

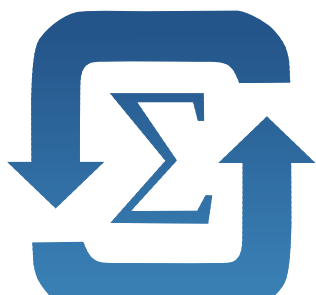
# Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 3  
по дисциплине «Компьютерное моделирование»



|                |   |
|----------------|---|
| ФАКУЛЬТЕТ:     | ПМИ                                     |
| ГРУППА:        | ПМИ-61                                  |
| СТУДЕНТЫ:      | Ершов П.К., Мамонова Е.В., Цыденов З.Б. |
| ПРЕПОДАВАТЕЛЬ: | Черникова О. С. Карманов В. С.          |

Новосибирск

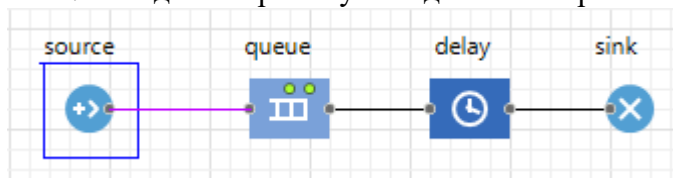
2020

## 1. Цель работы

Провести имитационное моделирование системы массового обслуживания и научиться оптимизировать ее параметры.

## 2. Ход работы

1. Создаём первичную модель и настраиваем её.



### source - Source

Имя:  ☒ Отображать имя ☐ Исключить

Прибывают согласно:

Время между прибытиями:

### queue - Queue

Имя:

Вместимость:

Максимальная вместимость:

Место агентов:

#### Специфические

Очередь:

Разрешить уход по таймауту:

Разрешить выпеснение:

Вернуть агента в исходную точку: ☒

Включить сбор статистики: ☒

### delay - Delay

Имя:  ☒

Тип задержки: ☒ Определенное время  
☐ До вызова функции stopD

Время задержки:

Вместимость:

Максимальная вместимость:

Место агентов:

#### Специфические

Выталкивать агентов:

Вернуть агента в исходную точку: ☒

Включить сбор статистики: ☒

### Simulation - Простой эксперимент

Имя:  ☐ Исключить

Агент верхнего уровня:

Максимальный размер памяти:  M6

#### Модельное время

Режим выполнения: ☒ Виртуальное время (максимальная скорость)

☐ Реальное время со скоростью

Остановить:

Начальное время:

Конечное время:

Начальная дата:

Конечная дата:


#### Случайность

Генератор случайных чисел:

☐ Случайное начальное число (уникальные "прогоны")

☒ Фиксированное начальное число (воспроизводимые "прогоны") Начальное число:

Проверяем работоспособность:


 **Ошибка при выполнении дискретного события**

Логическая ошибка в модели:

root.source:  
Агент не смог покинуть этот порт: root.source.out в момент времени 26.277 / дату 1 мар. 2020 г., 0:26:16 (текущее модельное время: 31.615)  
Увеличьте вместимости объектов и/или пропускную способность последующих объектов

Подробности смотрите в [Консоли](#)

Изменяем параметры очереди:

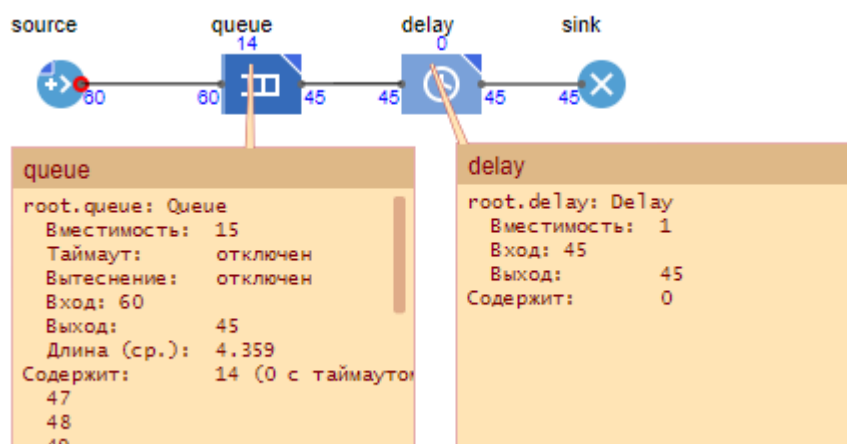
 **queue - Queue**


Имя:

Вместимость:

Максимальная вместимость: ☐

Проверяем работоспособность:



 **Ошибка при выполнении дискретного события**


Логическая ошибка в модели:

root.source:  
Агент не смог покинуть этот порт: root.source.out в момент времени 134.773 / дату 1 мар. 2020 г., 2:14:46 (текущее модельное время: 135.716)  
Увеличьте вместимости объектов и/или пропускную способность последующих объектов

Подробности смотрите в [Консоли](#)

Выбираем работоспособные параметры модели:

Модель успешно выполняется при следующих параметрах:

 **source - Source**

Имя:  ☒ Отображать имя

Прибывают согласно:

Время между прибытиями:

## queue - Queue

Имя: queue

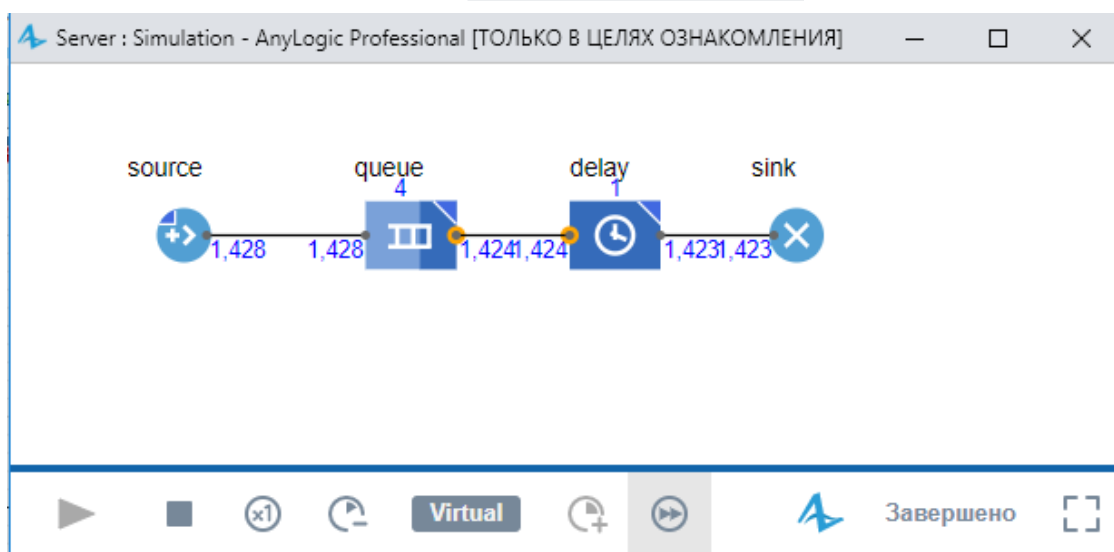
Вместимость: 20

## delay - Delay

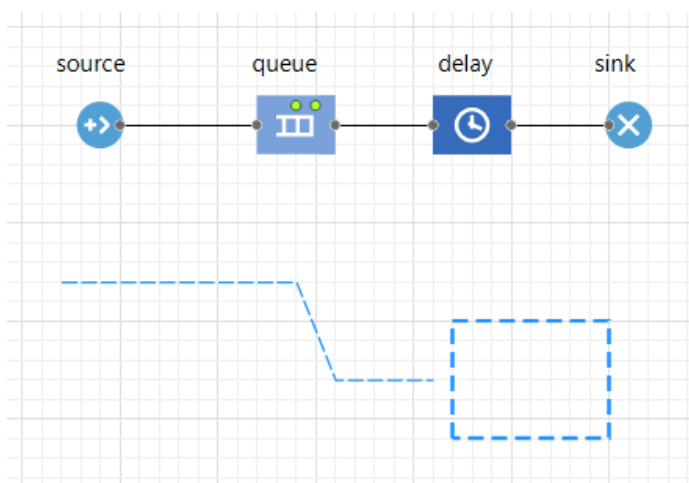
Имя: delay ☒ От

Тип задержки: ☒ Определенное время  
☐ До вызова функции stopDel

Время задержки:



2. Создаём анимацию модели.



## queue - Queue

Имя: queue

Вместимость: 20

Максимальная вместимость:

Место агентов: path

## delay - Delay

Имя: delay

Тип задержки: ☒ Определенное время  
☐ До вызова функции stopDel

Время задержки:

Вместимость:

Максимальная вместимость:

Место агентов: node

**node - Прямоугольный узел**

Имя:  ☐ Искать

☐ Блокировать

Видимость: ☒ да

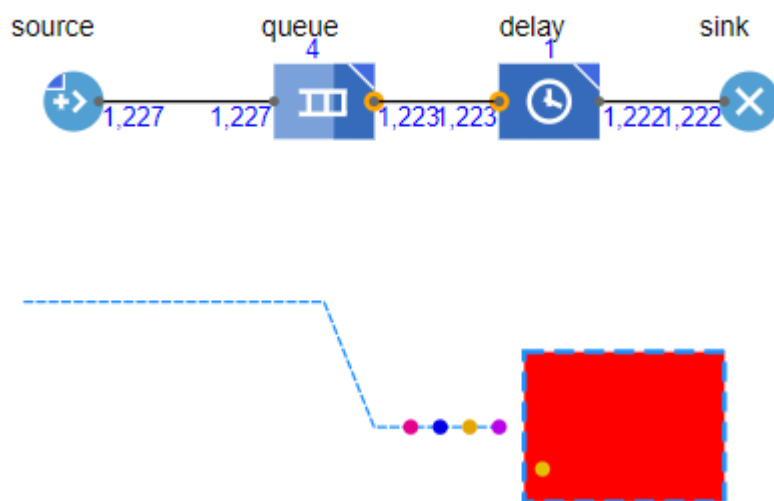
☐ Ограничение скорости для транспортеров

Расположение внутри:

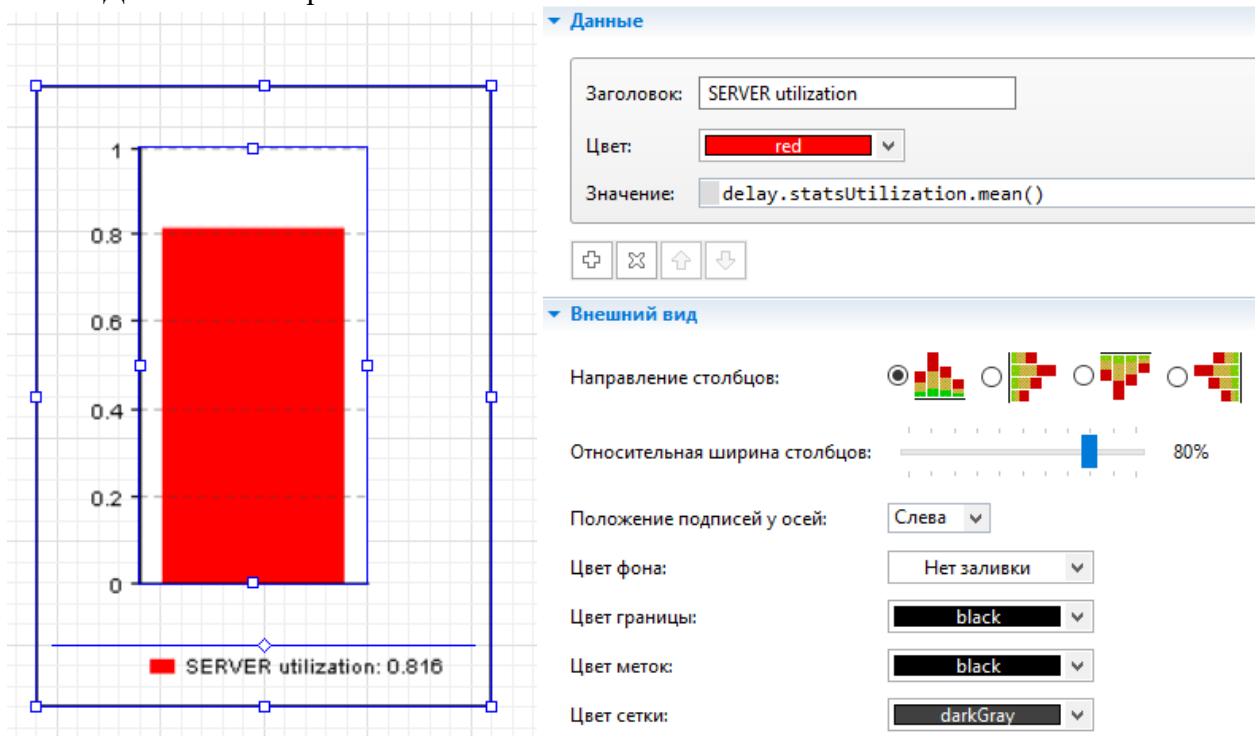
**Внешний вид**

Цвет заливки:

Проверяем работоспособность:



3. Добавляем сбор статистики.



▼ Местоположение и размер

Уровень:

X:       Ширина:

Y:       Высота:

▼ Легенда

☒ Отображать легенду

Высота:

Цвет текста:

Расположение: ☒ ☐ ☐ ☐

▼ Область диаграммы

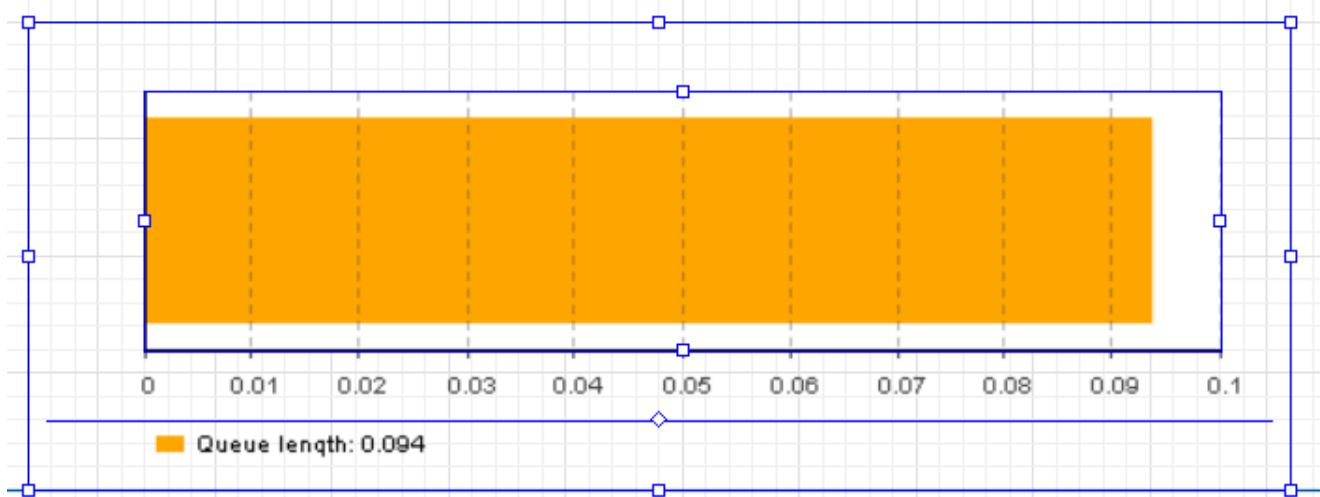
Смещение по оси X:       Ширина:

Смещение по оси Y:       Высота:

Цвет фона:

Цвет границы:

Добавляем ещё одну диаграмму:



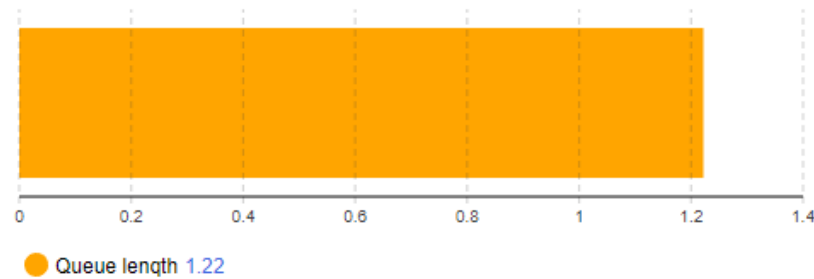
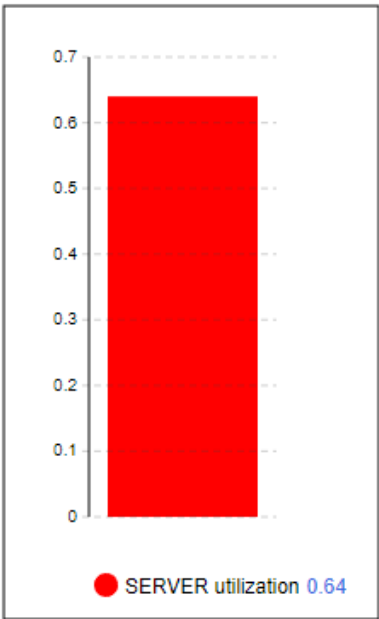
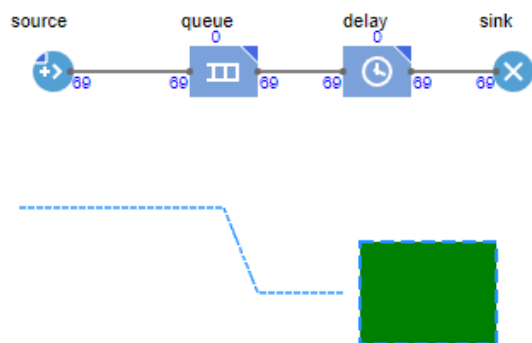
▼ Данные

Заголовок:

Цвет:

Значение:

Проверка работоспособности:



4. Доработка модели.

queue - Queue

Имя:

queue

☐ Исключить

Вместимость:

=

5

Максимальная вместимость:

=

Место агентов:

=

path

▼ Специфические

Очереди:

=

FIFO

Разрешить уход по таймауту:

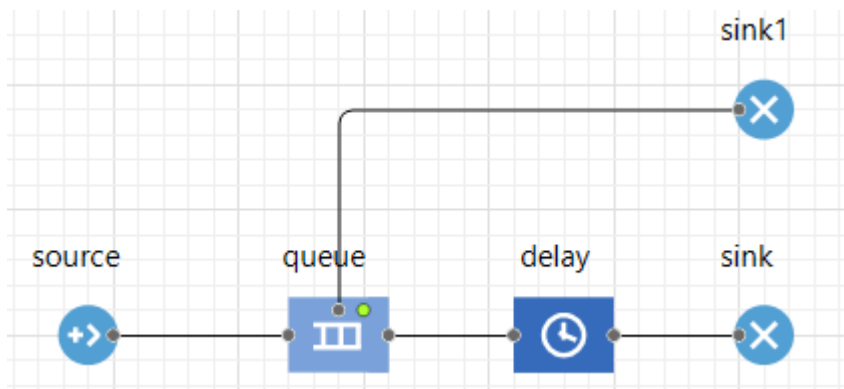
=

☐

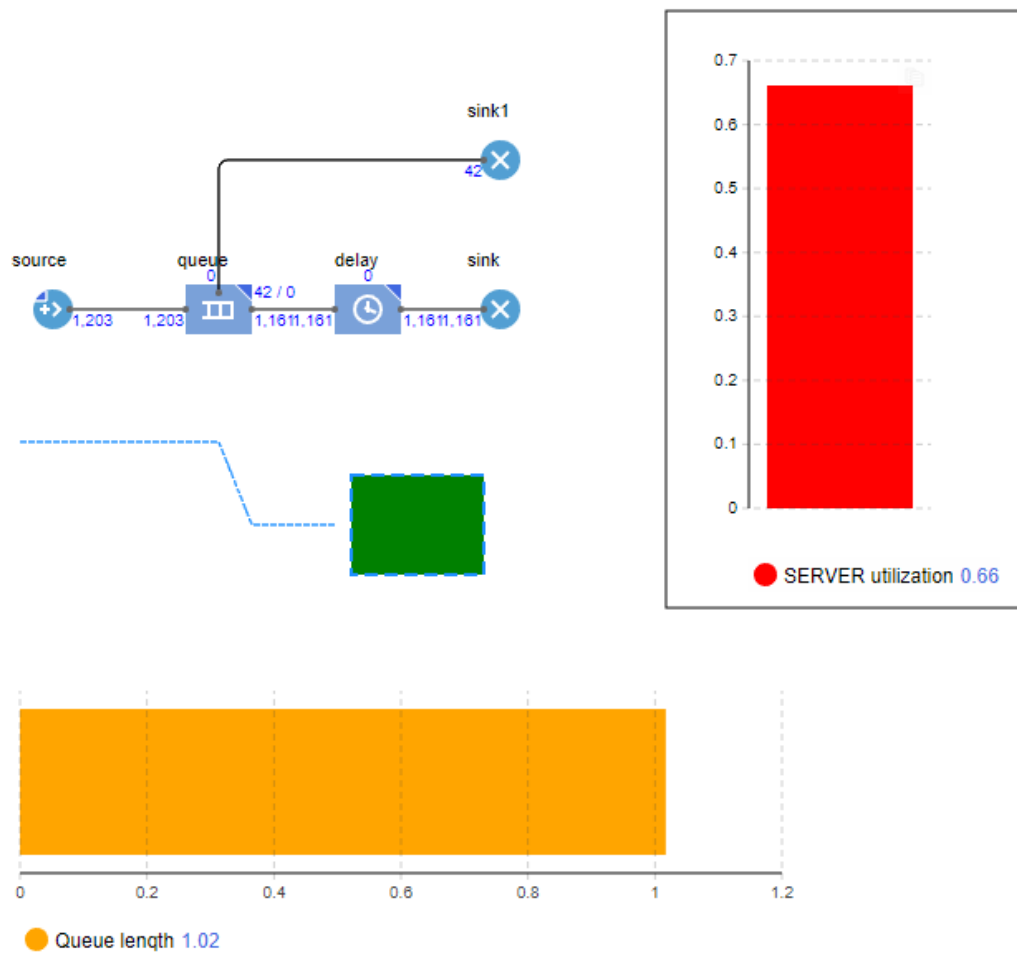
Разрешить вытеснение:

=

☒



Проверяем работоспособность:



5. Создадим новый тип заявки Inquiry и добавим в него данные диаграммы.



### time\_obrabotki - Данные гистограммы

Имя:

☐ Исключить

Видимость: ☒ да

Значение:

Кол-во интервалов:

### ver\_obrabotki - Данные гистограммы

Имя:

☐ Исключить

Видимость: ☒ да

Значение:

Кол-во интервалов:

Изменим тип заявки в source:

**Агент**

Новый агент: Inquiry

**Действия**

До прибытия:

При подходе к выходу:

При выходе: agent.time\_vxod=time();

Выполним такие же замены в других элементах системы.

Добавим в sink следующие коды:

```
time_obrabotki.add(time()-agent.time_vxod); agent.col_vixod=sink.count();
agent.col_vxod=source.count();ver_obrabotki.add(agent.col_vixod/ agent.col_vxod);
```

**Свойства**


**sink - Sink**

Имя:  ☒ Отображать имя

**Действия**

При входе: time\_obrabotki.add(time()-agent.time\_vxod); agent

Создадим для элемента delay параметр time\_mean с начальным значением 180 и свяжем параметр с бегунком:

 **time\_mean - Параметр**

Имя:

time\_mean


Видимость:

☒ да


Тип:

double

Значение по умолчанию:



180

 **delay - Delay**


Имя:

delay

☒ Отобра


☐ Исключить

Тип задержки:




☒ Определенное время  
☐ До вызова функции stopDelay()

Время задержки:



exponential(1/time\_mean)

 **slider - Бегунок**

Имя:

slider


☒ Отображается на верхнем агенте


☐ Блокировать

Ориентация:

☐ Вертикальная ☒ Гори.

☒ Связать с:



 time\_mean


Минимальное значение:

1

Максимальное значение:

300

Также добавим бегунок для контроля ёмкости очереди:

 **slider1 - Бегунок**

Имя:

slider1


☒ Отображается на верхнем агенте

☐ Блокирова

Ориентация:


☐ Вертикальная ☒ Гор

☒ Связать с:



queue

Параметр:



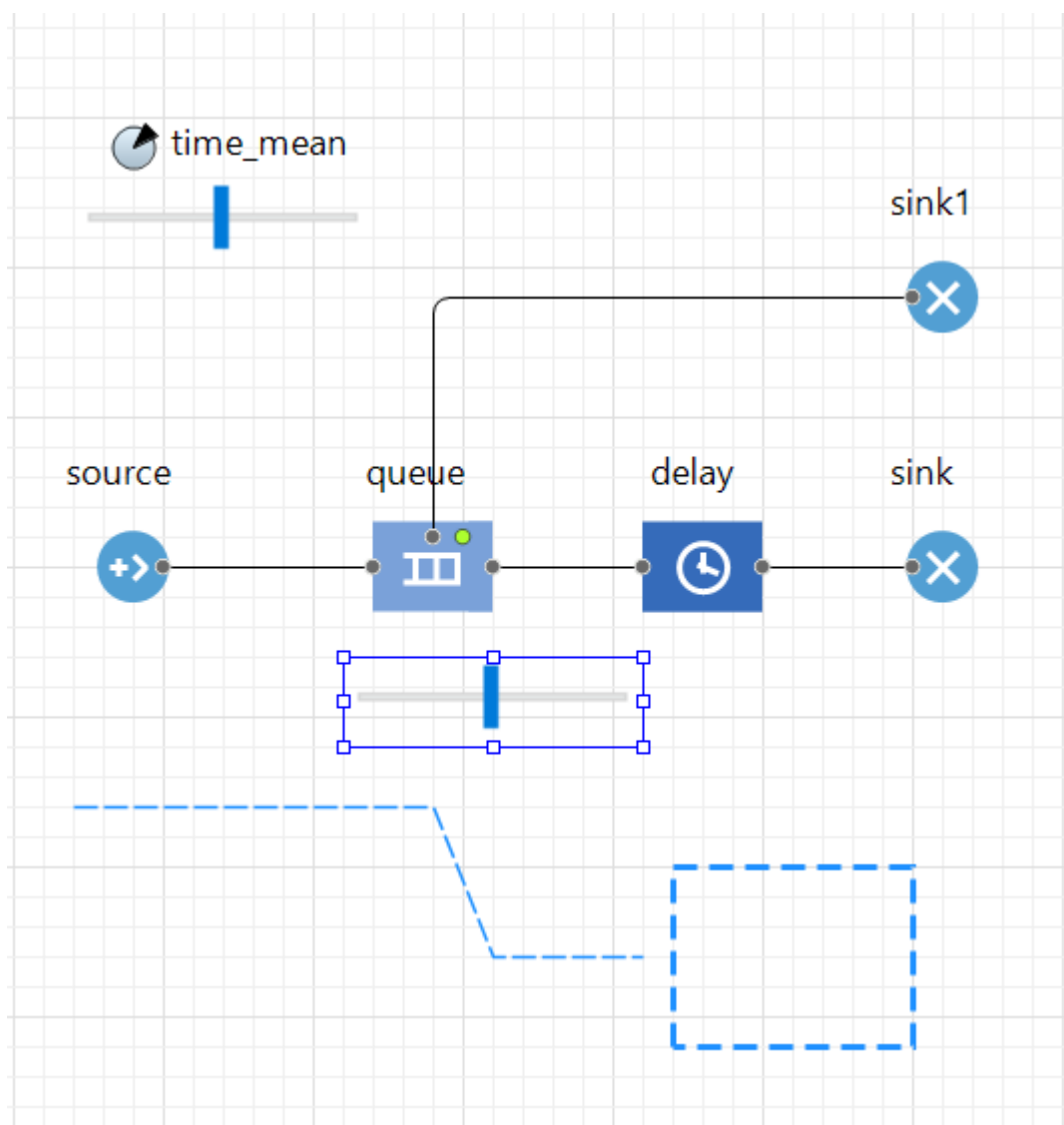
capacity

Минимальное значение:

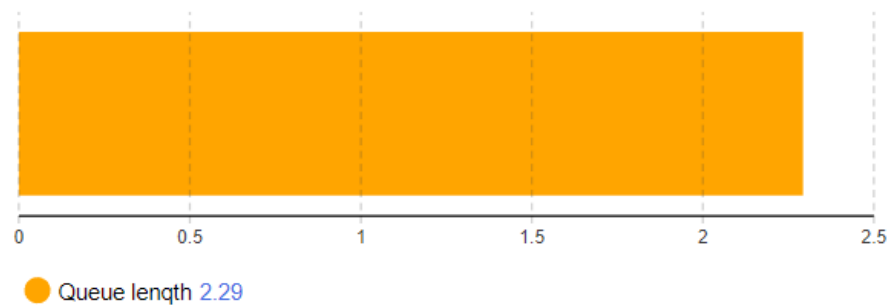
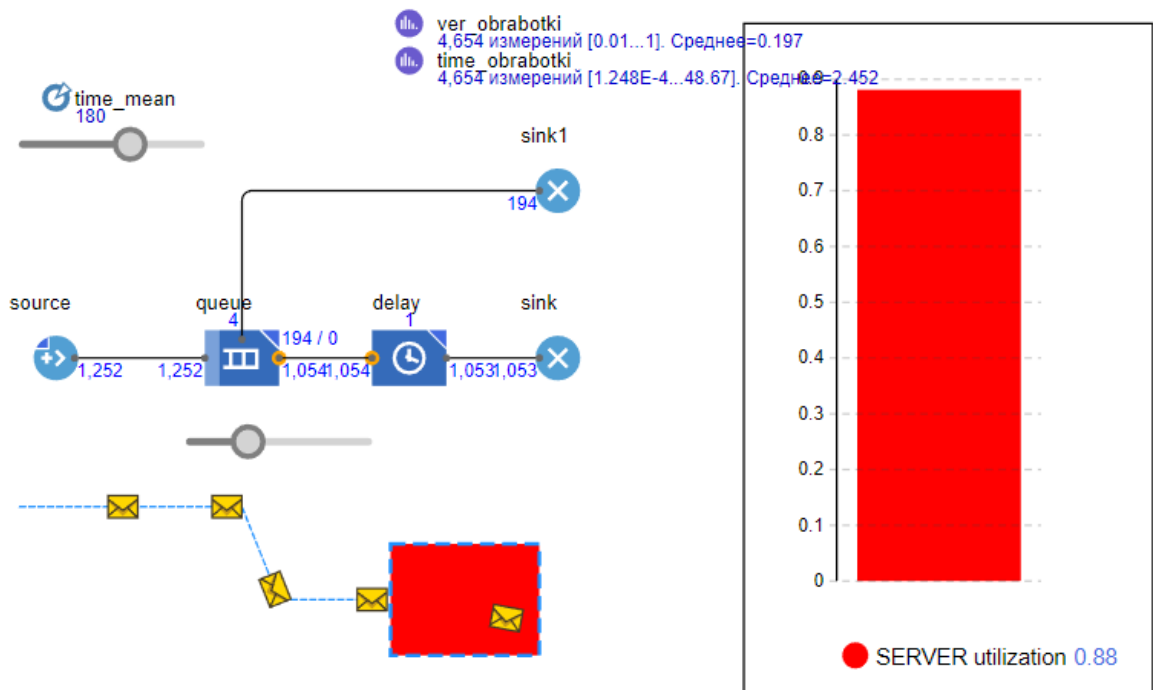
0

Максимальное значение:

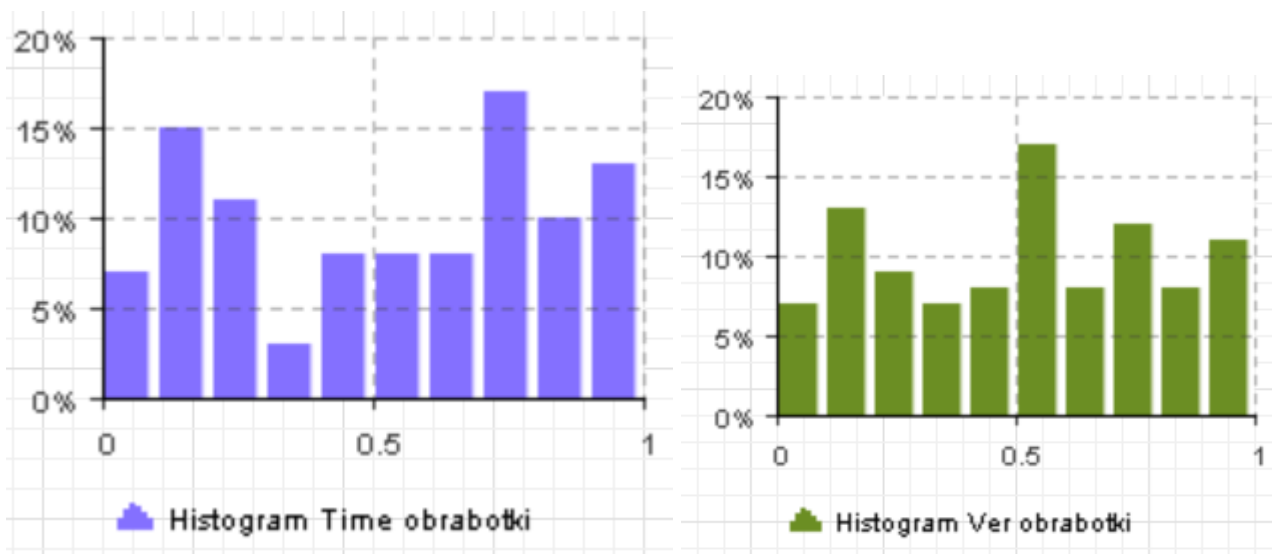
15



Проверка работоспособности:



6. Добавим две гистограммы:



Данные

Заголовок: Histogram Time obrabotki

Данные: time\_obrabotki

Цвет плотности вер-ти: lightSlateBlue

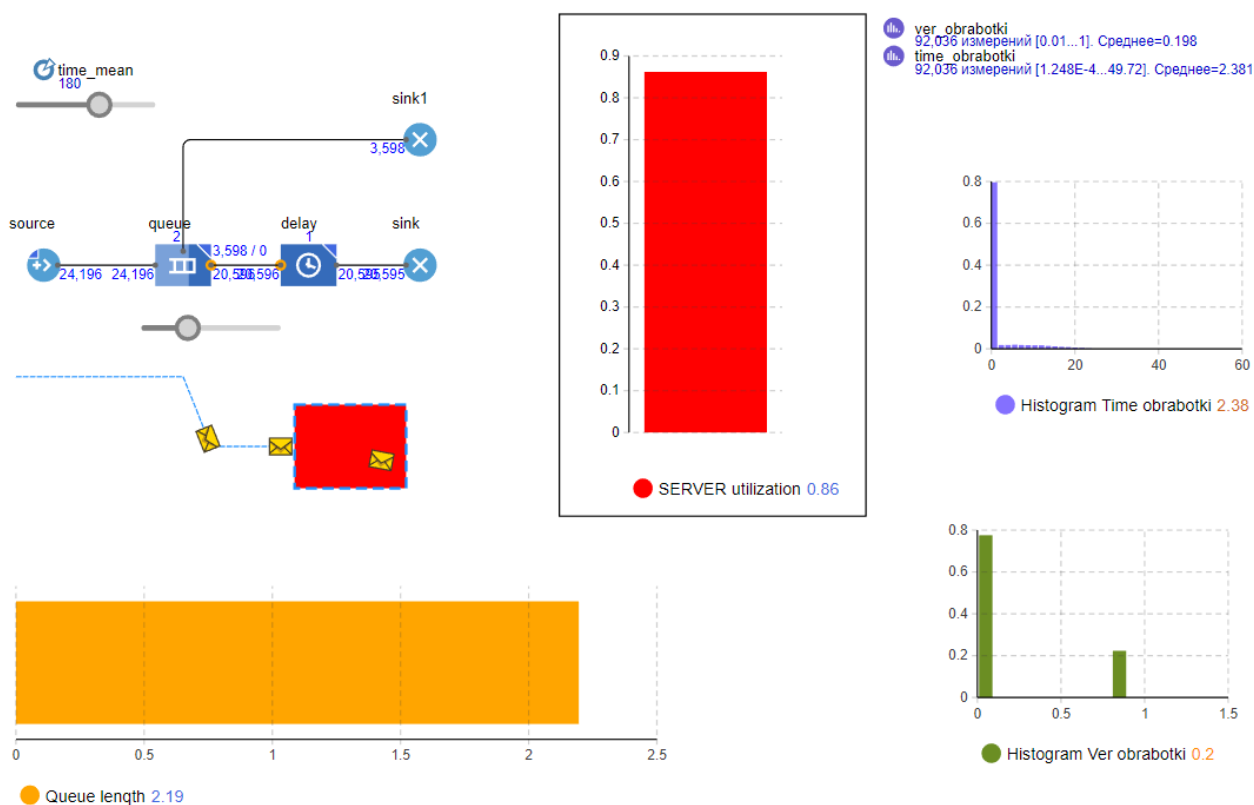
Данные

Заголовок: Histogram Ver obrabotki

Данные: ver\_obrabotki

Цвет плотности вер-ти: oliveDrab

Проверка работоспособности:



### 3. Выводы

В ходе проведённой работы была построена модель сервера. Анализируя работу построенной модели, можно сделать вывод, что оптимальные параметры не всегда удаётся предугадать. Поэтому, модель должна обладать инструментами контроля параметров.