν στα εξής:

1. Τρόπος Ανάθεσης, Απόκτησης και Κατάργησης Ιδιοκτησιών
2. Χρόνος Απόκτησης Ιδιοκτησιών
3. Τρόποι Αποτροπής ή Ανίχνευσης και Επίλυσης Συγκρούσεων και Μηχανισμός Ελέγχου Συνέπειας
4. Πλήθος Ενδιάμεσων Επιπέδων
5. Μοντέλο Μνήμης

Τρόπος Ανάθεσης, Απόκτησης και Κατάργησης Ιδιοκτησιών

Απλούστερος Αλγόριθμος: χρησιμοποιεί μια μόνο ιδιοκτησία για όλες τις t-μεταβλητές

Καλύτερη Απόδοση $\Rightarrow$ μια ιδιοκτησία ανά t-μεταβλητή

- Ανάθεση ανά t-μεταβλητή – ανάθεση ανά shared object – ανά σύνολο ανάθεση

Κάθε δοσοληψία μπορεί να είναι είτε ενεργή ή ολοκληρωμένη. Στα metadata για τη δοσοληψία αποθηκεύεται μια μεταβλητή status.

Επώνυμη πληροφορία ιδιοκτησιών

Στα metadata μιας t-μεταβλητής αναφέρεται ποια δοσοληψία έχει υπό την κατοχή της την ιδιοκτησία για αυτή την μεταβλητή.

Ανώνυμη πληροφορία δοσοληψιών

Κάθε t-μεταβλητή μπορεί να είναι είτε υπό κατοχή ή ελεύθερη

Ορατές – Μη ορατές αναγνώσεις

Μόνιμη – Προσωρινή Ιδιοκτησία $\Rightarrow$ blocking – non blocking algorithms

Χρόνος Απόκτησης Ιδιοκτησιών

- **Εκ των προτέρων απόκτηση** $\Rightarrow$ άμεση ενημέρωση (undo logs)
- **Εκ των υστέρων απόκτηση** $\Rightarrow$ ετεροχρονισμένη ενημέρωση (redo logs)

Κάθε δοσοληψία διατηρεί δυο λίστες t-μεταβλητών:

- λίστα ενημερώσεων
- λίστα αναγνώσεων

Τρόποι Αποτροπής ή Ανίχνευσης και Επίλυσης Συγκρούσεων – Μηχανισμός Ελέγχου Συνέπειας

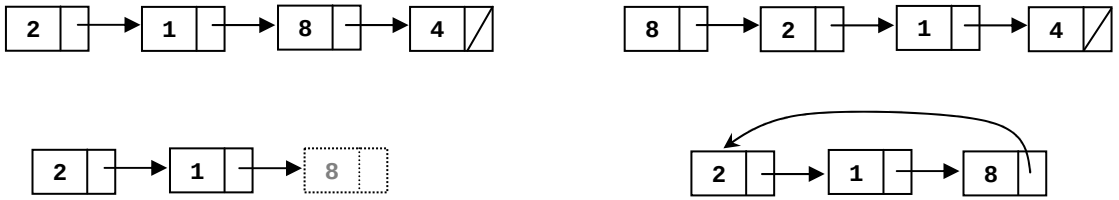
Σύγκρουση R-W

Όταν το σύστημα χρησιμοποιεί μη-ορατές αναγνώσεις μπορεί να προκύψει ασυνέπεια.

Μηχανσμός Ελέγχου Συνέπειας: έλεγχος συνέπειας των t-μεταβλητών που μια δοσοληψία έχει αναγνώσει.

Έκδοση t-μεταβλητής (version): Μια δοσοληψία που ενημερώνει την t-μεταβλητή ενημερώνει και το version της t-μεταβλητής.

Πρόβλημα που μπορεί να προκύψει αν δεν εφαρμοσθεί ο μηχανισμός ελέγχου συνέπειας:



Τρόποι Αποτροπής ή Ανίχνευσης και Επίλυσης Συγκρούσεων – Μηχανισμός Ελέγχου Συνέπειας

Αυξητικός Έλεγχος Συνέπειας

Άλλες τεχνικές (έλεγχος συννέπειας πραγματοποιείται κάθε ένα πεπερασμένο πλήθος λειτουργιών)

Αυτόματοι – Μη Αυτόματοι Μηχανισμοί Ελέγχου Συνέπειας

Ο μη αυτόματος έλεγχος της συνέπειας απαιτεί την εμπλοκή του χρήστη (και επαύξηση των λειτουργιών που παρέχει το διαμοιραζόμενο αντικείμενο STM)

Επιλογές όταν μια δοσοληψία T_1 συγκρούεται με μια άλλη T_2

- Η T_1 περιμένει την T_2 να ολοκληρωθεί
- Η T_1 μπορεί να ολοκληρωθεί ως μη-επιτυχημένη
- Η T_1 μπορεί να τερματίσει βίαια την T_2 και να αποκτήσει εκείνη την ιδιοκτησία της t-μεταβλητής που χρειάζεται
- Η T_1 μπορεί να βοηθήσει την T_2 να ολοκληρωθεί

Πλήθος Ενδιάμεσων Επιπέδων – Μοντέλο Μνήμης

Πλήθος Ενδιάμεσων Επιπέδων

Ορίζεται να είναι το μέγιστο πλήθος δεικτών που πρέπει να ακολουθηθούν από μια λειτουργία ανάγνωσης οποιασδήποτε t-μεταβλητής σε οποιαδήποτε εκτέλεση, βάσει της αναπαράστασης της μεταβλητής από τον εκάστοτε αλγόριθμο STM.

Το πλήθος αυτό εξαρτάται από το είδος των metadata που διατηρεί ο εκάστοτε αλγόριθμος.

Μοντέλο Μνήμης

Ανοιχτό μοντέλο μνήμης

Κλειστό μοντέλο μνήμης