

УДК 004.65

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ WEB-ПРИЛОЖЕНИЙ СРЕДСТВАМИ СУБД REDIS

И. П. Костенко, М. В. Ступина

Донской государственной технической университет (г. Ростов-на-Дону, Российская Федерация)

Рассмотрены реляционные и нереляционные базы данных, способы повышения производительности средствами NoSQL СУБД Redis. Проведен анализ существующих видов реляционных и нереляционных баз данных, приводятся примеры их использования. Рассматривается применение СУБД Redis и преимущества его использования.

Ключевые слова: реляционная БД, нереляционная БД, база данных, СУБД, SQL, NoSQL, Redis.

IMPROVING THE PERFORMANCE OF WEB-APPLICATIONS BY MEANS OF DBMS REDIS

I. P. Kostenko, M. V. Stupina

Don State Technical University (Rostov-on-Don, Russia Federation)

The paper considers relational and non-relational databases, the ways to improve performance using NoSQL DBMS Redis. The authors analyze the existing types of relational and non-relational databases, consider specific types, and provide examples of their use. In particular, the application of the Redis DBMS is considered, and the advantages of its use.

Keywords: relational database, non-relational database, database, DBMS, SQL, NoSQL, Redis.

Введение. В современном мире в каждом веб-приложении используются базы данных. В них хранятся не только данные, необходимые для работы приложения, но и данные пользователей. У разных пользователей хранятся разные данные, они могут меняться, удаляться или изменяться. Каждый раз при обращении пользователя к системе эти данные выгружаются и отправляются на сторону пользователя. Реляционные базы данных справляются с этой задачей, но зачастую этот процесс можно ускорить с помощью нереляционных баз данных. Для начала определим, что такое реляционные и нереляционные базы данных, в чем их различия.

Реляционные базы данных. Реляционные базы данных реализуют реляционную систему хранения данных. Данные организованы в таблицах. Столбцами таких таблиц являются типы данных и название хранимых данных, а строками являются набор связанных значений, относящихся к одному объекту. Также определенный столбец может быть отмечен ключевым, чтобы связать его с другой таблицей посредством соединения ключевого поля с так называемым внешним ключом, которым является столбец. Таким образом, данные в реляционных базах хранятся строго структурировано без возможностей изменения данной структуры [1].

Преимуществами реляционных баз данных являются [2]:

– минимальная избыточность данных при соблюдении нормальных форм, что гарантирует целостность информации при добавлении или изменении;

– механизм транзакций, обеспечивающий согласованность данных. К таким механизмам предъявляются свойства ACID: атомарность, согласованность, изолированность и надежность. Благодаря транзакциям можно выполнять параллельные запросы, не боясь за возникновение конфликтов в процессе;

– язык запросов SQL, который является стандартным для всех реляционных баз данных.

Но также такой подход имеет и свои недостатки:

- несоответствие реляционных принципов принципам объектно-ориентированного программирования, который используется во всех современных языках программирования;
- невозможность масштабирования уже созданной базы.

Нереляционные базы данных. В свою очередь нереляционные базы данных или, как их еще называют, NoSQL решают эту проблему. Выделяют несколько основных способов организации данных [3]:

- ключ-значение — данные представляют собой хэш таблицу, содержащую значение и соответствующую этому значению ключ. Подобные СУБД чаще всего используются для хранения кэша или изображений. Наиболее распространенными являются Memcached и Redis [4];

- документно-ориентированные базы — данные хранятся в иерархическом виде. Используются для хранения больших объемов данных, независимых документов без поддержания ссылочной целостности, а также используются в системах управления содержимым. Данные чаще всего представляются в формате JSON или XML. Популярными являются MongoDB и CouchDB [5];

- графовые базы данных — данные представляются в виде графов (вершин), которые объединяются ребрами (связями между ними). Чаще всего используются для социальных сетей. Популярным является Neo4j [6];

- колоночные базы данных — данные хранятся в столбцах (а не в строках, как в реляционных базах). Их преимуществом является хранение большого объема данных. Они быстро читают данные, но долго записывают их. Представителями такого рода СУБД являются HBase, Cassandra, Vertica [7].

Исходя из описания основных моделей хранения данных в NoSQL-базах можно сделать вывод, что данные в них хранятся не структурировано. Это дает преимущество в скорости обработки и выдачи информации, а также позволяет хранить файлы и изображения.

Повышение производительности средствами СУБД Redis. Во многих современных системах пользуются популярностью NoSQL-базы, хранящие ключ-значение. Чаще всего они хранят информацию в кэше, что значительно увеличивает скорость обработки запросов. Например, с такой целью используется СУБД Redis. На практике в ней хранятся самые востребованные пользователями данные, что значительно ускоряет их обработку. Но стоит отметить, что выгодно использовать Redis совместно с реляционной базой данных. В ней должны храниться все данные, а в Redis переноситься только часто используемые [8].

Популярность СУБД Redis обусловлена широким спектром вариантов использования. Как было сказано ранее, ее можно использовать для кеширования данных, что повышает производительность системы в несколько раз за счет того, что данные хранятся в оперативной памяти. Это позволяет снизить нагрузку на реляционную СУБД и перенести часто запрашиваемые данные в Redis, а также повысить скорость обработки запросов. Помимо этого, Redis располагает инструментарием для лёгкого масштабирования в зависимости от текущих нужд сервиса [9].

Также Redis зачастую используют и для хранения сессий, это очень удобно за счет хранения вида ключ-значение. Большинство современных веб-приложений стараются переносить все данные о сессиях в кеш, с чем Redis отлично справляется. За счет этого повышается скорость обработки запросов пользователей к серверу.

Помимо этого, в данной СУБД удобно хранить данные для потоковой передачи в режиме реального времени. Например, при передаче геопространственных данных, видео, аудио и изображений. Также можно хранить метаданные, истории просмотров пользователей, токены и

данные аутентификации пользователей. За счет хранения этих данных в оперативной памяти скорость работы с ними повышается в разы, по сравнению с реляционными базами данных.

Подобное решение также можно использовать для создания брокеров сообщений. Они позволяют внедрить в архитектуру web-приложения дополнительную логику, отвечающую за адресацию запросов между серверами и постановку их в очередь при большом количестве поступающих заявок. За счет скорости записи и чтения, обеспечиваемой СУБД Redis, подобные операции занимают минимальное количество времени, что позволяет оптимизировать нагрузку на сервер и повысить общую производительность системы.

Заключение. Таким образом, можно сказать, что полностью заменять реляционные базы на нереляционные не получится. Необходимо понимать преимущества и недостатки каждого вида СУБД и уметь правильно применять их там, где они не только увеличат производительность системы, а также будут полностью соответствовать выполняемой функции по остальным критериям. Лучше всего использовать их совместно, для нивелирования недостатков друг друга. В частности, необходимо понимать специфику использования СУБД Redis. Хотя она и хранит данные в кеше, что в свою очередь увеличивает скорость работы с ними, но также не стоит забывать, что размер кеша невелик и в связи с этим необходимо использовать его только для часто используемых данных. Из преимуществ, которые предоставляет redis, стоит выделить: повышение скорости обработки запросов, предоставление инструментария для разработки брокера сообщений, что позволяет хранить данные сессий, а также данные для потоковой передачи.

Библиографический список

1. Использование СУБД Redis в качестве промежуточного хранилища данных для PostgreSQL / Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/ispolzovanie-subd-redis-v-kachestve-promezhutochnogo-hranilischa-dannyh-dlya-postgresql> (дата обращения : 20.03.2022).
2. Исследование способов применения NoSQL и реляционных баз данных / Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/issledovanie-sposobov-primeneniya-nosql-i-relyatsionnyh-baz-dannyh/viewer> (дата обращения : 25.03.2022).
3. Методы повышения производительности современных веб-приложений / Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/analiz-proizvoditelnosti-redis-v-mysql-dlya-veb-keshirovaniya> (дата обращения : 20.03.2022).
4. Анализ производительности Redis в MySQL для веб-кэширования / Cyberleninka : [сайт]. — URL : <https://proglib.io/p/monolitnaya-vs-mikroservisnaya-arhitektura-2019-09-16> (дата обращения : 23.03.2022).
5. Большие данные: современные подходы к хранению и обработке / Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/bolshie-dannye-sovremennye-podhody-k-hraneniyu-i-obrabotke/viewer> (дата обращения : 23.03.2022).
6. Обзор NoSQL решений управления данными / Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/obzor-nosql-resheniy-upravleniya-dannymi/viewer> (дата обращения : 25.03.2022).
7. NoSQL — коротко о главном / Хабр : [сайт]. — URL : <https://habr.com/ru/company/oleg-bunin/blog/319052> (дата обращения : 25.03.2022).
8. Сравнительный анализ производительности SQL И NOSQL СУБД / Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-proizvoditelnosti-sql-i-nosql-subd> (дата обращения : 24.03.2022).
9. Об использовании NoSQL-хранилищ данных // Cyberleninka : [сайт]. — URL : <https://cyberleninka.ru/article/n/ob-ispolzovanii-nosql-hranilisch-dannyh> (дата обращения : 25.03.2022).



Об авторах:

Костенко Игорь Павлович, магистрант кафедры «Информационные технологии» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), igor.kostenko.1999@mail.ru

Ступина Мария Валерьевна, доцент кафедры «Информационные технологии» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат педагогических наук, masamvs@bk.ru

About the Authors:

Kostenko, Igor P., Master's degree student of the Department of Information Technology, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), igor.kostenko.1999@mail.ru

Stupina, Mariya V., Associate professor of the Department of Information Technologies, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), Cand. Sci. (Pedagog.), masamvs@bk.ru