

Ανάπτυξη Λογισμικού για Υπολογιστικά Συστήματα (PROJECT)

Μέρος Πρώτο: Τελεστής Radix Hash Join

Φοιτητές:

Ονοματεπώνυμο: Κατσορίδας Ιωάννης
Αριθμός Μητρώου: 1115201400066
SDI: sdi1400066

Ονοματεπώνυμο: Μητράκης Γεώργιος
Αριθμός Μητρώου: 1115201400107
SDI: sdi1400107

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2018-2019

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Εκτελέσιμα:

`./DataGenerator <n>`

όπου:

$n = 1$ για τη δημιουργία relations με increasing payloads ($== 0,1,2...REL_SIZE$),

$n = 2$ για τη δημιουργία relations με identical payloads ($== 1$),

$n = 3$ για τη δημιουργία relations με exclusive per relation payloads ($R \rightarrow 1$ και $S \rightarrow 2$),

$n =$ οτιδήποτε άλλο για τη δημιουργία relations με random payloads

`./Caramel`

για την δημιουργία των relations και την πράξη του join.

`./UnitTests`

για το unit testing των συναρτήσεων

Μεταγλώττιση:

`make`

Αρχεία και περιεχόμενά τους:

- `DataGenerator.c`, `DataGenerator.h`

Το πηγαίο και το αρχείο βιβλιοθήκης του, από τα οποία προκύπτει το εκτελέσιμο που δημιουργεί 2 αρχεία με ονόματα “DataRelationR.txt” και “DataRelationS.txt” με τυχαία payloads. Το εκτελέσιμο αυτό εκτελείται ανεξάρτητα από τα υπόλοιπα κομμάτια και εκτελείται μόνο αν θέλει ο χρήστης να αλλάξει τα δεδομένα με τα οποία τρέχει τον αλγόριθμο.

- `DataParse.c`, `DataParse.h`

Το πηγαίο και το αρχείο βιβλιοθήκης του που περιλαμβάνει τις συναρτήσεις για δημιουργία relation από τα παραπάνω txt files, δημιουργία ιστογράμματος, αθροιστικού ιστογράμματος και δημιουργία νέου κατακερματισμένου relation από ένα ήδη υπάρχον, όπως αυτά περιγράφονται στην εκφώνηση

- `Globals.h`

Το κύριο header file, που περιλαμβάνει ορισμούς global δομών συναρτήσεων και μεταβλητών και γίνεται included σε όλα τα πηγαία.

- `main.c`

Περιλαμβάνει την `main` συνάρτηση του εκτελέσιμου, η οποία δημιουργεί δύο relations `R` και `S`, κάνει Radix Hash Join , αποθηκεύει το αποτέλεσμα του join αυτού και επιστρέφει.

- `RadixHashJoin.c`

Το πηγαίο που κάνει Join δύο relations. Δημιουργεί ιστόγραμμα και αθροιστικό ιστόγραμμα για κάθε relation, τα κατακερματίζει δημιουργώντας δύο νέα relations και κάνει αναζήτηση για ίσα payloads μεταξύ στοιχείων του `R` και του `S` που ανήκουν στο ίδιο bucket. Δημιουργείται ευρετήριο για το bucket του relation με τα λιγότερα tuples και για κάθε tuple του `R` που είναι ίσο με κάποιο tuple του `S` στο ίδιο bucket, προστίθενται τα keys στη λίστα αποτελεσμάτων, η οποία και επιστρέφεται.

- `Results.c`, `Results.h`

Περιέχουν συναρτήσεις για διαχείριση της λίστας αποτελεσμάτων του Join.

- `Index.c`, `Index.h`

Περιέχουν τη δομή του `index {bucket array, chain array}` και συναρτήσεις που διαχειρίζονται τη δημιουργία, την αρχικοποίηση, την αναζήτηση στον εκάστοτε κάδο και τη διαγραφή του.

UnitTests

Όλες οι συναρτήσεις που υλοποιήθηκαν δοκιμάζονται για την αποτελεσματικότητά τους μέσω unit test. Τα test αυτά υλοποιήθηκαν με τη χρήση του Framework: CUnit και βρίσκονται στο αρχείο `UnitTests.c`. Ως είσοδο παίρνουν καρφωτά συγκεκριμένες τιμές και ελέγχεται η έξοδος από τις διάφορες συναρτήσεις ως προς την εγκυρότητά τους με βάση την ομοιότητά τους με τα αναμενόμενα αποτελέσματα.