Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 681.5.015

Романович
Максим Анатольевич

Имитационное моделирование системы поддержки партнёрской сети

АВТОРЕФЕРАТ
на соискание степени магистра технических наук
По специальности 1-53 80 01 «Автоматизация и управление технологическими процессами и производствами»

_____

Научный руководитель

Лукьянец Степан Валерьянович
к.т.н., профессор

_____

Минск 2015

ВВЕДЕНИЕ

В настоящее время широкое распространение получили системы автоматизированного управления сложными технологическими производствами, складскими и транспортными потоками, информационно поисковыми и другими процессами. Они требуют комплексных клиент-серверных программных решений. Web-приложения становятся наиболее востребованными системами массового обслуживания информационного пространства. 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.

Целью данной работы является анализ производительности Web-приложений и выработка методов её оптимизации путём моделирования нагрузки на сервер, а также расчёт предполагаемого эффекта от масштабирования (развёртывания системы на серверном кластере). Для достижения поставленной цели необходимо решение следующих задач:

1. Рассмотрение методов анализа производительности сервера.
2. Создание необходимых программных решений для реализации моделирования нагрузки.
3. Анализ теоретических сведений о подходах повышения производительности и их реализация в выбранном модуле системы.
4. Моделирование работы системы на серверном кластере.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Диссертация состоит из введения, трёх разделов и заключения, в которых решены перечисленные задачи. При их рассмотрении использованы современные методы анализа и организации параллельных и распределённых вычислительных сетей, а также подходы имитационного моделирования сложных систем в случае работы системы на серверном кластере. Полученные в работе результаты являются ключевыми моментами для обеспечения работоспособности систем массового обслуживания любых размеров с различной интенсивностью нагрузки, что в настоящее время высоко востребовано в динамично развивающихся организациях и различных Webструктурах с растущей клиентской базой: социальные сети, поисковые агенты, системы облачных хранилищ и другие СМО. Основные положения диссертации отражены в работах автора [1-А, 2-А].

Партнёрская сеть как объект исследования. Проведенный анализ понятия партнёрской сети свидетельствует о пригодности его концепций для различного применения. Согласно проведённым исследованиям, наиболее пригодной концепцией сетевого взаимодействия партнёров является концепция,базирующаяся на единой системе поддержки. Она представляет собой механизм, в котором каждый партнёр взаимодействует только лишь с системой, которая, в свою очередь, обеспечивает связь со всеми другими объектами сети Рассматриваемый продукт включает в себя единое информационное пространство партнёрской сети, единое программное обеспечение, доступ к которому можно получить как через локальную сеть, так и через Интернет. Структура сетевого взаимодействия представлена на рисунке 1 [1-А].

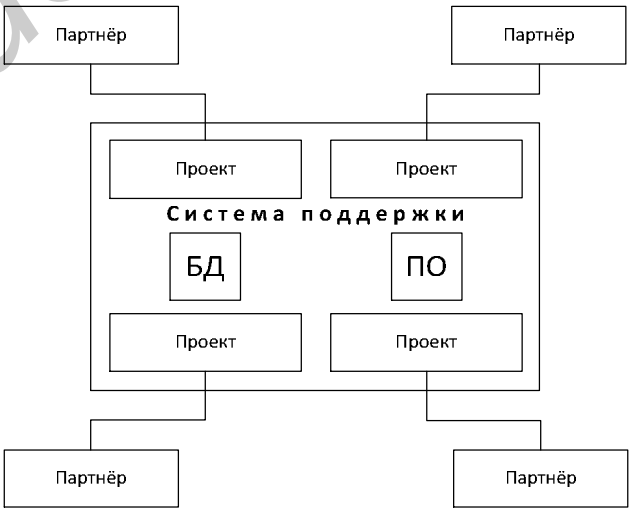


Рисунок 1 – Взаимодействие партнёров посредствам единой системы поддержки

В рамках данной работы рассматривается система поддержки партнёрской сети, разработка которой осуществляется на базе технологии ASP.NET. В ходе реализации данной системы созданы модули, включающие в себя интерфейс и определённую последовательность действий, служащие для решения ряда задач.Назначение системы подразумевает и предусматривает существование определённого числа пользователей, работающих с ней одновременно.

Ввиду того, что разработка системы предполагает реализацию процессов и алгоритмов Web-приложения, а также взаимодействие с базой данных, основное внимание будет уделено анализу и оптимизации обработки запросов серверным приложением, которая, в свою очередь, включает инициирование информационныхпотоковобмена данными.

Для тестирования и наблюдения за производительностью Web-приложений используется несколько средств. В ASP.NET включены счетчики производительности, которые можно использовать для отслеживания работы сервера приложений. Ниже приведены некоторые из них.

1. Запросы в очереди *(Requests Queued)*.
2. Время ожидания для запроса*(Request Wait Time)*.
3. Отклонённые запросы*(Requests Rejected)*.
4. Ошибкиприобработке*(Errors During Preprocessing)*.
5. Выполнение запросов*(Requests Executing)*.
6. Интенсивность обработки запросов*(Requests/Sec)*.

Чтобы определить, насколько хорошо программное обеспечение отвечает различным режимам использования, осуществляется нагрузочное тестирование. Комплексная среда разработки Visual Studio Ultimate позволяет использовать неограниченное число виртуальных пользователей для локального и удаленноговыполнения нагрузочного теста. Для каждого теста имеются свойства шаблона нагрузки, которые определяют способ корректировки моделируемой пользовательской нагрузки во время тестирования.

Для создания нагрузочного теста необходимо записать определённую последовательность пользовательских действий, которая в дальнейшем будет использоваться в качестве блока сценария нагрузки. В основу данного теста производительности положен Web-тест создания (вставки) объекта системы.

На следующем этапе определяется шаблон нагрузки. В контексте данного исследования выбран шаговый режим моделирования нагрузки, а также следующие метрики производительности:загрузка процессора, интенсивность запросов, количество запросов в очереди, число транзакций с БД.Показания монитора производительности сервера представлены на рисунке 3.

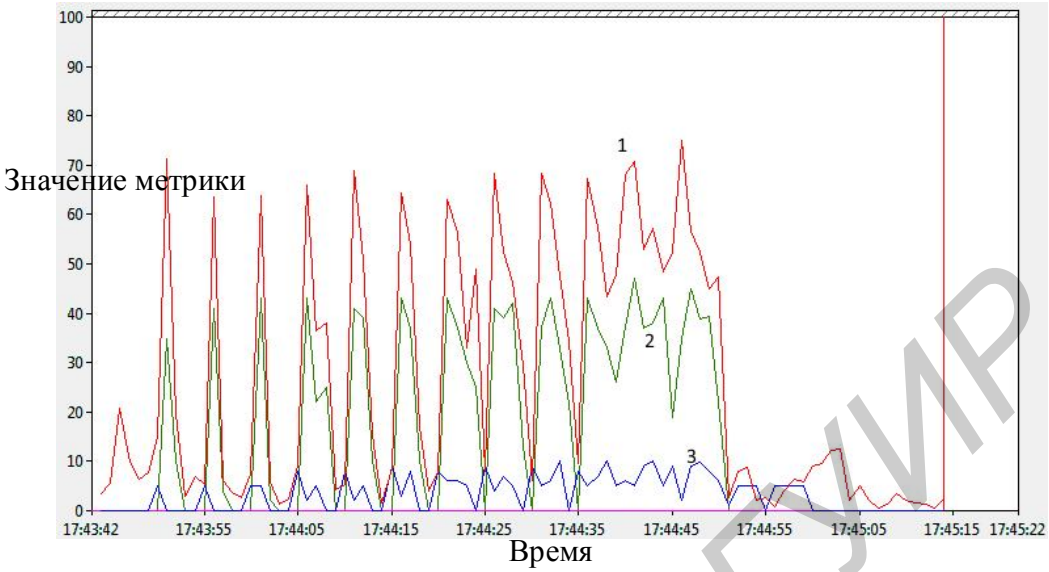


Рисунок 3 – Показания монитора производительности сервера при синхронном выполнении процессов: 1 - загрузка процессора, %; 2 - загрузка СУБД, %; 3 - скорость обработки запросов, запрос/с

Состояние выполнения теста отражается в инструменте моделирования Visual Studio в виде графиков (рисунок 4).

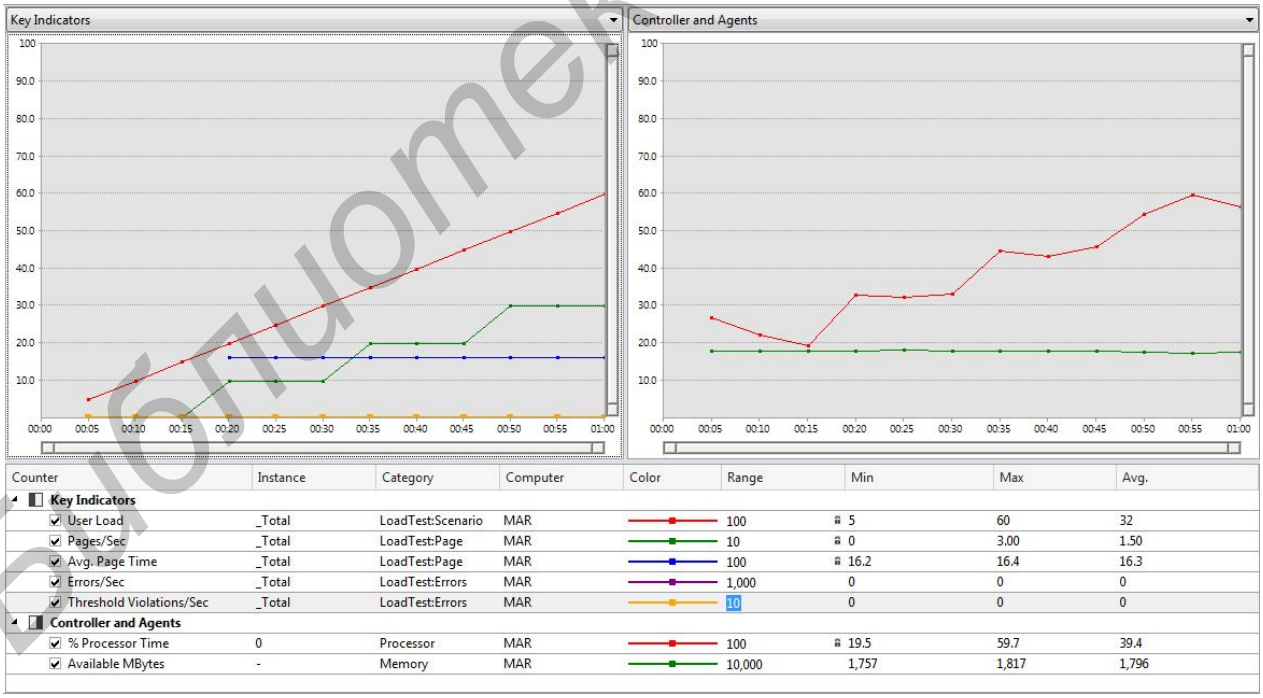


Рисунок 4 – Графический результат выполнения моделирования синхронных процессов

Среднее время ответа сервера достаточно высоко – 16,3 секунды и остаётся постоянным при возрастающем количестве пользователей. Это говорит о

неполном использовании ресурсов сервера, о чём также свидетельствует график загрузки процессора. Основываясь на этой информации, сделан вывод, что основным узким местом в процессе создания объекта являются длительные операции с внешними ресурсами, во время которых сервер приложения простаивает.

Анализ и реализация подходов к повышению производительности.Существует ряд подходов к увеличению производительности работы приложения:

1. Параллельные и распределённые (асинхронные) модели программирования.
2. Масштабирование системы.
3. Партиционирование, шардинг и репликация базы данных.
4. Настройка оптимальных параметров сервера.
5. Кеширование и др.

При реализации параллельных или распределённых вычислительных систем возникает ряд важных вопросов: в какой степени это позволяет ускорить решение задачи, насколько оптимально используются ресурсы, какова эффективность применяемого для параллельной обработки программного и аппаратного обеспечения.

Ускорение (speedup), т.е. относительный выигрыш во времени выполнения параллельного алгоритма для p процессоров по сравнению с последовательным вариантом вычислений, определяется величиной$S_p(n) = \frac{T_1(n)}{T_p(n)}$,где$T_1$ – затраты времени на выполнение алгоритма на одном процессоре,T_p – время выполнения алгоритма на p процессорах,n – абстрактный показатель вычислительной сложности решаемой задачи, например, количество входных данных задачи.

Эффективность (efficiency)– среднее время выполнения алгоритма, в течение которого процессоры реально задействованы для решения задачи [12]. Численно эффективность равна отношению ускорения к количеству процессоров в системе $E_p(n) = \frac{T_1(n)}{pT_p(n)} = \frac{S_p(n)}{p}$.

Из этих формул следует, что наилучшее значение ускорения $\mathrm{S_p(n) = p}$, а для эффективности $\mathrm{E_p(n) = 1}$. Однако на практике в некоторых случаях данные соотношения могут нарушаться. Оценить максимально достижимое ускорение помогает *закон Амдала*, который позволяет учесть влияние доли последовательных вычислений, которые не могут быть распараллелены, на максимально достижимое ускорение при параллельном выполнении алгоритма.

Он формулируется следующим образом. Пусть f – доля последовательных вычислений в применяемом алгоритме обработки данных, тогда ускорение процесса вычислений при использовании p процессоров ограничивается величиной ——— – Графики, иллюстрирующие ограничение закона Амдала при различной доле параллельных вычислений, приведены на рисунке 5.

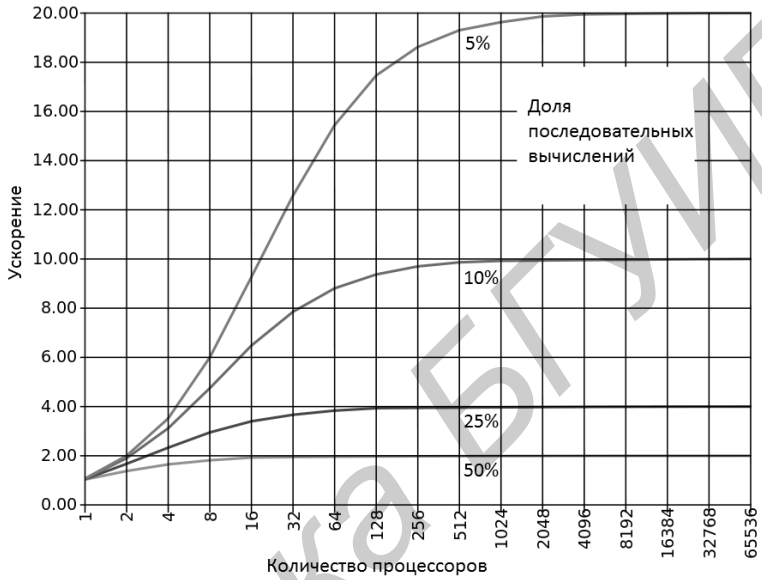


Рисунок 5 – Иллюстрация закона Амдала

Рассмотриммасштабирование системы. Масштабируемость (англ. scalability) – в электронике и информатике означает способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов (обычно аппаратных).

Выделяют два вида масштабирования информационных систем [15]:

1. Вертикальное масштабирование – увеличение производительности каждого компонента системы с целью повышения общей производительности.

2. Горизонтальное масштабирование – разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам (или их группам), и (или) увеличение количества серверов, параллельно выполняющих одну и ту же функцию. Данный вид масштабирования сервера приложения, реализация которого будет исследована далее, предусматривает работу на многомашинных комплексах, связанных сетью, поэтому необходимо оценить затраты времени на передачу данных по сети. Основные составляющие времени передачи данных – это следующие величины:

– tн – время начальной подготовки, т.е. длительность подготовки сообщения для передачи, поиска маршрута в сети и т. п.;

− tc – время передачи служебных данных (заголовок сообщения, блок данных для обнаружения ошибок передачи и т. п.);

− tк – время передачи одного слова данных по одному каналу передачи данных, длительность подобной передачи определяется полосой пропускания коммуникационных каналов в сети.

Зачастую узким местом в современных приложениях является БД. Проблемы с ней делятся, как правило, на два класса: производительность и необходимость хранения большого количества данных. В качестве решения данных проблем рассмотрены и даны подробные определения основным подходам оптимизации хранения информации и доступа к данным: партиционированию, репликации и шардингу.

Максимально эффективно решить проблему простоя сервера, выявленную при моделировании, способна реализация асинхронности при выполнении рабочего процесса (создания объекта), алгоритм которого представлен на рисунке 6а[2-А].

Выделены 2 действия, выполнение которых занимает длительное время и слабо коррелирует с ресурсами сервера:выполнение внешнего запроса для синхронизации данных со сторонними приложениями;рассылка уведомлений по электронной почте – использование SMTP сервера.

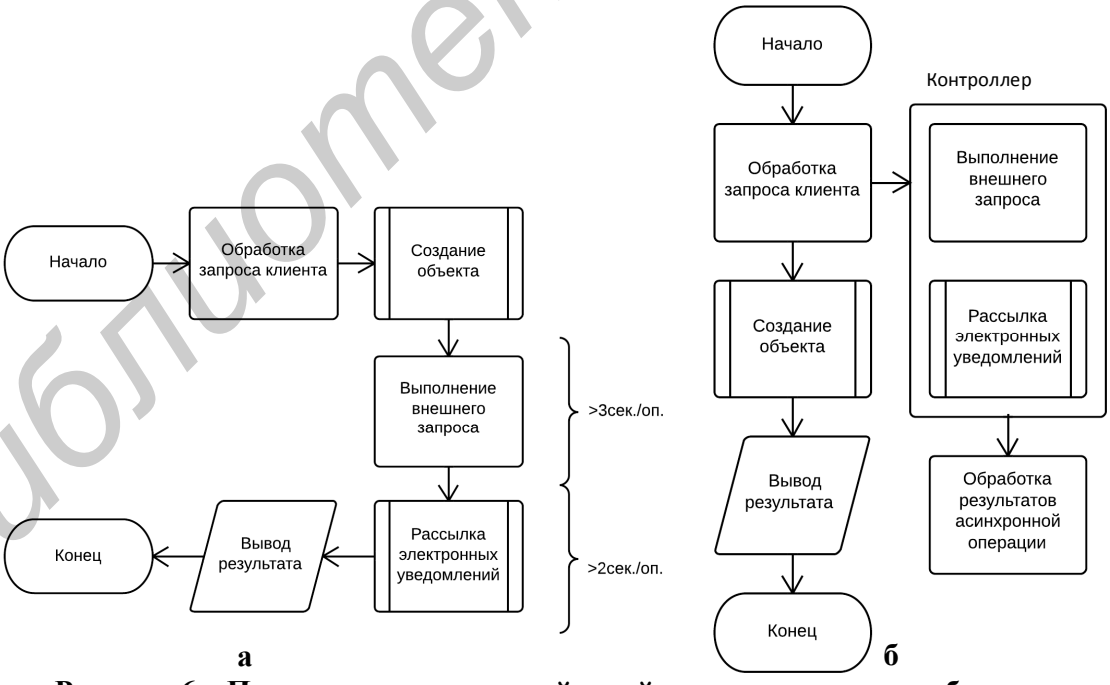


Рисунок 6 – Последовательность действий процесса создания объекта: а – в синхронном режиме, б – в асинхронном режиме.

Для реализации концепциипараллельных вычислений создан контроллер асинхронных методов, отвечающий за выполнение действия в фоновом процессе с последующей обработкой результатов: журналирование, выставление флагов. Схема последовательности действий рисунка 6а примет вид, представленный на рисунке 6б.

При моделировании асинхронных процессов использованпрофиль нагрузки, описанный ранее. Графики исследуемых процессов приведены на мониторе производительности сервера (рисунок7) и в системе моделирования (рисунок 8).

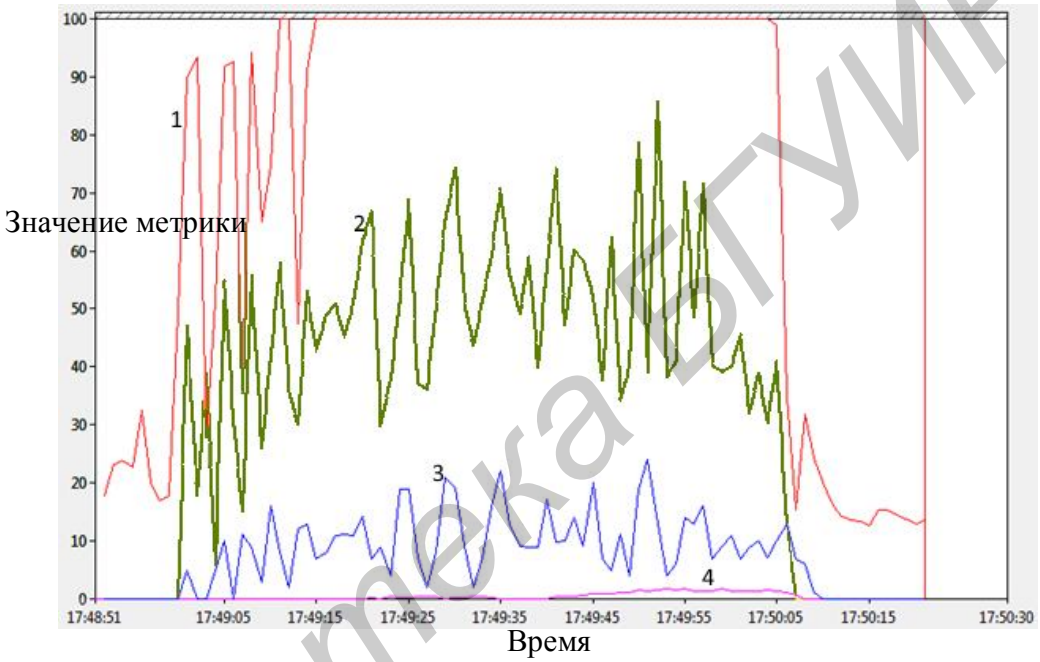
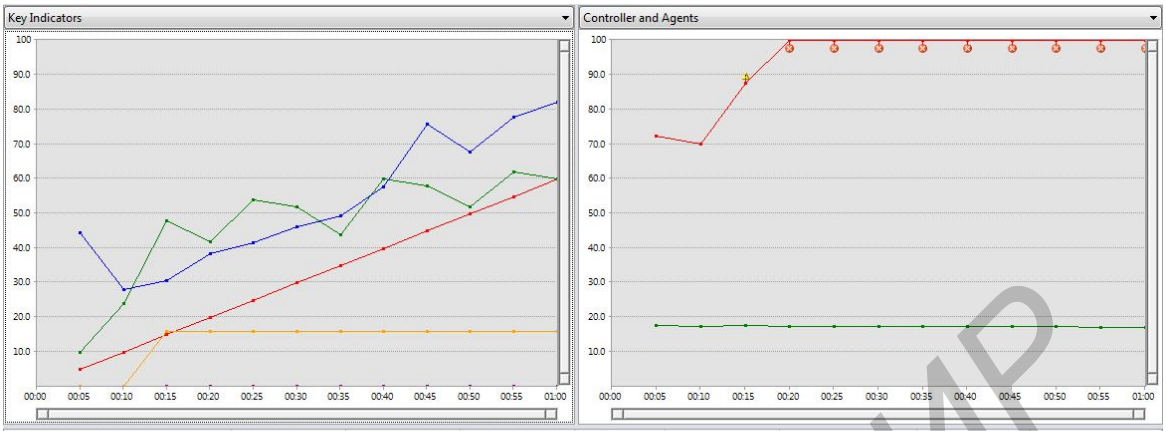


Рисунок 7 – Показания монитора производительности сервера при асинхронном выполнении процессов: 1 - загрузка процессора; 2 - загрузка СУБД; 3 - скорость обработки запросов (запрос/с); 4 - количество запросов в очереди



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg.
Key Indicators								
User Load	_Total	LoadTest:Scenario	MAR		100	5	60	32
Pages/Sec	_Total	LoadTest:Page	MAR		10	1.00	6.20	4.72
Avg. Page Time	_Total	LoadTest:Page	MAR		10	2.80	8.24	5.69
Errors/Sec	_Total	LoadTest:Errors	MAR		1,000	0	0	0
Threshold Violations/Sec	_Total	LoadTest:Errors	MAR		10	0	1.60	1.33
Controller and Agents								
% Processor Time	0	Processor	MAR		100	70.1	100	94.2
Available MBytes	-	Memory	MAR		10,000	1,707	1,776	1,739

Рисунок 8 – Графический результат выполнения моделирования асинхронных процессов

На графиках видно, что скорость обработки запросов выше, чем в синхронном режиме. При этом процессор загружен на 100% при достижении отметки в 15 одновременных пользователей. В середине процесса тестирования начала формироваться очередь из запросов, ожидающих обработку. Сделаны выводы, что показатели производительности существенно улучшаются,пропускная способность системы растёт; анализ метрик работы сервера позволяет выделить узкое место (недостающий ресурс).

Таким образом, встаёт вопрос о рассмотрении способов увеличения производительности путём устранения недостатка ресурсов сервера.Для этого необходимо произвести оценку максимального показателя производительности при текущих параметрах системы. В качестве сценария нагрузочного моделирования выберем нагрузку по целевому параметру – определим характеристики системы, при которых процессор загружен на 70-90%, что соответствует нормальному режиму работы.

Показатели производительности сервера при моделировании нагрузки по целевому параметру (рисунок 9) свидетельствуют об отсутствии очереди на фоне оптимальной загрузки процессора.Результаты моделирования (рисунок 10) показывают, что среднее количество одновременных клиентов при запрашиваемых параметрах работы равно 11 при среднем времени ответа 2,55 секунды. За 1 минуту сервер способен обработать 252 запроса, что превышает показатель при синхронном режиме работы (90 запросов в минуту) на 180%. Для

обеспечения одновременной работы большего числа пользователей необходимо увеличивать производительность на аппаратном уровне.

Дли систематизации полученных числовых результатов приводится сводная таблица 2, в которой за исходное значение приняты характеристики сервера в синхронном режиме, а также рассчитано их относительное увеличение при асинхронном режиме работы.

Таблица 2 – Анализ результатов моделирования при синхронном и асинхронном режимах работы

Метрика	Исходное значение	Сравниваемое значение	Отношение
Среднее время обработки запроса	16,4	2,55	6,43
Запросов в секунду	1,5	4,20	2,8
Общее количество запросов	90	252	2,8
Коэффициент загрузки процессора, %	39,4	73	1,85

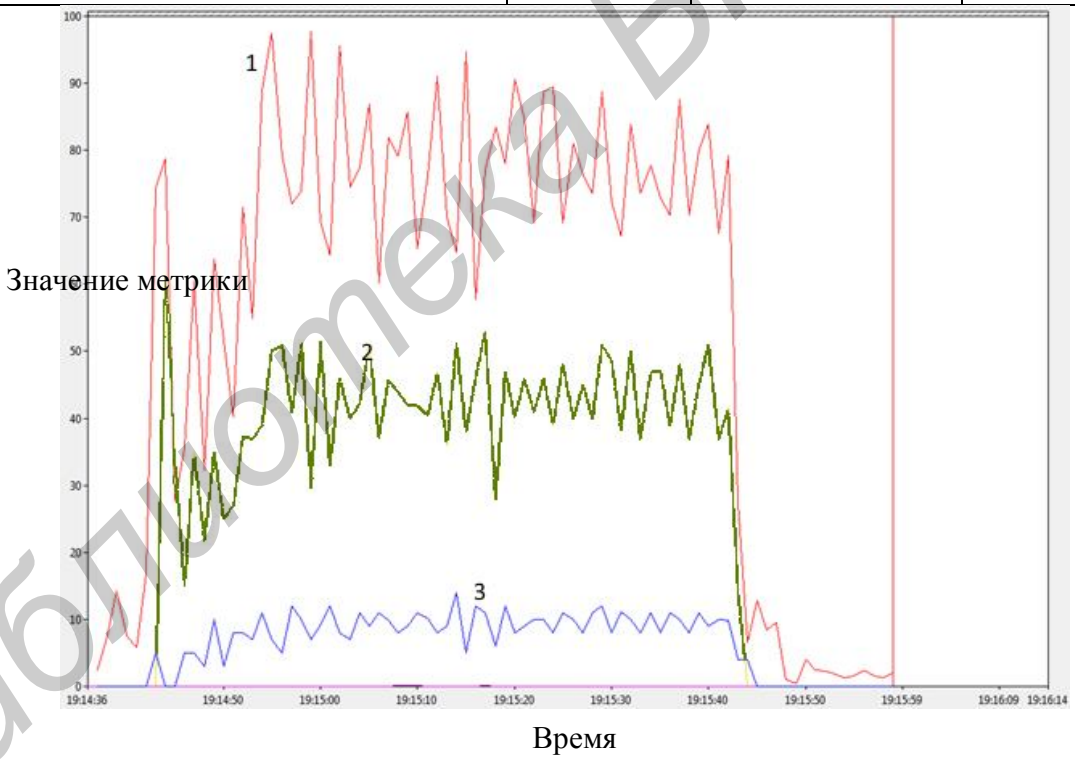
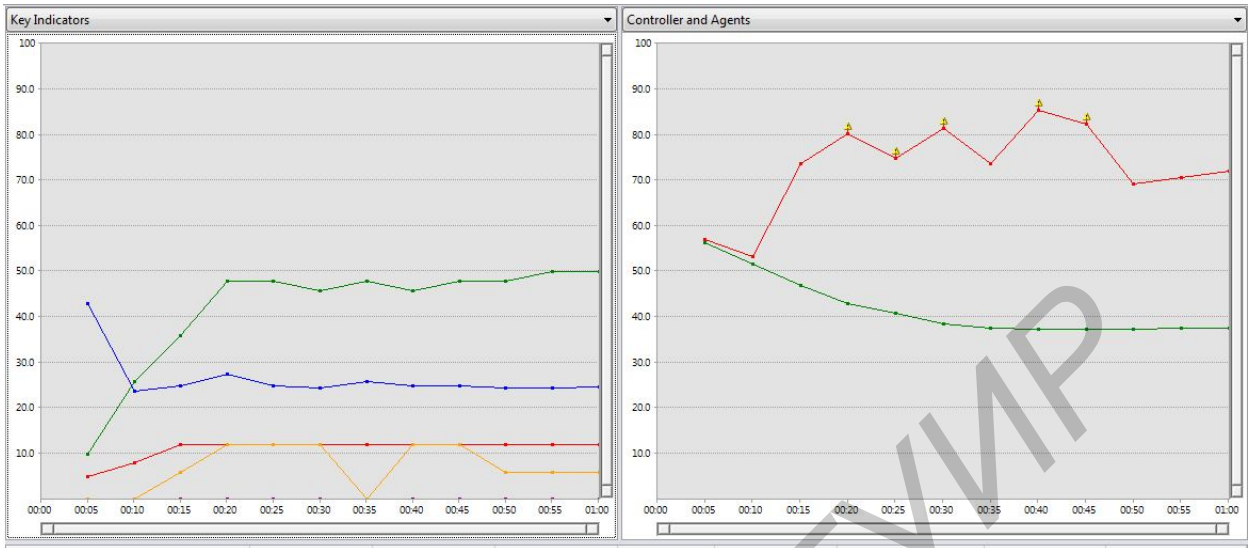


Рисунок 9 – Показания монитора производительности сервера при нагрузке по целевому параметру: 1 - загрузка процессора; 2 - загрузка СУБД; 3 - скорость обработки запросов (запрос/с);



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg.
Key Indicators								
User Load	_Total	LoadTest:Scenario	MAR		100	5	12	11
Pages/Sec	_Total	LoadTest:Page	MAR		10	1.00	5.00	4.20
Avg. Page Time	_Total	LoadTest:Page	MAR		10	2.39	4.32	2.55
Errors/Sec	_Total	LoadTest:Errors	MAR		1,000	0	0	0
Threshold Violations/Sec	_Total	LoadTest:Errors	MAR		10	0	1.20	0.70
Controller and Agents								
% Processor Time	0	Processor	MAR		100	53.5	85.6	73.0
Available MBytes	-	Memory	MAR		10,000	3,744	5,637	4,194

Рисунок 10– Результаты моделирования нагрузки по целевому параметру

Анализ результатов моделирования свидетельствует об оптимальной загрузке сервера, для дальнейшей оптимизации системы необходимо увеличивать производительность на аппаратном уровне. Рассмотренные теоретические сведения, являются предпосылкой к осуществлению моделирования работы системы на серверном кластере.

Под кластером понимают группукомпьютеров, объединённых высокоскоростными каналами связи, представляющую с точки зрения пользователя единый аппаратный ресурс.Специфика работы рассматриваемой системы поддержки партнёрской сети предусматривает взаимодействие с ней большого числа пользователей, территориально удалённых друг от друга. Для обеспечения масштабирования системы, а также повышения надёжности работы с ней возникает необходимость применения отказоустойчивых кластеров с балансировкой нагрузки.Принципиальная модель кластера для рассматриваемой системы приведена на рисунке 11.

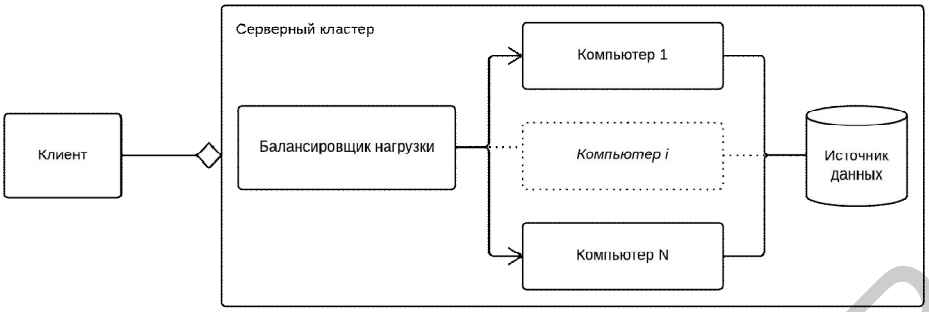


Рисунок 11 – Модель серверного кластера

Согласно данному представлению, запрос клиента не поступает непосредственно на сервер приложения, а направляется системой балансировки нагрузкипо наиболее оптимальному пути к наименее загруженному компьютеру.

В настоящее время существует ряд готовых решений для масштабирования и балансировки нагрузки Web-приложений при поддержанииих работы на комплексе серверов. Рассмотрено средство балансировки сетевой нагрузки NLB компании Microsoft, предназначенное для повышения надежности и масштабируемости серверных приложений, используемых на Web-серверах под управлением операционной системы WindowsServer.NLB включает следующие возможности:

1. Масштабируемость.
2. Высокая надежность.
3. Управляемость.
4. Простота использования.

Данная система позволяет добиться практически линейного повышения производительности при масштабировании без необходимости оптимизации аппаратного обеспечения сервера. Однако, для внедрения и сопровождения подобного рода систем необходимы дополнительные затраты на приобретение вычислительных ресурсов и их настройку, поэтому целесообразно произвести имитационное моделирование работы системы поддержки партнёрской сети на серверном кластере, с целью уточнения зависимости повышения производительности от количества дополнительных серверов.

Инструментарием для проведения эксперимента выбрана широко известная система моделирования общего назначения и одноимённый язык GPSS. Достоинством этого языка является подобие объектной модели реальному объекту исследования, а также ряд гибких возможностей по управлению временем и параметрами моделирования. Для интерпретации процессовфункционирования

системы в язык GPSSсоставим соответствующий алгоритм, представленный на рисунке 12.

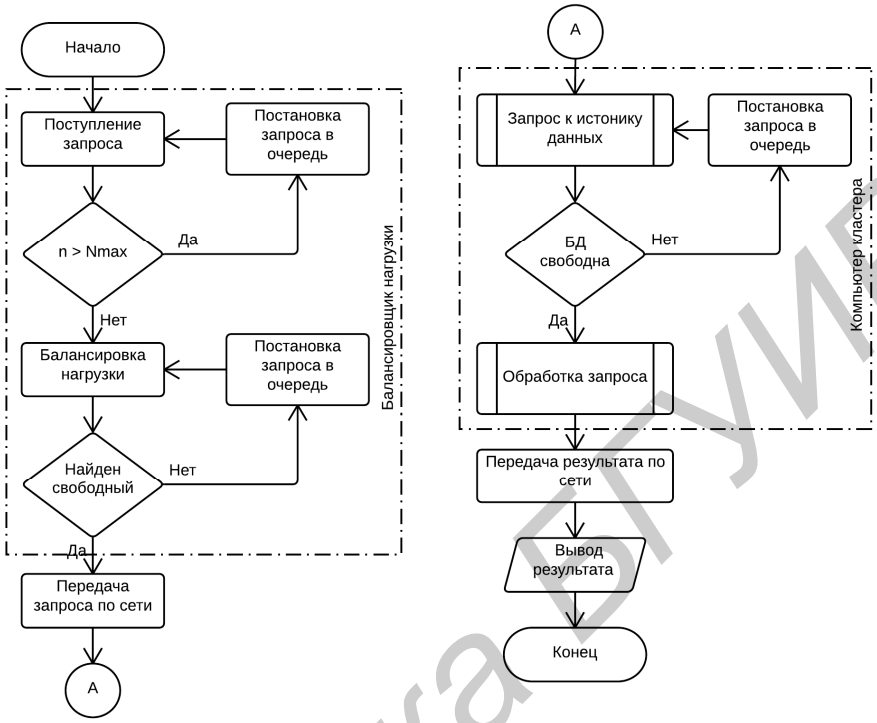


Рисунок 12 – Схема алгоритма обработки запросов на серверном кластере

Для описания приведённого алгоритма с последующей интерпретацией его блоков в язык GPSS составляется временная диаграмма, представленная на рисунке 13.

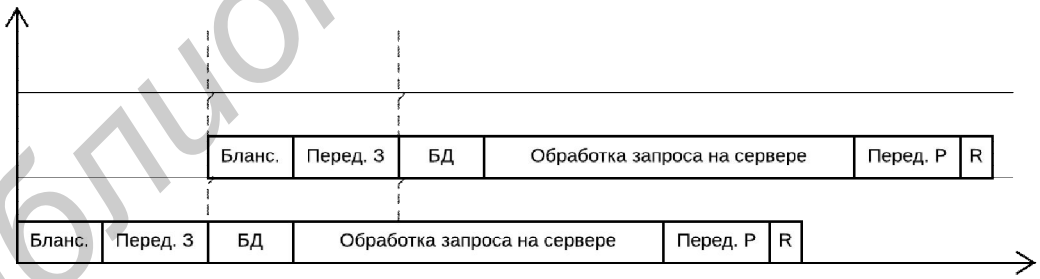


Рисунок 13 – Временная диаграмма обработки поступающих запросов

Интерпретируя алгоритм с учётом действующих ограничений и исходных данных в язык GPSS, получаем готовую для исследований модель системы в виде программного кода, блок моделирования которого представлен ниже:

```
; Генерация входного воздействия
          GENERATE (normal(1,3000,500))
abc       advance
          split     (n$abc#10+50-1),System_IN
```

```
; Блокалгоритма
System_IN ENTER     System
          queue     SBN_Q
          ENTER     SBN
          depart    SBN_Q
          advance   30,20
          SELECT MIN 5,com1,com10,,SR
          queue     p5
          ENTER     p5
          depart    p5
          advance   80,70
          LEAVE     SBN
          queue     BD
          ENTER     BD
          depart    BD
          advance   100
          LEAVE     BD
          advance   2000
          advance   80,70
          LEAVE     p5
          LEAVE     System
          terminate
```

При моделировании для определения входного воздействия будет применена особенность языка GPSS, которая позволяет генерировать величины согласно известным законам распределения. Наиболее часто встречающимся в СМО и целесообразным в контексте данной системы является нормальное (гауссово) распределение. В таблице 3представлены статистические результаты моделирования работы системы при нормальном законе распределения нагрузки.

Таблица 3 Статистика по моделированию при нормальном законе распределения нагрузки

Общее количество сгенерированных транзактов	2310
Количество обработанных за время моделирования транзактов	2200
Средняя очередь к системе	0
Среднее время пребывания транзакта в очереди к системе, мс	0
Средняя очередь к узлам кластера	5
Среднее время пребывания транзакта в очереди к узлу кластера, мс	164
Коэффициент использования оборудования, %:	
1.Система балансировки	17,6
2.Узлы кластера	87,4
3.Источник данных	10

Графические показатели характеристик системы свидетельствуют об отсутствии очереди и постоянном времени ответа. Коэффициент загрузки узлов кластера стремится к максимальному показателю (рисунок 14).

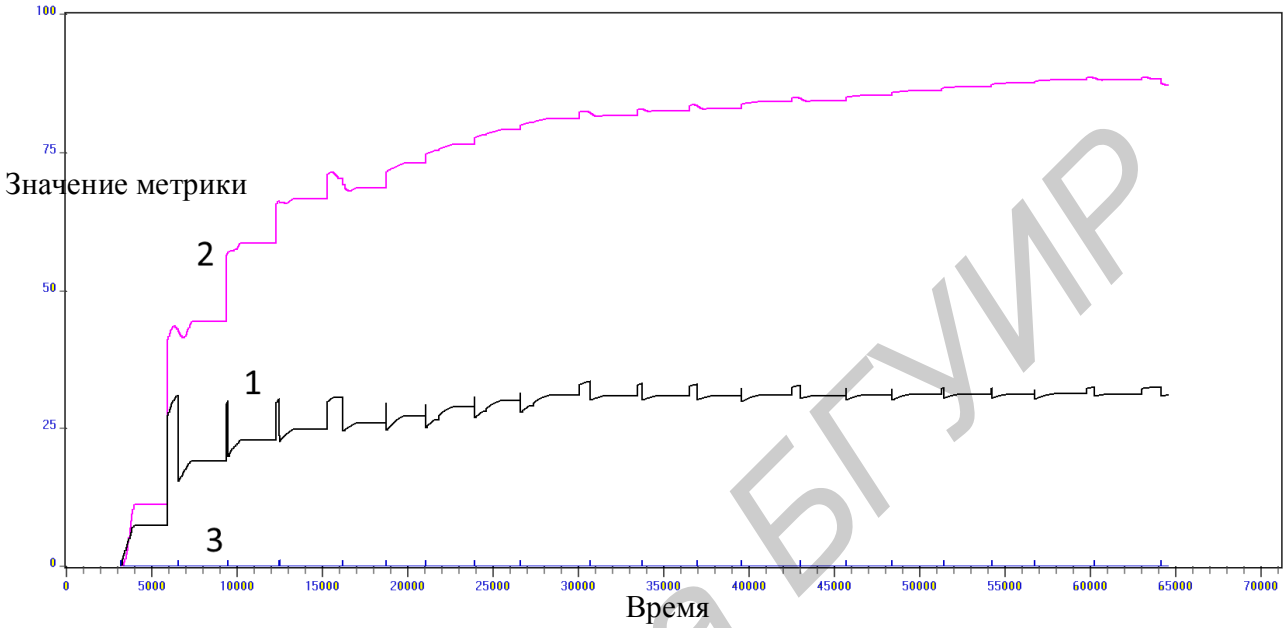


Рисунок 14 – Динамические характеристики системы при нормальном законе распределения нагрузке: 1 – среднее время пребывания транзакта в системе (время ответа), с (x10); 2 – коэффициент загрузки оборудования; 3 – очередь к системе

Проанализированы характеристики системы при конфигурации аппаратного обеспечения кластера, которое соответствует требованиям разработанной и функционирующей системы поддержки партнёрской сети COLLIOS: 32 вычислительных узла (максимальное количество для системы NLBкомпании Microsoft) и интенсивностью нагрузки, распределённой по нормальному закону с математическим ожиданием 0,7 с и отклонением 0,05 с.

Таблица 4Результаты моделирования работы системы с 32 вычислительными узлами

Количество обработанных за время моделирования транзактов	9018
Средняя время ответа сервера, с	2,7
Коэффициент использования оборудования, %:	
1.Система балансировки	56
2.Узлы кластера	91
3.Источник данных	41

Результаты моделирования (таблица 4) отражают способность системы обрабатывать до 9018 запросов в минуту при нормальном законе их поступления. Время ответа остаётся постоянным и приемлемым для комфортной работы

конечного пользователя. Следует отметить запас производительности СБН и СУБД, что позволяет проводить дальнейшее масштабирование увеличением количества вычислительных узлов, либо применять менее мощные средства для обеспечения балансировки и обмена данными, что позволит получить дополнительную экономическую эффективность.

ЗАКЛЮЧЕНИЕ

В ходе работы рассмотрено понятие партнёрских сетей. Проанализирована структура взаимодействия партнёрских организаций и обосновано решение о внедрении информационных технологий для реализации и поддержания её работоспособности, а именно использование Web-приложений в качестве СМО.Данные аспекты подробно освещены в авторской исследовательской работе [1-А], результаты которой были рекомендованы на Республиканский конкурс научных работ 2012 года и получили 2 категорию конкурса.

С целью оптимизации показателей работы системы рассматривались методы повышения производительности серверной части. Даны определения и исчерпывающие характеристики параллельным и распределённым вычислениям, а также обоснование их использования. Приведён закон Амдала, позволяющий учесть эффект от внедрения асинхронных вычислений.

Рассмотрено понятие и критерии масштабируемости системы, описаны концепции вертикального и горизонтального масштабирования. Перечислены накладные расходы и специфика реализации горизонтального масштабирования. Даны рекомендации по оптимизации работы с базами данных.

Для обеспечения требуемых показателей производительности системы реализованы методики параллельно программирования в рассматриваемый процесс. Разработаны необходимые классы и алгоритмы для достижения поставленной цели. Моделирование и анализ производительности после внедрение асинхронных операций показалприемлемый результат, а именно среднее время ответа сервера уменьшилось до 2,5 секунд, процессор перешёл в оптимальный режим работы. За 1 минуту один сервер способен обработать более 250 запросов клиентов.

Согласно спецификеобъекта исследования, при динамически растущем количестве пользователей,принято решение о рассмотрении эффекта внедрения горизонтального масштабирования. Дано определение понятия серверный кластер, рассмотрены основные типы кластеров. Решено использовать отказоустойчивый кластер с балансировкой нагрузки.

Проанализированы существующие методы и алгоритмы балансировки, произведён обзор и описание работы средств балансировки, выработан ряд допущений для последующего моделирования.

В качестве инструментария для имитационного моделирования выбран язык GPSS, ввиду достоинств, освещённых при анализе литературы, разработан алгоритм и составлена временная диаграмма работы кластера с СБН. На

основании этого создана модель и проведено моделирование работы комплекса, состоящего в первом случаем из 10, во втором – из 32 компьютеров. Статистические данные, полученные в процессе моделирования,свидетельствуют о практически линейном масштабировании – пропорциональном возрастании производительности при подключении к кластеру дополнительных узлов.

Исследованные характеристикразработанной системы поддержки партнёрской сети COLLIOSпутём моделирования её работы на серверном кластере, состоящем из 32 вычислительных узлов представляют максимальную производительность в 9018 обработанных за 1 минуту запросов при среднем времени ответа равном 2,8с.

Таким образом, в рамках данной работы рассмотрены методы анализа производительности Web-приложений, осуществлена их реализация, проведено имитационное моделирование с целью обоснования эффекта от масштабирования информационных систем, выработаны рекомендации по использованию инструментов и методик для решения поставленных задач.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

[1-А] Романович, М.А. Система поддержки партнёрской сети / М.А. Романович, П.М. Гунич // Информационные технологии и управление : материалы 48 научной конференции аспирантов, магистрантов и студентов. – Мн.: БГУИР, 2012. 95 с.

[2-А] Романович, М.А. Исследование асинхронной модели программирования / М.А. Романович, С.В. Лукьянец // Информационные технологии и системы 2014 (ИТС 2014) : материалы международной научной конференции. – Мн.: БГУИР, 2014. 80-81 с.