Βασίλης Μπούργος ΑΜ: 197

Εισαγωγή

Το πρόβλημα με το οποίο ασχολείται η παρούσα δημοσίευση αφορά την ευρετηριοποίηση μεγάλων συλλογών εγγράφων πάνω σε ένα P2P δίκτυο. Ένας σημαντικός περιορισμός σε ένα P2P δίκτυο είναι η κατανάλωση bandwidth (εύρος ζώνης) καθώς επίσης και ο περιορισμένος χώρος μνήμης των κόμβων.

Ο αλγόριθμος που παρουσιάζεται είναι query driven, δηλαδή για την ευρετηριοποίηση χρησιμοποιεί πληροφορία που εξάγεται από τις επερωτήσεις. Το ευρετήριο που κατασκευάζεται αποθηκεύει αναφορές στα top-k έγγραφα. Επιπλέον, χρησιμοποιούνται στατιστικά στοιχεία για την ευρετηριοποίηση των συνδυασμών των όρων που αναζητούνται πιο συχνά από τους χρήστες. Τέλος, υπάρχει ένας μηχανισμός για την ανανέωση των περιεχομένων του ευρετηρίου ανάλογα με το περιεχόμενο των επερωτήσεων.

Υπάρχουν δύο προσεγγίσεις όσον αφορά τον τρόπο δημιουργίας και συντήρησης του ευρετηρίου: η HDK και η DCT οι οποίες έχουν χρησιμοποιηθεί σε παλιότερες εργασίες. Με την HDK (*Highly Discriminative Keys*) προσέγγιση κάνουμε ευρετηριοποίηση των όρων ή συνδυασμών όρων *(κλειδιά-keys*) που εμφανίζονται πιο συχνά με βάση ένα κατώφλι DF_{max}. Αυτή η τεχνική έχει το μειονέκτημα ότι απαιτεί πολύ χώρο. Με την DCT (*Distributed Cache Table*) προσέγγιση δημιουργείται ένα ευρετήριο on-the-fly στο οποίο γίνονται caching τα αποτελέσματα των πιο συχνών επερωτήσεων. Έτσι, για κάθε επερώτηση γίνεται πρώτα αναζήτηση σε αυτό το ευρετήριο και αν δεν υπάρχει εκεί αποτέλεσμα, το ερώτημα γίνεται broadcast στο υπόλοιπο δίκτυο. Σε αυτή την περίπτωση απαιτείται λιγότερος χώρος, αλλά αυξάνεται το φορτίο του δικτύου.

Η προσέγγιση που χρησιμοποιείται εδώ είναι συνδυασμός της HDK και της DCT. Στόχος είναι να μειωθεί ο χώρος που καταλαμβάνεται με την HDK άλλα και να περιορίσουμε τις broadcast εκπομπές των επερωτήσεων που είναι σπάνιες (κέρδος στο bandwidth).

Το μοντέλο-Γενική ιδέα

Το μοντέλο του δικτύου είναι το ακόλουθο. Έστω N κόμβοι P_i, $1<i<N$ και μια συλλογή εγγράφων D που περιέχει M έγγραφα. Με T_D συμβολίζουμε το λεξιλόγιο του D. Κάθε κόμβος έχει τις εξής αρμοδιότητες: Πρώτον, διατηρεί ένα τμήμα D_i της

συνολικής συλλογής D και δημιουργεί ένα ευρετήριο πάνω σε αυτό και δεύτερον, συμμετέχει στην συντήρηση και αποθήκευση του global ευρετηρίου το οποίο αφορά όλη τη συλλογή. Το κομμάτι του global ευρετηρίου για το οποίο είναι υπεύθυνος κάθε κόμβος P_i περιέχει ζεύγη της μορφής (k, PL(k)), όπου k το κλειδί και PL(k) η αντίστοιχη posting list, για όλα τα κλειδιά και τις posting lists που έχουν ανατεθεί στον P_i από τον κατανεμημένο πίνακα κατακερματισμού (Distributed Hash Table-DHT). Κατά τη επεξεργασία μιας επερώτησης ο κόμβος P_i αλληλεπιδρά με το δίκτυο για να ανακτήσει τα έγγραφα που ικανοποιούν την επερώτηση, ενημερώνει τα στατιστικά στοιχεία και ανάλογα με τα νέα στατιστικά μπορεί να ενεργοποιήσει το μηχανισμό ενεργοποίησης κλειδιού.

Αρχικά, κάθε κόμβος κάνει την τοπική ευρετηριοποίηση και εισάγει τα κλειδιά που προκύπτουν στο δίκτυο. Στη συνέχεια, όταν ένας κόμβος λαμβάνει μια επερώτηση αρχίζει να την αναλύει σε υποσύνολα ανάλογα με τους όρους που περιέχει (υποερωτήματα).

Για κάθε υποερώτημα q' ο κόμβος ελέγχει αν το q' σχετίζεται ήδη με κάποια posting list, ώστε να ανανεώσει/ανακτήσει την αναμενόμενη πιθανότητα χρήσης (estimated probability of use-EPU). Αν η EPU ξεπερνάει κάποιο κατώφλι EPU_{min} τότε το q' θα πρέπει όπως λέμε να γίνει ενεργό κλειδί. Αυτό σημαίνει ότι οι όροι που αποτελούν το q' είναι συχνοί, οπότε πρέπει να δημιουργήσουμε και να ευρετηριοποιήσουμε ένα νέο κλειδί. Για να ενημερώσουμε τους κόμβους που έχουν έγγραφα που περιέχουν το νέο κλειδί, χρησιμοποιούμε το μηχανισμό ONM (Opportunistic Notification Mechanism). Ο μηχανισμός αυτός έχει ως στόχο να κάνει την ενημέρωση με αποτελεσματικότερο τρόπο από ότι το broadcasting. Στηρίζεται στην ιδέα του shower multicast, δηλαδή η broadcast μετάδοση 'σπάει' σε multicast sessions τα οποία κάθε φόρα 'ψαλιδίζουν' τα έγγραφα με χαμηλή συχνότητα εμφάνισης των όρων. Επίσης, χρησιμοποιείται το piggybacking και οι επιβεβαιώσεις είναι μικρές και χαμηλής προτεραιότητας.

Φιλτράρισμα

Για τον περιορισμό του μεγέθους του ευρετηρίου εφαρμόζεται φιλτράρισμα των όρων των κλειδιών με κάποιον από τους τρόπους που περιγράφονται παρακάτω:

- Με βάση το μέγεθος (μέγιστο μέγεθος s_{max} στο πλήθος των όρων)
- Διαχωρισμός των κλειδιών. Κάθε κλειδί χαρακτηρίζεται ως discriminative ή non-discriminative. Ως discriminative ορίζουμε τα κλειδιά που εμφανίζονται στα έγγραφα

της συλλογής το πολύ DF_{max} φορές. Αντίθετα, τα non-discriminative κλειδιά εμφανίζονται πάνω από DF_{max} φορές.

- Φίλτρα πλεονασμού. Τα φίλτρα αυτά χρησιμοποιούν τον παραπάνω διαχωρισμό και την ιδιότητα ότι αν ένα discriminative κλειδί k_1 περιέχει ένα μικρότερου μεγέθους discriminative κλειδί k_2 τότε η posting list του k_1 μπορεί να παραχθεί με τοπική επεξεργασία της posting list του k_2, άρα αρκεί να κρατάμε στο global ευρετήριο πληροφορία μόνο για το k_2.

Αλγόριθμοι ευρετηριοποίησης/ανάκτησης

Στην παρούσα εργασία παρουσιάζονται δύο αλγόριθμοι που χρησιμοποιούνται, ο ένας για ευρετηριοποίηση και ο άλλος για ανάκτηση.

Ο πρώτος αλγόριθμος (ευρετηριοποίηση) εκτελείται σε κάθε κόμβο όταν ένα νέο κλειδί k γίνεται ενεργό. Ουσιαστικά ο κάθε κόμβος εκτελεί μια τοπική αναζήτηση του k και αν το αποτέλεσμα δεν είναι κενό στέλνει τη λίστα των εγγράφων που έχουν τιμή πάνω από minRank στον κόμβο που είναι υπεύθυνος για το κλειδί k.

Ο δεύτερος αλγόριθμος (ανάκτηση) εκτελείται στους κόμβους που είναι υπεύθυνοι για τα κλειδιά που εμφανίζονται στο q. Εκεί γίνεται ανανέωση των στατιστικών για κάθε υποκλειδί k που προκύπτει και γίνεται ανάκτηση των posting lists. Αν κάποιο υποκλειδί k δεν υπάρχει ως τότε και έχει $EPU(k)>EPU_{min}$ τότε δημιουργείται (μηχανισμός ενεργοποίησης κλειδιού) και προστίθεται στη λίστα νέων κλειδιών. Σε περίπτωση που η συχνότητά του είναι μικρότερη από DF_{max} τότε ανήκει στα discriminative οπότε πρέπει να διαγράψουμε όλα τα νέα κλειδιά k' που προέκυψαν και για τα οποία ισχύει $k \subset k'$. Τέλος, θα πρέπει να ενημερώσουμε τον DHT με τα νέα κλειδιά που δημιουργήθηκαν, με χρήση του ONM.

Scalability

Η προσέγγιση που μελετάμε κάνει το IR σύστημα να είναι scalable. Αυτό συμβαίνει επειδή οι posting lists που διακινούνται στο δίκτυο φράσσονται από την παράμετρο DF_{max}. Επιπλέον, ο δεύτερος παράγοντας που επηρεάζει το scalability, το πλήθος των κλειδιών που παράγονται κατά την ευρετηριοποίηση, είναι επίσης φραγμένος χάρη στο φιλτράρισμα που εφαρμόζεται με τους τρόπους που αναφέραμε πριν. Τελικά, οι συγγραφείς αποδεικνύουν ότι το πλήθος των κλειδιών αυξάνεται γραμμικά με το πλήθος των queries.

Πειράματα

Στα πειράματα χρησιμοποιήθηκαν επερωτήσεις που είχαν τεθεί στη μηχανή αναζήτησης της Wikipedia. Στο $1^{ο}$ χρησιμοποιήθηκαν 3000 επερωτήσεις ως σύνολο δοκιμής και ανακτήθηκαν τα top-20 αποτελέσματα με χρήση του Google και με τις ίδιες επερωτήσεις τις posting lists του μοντέλου. Κατόπιν, έγινε σύγκριση των δύο αποτελεσμάτων. Σημαντική παρατήρηση είναι το γεγονός ότι για μέγεθος κλειδιού πάνω από 3 δεν έχουμε ουσιαστική βελτίωση. Ωστόσο, η επικάλυψη με τα αποτελέσματα του Google, για συχνά ερωτήματα, έφτασε σε αρκετές περιπτώσεις το 80%. Στο $2^{ο}$ πείραμα μετριέται ο αριθμός των παραγόμενων κλειδιών κατά τη διάρκεια της επεξεργασίας των επερωτήσεων. Οι παράμετροι DF_{max}, EPU_{min} και s_{max} έχουν τιμές 100, 4/(2*M) και 3 αντίστοιχα. Αυτό που παρατηρήθηκε ήταν η μείωση του αριθμού των παραγόμενων κλειδιών σε σχέση με την HDK προσέγγιση, πράγμα που ισοδυναμεί με μείωση της κίνησης στο δίκτυο. Επίσης, έγιναν συγκρίσεις επικάλυψης αντίστοιχες με το $1^{ο}$ πείραμα, αλλά με τη μηχανή ανάκτησης Terrier. Τα αποτελέσματα ήταν εξίσου ικανοποιητικά.