



Memcached против Redis: Выбор In-Memory Cache

Описание

В проектах веб-разработки важную роль играет кэширование, при котором часто используемые ресурсы хранятся во временных базах данных в памяти (кэшах). Его цель – ускорить выполнение клиентских запросов и снизить нагрузку на сервер. Двумя наиболее популярными решениями для хранения данных in-memory в веб-разработке являются Redis и Memcached. Если сравнивать Memcached и Redis, то оба решения имеют открытый исходный код, мощные возможности кэширования и распределенную архитектуру для легкого масштабирования. Итак, давайте подробно рассмотрим Redis и Memcached, их плюсы, минусы и общие сценарии использования, чтобы вы могли выбрать подходящий вариант для своего проекта.

Memcached против Redis: В чем разница?

На первый взгляд, Memcached и Redis похожи. Их огромная популярность и открытый исходный код обеспечили широкую поддержку сообщества. Будучи одним из самых ранних решений для кэширования, Memcached получил широкое распространение во многих организациях. Redis – значительно более новое решение, но оно завоевало значительную популярность благодаря богатому набору функций.

Оба решения также предлагают клиентские библиотеки и драйверы, позволяющие легко интегрироваться с популярными языками программирования и фреймворками. Кроме того, они постоянно развиваются в соответствии с меняющимися потребностями экосистемы веб-разработки. Однако эти решения

имеют ряд заметных отличий, поэтому важно выбрать то, что лучше всего подходит для вашего проекта. Давайте рассмотрим некоторые ключевые особенности.

Что такое Memcached?

Memcached – это высокопроизводительная система кэширования объектов распределенной памяти. Ее основное назначение – кэширование часто используемых данных для снижения нагрузки на базу данных, что позволяет ускорить время отклика. Одним из основных принципов Memcached является ориентация на простоту и скорость. Она использует простую модель данных “ключ-значение”, которая позволяет легко получать доступ к данным без использования сложных языков запросов.



Что такое Redis?

Redis – это хранилище структур данных in-memory для кэширования. Его функциональность выходит за рамки традиционных систем кэширования, поддерживая различные типы данных. Также поддерживаются транзакции, которые обеспечивают последовательное и непрерывное выполнение сгруппированных команд для сохранения согласованности при модификации данных. Кроме того, Redis поддерживает обмен сообщениями по принципу “публикация/подписка”, обеспечивая высокомасштабируемую связь между

компонентами приложения.



Сравнение Memcached и Redis

Несмотря на то, что Memcached и Redis отличаются простотой использования и высокой производительностью, существуют ключевые различия, которые необходимо учитывать при выборе подходящего варианта для вашего проекта.

Структуры данных и хранение данных

Memcached – это простое хранилище ключевых значений, поддерживающее небольшие произвольные типы данных, такие как строки и объекты. Оно идеально подходит для приложений, которым требуется простое кэширование и быстрый поиск по ключу. Как следствие, оно не слишком подходит для более сложных приложений, требующих продвинутых методов манипулирования данными. В то же время Redis поддерживает более широкий спектр структур данных, включая строки, хэши, списки, множества и растровые изображения. Такая гибкость позволяет реализовать более сложные сценарии использования, например, аналитику в реальном времени или получение данных о местоположении.

Производительность и масштабируемость

Приоритетом Memcached является высокая производительность и исключительное время отклика. Кроме того, он масштабируется вертикально, позволяя добавлять в пул кэширования дополнительные серверы, чтобы удовлетворить растущий трафик и нагрузку на данные. Redis предлагает сопоставимую производительность для простых задач кэширования и дополнительные возможности для более сложных случаев использования.

Он позволяет выполнять асинхронные и неблокирующие операции ввода-вывода, что дает возможность приложениям обрабатывать большее количество одновременных задач. Эта функциональность повышает производительность при больших нагрузках. Redis также поддерживает кластеры для горизонтального масштабирования, что позволяет продолжать работу при выходе из строя некоторых узлов.

Постоянство и долговечность

Memcached работает только в памяти, то есть хранит все данные в оперативной памяти и не имеет встроенных средств сохранения. Такой подход обеспечивает максимальную производительность и низкую задержку доступа к данным. Однако Memcached не сохраняет данные автоматически в случае системного сбоя. К счастью, новые версии поддерживают восстановление данных после перезапуска и постоянную память через монтирование файловой системы DAX.

В отличие от этого, Redis предлагает опциональное сохранение данных двумя различными методами: моментальными снимками и файлом только с приложением

(AOF). Снапшотинг предполагает создание моментальных снимков данных в кэше и сохранение их на диске по истечении заданного времени. AOF – это более долговременный метод, который заключается в добавлении команд в AOF для модификации данных. В отличие от моментального снимка, этот метод позволяет Redis полностью восстановить данные в случае сбоя или перезапуска системы.

Возможности и расширяемость

Memcached ориентирован на простой и понятный подход к кэшированию. Несмотря на это, ограниченная поддержка типов данных ограничивает возможности Memcached в отношении более сложных требований к моделированию данных и выполнению специализированных операций. Такие ограничения ограничивают возможности расширения Memcached в сценариях, требующих развитых структур данных и методов манипулирования данными.

В то же время Redis предлагает множество функций, помимо базового кэширования, включая транзакции, обмен сообщениями по принципу “публикация/подписка” и сценарии на языке Lua. Транзакции позволяют автоматически выполнять несколько команд за один атомарный шаг для обеспечения согласованности данных. Обмен сообщениями Pub/sub обеспечивает взаимодействие между компонентами приложения на основе сообщений. Это позволяет реализовать обмен сообщениями в реальном времени, чаты и событийно-ориентированные архитектуры. Наконец, сценарии Lua позволяют выполнять сценарии на стороне сервера для сложных операций и преобразований данных.

Memcached против Redis в веб-разработке

Несмотря на то что Memcached и Redis имеют общие возможности кэширования, они обладают уникальными свойствами, которые нацелены на различные сценарии использования.

Примеры использования Memcached

Memcached лучше всего подходит для:

- Проектов, требующих простого кэширования для ускорения работы веб-приложений. Храня в памяти часто используемые данные, например, результаты запросов, Memcached снижает необходимость повторных запросов

к базе данных, что приводит к ускорению отклика и повышению производительности приложения.

- Высокая нагрузка на чтение и запись в базовом хранилище ключевых значений, когда необходимо простое хранилище ключевых значений и быстрый доступ к данным.
- Проекты с простыми потребностями в кэшировании и ограниченными типами данных, например, кэширование параметров конфигурации, маркеров аутентификации пользователей или статического содержимого.

Примеры использования Redis

Redis лучше всего подходит для:

- Продвинутое кэширование с поддержкой сложных структур данных, таких как списки, множества и отсортированные множества. Redis позволяет кэшировать более сложные модели данных и предоставляет специализированные операции для каждого поддерживаемого типа данных.
- Брокерская обработка сообщений с использованием технологии публикации/подписки, аналитика в реальном времени и поиск. Поддержка Redis функции обмена сообщениями “pub/sub” делает его оптимальным выбором для реализации брокеров сообщений и систем аналитики в реальном времени. Низкая задержка также делает его подходящим для аналитики в реальном времени и обработки потоковых данных. Кроме того, Redis предлагает мощные поисковые возможности благодаря таким функциям, как Redisearch, обеспечивающим эффективное индексирование и поиск данных.
- Проекты, требующие сохранения и долговечности данных. Redis предлагает дополнительные механизмы сохранения данных с помощью снапшотов и механизмов добавления файлов (AOF), которые позволяют сохранять данные на диск и восстанавливать их после сбоев и перезагрузок системы. Эти возможности делают Redis подходящим для проектов, требующих сохранения данных, таких как пользовательские сессии, транзакционные данные или кэширование критически важной информации.

Заключение

При выборе между Memcached и Redis для проекта веб-разработки не существует какого-то одного “лучшего” варианта, есть только тот, который лучше всего

подходит для уникальных потребностей вашего проекта. Понимание уникальных преимуществ и характеристик каждого решения необходимо для того, чтобы решить, что именно вам подходит.

Memcached и Redis – чрезвычайно популярные и хорошо поддерживаемые решения, каждое из которых имеет совершенно разные причины. Ориентированность Memcached на простоту и высокую производительность делает его отличным выбором для проектов с простыми потребностями в кэшировании. Однако если вам нужны более продвинутые функции кэширования и возможности, выходящие за рамки кэширования, следует рассмотреть возможность использования Redis.

Дата Создания

11.07.2023