NoSQL Database Technologies

1.0 Introduction

Since NoSQL database technologies entered the market in 1998, these technologies have challenged the widely accepted relational structures and database management practices supported by traditional SQL databases. Despite this trend, the perception that NoSQL seeks to eliminate the use of relational databases is a common misconception. NoSQL is best considered with the acronym “NOSQL” - Not Only SQL - which more accurately represents an approach that combines non-relational databases with the use of relational ones (Purdue, 2014; Penchikala, 2013).

1.1 Application cases

Social networking, big data and business intelligence applications, for instance, all pushed traditional relational databases to their limits, thus helping to spur the creation of nonrelational, horizontally scalable, distributed databases. This trend is made evident by companies that have adopted NoSQL technologies, including Facebook, Twitter, Digg, Amazon, LinkedIn and Google. Nearly fifty percent of respondents to a 2012 Couchbase survey cited the inflexibility of relational systems as the reason for adopting NoSQL systems, while 35 percent cited scalability, and 29 percent cited performance (Moniruzzaman & Hossain, 2013).

1.2 Strengths and Weaknesses

Cory, Travis, Reenu and Gary, 2013 paper was able to address some strength and weakness of NoSQL. Some of the highlighted strength are; NoSQL databases are highly scalable, reliable, have a simple data model, extremely bare query language, no mechanism for handling consistency and integrity amongst data, and almost no support for security at the database level (Okman, Gal-Oz, Gonen, Gudes, and Abramov, 2011). One of the most important advantages of NoSQL databases is that the databases can handle unstructured data. Unstructured data can be word documents, emails, audio, video, or even social network data. Also, NoSQL databases tend to scale very well on commodity hardware. Some even claim that NoSQL databases enable better performance (Leavitt, 2010), which is crucial for companies with large amounts of data. To enable faster performance, NoSQL databases typically don’t adhere to ACID (atomicity, consistency, isolation, durability) restraints that are used in relational databases. An example of NoSQL database’s performance is Facebook’s implementation (Cassandra) that is capable of handling over 100 million users continuously (Okman, Gal-Oz, Gonen, Gudes, and Abramov, 2011).

NoSQL databases are not without their faults and limitations. A common issue with these databases is lack of encryption support for data files, weak authentication between the client and the servers (and server members), lack of client communication encryption, and vulnerability to SQL injection or DOS (denial of service) attacks. Some NoSQL implementations are attempting to tackle these issues, but their solutions are not yet production ready. Cassandra provides an authenticate interface as an answer to authentication (Okman, Gal-Oz, Gonen, Gudes, and Abramov, 2011). Passwords are housed in a flat Java properties file and can be stored in plain text or as an unsalted MD5 hash. Storing passwords in plain text is a security risk for obvious reasons. MD5 hashing is not considered to be cryptographically secure and can be defeated using pre-calculated lists or rainbow tables (Okman, Gal-Oz, Gonen, Gudes, and Abramov, 2011). It is also important to note that Cassandra uses Apache's Thrift framework for client communication. As an unfortunate side effect of doing so, all passwords are transmitted in plain text. This allows an attacker the ability to use a packet sniffer to view a password in transmission. Implementations such as MongoDB only support authentication when running in a standalone or replica-set modes. Currently MongoDB's sharded mode does not support authentication.

2.0 Strategies and Challenges Associated with NoSQL.

According to article by (Michael, Mark, Joy and Cassie, 2015) who were able to address some deficiencies of the NoSQL also highlighted some strategies. Stonebraker (2010, 2012) questions NoSQL's ability to satisfy the rising need for large-scale online transaction processing (OLTP) applications, such as those demanded by online games, gambling sites and social networks. Stonebraker (2010) argues that NoSQL is not suited for OLTP due to the need for ACID guarantees to ensure data accuracy and integrity. The lack of native ACID guarantees within the NoSQL database places greater demand on the application layer and makes querying difficult. As an alternative to NoSQL, Stonebraker (2012) suggests that NewSQL systems are appropriate for responding to the needs of OLTP due to NewSQL's ability to preserve SQL ACID guarantees while continuing to offer high performance and scalability.

Another study investigated avenues to improve NoSQL's performance and address the relative immaturity of NoSQL scalability features (Thomson, et al., 2014). "Calvin" is a "transaction scheduling and data replication layer" that allows clusters of low-cost commodity machines to preserve their scalability while also maintaining transactional guarantees, even at strong consistency levels. Calvin provides a layer of ACID-compliant, transactional support on top of a NoSQL database.

A study by Ghosh (2010) encourages a multi-paradigm approach to database design. Ghosh recognizes that there are a large variety of NoSQL technologies, with varying strengths and weaknesses, and advocates using an effective combination of technologies to get the best of their strengths while offsetting their weaknesses. Ghosh further encourages the use of relational databases to support NoSQL systems while providing additional functions, including report generation and audit support. As a means for unifying the different technologies, Ghost (2010) recommends a system of asynchronous messaging, which would achieve eventual consistency between the technologies. However, the author recognizes that there are barriers to multi-paradigm design, including the rapidly evolving nature of NoSQL technologies, which could cause compatibility issues, and difficulties that could arise in syncing NoSQL and relational databases if the relational database enforces ACID transaction standards with which NoSQL is unable to comply.

3.0 Problem areas where NoSQL has been successfully utilised.

NoSQL systems tend to be adopted by companies after traditional SQL systems fall short of their requirements for performance, flexibility, or scalability (Stonebraker, 2010, Moniruzzaman & Hossain, 2013). Social networking, big data and business intelligence applications, for instance, all pushed traditional relational databases to their limits, thus helping to spur the creation of nonrelational, horizontally scalable, distributed databases. This trend is made evident by companies that have adopted NoSQL technologies, including Facebook, Twitter, Digg, Amazon, LinkedIn, and Google. Nearly fifty percent of respondents to a 2012 Couchbase survey cited the inflexibility of relational systems as the reason for adopting NoSQL systems, while 35 percent cited scalability, and 29 percent cited performance (Moniruzzaman & Hossain, 2013).

Business intelligence, traditionally based on relational online analytical processing (OLAP) databases, provides a case for the move away from relational systems. Muntean and Surcel (2013) suggest a transition from the traditional, relational-based business intelligence model to a new paradigm of agile business intelligence which, like NoSQL, will be able to utilize semi-structured data in addition to structured data. In addition, this new paradigm calls for the adoption of in memory technology (rather than disk-based technology) in order to drastically increase processing speeds (Muntean & Surcel, 2013)

Some other problem areas where NoSQL has been successfully utilized highlighted by (Barooah, 2022) are.

- Detecting Fraud in ecommerce - algorithm rules, client information / location, transaction information, last device usage etc.
- Digital data, Mobile data and Communication
- Analyses of huge unstructured data for social tracking and trends.
- Internet of Things. (IOT)
- Ecommerce, hotels, travel - Catalogue availability
- Real-Time Big Data Accessibility and management. - NoSQL can be deployed at Frontend that can store and serve operational data from multi source.
- NoSQL can also be deployed at the Backend to aggregate, store and create analysis and results

Referencing List

Madison, Michael; Barnhill, Mark; Napier, Cassie; and Godin, Joy (2015) "NoSQL Database Technologies," Journal of International Technology and Information Management: Vol. 24 : Iss. 1 , Article 1. Available at: https://scholarworks.lib.csusb.edu/jitim/vol24/iss1/1

Penchikala, S. (2013, July 23). NoSQL database adoption trends. Retrieved from Infoq.com: http://www.infoq.com/research/nosql-databases

Perdue, T. (2014, October 14). An overview of NoSQL. Retrieved from newtech.about.com: http://newtech.about.com/od/databasemanagement/a/Nosql.htm

Moniruzzaman, A. M., & Hossain, S. A. (2013). NoSQL database: new era of databases for big data analytics – classification, characteristics and comparison.

Stonebraker, M. (2012). New opportunities for new SQL. Communications of the ACM, 55(11), 10-11. doi:10.1145/2366316.2366319

Thomas, A., Diamond, T., Shu-Chun, W., Kun, R., Philip, S., & Abadi, D. J. (2014). Fast distributed transactions and strongly consistent replication for OLTP database systems. ACM Transactions on Database Systems, 39(2), 11:1. doi:10.1145/2556685

Ghosh, D. (2010). Multiparadigm data storage for enterprise applications. IEEE Software, 27(5), 57. doi:10.1109/MS.2010.87

Nance, Cory; Losser, Travis; Iype, Reenu; and Harmon, Gary, "NOSQL VS RDBMS - WHY THERE IS ROOM FOR BOTH" (2013). SAIS 2013Proceedings. 27. http://aisel.aisnet.org/sais2013/27

Moniruzzaman, A. M., & Hossain, S. A. (2013). NoSQL database: new era of databases for big data analytics – classification, characteristics and comparison.

Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., and Abramov, J. (2011) Security issues in NoSQL databases trust, security and privacy in computing and communications (TrustCom), IEEE 10th International Conference, 541-547, 16- 18.

Leavitt, N. (2010) Will NoSQL databases live up to their promise? Computer, 43, 2, 12–14.

Barooah, R., 2022. *Applications That Work Best With NoSQL Database*. [online] Clariontech.com. Available at: <https://www.clariontech.com/blog/applications-that-work-best-with-nosql-database> [Accessed 12 April 2022].