

Министерство науки и высшего образования Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого
Высшая школа программной инженерии

Отчёт по курсовой работе

"МЕТОДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ"

Выполнил
студент гр. 5130904/30103

Зайцев В.А.

Руководитель

Дробинцев Д.Ф.

Санкт-Петербург

2025

Оглавление

Постановка задачи	3
Исходные данные	4
Формализованная схема и описание СМО	4
Пример временной диаграммы функционирования системы	6
Обобщённая блок-схема	7
Пример технической системы, удовлетворяющей формализованному описанию	8
Ограничения и требуемые характеристики	8
Модульная структура	10
Описание работы программы	11
Результаты работы имитационной модели	15
Определение количества реализаций	15
Анализ результатов, выводы и рекомендации по выбору конфигурации системы.	16
Вывод	24
Список литературы	25

Постановка задачи

Целью курсовой работы является создание модели вычислительной системы (ВС) или ее части на некотором уровне детализации, описывающей и имитирующей ее структуру и функциональность.

Каждый реальный объект (реальная ВС) обладает бесконечной сложностью, множеством характеристик, внутренних и внешних связей. Модель есть приближенное описание объекта с целью получения требуемых результатов с определенной точностью и достоверностью.

При необходимости исследования поведенческих характеристик ВС в процессе исследования выгодно использовать не сам объект, а его модель. Степень приближения модели к описываемому объекту может быть различной и зависит от требований задачи.

Существуют различные типы моделей:

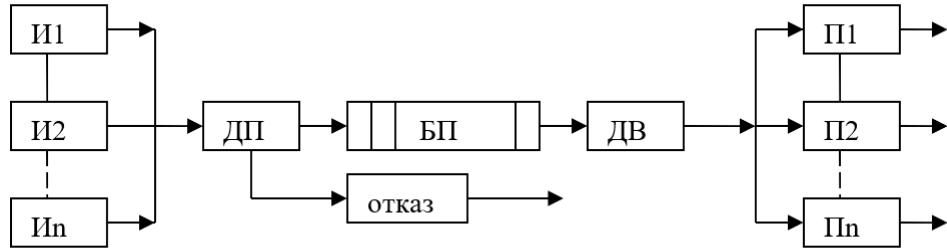
- Аналитические (математические) модели
- Аналоговые модели
- Физические модели
- Имитационные модели

Последний тип моделей является предметом нашего изучения.

Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО), с характерной для СМО терминологией: источник, буфер, прибор, диспетчер, заявка (требование).

Исходные данные

Формализованная схема и описание СМО

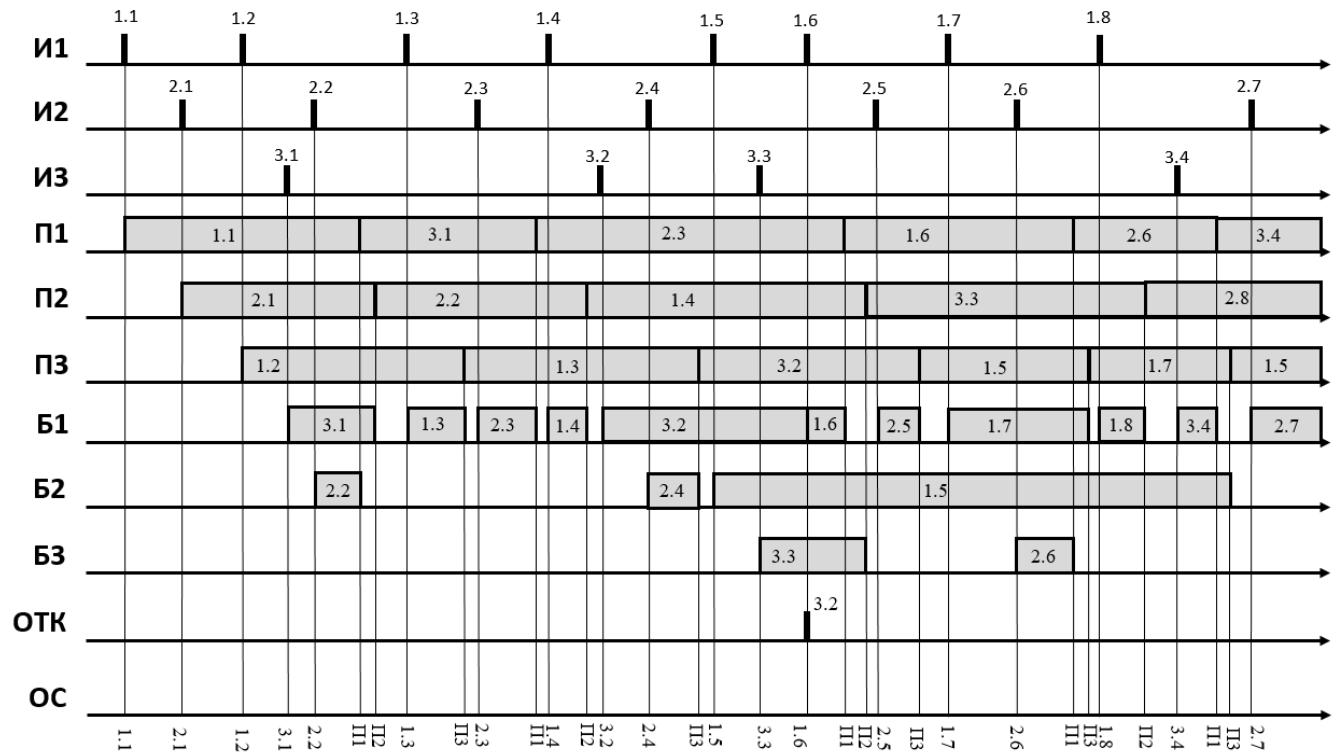


11	ИБ	ИЗ1	ПЗ2	Д10З3	Д10О3	Д2П1	Д2Б2	ОР1	ОД2
----	----	-----	-----	-------	-------	------	------	-----	-----

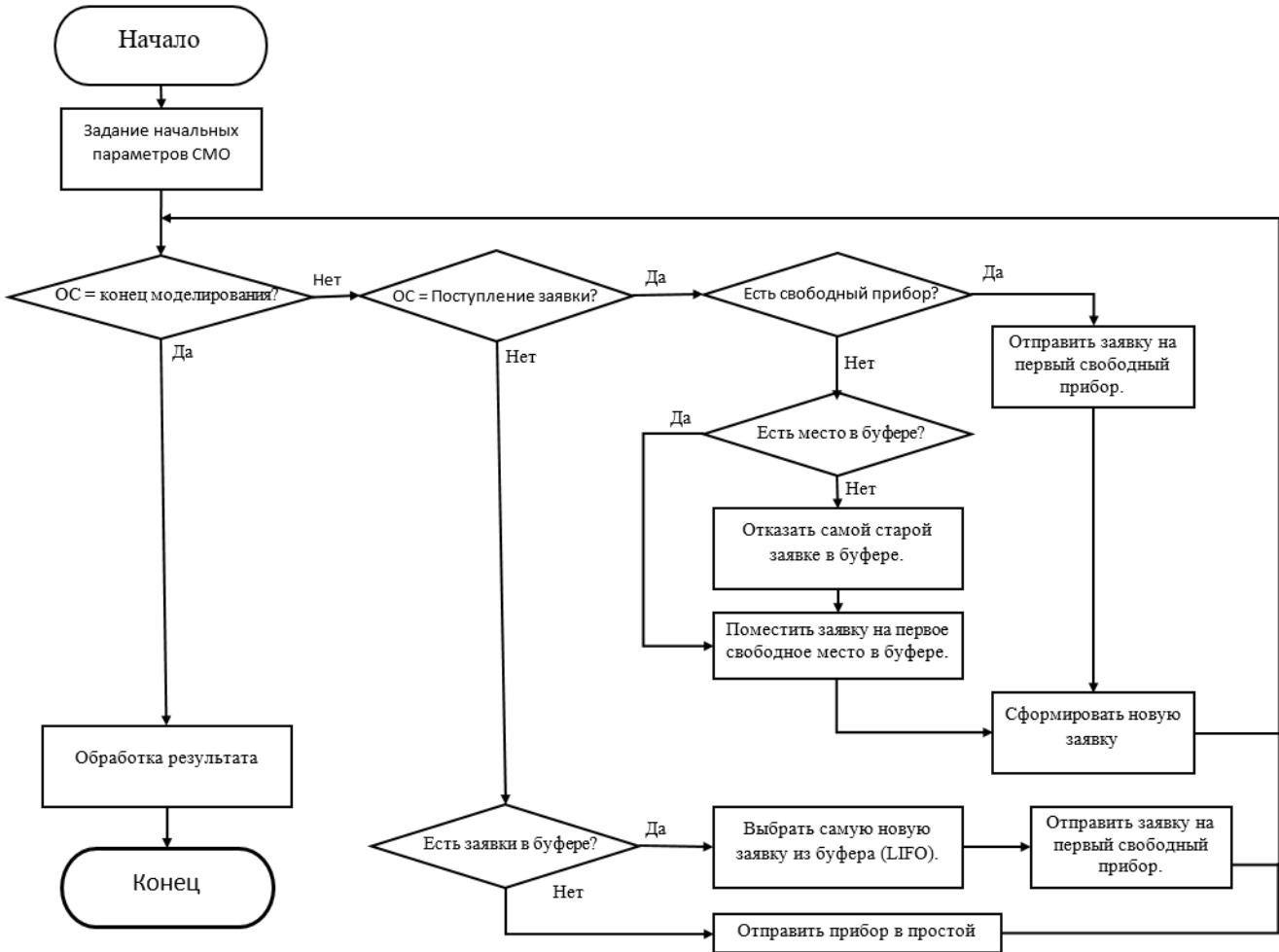
- Параметры элементов модели.
 - Источники:
 - * ИБ — бесконечный источник;
 - * ИЗ1 — пуссоновский закон распределения заявок;
 - Приборы:
 - * ПЗ2 — равномерный закон распределения времени обслуживания;
- Описание дисциплин постановки и выбора.
 - Буферизации:
 - * Д1ОЗ3 — постановка заявки в очередь на первое от начала свободное место;
 - Отказа:
 - * Д1ОО3 — заявка, раньше других вставшая в буфер получает отказ, уходит из системы и на её место встает пришедшая заявка;
 - Выбора заявки на обслуживание:
 - * Д2Б1 — LIFO (последним пришел — первым обслужен);
 - Выбора обслуживающего прибора:
 - * Д2П1 — приоритет по номеру прибора;
- Виды отображения результатов работы программной модели.

- Динамическое отражение результатов (пошаговый режим):
 - * ОД2 – формализованная схема модели, текущее состояние;
- Отражение результатов после сбора статистики (автоматический режим):
 - * ОР1 – сводная таблица результатов;

Пример временной диаграммы функционирования системы



Обобщённая блок-схема



Пример технической системы, удовлетворяющей формализованному описанию

Техническая система	Система видеонаблюдения
Источники	Камеры видеонаблюдения, которые отправляют кадры только при обнаружении движения в кадре. Количество камер может варьироваться от 20 до 50 устройств.
Приборы	Приборами являются видеокарты, которые анализируют кадры (поиск лиц, трекинг объектов между несколькими камерами).
Буфер	Выделенная область оперативной памяти, в которую записываются поступающие кадры.
Дисциплина постановки в буфер	Постановка на первое свободное место в буфере
Дисциплина выбора из буфера	LIFO - последние кадры обрабатываются первыми, так как они наиболее актуальны для текущего состояния системы.
Дисциплина отказа	Самые старые кадры в видеопамяти перезаписываются при нехватке места.
Дисциплина постановки на обслуживание	Выбирается первая свободная видеокарта.

Ограничения и требуемые характеристики

Количество камер	От 20 до 50
Размер заявки	1Мб
Размер буфера	от 25Мб до 50Мб
Количество видеокарт	от 1 до 10
Среднее время между кадрами	$\lambda = 1\text{с}$
Время обработки кадров	350-500 мс 150-220 мс 50-90 мс 40-60 мс

Вероятность отказа (пропуска кадров) должна составлять не более 10%.

Загрузка приборов более 90% при максимально возможном числе камер.

Стоимость компонентов системы

Наименование	Время обслуживания заявки	Цена
NVIDIA GeForce GTX 1650	350-500 мс	18 000 р.
NVIDIA GeForce RTX 3050	150-220 мс	40 000 р.
NVIDIA GeForce RTX 4080	50-90 мс	120 000 р.
NVIDIA RTX A4000	40-60 мс	150 000 р.

Поскольку число камер и размер кадров таковы, что память буфера значительно меньше оперативной памяти стандартного компьютера (современный ПК имеет от 4 Гб ОЗУ) на несколько порядков, то можем считать, что дополнительных затрат на память не будет.

Нам требуется найти самую дешёвую конфигурацию системы, которая будет удовлетворять всем требованиям при максимальной нагрузке т.е. при максимальном числе подключенных камер равном 50. Мы можем варьировать размер буфера и число камер, а также менять различные виды видеокарт.

Модульная структура

Разработка производилась в среде VisualStudio на языке C++ и среде VS Code на языке Python с использованием графической библиотеки Qt. Приложение является объектно-ориентированным и содержит следующие классы:

- Request — класс описывающий т пакет данных от датчика транспортного потока
- Event — класс описывающий тип события и время его наступления и заявку, с которой это событие связано
- RequestSource — класс источника, который создаёт объекты заявок Request
- Handler — прибора, который обрабатывает заявки Request
- Buffer — класс буфера, хранящий некоторое число заявок Request
- UniformGenerator — класс генератора случайных чисел с равномерным распределением и в интервале $[a, b]$
- PoissonGenerator — класс генератора случайных чисел с экспоненциальным распределением с параметром λ
- QueueingSystem — класс реализующий интерфейс взаимодействия источников, приборов и буфера, а также сбор статистики
- Набор классов MainWindow, Tab1Widget, Tab2Widget и другие, реализующие графический интерфейс приложения

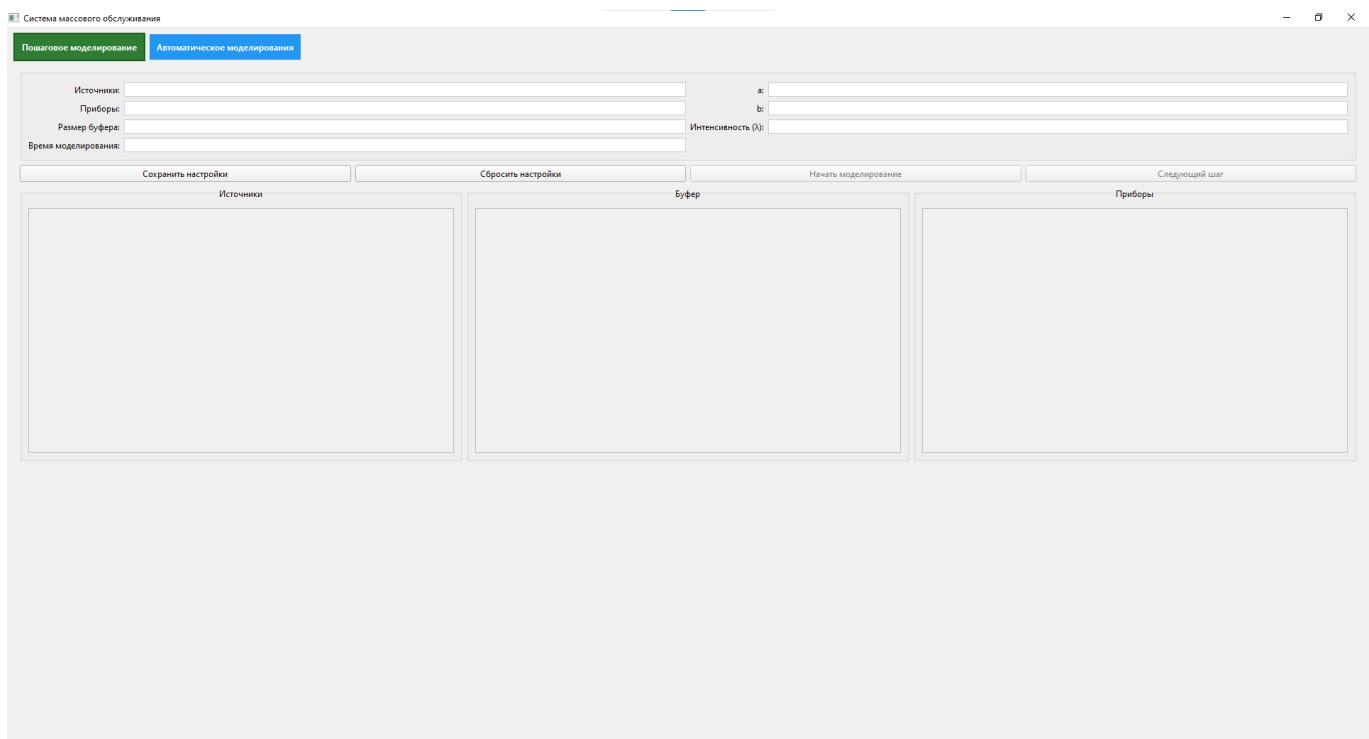
Описание работы программы

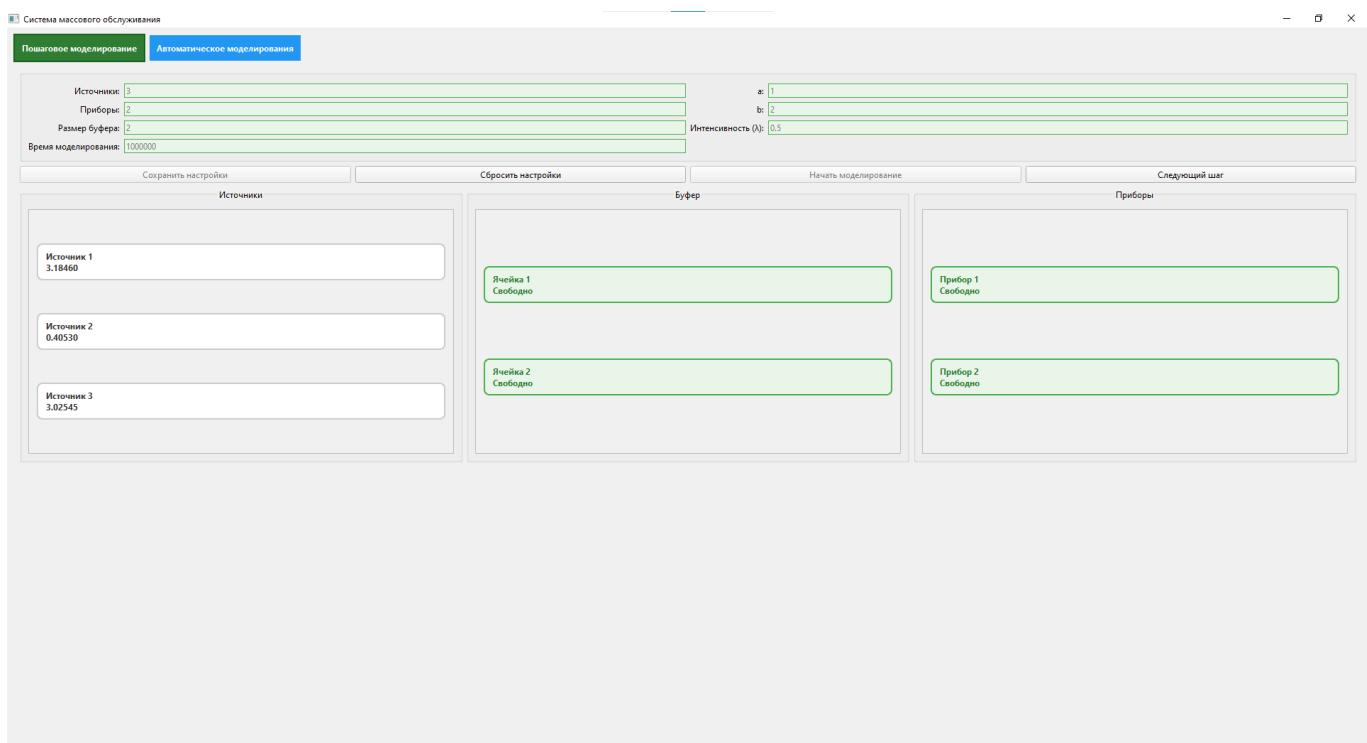
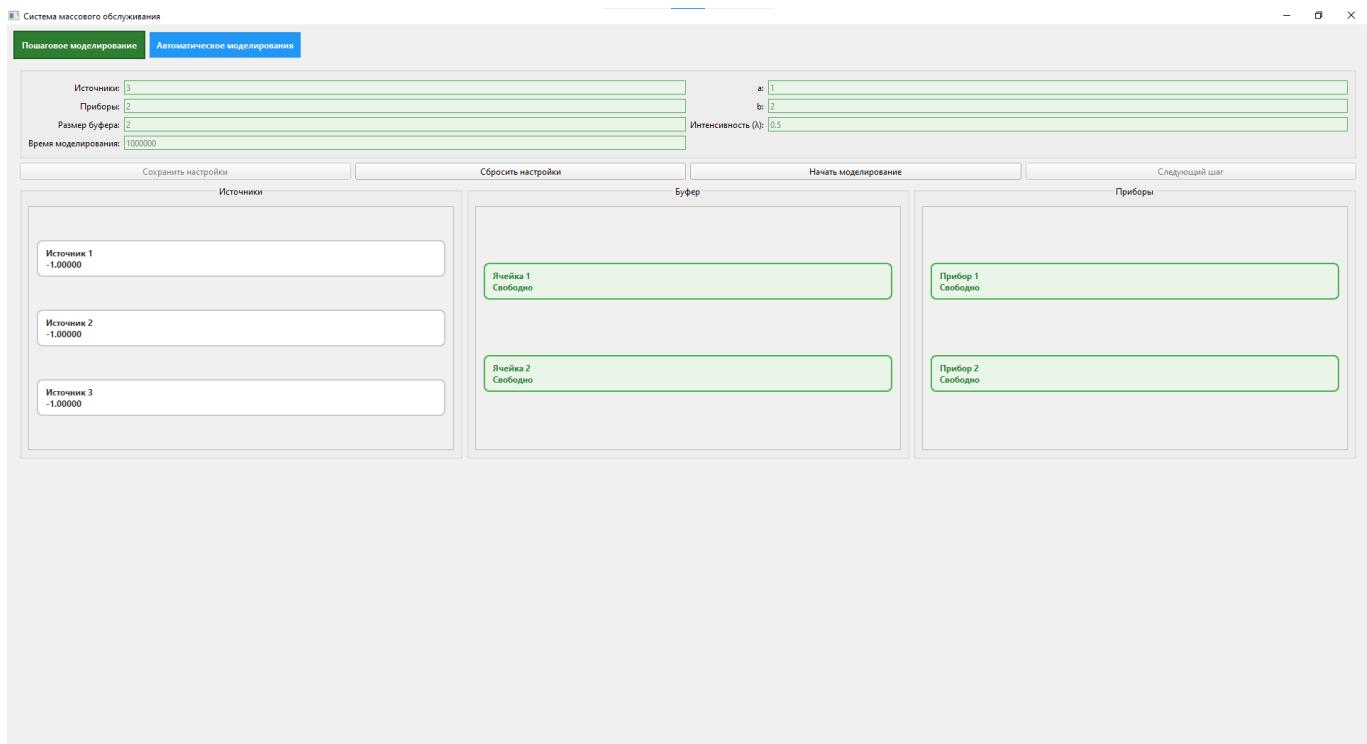
Графический интерфейс приложения содержит два основных окна: для пошагового моделирования и для моделирования в автоматическом режиме.

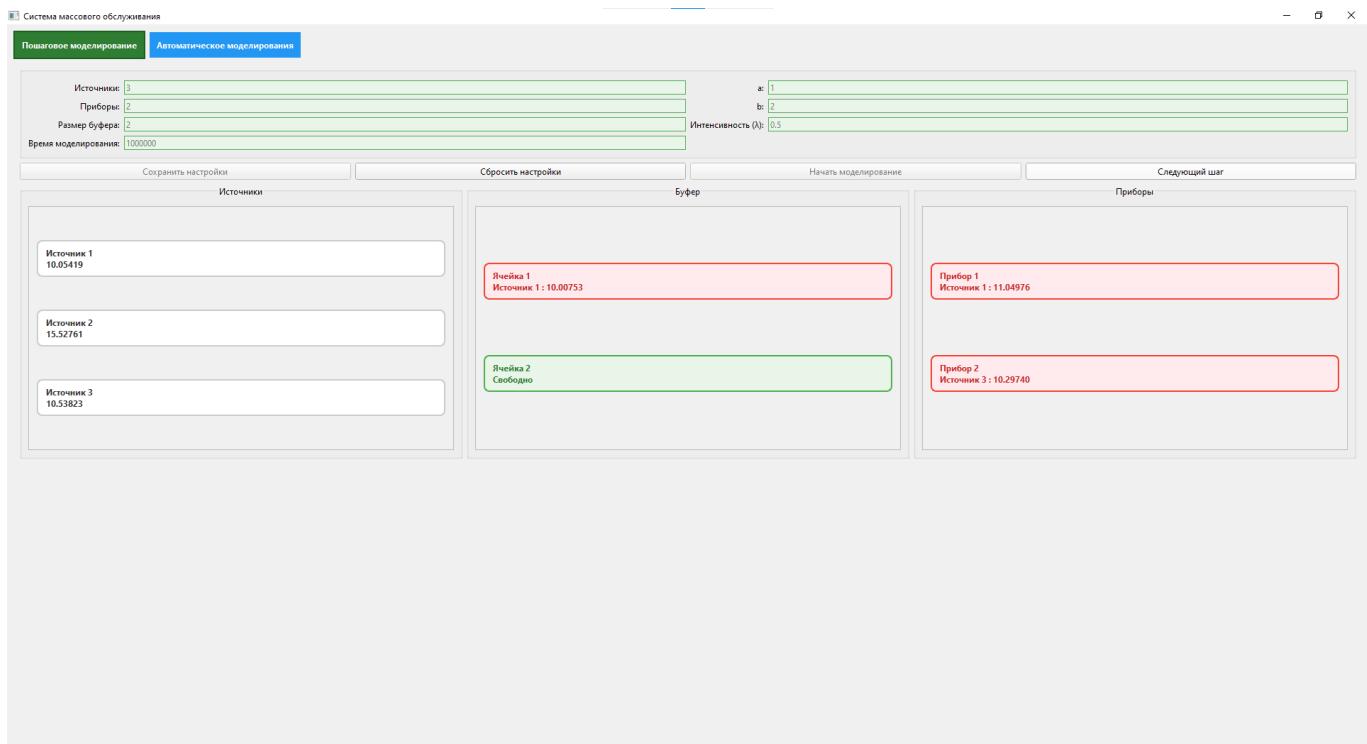
Оба окна содержат поля ввода всех основных параметров системы (число источников, число приборов, размер буфера, время моделирования, параметры времени обработки на приборе и интенсивности источников), а также набор кнопок для сохранения и сброса настроек, а также управления ходом моделирования.

В нижней части окна пошагового моделирования содержатся блоки элементов системы (блок источников, блок ячеек буфера и блок приборов), которые отражают состояние системы на момент очередного события.

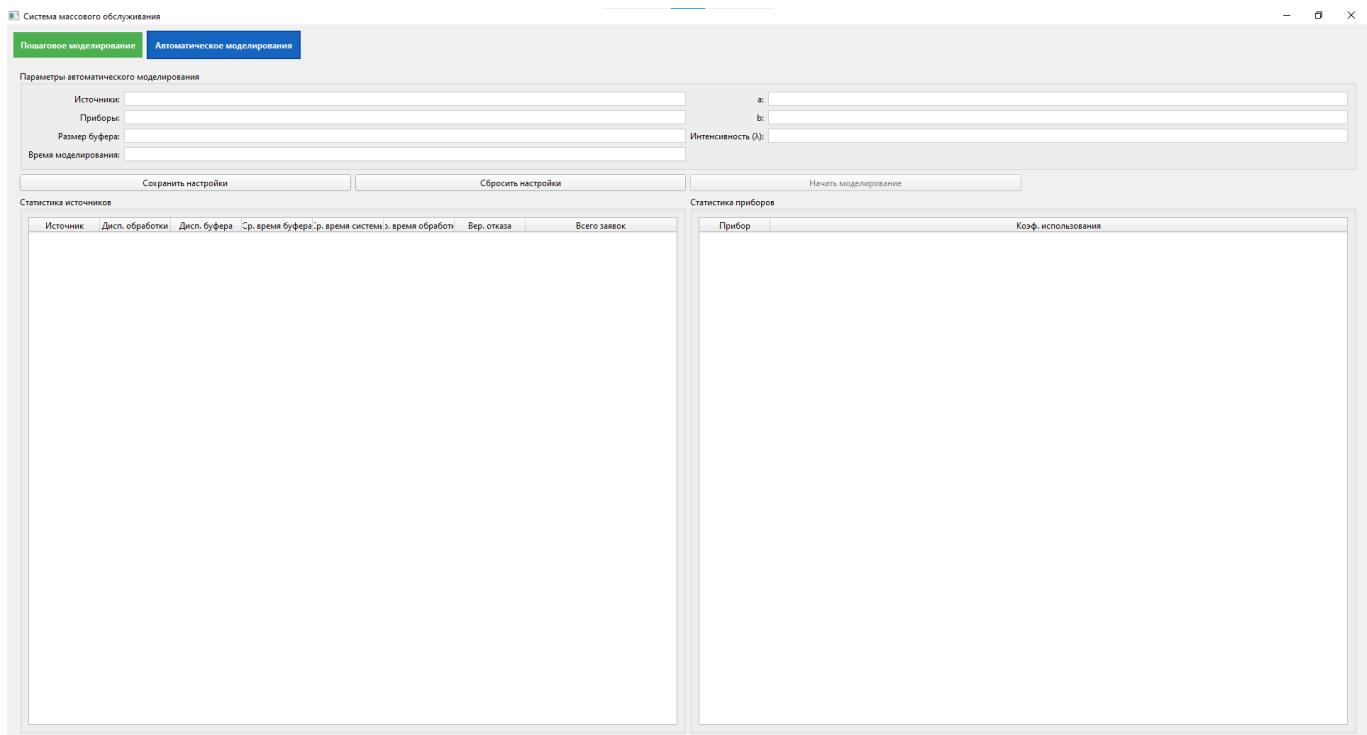
- Источники содержат информацию о времени формирования следующей заявки.
- Ячейки буфера содержат информацию о времени формирования хранимой заявки и номере прибора, от которого эта заявка поступила. Если ячейка буфера не содержит заявку, то указывается, что ячейка пуста.
- Приборы содержат информацию о времени своего освобождения. Если прибор не обрабатывает заявку, то указывается, что он свободен.







В нижней части окна автоматического моделирования содержатся таблицы, содержащие расширенную статистику по отдельным источникам и приборам. Также над ними отдельно показывается общая статистика по числу заявок, загруженности приборов и вероятности отказов заявок.



Система массового обслуживания

Параметры автоматического моделирования

Источник:	50	a:	0.15
Приборы:	9	b:	0.22
Размер буфера:	5	Интенсивность (λ):	1
Время моделирования:	100		

Сохранить настройки Сбросить настройки Начать моделирование

Средняя загруженность: 0.9336
Средняя вероятность отказа: 0.0963
Общее число заявок: 5029

Статистика источников

Источник	Дисп. обработки	Дисп. буфера	Ср. время буфера/ср. время системы	с. время обработки	Вер. отказа	Всего заявок
1 Источник 1	0.0004	0.0007	0.0176	0.2015	0.0693	101
2 Источник 2	0.0004	0.0021	0.0249	0.2130	0.1681	96
3 Источник 3	0.0004	0.0018	0.0185	0.2022	0.1836	98
4 Источник 4	0.0004	0.0028	0.0325	0.2192	0.1867	101
5 Источник 5	0.0003	0.0012	0.0196	0.2020	0.1824	101
6 Источник 6	0.0004	0.0020	0.0276	0.2123	0.1847	96
7 Источник 7	0.0004	0.0007	0.0179	0.2076	0.1897	101
8 Источник 8	0.0004	0.0033	0.0325	0.2196	0.1871	101
9 Источник 9	0.0004	0.0020	0.0226	0.2079	0.1853	106
10 Источник 10	0.0004	0.0032	0.0340	0.2212	0.1872	110
11 Источник 11	0.0004	0.0027	0.0333	0.2166	0.1833	94
12 Источник 12	0.0004	0.0024	0.0285	0.2100	0.1815	109
13 Источник 13	0.0004	0.0024	0.0307	0.2143	0.1836	100
14 Источник 14	0.0004	0.0016	0.0272	0.2139	0.1866	93
15 Источник 15	0.0003	0.0025	0.0301	0.2160	0.1859	122
16 Источник 16	0.0004	0.0046	0.0385	0.2240	0.1655	101
17 Источник 17	0.0004	0.0016	0.0194	0.2051	0.1857	107
18 Источник 18	0.0003	0.0011	0.0224	0.2076	0.1852	97
19 Источник 19	0.0004	0.0017	0.0194	0.2034	0.1840	106
20 Источник 20	0.0004	0.0017	0.0217	0.2117	0.1874	100

Статистика приборов

Прибор	Коэф. использования
1 Прибор 1	0.9254
2 Прибор 2	0.9128
3 Прибор 3	0.8958
4 Прибор 4	0.8793
5 Прибор 5	0.8628
6 Прибор 6	0.9254
7 Прибор 7	0.9128
8 Прибор 8	0.8958
9 Прибор 9	0.8793

Info

Автоматическое моделирование завершено.

OK

Результаты работы имитационной модели

Определение количества реализаций

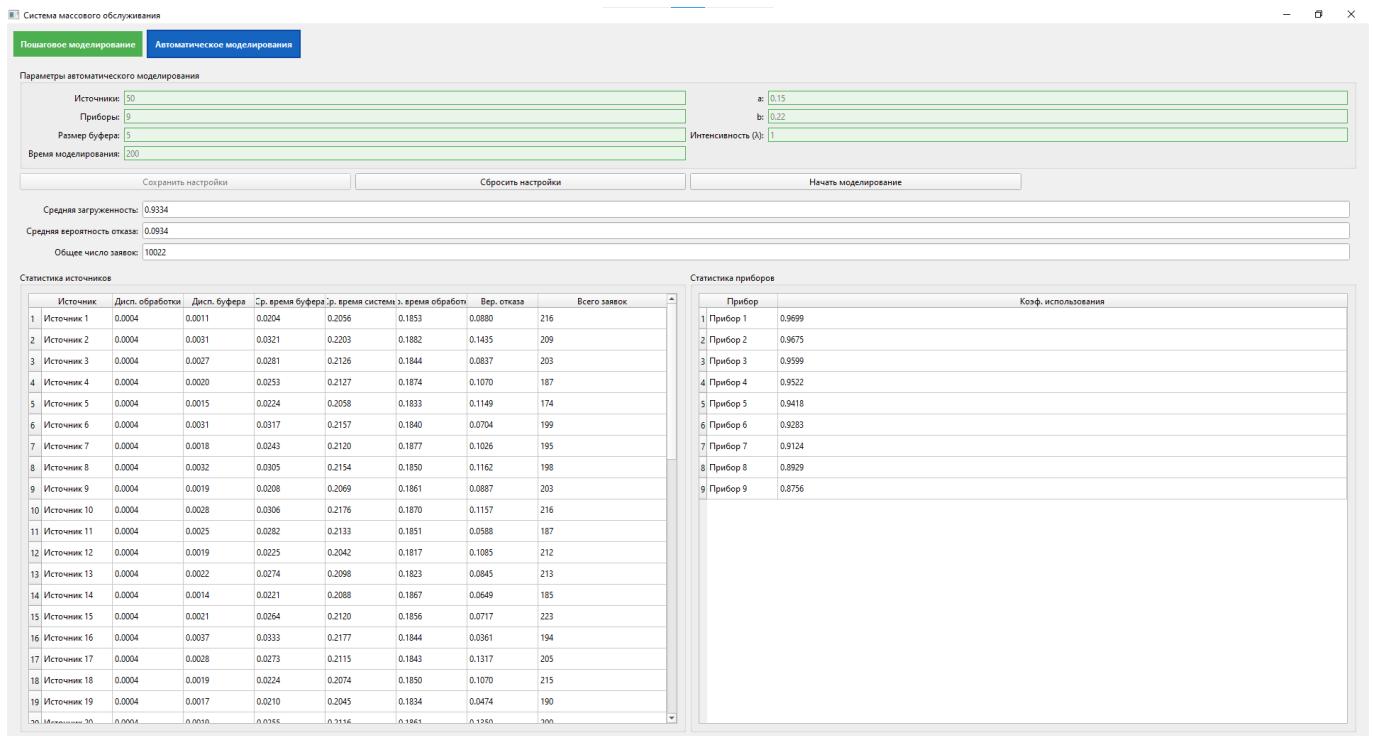
Количество реализаций, необходимое для получения нужной точности при заданной доверительной вероятности, можно оценивать по формуле:

$$N = \frac{t_{\alpha}^2(1-p)}{p\delta^2} = 2429,$$

где:

- p — вероятность отказа в обслуживании заявки;
- $t_{\alpha} = 1.643$ для $\alpha = 0.9$ (доверительная вероятность 90%);
- $\delta = 0.1$ — относительная точность (10%).

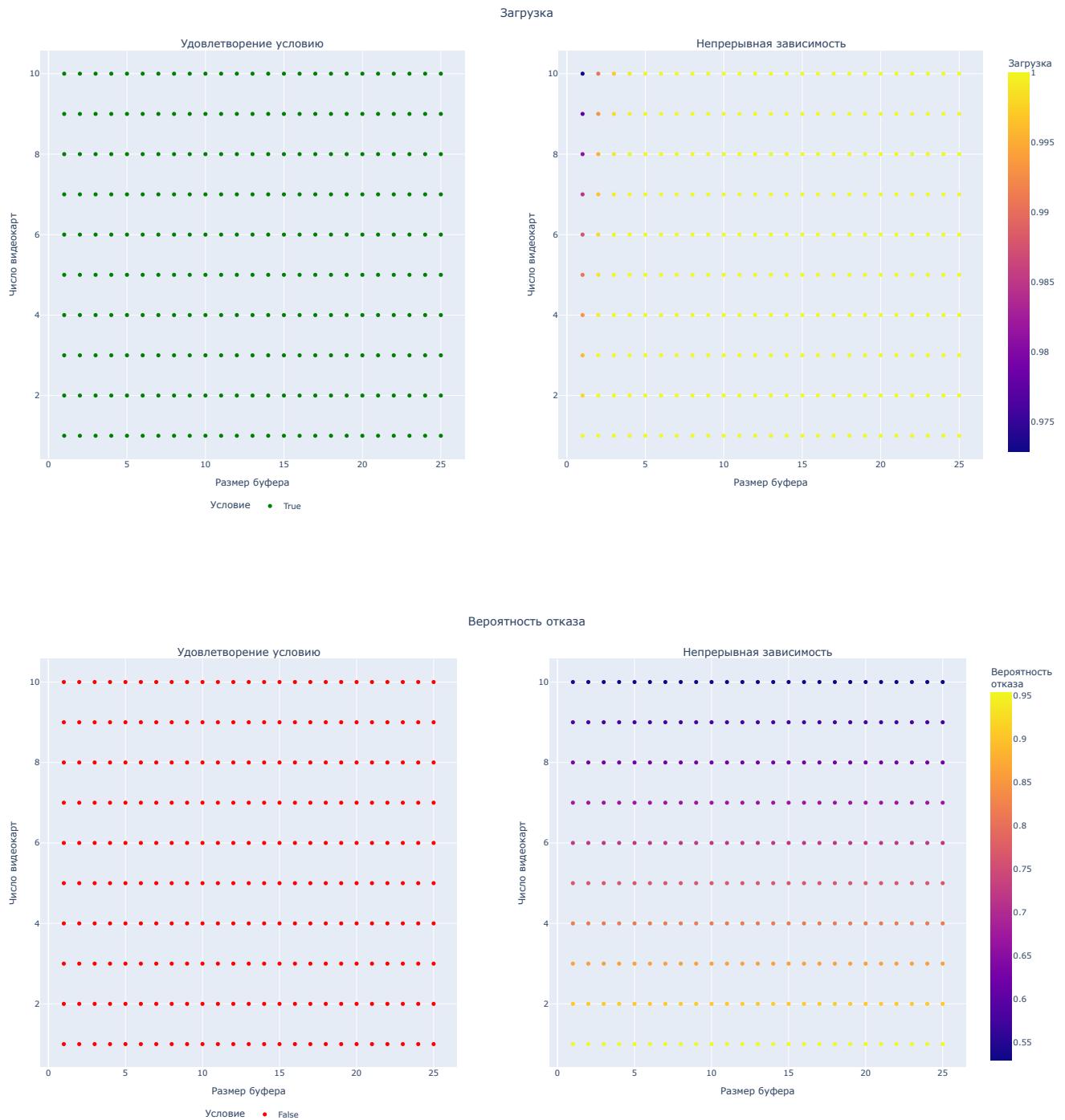
По результатам работы программы получено, что в большинстве случаев хватает около 10.000 заявок для достижения нужной точности. Однако, в зависимости от конфигурации, может потребоваться ощутимо большее количество заявок (до 25.000).



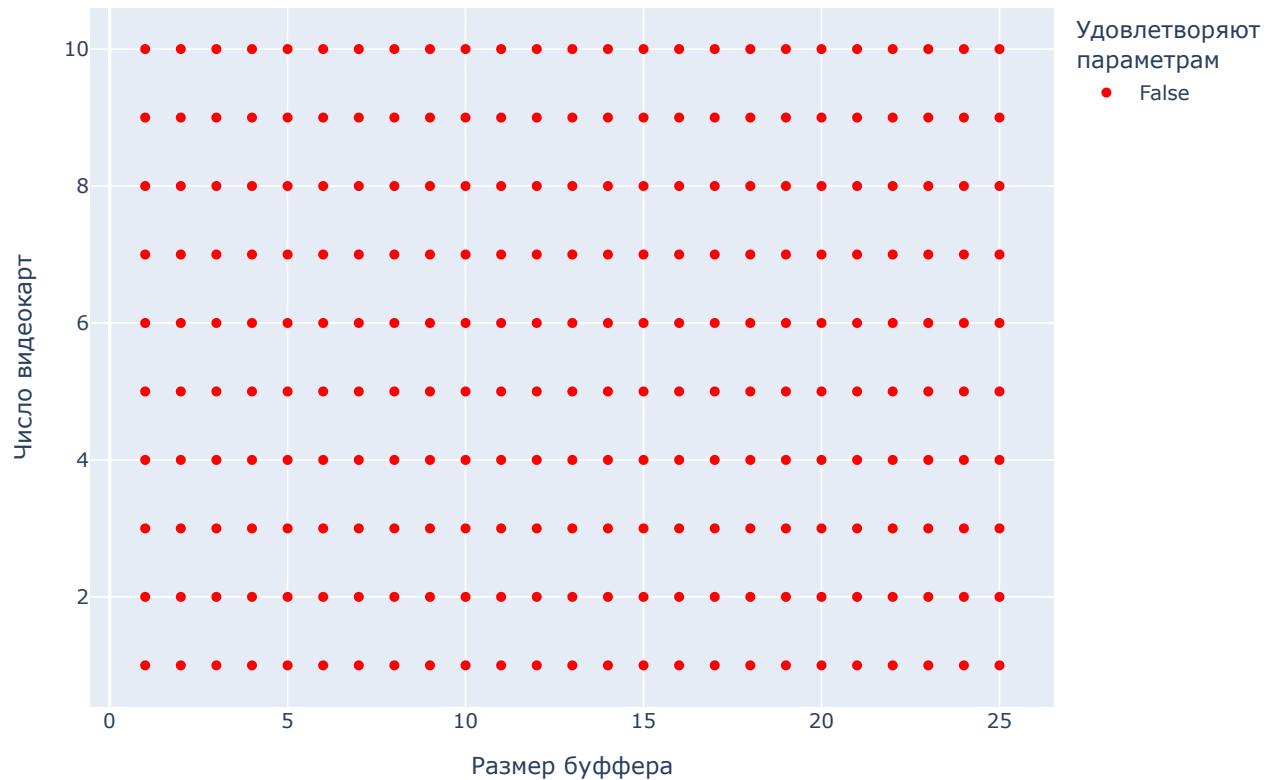
Анализ результатов, выводы и рекомендации по выбору конфигурации системы.

NVIDIA GeForce GTX 1650

В случае с NVIDIA GeForce GTX 1650 мы можем отметить, что на всех конфигурациях она выдаёт максимально возможную или близкую к максимальной загрузку всех видеокарт, но при этом ни одна конфигурация не позволяет снизить вероятность отказа хотя бы до 50% не говоря уже о целевых 10%.

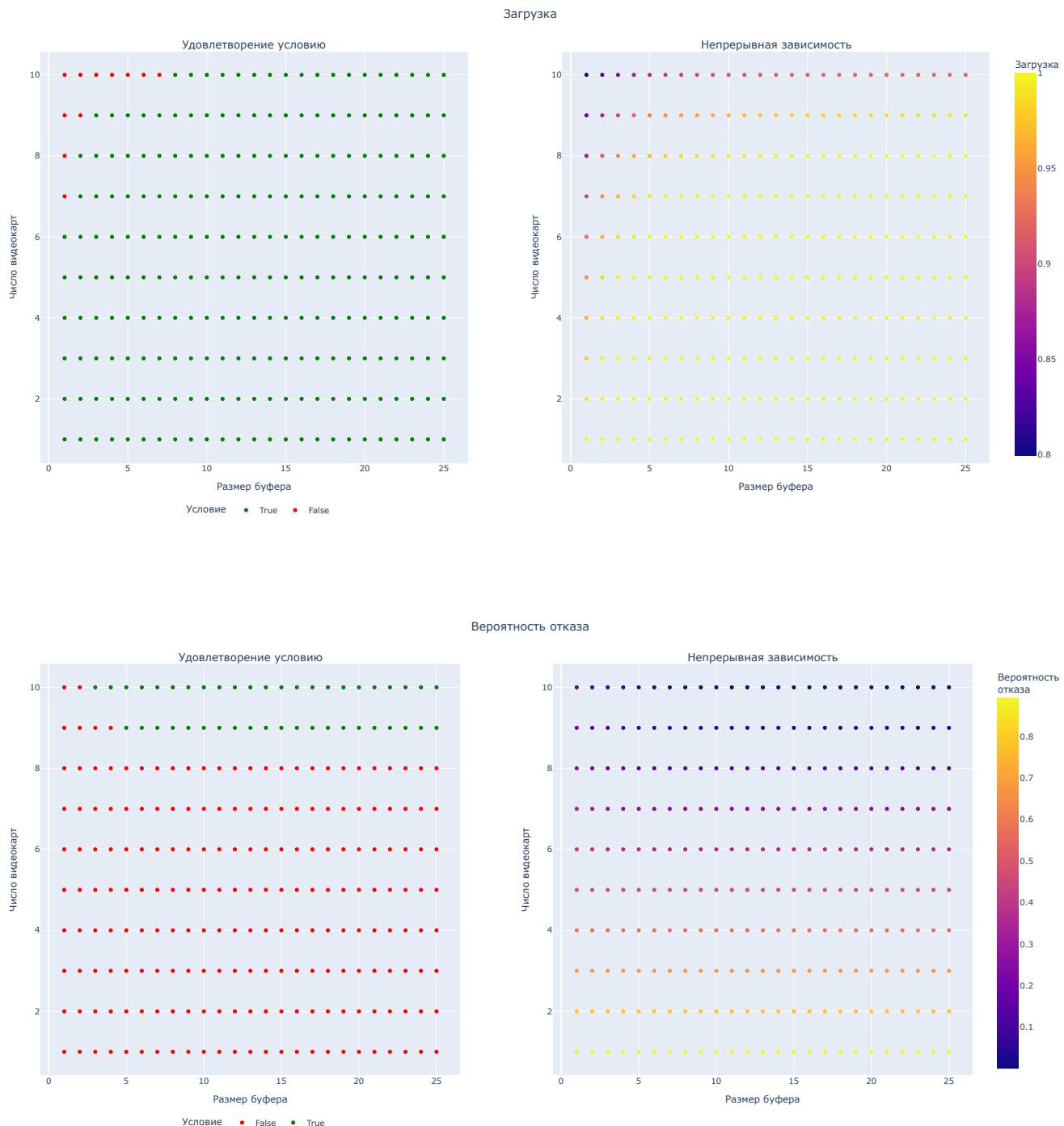


Соответствие всем условиям

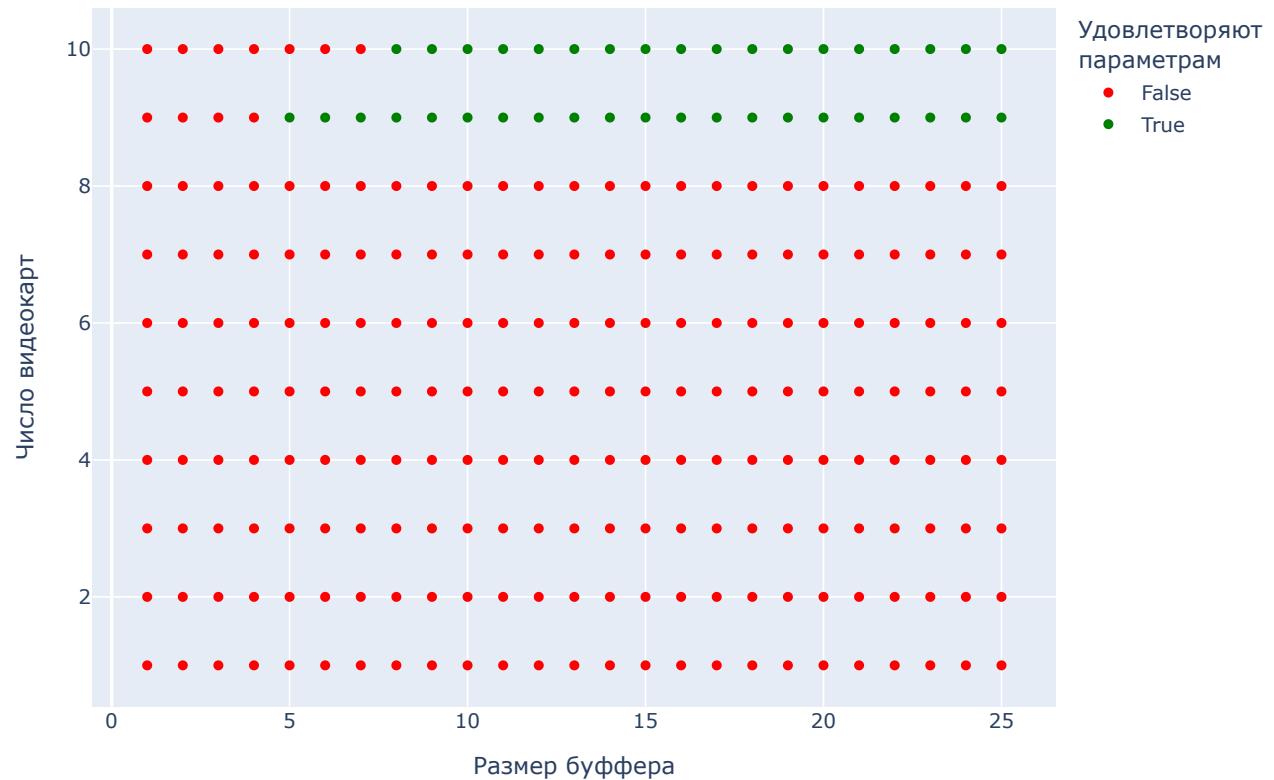


NVIDIA GeForce RTX 3050

Для NVIDIA GeForce RTX 3050 можно найти несколько конфигураций которые будут удовлетворять всем необходимым условиям. Первый набор конфигураций состоит из 9 видеокарт и буфера размером от 5 до 25 заявок. Второй набор состоит из 10 видеокарт буфера размером от 8 до 25 заявок. Самой оптимальной будет конфигурация из 9 видеокарт и 5 местного буфера, однако при необходимости можно будет дополнительно снизить вероятность отказа за счёт добавления ещё одной видеокарты.

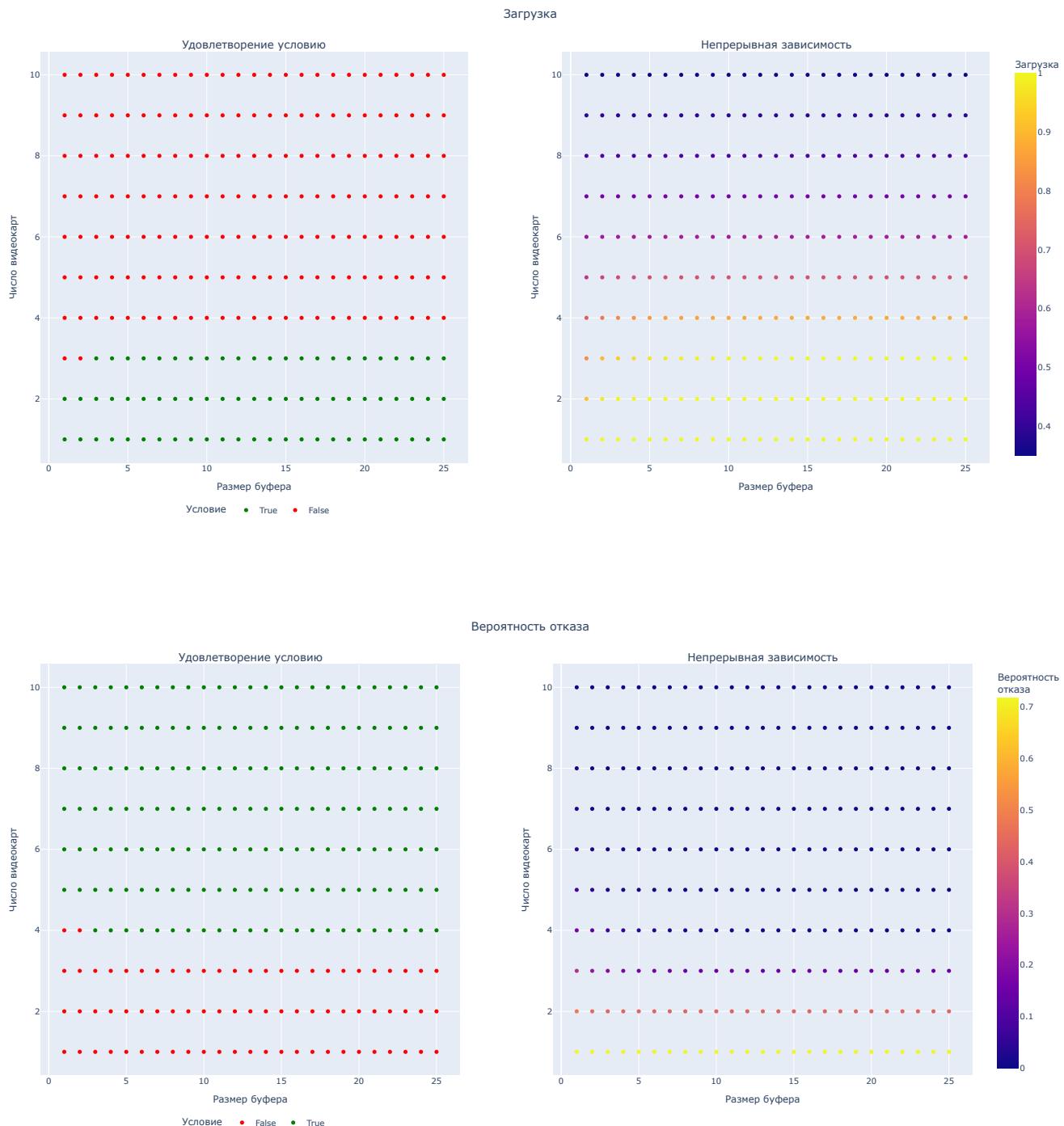


Соответствие всем условиям

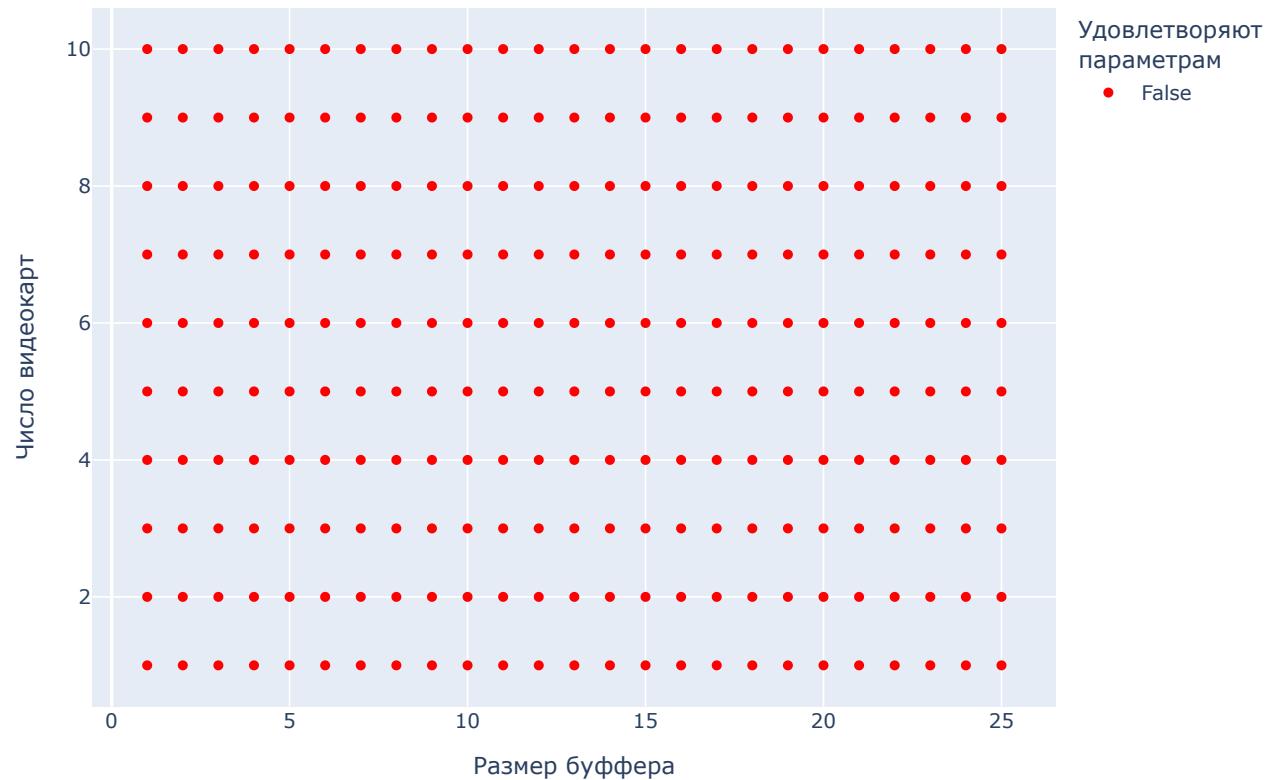


NVIDIA GeForce RTX 4080

В случае с NVIDIA GeForce RTX 4080 мы не можем найти баланс между загруженностью карт и вероятностью отказа. Если в системе меньше 4 видеокарт, то мы имеем высокую загрузку при высоком проценте отказов, а если 4 и более, то наоборот — загруженность становится менее 90% и процент отказов становится менее 10%.

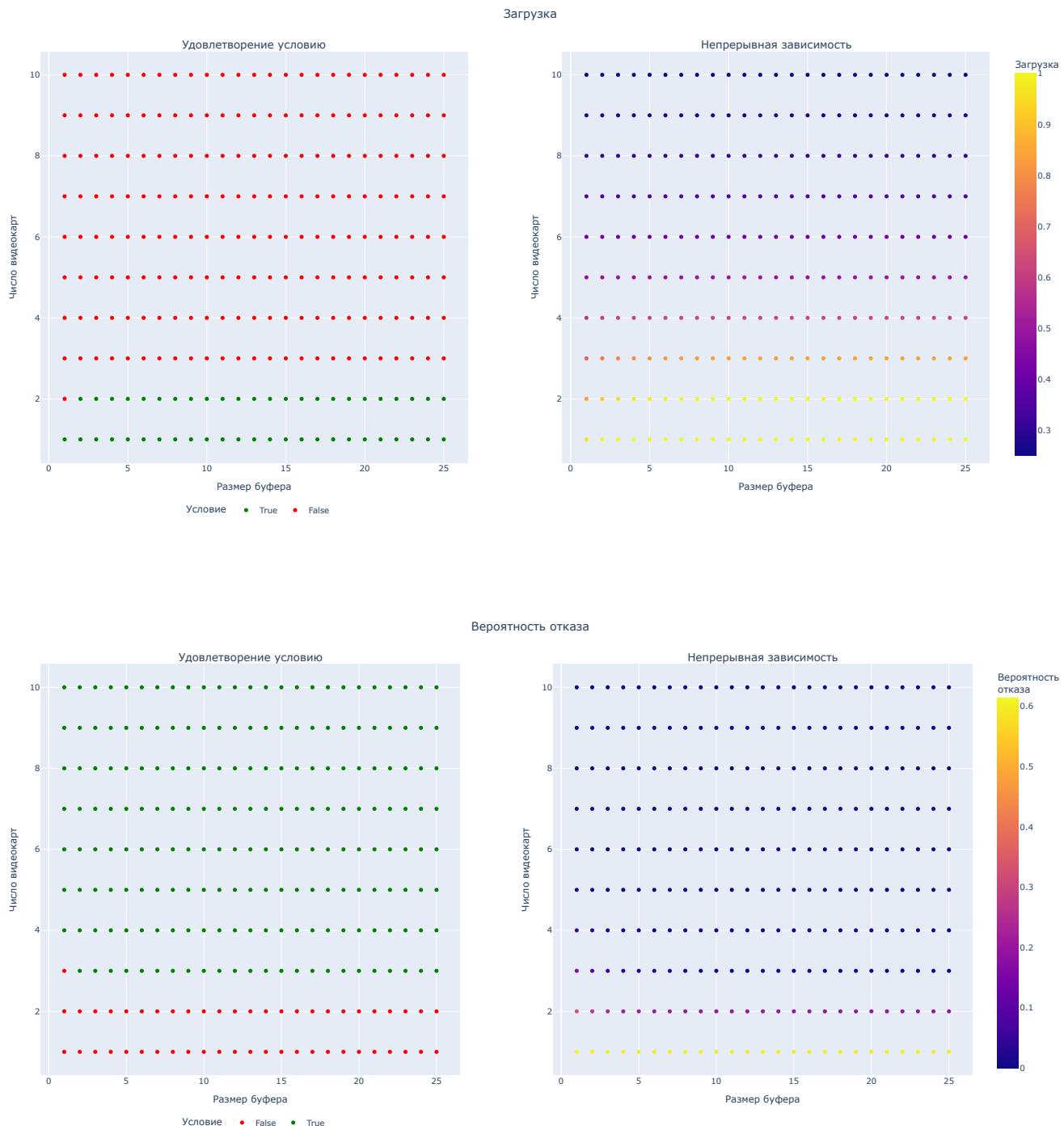


Соответствие всем условиям

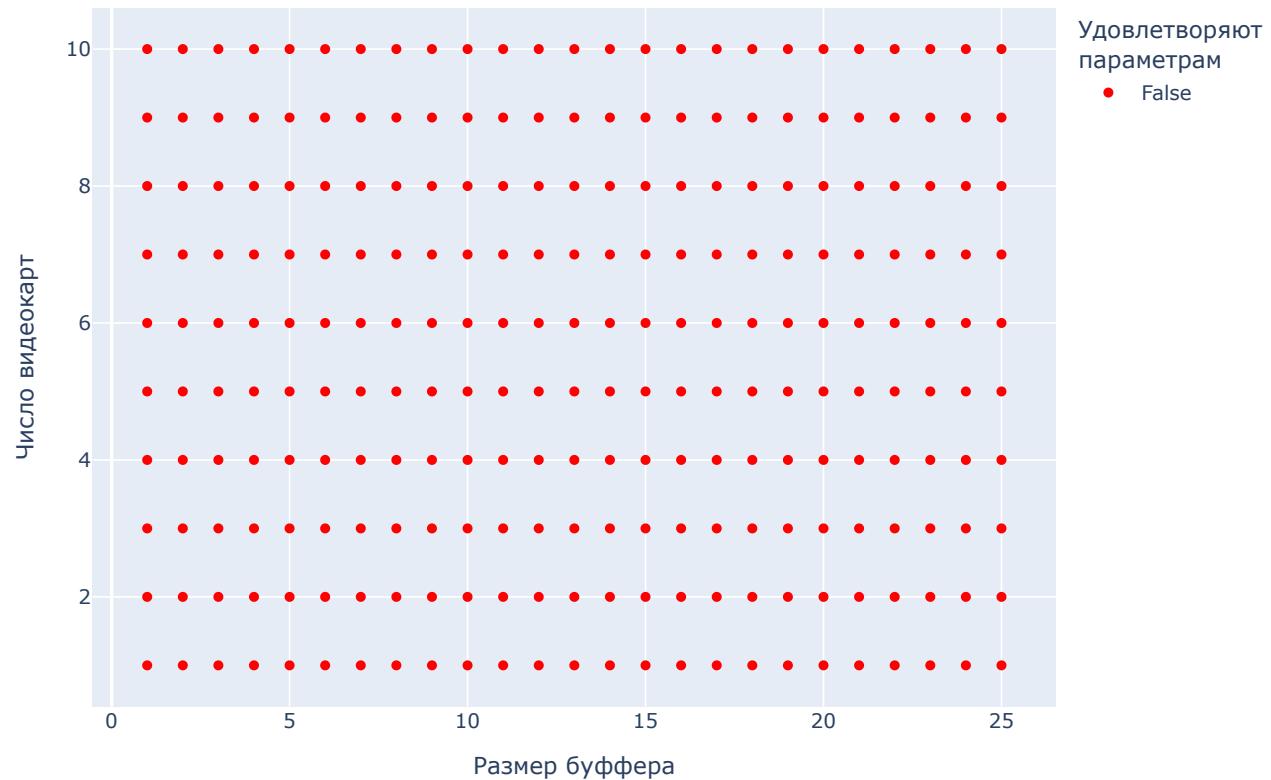


NVIDIA RTX A400

Для NVIDIA RTX A400 мы также не можем найти баланс между загруженностью карт и вероятностью отказа. Но если для NVIDIA GeForce RTX 4080 переход был между 3 и 4 видеокартами в системе, то теперь он находится между 2 и 3. Можно отметить, что проблема с более мощными NVIDIA GeForce RTX 4080 и NVIDIA RTX A400 состоит в том, что они дают слишком высокий прирост общей производительности системы при добавлении одной видеокарты, из-за чего не удается получить сбалансированные настройки для наших ограничений.



Соответствие всем условиям



Таким образом мы получили несколько вариантов конфигурации удовлетворяющих ограничениям. Все подходящие конфигурации используют NVIDIA GeForce RTX 3050. Самой оптимальной из подходящих является вариант с 9 видеокартами и 5 местами в буфере, однако можно добавить ещё одну видеокарту или увеличить число мест в буфере до максимальных 25 чтобы ещё сильнее уменьшить вероятность отказа оставаясь в рамках необходимой загрузки. Остальные видеокарты в любой конфигурации не могут удовлетворить хотя бы одному параметру.

Вывод

В ходе курсовой работы была написана имитационная модель системы массового обслуживания на языке C++, а также реализован графический интерфейс для неё на языке Python с использованием графической библиотеки Qt. С помощью данной программы была проанализирована реальная система видеонаблюдения и подобрана максимально выгодная комплектация системы, удовлетворяющая поставленным требованиям.

Список литературы

1. Гмурман В.Е. Теория вероятностей и математическая статистика : учеб. пособие для вузов
В. Е. Гмурман .— 12-е изд., перераб. — М. : Юрайт, 2010 .— 478 с.
2. Бусленко Н.П. Моделирование сложных систем. М.: Наука, 1978
3. Александрова О.В., Котляров В.П. Архитектура вычислительных систем. Методические
указания для курсового проектирования. СПбПУ, 2018г. — 73 с.