



## Redis vs MongoDB: что выбрать?

### Описание

При разработке приложения выбранная вами база данных может повлиять на скорость получения данных, масштабируемость и общую производительность. MongoDB и Redis являются популярными вариантами современного управления базами данных. Они обе являются масштабируемыми базами данных NoSQL с открытым исходным кодом, которые предлагают весьма сопоставимые результаты по простоте и скорости. Каждая из них обладает уникальными преимуществами, которые отвечают различным потребностям разработчиков. В этой статье мы рассмотрим сходства и различия между MongoDB и Redis, чтобы помочь определить, какая из них больше подходит для вашего проекта. Принятие правильного решения сейчас может предотвратить огромную головную боль в будущем.

## MongoDB против Redis: Разные хранилища для разных нужд

Различные приложения требуют различных архитектур баз данных, поэтому MongoDB и Redis могут наилучшим образом соответствовать уникальным сценариям. Наиболее существенное различие между ними заключается в их соответствующих моделях хранения данных, которые могут влиять на скорость поиска данных, объем хранилища и риск потери данных и сбоев. MongoDB по умолчанию хранит данные на диске в виде коллекций документов Binary JSON (BSON). Такое хранение данных на диске обеспечивает больший объем хранения и

меньший риск сбоев системы. Хранение данных в виде документов BSON позволяет MongoDB обрабатывать множество типов данных, которые обычный JSON не может разобрать. Кроме того, она может хранить данные в памяти или в облаке.

Кроме того, MongoDB не имеет схемы или фиксированной структуры данных и не требует создания структур документов. Это делает ее более доступной для новичков и позволяет быстрее обрабатывать большие объемы данных с течением времени. MongoDB также предлагает превосходную масштабируемость с широким спектром горизонтальных, вертикальных и эластичных опций. Эти моменты делают базу данных легко адаптируемой к последующим изменениям, что делает ее более подходящей для стартапов и предприятий, которым необходимо учитывать потенциальное развитие и рост.

Однако если ваш проект связан с быстро меняющимися данными, Redis – лучший вариант. Универсальность Redis обусловлена хранением данных в форматах ключ/значение, которые поддерживают широкий спектр типов данных. Он также может функционировать как брокер сообщений и кэш. В отличие от MongoDB, Redis использует хранение в памяти с сохранением на диске, что позволяет ему обрабатывать поступающие данные быстрее, чем MongoDB. Кроме того, хранение данных в памяти делает его более подходящим для аналитики в реальном времени, такой как таргетинг рекламы, аналитика социальных сетей, Apache Kafka и другие потоковые решения.

## Архитектура базы данных

Архитектура базы данных определяет правила сбора, хранения, извлечения, расширения и интеграции данных в рамках конкретной системы управления базой данных. Твердое понимание архитектуры базы данных необходимо, поскольку она влияет на производительность и масштабируемость базы данных. MongoDB использует документо-ориентированную архитектуру и неструктурированный язык запросов, что означает, что ей не нужно хранить данные в строках и столбцах. Документы имеют гибкую схему и структуру, которая позволяет вставлять или удалять поля с течением времени. Она прекрасно поддерживает иерархические и вложенные данные. MongoDB также использует BSON, что означает, что хранилище данных может поддерживать многие типы файлов, обычно не поддерживаемые базами данных SQL. Но она требует много места для хранения из-за высокой скорости репликации данных. В ней также отсутствуют внешние ключи и

---

---

соединения.

Однако Redis в значительной степени опирается на хранилища ключ/значение, в которых данные хранятся в отдельной коллекции, содержащей два элемента: ключ и присвоенное ему значение. Для хранения и извлечения конкретных данных используется уникальный идентификатор (ключ). Эти ключи хранятся в словаре и требуют парсера для обработки нескольких значений (поскольку по умолчанию каждый ключ может содержать только одно значение). Redis также поддерживает несколько структур данных и значений, таких как хэши, строки, множества, списки, наборы, сортированные множества, потоки, геопространственные индексы и растровые изображения. Но их хранение ограничено доступным пространством в оперативной памяти, и хранимые данные не так масштабируемы или легко доступны с течением времени.

## MongoDB против Redis: Ключевые различия

В таблице ниже приведены основные различия и сходства между MongoDB и Redis:

Аспект	MongoDB	Redis
<b>Скорость</b>	Бессхемная структура обеспечивает высокую скорость при хранении больших объемов данных на диске.	Намного быстрее, чем MongoDB, за исключением случаев, когда большие объемы данных находятся в ее хранилище in-memory.

<b>Масштабируемость</b>	<p>Встроенный шардинг позволяет масштабировать систему на несколько географических регионов и узлов.</p> <p>Поддерживаются хэш-шаринг, шаринг диапазонов, шаринг зон и кросс-шаринг.</p> <p>Последовательное многооблачное резервное копирование легко доступно с помощью MongoDB Atlas.</p> <p>Функция мультиязычности поддерживается всеми драйверами сообщества и официальными драйверами.</p>	<p>Кластер Redis обеспечивает масштабируемость. Только хешированный шардинг. Осколки обслуживаются вручную. Отсутствие согласованности в резервном копировании шардов. Поддержка драйверов ограничена.</p>
<b>Целостность транзакционных данных</b>	<p>Поддержка многодокументных транзакций ACID с синтаксисом многодокументных транзакций.</p>	<p>Поддержка многокомандных и многозаписных транзакций. Нет поддержки откатов по умолчанию.</p>
<b>Использование памяти</b>	<p>Хранение на диске. Потребляет много памяти – около 1 ГБ на 100 000 активов.</p>	<p>Хранилище в памяти. Требуется около 4 ГБ оперативной памяти. Более высокое потребление памяти, чем у MongoDB.</p>

---

<b>Индексы</b>	Создание индексов легко и разнообразно. Советник по производительности MongoDB Atlas может рекомендовать пользователям новые индексы. Вторичные индексы можно легко использовать для создания приложений для обработки данных различными способами.	Вторичные индексы создаются нелегко и поддерживаются вручную.
<b>Высокая доступность</b>	Высокая доступность благодаря репликации.	Высокая доступность благодаря первичной и вторичной репликации на нескольких узлах и центрах обслуживания.
<b>Язык запросов</b>	MongoDB Query API запрашивает документы с помощью текстового поиска, диапазонов, одного или нескольких ключей. Он создает материализованные представления полученных данных (по запросу), выполняет геопространственные запросы и обходы графов.	Только запросы по ключу-значению. Низкая функциональность запросов, которая может быть расширена с помощью внешних модулей Redis.
<b>Постоянное хранение</b>	Функция постоянных томов в Kubernetes обеспечивает постоянство данных/хранилищ.	Постоянное хранение данных Append-Only File и постоянное хранение данных с помощью моментальных снимков.

---

**Агрегация данных**

Одноцелевая функция, функция map reduce и конвейер агрегации позволяют агрегировать данные. MongoDB Atlas использует конструктор конвейеров агрегации для построения и обработки конвейеров агрегации.

Используются функции Map reduce и конвейеры агрегации.

## MongoDB vs Redis: идеальные варианты использования

Хотя при выборе между двумя технологиями необходимо сравнивать технические характеристики, выбор лучшей системы баз данных зависит от типа приложения, которое вы разрабатываете. Если решающими факторами являются высокая скорость и низкая задержка, то Redis – лучший претендент на оптимальную производительность. Он может справиться с нагрузкой таких приложений, как обнаружение мошенничества и разработка современных игр, которым необходимо быстро и эффективно обрабатывать большие объемы постоянно меняющихся данных в режиме реального времени. Между тем, MongoDB лучше с точки зрения масштабируемости и надежности, что делает ее идеальной для приложений, которые хранят большие объемы данных в течение длительного периода времени. В качестве примера можно привести веб-сайты электронной коммерции, приложения для обмена фотографиями и программы вознаграждения сотрудников.

## Заключение

MongoDB и Redis – отличные варианты управления базами данных. Их различия в хранении, модели, архитектуре и возможностях означают, что правильный выбор зависит от конкретных потребностей вашего проекта. MongoDB стабильна, надежна и лучше всего подходит для проектов общего назначения. Однако модель хранения данных на диске делает ее относительно медленной при обработке данных в реальном времени. С другой стороны, хранилище Redis в памяти позволяет ему лучше обрабатывать огромные объемы данных в реальном времени. Однако он не так хорошо масштабируется и имеет более высокий барьер входа для начинающих

разработчиков.

**Дата Создания**

08.05.2023