

ОПТИМИЗАЦИЯ СКОРОСТИ РАБОТЫ ВЕБ-ПРИЛОЖЕНИЙ ASP.NET ПРИ ПОМОЩИ АСИНХРОННЫХ МЕТОДОВ ПРОГРАММИРОВАНИЯ И КЭШИРОВАНИЯ

Чинь Куок Бао (Университет ИТМО), Широков И. (Университет ИТМО)
Научный руководитель – к.педагог.н., доцент Государев И.Б. (Университет ИТМО)

Аннотация. Производительность можно рассматривать как важный фактор успешного сайта или проекта. В данной работе описываются асинхронные методы программирования и распределенное кэширование для оптимизации скорости веб-приложений, построенных на ASP.NET.

Введение. В настоящее время мы наблюдаем множество революций в цифровых технологиях, трансформирующих платформы приложений, включая дизайн веб-приложений ASP.NET. Традиционные и классические веб-технологии постепенно забываются. Однако, ASP.NET всегда является оптимальным выбором, поэтому мы используем продукт Microsoft для разработки веб-сайтов. Благодаря многим преимуществам, приложения на ASP.NET применялись во многих областях, таких как интернет-магазины, образование, туризм, бизнес и здравоохранение. Таким образом, разработка приложений на ASP.NET актуальна сейчас и будет актуальна в будущем. Скорость загрузки страниц веб-приложений ASP.NET очень важна, особенно для некоторых веб-сайтов в сфере бизнеса и продаж, когда скорость загрузки страницы низкая, что делает сайт менее привлекательным для посещения. Есть много способов оптимизировать производительность приложений ASP.NET, оптимизировать front-end, back-end, запросы к базе данных. В этой статье, мы опишем методы оптимизации back-end, а именно использование методов асинхронного программирования и распределенного кеширования.

Основная часть. В прошлом процессы часто обрабатывались в синхронном режиме. Предположим, что последовательность задач выполняется в порядке A, B, C, D. Задача A выполняется перед задачей B, задачей B выполняется перед задачей C. Выполнение задачи D, в таком случае, возможно только после завершения задач A, B и C. Такая последовательность процессов может быть прервана, если задание не было выполнено или перешло в состояние отсутствия ответа, тогда работа не будет выполнена. Асинхронное программирование исправит указанные выше недостатки. В этой модели последний процесс не должен ждать завершения предыдущей задачи, поскольку может быть выполнен одновременно с каким-либо предшествующим заданием. Общее время выполнения задач больше не будет равно сумме временных отрезков, затраченных на выполнение каждой задачи, а лишь немного больше, чем время, которое потребуется для выполнения самой долгой по времени задачи. Очевидно, что у этой модели есть преимущество во времени выполнения. Использование конструкции асинхронного программирования `async/await` максимально увеличит скорость обработки, потому что оно будет обрабатывать множество запросов одновременно, без необходимости обрабатывать их один за другим, как при синхронном программировании.

Распределенная буферизация – это метод настройки буферов данных для масштабирования нескольких серверов, хранения общих запросов и обеспечения быстрого извлечения. Кэш используется как копия или часть базы данных, каждый раз, когда приложение запрашивает данные из базы данных, оно будет смотреть в кэш, если данные были загружены ранее, оно вернет данные немедленно, не тратя ресурсы на выполнение запросов к базе данных. Следовательно, веб-приложения ASP.NET будут ускорены.

Распределенные кэши используются на веб-серверах и серверах приложений для обеспечения нелокальной памяти для лучшей избыточности, обслуживания нескольких областей и обеспечения расширяемого объема для хранения данных и пропускной способности. Данные, хранящиеся в распределенном кэше, обычно определяются тем, к чему чаще всего обращаются с данного веб-сервера или приложения. Если ранее запрошенные части данных не требуются, то недавно запрошенные данные имеют приоритет; В конечном итоге старые данные удаляются из кэша.

Выводы. В этом исследовании мы используем модель MVC в сочетании с асинхронным программированием для достижения наилучших результатов. Результаты исследования показывают, что при использовании асинхронного программирования приложения дадут результаты в 3 раза лучше, чем приложения, не использующие асинхронное программирование.

Исследование показало, что использование распределенного кэширования увеличивает скорость более чем в 10 раз, что идеально подходит для информации, в которую будут вноситься незначительные изменения. К такому у виду информации относятся: личная информация пользователя, ярлыки и редко меняющийся контент.

Чинь Куок Бао (автор)

Подпись

Широков И. (соавтор)

Подпись

Государев И.Б. (научный руководитель)

Подпись