

ИССЛЕДОВАНИЕ АСИНХРОННОЙ МОДЕЛИ ПРОГРАММИРОВАНИЯ

М.А. Романович, С.В. Лукьянец

Кафедра систем управления,

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: mjromka@gmail.com

Исследованы подходы к повышению производительности Web-приложений путём анализа показателей системы при моделировании нагрузки на сервер. На основании полученных характеристик рассмотрены и реализованы способы увеличения эффективности работы программного обеспечения за счёт внедрения асинхронности и распараллеливания при обработке клиентских запросов.

ВВЕДЕНИЕ

Современные методики управления информационными потоками в бизнесе обуславливают широкое распространение клиент-серверных программных решений. Web-приложения становятся наиболее востребованными системами массового обслуживания информационного пространства. Наряду с этим, ввиду усложнения процессов в этих системах, увеличения объёмов информации и количества пользователей, возникает необходимость оптимизации их работы с точки зрения производительности – скорости обработки запросов и отказоустойчивости. В данной работе рассмотрены методы анализа показателей производительности Web-приложений на примере ASP.NET технологии компании Microsoft. На базе проводимых исследований выработаны методы оптимизации работы алгоритмов внутри системы [1].

I. МОДЕЛИРОВАНИЕ НАГРУЗКИ НА СЕРВЕР

Чтобы определить, насколько хорошо программное обеспечение отвечает различным режимам использования, осуществляется нагрузочное тестирование с целью определения основных характеристик производительности. Нагрузочные тесты моделируют ожидаемое использование программы путем имитации одновременной работы с ней множества пользователей. Для создания нагрузочного теста необходимо записать определённую последовательность пользовательских действий, которая в дальнейшем будет использоваться в качестве блока сценария нагрузки. На следующем этапе определяется шаблон нагрузки. В контексте данного исследования целесообразно выбрать шаговый режим с параметрами, представленными таблице 1.

Таблица 1 – Профиль нагрузки

Начальное число пользователей	5
Максимальное число пользователей	200
Шаблон	Шаговый
Продолжительность шага	5
Пользователей на шаг	5

II. АНАЛИЗ ХАРАКТЕРИСТИК ПРОИЗВОДИТЕЛЬНОСТИ МОДУЛЕЙ СИСТЕМЫ

Просмотр счётчиков системы в ходе выполнения моделирования осуществляется на мониторе производительности сервера. В данной работе для исследования выбраны следующие характеристики: загрузка процессора; запросов/с; запросов в очереди; число транзакций с БД [2].

На рис. 1 приведены показатели работы сервера при моделировании по сценариям, описанным в предыдущем разделе.

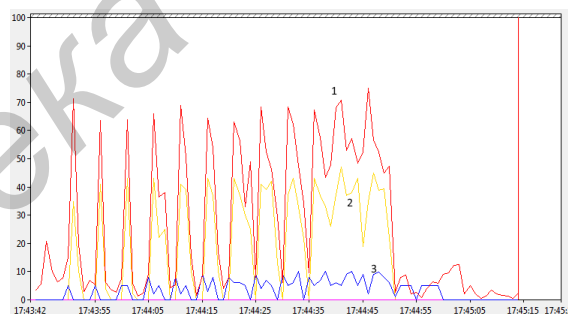


Рис. 1 – Показания монитора производительности сервера при синхронном выполнении процессов: 1 – загрузка процессора; 2 – загрузка СУБД; 3 – скорость обработки запросов (запрос/с)

Числовые и графические результаты свидетельствуют о неполном использовании ресурсов сервера (график загрузки процессора). Основываясь на этой информации, делаем вывод, что основным узким местом в процессе являются длительные операции со внешними ресурсами, во время которых сервер приложения простаивает [3].

III. РЕАЛИЗАЦИЯ АСИНХРОННЫХ ПРОЦЕССОВ

ASP.NET поддерживает методики асинхронного программирования, которые позволяют помещать длительную задачу в поток, который действует отдельно от основного потока приложения. Таким образом, пользователь получит ответ от сервера не дожидаясь окончания операции, запущенной в отдельном потоке. Тем самым уменьшится среднее время отклика

за счёт распараллеливания процессов [4]. Для реализации данной идеи создан контроллер асинхронных методов, отвечающий за выполнения действия в фоновом процессе с последующей обработкой результатов: журналирование, выставление флагов. Схема последовательности действий имеет следующий вид (рис. 2).



Рис. 2 – Асинхронная последовательность действий процесса

Идея контроллера заключается в том, что последовательность событий, которая должна быть выполнена асинхронно, описывается в делегате (C#, delegate). При этом пост-обработка результатов выполнения делегата выделена в метод-обработчик результата. Реализовав асинхронную модель для процесса создания объекта, проведём ряд тестов производительности для сравнения с результатами синхронной модели. При моделировании асинхронных процессов будем использовать такие же параметры нагрузки. Графики исследуемых метрик приведены на рис. 3.

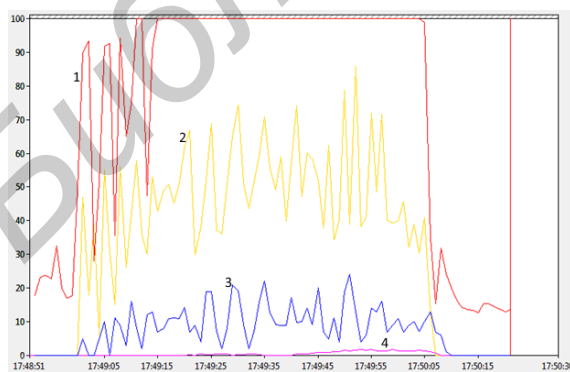


Рис. 3 – Показания монитора производительности сервера при асинхронном выполнении процессов: 1 - загрузка процессора; 2 - загрузка СУБД; 3 - скорость обработки запросов (запрос/с); 4 - количество запросов в очереди

На графиках видно, что скорость обработки запросов выше, чем в синхронном режиме. При этом процессор загружен на 100% при достижении определённого числа одновременных пользователей. В середине процесса тестирования начала формироваться очередь из запросов, ожидающих обработку. На основании результатов, полученных при моделировании нагрузки, можно сделать следующие выводы:

1. Показатели производительности существенно улучшаются, пропускная способность системы растёт;
2. Анализ характеристик работы сервера позволяет выделить узкое место (недостающий ресурс).

Для дальнейшего анализа производительности следует произвести оценку максимального показателя производительности при текущих параметрах системы. Для этого в качестве сценария нагрузочного моделирования выберем нагрузку по целевому параметру. Как показали тесты, узким местом системы является процессор, поэтому именно эта характеристика является целевым параметром профиля нагрузки. На рис. 4 представлены графики работы сервера, при котором процессор загружен на 70-90% (нормальный режим работы, характеризующийся отсутствием очереди запросов).

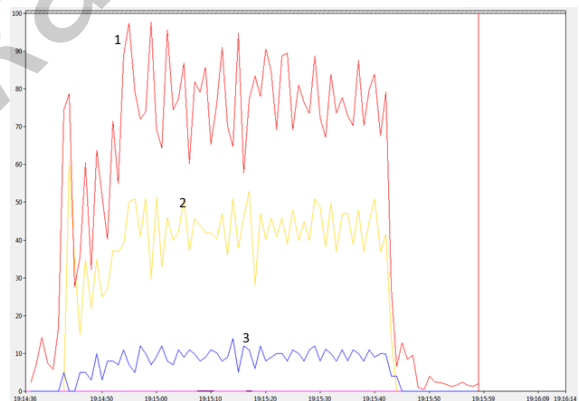


Рис. 4 – Показания монитора производительности сервера при нагрузке по целевому параметру: 1 - загрузка процессора; 2 - загрузка СУБД; 3 - скорость обработки запросов (запрос/с);

1. Особенности работы СУБД в WEB – Электронные данные – Режим доступа: <http://all4study.ru/sql/osobennosti-raboty-sudb-v-web.html>.
2. Счетчики производительности для ASP.NET / MSDN – Электронные данные – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/fxk122b4.aspx>.
3. Разработка и настройка для повышения производительности / MSDN – Электронные данные – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/ms227998.aspx>.
4. Asynchronous Programming Patterns / MSDN – Electronic resource – Mode of access: <http://msdn.microsoft.com/en-us/library/jj152938.asp>.