

Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα 2018-2019

Τελική Αναφορά

Συμμετέχοντες:

- | | |
|----------------------------|---------------|
| • Κωνσταντίνος Μαρκόπουλος | 1115201500091 |
| • Δημήτριος Ψήτος | 1115201500183 |
| • Λάζαρος Ζερβός | 1115201500042 |

Σύνοψη

Στο σύνολο της η εργασία εκπονήθηκε σύμφωνα με τις οδηγίες που δόθηκαν τόσο από τις εκφωνήσεις όσο και από την καθοδήγηση των επιμελητών. Μοναδική εξαίρεση αποτέλεσαι το τρίτο μέρος της εργασίας. Σε αυτό, και συγκεκριμένα στο κομμάτι του Join Enumeration, υλοποιήθηκαν δύο διαφορετικές προσεγγίσεις, οι οποίες αναφέρονται αναλυτικότερα στη συνέχεια.

Πρώτη προσέγγιση - Brute Force:

Στην πρώτη προσέγγιση θα μπορούσε κανείς να κάνει λόγο για Predicate Enumeration και όχι Join Enumeration. Ο αλγόριθμος βασίστηκε στο γεγονός ότι μέχρι και το τρίτο μέρος της εργασίας, το πρόγραμμα διαχειριζόταν Queries μέσω Predicates και Filters, και μέσω αυτών πραγματοποιούνταν και οι Εκτιμήσεις Πληθικότητας (βλ. cardinality.cpp - .hpp).

Μέσω της προσέγγισης αυτής ορίζεται η σειρά με την οποία θα πρέπει να εκτελεστούν τα Predicates έτσι ώστε να υπάρξουν τα λιγότερα ενδιάμεσα αποτελέσματα. Κατ'επέκταση δε χρειάζονται περαιτέρω υπολογισμοί και ορίζοντας αυτή σειρά, ορίζεται με έμμεσο τρόπο και η σειρά με την οποία θα πρέπει να γίνουν οι ζεύξεις.

Δεύτερη προσέγγιση - DP-Linear-1:

Στη δεύτερη προσέγγιση ακολουθήθηκε ο αλγόριθμος DP-Linear-1, όπως αυτός ορίζεται στη στο [παρόν έγγραφο](#) (σελ.84 - 85 του PDF) που δόθηκε μέσω της εκφώνησης του τρίτου μέρους.

Σύμφωνα με τον αλγόριθμο αυτό, εξετάζονται σταδιακά όλοι οι πιθανές ακολουθίες σχέσεων που μπορεί να υπάρξουν. Οι ακολουθίες αυτές αναφέρονται στη σειρά με την οποία θα πρέπει να πραγματοποιηθούν οι ζεύξεις μεταξύ των σχέσεων, και φυλάσσεται αυτή με το μικρότερο κόστος (πλήθος ενδιάμεσων αποτελεσμάτων).

Benchmarking

Small dataset

Execution method*	Loading	Optimizer	Filters	Radix Join	Interm. build	Interm. update	Selectors	Total
BF-1	0.024120	0.000503	0.009566	0.257115	0.007357	0.695396	0.096435	1.100
BF-2	0.025517	0.000444	0.010496	0.311914	0.010903	0.695483	0.097938	1.100
BF-4	0.009161	0.000379	0.010087	0.352241	0.018335	0.739711	0.097613	1.200
DPL-1	0.026288	0.000069	0.009445	0.300101	0.020547	0.728464	0.096047	1.200
DPL-2	0.028802	0.000072	0.009636	0.349601	0.016368	0.766361	0.097328	1.250
DPL-4	0.025655	0.000071	0.009653	0.396944	0.016923	0.757884	0.096959	1.200

*[method] - [# of threads]

Tested on i5-8250U CPU @ 1.60GHz × 8, single-channel 8GB @ 2400MHz

Most heavy

Από τα αποτελέσματα που παρουσιάζονται στον παραπάνω πίνακα, παρατηρούμε ότι ο περισσότερος χρόνος καταναλώνεται κατά την ενημέρωση των ενδιάμεσων αποτελεσμάτων.

Threads

Δεν παρατηρούνται διαφορές στους χρόνους εκτέλεσης με διαφορετικό πλήθος threads. Αυτό οφείλεται και στο γεγονός ότι οι διαδικασίες είναι memory heavy και λόγω single channel memory, συμβαίνει bottleneck. Επίσης, τα jobs των threads έχουν μικρό φόρτο για το small dataset και σπαταλάται χρόνος στον συγχρονισμό των threads.

Παρατηρήσεις

Αξιοσημείωτες αποτελούν ορισμένες διαφορές που παρατηρήθηκαν κατά τη σύγκριση της λειτουργίας των δύο προσεγγίσεων (BF και DPL). Μετά την εξαγωγή των αποτελεσμάτων σε αρχεία, οι εκτιμήσεις που παρήγαγαν οι δύο αλγόριθμοι φαίνεται να διαφωνούν σε 9 από τα 50 queries. Για 5 από αυτά η διαφορά είναι αμελητέα καθώς τα ενδιάμεσα αποτελέσματα παραμένουν κοινά σε αριθμό.

Σημείωση: Όσα αναφέρονται παραπάνω αφορούν το small dataset.