Υλοποίηση Συστημάτων Βάσεων Δεδομένων

Εργασία 2

ΑΜ: 1115202000234 - ΚΡΑΤΗΜΕΝΟΣ ΧΡΗΣΤΟΣ

ΑΜ: 1115202000282 - ΦΥΚΑΣ ΑΘΑΝΑΣΙΟΣ

1. ΤΙ ΕΧΕΙ ΥΛΟΠΟΙΗΘΕΙ

Εχουμε υλοποιήσει όλες τις απαραίτητες συναρτησεις που αναφερει η εκφώνηση της εργασιας για τα B+ δεντρα. Ο κώδικας υποστηρίζει τη δημιουργία αρχείων, το άνοιγμα και το κλείσιμο του αρχείου, την εισαγωγή νέων εγγραφών, την αναζήτηση για υπάρχουσες εγγραφές και τον αυτόματο διαχωρισμό nodes του δεντρου.

2. ΠΑΡΑΔΟΧΕΣ:

Η προσέγγιση μας προϋποθέτει ότι τα IDs των records είναι μοναδικά και ότι εάν υπάρχει ήδη μια εγγραφή με το ίδιο ID η εισαγωγή του θα αγνοηθεί. Ο κώδικας προϋποθέτει επίσης ότι η ριζα του δεντρου, αφού δημιουργηθεί, παραμένει σε μια σταθερή θέση στη δομή του δέντρου και ότι το πρώτο μπλοκ δίσκου (block 0) χρησιμοποιείται πάντα ως μπλοκ metadata που δεν θα αποθηκεύσει ποτέ δεδομένα leaf ή index. Ως εκ τούτου, δεν διαχειριζομαστε πολλα edge cases.

3. ΜΑΙΝ ΣΥΝΑΡΤΗΣΕΙΣ

Η συνάρτηση BP_CreateFile() δημιουργει ένα νέο αρχείο δέντρου B+ δημιουργώντας ένα metadata block όπου αποθηκεύονται πληροφορίες όπως ο αριθμός του root node και το ύψος του δέντρου, στοιχεια απαραιτητα για την επεξεργασια του δεντρου. Στη συνέχεια, η συνάρτηση BP_OpenFile() ανοίγει αυτό το αρχείο και διαβάζει τα metadata σε μια δομή BPLUS_INFO στη μνήμη, ενώ η BP_CloseFile() χειρίζεται την εκκαθάριση και ελευθερώνει τυχόν πόρους που καναμε allocate κατα την εκτελεση του κωδικα μας.

Η εισαγωγή records εκτελείται από τη συνάρτηση BP_InsertEntry(). Εάν το δέντρο είναι κενό, αυτή η συνάρτηση χρησιμοποιεί την CreateRootNode()(δικη μας συναρτηση) για να δημιουργήσει έναν νέο κόμβο φύλλου και να τοποθετήσει το records στο δεντρο. Όταν το δέντρο δεν είναι κενό, ο κώδικας διασχίζει εσωτερικούς κόμβους χρησιμοποιώντας την TraverseToLeaf()(δικη μας συναρτηση) μέχρι να φτάσει στον σωστό κόμβο φύλλου. Σε εκείνο το σημείο, η InsertIntoLeaf()(δικη μας συναρτηση) εισάγει το record στην κατάλληλη θέση. Εάν δεν υπάρχει χώρος στον κόμβο φύλλου, η SplitLeafNode()(δικη μας συναρτηση) τον χωρίζει σε δύο κόμβους, κανει sort τις εγγραφές και επιστρεφει ένα κλειδί μέχρι τον γονικό κόμβο μέσω της UpdateParent(). Αυτή η διαδικασια έχει ως αποτέλεσμα την προσαρμογή

των εσωτερικών κόμβων ή ακόμα και τη δημιουργία μιας νέας ρίζας εάν η τρέχουσα ρίζα πρέπει να χωριστεί.

Η ανάκτηση records γίνεται από τη συνάρτηση BP_GetEntry(). Αυτή η συνάρτηση λειτουργει παρομοια με την insert, κανοντας traverse εσωτερικούς κόμβους μέχρι να βρει τον σωστό κόμβο φύλλου. Μόλις φτάσει στο τελευταιο φύλλο, σαρώνει τις εγγραφές που είναι αποθηκευμένες εκεί για να βρει μια αντιστοιχία με το ID Εάν βρεθεί, η εγγραφή επιστρέφεται στον καλούντα. Εάν όχι, η συνάρτηση υποδεικνύει ότι δεν υπάρχει αντίστοιχη εγγραφή. Σε όλες αυτές τις λειτουργίες, οι λειτουργίες χειρισμού block όπως BF_GetBlock(), BF_UnpinBlock() και BF_AllocateBlock() από τη βιβλιοθήκη BF καλούνται μέσω της CALL_BF().

.

4. ΣΥΝΑΡΤΗΣΕΙΣ ΓΙΑ ΒΟΗΘΕΙΑ

Εχουμε δημιουργησει μερικες συναρτησεις για την ευκολοτερη διαχειριση των operations στο B+ δεντρο, οπως επισης και ευκολοτερο debugging.

SplitLeafNode:
Όταν ένας κόμβος φύλλου είναι γεμάτος και δεν μπορεί να κρατησει ενα νεο record, η συνάρτηση SplitLeafNode() καλείται για να χειριστεί την υπερχείλιση. Αυτή η συνάρτηση εκχωρεί ένα νέο leaf block και μετακινει τις μισές εγγραφές από το αρχικό leaf στο νέο. Με αυτόν τον τρόπο, αναδιανέμει τα δεδομένα. Μετά την ανακατανομή, η συνάρτηση καθορίζει ποιο φύλλο θα λάβει το record που προκαλεσε το split. Ενημερώνει επίσης τους pointers για να διατηρήσει την αλυσιδα μεταξυ κόμβων. Τέλος, προωθεί ένα κλειδί μέχρι το parent node. Εάν δεν υπάρχει γονέας (δηλαδή, εάν αυτή η διαίρεση συμβαίνει στη ρίζα), δημιουργείται ένας νέος κόμβος ρίζας.

CreateRootNode:
Όταν το δέντρο είναι κενό και πραγματοποιείται η πρώτη εισαγωγή ή όταν ένας διαχωρισμός γινεται μέχρι το τέλος χωρίς να βρεθεί χώρος, η συνάρτηση CreateRootNode() χρησιμοποιείται για τη δημιουργία μιας νέας ρίζας. Αρχικά, εάν το δέντρο δεν έχει ρίζα, αυτή η συνάρτηση εκχωρεί ένα νέο block kai το ορίζει ως φύλλο, εισάγει την πρώτη εγγραφή και ενημερώνει τα μεταδεδομένα του δέντρου ώστε να αντικατοπτρίζει ότι αυτό το νέο block είναι τώρα η ρίζα και ότι το ύψος του δέντρου είναι μηδέν. Σε περιπτώσεις όπου ο διαχωρισμός κινείται προς τα πάνω, η CreateRootNode() μπορεί να κληθεί για να εισαγει μια ρίζα υψηλότερου επιπέδου, μετατρέποντας ένα δέντρο που βασιζόταν σε φύλλα στο παρελθόν σε δέντρο με εσωτερικούς κόμβους.

UpdateParent:
Μετά τον διαχωρισμό ενός κόμβου, μπορει να χρειαστει ενα κλειδί να "προωθηθεί" σε έναν εσωτερικό κόμβο. Η συνάρτηση UpdateParent() είναι υπεύθυνη για την εισαγωγή ενός προωθημένου κλειδιού στον σωστό εσωτερικό (γονικό) κόμβο. Αυτό το κλειδί λειτουργεί ως όριο που διαχωρίζει τα records που πηγαίνουν στον αριστερό και τον δεξιό κόμβο. Η

συνάρτηση βρίσκει τη σωστή θέση για το προωθημένο κλειδί στην ταξινομημένη λίστα κλειδιών του γονικού κόμβου, μετατοπίζει τα υπάρχοντα κλειδιά και τους δείκτες για να δημιουργήσει χώρο και, στη συνέχεια, εισάγει το νέο κλειδί μαζί με τους αντίστοιχους δείκτες στους νεους κόμβους. Χρειαζοματε αυτην τη συναρτηση καθως διασφαλίζει ότι οι επόμενες αναζητήσεις θα περιηγηθούν σωστά στην ιεραρχία του δέντρου.

InsertIntoLeaf:
Η συνάρτηση InsertIntoLeaf() χειρίζεται την περίπτωση της εισαγωγης μιας νέας εγγραφής σε έναν κόμβο φύλλου που έχει διαθέσιμο χώρο. Σαρώνει τις υπάρχουσες εγγραφές σε αυτό το φύλλο για να βρει το σωστό σημείο εισαγωγής, διατηρώντας τη σειρά ταξινόμησης κατά το redord ID. Εάν εντοπίσει ένα διπλότυπο κλειδί, απορρίπτει την εισαγωγή. Εάν δεν υπάρχει χώρος στο φύλλο, καλειται η SplitLeafNode() για να εξισορροπήσει ξανά το δέντρο.

GetBlockNumber:
Κάθε φορά που εκχωρείται ένα νέο block η συνάρτηση GetBlockNumber() βοηθά στον προσδιορισμό του πιο πρόσφατα εκχωρημένου block . Η συναρτηση ανακτά τον τρέχοντα συνολικό αριθμό block στο αρχείο και υπολογίζει το αναγνωριστικό του τελευταίου εκχωρημένου block.

5. ΠΡΟΒΛΗΜΑΤΑ ΤΗΣ ΥΛΟΠΟΙΗΣΗΣ

Οταν η BP_GetEntry() προσπαθει να διασχίσει το δέντρο, βλέπει μόνο αυτό που είναι ουσιαστικά μια δομή ενός επιπέδου, επιστρέφοντας επανειλημμένα στο ίδιο μπλοκ φύλλου (block 5) αντί να περιηγηθεί σε ολους τους κόμβους φύλλων. Σαν αποτελεσμα, δεν βρισκει ολα τα IDs που ψαχνουμε.