

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Системное программирование и компьютерные технологии

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему Модель системы массового обслуживания заявок визового центра

Студент

И.В. Саранчук

(И.О. Фамилия)

(личная подпись)

Руководитель

С.В. Мкртычев

(И.О. Фамилия)

(личная подпись)

Консультанты

Н.В. Яценко

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 20 _____ г.

Тольятти 2018

АННОТАЦИЯ

Тема: «Модель системы обслуживания заявок визового центра».

Целью выпускной квалификационной работы разработка модели системы массового обслуживания заявок визового центра.

Объектом выпускной квалификационной работы является процесс обслуживания заявок визового центра.

Предмет исследования – построение модели системы массового обслуживания заявок визового центра.

В данной выпускной квалификационной работе исследуется построение систем массового обслуживания. Рассматриваются основные виды моделирования систем и методы реализации систем.

В работе представлены результаты тестирования нескольких систем массового обслуживания.

Структура выпускной квалификационной работы состоит из введения, трех глав, заключения, списка литературы.

Во введении описывается актуальность проводимого исследования, формулируется цель и ставятся задачи, которые необходимо решить.

В первой главе приводится анализ предметной области и общее описание моделирования систем.

Во второй главе рассматривается процесс выбора необходимой модели системы массового обслуживания для дальнейшей реализации и подготовка исходных данных.

В третьей главе рассматривается процесс реализации и тестирование имитационной модели системы массового обслуживания.

Бакалаврская работа состоит из 50с. пояснительной записки, включая 10 рисунков, 3 таблицы, 20 источников и 2 приложения.

ABSTRACT

The title of the graduation work is “The Model of the Visa Application Service System”.

The graduation work consists of an explanatory note, including 7 figures, 3 tables, 3 graphic, 5 foreign sources in a language and 2 appendicies.

The goal of the graduation work is to develop the model of the queuing system of application.

The object of the work is the process of servicing applications for the visa center.

The subject of the work is the development a model of the queuing system for visa center applications.

In this graduation work the results of testing several queuing systems are presented.

The main tasks of recognition systems and methods for increasing recognition level are considered. In one model, a single-channel queuing system with limited queue systems is considered and in another one-channel queuing system with unlimited queue.

In the first part the review of recognition systems analysis of methods is provided.

The second part considers the process of selecting the necessary model of the queuing system for further implementation and the preparation of the initial data.

The third part examines the implementation and testing of the simulation model of the queuing system.

Оглавление

| | |
|--|----|
| ВВЕДЕНИЕ | 6 |
| ГЛАВА 1 МЕТОДЫ МОДЕЛИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ | 8 |
| 1.1 Обоснование подхода к моделированию системы массового обслуживания заявок визового центра..... | 9 |
| 1.2 Математическая модель системы массового обслуживания | 13 |
| 1.3 Разработка алгоритма системы массового обслуживания..... | 15 |
| ГЛАВА 2 СОЗДАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ ЗАЯВОК ВИЗОВОГО ЦЕНТРА..... | 16 |
| 2.1 Выбор модели системы массового обслуживания для реализации | 17 |
| 2.2 Описание одноканальных систем массового с ограниченной и неограниченной длинной очереди с ожиданием | 19 |
| 2.2.1 Одноканальная система массового обслуживания с ограниченной длинной очереди..... | 19 |
| 2.2.2 Одноканальная система массового обслуживания с неограниченной длинной очереди..... | 21 |
| 2.3 Составление одноканальной системы массового обслуживания на примере визового центра..... | 22 |
| ГЛАВА 3 ОЦЕНКА АДЕКВАТНОСТИ МОДЕЛИ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ ЗАЯВОК ВИЗОВОГО ЦЕНТРА | 29 |
| 3.1.1 Выбор среды программирования | 29 |
| 3.1.2 Выбор системы управления базы данных..... | 32 |
| 3.2 Реализация имитационной модели системы массового обслуживания заявок визового центра..... | 37 |
| 3.3 Проверка адекватности системы массового обслуживания заявок визового центра | 39 |

| | |
|--------------------------------------|----|
| ЗАКЛЮЧЕНИЕ | 42 |
| СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ | 43 |
| Приложение А | 46 |
| Приложение Б..... | 49 |

ВВЕДЕНИЕ

Всем без исключения не по нраву ожидание. Но сокращение времени ожидания, как правило, требует дополнительных затрат. Чтобы понять, есть ли надобность в инвестировании, важно знать влияние инвестиции в ожидаемое время ответа системы. По этой причине нам необходимы модели и методы, с целью рассмотрения подобных состояний. Из этого рассматриваются несколько незамысловатых моделей систем массового обслуживания.

Сосредоточенность уделяется к методам анализа этих моделей. Значимыми областями применения моделей очередей считаются производственные системы, транспортные и системы хранения, системы связи и системы обработки данных. День ото дня, количество подключенных устройств к интернету увеличивается, все это пользуется большим спросом потому, что устройства совершенствуются с каждым днем. Все это ориентировано на потребность более скоростного, чем есть интернет-соединения, дабы запрашиваемый запрос клиента потреблял меньше затрат, а т.е времени и трафика.

Для подключения к устройствам различным по размеру, форме, емкости, мощности, пользовательскому языку и операционной системе необходим весьма немудреный и глобальный по размерам интерфейс, который будет способен подключаться к устройствам с использованием простых протоколов. На сегодняшний день, практически любое устройство может подключиться к Интернету и обмениваться данными через веб-интерфейс. Из этого следует, что если бы приложения имели веб-интерфейс, то пользователю было намного проще перенести свои личные данные в облако.

Поскольку вся идея перехода в систему массового обслуживания заключается в том, что появляется возможность получать доступ в любое время из любого устройства с максимальной надежностью и неиссякаемой вычислительной мощностью.

Целью данной выпускной квалификационной работы является разработка модели системы массового обслуживания заявок визового центра.

Объект исследования– процесс обслуживания заявок визового центра.

Предмет исследования – построение модели системы обслуживания заявок визового центра.

Таким образом, для успешной реализации этой цели выполняется разработка обобщенного и детального алгоритма моделирования работы СМО. Практическое осуществление поставленной задачи выполняется с помощью программы, написанной на языке программирования java. В ходе нескольких осуществлений работы СМО фиксируются результаты функционирования системы. На основе приобретенных данных выполняется построение графических зависимостей, позволяющих произвести исследование СМО, подвергнуть анализу характеристики ее деятельности. В результате производятся заключения , в совокупности определяющие деятельность СМО, воссоздающие ее плюсы и минусы, определяются направления улучшения ее качества.

ГЛАВА 1 МЕТОДЫ МОДЕЛИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Имитационное моделирование - представляет собой процесс создания и анализа цифрового прототипа физической модели для прогнозирования ее производительности в реальном мире. Моделирование моделирования используется, чтобы помочь дизайнерам и инженерам понять, в каких условиях и в какой степени часть может потерпеть неудачу и какие нагрузки она может выдержать. Моделирование моделирования также может помочь предсказать модели потока жидкости и теплопередачи. Он анализирует приблизительные условия работы, применяя программное обеспечение для моделирования.

Система массового обслуживания – это математическое исследование систем ожидания или очередей. Модель очередей построена таким образом, что можно прогнозировать длину очереди и время ожидания. Теория массового обслуживания обычно рассматривается как отрасль исследований операций, поскольку результаты часто используются при принятии деловых решений о ресурсах, необходимых для предоставления услуги.

Во время работы системы может возникнуть ситуация отказа в обслуживании. Она возникает в результате переполнения всех ячеек буфера. Заявка, поданная в этот момент, теряется.

Среди основных характеристик системы обслуживания выделяют: время, затраченное на ожидание начала обслуживания заявки, время обслуживания одной заявки, длина очереди и т.д.

Рассмотрим основные термины систем массового обслуживания. Прибор, который обслуживает заявки – эвм. Он решает задачи связанные с обработкой информации. Заявки – это задачи, которые необходимо решить.

Выдача заявки – промежуток времени, отсчитываемый от нажатия кнопки подачи до конца обработки заявки.

При выполнении задач также фиксируется очередность заявок. При имитационном моделировании СМО играет роль оригинала, а в качестве модели представляется программа имитационным моделированием. [3]

Основным недостатком моделирования системы массового обслуживания является численный характер результатов. Аналитическое исследование на выходе дает результат в форме формулы, а метод моделирования - это число, которое зависит от входных данных. В моделируемых системах учитываются случайные факторы, но для моделирования требуются большие временные затраты.

Имитационное моделирование выполняется с помощью специальных языков и не требует тщательного изучения данного вопроса специалистом. Совместное применение моделирования и аналитического исследования дают лучший результат, но для их совместного использования необходимы знания, опыт и квалификация.

1.1 Обоснование подхода к моделированию системы массового обслуживания заявок визового центра

Моделирование – это процесс, используемый для определения и анализа требований к данным, необходимых для поддержки бизнес-процессов в рамках соответствующих информационных систем в организациях. Таким образом, процесс моделирования данных включает в себя профессиональных обработчиков данных, работающих в тесном контакте с заинтересованными сторонами бизнеса, а также потенциальными пользователями информационной системы.

Аналитическое моделирование представляет из себя процесс описания существующего объекта и поиска решений с помощью аналитических формул. Для создания модели помимо формул, дифференциальных и интегро дифференциальных уравнений или неравенств используются правила из различных точных наук. Основные типы математического моделирования показаны на рисунке 1.1.



Рисунок 1.1– Типы математического моделирования

В имитационном моделировании вместо использования реального прототипа системы или объекта применяется модель системы в виде программы для ЭВМ. Такое моделирование применяется для проведения экспериментов, если невозможно учитывать различные причинные связи, стохастические переменные и результат. Модель позволяет рассмотреть поведение модели при изменении внутренних и внешних условий в течении продолжительного времени [2].

Модель - упрощенное воспроизведение реального объекта. Все детали, а также процессы и явления, протекающие в системе упрощаются.

Имитационное моделирование имеет различные направления. Основные из них показаны на рисунке 1.2.



Рисунок 1.2 – Разновидности(направления) имитационного моделирования

Статистическое моделирование заключается в проведении статистических испытаний над моделью. Моделирование учитывает различные явления и процессы, подчиняющиеся вероятностным законам. В основе модели лежит генерация псевдослучайных чисел с помощью ЭВМ.

Имитация с помощью компьютера позволяет изучить объект в различные моменты времени. Чтобы найти точное решение, нужно множество раз повторять исследования.

Динамическая система состоит из закона, описывающего различные изменения состояния объекта в течении определенного времени и множества величин, которые изменяются во времени. Система описывает процесс, явление или объект. Среди динамических объектов рассматривают производственные, механические, физические и другие объекты [5].

Дискретно–событийное моделирование - моделирование с помощью которого может быть описана функциональность протекающего процесса в виде события.

Системная динамика - подход к пониманию нелинейного поведения сложных систем с течением времени с использованием запасов, потоков, внутренних циклов обратной связи, табличных функций и временных задержек. Диаграммы, созданные с использованием модели, могут быть перенесены на компьютер.

Агентное моделирование - модель на основе АВМ представляет собой класс вычислительных моделей для моделирования действий и взаимодействий автономных систем (как отдельных, так и коллективных объектов, таких как организации или группы) с целью оценки их влияния на систему в целом. Он объединяет элементы теории игр, сложных систем, возникновения, вычислительной социологии, мультиагентных систем и эволюционного программирования. Методы Монте-Карло используются для введения случайности. В частности, в рамках экологии ПРО также называются индивидуальными моделями (IBM), и лицами в рамках ПРО. Обзор недавней литературы по индивидуальным моделям, основанным на агентах моделям и мультиагентным системам показывает, что ПРО используются в неиндустриальных смежных областях, включая биологию, экологию и социальную науку. Моделирование на основе агентов связано с концепцией многоагентных систем или многоагентной имитацией, но отличается от нее, поскольку целью ПРО является поиск объяснительного понимания коллективного поведения агентов.

В отличие от системной динамики поведение системы рассматривается как результат работы различных агентов. Поведение агентов определяется с помощью специалиста на компьютере.

Когнитивное моделирование – это то приближение когнитивных процессов животных (преимущественно человеческих) для целей понимания и прогнозирования. Когнитивные модели могут быть разработаны внутри или без когнитивной архитектуры, хотя они не всегда легко различимы.

В отличие от когнитивных архитектур, когнитивные модели, как правило, ориентированы на единое когнитивное явление или процесс (например,

изучение списка), как взаимодействуют два или более процесса (например, визуальный поиск и принятие решений) или делают предсказания поведения для конкретной задачи или инструмент (например, как внедрение нового программного пакета повлияет на производительность). Когнитивные архитектуры, как правило, сосредоточены на структурных свойствах моделируемой системы и помогают сдерживать развитие когнитивных моделей в архитектуре. Аналогичным образом, разработка модели помогает информировать об ограничениях и недостатках архитектуры. Некоторые из наиболее популярных архитектур для когнитивного моделирования включают ACT-R, Clarion и Soar.

1.2 Математическая модель системы массового обслуживания

Система массового обслуживания - система с помощью которой обрабатываются различные заявки. Такие системы применяются в области торговли, образования, медицины. В системе решаются задачи обслуживания различных требования.

Вероятность поступления k заявок за время t вычисляется по формуле:

$$P_k(t) = \frac{\mu^k t^k}{k!} e^{-\mu} \quad (1)$$

Основным параметром системы является время, затраченное на обслуживание заявок. Для распределения по экспоненциальному закону время описывается с помощью функции распределения. Функция определяется по формуле:

$$F(t) = 1 - e^{-\mu} \quad (2)$$

Среди систем обслуживания выделяют замкнутые и незамкнутые. В замкнутых системах заявки поступают из самой системы, а в незамкнутых заявки поступают из других источников.

Основная эффективность СМО определяется количеством свободных каналов способных обработать заявку. Также для определения эффективности используется длина очереди и различные коэффициенты, определяющие простой каналов.

Рассмотрим одну из основных характеристик системы массового обслуживания $\alpha = \lambda/m$. Если $\alpha/n < 1$, то, следует, что очередь не может расти бесконечно. Важные характеристики системы массового обслуживания:

$$1. P_0 = \left[\sum_{k=0}^{n-1} \frac{a^k}{k!} \times \frac{a^n}{n!(1-\frac{a}{n})} \right]^{-1} \quad (3)$$

2. Вероятность того, что ровно k каналов обслуживания заняты, при условии, что общее количество требований к обслуживанию не превышает количество единиц обслуживания:

$$P_k = \frac{a^k}{k!} P_0, \quad (4)$$

при $1 \leq k \leq n$

3. Вероятность наличия заявок в системе в случае, когда их число больше, чем количество обслуживающих каналов:

$$P_k = \frac{a^k}{n! n^{k-n}} P_0, \text{ при } k \geq n \quad (5)$$

4. Вероятность того, что все обслуживающие каналы заняты:

$$P_n = \frac{a^n}{n!(1-\frac{a}{n})} P_0; \left(\frac{a}{n} < 1\right) \quad (6)$$

5. Среднее время ожидания для начала обслуживания в системе:

$$t_{об} = \frac{P_n}{\mu(n-a)} \quad (7)$$

6. Средняя длина очереди:

$$l_{ср} = \frac{a P_n}{n(1-\frac{a}{n})} \times \frac{a P_n}{n! n(1-\frac{a}{n})^2} \times P_0; \left(\frac{a}{n} < 1\right) \quad (8)$$

7. Среднее число свободных от обслуживания каналов:

$$N_0 = \sum_{k=0}^{n-1} \frac{n-k}{k!} \quad (9)$$

8. Коэффициент простоя каналов:

$$K_{пр} = \frac{N_0}{n} \quad (10)$$

9. Среднее число занятых обслуживанием каналов:

$$N_0 = n \quad (11)$$

10. Коэффициент загрузки каналов:

$$K_a = \frac{N_a}{n} \quad (12)$$

1.3 Разработка алгоритма системы массового обслуживания

Рассмотрим процесс разработки СМО. Алгоритм обслуживания представлена следующим образом:

1. Определение требований к автоматизированной системе.

- этап исследования концепции автоматизированной системы;
- этап исследования и утверждения технического задания.

2. Обоснование выбора тех. средств и проектирование:

- исследование моделей автоматизированной системы;
- стадия проектирования программного обеспечения;
- стадия проектирования интерфейса;
- стадия разработки программного кода;
- стадия создания и оформления документации.

3. Внедрение системы.

4. Сопровождение системы.

Основная цель проектирования – определение структуры данных, определение системных требований и свойств системы.

В процессе проектирования реализуется модель исходной системы, учитывая различные аспекты её функционирования.

Моделирование позволяет рассмотреть основные решения рассматриваемых вопросов, обнаружить различные ошибки на ранних стадиях, ускорить процесс разработки. При разработке модели также рассматриваются функции преобразования входной информации и выходной.

ГЛАВА 2 СОЗДАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ ЗАЯВОК ВИЗОВОГО ЦЕНТРА

Изучение линий ожидания, называемое теорией массового обслуживания, является одним из наиболее старых и наиболее широко используемых методов анализа.

Система массового обслуживания – это обычное явление, затрагивающее людей, покупающих продукты, покупающие бензин, внесение банковского вклада или ожидая по телефону, чтобы первые доступные резервисты компании ответили. Очереди, еще один термин для систем массового обслуживания, также могут принять форму машин, ожидающих ремонта, грузовиков в очереди для разгрузки или самолетов, выстроившихся на взлетно-посадочную полосу, ожидающих разрешение на вылет. Три основных компонента процесса очередей заявок – это прибытие, обслуживание и фактическая линия ожидания.

Модели линейного программирования, как представляется, особенно подходят для теории массового обслуживания, поскольку требуется время для решения задач.

Мы кратко изучаем характеристики и используем преимущества и недостатки некоторых из наиболее актуальных моделей теории массового обслуживания. Обсуждается как аналитические модели систем ожидания могут помочь менеджерам оценить стоимость и эффективности систем обслуживания, мы начинаем с рассмотрения ожидающих затрат на систему, а затем описываем характеристики системы массового обслуживания и основные математические предположения, используемые для разработки моделей очередей; мы также предоставляем уравнение, необходимое для вычисления эксплуатационные характеристики системы обслуживания и показать в нашем случае, как они используются.

2.1 Выбор модели системы массового обслуживания для реализации

Каждая система массового обслуживания состоит из определенного числа обслуживающих единиц (приборов, устройств, пунктов, станций), которые будут обозначаться как каналы. Как правило, она состоит из одного или нескольких каналов обслуживания заявок клиентов, прибывающих в некотором роде и имеющих некоторые требования к обслуживанию. Клиенты (заявки) представляют пользователей, рабочие места, транзакции или программы. Они прибывают в СМО для обслуживания, ожидая обработки, если есть комната ожидания, и покидают систему после обработки. Иногда заявки теряются, если отсутствует буфер для хранения заявок. Системы массового обслуживания описываются распределением времени между поступлением заявки, распределением времени на обслуживание поступившей заявки, количества каналов и т. д. По числу каналов системы массового обслуживания подразделяют на одноканальные и многоканальные.

Заявки вводятся в систему массового обслуживания как правило не на постоянной основе, а в случайном порядке, создавая таким образом так называемый случайный поток системных требований (требования). Запросы на обслуживание также прерываются в течение некоторого случайного времени. Случайный характер потока заявок и время обслуживания приводит к тому, что система массового обслуживания загружается неравномерно: одновременно происходит накопление очень большого количества заявок (они либо находятся в очереди, либо покидают систему массового обслуживания), в другой период СМО работает с недогрузкой или бездействием.

Предметом исследований теории массового обслуживания являются операции, которые касаются проблемы предоставления качественных, но экономичных услуг, связанных с непостоянными числами и временем или последовательностями. В теории массового обслуживания термин «клиенты» используется, будь то обращение к людям или вещам, в сопоставлении таких переменных, как то, как поступают клиенты, как удовлетворяет их требования, среднее время обслуживания и степень вариаций, а также время простоя. Когда

такие переменные идентифицируются как для клиентов, так и для объектов, выбор может быть сделан на основе экономических преимуществ.

В качестве показателей эффективности СМО используется:

- вероятность того, что в системе нет заявок;
- среднее время обработки заявки;
- среднее количество поступивших заявок в системе (заявки стоящие в очереди на обслуживание и заявки, которые обрабатываются);
- среднее время, которое заявка проводит в очереди ожидания;
- среднее время, которое заявка проводит в системе (состоит из времени ожидания, кроме время обслуживания);
- вероятность того, что заявка должна ждать своей очереди;
- вероятность того, что заявка будет необслужена и покинет очередь.

Системы массового обслуживания делятся на два основных типа (класса): система массового обслуживания с отказами и система массового обслуживания с ожиданием (очередью). В системе массового обслуживания с отказами заявка, которая прибыла в то время, когда все каналы заняты, получает отказ, оставляет систему массового обслуживания и в будущем процессе обслуживания не участвует. В системе массового обслуживания с ожиданием заявка, которая прибыла в тот момент, когда все каналы заняты, не получает отказ в обслуживании, но переходит в очередь обработки.

СМО с ожиданием делятся на разные типы в зависимости от того, как организована очередь: с ограниченной или неограниченной длиной очереди, с ограниченным временем ожидания и т.п.

Существует стандартная система для классификации систем массового обслуживания:

- по распределению вероятностей для процесса прибытия заявок;
- по распределению вероятностей для процесса обслуживания заявок;
- по количеству каналов (серверов);

- по максимальному количеству клиентов, размещенных в системе ожидания (обслуживаемых или ожидающих обслуживания).

Приоритет заявок(клиентов) может быть как абсолютным, когда более важная заявка "вытесняет" из-под обслуживания обычную заявку, так и относительным, когда более важная заявка получает лишь "лучшее" место в очереди.

В данной выпускной квалификационной работе будет рассмотрена одноканальная система массового обслуживания.

2.2 Описание одноканальных систем массового с ограниченной и неограниченной длиной очереди с ожиданием

2.2.1 Одноканальная система массового обслуживания с ограниченной длиной очереди

В системе массового обслуживания с ограниченной очередью число мест m в очереди ограничено. Откуда следует, что заявка, которая поступила в момент времени, когда очередь загружена, отклоняется и покидает систему массового обслуживания. Пример схемы одноканальной системы может быть представлен в упрощенной форме с использованием следующего графического представления, который показан на рисунке 2.1.

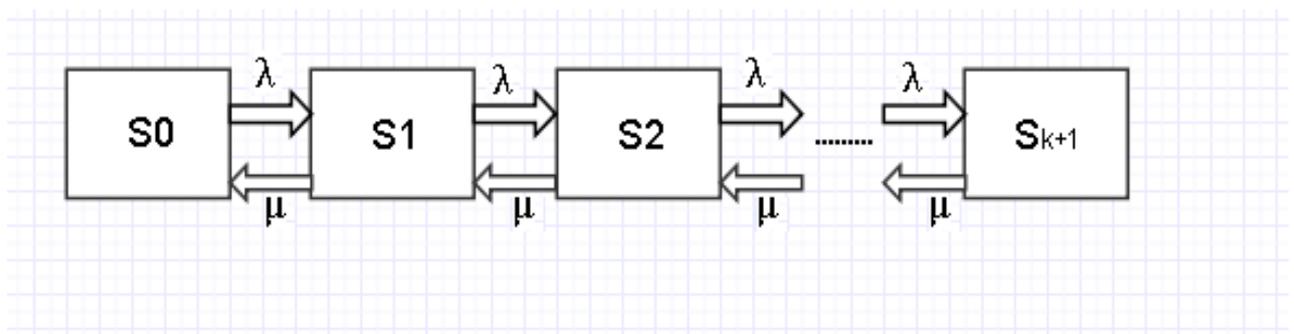


Рисунок 2.1– Графическое представление одноканальной системы массового обслуживания с ограниченной длиной очереди

Чтобы получить необходимые формулы, можно воспользоваться тем, что если взять $g = m + 1$ и

$$\lambda_i = \lambda, \mu_i = \mu, (i = \overline{0, m}) \quad (13)$$

Окончательные вероятности состояния одноканальной системы массового обслуживания выглядят следующим образом:

$$p_0 = (1 + \rho + \rho^2 + \dots + \rho^{m+1})^{-1} = \frac{1-\rho}{1-\rho^{m+2}} \quad (14)$$

$$p_k = \rho^k \times p_0, (k = \overline{1, m+1}) \quad (15)$$

Вероятность того, что в системе будет k транспортных единиц, будет равна:

$$p_0 = p_k = \rho^{m+1} \times p_0 \quad (16)$$

При $m = 0$ (канал обслуживания занят, но при этом нет очереди ожидания) формулы (15), (16) переходят в формулы для одноканальной системы массового обслуживания с отклонением.

Запрос, который поступила в систему массового обслуживания, получает отказ в обслуживании, если система массового обслуживания находится в состоянии S_{m+1} , т.е. вероятность получить отказ в обработке:

$$p_{\text{отк}} = p_{m+1} = \rho^{m+1} \times p_0 \quad (17)$$

Условная пропускная способность системы

$$Q = p_{\text{отк}} = 1 - p_{\text{отк}} = \rho^{m+1} \times p_0 \quad (18)$$

Окончательная пропускная способность системы равна

$$A = \lambda \times Q = \lambda \times (1 - \rho^{m+1} \times p_0) \quad (19)$$

Среднее число заявок, стоящих в очереди $L_{\text{оч}}$, рассчитывается по формуле

$$L_{\text{оч}} = 1 \times p_2 + 2 \times p_3 + \dots + m \times p_{m+1} \quad (20)$$

и может быть записано в виде

$$L_{\text{оч}} = p_2 \times \frac{1-\rho^m [m \times 1-\rho + 1]}{(1-\rho)^2} \times p_0 \quad (21)$$

При $\rho = 1$ формула (21) преобразовывается к виду,

$$L'_{\text{оч}} = \frac{m \times (m+1)}{2 \times (m+2)}, \text{ при } \rho = 1 \quad (22)$$

$L_{\text{обс}}$ – среднее число заявок, находящихся в системе массового обслуживания, рассчитывается по формуле (23)

$$L_{\text{обс}} = 1 \times p_1 + 1 \times p_2 + 1 \times p_3 + \dots + m \times p_{m+1} = p_1 + L_{\text{оч}} \quad (23)$$

и может быть преобразовано к виду

$$L_{\text{обс}} = \rho \left(1 + \rho \times \frac{1 - \rho^m}{1 - \rho^2} \right) \times \frac{m \times (1 - \rho) + 1}{1 - \rho^2} \rho_0 \quad (24)$$

При $\rho = 1$, из (24) получаем:

$$L'_{\text{обс}} = \frac{m^2 + m + 2}{2 \times (m + 2)}, \text{ при } \rho = 1 \quad (25)$$

2.2.2 Одноканальная система массового обслуживания с неограниченной длиной очереди

Примером такой СМО может служить способ обслуживания клиентов в ресторане быстрого питания с одним официантом. То есть заказ клиента берется и по завершении транзакции берется заказ следующего клиента в очереди ожидания. Таким образом, каждый клиент ресторана проходит через один канал, где он размещает заказ, оплачивает и собирает продукты. Если клиентов больше, чем можно обслуживать, возникает очередь ожидания. Пример схемы одноканальной системы может быть представлен в упрощенной форме с использованием следующего графического представления, который показан на рисунке 2.2.

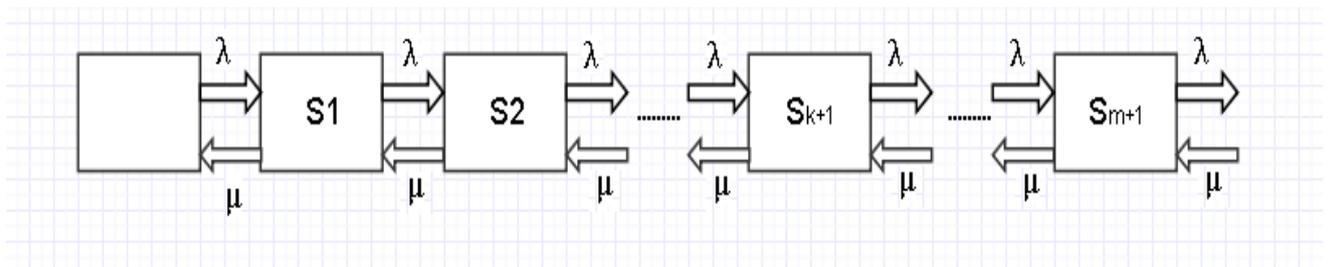


Рисунок 2.2– Графическое представление одноканальной системы массового обслуживания с неограниченной длиной очереди

Все характеристики такой системы массового обслуживания можно получить из формул предыдущего раздела, полагая в них $m \rightarrow \infty$. При этом необходимо различать два существенно разных случая: а) $\rho \geq 1$; б) $\rho < 1$. В первом случае, как это видно из формул (23), (24), $\rho_0 = 0$ и $\rho_k = 0$ (при всех конечных значениях k). Это означает, что при $t \rightarrow \infty$ очередь неограниченно возрастает, т.е. этот случай практического интереса не представляет.

Проанализируем случай, когда $\rho < 1$. Формулы (23) и (24) при этом запишутся в виде

$$p_0 = 1 - \rho, \quad (26)$$

$$p_k = \rho^k \times 1 - \rho, \quad k = 1, 2, \dots \quad (27)$$

Поскольку в системе очередности отсутствует ограничение на длину очереди, последующая заявка может быть подана, т.е. условная пропускная способность равна

$$Q = p_{обс} = 1 \quad (28)$$

окончательная пропускная способность равна

$$A = \lambda \times Q = \lambda \quad (29)$$

Среднее число заявок в очереди выведем из формулы(30) при $m \rightarrow \infty$

$$L_{оч} = \frac{\rho^2}{(1-\rho)} \quad (30)$$

Среднее число обрабатываемых заявок есть

$$L_{обс} = \rho \times Q = \rho \quad (31)$$

а среднее число заявок, которые находятся в системе массового обслуживания, равно

$$L_{смo} = L_{оч} + L_{обс} = \frac{\rho}{1-\rho} \quad (32)$$

Среднее время нахождения заявки в системе массового обслуживания и в очереди рассчитывается по формулам представленным выше.

2.3 Составление одноканальной системы массового обслуживания на примере визового центра

Рассмотрим пример одноканальной СМО с ожиданием– визовый центр. Количество заявок, ожидающих своей очереди, ограничено и равно 3 ((N– 1) = 3). Если все каналы заняты, то есть в очереди уже находятся три заявки, то очередная заявка, поступившая в канал, в очередь не становится. Поток заявок, прибывающих в визовый центр, распределен по закону Пуассона и имеет интенсивность $\lambda = 0,85$ (заявок в час). Время обслуживания заявки распределено по показательному закону и в среднем равно 1,05 ч.

Требуется определить вероятностные характеристики визового центра, работающего в стационарном режиме.

1. Параметр потока обслуживаний заявок:

$$\mu = \frac{1}{\tau} = \frac{1}{1,05} = 0,95$$

Из условия задачи следует, что анализируемый процесс характеризуется потоком приложений Пуассона (значение параметра $\lambda = 0,85$). Мы предполагаем, что время обслуживания получило экспоненциальное распределение ($\mu = 0,95$). Таким образом, интенсивность движения (потока) берет значение

$$\rho = \frac{\lambda}{\mu} = \frac{0,85}{0,95} = 0,89$$

2. Вычислим финальные вероятности системы:

$$P_0 = \frac{1-\rho}{1-\rho^{N+1}} = \frac{1-0,89}{1-0,89^5} = 0,25$$

$$P_1 = \rho \times P_0 = 0,89 \times 0,25 = 0,22$$

$$P_2 = \rho^2 \times P_0 = 0,89^2 \times 0,25 = 0,2$$

$$P_3 = \rho^3 \times P_0 = 0,89^3 \times 0,25 = 0,18$$

$$P_4 = \rho^4 \times P_0 = 0,89^4 \times 0,25 = 0,16$$

4. Вероятность отказа в обслуживании заявки:

$$P_{\text{отк}} = P_4 = 0,16$$

5. Условная пропускная способность визового центра:

$$q = 1 - P_{\text{отк}} = 0,84$$

6. окончательная пропускная способность визового центра:

$$A = q \times \lambda = 0,716 \text{ (заявок в час)}$$

7. Среднее число заявок, находящихся на обслуживании и в очереди (то есть в моделирование СМО):

$$\begin{aligned} L_s &= \frac{\rho \times 1 - N - 1 \times \rho^N + N \times \rho^{N+1}}{1 - \rho \times 1 - \rho^{N+1}} \\ &= \frac{0,89 \times 1 - 4 + 1 \times 0,89^4 + 4 \times 0,89^5}{1 - 0,89 \times 1 - 0,89^5} = 1,8 \end{aligned}$$

8. Среднее время пребывания заявки в системе:

$$W_s = \frac{L_s}{\lambda \times (1 - P_N)} = \frac{1,8}{0,85 \times (1 - 0,16)} = 2,5$$

9. Средняя продолжительность пребывания заявки в очереди на обслуживание:

$$W_q = W_s - \frac{1}{\mu} = 2,5 - \frac{1}{0,95} = 1,4$$

10. Среднее число заявок в очереди (длина очереди):

$$L_q = \lambda \times 1 - P_N \times W_q = 0,85 \times 1 - 0,16 = 1,02$$

Оценивая работу визового центра, приведенного в примере, можно отметить, что оно удовлетворительное, поскольку визовый центр не обслуживает заявки в среднем в 16% случаев. Проанализируем дальше, можно ли повысить качество работы одноканальной системы массового обслуживания путем увеличения длины очереди (в приведенном примере – расширение коллическвa заявок, ожидающих своей очереди). Примем, что $N \rightarrow \infty$, остальные параметры системы массового обслуживания остаются неизменными.

Система алгебраических уравнений, описывающих работу системы массового обслуживания при $t \rightarrow \infty$ для любого $n = 0, 1, 2, \dots$, следующая:

$$\begin{aligned} \lambda \times P_0 + \mu \times P_1 &= 0, n = 0 \\ \lambda \times P_{n-1} + \mu \times P_{n+1} - (\lambda + \mu) \times P_n &= 0, n > 0 \end{aligned} \quad (33)$$

Решение такой системы уравнений имеет вид:

$$P_n = 1 - \rho \times \rho^n, n = 0, 1, 2, \dots, \text{ где } \rho = \frac{\lambda}{\mu} < 1 \quad (34)$$

Характеристики одноканальной СМО с ожиданием, без ограничения на длину очереди следующие:

- среднее число находящихся в системе клиентов (заявок) на обслуживание:

$$L_s = \sum_{n=0}^N n \times P_n = \frac{\rho}{1-\rho}; \quad (35)$$

- средняя продолжительность пребывания клиента в системе:

$$W_s = \frac{L_s}{\lambda \times (1 - P_N)} = \frac{1,8}{\mu \times (1 - \rho)} = 2;$$

- среднее число клиентов в очереди на обслуживание:

$$L_q = L_s - \frac{\lambda}{\mu} = \frac{\rho^2}{1-\rho}; \quad (36)$$

- средняя продолжительность пребывания заявок клиента в очереди:

$$W_q = \frac{L_q}{\lambda} = \frac{\rho}{\mu(1-\rho)}. \quad (37)$$

Пусть рассматриваемый визовый центр располагает неограниченной длиной очереди для прибывающих в систему заявок, то есть имеет неограниченную длину очереди. Требуется определить конечные значения следующих вероятностных характеристик:

- вероятность состояний системы;
- среднее число заявок в системе (в исполнении и в очереди);
- средняя продолжительность пребывания заявки в системе (в исполнении и в очереди);
- среднее число заявок в очереди на исполнение;
- средняя продолжительность пребывания заявки в очереди.

Основные показатели одноканальной системы массового обслуживания с ограниченной очередью представлены на графике 1.

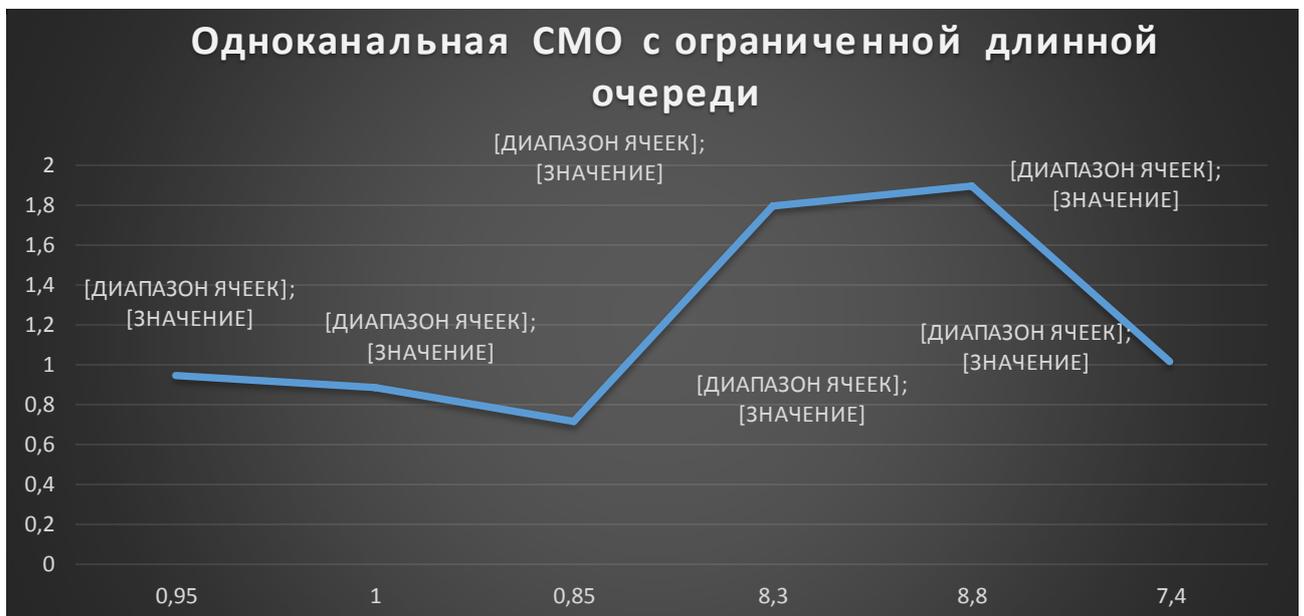


График 2. Основные показатели одноканальной системы массового обслуживания с ограниченной очередью

1. Параметр потока обслуживания μ и приведенная интенсивность потока заявок ρ определены выше:

$$\mu=0,95; \rho=0,89.$$

1. Вычислим предельные вероятности системы по формулам:

$$P_0 = 1 - \rho = 1 - 0,89 = 0,11$$

$$P_1 = (1 - \rho) \times \rho = (1 - 0,89) \times 0,89 = 0,1$$

$$P_2 = (1 - \rho) \times \rho^2 = (1 - 0,89) \times 0,89^2 = 0,1$$

$$P_3 = (1 - \rho) \times \rho^3 = (1 - 0,89) \times 0,89^3 = 0,08$$

$$P_4 = (1 - \rho) \times \rho^4 = (1 - 0,89) \times 0,89^4 = 0,07$$

$$P_5 = (1 - \rho) \times \rho^5 = (1 - 0,89) \times 0,89^5 = 0,06$$

и т. д.

Следует отметить, что P_0 определяет долю времени, в течение которого обслуживающий заявку аппарат вынужденно бездействует (простаивает). В данном случае она составляет 11%, так как $P_0 = 0,11$.

2. Среднее число заявок, находящихся в системе (в обслуживании и в очереди):

$$L_s = \frac{\lambda}{\mu} = \frac{\rho}{1-\rho} = \frac{0,89}{1-0,89} = 8,3.$$

3. Средняя продолжительность пребывания заявки в системе:

$$W_s = \frac{L_s}{\lambda} = \frac{1}{\mu \times (1-\rho)} = 9,8.$$

4. Среднее число заявок в очереди на обслуживание:

$$L_q = L_s - \frac{\lambda}{\mu} = \frac{\rho^2}{1-\rho} = 7,4.$$

5. Средняя продолжительность пребывания заявки в очереди:

$$W_q = \frac{L_q}{\lambda} = \frac{1}{\mu \times (1-\rho)} = 8,8.$$

6. Условная пропускная способность системы:

$$q=1$$

то есть каждая заявка, пришедшая в систему, будет обслужена.

7. Окончательная пропускная способность:

$$A = q \times \lambda = 0,85(\text{заявок в час})$$

Следует отметить, что организация, занимающаяся приёмом документов на визу, прежде всего, интересуется количеством клиентов, которые подадут

заявку на подачу документов на визу, при снятии ограничения на длину очереди. Допустим, в первоначальном варианте количество обслуживающихся заявок было равно 3. Частота m возникновения ситуаций, когда прибывающая в очередь заявка не имеет возможности присоединиться всегда к очереди:

$$m = \lambda \times P_N. \tag{38}$$

В нашем примере при $N=3+1$ и $\rho=0,89$

$$m = \lambda \times P_N \times \rho^4 = 0,85 \times 0,25 \times 0,89^4 = 0,13 \text{ заявок в час.}$$

При 12-ти часовом режиме работы визового центра это эквивалентно тому, что он в среднем за смену (день) будет терять $12 \times 0,13 = 1,6$ клиентов. Снятие ограничения на длину очереди позволяет увеличить количество обслуженных клиентов в нашем примере в среднем на 1,6 заявку за смену (12 ч работы).

Основные показатели одноканальной системы массового обслуживания с неограниченной очередью представлены на графике 2.

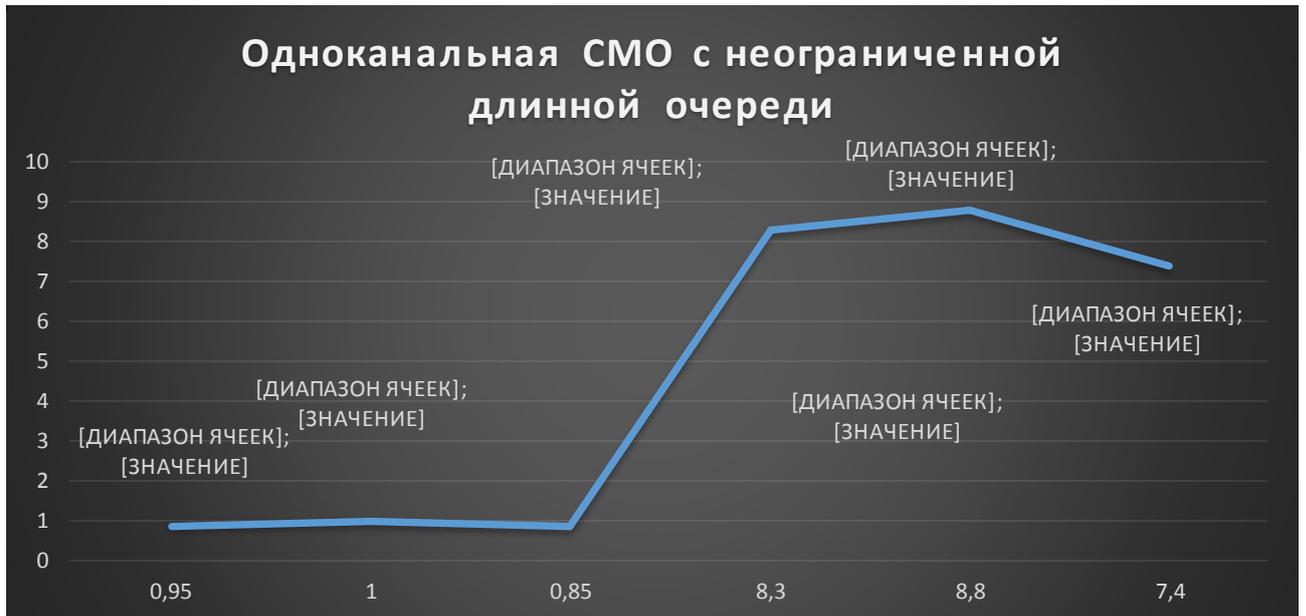


График 2. Основные показатели одноканальной системы массового обслуживания с неограниченной очередью

Ясно, что решение относительно расширения очереди для поступающих заявок, прибывающих от клиентов, должно основываться на оценке экономического ущерба, который обусловлен потерей клиентов.

Данные полученные после сравнения конечных результатов, полученных в ходе вычисления, представлены на графике 3.

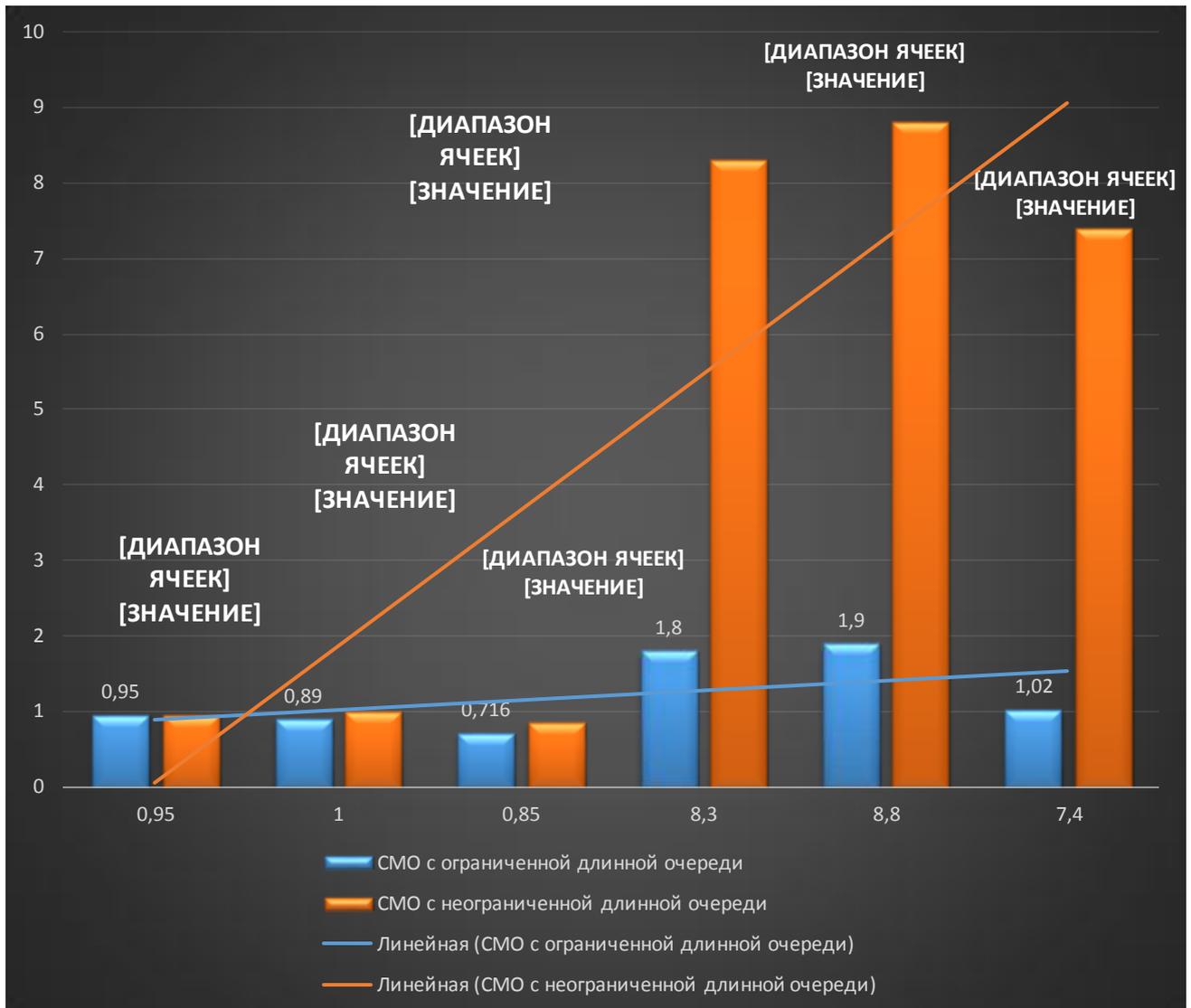


График 1. Сравнение одноканальных СМО с ограниченной и неограниченной длиной очереди

Как видно по графику 1, одноканальная система массового обслуживания с неограниченной длиной очереди является лучшим выбором для визового центра. По основным параметрам систем видно, что ограничение очереди существенно снижает показатель эффективности работоспособности модели.

ГЛАВА 3 ОЦЕНКА АДЕКВАТНОСТИ МОДЕЛИ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ ЗАЯВОК ВИЗОВОГО ЦЕНТРА

3.1.1 Выбор среды программирования

Для разработки был произведен анализ нескольких языков программирования.

Рассмотрим основные из них.

C++ - язык общего назначения, в котором используются разные парадигмы, такие как процедурное и объектное программирования. Так же язык имеет множество библиотек, поддержку многопоточности и строгую типизацию.

Достоинствами объектно– ориентированного языка программирования C++:

- масштабируемость. На языке C++ разрабатывают приложения для самых разных платформ;
- возможность создания обобщенных алгоритмов для различных типов данных, их специализация, и вычисления на шаге компиляции, которые используют шаблоны.

Недостатки объектно– ориентированного языка программирования C++:

- Наличие большого количества нарушений безопасности приводит к различным ошибкам, которые трудно обработать;
- Разработчик вынужден придерживаться специальных правил кодировки. Также в языке присутствует множество проблем унаследованных от языка C. В языке отсутствует сборщик мусора, и C++ не защищен от ошибок типа "переполнение буфера";
- плохая поддержка модульности. Подключение интерфейса наружного модуля через препроцессорную вставку заголовочного файла (#include) серьезно приостанавливает обработку информации, при подключении огромного количества модулей;

- недостаток информации о типах данных во время обработки информации (СТТИ);

- язык С++ является сложным для изучения и для компиляции.

С++ является популярным языком и применяется для различных разработок ПО. С помощью языка создаются операционные системы, различные программы для встраиваемых систем и серверов.

Существует набор команд, который чаще всего называют исходный код или просто код. Командами являются или «функции» или «ключевые слова». Ключевые слова – это основные строительные блоки данного языка. Характеристики функции – это сложные строительные блоки, потому что они записаны в терминах более простых функций. С++ позволяет использовать зарезервированные слова, которые необходимы для написания своих функций.

PHP – широко используемый язык сценариев общего назначения с открытым исходным кодом, который особенно подходит для веб-разработки и может быть встроен в HTML [15].

PHP можно использовать во всех основных операционных системах, включая Linux, многие варианты Unix (включая HP-UX, Solaris и OpenBSD), Microsoft Windows, Mac OS X, RISC OS и, возможно, другие. Сегодня PHP также поддерживает большинство веб-серверов. Это включает Apache, IIS и многие другие. И это включает в себя любой веб-сервер, который может использовать двоичный файл FastCGI PHP, например lighttpd и nginx. PHP работает либо как модуль, либо как процессор CGI.

Таким образом, с PHP есть свобода выбора операционной системы и веб-сервера. Кроме того, у вас также есть выбор использования процедурного программирования или объектно-ориентированного программирования (ООП) или их сочетание [19].

Преимущества языка PHP:

- встроенные модули подключения к базе данных: возможность легко подключаться к базе данных с помощью PHP, поскольку многие веб-сайты управляются данными и контентом;

- поддержка связи с другими службами, используя такие протоколы, как LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (в Windows) и многие другие;
- прост в использовании: он использует синтаксис типа C, поэтому для тех, кто знаком с C, их очень легко подобрать, и очень легко создать скрипты веб-сайта.;
- работа с cookies и сессиями;
- полезные функции обработки текста, в том числе совместимые с Perl регулярные выражения (PCRE), а также множество расширений и инструментов;
- легко найти поставщиков услуг хостинга;

Недостатки языка PHP:

- безопасность: поскольку он открыт, поэтому все люди могут видеть исходный код, если в исходном коде есть ошибки, он может использоваться людьми для изучения слабости PHP;
- не подходит для больших применений: трудно поддерживать, поскольку он не очень модульный;
- изменение глобальных параметров конфигурации оказывают влияние на базовый синтаксис языка, что создает проблемы с настройками сервера и разворачиванием приложений;
- объекты передаются по значению, что не очень удобно;

Java – является универсальным компьютерным языком программирования, который является параллельным, основанным на классах, объектно-ориентированным и специально разработанным, чтобы иметь как можно меньше зависимостей в реализации.

Так называют не только сам язык, но и платформу для создания приложений уровня предприятий на основе данного языка [17, 18].

Программы на Java транслируются в байт– код, который выполняется виртуальной java–машиной (JVM) — это программа, которая обрабатывает байтовый код и передающей инструкции оборудованию видно интерпретатор,

но с тем отличием, что байтовый код, в отличие от текста, обрабатывается значительно быстрее.

В таблице 3.1 приведены результаты проведенного анализа по перечисленным языкам программирования.

Таблица 3.1 – Сравнительный анализ языков программирования

| Характеристика | C++ | PHP | Java |
|--|-----|-----|------|
| Простота синтаксиса | + | – | + |
| Простота освоения | – | – | + |
| Опыт работы с данным языком | + | + | – |
| Популярность | + | – | + |
| Поддержка языка на большинстве хостингах | – | – | + |
| Множество русифицированной документации | + | – | + |
| Итог | 4 | 1 | 5 |

Таким образом, с помощью сравнительного анализа были выявлены основные преимущества и недостатки рассмотренных языков программирования.

Для написания основной части скриптов было решено использовать язык java.

3.1.2 Выбор системы управления базы данных

Разработка системы массового обслуживания заявок визового центра предполагает наличие сервера баз данных.

Система управления базами данных (СУБД) - это системное программное обеспечение для создания и управления базами данных. СУБД предоставляет пользователям и программистам систематический способ создания, извлечения, обновления и управления данными.

Использование СУБД для хранения и управления данными имеет свои преимущества, но также и свои недостатки. Одним из самых главных преимуществ использования СУБД является то, что она позволяет конечным пользователям и прикладным программистам получать доступ и использовать одни и те же данные при управлении целостностью данных. Данные лучше защищены и поддерживаются, когда их можно использовать с использованием СУБД вместо создания новых итераций тех же данных, которые хранятся в новых файлах для каждого нового приложения.

Центральное хранилище и управление данными в СУБД обеспечивает:

- абстракция и независимость данных;
- безопасность данных;
- механизм блокировки для одновременного доступа;
- эффективный обработчик для сбалансирования потребностей нескольких приложений с использованием одних и тех же данных;
- возможность быстрого восстановления после сбоев и ошибок, включая перезапуск и возможность восстановления;
- надежные возможности целостности данных;
- регистрация и аудит деятельности;
- простой доступ с использованием стандартного интерфейса прикладного программирования (API);
- единообразные процедуры администрирования данных.

На данный момент существует множество СУБД – серверов баз данных. Самыми популярными являются MySQL, OracleExpress и PostgreSQL.

PostgreSQL – свободная объектно– реляционная система управления базами данных [10].

Существует в реализациях для множества UNIX – подобных платформ, включая AIX, различные BSD– системы, HP– UX, IRIX, Linux, Mac OS X, Solaris/OpenSolaris, Tru64, QNX, а также для MicrosoftWindows.

Достоинства PostgreSQL:

- СУБД с открытым исходным кодом, совместимая со стандартами SQL;

- расширяемость, можно программно расширять PostgreSQL с помощью хранимых процедур, например, для расширенной RDBMS;

- объективность, PostgreSQL - это не просто система управления реляционными базами данных, а объективная - с поддержкой и многое другое;

Недостатки PostgreSQL:

- для простых операций с чтением, PostgreSQL может быть чрезмерным и может казаться менее результативным, чем аналоги, например MySQL;

- непопулярность;

- нет встроенной поддержки в plpgsql для создания веб-процедур;

- если все, что вам требуется, это операции быстрого чтения, PostgreSQL не является инструментом для продолжения;

- сложности с поиском хостинга с поддержкой PostgreSQL.

Oracle Application Express (сокращенно APEX, ранее называемая Oracle HTML DB) - это веб-среда разработки программного обеспечения, которая работает в базе данных Oracle. Он полностью поддерживается и поставляется стандартно (без каких-либо дополнительных затрат) со всеми выпусками Oracle Database и, начиная с Oracle 11g, устанавливается по умолчанию как часть базовой базы данных [3].

OracleDatabase не имеет ограничений в применении – от простой базы данных, обслуживающей сайт или небольшую СМО, до огромных и мощных хранилищ данных со встроенными решениями задач класса OLAP или DataMinig, хранящих любые данные: от простых таблиц до документов, видео-файлов, геоинформационных данных и т. п.

Достоинства OracleDatabase:

- администраторы баз данных, знакомые с PL / SQL, могут использовать свой набор навыков для разработки веб-приложений;

- легко создавать макеты с использованием предварительно построенных тем;

- простота развертывания (конечный пользователь открывает URL-адрес для доступа к приложению APEX);

- масштабируемость (может быть развернута на ноутбуках, автономных серверах или установках Oracle RAC);

- обработка и проверка на стороне сервера;

Недостатки OracleDatabase:

- приложения APEX создаются с использованием собственных инструментов Oracle и могут размещаться только в базе данных Oracle, что делает реализатора восприимчивым к блокировке поставщика;

- очень немногие веб-хосты предлагают APEX (Oracle Database) в своем пакете услуг хостинга (большинство из них предлагают PHP + MySQL или ASP + Microsoft SQL Server). В результате приложения APEX ограничены в выборе веб-хостов;

- проекты, требующие, чтобы несколько разработчиков касались одной и той же веб-страницы, должны сообщать о своих намерениях друг с другом. Нет встроенного контроля версий, и все компоненты должны быть отредактированы через веб-интерфейс. Блокировка страниц может помочь защитить от физических зависимостей.

MySQL – является системой управления реляционными базами данных с открытым исходным кодом (РСУБД). Проект разработки MySQL сделал исходный код доступным в соответствии с GNU General Public License, а также в соответствии с различными соглашениями о собственности. MySQL принадлежал и спонсировался одной коммерческой некоммерческой фирмой, шведской компанией MySQL AB, которая теперь принадлежит корпорации Oracle [8]. Для частного использования доступно несколько платных выпусков и предлагают дополнительную функциональность [4].

Преимущества базы данных MySQL:

- повышенная производительность приложений: как известно, после создания данных они компилируются и хранятся в базе данных. Но MySQL реализует хранимые данные, несколько отличающиеся, что помогает повысить производительность приложений. Хранимые данные MySQL компилируются по требованию;

- скорость работы: MySQL Хранимые данные выполняются быстро, потому что сервер MySQL использует некоторые преимущества кеширования;
- простота установки: MySQL не обязывает нас устанавливать дополнительный пакет среды выполнения или устанавливать разрешения для выполнения программы в операционная система;
- масштабируемость;
- безопасность: Хранимые данные MySQL являются безопасными, поскольку администратор базы данных может предоставлять соответствующие разрешения для приложений, которые получают доступ к хранимым процедурам в базе данных, без предоставления каких-либо разрешений для базовых таблиц базы данных.

Недостатки базы данных MySQL:

- использование памяти увеличилось: если мы используем много хранимых данные, использование памяти каждым соединением, использующим эти хранимые данные, значительно увеличится;
- ограничено для сложной бизнес-логики: на самом деле конструкции хранимых данных не предназначены для разработки сложных и гибких бизнес-логик;
- трудная настройка: трудно настраивать хранимые данные. Только несколько систем управления базами данных позволяют отлаживать хранимые данные. К сожалению, MySQL не предоставляет возможности для отладки хранимых данных.

Среди всех СУБД было решено воспользоваться MySQL. Такая СУБД обладает рядом полезных преимуществ, необходимых для разработки. Среди основных преимуществ выделяет скорость работы, богатый функционал и простоту использования [9, 16].

Результаты анализа СУБД представлены в таблице 3.2.

Таблица 3.2 – Сравнительный анализ СУБД

| Характеристика | MySQL | PostgreSQL | Oracle Express |
|--------------------------|-------|------------|----------------|
| Масштабируемость | + | + | – |
| Простота в использовании | + | – | – |
| Стабильность | + | – | – |
| Быстродействие | + | + | + |
| Итог | 4 | 2 | 1 |

Проанализировав различные СУБД, был выбран MySQL для создания модели массового обслуживания. Ее возможности и простота в работе модель достаточно полно отвечают необходимым требованиям.

3.2 Реализация имитационной модели системы массового обслуживания заявок визового центра

Для локальной отладки фалов перед загрузкой на хостинг, был использован локальный сервер на основе Denwer. Он позволяет локально запускать PHP скрипты. Во время разработки было обнаружено, что Denwer не позволяет отправлять электронные сообщения для проверки статуса заявки. На рисунке 3.1 показан фрагмент класса “MyModel.java”. Здесь с помощью программного языка описывается модель системы обслуживания:

```

1 package Z1;
2
3 import java.awt.Container;
4 import SimJava.*;
5 public class MyModel3 extends Sys implements Runnable{
6     Storage IS; // Ссылка на многоканальное устройство
7     Queue IQ; // Ссылка на очередь
8
9     int hy;
10    public MyModel3(int tgl, Container con) // Конструктор класса
11    {
12        super(tgl, // Значение счетчика завершения
13            null, // Ссылка на файл с результатами
14            con); // Ссылка на контейнер для вывода данных
15        Thread th = new Thread(this); // Создание потока запуска модели
16        // Создание модельной среды
17        IQ=new Queue(this, "Очередь заявок"); // Создаем очередь
18        IS=new Storage(this, "Вых. центр", 4); // Создаем многоканальное
19        // устройство
20    }
21    public void run()
22    {
23        initGenerate(1, 0); // Инициализация генератора транзактов - задач
24        initGenerate(8, 60*24*100); // Инициализация генератора транзактов
25        int tgl=0;
26        // Ограничиваем время моделирования транзакт

```

Рисунок 3.1 – Фрагмент программного кода класса “ MyModel.java” выполняющий описание программным языком имитационной модели СМО

На рисунке 3.2 представлен фрагмент программного кода класса “MyFrame.java”, выполняющего описание графического интерфейса реализуемой СМО.

```
1 package Z1;
2
3 import java.awt.Color;
4 import java.awt.Graphics;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import javax.swing.JFrame;
8 import javax.swing.JMenu;
9 import javax.swing.JMenuBar;
10 import javax.swing.JMenuItem;
11
12 public class MyFrame3 extends JFrame {
13     MyModel3 model;
14     JMenuItem m1=new JMenuItem("Запустить модель");
15     public MyFrame3 ()
16     {
17         super("Многоканальная СМО с неограниченной очередью");
18         setSize(800, 600); // Установить размеры
19         setVisible(true);
20         getContentPane().setBackground(new Color(200, 200, 200)); // Задать фон окна
21         // серый
22         JMenu M1=new JMenu("Моделирование");
23         M1.add(m1);
24         JMenuBar MB=new JMenuBar ();
25         MB.add(M1);
26         setJMenuBar (MB);
27         m1.addActionListener(new ActionListener() {
28             public void actionPerformed(ActionEvent e)
```

Рисунок 3.2 – Фрагмент программного кода класса “ MyFrame.java” выполняющего описание графического интерфейса реализуемой системы массового обслуживания

При проектировании системы массового обслуживания необходимо учитывать, что она должна соответствовать тем функциональным требованиям, которые выдвигает компания и ее сотрудники.

На рисунке 3.3 представлен результат реализации java– классов и реализованный графический интерфейс.

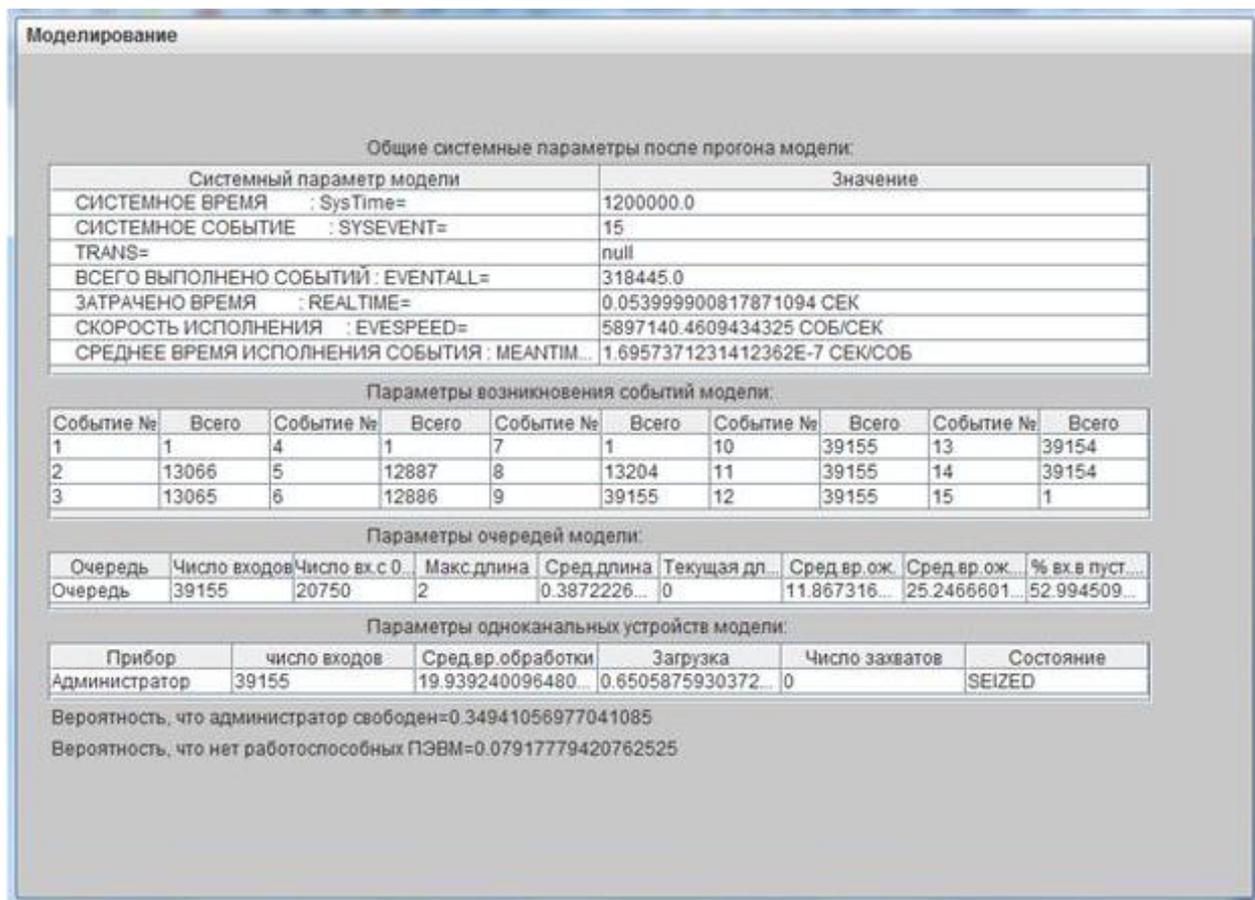


Рисунок 3.3 – результат реализации java – классов и реализованный графический интерфейс.

Из представленного рисунка видно, что полученные результаты с определенной погрешностью соответствуют расчетам на математической модели.

3.3 Проверка адекватности системы массового обслуживания заявок визового центра

Получается, что математически поставленную задачу можно выразить, как минимизацию времени на обработку заявки.

В качестве объекта исследования был взят среднестатистический визовый центр, выполняющий приём заявок на подачу документов для оформления визы. В частности были использованы следующие данные. Интенсивность обслуживания заявок одним окном - $A = 0,716$ заявки в час.

Средняя продолжительность пребывания заявки в очереди на обслуживание $W_s = 2,5$ заявки в час.

В ходе моделирования на ЭВМ были получены следующие результаты.

Для случая, когда $A = 8,3$ чел./час, оптимальное число, рассчитанное по модели, равно 8,8. Вероятность отказа в обслуживании составит 0%, средняя длина очереди – неограничена. Для случая, когда $A = 0,716$ человек в час, оптимальное число каналов, рассчитанное по модели, равно 3. Вероятность отказа в обслуживании составит 0,16, средняя длина очереди – 1,8 человека.

Сравнение важнейших для системы массового обслуживания характеристик представлено в таблице 3.3.

Таблица 3.3 – Сравнение важнейших для системы массового обслуживания характеристик

| Важнейшие характеристики СМО | СМО с ограниченной длиной очереди | СМО с неограниченной длиной очереди |
|---|-----------------------------------|-------------------------------------|
| Параметр потока обслуживаемых заявок | 0,95 | 0,95 |
| Условная пропускная способность | 0,89 | 1 |
| Окончательная пропускная способность(заявки в час) | 0,716 | 0,85 |
| Среднее число заявок на обслуживание в очереди | 1,8 | 8,3 |
| Средняя продолжительность пребывания заявки в очереди на обслуживание | 1,9 | 8,8 |
| Среднее число заявок в очереди(длина очереди) | 1,02 | 7,4 |

Анализ произведенных вычислений показывает, что система в полной мере обрабатывает установленные для реализации требования.

В результате проведенного эксперимента можно сделать выводы:

Использование компонентов концепции систем массового обслуживания в комбинации с экономическими аспектами способны предоставить возможность для уменьшения расходов визового центра посредством

оптимизации выбора одноканальной системы массового обслуживания с неограниченной очередью.

Проверка адекватности предложенной модели на данных среднестатистического визового центра обнаружила, что этот аспект может быть использован в практике оперативного управления. А это значит, что итоги экспериментов на модели можно считать правильными при подлинности начальных данных.

ЗАКЛЮЧЕНИЕ

Настоящая выпускная квалификационная работа посвящена теме выбора модели системы массового обслуживания заявок визового центра.

Выпускная квалификационная работа выполнена на основе материалов полученных в интернете на примере среднестатистического визового центра.

Путем решения поставленной задачи и произведенного анализа извлекли следующие результаты:

- совершен анализ предметной области;
- произведено сравнение двух одноканальных моделей систем массового обслуживания и выбор более эффективной;
- реализована модель системы массового обслуживания.

Для успешного завершения цели была произведена разработка подытоженного и досканального алгоритма моделирования работы СМО. В ходе некоторых осуществлений работы СМО закрепляются предварительные итоги работы системы. Используя эти данные были построены графические зависимости, которые помогли исследовать СМО. Был сделан анализ характеристик работы разрабатываемой СМО. По итогу работы были выявлены достоинства и недостатки, а так же сделаны выводы по улучшению качества работы разработанной СМО.

Выбранная модель системы массового обслуживания прошла успешное тестирование.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Кельтон В., Лоу А. Имитационное среднее моделирование. Классика CS. 3-е изд. СПб.: Питер; Киев: Изд. группа BHV, 2004. 847 с.
2. Власов С.А., Девятков В.В., Кобелев Н.Б. Имитационные исследования: от классических технологий до облачных вычислений // Пятая (юбилейная) Всероссийская науч.– практич. конф. по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» ИММОД– 2011(Санкт - Петербург, 2011 г.): сб. докл. Т. 1. С. 42– 50.
3. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. М.: ДМК Пресс, 2004. 320 с.
4. Марков А.А. Моделирование информационно– вычислительных процессов: учеб. пособие для вузов. М.: Изд– во МГТУ им. Н.Э. Баумана, 1999. 360 с.
5. Медведев Н.В., Быков А.Ю., Гришин Г.А. Имитационное моделирование систем массового обслуживания с использованием меж– платформенной библиотеки функции языка Си++ // Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2005. № 4. С. 85– 93.
6. Журавлев А.М., Медведев Н.В., Быков А.Ю. Использование межплатформенной библиотеки функции языка Си++ для реализации имитационных моделей типовых систем массового обслуживания // Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2008. № 4. С. 3– 15.
7. Ноутон П., Шилдт Г. Java 2. Наиболее полное руководство. СПб.: BHV– Санкт– Петербург, 2007. 1072 с.
8. Клейнрок Л. Теория массового обслуживания. М.: Машиностроение, 1979 432 с.
9. Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web– мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд. – СПб.: Изд– во «БХВ– Петербург», 2015. – 766с.

10.Ригс, С. Администрирование PostgreSQL 9. Книга рецептов / С. Ригс, Х. Кроссинг. – ДМК Пресс, 2013. – 368с.

11.Скляр, Д. PHP. Рецепты программирования/ Д. Скляр, А. Трахтенберг. – 3– е изд. – СПб.: Изд– во «Питер», 2015. – 784с.

12.Кирпичников А.П., Титовцев А.С. Открытая одноканальная система массового обслуживания с отказами и неорганической очередью // Вестник Казанского государственного технологического университета – Казань: Изд– во Казан. гос. технол. ун– та, 2006 – № 4 – С. 78 – 85.

13. Валеев И.Н., Кирпичников А.П. Многоканальные системы массового обслуживания с отказами // Вестник Казанского государственного технологического университета – Казань: Изд– во Казан. гос. технол. ун– та, 2006 – № 4 – С. 66– 71.

14. Гильмутдинов Р.Ф., Кирпичников А.П. Замкнутые модели систем массового обслуживания с ограничениями // Вестник Казанского государственного технологического университета – Казань: Изд– во Казан. гос. технол. ун– та, 2006 – № 4 – С. 220– 224.

15. Кирпичников А.П. Прикладная теория массового обслуживания. – Казань: Изд– во КГУ, 2008. 112 с.

Литература на иностранном языке

16.Modeling digital library management system based on the methods of queuing theory, Ivo Adan and Jacques Resing, Introduction to Discrete Event Systems, Springer, 2007 S.M. Ross, Stochastic Process, John Wiley & Sons, URL: http://www.iitg.ac.in/nselvaraju/ma597_2007/Assignments/06212310–_assii.pdf

17. Time Characteristics of Queuing System with Renovation and Reservice C. De Nicola, R. Manzo, R. Razumchik, Z. I., URL: https://books.google.ru/books?id=xM–zCwAAQBAJ&pg=PA377&lpg=PA377&dq=Time+Characteristics+of+Queuing+System+with+Renovation+and+Reservice.&source=bl&ots=lckjV9L6RO&sig=3d6_hS45qBdDIYUSZ9nG3–_6–_

18. Testing of Software for Calculating a Multichannel Queuing System with “Cooling” and E2– approximation , 3085 зН., C. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, Springer, 2007S.M. Ross, Stochastic Process, John Wiley & Sons, URL: <https://doi.org/10.1145/2745844.2745882>

19. Modeling digital library management system based storage on the methods of queuing theory – C. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, Springer, 2007 S.M . Ross, Stochastic Process, University of Texas at El Paso, URL: <https://www.scss.tcd.ie/Brett.Houlding/Domain.sites2/sslides9.pdf>

20. Queueing Systems, NLP, and Representations, 2017, Bo Jiang, Yongge Tian, and Xuan Zhang, URL: <http://www.win.tue.nl/~iadan/queueing.pdf>

Приложение А

```
import java.awt.Container;
import java.awt.Label;
import SimJava.*;
public class MyModel5 extends Syst implements Runnable {
    Facility IF;
    Queue IQ;
    public MyModel5(int tg1, Container con)
    {
        super(tg1,
                null,
                con);
        th=new Thread(this);
        IQ=new Queue(this, "Очередь");
        IF=new Facility(this, "Администратор");
    }
    public void run()
    {
        double T_0=0, T_M=0;
        double t1=0, t2=0;
        for(int i=0; i<3; i++) initGenerate(1+3*i, 0);
        initGenerate(15, 60*20000);
        while(tg1!=0)
        {
            plan();
            switch(sysEvent)
            {
                case 1:    trans.pI[0]=2; break;
                case 2: advance(v1.randExp(1./60, false)); break;
            }
        }
    }
}
```

```

case 3: next(9); break;
case 4: trans.pI[0]=5; break;
case 5: advance(v2.randExp(1./60, false)); break;
case 6: next(9); break;
case 7: trans.pI[0]=8; break;
case 8: advance(v3.randExp(1./60, false)); break;
case 9:
    if (IQ.IQ==0 && IF.status==Syst.Status.FREE) T_0+=(sysTime-t1);
        IQ.queue(1);
    if (IQ.IQ==2) t2=sysTime;
        break;
case 10: IF.seize(); break;
case 11: if (IQ.IQ==2) T_M+=sysTime-t2;
        IQ.depart(1); break;
case 12: advance(v4.randExp(0.05, false)); break;
case 13: if (IQ.IQ==0) t1=sysTime;
        IF.release();
        break;
case 14: next(trans.pI[0]); break;
case 15:
    if (IQ.IQ==0 && IF.status==Syst.Status.FREE) T_0+=(sysTime-t1);
    if (IQ.IQ==2) T_M+=sysTime-t2;
    terminate(1);
}
}
printAllScreen();
String str1="Вероятность, что администратор свободен="+T_0/sysTime;
String str2="Вероятность, что нет работоспособных
ПЭВМ="+T_M/sysTime;
Label l1=new Label(str1);

```

```
    Label l2=new Label(str2);  
    l1.setBounds(hx, hy, 400, h); hy+=h;  
    l2.setBounds(hx, hy, 400, h); hy+=h;  
    con.add(l1);  
    con.add(l2);  
    }  
}
```

Приложение Б

```
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
public class MyFrame5 extends JFrame {
    MyModel5 model;
    JMenuItem m1=new JMenuItem("Запустить модель");
    public MyFrame5()
    {
        super("Замкнутая СМО с m источниками заявок");
        setSize(800, 600);
        setVisible(true);
        getContentPane().setBackground(new Color(200, 200, 200));
        JMenu M1=new JMenu("Моделирование");
        M1.add(m1);
        JMenuBar MB=new JMenuBar();
        MB.add(M1);
        setJMenuBar(MB);
        m1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                model=new MyModel5(1, MyFrame5.this.getContentPane());
                model.hy=50;
                model.th.start();
                try { model.th.join(); } catch(Exception e2) {}
            }
        });
    }
}
```

```
        }
    }
    );
}
public static void main(String[] args) {
    MyFrame5 F=new MyFrame5();
}
}
```