Οικονομικό Πανεπιστήμιο Αθηνών

Υλοποίηση Two-Phase Locking με Undo/Redo

Βάσεις Δεδομένων Project Report

Δεδούσης Δημήτρης, Μπακοπούλου Εβίτα Ζωή

17/2/2015

1. Περιγραφή αρχείων

Ακολουθεί μια περιγραφή των αρχείων που περιέχουν την βάση δεδομένων, τα transactions τα οποία είναι προς εκτέλεση καθώς επίσης και του αρχείου που χρησιμοποιείται για την ανάνηψη της βάσης δεδομένων από κάποιου είδους αστοχίας κατά την διάρκεια της εκτέλεσης του προγράμματος μας.

1. Η βάση δεδομένων που χρησιμοποιούμε είναι αποθηκευμένη στο αρχείο db.txt το οποίο κατά την εκτέλεση του προγράμματος φορτώνεται στην μνήμη του υπολογιστή και εμπεριέχει αντικείμενα τύπου DbElement, σε μια δομή δεδομένων τύπου SET, στα οποία απεικονίζεται το όνομα της μεταβλητής καθώς και η τιμή της έτσι όπως φαίνεται στην βάση.

2. Το αρχείο transactions.txt περιέχει τα transactions που πρόκειται να εκτελεστούν. Όπως και με το db.txt έτσι και αυτό το αρχείο φορτώνεται στην μνήμη και κάθε transaction αντιστοιχίζεται με αντικείμενα τύπου Transaction τα οποία και αποθηκεύονται σε μια δομή δεδομένων τύπου λίστας. Το αρχείο αυτό έχει την ακόλουθη μορφή:

   TRANSACTION T1;
   r(X);r(A);
   uX=X+A;
   w(X);
   r(Y);
   uY=Y+1;
   w(Y)
   TRANSACTION T2;
   r(A);r(X);
   ....

   Όπου, r(X) είναι ανάγνωση της μεταβλητής X, w(X) είναι εγγραφή της μεταβλητής X και uX=X+1, είναι αλλαγή της τιμής της μεταβλητής X.

3. Το αρχείο undo_redo.log περιέχει τις αλλαγές που έχει υποστεί η βάση δεδομένων από τα transaction τα οποία έχουν εκτελεστεί. Κάθε ετικέτα περιέχει το transaction το οποίο έκανε την αλλαγή σε κάποιο στοιχείο της βάσης δεδομένων, το στοιχείο το οποίο υπέστει της αλλαγές, την αρχική τιμή του στοιχείου αυτού και τέλος την τελική τιμή του στοιχείου μετά την εκτέλεση του transaction. Το αρχείο αυτό έχει την ακόλουθη μορφή:

   <START T1>
   <T1 , X , 2 , 5>
   <T1 , Y , 4 , 5>
   <COMMIT T1>
   <START T2>
   ....

2. Σύντομη περιγραφή εκτέλεσης προγράμματος

Κατά την εκτέλεση του προγράμματος αρχικά φορτώνεται το αρχείο undo_redo.log, το οποίο αν περιέχει στοιχεία (δηλαδή έχει προηγηθεί σφάλμα στην προηγούμενη εκτέλεση), τότε εκτελείται η αποκατάσταση και αλλάζουν αναλόγως οι τιμές στην βάση δεδομένων (db.txt) καθώς και προστίθεται ετικέτα τύπου <ABORT> για κάθε transaction το οποίο δεν ήταν επιτυχημένο. Έπειτα, φορτώνεται η βάση δεδομένων(db.txt) και τα transactions(transactions.txt) στη μνήμη. Μόλις φορτωθούν τα transactions στην μνήμη και δημιουργηθούν αντικείμενα τύπου Transaction, για κάθε ένα transaction δημιουργείται και ένα νήμα το οποίο θα εκτελέσει το συγκριμένο transaction μέσω της κλάσης TransactionInitilizer οπου κάθε νήμα είναι και ένα transaction του αρχείου transactions.txt. Το κάθε νήμα εκτελεί τις ενέργειες του transaction που του αντιστοιχεί, αφού λάβει locks με βάση το two phase locking και γράφει στο undo_redo.log, την κάθε ενέργειά του και στη βάση δεδομένων τις αλλαγές που έχουν συμβεί επί κάποιου στοιχείου, αν πρόκειται για ενέργεια εγγραφής. Τέλος, μετά την εκτέλεση του προγράμματος, μπορούμε να δούμε τις αντίστοιχες αλλαγές στο αρχείο db.txt, και τις εγγραφές στο undo_redo.log.

3. Managing Locks

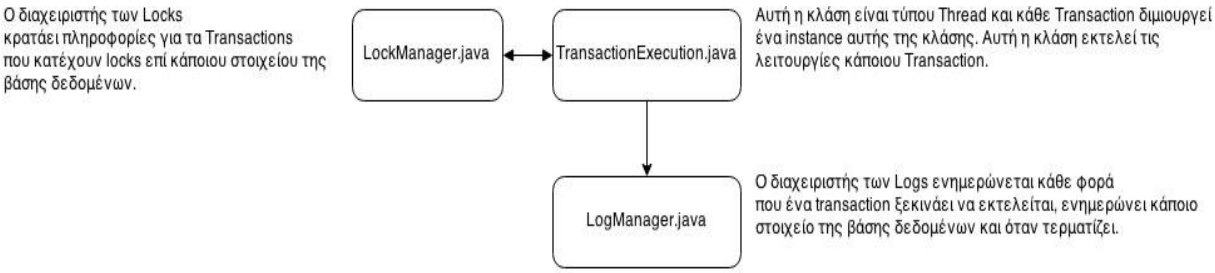
Το κάθε transaction για να κάνει την οποιαδήποτε ενέργεια επί κάποιου στοιχείου της βάσης δεδομένων πρέπει να κάνει μια αίτηση για ένα lock πάνω στο στοιχείο το οποίο θέλει να επεξεργαστεί. Οι ενέργειες που μπορεί να κάνει ένα transaction σε ένα στοιχείο της βάσης δεδομένων είναι δύο. Η πρώτη ενέργεια είναι να διαβάσει ένα στοιχείο από την βάση δεδομένων και η δεύτερη ενέργεια είναι να το εγγράψει το στοιχείο το οποίο έχει διαβάσει. Τα locks τα οποία χρησιμοποιούνται είναι δύο, ένα για κάθε ενέργεια που θέλει να κάνει κάθε transaction, και είναι τύπου shared lock για την περίπτωση της ανάγνωσης και exclusive lock για την περίπτωση της εγγραφής. Από τα παραπάνω είναι προφανές πως μια ενέργεια ανάγνωσης προηγείται πάντα μιας ενέργειας εγγραφής για ένα συγκεκριμένο στοιχείο.

Κατά την εκτέλεση ενός νήματος (δηλ. την εκτέλεση ενός transaction) γίνονται αιτήσεις για νέα locks στην κλάση LockManager η οποία και είναι υπεύθυνη να γνωρίζει σε ποια transactions και σε ποια στοιχεία της βάσης έχει αναθέσει locks καθώς επίσης και το είδος του lock που έχει αναθέσει. Σε ένα συγκεκριμένο στοιχείο της βάσης δεδομένων μπορούν να ανατεθούν πολλά shared locks από διαφορετικά transaction αλλά στην περίπτωση που ένα στοιχείο έχει exclusive lock τότε δεν μπορεί να του ανατεθεί κανένα άλλο lock οποιουδήποτε τύπου. Μόλις ένα transaction έχει ένα lock τύπου shared για κάποιο στοιχείο της βάσης δεδομένων και θελήσει να εγγράψει το στοιχείο αυτό, κάνει μια νέα αίτηση για να αναβαθμίσει το shared lock σε exclusive lock.

Μόλις ένα transaction ολοκληρώνεται τότε στέλνει μήνυμα στον LockManager ούτος ώστε να απελευθερωθούν όλα τα locks τα οποία κατείχε το συγκεκριμένο transaction ώστε να είναι διαθέσιμα για κάποιο άλλο transaction.

4. Managing Logs

Στην έναρξη της εκτέλεσης ενός νέου transaction τοποθετείται στο αρχείο undo_redo.log μια νέα ετικέτα της μορφής <START T> η οποία υποδηλώνει την έναρξη του transaction. Σε κάθε εγγραφή ενός στοιχείου της βάσης δεδομένων προστίθεται και μια νέα ετικέτα με τα στοιχεία της εγγραφής που έγιναν. Τέλος αφού γραφούν όλα τα νέα δεδομένα του transaction στην βάση δεδομένων προστίθεται μια ακόμα ετικέτα που έχει την μορφή <COMMIT T> έτσι ώστε να υποδηλώσει πως το συγκεκριμένο transaction ολοκληρώθηκε επιτυχώς. Για να μπορέσουμε να εξετάσουμε την περίπτωση αστοχίας του συστήματος έχουμε προσθέσει το keyword «fail» οπού κατά την εκτέλεση των transactions αν ανιχνευτεί αυτό το keyword τότε το πρόγραμμα τερματίζεται αμέσως. Κατά την επανέναρξη του προγράμματος γίνεται η διαδικασία της αποκατάστασης της βάσης δεδομένων με τις σωστές τιμές για κάθε στοιχείο της και τοποθετείται στο αρχείο, που κρατούνται τα logs , μια ετικέτα <ABORT T> για κάθε μη ολοκληρωμένο transaction.



Διάγραμμα 4.1. Βασικές λειτουργίες ενός transaction.