COMBINATION OF NON HADOOP-BIGDATA & RELATIONAL DB

Fahmi Amhar [1)], Winhard Tampubolon [2)], Florence Silalahi [3)]

[1) 3)] Center for Research, Promotion &Cooperation
[2)] Center for Standardization and Institution of Geospatial Information

Geospatial Information Authority of Indonesia
Jl. Jakarta-Bogor Km. 46 Cibinong 16911, Indonesia
Email: famhar@yahoo.com; winhardt@gmail.com; florenceelfriede@gmail.com;

ABSTRACT

A Relational Database Management System (RDBMS) has the major strength in handling data with a fix schema through relational tabular structure. Consequently, RDBMS has no ability to accommodate irregularities of data structure and schema in such way social media or big data implements. Therefore, it is generally not suitable for an application without a predefined structure where flexible schema is a requirement to cope with many ways of data collection and maintenance. Therefore, NoSQL (Not only SQL) databases have been raised as a trend offering potential alternative solution for a provision of free schema applications. On the other hand, Hadoop is actually rather an environment to enable parallel computation on a large scale, servers and coverages. In addition, Hadoop doesn't always fit with both RDBMS and NoSQL approaches. As an example of Non-Hadoop big data structure, some parts of NoSQL approach can be best fit to the aforementioned purposes. This type of database has the potential to manage various information from Volunteered Geographic Information (VGI) with different structures and schemes into a single data processing platform. Hence, MongoDB uses open source NoSQL document-based storage with replication functionalities using geospatial data partition and index. This approach enables robust task-specific queries using a simple point feature geospatial index. In this paper, a hybrid concept between NoSQL and RDBMS implementation is assembled in order to provide necessary tasks, namely Create, Read, Update, and Delete (CRUD) operations. To test the concept and get some results, GIS desktop is used as a test bed. Finally, it shows reliable performance for handling big data from data sets such as Open Street Map (OSM) and Indonesian Official Topographic Data (RBI).

KEYWORDS: NoSQL, relational, geospatial index, document-based, GIS.

INTRODUCTION

A Relational Database Management System (RDBMS) has the prominent advantage of handling data with a clear structure. When the data is grown in huge, Hadoop database is the solution for Bigdata of RDBMS data which could be store in many data servers [1]. Topographical Data (RBI) from the Indonesian Geospatial Information Authority is no exception. In other hand this solution could not yet accommodate the free format data which is not suit in RDBMS schema.

RDBMS are generally not suitable for data management without a predefined schema where schema flexibility is a requirement to deal with data variations and inconsistencies [2]. Meanwhile NoSQL (Not only SQL) databases have been raised as a fast growing potential alternative solution to free schema applications [3]. NoSQL is a good example of the Non-Hadoop Bigdata structure. This type of database has the potential to accommodate various spatial information from Volunteered Geographic Information (VGI) with different structures and schemes into a single geodatabase.

VGI applications, which based on Open Street Map use Java Script Object Notion (JSON) as their data model [5]. NoSQL database performance with the ability to parse, store and query has several advantages over RDBMS [6]. Meanwhile, many database systems provide desktop applications to localize data processing to speed up big data operations [7]. With desktop GIS as the basic processing platform, this paper discusses the combination of RDBMS and NoSQL approaches to provide more efficient GIS operations.

The main outcome expected from this combination is a hybrid GIS operating platform. Therefore, the objective of this paper is to integrate the NoSQL database into a GIS desktop which mostly uses RDBMS. Apart from that, he also described the capabilities of NoSQL databases as a reliable alternative solution for future GIS applications.

MATERIAL & METHODS

Unlike traditional RDBMS which have multiple tables with rows and columns, NoSQL databases store information in a non-standard format. This format later enables the actual query pattern to be as simple as possible. NoSQL databases, like key-value and document-based storage, appear to establish a continuous incremental development methodology. This is different from the relational method which uses Entity Relationship diagrams, normalization and a conceptual design framework [8].

The hybrid database concept proposed in this paper uses RDBMS and NoSQL databases in GIS operations. RDBMS has been used in GIS Desktop whereas NoSQL database is used for data storage and transfer based on JSON documents.

Even though there are dozens of NoSQL databases, it was found that most of them still had a deep niche for their deployment. Others try to keep developing and running as Hadoop.

Three NoSQL softwares primarily with general purpose capability and low cost include:

- MongoDB. MongoDB is popular because of how easy it is for developers to build projects on top of it. It features an ad-hoc request model that allows it to serve a variety of data processing and analytical use cases.
- MarkLogic. MarkLogic is popular with companies that require strong operational behavior and query capabilities. The database supports nearly all data types (XML, JSON) and has broad support for ad hoc analytics.
- Couchbase. Couchbase is a lesser-known MongoDB competitor with a superior query system and an ideal architecture for multiple uses for operations and analytics. The new database supports JSON documents.

For example, MongoDB introduced open source NoSQL document-based storage with replication using a data partitioning approach. This approach supports robust task-specific queries using a simple point feature geospatial index. In this paper, we propose a hybrid concept between NoSQL and a relational database to complete the necessary tasks, namely Create, Read, Update, and Delete (CRUD) operations using a GIS desktop platform.

Even though the concept of traditional RDBMS is still applied as a preferable solution for a more consistent and persistent approach to implement geospatial data, NoSQL database can provide comparable approach by using other items. As included in Table 1, an overview of the most recent definitions i.e. Indonesian Geographic Feature Catalogue (KUGI) example available on the RDBMS compared to the NoSQL platform i.e. MongoDB is presented. For example, data representation including their main technical specifications as stated in the database schemas has been introduced in atomic way of embedded document. It shows clearly how NoSQL can be very flexible to represent each feature independently without following the table, row, column structure as undertaken in RDBMS. Therefore, the object-oriented approach can be more easily implemented as the key feature to manage each attribute on a document-based data structure.

Table 1: Implementation for KUGI data structure.

RDBMS		MongoDB	
Definition	Example	Definition	Example
Database	Category	Database	Category
Table	Sub-category	Collection	Flexible
Row	Feature Code	Document	Flexible
Column	Attribute	Field	Flexible
Index	Geospatial (2D)	Index	Geospatial (2D)
Join/Relate	Table-based	Embedded Document	Document-based unit

Source: www.mongodb.com/collateral/rdbms-mongodb-migration-guide [9]

In this paper, a CRUD scheme is set up with Python programming language. This scheme uses three components: ArcGIS Desktop 10.3, MongoDB 3.0.9 and the JSON format [10]. A MongoDB toolbar is created in ArcGIS Desktop (Figure 1). A consumer grade notebook with Intel Core Duo 2.8 Gigahertz processor, 4 GB of RAM is used.

Fig. 1 MongoDB toolbar

The schematic runs in the following steps (Figure 2):
1. Read data from MongoDB to ArcGIS via JSON transfer.
2. Create and Delete features in the in_memory database of ArcGIS.
3. Reading data from in_memory database via JSON transfer.
4. Update to MongoDB via JSON transfer.

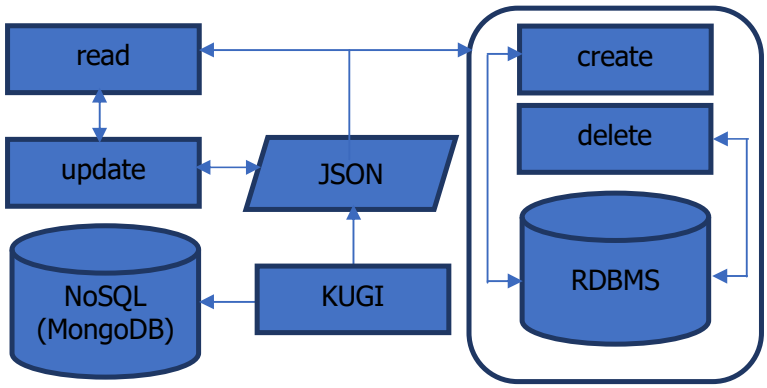

Fig. 2 CRUD-Schema

The data sample was taken from an OSM data. As is well known, the VGI PetaKita application from the Geospatial Information Authority is also an application built on top of OSM [Figure 3]. But if needed, the underlayed map could be switched to the standard topographic data (RBI) [Figure 4]..

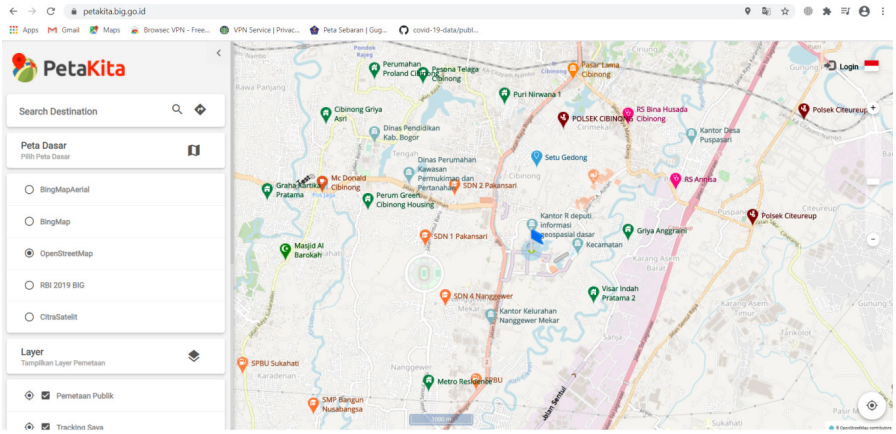

Fig. 3. PetaKita with background OpenstreetMap

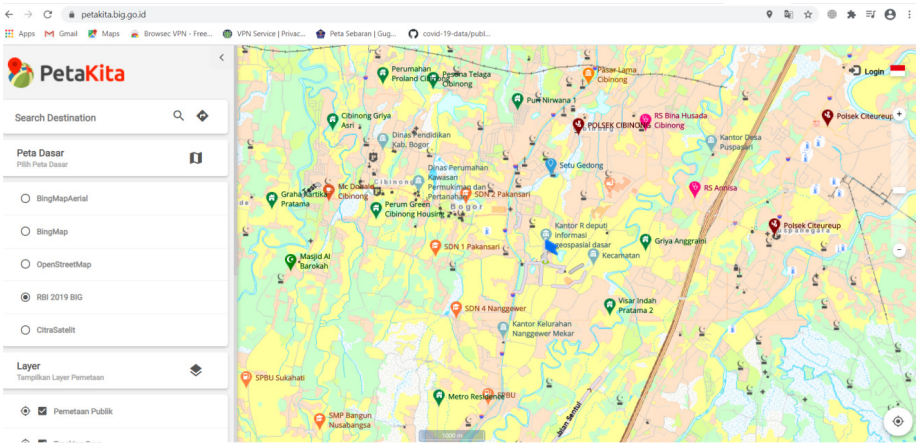

Fig. 4. PetaKita with background standard topographic map (RBI)

RESULTS & DISCUSSION

The critical role of RDBMS is a mandatory rule to ensure data consistency. In this experiment, ArcGIS Python Add-In toolbar consists of four components, i.e. connector (to Create), selector (to Read), editor (to Update and Delete), and committer (to Store), and uses PyMongo library as its key elements.

In ArcGIS, the corresponding elapsed time for each process can be informed in Python Toolbox. The number of features are about 35,000 points, more than 185,000 lines, and 55,000 polygons. The performance test is done by measuring the time needed for one cycle of CRUD operation (Table 2).

Table 2: Performance Test (in second).

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

It appears here that the performance of the hybrid model (RDBMS + NoSQL) is still better than pure RDBMS. As explained in the previous chapter, the CRUD schema is applied to all

features. The difference for our approach is in the Update process, in which JSON-based transfer are used rather than table-based as applied in RDBMS.

While not applicable to all situations, it has also been found that NoSQL as an alternative to Hadoop can save time, money and reduce risk. The following Table 3 shows a comparison between Hadoop and NoSQL platform.

Table 3: Comparison Hadoop to NoSQL.

Hadoop	NoSQL
Hadoop is interesting in bigdata projects because it is open source and can be scalable to hundreds of nodes with built-in redundancy. The general analysis and detail technical settings can be adopted for bigdata environments.	The NoSQL databases combine the positive traits of Hadoop with easy development and operational use. They are called NoSQL for their capability to store and process geospatial data, including unstructured data.
Hadoop may still be the only solution that fulfills all of these characteristics for the desired bigdata. However, there are different classes of solutions that also meet most requirements. And unlike Hadoop, this solution requires a bit of domain knowledge in order to keep the infrastructure running without much expense. Some developers have been seen building projects faster and with higher success rates.	This capability makes NoSQL more flexible, so it can store rich data without having to structure tables. This can help the database go further than traditional relational scales. On the other hand, NoSQL supports constructs that are similar to classic relational databases, such as: classic user/password authentication to provide security, collections to organize data, and indexes to improve performance.

However, NoSQL is not a panacea for everything. For pure analytical applications, Hadoop is often more suitable. For example, data server giant like Facebook uses Hadoop to support analytics across a petabyte-sized dataset. Meanwhile NoSQL is not designed for pure analysis and does not offer the same scalability and performance as Hadoop. But there are a number of use cases where NoSQL is more powerful [11].

Among others are:

- Data Hub: If we need to consolidate information from multiple online data sources, the NoSQL database is a perfect fit. For example, creating a single customer data view or linking together data from many different parts of a large organization. A national mapping agency like BIG, which has to be the hub for various geospatial data in Ministries, Agencies and Local Governments, seems to really need this.
- IoT: The Internet of Things is full of devices and sensors that create large amounts of data that are structurally changeable and whose primary purpose is for large-scale monitoring, alerting and analytics. NoSQL databases excel at this and can handle massive volumes of data with ease. At BIG, the gravity, tide, CORS sensors are jointly used with the navigation device built into the smartphone, as well as the aerial photo sensor and LIDAR on the drone.
- Real-Time Analysis: Most NoSQL databases are capable of real-time aggregation of high-volume data streams. A lot of low-value data needs to be aggregated prior to analysis. Some of the drones used for post-disaster emergency response surveys require real time analysis when they are in the field.

Hadoop has been around for quite a while and it has become clear that the kinds of applications that make sense to invest time and money, such as large-scale batch data processing, interactive queries over petabytes of data, and artificial intelligence workflows.

But at the same time, it was found that Hadoop was not the solution for all Bigdata initiatives. NoSQL currently shares many of the characteristics with Hadoop, but in some cases, it is relatively easier to manage and develop. So, if we're going to start a big data project, it makes sense to investigate using NoSQL. Since for data hubs, IoT and real-time analytics using cases, one may be a wiser choice than the solution from Hadoop.

CONCLUSSION

A hybrid NoSQL and RDBMS approach to GIS operations using ArcGIS Desktop was created. The provisional results show that the hybrid database approach with 2D Index provides an effective CRUD performance compared to RDBMS. In the future, this hybrid database approach will be expanded to more advance GIS analysis and visualization including cartography purposes.

It is hoped that this will also demonstrate reliable performance for handling big data from data sets such as PetaKita which is based on Open Street Map (OSM) and Official Topographic Data of Indonesia (RBI).

ACKNOWLEDGMENTS

This on going research has been supported by the Geospatial Information Authority of Indonesia and the Ministry for Research & Technology (PRN National Bigdata) of Indonesia.

REFERENCES

[1] P. Judge (22 October 2012). Doug Cutting: Big Data Is No Bubble. silicon.co.uk.

[2] K. Grolinger, W.A. Higashino, A. Tiwari, M.A. Capretz (2013). Data management in cloud environments: NoSQL and NewSQL data stores. Aira, Springer.

[3] Lawrence (2014). Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB. International Conference on Computational Science and Computational Intelligence 1.

[4] P. Shah, S. Chaudhary (2018) Big Data Analytics Framework for Spatial Data. Lecture Notes in Computer Science book series (LNCS, volume 11297)

[5] Z. Liu, B. Hammerschmidt and D. McMahon (2014). JSON Data Management-Supporting Schema-less Development in RDBMS. In Proceedings of the ACM SIGMOD. International Conference on Management of Data.

[6] X. Zhang, W. Song and L. Liu (2014). An implementation approach to store GIS spatial data on NoSQL database. In International Conference on Geoinformatics (GeoInformatics) 22nd, pp.1-5, Kaohsiung.

[7] A. Eldawy, M.F. Mokbel (2015). The Era of Big Spatial Data: Challenges and Opportunities. In International Conference on Mobile Data Management 16th.

[8] ENTERPRISEDB (2015., Using the NoSQL Capabilities in Postgres, Whitepaper, http://info.enterprisedb.com/rs/enterprisedb/images/EDB_White_Paper_Using_the_NoSQL_Features_in_Postgres.pdf

[9] MongoDB (2018): RDBMS to MongoDB Migration Guide. A MongoDB White Paper. https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide

[10] W. Tampubolon (2016): Hybrid Concept of NoSQL and Relational Database for GIS Operation 19th Agile Int. Conf. Geogr. Inf. Sci. Helsinki, Finlandia.

[11] J. De Goes (2016): Have Your Cake And Eat It: Big Data Without Hadoop. https://www.forbes.com/sites/forbestechcouncil/2016/09/19/have-your-cake-and-eat-it-big-data-without-hadoop/#21f2a4001d84