

Nearby Locations Query Performance Improvement

I'm working on Parkapps project(<http://parkapps.kent.edu/about>) which is a location base education project(using LBS). This project needs a features to query user's current location nearby locations, then push all nearby location to user's devices. The performance of query nearby locations in MySQL is not very good, sometimes when multi queries come to the server, there is heavy load for the DBSM. In Mongo DB, there are some existing solutions for location index. In this final project, I will test location query in Mongo DB system and migrate date from MySQL to Mongo DB.

In details, user will send their current location from mobile devices, then server will query database to select nearby points which already existing in the database.

Some requirements for this project:

1. Query all nearby location in location tables;
2. Order by the distance from near to far;
3. Get the distance from current location to each nearby locations;

Some methods overall:

1. Current MySQL based database is main methods in this project query statements like this:

```
SELECT id, ( 6371 * acos( cos( radians(37) ) * cos( radians( lat ) ) * cos( radians( lng ) - radians(-122) ) + sin( radians(37) ) * sin( radians( lat ) ) ) ) ) AS distance

FROM places HAVING distance < 25 ORDER BY distance LIMIT 0 , 100;
```

Obviously each query needs lots of calculation, the CPU load is very heavy.

1.1 Some optimization on SQL statements:

Above methods will calculate all points existing in circle, if we try to calculate the rectangle in green. The SQL will be simplify to:

```
SELECT * FROM places WHERE ((lat BETWEEN ? AND ?) AND (lng BETWEEN ? AND ?))
```

Although the consequence of this method is not very accurate, the performance is optimized. After index the attribute on latitude and longitude, the query speed will



be faster. However, this method will not get ordered result sets. Even sort result sets after query will cause use intermediate table then cause performance problem again.

2. GeoHash index, based MySQL.

GeoHash is a code format for geology position. MongoDB geo-index inner implementation use this method. GPS coordinate translate to a String indeed is effective method for represent locations. <http://geohash.org/> After transform the code to GeoHash, MySQL can to query by using LIKE%string% to find nearby locations. Seems good right? However, we ned to do 9 times to query all nearby points, cause the code format just can represent the adjacent points in square. Therefore this method is not for this project migration.

3. MongoDB geo location index.

MongoDB support store GeoJSON data directly in database without any transform. The most important is the MongoDB handle all the requirement for this project already!

```
db.runCommand( { geoNear: "places", near: [ 121.4905, 31.2646 ], num:100 })
```

This simple command will calculate all the nearby place. The mongoDB has some optimization on DB level, therefore we don't need to care about it.

In this final project, I will use iOS to read API on pythonanywhere and then to calculate different time using different DB.

There are 1000+ data in DB, I will set timer to count the time query and the time in network communications.

In the end I will do some comparison between those methods.