Neo4j and MySql comparison experiment

With caching

The experiment is run on a scientific linux machine with both databases running on it locally. The Neo4j server is embedded in the java application and the MySql server is running on it, too.

The experiment is performed separately for neo4j and mysql, to exclude any possibility of interaction between them. Before the experiment is performed and any measurements are recorded the system and the database is warmed up. The warm up consists of connecting to an instance of mysql or an instance of neo4j graph and running some queries on them. The queries are taken in order from the actual queries that will be executed to allow the database to do some caching, and the length of the warm up is at least one second.

The experiment is performed on random trees that have certain size and depth. The trees are generated using random seeds, so the same tree can be generated again using the same depth, size and random seed. It is planned to use trees of sizes 1 000, 10 000, 100 000, 1 000 000 and depths 30, 100, 3 000, 10 000, respectively. For each size/depth combination there are 10 different trees generated from different random seeds. The number of trees could be increased later on, if the experiment execution times allows for it.

Two types of queries are run for each tree: finding neighbors and finding lowest common ancestors. The find neighbors query takes a vertex and finds all its neighbors exactly at given depth. The find lowest common ancestors query takes two vertices and finds their lowest common ancestors or terminates upon reaching a given depth. Both queries use depths 5, 10, 15, 20, 25.

Due to some queries being very short they are executed and measured together in sets. Only the whole execution time of the set queries and nothing else is captured. The aim is to have the execution time of the set to be at least 20 ms, preferably around 100. Of course the same sets are used when measuring mysql and neo4j performance. A set still might contain only one query if it is long enough for both mysql and neo4j.

Each set contains queries only of the same type and depth, so an average execution time for the query type and depth is measured on each tree. Multiple different sets of the same depth/type will be used if the time allows for it to see how the results vary.

The whole experiment will be repeated enough times to determine how consistent the results are.

Summary

1. Start java via shell script and open a graph (tree) in mysql/neo4j.
2. Warm up.
3. Execute all query sets of that graph and record their times together with the parameters of all queries.
4. Quit and repeat step 1 with other graphs.
5. Repeat the whole experiment.

Without caching

Another similar comparison experiment is performed, except that now the database is not warmed up and only a single query is executed and measured in one java instance.

Data analysis

- Calculate the mean and confidence intervals of 95% for queries of the same type and depth on trees of the same size and depth.
- Make separate scatter plots for the query time against tree size and against the tree depth, and against the depth of the query for both databases. Try curve fitting to see if there is a correlation between them.
- Calculate the time differences (mysql time divided by neo4j time) to see how many times one outperforms the other. These differences are then plotted against the tree size to see how well they scale.