Κατανεμημένα Συστήματα

Διευκρινήσεις Project Μαθήματος

Στα πλαίσια του project του μαθηματος σας δίνεται η επιλογή να υλοποιήσετε **MIA** από τις 2 προσεγγίσεις της εργασίας του μαθήματος:

Πρώτη προσέγγιση (απλή version του memcache συστήματος)

Σε αυτήν την προσέγγιση, θα αξιοποιήσετε την χρήση του ταχυδρομικού κώδικα για την αναζήτηση διαδρομών.

- Ο χρήστης θα δίνει ένα αρχικό και ένα τελικό σημείο για το Query του. Αυτό μπορεί να είναι είτε ένα pair σαν αυτό ((StartLat,StartLon),(EndLat,EndLon)), είτε ένα pair από ταχυδρομικούς κώδικες (ταχυδρομικός κώδικας αρχικής, ταχυδρομικός κώδικας τελικής θέσης).
- Ρωτάει στην cache αν υπάρχει η διαδρομή που να ανταποκρίνεται σε αυτό το pair (είτε ταχ. Κώδικα είτε LatLon). Αν δεν υπάρχει, θα πρέπει να ρωτήσει τους workers. Σε αυτήν την περίπτωση, μπορείτε να κάνετε hash τον ταχυδρομικό κώδικα του αρχικού σημείου + τον ταχυδρομικό κώδικα του τελικού σημείου(αν το query σας είχε τη μορφή (ταχ. κώδ. Αρχής, ταχ. κώδ. Τέλους)), και με τον τρόπο που έχουμε αναφέρει (hash των ip+port και comparison αυτών των values) θα βρίσκετε ποιος worker είναι υπεύθυνος (ή κατά αντιστοιχία με παρόμοιο τρόπο με τα lat lon).
- Ο worker που είναι υπεύθυνος για το συγκεκριμένο hash θα πηγαίνει και θα αναζητά σε ένα τοπικό του αρχείο (μπορείτε και τοπική βάση στο μηχάνημα να στήσετε, είναι στην δική σας διακριτική ευχέρεια να το υλοποιήσετε όπως θέλετε), την διαδρομή που έχει το ίδιο ζεύγος (είτε αυτό είναι lat lon είτε αυτό είναι ταχ. κώδικες). Την στέλνει στον reducer και εκείνος με τη σειρά του στο master και συνεπώς στη συνέχεια και στον ίδιο τον χρήστη. **Για την δημιουργία του τοπικού αρχείου με τις διαδρομές** μπορείτε αξιοποιήσετε κάποια βάση δεδομένων, αλλά αυτή θα είναι μόνο για να έχετε μια αρχικοποίηση με διαδρομές στο τοπικό αρχείο που έχει ο κάθε worker ώστε να μην έχετε ένα άδειο σύστημα (χωρίς περαιτέρω να χρειάζεται να διαβάζετε από τη βάση δεδομένων ή να γράφετε σε αυτήν).
- Αν δεν βρει κάποια διαδρομή μέσα στο αρχείο που αναζητά, τότε **ο worker ρωτά** το Google Directions API (και όχι ο master), βρίσκει τη διαδρομή, την στέλνει στον reducer και εκείνος την επιστρέφει στον master (για να την αποθηκεύσει εκείνος στην cache του) καθώς επίσης ο worker την αποθηκεύει τοπικά στο αρχείο του με τις διαδρομές.

Δεύτερη προσέγγιση (bonus 10% στη βαθμολογία του project)

Σε αυτή την προσέγγιση, ακολουθείτε τα βήματα όπως ακριβώς τα έχουμε εξηγήσει στα εργαστήρια με τις εξής αλλαγές:

- Ο κάθε worker **διαβάζει από ένα τοπικό αρχείο** τις εγγραφές του με τις διαδρομές. Αν επιθυμείτε να αξιοποιήσετε κάποια βάση δεδομένων αυτή θα είναι μόνο για να έχετε μια αρχικοποίηση από διαδρομές στο τοπικό αρχείο

με τις διαδρομές που έχει ο κάθε worker ώστε να μην έχετε ένα άδειο σύστημα (χωρίς περαιτέρω να χρειάζεται να διαβάζετε από τη βάση δεδομένων ή να γράφετε σε αυτήν).

- Αν κατά την διάρκεια αναζήτησης της διαδρομής, αυτή δεν βρεθεί στην cache του master node και ούτε ο reducer μας επιστρέψει αποτέλεσμα(δηλαδή το result είναι null) τότε ο master node (αντί να ζητήσει prompt από τον χρήστη όπως είχαμε αναφέρει), βρίσκει με την γνωστή διαδικασία του hash, ποιος worker είναι υπεύθυνος γι'αυτό το ζεύγος ((StartLat,StartLon),(EndLat,EndLon)). **Αυτός** (**ο worker** που είναι υπεύθυνος) **αναλαμβάνει να ρωτήσει στο Google Directions API** γι'αυτήν την διαδρομή την οποία ψάχνουμε και να την στείλει τόσο στον master node για να την αποθηκεύσει στην cache του, όσο επίσης την αποθηκεύει στο τοπικό του αρχείο με τις διαδρομές του.
- Υπενθυμίζουμε ότι το query πάει σε όλους τους workers και καθένας ψάχνει το τοπικό του αρχείο με τις διαδρομές. Τα αποτελέσματα στέλνονται στον reducer ο οποίος αναλαμβάνει να κάνει την συγχώνευση σε ένα object το οποίο θα στείλει στον master. Από εκεί αυτός θα βρίσκει τις πιο κοντινές διαδρομές όπως ακριβώς έχουμε αναφέρει (βάσει του distance από την αρχική και τελική θέση).