Impact of read and write operations on NoSQL schema design: First insights

André Conrad^{1,*}, Uta Störl¹

¹University of Hagen, Universitätsstraße 47, 58097 Hagen, Germany

Abstract

Document stores like MongoDB are among the most popular NoSQL data stores. Due to the nested document structure and possible redundancy, a large amount of alternative modeling possibilities arise in contrast to relational data stores, even for simple applications. Therefore, designing an optimal schema is a complex task. In [1] we presented a visionary approach to automatic schema migration and optimization for migrating relational data stores to various NoSQL stores. This is essentially based on an in-depth data and workload analysis. A big challenge, however, is to find the optimal compromise between read and write optimization. In this paper, we show the impact of different workload profiles (mix of read and write operations) on the schema design using initial measurements. This illustrates the fundamental need to consider realistic workload profiles in the schema design and optimization process.

Keywords

Document Store, NoSQL Schema Design, Database Benchmark, Optimization

1. Introduction

NoSQL data stores such as MongoDB are gaining in popularity. However, designing a suitable schema is a major challenge due to the large number of modeling alternatives, which is often based on a trial and error approach. Thus, there are several works that address automatic schema optimization and design, respectively [1, 2, 3, 4].

For the evaluation of automatically generated, but also manually created schemas, it is therefore absolutely necessary to use realistic workload profiles in order to be able to estimate the behavior in a productive environment.

Contribution: First measurements showing the different behavior of realistic workload profiles compared to single execution (one by one) of queries.

This paper is organized as follows: Section 2 presents work discussing the evaluation of different schema modeling and optimization approaches. Section 3 describes the environment and the results of our first measurements. Section 4 provides the conclusion as well as an outlook to our future work.

 ^{0000-0001-6681-2798 (}A. Conrad); 0000-0003-2771-142X (U. Störl)
0 000 • 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

ER2023: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, Project Exhibitions, Posters and Demos, and Doctoral Consortium, November 06–09, 2023, Lisbon, Portugal *Corresponding author.

andre.conrad@fernuni-hagen.de (A. Conrad); uta.stoerl@fernuni-hagen.de (U. Störl)

2. Related work

In the following, related work on the evaluation of schema modeling and optimization approaches is described. Since the focus of this paper is on the evaluation and the necessity of considering realistic workload profiles, only papers regarding the evaluation of NoSQL schema design methods are considered.

In [5] an *e-commerce* scenario is used. They define 6 different access operations. The workload profile has been described in detail, but it is not clearly defined if a concurrent workload with multiple threads (clients) is used. In [6] the Rice University Bidding System (*RUBiS*), a web application benchmark, was adapted from [7]. It consists of a read-only workload with 7 queries and a mixed workload with 9 additional write queries. Although the measurements were originally performed with different numbers of threads (clients), this is not discussed in detail. In [2] and [4], the *RUBiS* scenario is also used, but again without going into detail about the measurements (number of threads) and only with read queries. The authors of [3] use the *e-commerce* and *RUBiS* scenario for comparison with other work, but also without going into detail about the measurements are made with respect to the different workload profiles and number of threads (clients).

None of the works describes in detail how the measurements were made with respect to the workload. In particular, the number of concurrent threads (clients) is not discussed. Therefore, it is difficult to understand exactly what the benchmark environment looks like.

3. Experimental environment and results

The goal of this work is to show the impact of executing queries one by one compared to different workload mixes of read and write queries on multiple clients. This is done using two different schema designs. One has been optimized for read-only queries and one for write-only queries.

The schema of a prototypical application¹ to manage digital print media is used because it is a real world scenario that is not too complex but has sufficiently complex relationships to illustrate the impact on the two schema designs (see Figure 1).

The test data was generated using our test data generator [8]. Figure 1 shows the conceptual model of the application.



Figure 1: Conceptual model of the application (the optimized physical models are available here: https://zenodo.org/doi/10.5281/zenodo.10009801).

For modeling the physical schemas and defining different workload mixes, 3 read and 3 write queries are used: (R_1) Given a Collection.id, return the Collection, related Items, Metadata,

¹https://sammlungen.hebis.de/Handbuecher/

Publisher and Authors. (\mathbf{R}_2) Given an Author.id, return the Author, related Items, Publisher and Collections. (\mathbf{R}_3) Given an Item.id, return the Item related Metadata, Pages, Publisher and Authors. (\mathbf{W}_4) Given an existing Publisher.id and multiple existing Collection.ids, add one or more new Authors and a new Item with corresponding Metadata and Pages. (\mathbf{W}_5) Given multiple existing Item.ids and Collection.ids, add the Items to the Collections. (\mathbf{W}_6) Given an existing Item.id and changed Metadata.title, update the Metadata.

Due to the various relationship modeling possibilities, designing the optimal schema for a given workload in document stores such as MongoDB is a complex task.

Here, optimizations with respect to reading queries result in redundancy in the *Collection*, *Item* and *Author* documents so that all attributes needed for the queries are present in the corresponding documents. This increases the complexity of the write queries (W_4 to W_6) since redundant data must be written in several collections. To avoid inconsistencies, transactions were used for the write operations in the read-optimized schema.

For the optimization of the write queries, redundancy must be avoided, since the performance decreases the more redundant data has to be written. Here, only W_5 is realized as a transaction, since several documents are updated, which ensures a rollback in the case of an error.

All measurements were done on a single node of MongoDB version 6.0.9 with 64 GB RAM and 16 CPU cores. For the benchmark, py- $tpcc^2$, a Python implementation of the TPC-C³ benchmark was extended, which already has a MongoDB driver as part of performance measurements on multi-document transactions [9]. For the measurements regarding workload mixes different probabilities of the queries are used (from 99% read and 1% write to 1% read and 99% write). $r90_w10$, for example, means 90% read (R_1 , R_2 or R_3) and 10% write (W_4 , W_5 or W_6).

Results: First, it can be observed that for sequential execution (single thread) of queries, the read queries in the read-optimized schema and the write queries in the write-optimized schema always have the best performance (see R_3 and W_6 one_by_one in Figure 2 as an example).

However, this may be different for individual queries when running mixed workloads on multiple concurrent clients. For example, query W_6 (write) performs better in the read-optimized schema than in the write-optimized schema for mixed workloads (see Figure 2b).



Figure 2: Execution one by one (single thread) and 3 workload profiles with 32 threads (clients).

Finally, Figure 3 shows the behavior of different workload mixes on the read and write optimized schemas. Here, as expected, the correspondingly optimized schemas show better performance for a higher amount of read or write operations. For better readability the plot has been split, since the performance is much higher in the read-optimized schema for workload *r99_w1*.

²https://github.com/apavlo/py-tpcc

³https://tpc.org/tpcc/



Figure 3: Execution of different workload profiles (mixes) with 32 threads (clients).

4. Conclusion and future work

Based on initial measurements, it could be shown that when looking at the execution performance of individual queries, significant differences exist between queries executed sequentially (one by one) or, in a more realistic scenario, on several concurrent clients in a workload mix.

This shows the importance to consider realistic workload profiles (mixes) with concurrent clients when evaluating schema optimization approaches.

There are many dimensions to be taken into account when executing database benchmarks. We therefore plan to perform more extensive measurements. Our current vision is to use a MongoDB cluster with replication and sharding. We also plan to further examine important variables and their effects, such as the different number of clients, different document sizes, and different number of related documents.

Acknowledgments

This work has been funded by *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) grant #385808805.

References

- [1] A. Conrad, et al., Towards Automated Schema Optimization, in: ER'21, 2021.
- [2] V. Reniers, et al., A Workload-Driven Document Database Schema Recommender (DBSR), in: ER'20, 2020.
- [3] L. Chen, et al., A workload-driven method for designing aggregate-oriented NoSQL databases, DKE'22 (2022).
- [4] M. Hewasinghage, et al., Automated database design for document stores with multicriteria optimization, KAIS'23 (2023).
- [5] C. de Lima, et al., On proposing and evaluating a NoSQL document database logical approach, IJWIS'16 (2016).
- [6] M. J. Mior, et al., NoSE: Schema design for NoSQL applications, in: ICDE'16, 2016.
- [7] E. Cecchet, et al., Performance and scalability of EJB applications, in: OOPSLA'02, 2002.
- [8] V. Restat, et al., GouDa generation of universal data sets: improving analysis and evaluation of data preparation pipelines, in: DEEM'22, 2022.
- [9] A. Kamsky, Adapting TPC-C Benchmark to Measure Performance of Multi-Document Transactions in MongoDB, VLDB'19 (2019).