Оптимизация хранения данных испытания сложных технических изделий в документо-ориентированной базе данных

А. П. Попов[1], И. А. Шмидт[2]
Электротехнический факультет
Пермский национальный исследовательский политехнический университет
apopov2106@gmail.com[1], shmidt@msa.pstu.ac.ru[2]

***Аннотация*. Статья посвящена хранению данных испытаний в документо-ориентированной базе данных. Так как существующие решения не обеспечивают оптимального уровня производительности СУБД, автором предложен новый подход к организации модели данных, в рамках которого предполагается разделять временные ряды (основной тип хранимой информации) на несколько документов оптимального размера. Был поставлен ряд экспериментов, в ходе которых была доказана целесообразность применения данного подхода и определена зависимость оптимального размера документа от количества параметров в кадре для произвольного временного ряда. Использование оптимальной модели данных позволяет значительно увеличить производительность СУБД при выполнении операций записи и чтения, а также уменьшить размер самой базы данных. Результаты проведенного исследования будут полезны при разработке и оптимизации систем сбора и анализа данных испытаний сложных технических изделий.**

Ключевые слова: измерения; испытания сложных технических изделий; документо-ориентированная СУБД; mongoDB; оптимизация

I. Введение

В настоящее время при проведении испытаний сложных технических изделий, образуется большой поток информации. Полученные значения имеют привязку к отметке измерения и располагаются в четкой хронологической последовательности. Набор значений снимаемых в определенный момент времени называется кадром, а сама последовательность значений – временным рядом или трендом. [1].

При проведении испытания, скорость сбора данных достигает нескольких десятков кГц, а в каждый момент времени регистрируется более сотни различных параметров. Так как время проведение испытания может составлять несколько часов и даже дней, то объем полученной информации исчисляется несколькими миллионами значений и более [1, 2]. Поскольку база данных должна накапливать результаты множества испытаний различных изделий, а срок хранения данных должен соответствовать полному времени жизненного цикла изделия, то требуется применение высокопроизводительной СУБД. Задача усложняется еще и тем, что необходимо хранить не только временные ряды, но и атрибуты испытания, например, вид испытываемого изделия, дату проведения испытания и др. Для построения такой системы хранения необходимо наличие хорошо организованной модели данных, которая обеспечивала бы оптимальную скорость выполнения операций записи и считывания при минимальных затратах ресурсов.

Как известно из проведенных ранее исследований, применение реляционного подхода к организации таких систем хранения данных испытаний нерационально [3, 4]. Для этих задач рекомендуется использовать NoSQL решения, например, документо-ориентированные базы данных. Модель данных подобных хранилищ позволяет объединять множество пар ключ-значение в абстракцию, называемую «документ» [5].

В данной работе рассмотрены варианты организации структуры «документов» для хранения данных испытаний с точки зрения оптимизации производительности работы СУБД. Для реализации полученных решений используется документо-ориентированной СУБД mongoDB [5, 6].

II. Структура хранения данных в mongoDB

Для хранения документов в mongoDB используется формат BSON, двоичное представление JSON-формата. Документы могут иметь вложенную структуру и объединяются в коллекции [5, 6].

Касаемо задачи хранения данных испытаний, нам необходимо хранить как тренды (основной вид получаемой информации), так и атрибуты самого испытания. Рациональнее всего реализовать хранение атрибутов испытания в отдельной коллекции. Связь трендов с этими документами будет осуществляться по одному из ключей, например по идентификатору испытания. Данное решение позволит сократить занимаемое пространство, частично нормализовать хранимую информацию и избавиться от противоречивости [7].

Вернемся к проблеме хранения временных рядов. Существует несколько способов представления временных рядов в виде документов. Давайте рассмотрим некоторые из них.

A. Реляционный метод хранения

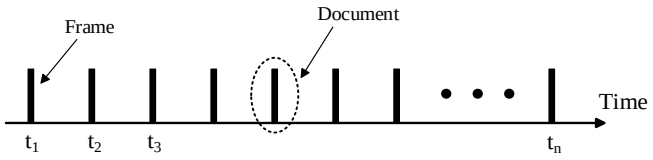

Рис. 1. Реляционный способ организации документа

Если мы применим реляционный способ организации документа, т. е. представим кадр в виде строки таблицы и реализуем этот формат в документо-ориентированной СУБД, то мы будем хранить каждое событие в отдельном документе [7]. Описание реляционного подхода представлено на рис. 1.

Хотя этот подход реализуем в mongoDB, он не использует всех возможностей документо-ориентированной модели данных [5, 6]. Напротив, он больше подходит к применению в реляционных СУБД, нежели в документо-ориентированных.

B. Документо-ориентированный метод хранения

По своей реализации данный метод диаметрально противоположен предыдущему. При таком подходе к организации данных в одном документе будут храниться все кадры принадлежащему одному временному ряду в виде массива. Документо-ориентированный подход реализует вложенную структуру документов в mongoDB [6]. Описание документо-ориентированного метода представлена на рис. 2.

Однако на практике этот метод имеет ряд особенностей, одной из которых является наличие ограничения на максимальный размер документа в mongoDB [5, 7]. При этом если размер временного ряда больше максимального размера документа, то его разбивают на несколько частей и записывают в разные документы.

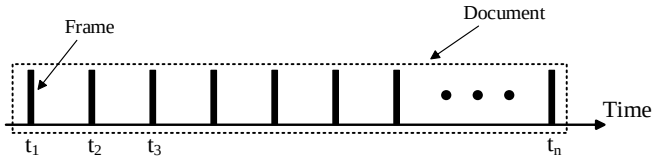

Рис. 2. Документо-ориентированный метод хранения

III. Оптимизация документо-оиентированного метода хранения

Описанный выше документо-ориентированный метод хранения обладает рядом преимуществ перед реляционным. Однако он имеет один существенный недостаток – низкую скорость считывания данных.

Так как основной задачей испытания является проведение комплексного анализа полученных данных, то необходимо обеспечить максимальную скорость считывания [2]. Чтение одного большого документа занимает много времени. Скорость считывания данных можно увеличить, уменьшив количество кадров записываемых в документ и увеличив количество самих документов, однако это может отразиться на скорости записи. В связи с этим встает вопрос о нахождении такого размера документа (количества кадров в одном документе), при котором будет обеспечиваться оптимальная скорость записи и считывания данных.

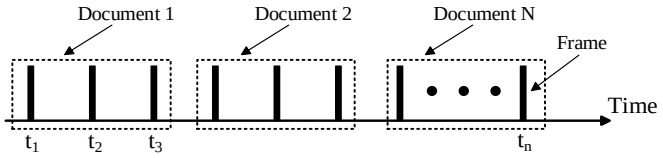

Рис. 3. Оптимальный метод хранения

При использовании данного метода временной ряд разделяется на несколько частей оптимального размера, каждая из которых храниться в отдельном документе. Описание этого оптимального метода хранения представлено на рис. 3.

Для нахождения оптимального количества кадров в документе было разработано приложение, которое позволяет записывать в базу данных документы с разным количеством кадров, а также измерять и выводить время выполнения операций над данными.

В качестве исходной информации используем несколько временных рядов одинаковой длины содержащих разное количество параметров в кадре (10, 50, 121, 300 и 500). После предварительной подготовки данных проведем серию экспериментов, в рамках которой для каждого случая определим такой размер документа, при котором обеспечивалась бы оптимальные скорость записи и считывании данных, а также размер базы данных.

Эксперименты проводились на оборудовании со следующими техническими характеристиками:

- Процессор – Intel Core i7-3630QM 2,4 GHz;
- ОЗУ – 8Gb DDR3;
- ПЗУ – HDD 1Tb;
- ОС – Windows 8.1 x64;
- Версия MongoDB – 3.4.10.

IV. Результаты экспериментов

Ниже представлены результаты эксперимента для временного ряда со 121 параметром в кадре.

На рис. 4 представлена зависимость времени выполнения операции считывания от количества кадров, хранимых в одном документе. Время, затрачиваемое на считывания данных, можно разделить на две составляющие: время доступа к документу на дисковой подсистеме и время считывания самого документа.

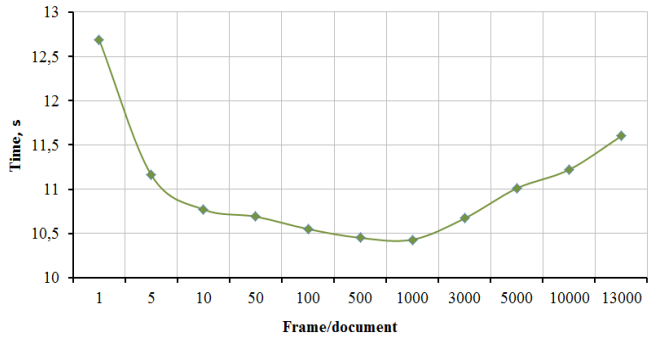

Рис. 4. Зависимость времени считывания от количества кадров в документе

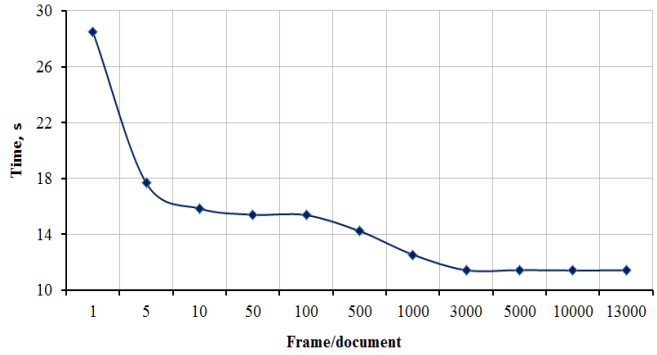

Рис. 5. Зависимость времени записи от количества кадров в документах

С одной стороны, при наличии большого количества документов увеличивается частота обращений к дисковой подсистеме, которая обладает низкой производительностью. При этом основная часть времени тратиться на доступ к документу, а не на его чтение [5].

С другой стороны, при увеличении размера документа, резко возрастает время на его чтение. Так как документ обладает вложенной структурой, то при чтении нам необходимо выполнить множество переходов внутри неё для считывания данных, что занимает достаточно много времени.

По графику видно, что минимальное время считывания данных достигается при хранении 1000 кадров в документе, т.е. при данном размере документа обеспечивается оптимальное соотношение времени доступа к данным на дисковой подсистеме и времени их считывания.

На рис. 5 представлена зависимость времени выполнения операции записи от количества кадров, хранимых в одном документе. По графику видно, что уменьшение время выполнения записи достигается увеличением количества кадров в документе. Это связано с особенностями заполнения документо-ориентированной структуры данными. Ключевой особенностью mongoDB является наличие особого механизма обновления данных: обновления требуют намного меньше времени, чем вставка [7]. При добавлении первого кадра в документ происходит его запись, а в дальнейшем, при добавлении следующих кадров, происходит обновление этого документа. Оптимальное соотношение операций вставки и обновления достигается при размере документа в 3000 кадров. Дальнейшее увеличение количества кадров в документе не приводит к повышению производительности.

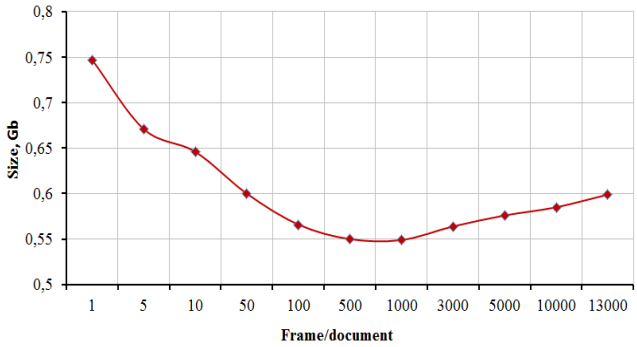

Рис. 6. Зависимость размера БД от количества кадров в документе

На Рис. 6 представлена зависимость размера базы данных от количества кадров, хранимых в одном документе. В mongoDB имеется фиксированный минимальный размер документа [5, 6], при этом не имеет значения, занимают ли хранимые данные этот объем памяти или нет. На графике видно, что увеличение количества кадров в документе повышает эффективность использования памяти. Однако при хранении большого количества кадров в одном документ наблюдается увеличение размера БД из-за нерационального использования памяти вложенной структурой.

V. Оптимальный размер документа для произвольного временного ряда

Результаты проведенных экспериментов показали целесообразность использования предложенного метода хранения временных рядов в документо-ориентированной базе данных. Для обобщения полученных результатов определим оптимальный размер документа для произвольного временного ряда.

С увеличением количества параметров в кадре временного ряда уменьшается скорость выполнения операций записи и чтения. Кроме того, из-за наличия ограничения на максимальный размер документа, уменьшается количество кадров, которое может вместить один документ.

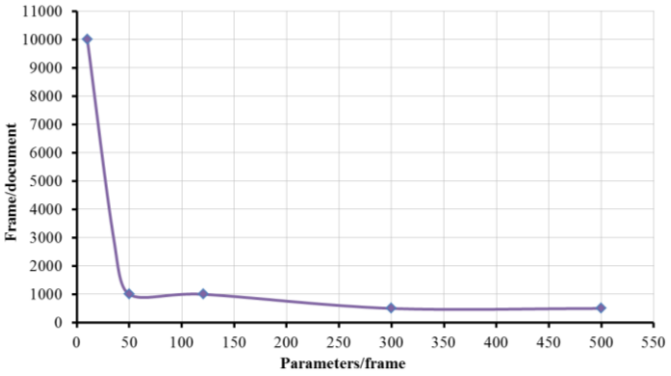

Рис. 7. Зависимость оптимального размера документа от количества параметров в кадре для произвольного временного ряда

В сфере испытаний сложных технических изделий ключевым критерием оптимальной производительности СУБД является минимизация времени выполнения операций считывания, что позволяет ускорить последующие обработку и анализ данных. В соответствии с этим критерием выберем оптимальное количество кадров в документе для каждого из исходных наборов данных на основе результатов экспериментов. После чего построим кривую отражающую зависимость оптимального размера документа от количества параметров в кадре временного ряда. Данный график представлен на рис. 7. Исходя из этой зависимости, можно приблизительно определить оптимальный размер документа для хранения произвольного временного ряда.

VI. Заключение

В настоящей работе рассмотрены вопросы хранения данных испытаний в документо-ориетированной базе данных. Для повышения уровня производительности СУБД был предложен новый подход к хранению данных испытания, в рамках которого предлагается разделять временные ряды (основной вид хранимой информации) на несколько документов оптимального размера. Для подтверждения этой гипотезы был поставлен ряд экспериментов. На рис. 8 и 9 представлены результаты проведенного эксперимента в сравнении с существующими методами хранения для временного ряда со 121 параметром в кадре.

Эксперимент показал, что использование предложенного подхода позволяет:

- сократить время записи до 125 %;
- сократить время чтения на 11–22 %;
- уменьшить размер базы данных на 10–35 %;

Кроме того, была получена зависимость оптимального размера документа от количества параметров в кадре для произвольного временного ряда. Следует отметить, что форма кривой не зависит от технических характеристик используемого оборудования, поскольку скорость выполнения операций прямо пропорционально мощности оборудования.

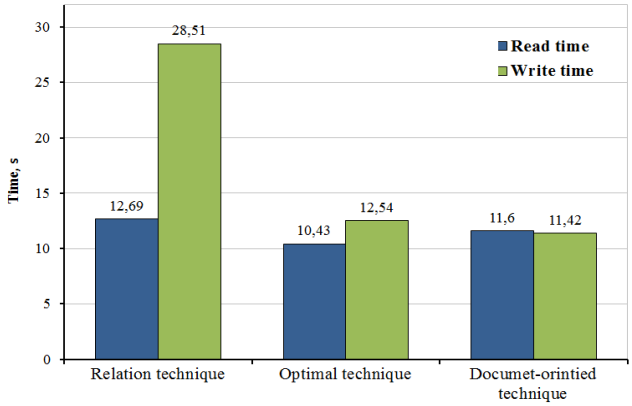

Рис. 8. Сравнения различных методов хранения исходя из времени выполнения операции

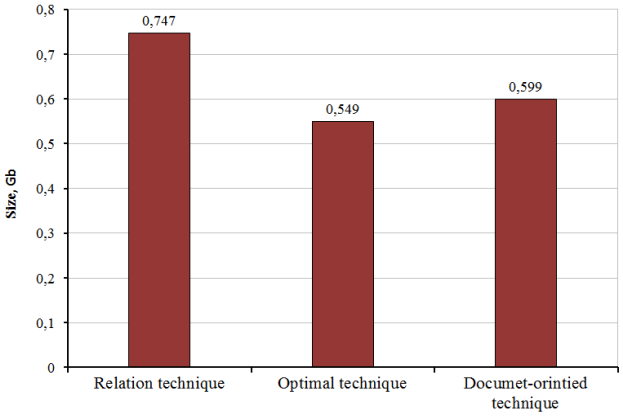

Рис. 9. Сравнения различных методов хранения исходя из размера БД

Дальнейшие исследования в этой области будут направлены на уточнение вида полученной кривой, поскольку имелось ограниченное количество данных, а также на нахождение уравнения, описывающего данную функцию.

Список литературы

[1] Попов Д.А., Шмидт И.А. Разработка функциональной структуры программного комплекса испытаний газотурбинных установок мощностью до 40 мвт // Научные исследования и инновации. 2012. Т. 6, № 1–4. С. 264-270.

[2] Шмидт И.А. Хранение результатов испытаний газотурбинных установок // В мире научных открытий. 2015. № 10.2(70) С. 1014-1026.

[3] I. Shmidt. Storing Data in The Trial of Complex Technical Products // Proceedings of the 2nd International Conference on Applied Innovations in IT. 2014. V. 1. I. 2. pp. 85-87. DOI: 10.13142/kt10002.14

[4] Шмидт И.А., Васенёв Н.В. Система регистрации параметров испытаний сложных изделий на основе документно-ориентированной базы данных // Фундаментальные исследования. 2016. № 11. ч. 3. С. 500-504.

[5] Кайл Бэнкер. MongoDB в действии. М.: ДМК Пресс, 2012. 395 с.

[6] Kristina Chodorow. Scaling MongoDB. O'Reilly Media, 2011. 66 p.

[7] Schema Design for Time Series Data in MongoDB, [Электронный ресурс]. URL: https://www.mongodb.com/blog/post/schema-design-for-time-series-data-in-mongodb. (Дата обращения: 03.04.2018).