Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

Курсовая работа

Методы имитационного моделирования

по дисциплине «Архитектура программных систем»

вариант 12

Выполнил студент гр. 5130904/30102 Мальцев А. Л.

Руководитель Дробинцев Д. Ф.

Санкт-Петербург, 2025 г.

Содержание

1 Введение

В данной курсовой работе разработана имитационная модель системы массового обслуживания (СМО) для исследования характеристик вычислительной системы с множественными источниками заявок, буферной памятью и несколькими приборами обслуживания.

Цель работы: разработка и исследование программной модели СМО с заданными дисциплинами работы для анализа эффективности функционирования вычислительной системы.

Задачи работы:

- Реализация имитационной модели СМО с использованием метода особых событий
- Исследование влияния входных параметров на выходные характеристики системы
- Синтез реальной вычислительной системы, соответствующей моделируемой СМО
- Определение оптимальной конфигурации системы, удовлетворяющей заданным требованиям

Актуальность: моделирование систем массового обслуживания позволяет оптимизировать конфигурацию вычислительных систем, минимизировать отказы в обслуживании и обеспечить эффективное использование ресурсов без необходимости дорогостоящих экспериментов на реальном оборудовании.

2 Постановка задачи

Формализованная формула варианта:

ИБ-ИЗ2-ПЗ1-Д10З3-Д10О4-Д2П2-Д2Б3-ОР1-ОД3

2.1 Расшифровка формулы

Источники

- ИБ — бесконечный источник.
- ИЗ2 — равномерный закон генерации заявок (постоянный интервал между заявками).

Приборы

- ПЗ1 — экспоненциальный закон распределения времени обслуживания.

Описание дисциплин постановки и выбора

- Д10З3 — дисциплина буферизации: первое свободное место в буфере.
- Д10О4 — дисциплина отказа: выбивание последней поступившей заявки при переполнении буфера.

Дисциплины постановки на обслуживание

- Д2П2 — дисциплина выбора прибора: выбор прибора по кольцу.
- Д2Б3 — дисциплина выбора из буфера: выбор заявки из буфера по кольцу.

Виды отображения результатов работы программной модели

- ОР1 - отображение результатов: сводная таблица результатов.
- ОД3 - отображение динамики функционирования модели: временные диаграммы, текущее состояние.

3 Формализованная схема и описание СМО

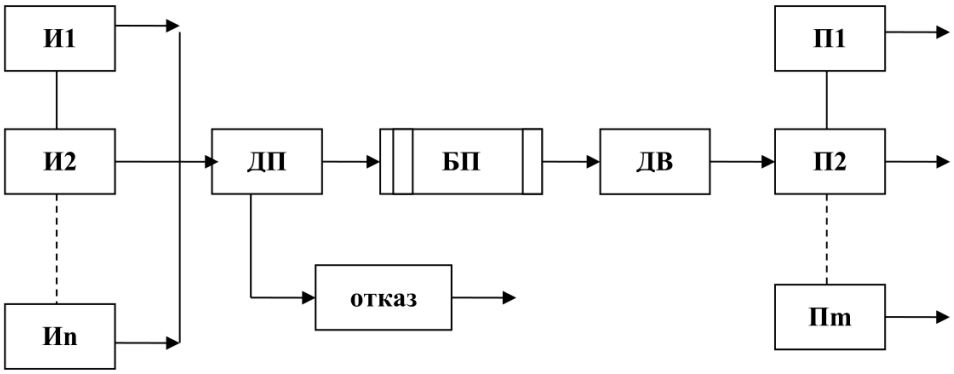

Рис. 1: Формализованная схема системы массового обслуживания.

Описание элементов системы:

- $И_i$ $(i = 1..n)$ — *источники заявок*. Каждый источник генерирует поток заявок, характеризующихся временем поступления и номером источника. Совокупность всех источников формирует суммарный входной поток системы.
- ДП — *диспетчер постановки заявок*. Он определяет дальнейшую судьбу каждой вновь поступившей заявки: при наличии свободных приборов направляет её непосредственно на обслуживание, при их отсутствии — помещает в буферную память. Если буфер заполнен, диспетчер реализует дисциплину отказа или выбивания заявки из буфера.
- БП — *буферная память*, предназначенная для временного хранения заявок, ожидающих обслуживания. Заполнение буфера происходит в соответствии с заданной дисциплиной постановки заявок.
- ДВ — *диспетчер выбора заявок*. Он осуществляет выборку заявок из буфера и их распределение по свободным приборам согласно выбранной дисциплине обслуживания.
- $П_j$ $(j = 1..m)$ — *приборы обслуживания*. Каждый прибор выполняет обработку заявок, формируя выходной поток обслуженных требований. После завершения обслуживания прибор освобождается и готов к приёму следующей заявки.

4 Временная диаграмма функционирования

Временная диаграмма строится методом особых событий, где каждое событие (поступление заявки или завершение обслуживания) отмечается на временной оси.

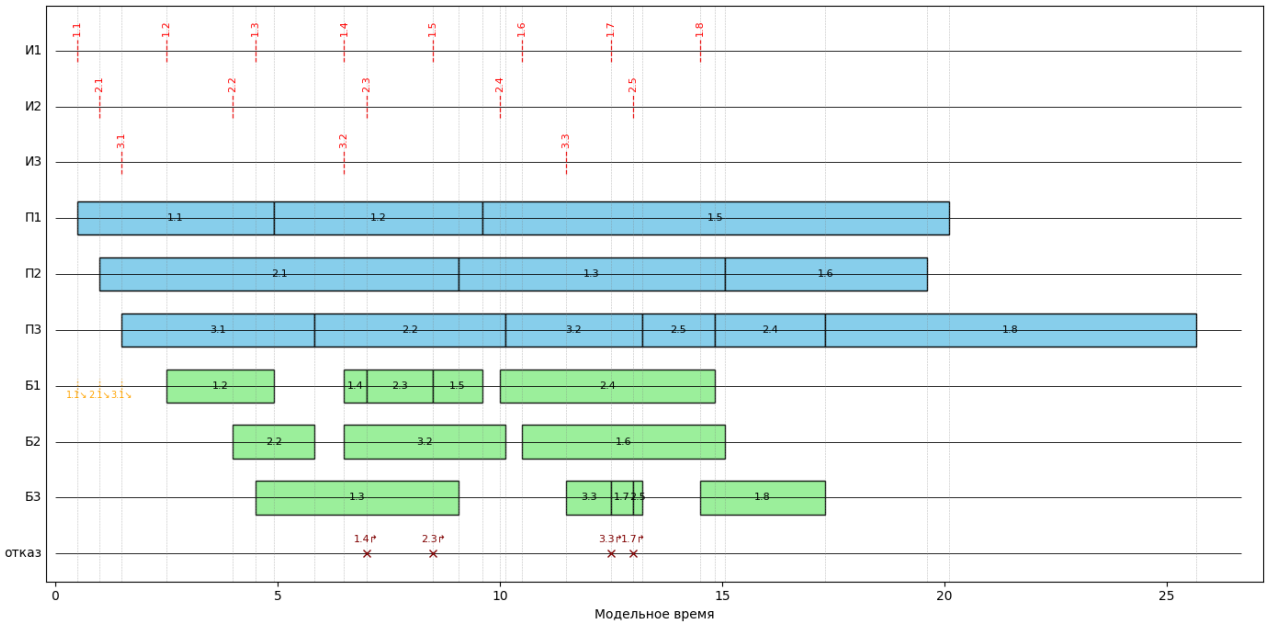

Рис. 2: Временная диаграмма функционирования СМО (метод особых событий).

Характерные моменты на диаграмме:

1. **Моменты поступления заявок:** отмечаются вертикальными линиями на временной оси для каждого источника. Заявки поступают с постоянными интервалами τ_i.

2. **Постановка в буфер:** если все приборы заняты, заявка размещается в первое свободное место буфера. На диаграмме это отображается как переход заявки в состояние ожидания в буфере.

3. **Отказ (выбивание):** при переполнении буфера последняя поступившая заявка вытесняется. На диаграмме это отмечается как разрыв линии заявки и переход в состояние отказа.

4. **Выбор на обслуживание:** следующая заявка из буфера выбирается по кольцу. На диаграмме это соответствует переходу из состояния ожидания в состояние обслуживания.

5. **Освобождение прибора:** завершение обслуживания заявки отмечается как момент выхода заявки из системы. Прибор освобождается и может принять следующую заявку из буфера.

Диаграмма демонстрирует работу всех дисциплин: заполнение буфера по порядку (Д10З3), выбивание последней заявки при переполнении (Д10О4), кольцевой выбор приборов (Д2П2) и заявок из буфера (Д2Б3).

5 Разработка и отладка программы

5.1 Обобщённая блок-схема

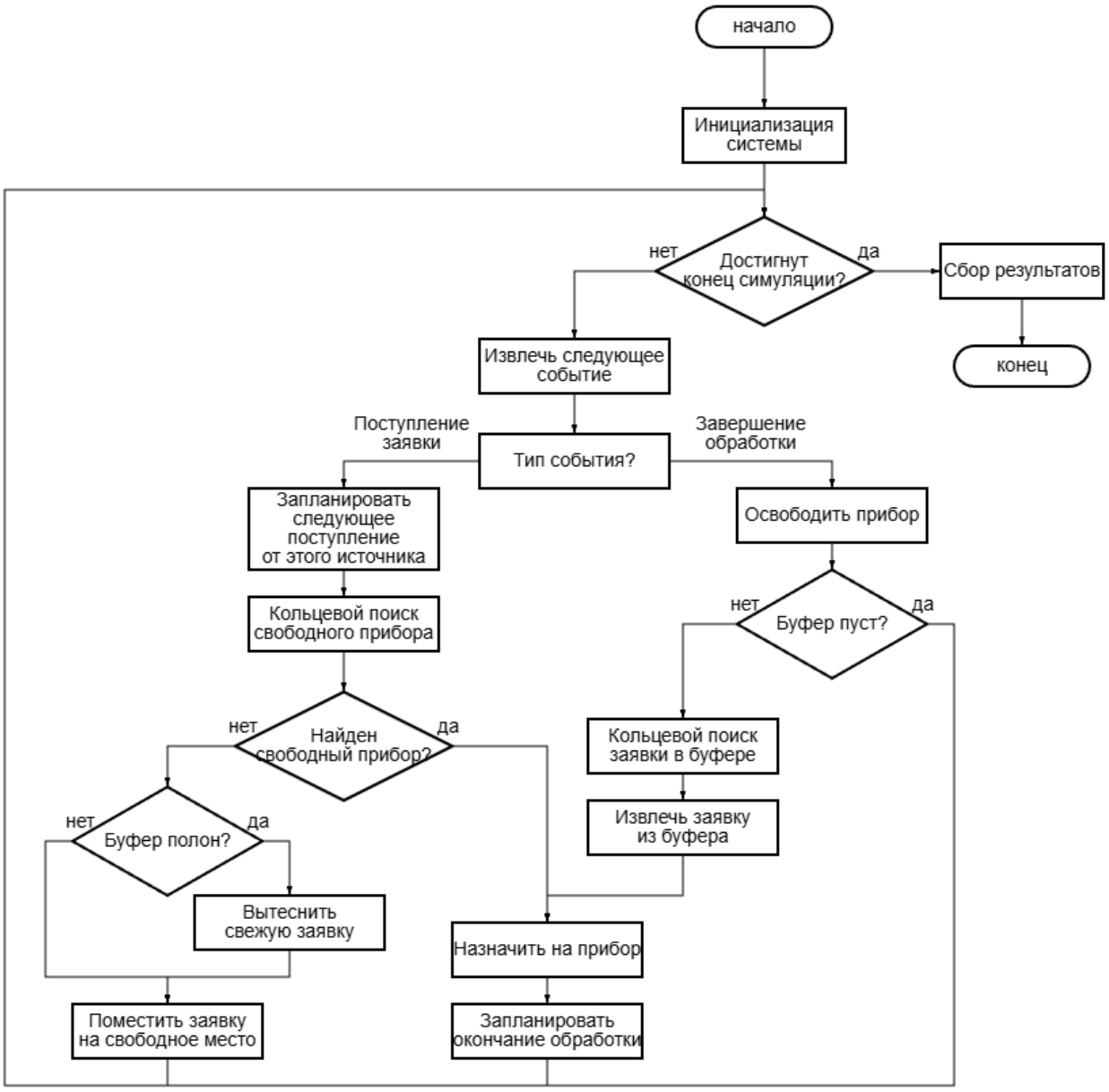

Рис. 3: Блок-схема, описывающая схему функционирования СМО.

5.2 Описание модели и подход к реализации

Моделирование реализовано методом особых событий с использованием календаря событий (event calendar) — приоритетной очереди событий, упорядоченной по времени.

Календарь событий: реализован как приоритетная очередь (min-heap), хранящая события типа (время, тип события, заявка, прибор, источник). События извлекаются в порядке возрастания времени.

Обработка события поступления:

1. Создание новой заявки с уникальным ID и временем поступления.
2. Поиск свободного прибора по кольцу.

3. Если прибор найден: начало обслуживания, генерация времени обслуживания, планирование события завершения.
4. Если приборов нет: попытка размещения в буфер (первое свободное место), при переполнении — вытеснение последней заявки.
5. Планирование следующего поступления от того же источника.

Обработка события завершения обслуживания:

1. Завершение обслуживания на приборе.
2. Запись статистики: время пребывания, время ожидания, время обслуживания.
3. Если буфер не пуст: выбор заявки из буфера по кольцу, начало обслуживания.

5.3 Законы распределения

Равномерный закон генерации (ИЗ2): заявки генерируются с постоянным интервалом τ_i. Время следующего поступления:

$$t_{next} = t_{current} + \tau$$

где τ — интервал генерации для данного источника, задаётся константой.

Экспоненциальный закон обслуживания (ПЗ1): время обслуживания T распределено по экспоненциальному закону с параметром μ:

$$f(t) = \mu e^{-\mu t}, \quad t \geq 0$$

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

Генерация случайной величины по методу обратной функции:

$$T = -\frac{1}{\mu}\ln(1 - U) = -\frac{1}{\mu}\ln(U')$$

где U — случайная величина, равномерно распределённая на $[0, 1]$, $U' = 1 - U$ также равномерно распределена на $[0, 1]$.

В программе используется стандартная библиотека C++: `std::exponential_distribution<double>` с параметром μ.

5.4 Динамическое и автоматическое отображение

ОД3 (динамическое отображение — временные диаграммы): реализовано в виде виджета `TimelineWidget`, который отображает временную диаграмму функционирования системы в реальном времени. Диаграмма показывает:

- Моменты поступления заявок от каждого источника
- Время пребывания заявок в буфере
- Время обслуживания заявок на приборах
- Моменты отказов (выбивания заявок)
- Состояние системы в каждый момент времени

Диаграмма обновляется при каждом шаге моделирования и позволяет визуально отслеживать работу системы.

ОР1 (автоматическое отображение — сводная таблица результатов): реализовано в виде виджета `AnalyticsWidget`, который отображает таблицы результатов после завершения моделирования:

- **Таблица 1:** характеристики источников (количество заявок, вероятность отказа, среднее время пребывания, среднее время ожидания, среднее время обслуживания, дисперсии)

Характеристики источников ВС							
№ источника	Количество заявок	$p_{\text{отк}}$	$T_{\text{преб}}$	$T_{\text{БП}}$	$T_{\text{обсл}}$	$Д_{\text{БП}}$	$Д_{\text{обсл}}$
И1							
И1							
...							
Иn							

- **Таблица 2:** характеристики приборов (коэффициент использования)

Характеристики приборов ВС	
№ прибора	Коэффициент использования
П1	
П2	
...	
Пm	

Также реализован виджет `EventCalendarWidget` для отображения календаря событий в пошаговом режиме, показывающий запланированные события и текущее состояние системы.

Календарь событий		Число заявок	Число отказов
Событие	Время		
И1			
...			
Иn			
П1			
...			
Пm			

6 Структура проекта

Основные компоненты:

- **Core (ядро моделирования):** основные классы и алгоритмы моделирования (`Simulator`, `EventCalendar`, `Buffer`, `Device`, `Request`, `Event`, `Metrics`). Эти классы находятся в библиотеке `sim_core` (папка `libs/sim_core/`).
- **GUI (графический интерфейс):** Qt-виджеты и окно приложения — `MainWindow`, `TimelineWidget`, `EventCalendarWidget`, `AnalyticsWidget`. Код расположен в `apps/gui/` и использует API ядра для визуализации динамики и вывода аналитики.
- **CLI (консольный интерфейс и утилиты):** консольные утилиты для отладки и пакетных расчётов: пошаговый отладочный запуск (`apps/cli/src/main.cpp`) и runner для перебора конфигураций (`apps/cli/src/sim_sweep.cpp`).

6.1 Модульная структура и описание модулей

Ядро моделирования (`libs/sim_core/`):

- `Simulator.h/cpp`: основной класс симулятора, управляющий циклом обработки событий методом особых событий. Отвечает за создание источников, приборов, запуск обслуживания на приборах, работу диспетчеров и взаимодействие с календарём событий.

- `Event.h/cpp`: представление события (время, тип события — поступление/завершение, связанная заявка/прибор/источник).
- `EventCalendar.h/cpp`: календарь событий — min-heap / приоритетная очередь для планирования и извлечения следующих по времени событий.
- `Buffer.h/cpp`: реализация буфера фиксированной ёмкости с дисциплинами постановки и выбора: Д10З3 (первое свободное место) и Д2Б3 (кольцевой выбор). Методы: `placeRequest()`, `takeRequest()`, `displaceRequest()`.
- `Device.h/cpp` и `DevicePool.h/cpp`: представление прибора обслуживания, состояние (свободен/занят), учёт времени занятости, методы `startService()`, `finishService()`, `isFree()` и пул приборов с поддержкой дисциплины выбора Д2П2 (по кольцу).
- `Request.h/cpp`: класс заявки с идентификатором, номером источника, временем поступления, временем начала и окончания обслуживания.
- `Metrics.h/cpp`: сбор статистики по системе и по каждому источнику/прибору: число поступивших, отказанных, завершённых, суммарные и средние времена, дисперсии, коэффициенты использования приборов.
- **Дополнительные модули:** `SourcePool` (управление источниками), `EventDispatcher` (правила постановки/выбора), и интерфейсы наблюдателей (Observers) для GUI/логирования.

Графический интерфейс (`apps/gui/`):

- `MainWindow.h/cpp`: окно приложения, элементы управления симуляцией (Step, Run, Pause, Reset), запуск и остановка симулятора, передача событий и статистики виджетам.
- `TimelineWidget.h/cpp`: визуализация временной диаграммы (ОД3) — отображение поступлений, ожидания в буфере, обслуживания на приборах и отказов.
- `EventCalendarWidget.h/cpp`: табличное представление календаря событий и ближайших планируемых событий.
- `AnalyticsWidget.h/cpp`: отображение итоговой сводной таблицы (ОР1) — характеристики источников и приборов.

Консольный интерфейс и утилиты (`apps/cli/`):

- `main.cpp`: пошаговый режим отладки с выводом состояния симулятора в консоль.
- `sim_sweep.cpp`: утилита пакетного перебора конфигураций, реализующая сеточный поиск по входным параметрам, запуск симуляции для каждой конфигурации, отбор по ограничениям и экспорт результатов в CSV (`sweep_results.csv`).
- **CMake:** каждый исполняемый файл собирается отдельной целью (`sim_cli`, `sim_sweep`), подключающей статическую библиотеку ядра `sim_core`.

6.2 Формат входных и выходных данных

Входные данные: конфигурация моделирования задаётся структурой `SimulationConfig` (файл `sim/simulator/SimulationConfig.h`) и включает параметры:

- ёмкость буфера (`buffer_capacity`);
- число поступлений/максимальное число заявок (`max_arrivals`);

- зерно генератора (`seed`);
- список источников (`sources`): для каждого — `id`, параметр интервала, тип распределения (постоянный/экспоненциальный);
- список приборов (`devices`): для каждого — `id`, параметр обслуживания (параметр `mu` для экспоненциального распределения), тип распределения.

Выходные данные: через класс `Metrics` доступны агрегированные и по-источниковые показатели:

- общие характеристики: число поступивших, отказанных и завершённых заявок; вероятность отказа; средние времена пребывания/ожидания/обслуживания;
- по источникам: количество поступлений, вероятность отказа, среднее и дисперсии времени ожидания и обслуживания;
- по приборам: коэффициент использования (загруженность) для каждого прибора;
- в пакетном режиме (утилита `sim_sweep`): CSV-файл `sweep_results.csv` с колонками конфигурации, метриками и отметкой соответствия ограничениям (passes).

7 Пример реальной вычислительной системы (синтез модели)

7.1 Описание

В данном разделе приводится пример потенциально существующей вычислительной системы (ВС), полностью соответствующей заданным дисциплинам и законам.

Цель — задать количественные значения входных параметров и сформулировать требования к выходным характеристикам, чтобы в дальнейшем оценивать соответствие модели этим требованиям.

7.1.1 Определение эффективности ВС

Эффективность системы оценивается по трём выходным характеристикам:

$p_{\text{отк}}$ — вероятность отказа в обслуживании;

$T_{\text{преб}}$ — среднее время пребывания заявки в системе;

$K_{\text{исп}}$ — коэффициент использования приборов.

Задание желаемых значений этих величин формирует требования к работе ВС.

7.1.2 Подход к проектированию модели

Модель строится путём фиксации входных параметров в допустимых диапазонах, исходя из физического смысла компонентов системы. Это позволяет сформировать архитектуру, удовлетворяющую заданным требованиям.

7.2 Пример ВС, соответствующей модели

7.2.1 Описание ВС

В качестве реальной вычислительной системы рассматривается **система агрегации телеметрии от распределённых датчиков**, обеспечивающая непрерывный сбор, временное буферирование и обработку данных (температуры, давления, вибрации, расхода, уровня жидкости и др.) на промышленном объекте. Каждый датчик периодически передаёт пакет фиксированного размера на центральный модуль обработки. Центральный модуль распределяет поступающие пакеты по вычислительным узлам (приборы обслуживания) для выполнения фильтрации, сглаживания, детекции аномалий и агрегации.

7.2.2 Соответствие модели с ВС

Элемент системы	**Дисциплина / закон**	**Соответствие в ВС**
Источники	ИБ–ИЗ2	Датчики работают непрерывно и передают пакеты (2 Кбайт) с фиксированным интервалом — что соответствует бесконечному источнику с равномерной генерацией
Приборы	ПЗ1	Время обработки пакета случайно из-за вариативности данных и нагрузки, что моделируется экспоненциальным распределением
Буфер	—	Массив фиксированных ячеек в модуле агрегации (ring-buffer)
Постановка в буфер	Д10З3	Новый пакет размещается в первое свободное место — минимизирует задержку вставки и упрощает реализацию в промышленных контроллерах
Отказ в буфере	Д10О4	При переполнении вытесняется последняя добавленная заявка: новые сэмплы шумнее, а старые необходимы для усреднения
Выбор из буфера	Д2Б3	Чтение заявок — кольцевым обходом, что соответствует аппаратной реализации ring-buffer
Выбор прибора	Д2П2	При освобождении прибора следующая заявка направляется на следующий по кольцу модуль — обеспечивает равномерную загрузку

7.2.3 Ограничения и требуемые характеристики

Требования к выходным характеристикам:

$p_{\text{отк}} \leq 0.10$ — не более 10% потерянных пакетов;

$T_{\text{преб}} \leq 200$ мс — задержка не должна нарушать требования реального времени;

$K_{\text{исп}} \geq 0.9$ — приборы должны быть загружены не менее чем на 90%.

Входные параметры и их диапазоны:

Параметр	**Диапазон**
Количество датчиков	4–20
Интервал генерации телеметрии	50–500 мс (шаг 50 мс)
Количество приборов	1–6
Тип прибора	Тип 1 (120 мс), Тип 2 (90 мс), Тип 3 (60 мс)
Размер буфера	8–40 ячеек (шаг 8)
Размер заявки	2 Кбайт

Стоимость компонентов:

Компонент	Характеристики	Цена, руб.
Прибор Тип 1	среднее время 120 мс	6000
Прибор Тип 2	среднее время 90 мс	9000
Прибор Тип 3	среднее время 60 мс	15000
Блок буфера	+8 ячеек	800

8 Анализ конфигураций

8.1 Описание

Цель этапа — исследовать работу модели при различных конфигурациях входных параметров, отсеять недопустимые по требованиям и выбрать оптимальную по минимальной стоимости.

Необходимо:

- определить достаточное число заявок для заданной точности;
- провести моделирование по сетке параметров;
- отфильтровать конфигурации по ограничениям;
- выбрать конфигурацию с минимальной стоимостью среди допустимых.

8.2 Определение оптимального числа заявок

Требуемая точность:

относительная точность: $\delta = 0.1$ (10%);

доверительная вероятность: $\alpha = 0.9$ ($t_\alpha = 1.643$).

Число заявок определяется по формуле:

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

где p — оценка вероятности отказа.

8.2.1 Пристрелка

Алгоритм определения необходимого числа заявок («пристрелка»):

1. Назначить начальное количество заявок $N_0 = 200$.
2. Провести моделирование: пропустить N_0 заявок через систему, получить оценку вероятности отказа p_0.
3. Вычислить новое значение N_1.
4. Выполнить моделирование с N_1 заявками и получить новую оценку p_1.
5. Проверить условие стабилизации:
$$|p_1 - p_0| < 0.1 \cdot p_0.$$
6. Если условие выполнено, принять $N = N_1$ как достаточное. В противном случае положить $p_0 := p_1$ и повторить шаги 3–6.

На практике для всех конфигураций принято фиксированное значение:

$$N = 10000,$$

что обеспечивает точность даже при малых $p_{\text{отк}}$.

8.3 Теория для тестирования

Варьируемые параметры:

- количество датчиков: 4—20;
- интервал генерации: 50—500 мс (с шагом 50 мс);
- количество приборов: 1–6;
- тип прибора: 1, 2 или 3;
- размер буфера: 8—40 (с шагом 8).

Общее число конфигураций — несколько тысяч. Для каждой:

1. проводится моделирование с $N = 10000$ заявок;
2. вычисляются $p_{\text{отк}}$, $T_{\text{преб}}$, $K_{\text{исп}}$;
3. проверяется выполнение ограничений;
4. вычисляется стоимость $C_{\text{сист}}$.

Стоимость системы:

$$C_{\text{сист}} = (\text{число приборов}) \cdot (\text{стоимость данного типа приборов}) + \frac{K}{8} \cdot 800.$$

9 Анализ результатов и выбор оптимальной конфигурации

Процесс фильтрации конфигураций:

Проведён полный перебор всех **15 300** возможных конфигураций вычислительной системы. Отбор допустимых решений осуществлялся последовательно по трём критериям эффективности:

1. Вероятность отказа не превышает 10%: $p_{\text{отк}} \leq 0.10$ — прошли **6 370** конфигураций;
2. Среднее время пребывания заявки в системе не превышает 200 мс: $T_{\text{преб}} \leq 200$ мс — осталось **4 938** конфигураций;
3. Коэффициент использования приборов не ниже 90%: $K_{\text{исп}} \geq 0.90$ — осталось **75** конфигураций.

Из этих 75 удовлетворяющих всем условиям конфигураций отобраны 10 с наименьшей стоимостью. Их параметры и характеристики приведены в таблице 1.

Таблица 1: 10 лучших конфигураций по критерию минимальной стоимости

Датчики	**Интервал, мс**	**Приборы**	**Тип прибора**	**Буфер**	$p_{\text{отк}}$	$T_{\text{преб}}$, мс	$K_{\text{исп}}$	**Стоимость, руб.**
5	150	3	2	8	0.0673	196.28	0.9337	27 800
7	200	3	2	8	0.0842	192.39	0.9262	27 800
4	100	5	1	8	0.0344	177.92	0.9044	30 800
5	150	2	3	8	0.0544	157.88	0.9131	30 800
6	150	5	1	8	0.0427	183.59	0.9040	30 800
8	200	5	1	8	0.0678	197.47	0.9043	30 800
5	100	6	1	8	0.0632	189.00	0.9478	36 800
7	150	4	2	8	0.0998	174.23	0.9430	36 800
7	150	6	1	16	0.0059	197.09	0.9076	37 600
8	200	4	2	16	0.0085	199.92	0.9039	37 600

Наилучшей по стоимости признана конфигурация:

$$\text{датчики} = 5, \quad \text{интервал} = 150 \text{ мс}, \quad \text{приборы} = 3, \quad \text{тип прибора} = 2, \quad \text{буфер} = 8 \text{ ячеек},$$

с общей стоимостью **27 800 руб.**. Для неё получены следующие характеристики:

$$p_{\text{отк}} = 6.73\%, \qquad T_{\text{преб}} = 196.3 \text{ мс}, \qquad K_{\text{исп}} = 93.4\%.$$

Анализ влияния параметров:

- **Увеличение числа приборов:** снижает вероятность отказа и время ожидания, увеличивает стоимость системы.
- **Увеличение ёмкости буфера:** снижает вероятность отказа при переполнении, незначительно влияет на стоимость.
- **Увеличение интенсивности обслуживания:** снижает время пребывания и вероятность отказа, значительно увеличивает стоимость (более производительные серверы).
- **Оптимальный баланс:** достигается при комбинации параметров, обеспечивающей выполнение ограничений при минимальной стоимости.

10 Вывод

В ходе выполнения курсовой работы была разработана и исследована имитационная модель системы массового обслуживания с заданными дисциплинами работы.

Достигнутые цели:

- С использованием метода особых событий реализована имитационная модель СМО с заданными дисциплинами работы.
- Реализован графический интерфейс, позволяющий пользователю производить гибкую конфигурацию системы, наблюдать за динамикой работы системы (временные диаграммы, календарь событий) и результатами моделирования (сводные таблицы).
- Проведено исследование влияния входных параметров на выходные характеристики системы.

Оптимальные конфигурации: в результате исследования найдены конфигурации системы, удовлетворяющие требованиям к вероятности отказа ($p \leq 0.10$), времени пребывания ($T \leq 200$ мс) и коэффициенту использования ($K \geq 0.90$) при минимальной стоимости.

A Приложение А: Формулы и определения

A.1 Расчёт числа реализаций N

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

где $t_\alpha = 1.643$ для $\alpha = 0.9$, $\delta = 0.1$ (10% точность), p — вероятность отказа.

A.2 Расчёт вероятности отказа

$$p = \frac{m}{n}$$

где m — количество отказов, n — количество поступивших заявок.

A.3 Расчёт среднего времени пребывания

$$T_{\text{преб}} = T_{\text{БП}} + T_{\text{обсл}}$$

где $T_{\text{БП}}$ — среднее время пребывания в буфере, $T_{\text{обсл}}$ — среднее время обслуживания.

A.4 Расчёт коэффициента использования

$$K_{\text{исп}} = \frac{T_{\text{занятости}}}{T_{\text{общее}}}$$

где T — суммарное время занятости прибора, T — общее время моделирования.

A.5 Расчёт загрузки системы

$$\rho = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{m} \mu_j}$$

где λ_i — интенсивность i-го источника, μ — интенсивность обслуживания прибора, m — количество приборов.

A.6 Расчёт дисперсии

$$D = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 = \overline{x^2} - \overline{x}^2$$

где x_i — значения случайной величины, n — количество значений.

A Приложение B: Скриншоты

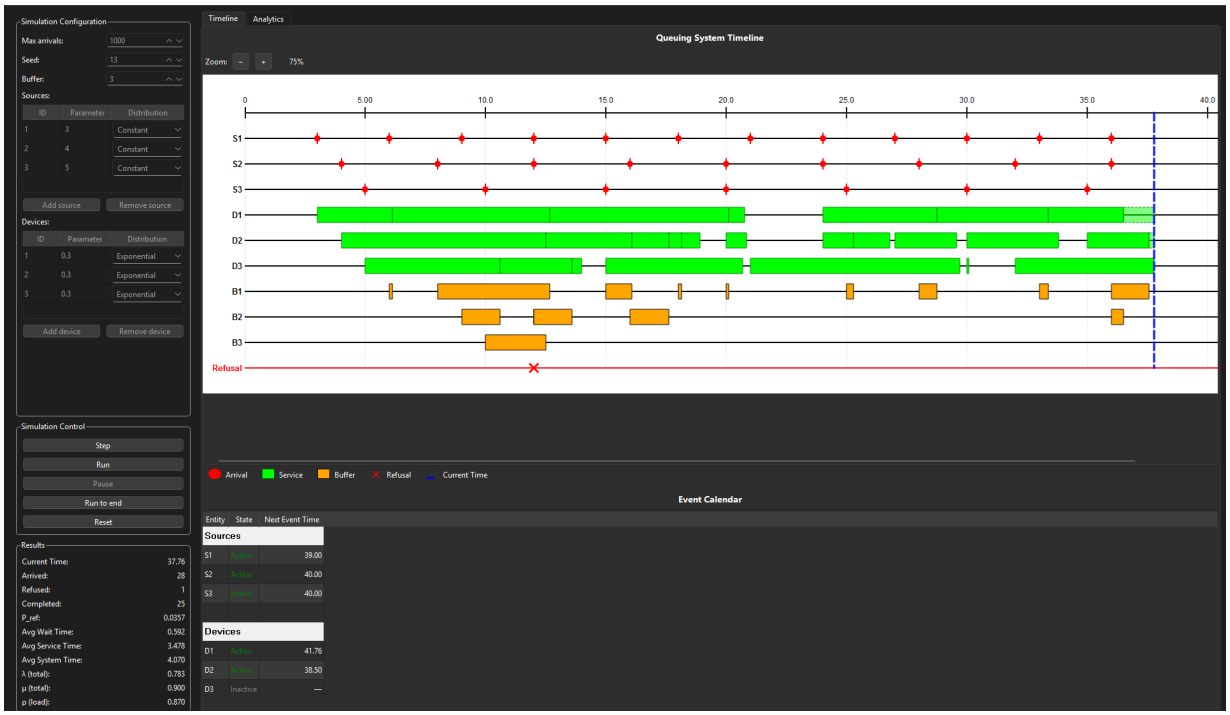

Рис. 4: Главный экран перед запуском симуляции: текущая конфигурация, результаты работы в реальном времени, календарь событий и графики.

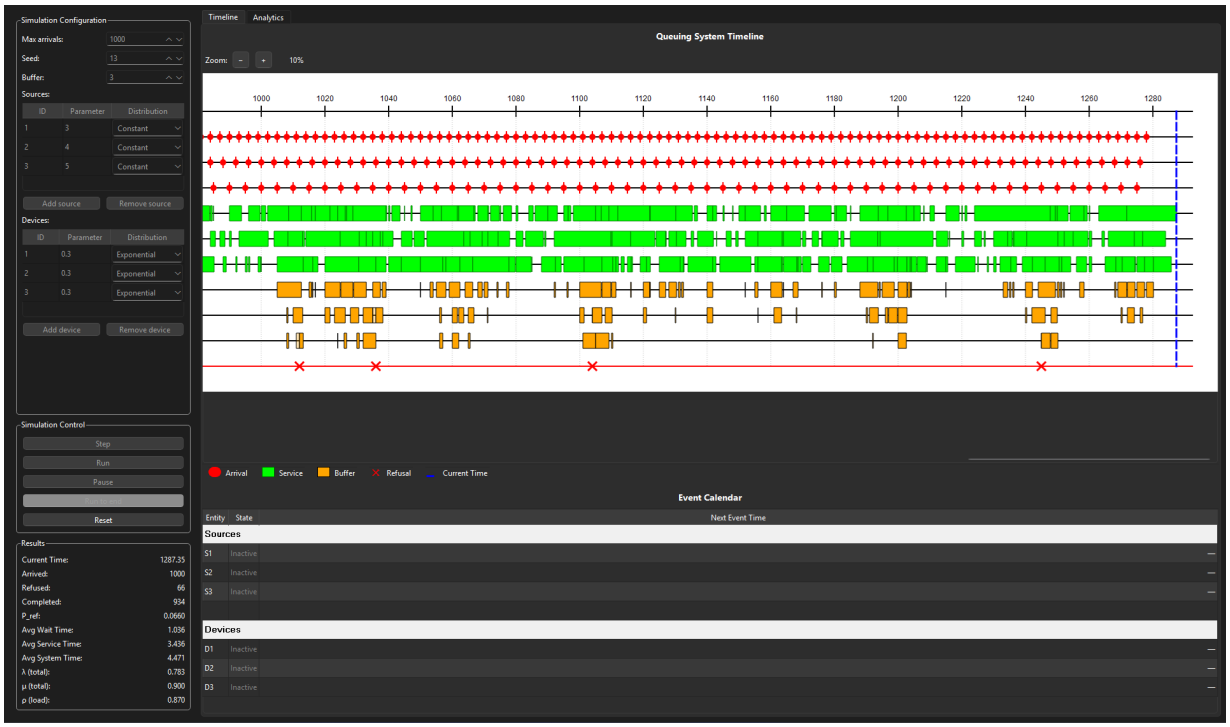

Рис. 5: Главный экран после завершения симуляции.

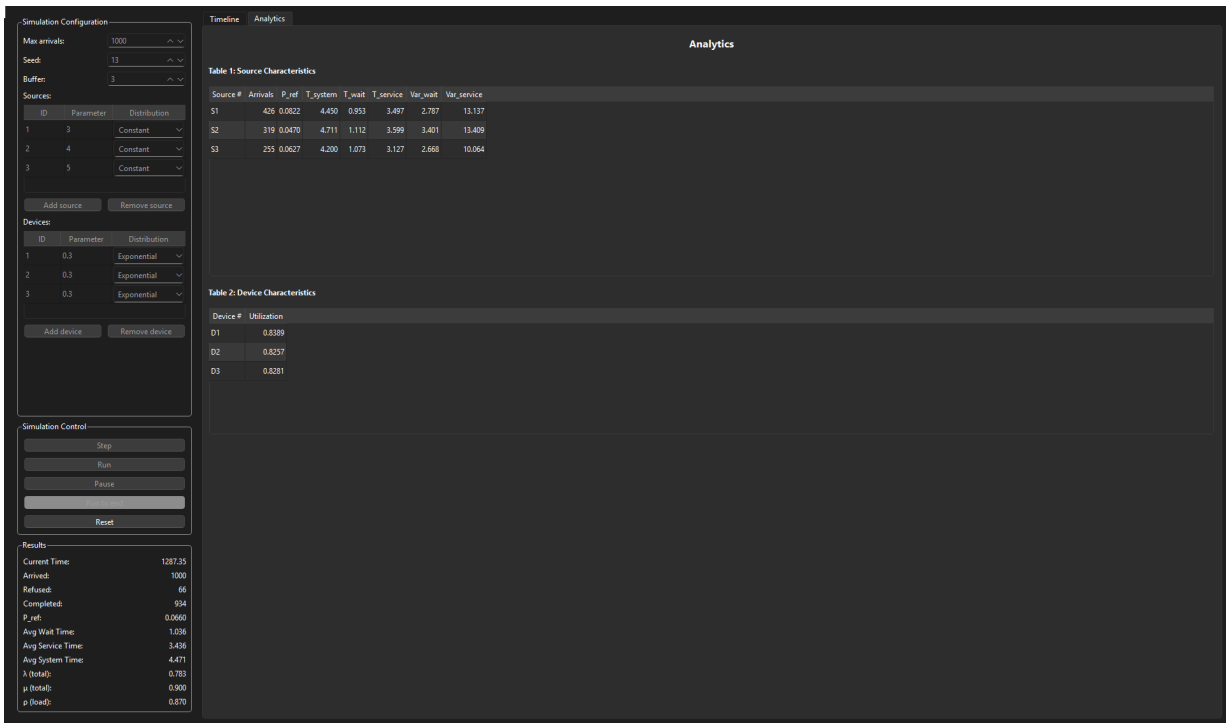

Рис. 6: Экран аналитики: итоговые метрики