Σχολή Η.Μ.Μ.Υ. Πολυτεχνείου Κρήτης
ΠΛΗ 211 - Δομές Δεδομένων και Αρχείων (Εαρινό εξάμηνο 2016-2017)
Διδάσκων: Ε. Πετράκης
Όνομα φοιτητή : Χατζηνικολάου Μιχαήλ (Α.Μ. 2011030111)

1 B-Tree

Σε αυτό το μερος υλοιποιήθηκε μια δομή αρχείου τύπου $B-Tree$, η οποία αποθηκεύεται στο δίσκο, αφότου έχουν διαβαστει όλες οι λέξεις ενός αρχείου εισόδου τύπου ascii ".txt" .

1.1 Μέγεθος σελίδας.

Το μέγεθος σελίδας πρέπει να είναι υποχρεωτικά μεγαλύτερο από 512 bytes, καθώς κατα το Serialization των δεδομένων ενός πλήρους κόμβου, το μέγεθος των δεδομένων ένα κόμβο είναι περίπου ίσο με 340 bytes. Χρησιμοποιόντας τη γνώση ότι τα μεγέθη μνήμης ακολουθούν κάποια δύναμη του 2 ως μέγεθος, το N=512 συνίσταται να τεθεί ως κατώτατο όριο .

1.2 Βαθμός δέντρου N

Η δέσμευση του χώρου του δίσκου στο αρχείο εξόδου γίνεται ως εξής:

Το αρχείο αποτελείται από κόμβους του δυαδικού δέντρου, οι οποίοι δημιουργούνται κάθε φορά που υπάρχει υπερχείλιση του δέντρου (συγκεκριμένα της ρίζας του). Καθώς το αρχείο αποθηκεύεται σε serialized μορφή, υπάρχει κάποιο overhead το οποίο είναι

1. απρόβλεπτο, σε περίπτωση που μερος του κόμβου-σελίδας ειναι κάποιο αντικείμενο μη πρωταρχικού τύπου δεδομένων.

2. μεταβαλλόμενο, σε περίπτωση που κάποιος μη αρχικοποιημένος πίνακας του κόμβου αλλάζει σε μεγεθος.

Για τους παραπάνω λόγους, ο υπολογισμός του βαθμού N του δέντρου, υπολογίστηκε με την εξής μέθοδο:

Δημιουργούνται δοκιμαστικοί , αρχικοποιημένοι με ουδέτερες τιμές κόμβοι, με αυξανόμενο βαθμό / αριθμό κλειδιών. Στη συνέχεια, αποθηκεύονται σε ένα δοκιμαστικό αρχείο. Όταν υπάρξει υπέρβαση του μεγέθους σελίδας κατα την αποθήκευση των serialized δεδομένων, κατοχυρώνουμε ως βαθμό του δέντρου τον προηγούμενο βαθμό που δεν οδήγησε σε υπερχείλιση.

2 Ευρετήριο/Posting-List

Το ευρετήριο υλοποιήθηκε ως αρχείο δομής Posting-List και περιέχει μια συνδεδεμένη λίστα σελίδων με τις θέσεις κάθε λέξης που έχει βρεθεί στα αρχεία εισόδου. Η υλοποίηση αποτελείται από δυο μερη:

1. Υλοποίηση της λίστας στο δίσκο. Η δομή υλοποιήθηκε παρόμοια με τους κόμβους του δέντρου B-Tree.

   (α΄) Αρχικά, υπολογίσθηκε με διαδοχικές δοκιμές εγγραφής, το ανώτατο επιτρεπτό όριο κλειδιών σε κάθε κόμβο όπως και στο προηγούμενο ερώτημα.

   (β΄) Κάθε υπερχείλιση σελίδας εγγράφεται στο τέλος του αρχειου (ως νεα σελίδα) και η προηγούμενη δείχνει σε αυτή. Αυτό επαναλαμβάνεται κάθε φορά που μια σελίδα υπερχείλισης γεμίζει.

2. Διασύνδεση με το Λεξικό τύπου B-Tree.

   Κάθε φορά που εισάγουμε νέα δεδομένα προς καταγραφή, ελέγχεται αν υπάρχει η συγκεκριμένη λέξη καταγεγραμμενη μεσα στο αρχείο Λεξικού. Αν υπάρχει, τότε εγγράφεται στο ευρετήριο η νέα καταχώρηση θέσης. Αν δεν υπάρχει, δημιουργείται νέα σελίδα ευρετηρίου και εγγράφεται σε αυτή.

3 Αναζήτηση λέξεων στο δίσκο.

Κατά την αναζήτηση των λέξεων στα αρχεία εισόδου, χρησιμοποιείται η δομή που περιγράφηκε στα προηγούμενα μερη και την εκφώνηση.

Τα αρχεία αυτά βρίσκονται στον σκληρό δίσκο. Η αναζήτηση χρησιμοποιεί μία λέξη εισόδου και αναζητά αρχικά την καταχώρηση στο Λεξικό. Κατοπιν, διαβάζονται όλες οι καταχωρήσεις του ευρετηρίου (αρχική σελίδα και οι αντίστοιχες σελίδες υπερχείλισης) και τα αποτελέσματα εμφανίζονται στην έξοδο. Το κόστος προσβάσεων στο δίσκο καταμετράται σε static τύπου μεταβλητές, μία για τις προσβάσεις (τόσο αναγνώσεις όσο και εγγραφές) στο Λεξικό και μια για τις προσβάσεις στο Ευρετήριο.

3.1 Αποτελέσματα μετρήσεων. (Μέγεθος σελίδας = ελάχιστο δυνατό = 512 *bytes*)

1. Α. Μέσος όρος προσβάσεων στο B-Tree και την Posting List κατά την ανάγνωση των 3 αρχείων:

   17.07 **προσβάσεις ανά λέξη.** (7.13 Posting List, 9.94 B-Tree).

2. Β. Μέσος όρος προσβάσεων για αναζήτηση γνωστών λέξεων. (Χρησιμοποιήθηκαν αποκόμματα των 3 αρχείων σε ένα κείμενο Excerpts.txt .

   6.29 **Posting List** 4.47 **B-Tree.**

3. Γ. Μέσος όρος προσβάσεων για αναζήτηση τυχαίων λέξεων. Γι αυτό τον σκοπό χρησιμοποιήθηκαν οι πρώτες 100 λέξεις την ορκομωσίας ενός ακόμα προέδρου των ΗΠΑ.

   7.07 **Posting List** 4.48 **B-Tree.**

3.2 Συμπεράσματα/Παρατηρήσεις απόδοσης

1. B-Tree

   Κατά τη δημιουργια του B-Tree , υπάρχει το ενδεχόμενο υπερχείλισης κόμβων. Αυτό κοστίζει 2 προσβάσεις δίσκου για την αποθήκευση των δύο κόμβων που προκύπτουν, και άλλες δύο για την ανάκτηση και αποθήκευση του κόμβου-γονέα. Αυτό συμβαίνει κατά την εισαγωγή δεδομένων. Στην αναζήτηση όμως έχουμε O($log_m n$) αναζητήσεις , όπου μ ο βαθμός του δεντρου και ν οι συνολικές καταχωρήσεις τη χρονική στιγμή της αναζήτησης.

2. List

   Κατά τη δημιουργία αλλά και την αναζήτηση έχουμε σταθερά κάποιο χρόνο $O(n)$, καθώς ο μόνος τρόπος να υπάρξουν περισσότερες προσβάσεις είναι η υπερχείλιση. Μπορούμε να πούμε , πως οι λειτουργίες αυτές έχουν πολυπλοκότητα κάποια συνάρτηση $O(n/p)$, με p τη χωρητικότητα σελίδας σε κλειδιά.

4 Παρατηρήσεις πάνω στο πρόγραμμα.

4.1 Επιπλέον/Διαφορετικές λειτουργίες.

Το πρόγραμμα έχει τη δυνατότητα να προσαρμόζεται στο μέγεθος σελίδας του δισκου. Κατα τη δημιουργία τους, τόσο το B-tree όσο και η Posting List όπως περιγράφεται παραπάνω, προσαρμόζονται ούτως ώστε να έχουν τη μέγιστη δυνατή χωρητικότητα σε καταχωρήσεις. Αυτός είναι άλλωστε και ο κύριος λόγος χρήσης των B-Trees, η ταύτιση (όσο περισσοτερο το δυνατόν) του μεγέθους κόμβου με τη σελίδα δισκου.

Επιπλέον , το πρόγραμμα μπορει να διαβάσει οποιοδήποτε αρχείο εισόδου τύπου txt μπει ως επιπλέον είσοδος (αντίστοιχο runtime menu).

4.2 Σφάλματα.

Όταν λέξεις ακολουθούνται από σημεία στίξης, αποθηκεύονται μαζι με αυτά ως νέες καταχωρήσεις. (Δε γίνεται επαρκώς το parsing) Επίσης, το πρόγραμμα δε λειτουργεί για μεγέθη σελίδας μικρότερα των 512 bytes. Αυτό οφείλεται στο γεγονός ότι αποθηκεύονται πολλά δεδομένα κατά την εγγραφή στο δίσκο για κάθε κόμβο του δέντρου ή σελίδα της λίστας (δείκτες σε συνδεδεμένες σελίδες, χωρητικότητα κλειδιών κλπ.).