NoSQL Suitability for SWE-enabled Sensing Architectures

Bernd Resch
University of Heidelberg
Berliner Strasse 40, 69120 Heidelberg
+49-6221-54-5560
bernd.resch@geog.uni-heidelberg.de

ABSTRACT
The emergence of a variety of new sensing technologies including geo-sensor networks, cameras or RFID-based systems raises the need for efficient measurement data storage and retrieval. This paper illustrates the suitability of NoSQL databases for distributed geo-sensor webs. Amongst others, a specific advantage of NoSQL databases in SWE-based networks is that complex data structures can be directly mapped into the database. Concluding, it can be stated that NoSQL is a very good fit for SWE-based geo-sensor webs, but has its limitations in terms of standardised semantics.

Categories and Subject Descriptors
H.2.8 [**Information Systems**]: Database Applications – *spatial databases and GIS.*

General Terms
Management, Measurement, Performance, Standardisation.

Keywords
NoSQL, Sensor Web Enablement, performance.

1. INTRODUCTION AND MOTIVATION
The emergence of a variety of new sensing technologies including geo-sensor networks, cameras or RFID-based systems raises the need for efficient measurement data storage and retrieval. In geo-sensor webs, a major focus is on scalability and performance of the data storage system with simultaneous multi-user access as potentially numerous read and write processes can access the data repository – especially in large sensor webs.

Particularly the emerging idea of decentralised geo-sensor webs requires optimised data storage in federated environments. This is mainly due to the fact that distributed sensor pods are mostly characterised by constrained energy and processing resources.

Decentralised data handling approaches substantially diverge from current centralised methods in large-scale databases, which are often not suitable for sensor-based real-time applications in dynamic environments. This is particularly true for Sensor Web Enablement (SWE) [1] based systems, as SWE aims to make sensors discoverable, accessible and controllable over the Web.

The First ACM SIGSPATIAL International Workshop on Sensor Web Enablement (SWE2012) 2012, November 6, 2012. Redondo Beach, CA, USA.

2. DATA STORAGE IN DISTRIBUTED GEO-SENSOR WEBS
Traditionally, database management systems (DBMS) have to fulfil seven main requirements: data security (avoid data loss and administer access rights), transactions (preserve data consistency even under concurrent multi-user access), data integrity (through defining constraints and access rules), request optimisation (translation of user requests into logical operators and relational algebra), application support (stored procedures and triggers), query languages and multi-user capabilities.

Relational database models, which are currently widely used in geo-sensor webs, are conceived to fulfil the so-called ACID properties – atomicity (transactions follow the “all or nothing” principle), consistency (write only “valid” data to the database), isolation (non-interfering transactions) and durability (prevention of data loss) [2]. However, ACID requirements can often not be fulfilled in geo-sensor webs due to a number of restrictions such as unreliable radio connections, sensor failure due to power outages, measurement errors or in-memory database management.

One solution for this problem is the approach of storing data in decentralised databases locally on the sensor pod. This substantially diverges from current centralised data storage methods in large-scale databases, which are often not suitable for sensor-based real-time applications in dynamic environments. Especially SWE conformant infrastructures need to follow this decentralised approach, as SWE aims to make sensors discoverable, accessible and controllable over the Internet. In other words, each sensor shall be accessible separately over the Internet, which requires that every sensor is accessible over a separate data service instance and that measurement data are stored directly on the sensing device.

Another important aspect, which requires data storage directly on the sensing device, is quality assurance, and error detection and correction in geo-sensor webs. Advanced quality assurance involving Complex Event Processing (CEP) requires a number of historical measurement values (for time series checks) and inter-relations between geographically distributed sensors to compute statistical algorithms involving the dimensions time, location, cause and aggregation [4].

Another issue with RDBMS is that relational data does not map well to typical programming structures, which oftentimes involve complex data types and hierarchic data. This is a specific issue with SWE-based sensing infrastructures as the hierarchic nature of XML-based communication induces a non-trivial problem: Complex objects which comprise other objects and lists inside the XML structure do not always correspond directly to a single row in a database table.

3. NOSQL SUITABILITY FOR STORING AND QUERYING SWE SENSOR DATA

NoSQL (Not Only SQL) stands for new database concepts to overcome disadvantages of traditional relational DBMS approaches. In contrast to SQL-based approaches, NoSQL databases do not store data in relational schemas using tables and complex joins between them, but in a data structure oriented manner. Thus, NoSQL databases are particularly suitable for settings where the performance and real-time characteristics are predominantly important.

As NoSQL databases are not built up in a schema-based structure, there are other data architectures, which have been created according to the specific application scenario. NoSQL database architectures comprise document-oriented databases, graph databases, key-value stores, multi-value databases, object databases, Resource Description Framework (RDF) databases, and tabular NoSQL databases.

NoSQL databases have been conceived to optimise performance and scalability. Optimised performance means fast read and write processes with concurrent database access involving large amounts of data and queries. Scalability means that performance is not decreasing exponentially with increasing amounts of data, the number of data sets and the number of concurrent accesses.

NoSQL-based systems scale particularly well horizontally (i.e. performance when adding more nodes to the sensor web), which clearly distinguishes them from SQL-based systems [3]. This fact is essential for use in distributed geo-sensor webs because data storage is also possible using numerous cheap and low-capability systems instead of one high-performance system.

A crucial feature for databases used in geo-sensor webs are spatial queries. As most NoSQL databases have not emerged in the area of geoinformation, they are not primarily designed for spatial data handling. Yet, spatial extensions exist for a number of NoSQL databases for MongoDB, CouchDB, or Apache Cassandra, including spatial indexing functionality.

To assess NoSQL performance, a number of tests have been carried through [3]. A series of 5k *inserts* has been performed by 10 parallel clients (50k inserts in total). Each dataset comprised 600 bytes. Figure 1 shows the insert times for different stages in the insert process. It is evident that NoSQL is faster by magnitudes, which is also true for query procedures. Insert times are in seconds. Detailed settings for the tests can be found in [3].

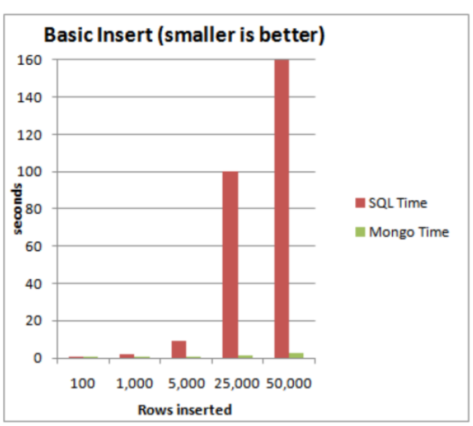

Figure 1. Required Time for 100-50k Inserts. [3]

4. DISCUSSION AND CONCLUSION

This paper sketches the usability of NoSQL databases for distributed geo-sensor webs. NoSQL (Not Only SQL) stands for new database concepts to overcome disadvantages of traditional relational DBMS approaches. In contrast to SQL-based approaches, NoSQL databases do not store data in relational schemas using tables and complex joins between them, but store data in a data structure oriented manner, which is an essential advantage in SWE-compliant measurement infrastructures. Moreover, NoSQL databases are particularly suitable for settings where performance and real-time characteristics are vital.

The performance tests briefly described in section 3 show that NoSQL databases scale very well horizontally (i.e. good performance when adding more nodes to the sensor web), which is a crucial requirement for geo-sensor webs. Horizontal scalability also allows for distributing a single measurement database on several sensor nodes. This possibility could not be validated in the presented research. Distributing the database would probably lead to reliability issues, especially in harsh environments where sensors are not always on, data connections are unreliable and sensor failure is more likely.

A specific advantage of NoSQL databases in SWE-based infrastructures is that complex objects and data structures (such as O&M-encoded measurements, features of interest [FoI] or sensor metadata) can be directly mapped into the database. This eliminates a major issue of current RDBMS-based approaches that the data stored in the database have to be transformed into SWE service-compliant data structures in a heavy-weight and computationally expensive process. Using NoSQL, large parts of O&M-encoded data can be inserted directly into the database via SOS-T (transactional) without having to extract single values out of the XML structure, minimising computational requirements.

On the other hand, issues with NoSQL databases in distributed geo-sensor webs comprise non-normalised models (through potential data object duplication) and non-standardised queries (due to non-standardised query semantics and interfaces).

Concluding, it can be stated that NoSQL is a very good fit for SWE-based geo-sensor webs, but has its limitations in terms of standardised semantics. Thus, a combined SQL-NoSQL approach could be a viable choice combining the advantages of NoSQL (high performance, horizontal scalability and direct mapping of data objects to database entries) and SQL (standardised semantics and interoperability).

5. REFERENCES

[1] Botts, M., Percivall, G., Reed, C. and Davidson, J. (Eds.) (2007) OGC Sensor Web Enablement: Overview And High Level Architecture. http://www.opengeospatial.org, OpenGIS White Paper OGC 07-165, Version 3, 28 December 2007. (17 August 2012)

[2] Chapple, M. (2012) The ACID Model. http://databases.about.com, 2012.

[3] Kennedy, M. C. (2010) MongoDB vs. SQL Server 2008 Performance Showdown. http://www.michaelckennedy.net, 29 April 2010. (14 August 2012)

[4] Resch, B., Lippautz, M. and Mittlboeck, M. (2010) Pervasive Monitoring - A Standardised Sensor Web Approach for Intelligent Sensing Infrastructures. Sensors - Special Issue "Intelligent Sensors 2010", 10(12), 2010, pp. 11440-11467.