Εργασία: Ανάκτηση Πληροφορίας
ACM SIGMOD 2013 Programming Contest

Ομάδα εργασίας:
Χρήστος Μελίδης ΑΕΜ 1816
Κυριάκος Αδάμ ΑΕΜ 1890
Ηρακλής Μουτίδης ΑΕΜ 1214

Για την εργασία υλοποιήθηκαν ένα prefix-trie (trie.hpp) και ένα hash table(HashTable.hpp), μια βάση για τις λέξεις, που βασίζεται στο trie (wordTrie.hpp) και μια δομή word για τις λέξεις (word.hpp).

Κάθε λέξη που έρχεται ασχέτως αν ανήκει σε Document ή σε Query αποθηκεύεται στο Trie, όπου και αποκτά ένα global id. Έτσι απαλείφονται τα διπλότυπα και επιταχύνεται η διαδικασία της ανάκτησης της λέξης.

Για κάθε Document, κάθε λέξη του, αποθηκεύεται στο hash table του Document και έχουμε exact match σε O(1). Υλοποιήθηκαν 24 threads τα οποία και διαμοιράζονται τα Documents.

Κάθε thread έχει συγκεκριμένα documents που επεξεργάζεται ανάλογα με το ThreadId του, οπότε και διευκολύνεται ο παραλληλισμός. Αρχικά, εκκινούνται τα threads τα οποία και περιμένουν μέχρι να έρθει το πρώτο Query/Document. Όταν έρθει κάποιο από τα δύο εισάγεται στην αντίστοιχη λίστα.

Τα queries αναλύονται με το που έρθουν, ενώ τα Documents προστίθενται σε μια λίστα αναμονής για να αναλυθούν. Όταν γίνει η ανάλυση και ανάλογα με το Id τους προστίθενται στο workload του ανάλογου thread. Μετά καλείται η Prepare που αναλαμβάνει την συμπλήρωση της λίστας των Queries που ταιριάζουν σε κάθε Doc.

Η όλη διαδικασία διακόπτεται κάθε φορά που ζητούνται τα αποτελέσματα κάποιου Doc, οπότε και επιστρέφονται και διαγράφεται το Doc από τις ενεργές λίστες.
Η όλη προσπάθεια μείωσε τον χρόνο του small test στο μισό από την brute force υλοποίηση του διαγωνισμού.

Για το small_test ανάλογα με τον υπολογιστή που χρησιμοποιήθηκε ο χρόνος κυμαίνεται από τα ~22s:400ms μέχρι τα ~36s:000ms (σε παλιότερης τεχνολογίας μηχάνημα) και μια δοκιμή σε ένα καινούριο υπολογιστή έδωσε χρόνο ~16s:500ms

Εκτέλεση
σε terminal στον φάκελο που βρίσκεται το pdf
• make
• ./testdriver