

NoSQL MongoDB vs. SQL

Department of Computer Science, University of Pretoria

September 5, 2016

Grobler, Arno	Harvey, Matthias	Lochner, Amy
14011396	14027021	14038600
Madigoe, Pricilla	Maree, Armand	Obo, Diana
13049128	12017800	13134885

Overview

- 1 Introduction
- 2 Database Problem Area
- 3 Advantages
- 4 Disadvantages
- 5 Relevance to COS326 and the business industry
- 6 Conclusion

Introduction

- Misuse of database systems in industries
- Studied research paper: Comparing NoSQL MongoDB to an SQL DB. Parker, Z. Poe, S. Vrbsky, S.V. (2013)
- Purpose: To find where NoSQL and SQL databases can be used the most effectively.

Database Problem Area

- Most common database implementation is **relational model**.
- Not effective for **large** and **unstructured data**.
- Solution for large database could be to use **NoSQL**.
- However industry still needs **modest sized databases**.
- Can NoSQL be used then for modest sized databases?

Advantages

MongoDB has better run-time performance for updates, deletes and selects over all where as SQL has better run-time performance for inserts overall.

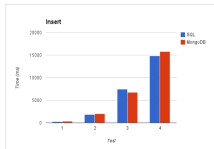


Figure 3. Insert Speed

Figure 3 shows the difference in insert speed between the two database systems. Neither database performs consistently better

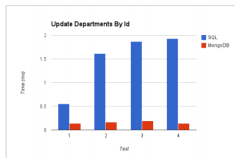


Figure 5. Update – Department ID

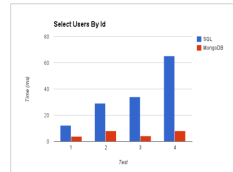


Figure 9. Select – User ID

Disadvantages

- MongoDB performs poorly for aggregate functions and querying based on non-key values
- SQL DB requires additional joins in more complex schemas

Relevance to COS326 and the business industry

- Knowing the run-time performance on each databases helps one learn their various effective and efficient usage in real-world applications
- Industry has evolved to require vast amounts of data
- This data is usually unstructured since it is not always known in hindsight
- Knowing about these database systems could help one choose the best database for large volumes of data in terms of scalability and efficiency

Conclusion

- The choice of NoSQL vs SQL is largely project dependent
- Projects with dynamic schemas are generally better off using NoSQL because of its flexibility
- The use of aggregate functions and querying based on non-key values is better suited to SQL
- When high scalability is required, NoSQL is better suited because of its suitability to distributed computing