

Hybrid Concept of NoSQL and Relational Database for GIS Operation

Winhard Tampubolon
UNIBW, Institute for Applied Computer Science
Werner-Heisenberg-Weg 39, 85577
Munich, Germany
winhard.tampubolon@unibw.de

Abstract

Relational Database Management Systems (RDBMS) have a strong point at handling data with clear-defined structures. Consequently, RDBMS are generally not suitable for the data management without pre-defined schema in which the schema flexibility is a requirement to cope with data variations and inconsistencies.

The term of NoSQL (Not only SQL) databases have been raised as a potential alternative solution which is growing rapidly to provide persistence for free schema application. They combine the capability of data sharding in a non-relational structure. This type of databases can potentially accommodate variety of information from Volunteered Geographic Information (VGI) with different structures and schemas into one single geodatabase. For instance, MongoDB introduces an open source NoSQL document-based storage with a replication using data partitioning approach. This approach supports a robust specific query task by using geospatial index of simple point feature.

In specific, JSON format in MongoDB is used as a data model on a basis of a document-based rather than on a tuple basis. This situation provides lightweight query and data transfer to support efficient operation especially for Geographical Information System (GIS). However, GIS is a mature system that is already extensively developed using RDBMS approach on the production lines.

Therefore in this paper, we propose a hybrid concept between NoSQL and relational database to perform complete necessary task i.e. Create, Read, Update, and Delete (CRUD) operations by using GIS desktop platform. It shows a reliable performance for handling big data from Open Street Map (OSM) datasets.

Keywords: NoSQL, relational, document-based, GIS desktop

1 Introduction

Most of the VGI applications are supporting Web 2.0 platform since they use Java Script Object Notion (JSON) as their data model [1]. Obviously, the performance of NoSQL database with JSON capability for parsing, storage and query purposes has some advantages compared to the relational database [2].

In general, many database systems provide a desktop application that localizes data processing in order to speed up the big data operation [3]. By using GIS desktop as a basic processing platform, this paper discusses the combination between relational and NoSQL database approach in order to provide more efficient GIS operation.

Main expected result from this combination is the hybrid GIS operation platform. Therefore, the main motivation of this paper is to incorporate the emerging NoSQL database approach into a GIS desktop that mostly uses relational database approach. In addition, it exposes also the NoSQL database capabilities as a reliable alternative solution for GIS applications.

2 NoSQL vs Relational database

Unlike traditional relational database that uses many tables with rows and columns, the NoSQL database stores information in a non-adhered format. This format can enable actual query patterns as simple as possible.

NoSQL database, such as document-based stores and key-value stores, emerged to establish the gradual development methodologies through on-going life cycle of agile technologies. This approach is contrary to the relational

methods of data models using Entity Relationship (ER) diagram, normalization and conceptual design frameworks [4].

3 Hybrid database approach

Hybrid database concept in this paper means that our proposed system uses both of RDBMS and NoSQL database in GIS operation. The RDBMS has been used in the GIS Desktop while the NoSQL database has been used for data storage and transfer on a basis of JSON document.

3.1 GIS desktop

We consider the critical role of RDBMS in GIS Desktop as a mandatory rule to ensure data consistency. Our ArcGIS Python Add-In toolbar consists of four components, connector, selector, editor, and committer, and uses PyMongo library as its key elements (Figure 1).



Figure 1: ArcGIS Python Add-In toolbar.

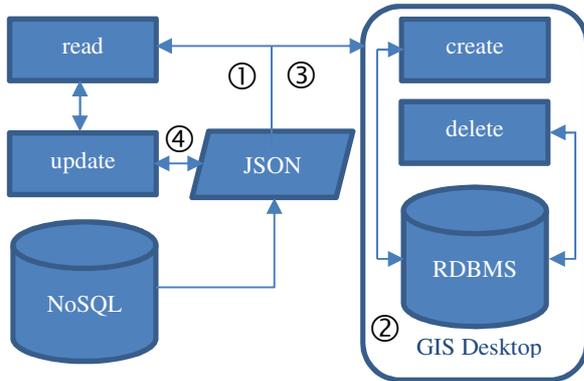
3.2 CRUD Schema

In this paper, we created a CRUD schema in a python programming language. This schema utilizes three major components namely MongoDB 3.0.9, ArcGIS Desktop 10.3 and JSON format.

The schema runs in the following steps (Figure 2):

1. Read from MongoDB via JSON transfer.
2. Create and Delete features in in_memory database.
3. Read from in_memory database via JSON transfer.
4. Update to MongoDB via JSON transfer.

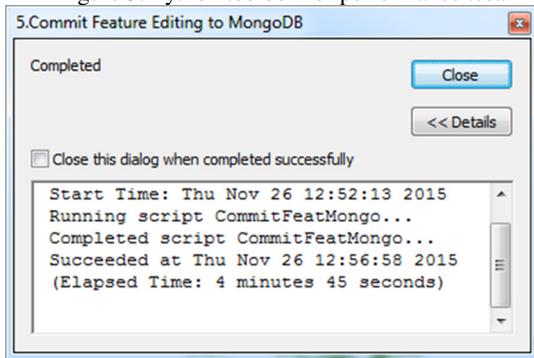
Figure 2: CRUD Schema.



3.3 Performance evaluation

The performance test is done by measuring the time needed for one cycle of CRUD operation (Table 1). We use consumer grade notebook with Intel Core Duo 2.8 Gigahertz processor, 4 Gigabyte RAM.

Figure 3: Python toolbox for performance test.



In ArcGIS, the corresponding elapsed time for each process can be informed in Python Toolbox (Figure 3). The counts of involved features are 35,000 (points), 186,866 (lines), 55,000 (polygons) from OSM datasets of Bavaria, Germany.

As explained in 3.2, the CRUD schema is applied to all features. The major difference for our hybrid approach is in the Update process, in which we use JSON-based transfer rather than table-based as implemented in RDBMS.

Table 1: Performance Test (in second).

Feature Type	RDBMS	Hybrid	
		No Index	2D Index
Point	335	245	135
Line	548	362	285
Polygon	756	453	376

4 Conclusions and Further Work

We have established the hybrid approach of NoSQL and RDBMS for GIS operation using ArcGIS Desktop. Our results show that the hybrid database approach with 2D Index presents effective CRUD performance compared to RDBMS. In the future, this hybrid database approach can be extended for cartographic and visualization purposes in GIS.

References

- [1] Z. Liu, B. Hammerschmidt and D. McMahon. JSON Data Management-Supporting Schema-less Development in RDBMS. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2014
- [2] X. Zhang, W. Song and L. Liu. An implementation approach to store GIS spatial data on NoSQL database. In International Conference on Geoinformatics (GeoInformatics) 22nd, pp.1-5, Kaohsiung, 2014
- [3] A. Eldawy, M.F. Mokbel. The Era of Big Spatial Data: Challenges and Opportunities. In International Conference on Mobile Data Management 16th, 2015
- [4] ENTERPRISEDB, Using the NoSQL Capabilities in Postgres, Whitepaper, (online last accessed on 06.04.2016 http://info.enterprisedb.com/rs/enterprisedb/images/EDB_White_Paper_Using_the_NoSQL_Features_in_Postgres.pdf), 2015