



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ

Кафедра программных систем

Дисциплина
Моделирование информационных процессов и систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе на тему
«МОДЕЛИРОВАНИЕ МНОГОТЕРМИНАЛЬНОЙ
ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ»

Вариант 15

Студент _____ Лалин К.С., группа № 6303-020302D

Руководитель _____ Баландин А.В.

Самара 2024

Содержание

| | | |
|-------|---|----|
| 1 | Описание исследуемой системы как предмета моделирования | 3 |
| 1.1 | Описание МВС..... | 3 |
| 1.2 | Стратегия сервера..... | 4 |
| 2 | Описание целей моделирования..... | 6 |
| 3 | Разработка математической модели | 7 |
| 4 | Формализация предмета моделирования | 8 |
| 4.1 | Параметры потока транзакции | 8 |
| 4.2 | Параметры сервера | 9 |
| 5 | Формализация целей моделирования | 11 |
| 5.1 | Критерии оценки потока транзакции | 11 |
| 5.1.1 | Время ожидания результата выполнения задания i -ой ТС | 11 |
| 5.1.2 | Производительность i -ой ТС..... | 11 |
| 5.1.3 | Эффективность работы i -ой ТС | 12 |
| 5.2 | Критерии оценки сервера | 12 |
| 5.2.1 | Абсолютная пропускная способность сервера | 12 |
| 5.2.2 | Эффективность загрузки сервера | 13 |
| 6 | Разработка имитационной модели | 14 |
| 6.1 | Описание агента..... | 14 |
| 6.2 | Описание имитационной модели..... | 15 |
| 7 | Оценка адекватности модели..... | 23 |
| 8 | Статистическая оценка характеристик системы | 26 |
| 8.1 | Пропускная способность сервера | 26 |
| 8.2 | Эффективность работы сервера | 26 |
| 8.3 | Производительность работы i -ой ТС | 27 |
| 8.4 | Эффективность работы i -ой ТС | 28 |
| 8.5 | Время ожидания результата выполнения i -ой ТС | 29 |
| 8.6 | Время ожидания заданием i -ой ТС начала выполнения | 30 |
| 9 | Эксперименты с моделью и оптимизация параметров системы | 32 |
| 10 | Итоговые результаты работы | 38 |
| 11 | Список использованных источников..... | 41 |

1 Описание исследуемой системы как предмета моделирования

1.1 Описание МВС

В данной курсовой работе предметом моделирования является многотерминальная вычислительная система (МВС) на базе локальной вычислительной сети, представленной на рисунке 1, которая состоит из центрального компьютера (сервера) и взаимодействующих с ним в интерактивном режиме терминальных станций (ТС). МВС работает под управлением сетевой многопроцессорной операционной системы, обеспечивающей взаимодействие между процессами-клиентами ТС, распределёнными по узлам локальной сети, и процессом-сервером посредством обмена сообщениями.

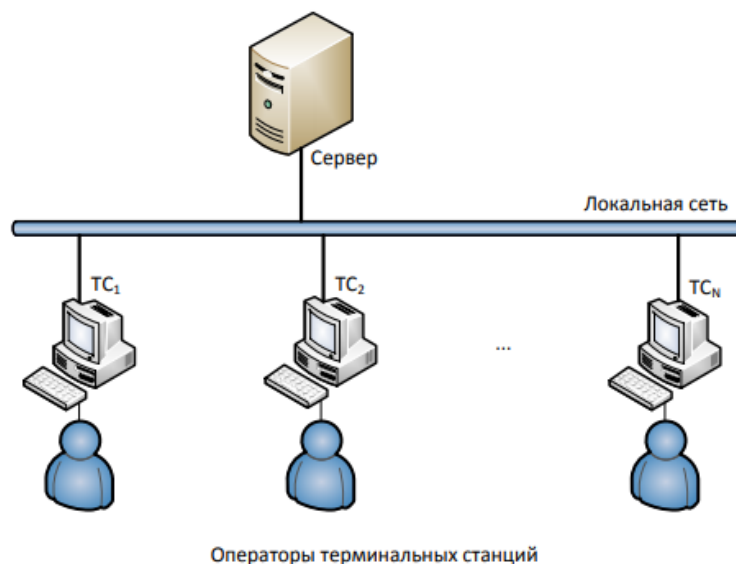


Рисунок 1 – Многотерминальная вычислительная сеть

Терминальные станции используются операторами для подготовки вычислительных заданий, отправляемых серверу для выполнения. Подготовленное оператором ТС вычислительное задание отправляется серверу для выполнения, после чего оператор ожидает получение результатов. Результаты выполненного вычислительного задания сервер отправляет на ТС оператора. После получения результатов и их анализа оператор подготавливает и отправляет серверу очередное задание. Далее всё повторяется.

Вычислительное задание формируется оператором на терминальной станции посредством текстового редактора в виде командных строк, предназначенных для их последовательного выполнения командным интерпретатором сервера. Оператор, завершив подготовку задания, автоматически инициирует на ТС программный процесс-клиент, реализующий транзакцию отправки задания по локальной сети серверу в виде текстового сообщения и приёма от сервера результатов выполнения. При этом процесс, реализующий транзакцию, выдаёт на экран оператора ТС сообщения о ходе её выполнения. Если сервер занят выполнением некоторого ранее принятого задания, то ТС ожидает начала приёма задания сервером.

1.2 Стратегия сервера

Сервер может реализовывать различные варианты стратегий учёта инициируемых ТС транзакций и мест нахождения заданий, ожидающих выполнения сервером. Стратегия сервера данного в задании варианта представлены ниже в таблице 1:

Таблица 1 – Стратегия сервера по обслуживанию транзакции

| Вариант № | Стратегия сервера по обслуживанию транзакции |
|------------------|--|
| 1 | 2 |
| <i>Вариант 5</i> | Сервер не имеет накопителя заданий, $N=0$. Подготовленное ТС скрипт-сообщение сохраняется на терминальной станции, а ТС выставляет флаг запроса обслуживания готового скрипт-сообщения. Освободившийся сервер по кругу тестирует ТС на наличие ожидающего обслуживания скрипт-сообщения. Если сервер выявил очередную такую ТС, то он сообщает ей о начале обслуживания, принимает скрипт-сообщение и сразу начинает выполнение задания. Результаты выполнения поступившего на обработку задания в виде |

Продолжение таблицы 1

| 1 | 2 |
|---|---|
| | отправляются соответствующей ТС, и сервер завершает транзакцию её обслуживания. После этого сервер переходит к тестированию наличия запроса на обслуживание от следующей по порядку ТС. |

Время выполнения сервером любого принятого на обработку вычислительного задания, складывается из времён последовательного выполнения сервером команд задания. Время выполнения сервером команды любого задания – τ_{cmd} , является случайным в диапазоне $[\tau_{cmd}^{min}, \tau_{cmd}^{max}]$.

Количество сформированных оператором в задании команд - k_{cmd} , является случайным в диапазоне $[k_{cmd}^{min}, k_{cmd}^{max}]$.

Среднее значение τ_{srv} определяется производительностью сервера – средним количеством команд, которые сервер обрабатывает в единицу времени.

Завершив выполнение очередного задания, сервер отправляет процессу-клиенту соответствующей ТС сообщения с результатами, которые выдаются на экран оператору. Получив результаты выполненного задания и осуществив их анализ, оператор приступает к подготовке следующего задания, после чего цикл подготовки оператором очередного задания и запуск транзакции по его выполнению повторяется.

Время анализа оператором полученного результата выполнения задания укладывается в диапазон $[\tau_{res}^{min}, \tau_{res}^{max}]$.

2 Описание целей моделирования

Целью моделирования МВС является нахождение оптимальной производительности сервера, понимая под производительностью среднее количество команд, выполняемых сервером в единицу времени, обеспечивающей его возможную предельную загрузку выполнением заданий ТС, при условии, что всем операторам ТС обеспечивается работа в режиме on-line. А именно, вероятность того, что среднее время ожидания оператором любой терминальной станции результата обработки задания не превысит 5 мин не должна быть ниже 0.9.

Для построенной модели необходимо найти:

- оптимальную производительность сервера;
- среднее время ожидания оператором i -ой ТС результата выполнения задания;
- долю, которую в начатой i -ой ТС транзакции составляет непосредственно время обработки сервером задания;
- долю занятости сервера выполнением вычислительных заданий от всего времени МВС (эффективность загрузки сервера);
- производительность оператора i -ой ТС – среднее количество результатов, которое оператор получает от сервера в единицу времени.

3 Разработка математической модели

В таблице 2 представлены параметры системы согласно заданию рассматриваемого варианта.

| ПАРАМЕТР | ЗНАЧЕНИЕ |
|---|---|
| 1 | 2 |
| Количество ТС, шт. | 3 |
| Среднее количество команд в задании $M[k_{cmd}]$, шт. | 17 |
| Закон распределения $F(k_{cmd})$ | Эрланга |
| Среднее время формирования командной строки $M[\tau_{cmd}]$, сек | 28 |
| Закон распределения $F(\tau_{cmd})$ | Треугольное |
| Среднее время анализа результата $M[\tau_{res}]$, сек | 180 |
| Закон распределения $F(\tau_{res})$ | Нормальное |
| Среднее время выполнения команды $M[\tau_{srv}]$, сек | 2,8 |
| Закон распределения $F(\tau_{srv})$ | Равномерное |
| Вариант стратегии сервера | Стратегия № 5. Сервер не имеет накопителя заданий. Подготовленное ТС скрипт-сообщение сохраняется на терминальной станции. Освободившийся сервер продолжает тестировать "готовые" ТС по кругу, начиная с ТС, следующей по порядку за только что обслуженной. |

4 Формализация предмета моделирования

На основе семантической модели можно выделить следующие параметры МВС как системы обслуживания транзакций.

4.1 Параметры потока транзакции

k_{cmd} – количество команд в задании – случайное натуральное число, которое находится в диапазоне $[k_{cmd}^{min}, k_{cmd}^{max}]$.

τ_{cmd} – время формирования одной команды – экзогенный случайный параметр, значение которого находится в диапазоне $[\tau_{cmd}^{min}, \tau_{cmd}^{max}]$.

τ_{res} – время анализа оператором полученного результата задания – экзогенный случайный параметр в диапазоне $[\tau_{res}^{min}, \tau_{res}^{max}]$.

$\tau_{task}(k_{cmd}, \tau_{cmd})$ – время формирования задания - эндогенный случайный параметр, который зависит от количества сформированных в задании команд k_{cmd} и от τ_{cmd} . $\tau_{task}(k_{cmd}, \tau_{cmd}) = \sum_{i=1}^{k_{cmd}} \tau_{cmd}^i$. Среднее время формирования задания будет равно $M[\tau_{task}] = M[\sum_{i=1}^{k_{cmd}} \tau_{cmd}^i]$.

τ_{wait} – время ожидания результата оператором ТС - непрерывная случайная величина, зависящая от времени обработки сервером одного задания и времени пребывания транзакции в серверном накопителе.

Тогда интервал времени инициирования транзакций оператором ТС:

$$\tau_{TC} = \sum_{i=1}^{k_{cmd}} \tau_{cmd}^i + \tau_{wait} + \tau_{res}.$$

Интенсивность потока транзакций зависит от среднего значения интервала поступления заявок: $M[\tau_{TC}] = M[\sum_{i=1}^{k_{cmd}} \tau_{cmd}^i + \tau_{wait} + \tau_{res}]$.

Положим $k_{cmd}, \tau_{cmd}, \tau_{res}$ независимыми, то есть в режиме работы МВС их математическое ожидание не зависит от времени. Тогда в установившемся режиме работы МВС математическое ожидание эндогенного случайного параметра τ_{wait} для некоторой выделенной ТС также не будет зависеть от времени, а следовательно:

$$M[\tau_{TC}] = M[\tau_{task}] + M[\tau_{wait}] + M[\tau_{res}].$$

Положим, что среднее значение случайного времени выполнения сервером любой команды любого задания любой ТС одно и то же - $M[\tau_{serv}]$, а также среднее время анализа результата оператором – одно и то же:

$$M[\tau_{res}^i] = M[\tau_{res}^j] = M[\tau_{res}], i, j \in [1, N].$$

А вот случайное количество команд, формируемое в задании операторами разных ТС, и, следовательно, среднее время ожидания ответа от сервера в установившемся режиме работы МВС будут различными для операторов разных ТС. Обозначим интенсивность потока транзакций от i -ой ТС как λ_i^{TC} , тогда при условии нулевого количества отказов, она будет равна:

$$\lambda_i^{TC} = \frac{1}{M[\tau_i^{TC}]}.$$

Если положить, что в установившемся режиме работы МВС средние значения всех случайных параметров не зависят от времени то:

$$\lambda_i^{TC} = \frac{1}{M[\tau_i^{TC}]} = \frac{1}{M\left[\sum_{i=1}^{k^{cmd}} \tau_{cmd}^i\right] + M[\tau_{wait}^i] + M[\tau_{res}]},$$

а интенсивность суммарного потока заявок, поступающих серверу со всех ТС будет:

$$\Lambda = \sum_{i=1}^N \lambda_i^{TC}, \text{ где } N - \text{ количество терминальных станций.}$$

Если положить, что все ТС формируют транзакции с одинаковой интенсивностью λ , то выражение примет вид: $\Lambda = N\lambda$.

Интенсивность потока транзакциями, поступающих серверу от всех ТС:

$$\Lambda = N * \frac{1}{M[\tau^{TC}]}.$$

4.2 Параметры сервера

Внутренним параметром сервера является его вычислительная производительность, характеризующаяся средним количеством команд вычислительного задания, выполняемых сервером в единицу времени $\mu \left(\frac{1}{c}\right)$.

Так как τ_{server} не зависит от того, к какому заданию и к какой ТС команды принадлежат, то:

$$\mu = \frac{1}{M[\tau_{srvr}]},$$

где τ_{srvr} – случайное время выполнения сервером любого задания, сек.

$$\tau_{srvr} = k_{cmd} * \tau_{server},$$

где k_{cmd} – количество сформированных в задании команд, $k_{cmd} = 1, 2, 3, \dots$;

τ_{server} – случайный непрерывный интервал времени выполнения сервером команды, сек.

Время выполнения команды от i -ой ТС:

$$\tau_{srv}^i = k_{cmd}^i * \tau_{server}.$$

Среднее время выполнения задания сервером $M[\tau_{srv}]$ (сек):

$$M[\tau_{srv}] = M[k_{cmd} * \tau_{server}].$$

Если положить, что в установившемся режиме работы МВС средние значения всех случайных параметров не зависят от времени то:

$$M[\tau_{srv}] = M[k_{cmd}] * M[\tau_{server}].$$

5 Формализация целей моделирования

τ_{serv} – эндогенный случайный параметр – среднее время выполнения сервером одного задания.

μ – внутренний параметр сервера – его производительность, которая характеризует среднее количество команд, выполняемых сервером в единицу времени.

Так как среднее время выполнения сервером одного задания - τ_{serv} , не зависит от того, заданию какой ТС команда принадлежит, то:

$$\mu = \frac{1}{M[\tau_{serv}]}$$

5.1 Критерии оценки потока транзакции

Для оценки потока транзакций рассмотрим следующие эндогенные параметры:

- W_i^{task} – время ожидания заданием i -ой ТС начала выполнения сервером;
- W_i^{rez} – время ожидания результата выполнения задания i -ой ТС;
- C_i^{TC} – производительность i -ой ТС (среднее количество заданий, формируемых в единицу времени);
- Q_i^{TC} – эффективность работы i -ой ТС (доля времени занятости подготовкой заданий и анализом результатов их выполнения).

5.1.1 Время ожидания результата выполнения задания i -ой ТС

W_i^{rez} – общее время нахождения транзакции на сервере, то есть время ожидания начала выполнения сервером и само время выполнения сервером.

$$\text{Тогда } W_i^{rez} = W_i^{task} + \tau_{srv}^i, i \in [1, N].$$

5.1.2 Производительность i -ой ТС

Определим производительность i -ой ТС C_i^{TC} как количество сформированных и обработанных заданий в единицу времени. Тогда

$$C_i^{TC} = \frac{(\text{сформированные}_i)}{t}, \text{ где}$$

сформированные_{*i*} – количество сформированных *i*-ой ТС заданий.

5.1.3 Эффективность работы *i*-ой ТС

Определим эффективность работы *i*-ой ТС Q_i^{TC} как отношение суммы времени формирования задания и анализа результата оператором к общему времени цикла. Тогда

$$Q_i^{TC} = \frac{(\tau_{CMD}^i + \tau_{res}^i)}{\tau_i^{TC}}, \text{ где}$$

τ_{CMD}^i – время формирования задания *i*-ой ТС, [сек];

τ_{res}^i – время анализа результата *i*-ой ТС, [сек];

τ_i^{TC} – общее время цикла транзакции, созданной *i*-ой ТС, [сек], $i \in [1, N]$.

5.2 Критерии оценки сервера

Для оценки работы сервера рассмотрим следующие эндогенные параметры:

- $C_{сервер}$ – абсолютная пропускная способность сервера;
- $Q_{сервер}$ – эффективность загрузки сервера.

5.2.1 Абсолютная пропускная способность сервера

Определим абсолютную пропускную способность сервера $C_{сервер}$ как среднее количество заданий, выполняемых сервером в единицу времени. Тогда,

$$C_{сервер} = \frac{\text{обработанные_задания}}{\text{время работы сервера}}, \text{ где}$$

время работы сервера – общее время обработки сервером заданий,

обработанные_задания – количество заданий, обработанных сервером.

$C_{отн_сервер}$ – относительная пропускная способность сервера:

$$C_{отн_сервер} = \frac{\text{обработанные_задания}}{\text{все_задания}}.$$

5.2.2 Эффективность загрузки сервера

Определим эффективность загрузки сервера $Q_{\text{сервер}}$ как отношение времени работы сервера к общему времени.

$$Q_{\text{сервер}} = \frac{\text{время работы сервера}}{t}.$$

Математическая модель будет иметь вид:

$$M_s = \otimes_s \left\{ \begin{array}{l} N = 3 \\ M[k_{cmd}] = 17 \\ M[\tau_{cmd}] = 28 \\ M[\tau_{rez}] = 180 \\ M[\tau_{server}] = 2.8 \\ \tau_{cmd} = \text{triangular}(24, 28, 32) \\ \tau_{rez} = \text{normal}(22.5, 80) \\ k_{cmd} = \text{erlang}(0.4, 17, 10) \\ \tau_{server} = \text{uniform}(2.52, 3.22) \\ M[\tau_{task}] = M[\sum_{i=1}^{k_{cmd}} \tau_{cmd}^i] \\ \tau_{TC} = \tau_{task} + \tau_{wait} + \tau_{res} \\ \tau_{TC} = t_{out}^{tr} \\ W_i^{res} = W_i^{task} + \tau_{srv}^i \\ W_i^{task} = \tau_{wait}^i \\ Q_i^{TC} = \frac{(\tau_{cmd}^i + \tau_{res}^i)}{\tau_i^{TC}} \\ \mu = \frac{1}{M[\tau_{srv}]} \\ \Lambda = \sum_{i=1}^N \lambda_i^{TC} \\ \lambda_i^{TC} = \frac{1}{M[\sum_{i=1}^{k_{cmd}} \tau_{cmd}^i] + M[\tau_{wait}] + M[\tau_{res}]} \\ C_{\text{сервер}} = \frac{\text{обработанные_задания}}{\text{время работы сервера}} \\ C_{\text{отн_сервер}} = \frac{\text{обработанные_задания}}{\text{все_задания}} \\ Q_{\text{сервер}} = \frac{\text{время работы сервера}}{t} \\ \text{обработанные_задания} = 1, 2, 3 \dots \\ \text{отправленные_задания} = 1, 2, 3 \dots \\ \text{сформированные}_i = 1, 2, 3 \dots \\ C_i^{TC} = \frac{(\text{сформированные}_i)}{t} \\ i \in [1, N] \\ t = \text{time}() \end{array} \right.$$

6 Разработка имитационной модели

Для определения характеристик оперативной обработки сервером вычислительных заданий и нахождения параметров системы, обеспечивающих операторам работу в режиме on-line, была разработана имитационная модель, представленная на рисунке 2, которая имитирует работу локальной вычислительной сети.

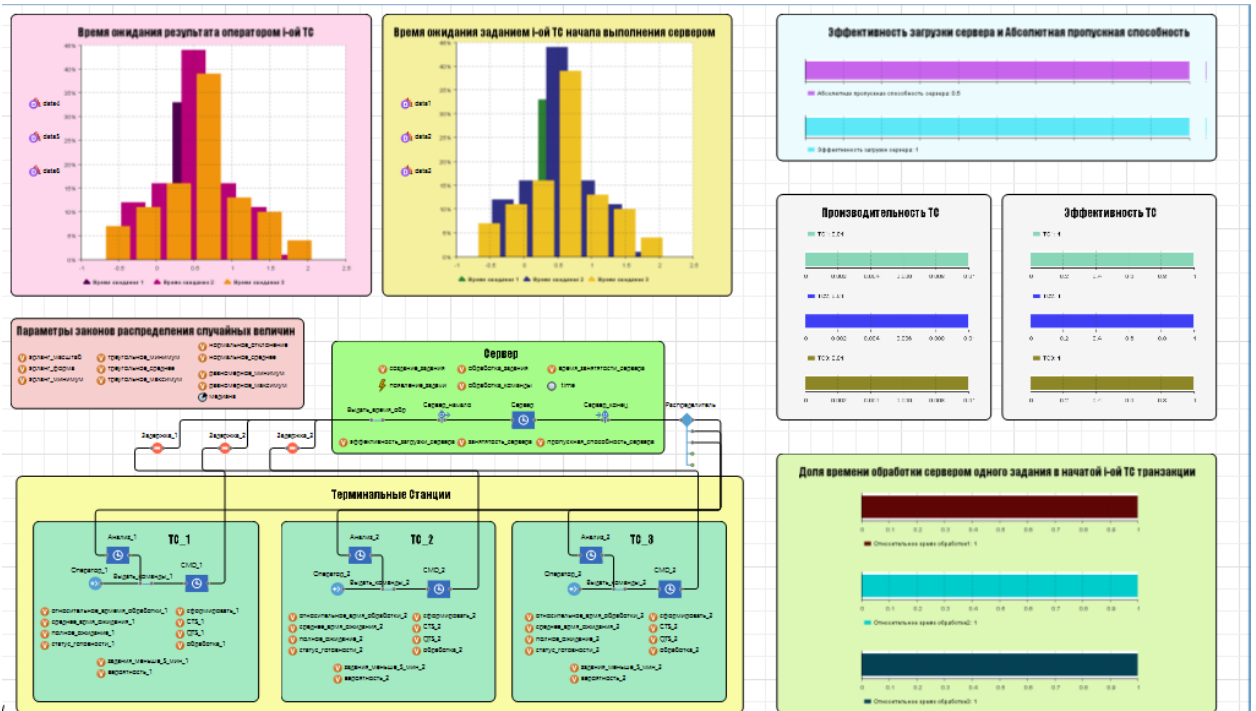


Рисунок 2 – Имитационная модель MVC

6.1 Описание агента

Для реализации работы модели был создан агент Агент_ТС. Хранимые им статистические параметры, представленные на рисунке 3:

- id – номер ТС, которая инициализировала данную транзакцию;
- k_{cmd} – количество командных строк в задании;
- t_{cmd} – время формирования оператором одного задания;
- t_{trans} – время ожидания транзакцией начала обработки сервером;
- t_{res} – время, затраченное на анализ ответа сервера;
- t_{serv} – время обработки задания сервером;
- t_{op} – общее время модельного цикла;

– t_{wait} – общее время ожидания оператором результата обработки задания сервером.

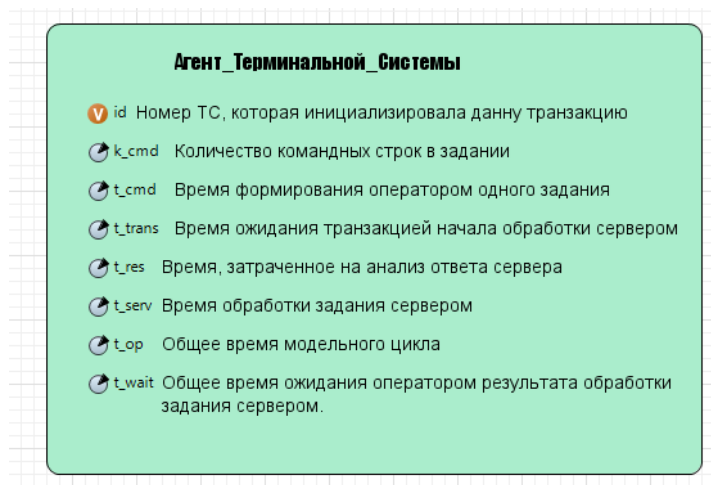


Рисунок 3 – Параметры агента Агент_ТС

6.2 Описание имитационной модели

Рассмотрим модельный цикл. В блоке Оператор_1 типа *Source* инициализируется Агент_ТС типа агент. При этом каждому агенту в переменную *id* присваивается номер ТС, для которого он создается, что видно из свойств Оператор_1 на рисунке 4 ниже.

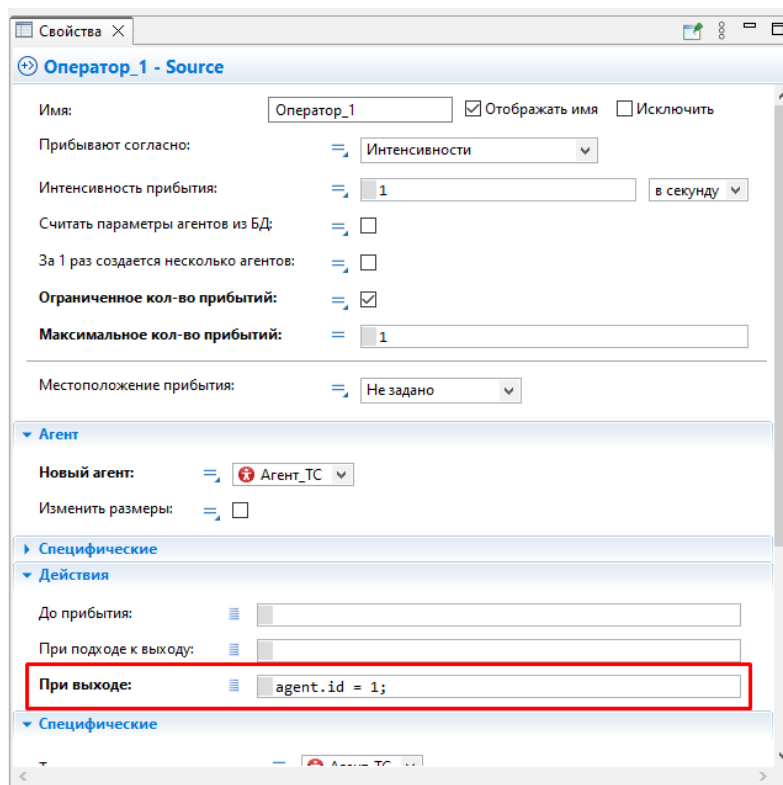


Рисунок 4 – Свойства блока под названием Оператор_1 типа Source

Затем агент попадает в блок под названием Выдать_команды_1 типа *Plain Transfer*, в котором при входе происходит присвоение агенту случайного количества команд по распределению Эрланга, а в цикле по количеству команд по треугольному распределению определяется t_{cmd} – общее время набора задания в терминале. Свойства блока Выдать_команды_1 приведены на рисунке 5.

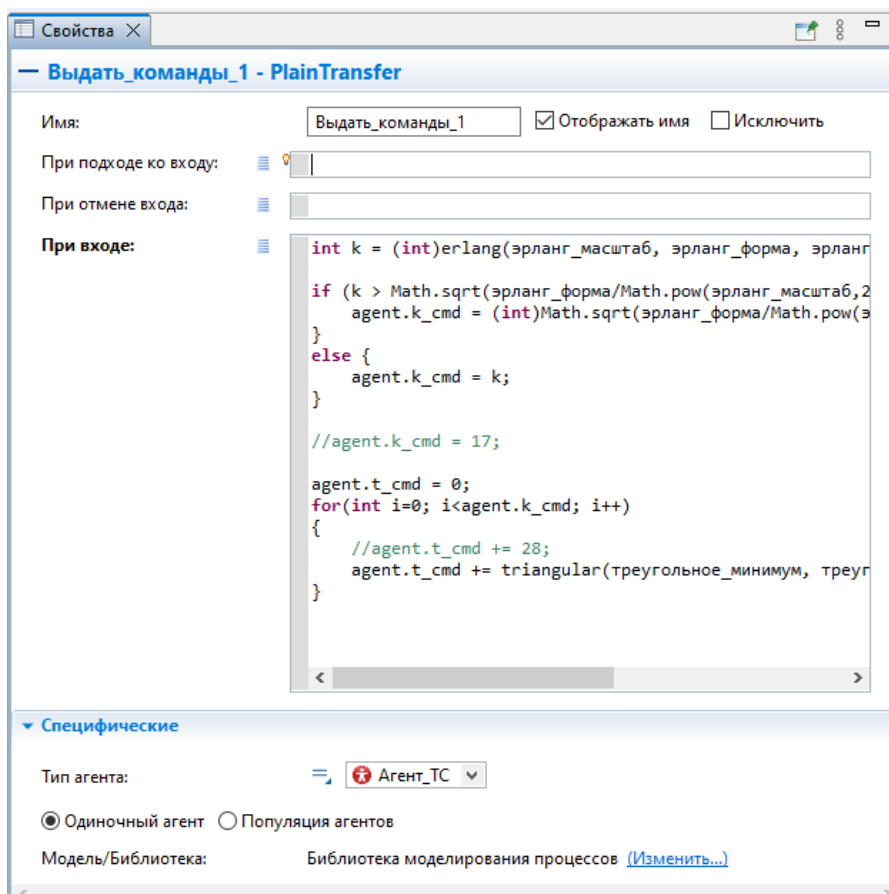


Рисунок 5 – Свойства блока под названием Выдать_команды_1 типа Plain Transfer

Затем агент попадает в блок *CMD_1* типа *Delay*. В нем происходит задержка агента на время формирования задания, а также подсчёт количества созданных ТС заданий и её производительность, также в этом блоке выставляется статус готовности задания для обработки сервером. Здесь же начинается отсчет времени ожидания начала обработки транзакции и общего времени ожидания оператором результата обработки задания сервером. Свойства блока *CMD_1* приведены на рисунке 6.

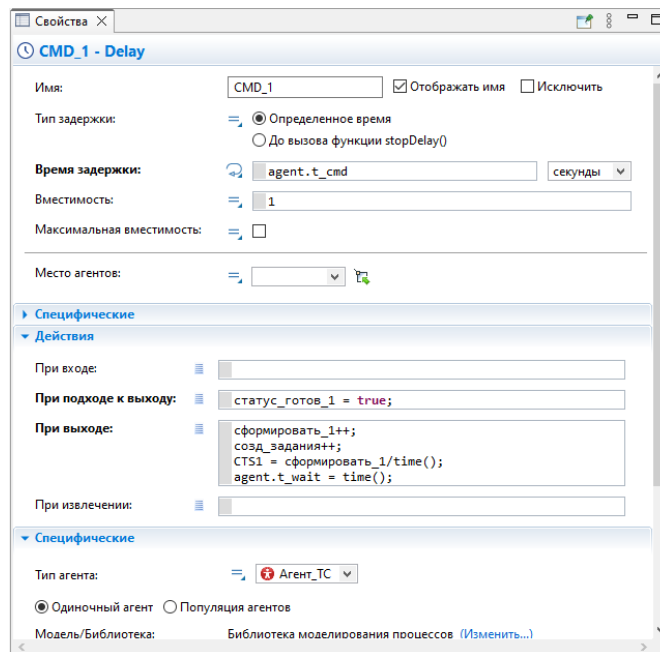


Рисунок 6 - Свойства блока под названием CMD_1 типа Delay

Далее агент проходит через блок Задержка_1 типа *Hold*, который разрешает отправлять задание, сформированное терминальной станцией, на сервер, но только в том случае, если сработало Событие появление_задачи, которое открыло блок Задержка_1 и агент смог пройти через него далее. Событие срабатывает только в том случае, если сервер не занят и одна из терминальных станций имеет готовое задание для сервера. При прохождении через блок Задержка_1, блок закрывается. Свойства блока Задержка_1 и События появление_задачи приведены соответственно на рисунке 7 и рисунке 8.

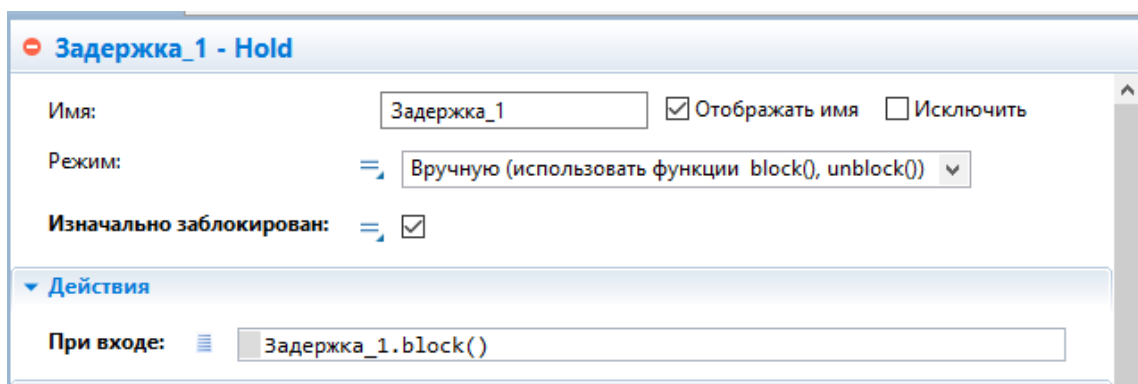


Рисунок 6 - Свойства блока под названием Задержка_1 типа Hold

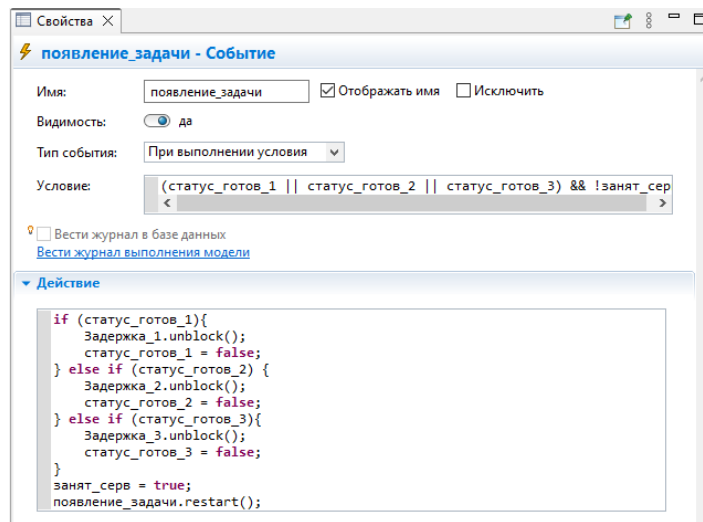


Рисунок 7 - Свойства События под названием появление_задачи

После прохождения блока Задержка_1 типа *Hold* агент попадает в блок Выдать_время_обработки типа *Plain Transfer*. В нем высчитывается время обработки команды сервером по равномерному распределению, в цикле находится общее время обработки команд сервером. Также здесь высчитывается время ожидания задачи с момента ее создания и до обработки сервером. Кроме того, здесь собирается статистика времени ожидания задания i -ой ТС начала выполнения сервером. Свойства блока Выдать_время_обработки приведены на рисунке 8.

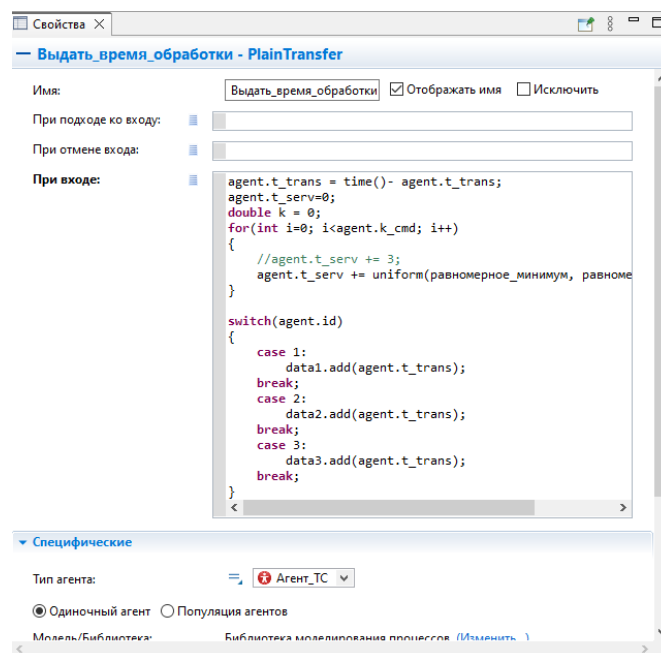


Рисунок 8 – Свойства блока под названием Выдать_время_обр типа Plain Transfer

Далее агент последовательно проходит через блок *TimeMeasureStart* с названием Сервер_начало. В нем автоматически замеряется время начала обработки сервером заданий. Свойства блока Сервер_начало приведены на рисунке 9.

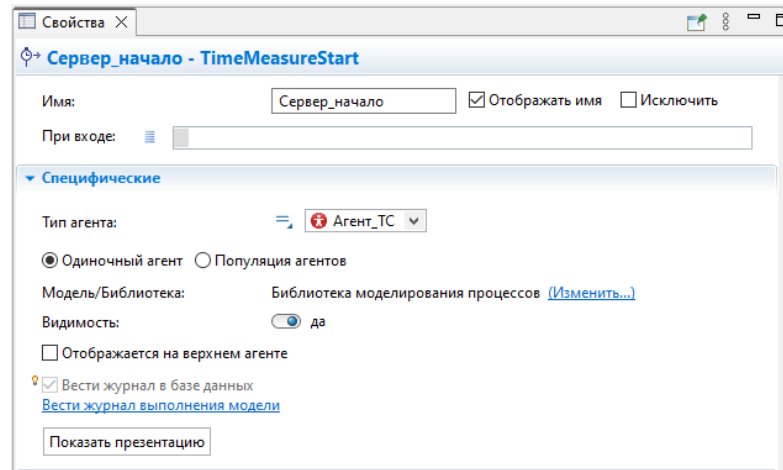


Рисунок 9 – Свойства блока под названием Сервер_начало типа TimeMeasureStart

Далее транзакция попадает в блок Сервер типа *Delay*, реализующий обработку задания сервером с задержкой, равной параметру t_{serv} . Здесь же изменяется параметр занятости сервера. Свойства блока Сервер приведены на рисунке 10.

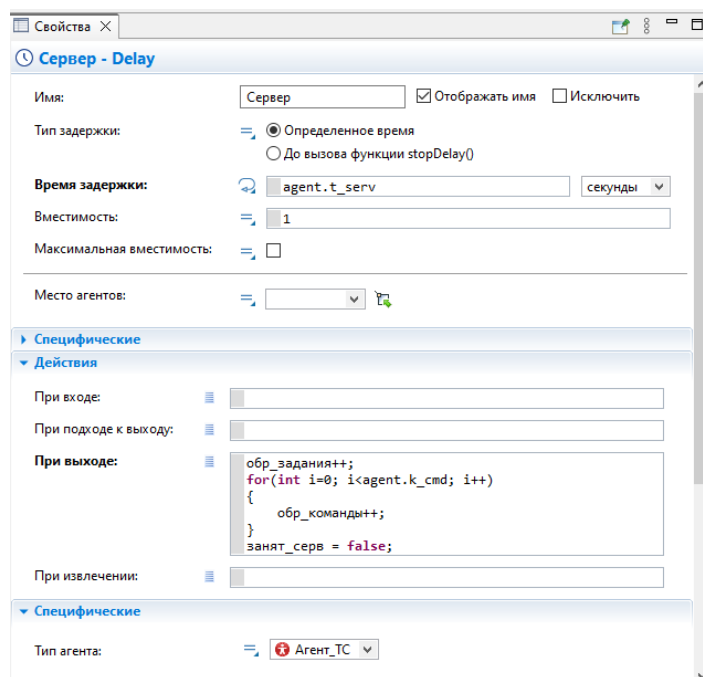


Рисунок 10 – Свойства блока под названием Сервер типа Delay

После окончания обработки на сервере агент попадает в блок *TimeMeasureEnd* с названием Сервер_конец. В нем автоматически замеряется время конца обработки сервером заданий. Свойства блока Сервер_конец приведены на рисунке 11.

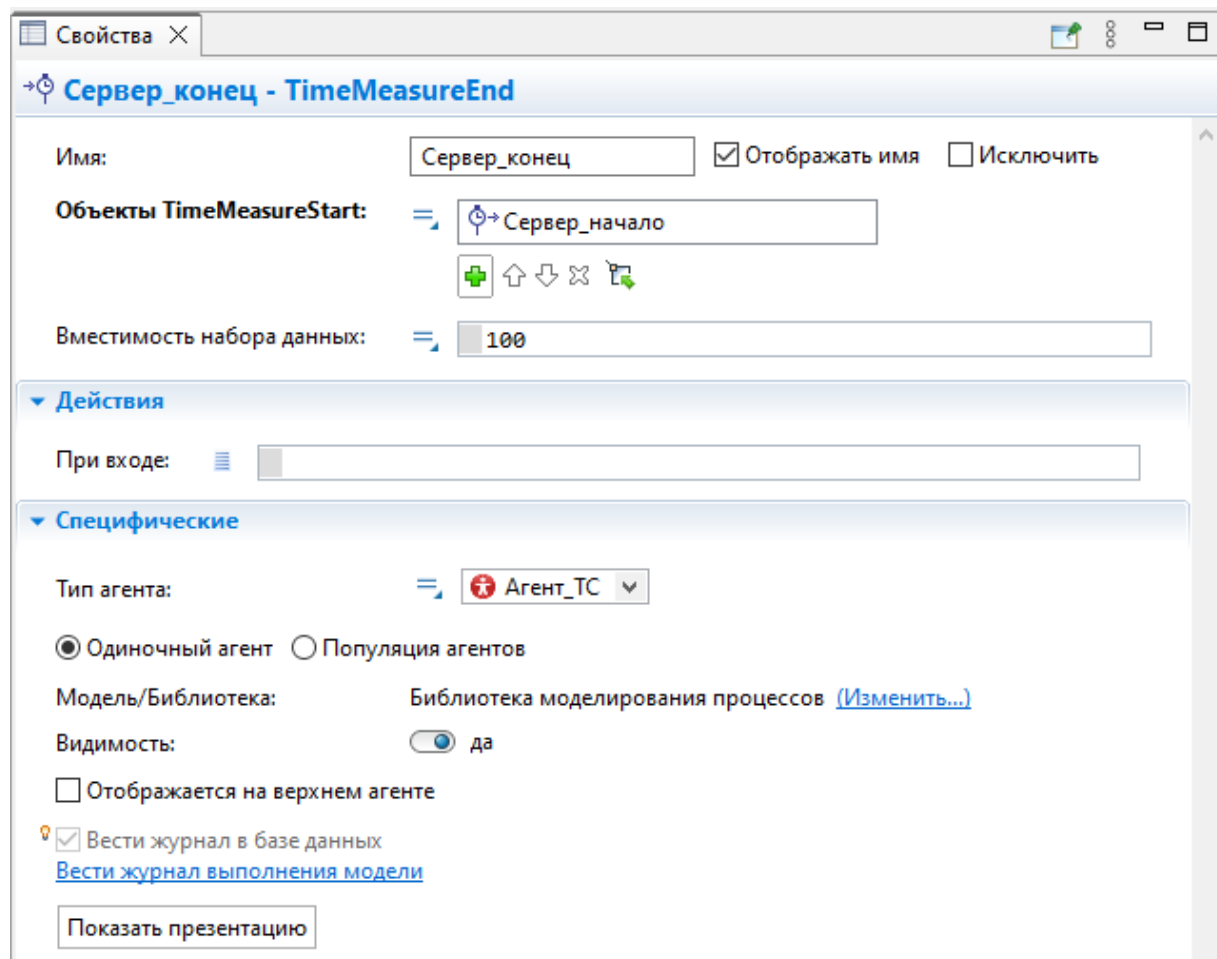


Рисунок 11 - Свойства блока под названием Сервер_конец типа TimeMeasureEnd

После окончания обработки на сервере агент попадает в блок Распределитель типа *SelectOutput5*, выполняющий функцию маршрутизатора и направляющий агента на ТС в соответствии со значением индекса *id* в блоки Анализ. Также в нем высчитывается эффективность сервера, абсолютная пропускная способность сервера, время занятости сервера и время анализа ответа сервера в соответствии с нормальным распределением. Свойства блока Распределитель приведены на рисунке 12.

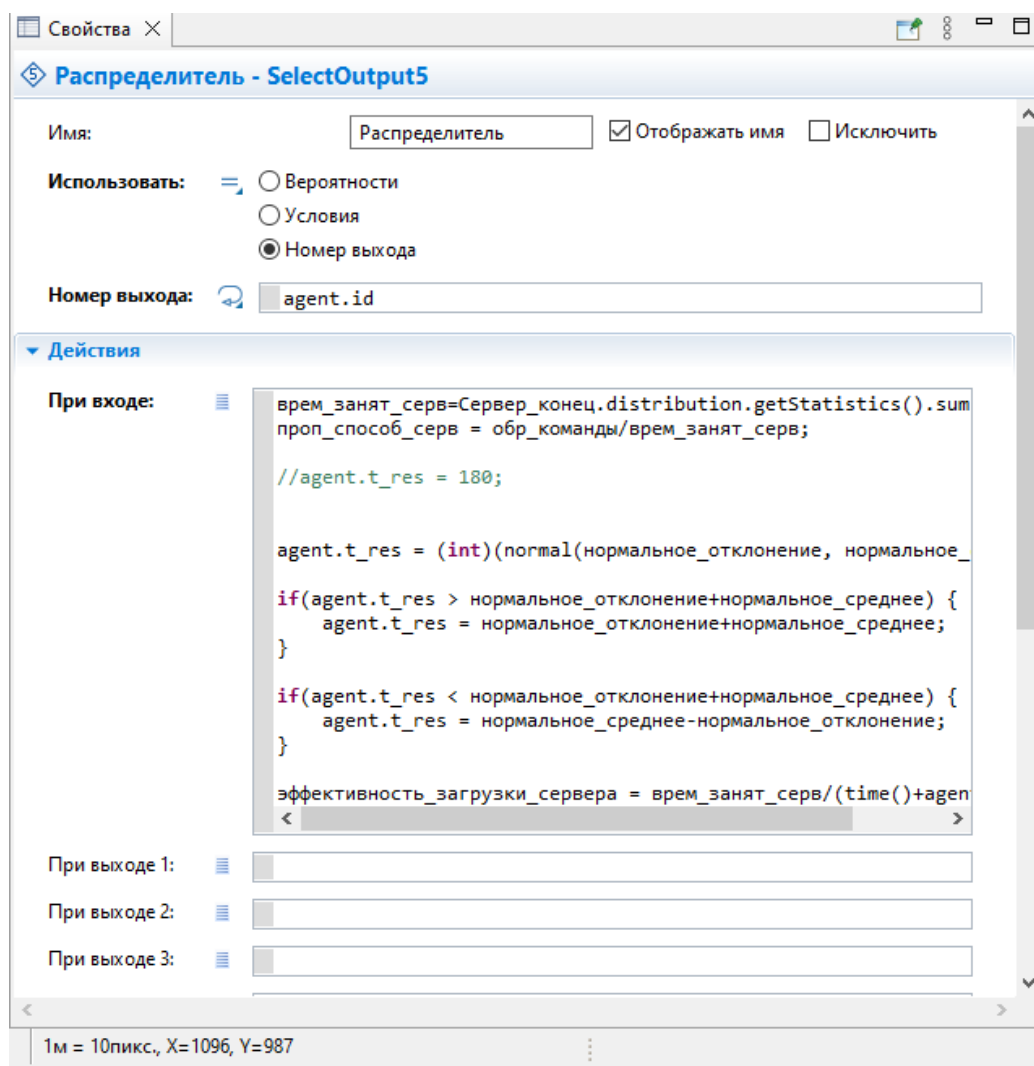


Рисунок 12 - Свойства блока под названием Распределитель типа Selectoutput5

Затем агент попадает в блок Анализ_1 типа *Delay*, где происходит имитация процесса анализа результатов оператором ТС. В нем происходит задержка на время t_{res} . А также высчитывается время ожидания результата, общее время прохождения транзакции и эффективность ТС. Также здесь происходит подсчет вероятности того, что среднее время ожидания оператором любой терминальной станции результата обработки задания не превысит 5 мин. Кроме того собирается статистика времени ожидания результата i -ой ТС Свойства блока Анализ_1 приведены на рисунке 13.

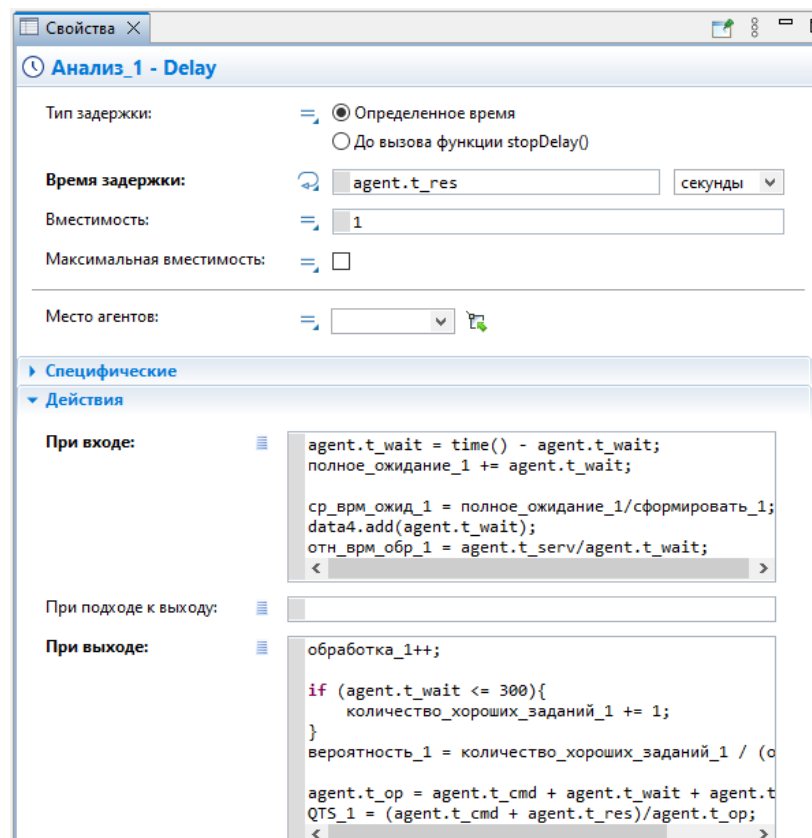


Рисунок 13 - Свойства блока под названием Анализ1 типа Delay

После обработки транзакции происходит инициализация новой транзакции и цикл повторяется. Также аналогичные действия происходят в других ТС, а именно в ТС – 2 и ТС – 3.

7 Оценка адекватности модели

Проверим адекватность модели, используя детерминированные параметры и только одну терминальную станцию. Пусть операторы создают задания, содержащие 17 команд, на ввод каждой команды в командной строке они тратят 28 секунд, сервер выполняет каждую команду 3 секунды, а на анализ у оператора уходит 180 секунд. Тогда:

- время ввода задания в командной строке равно $17 \cdot 28 = 476$ сек;
- время обработки транзакции сервером равно $17 \cdot 3 = 51$ сек;
- общее время цикла равно $476 + 51 + 180 = 707$ сек.

Согласно полученным значениям:

- эффективность загрузки сервера равна $\frac{51 \text{ сек}}{707 \text{ сек}} = 0.072$;

На рисунке 14 приведена диаграмма эффективности загрузки сервера, полученная с помощью модели. Вычисленное моделью значение равно полученному теоретическому значению.

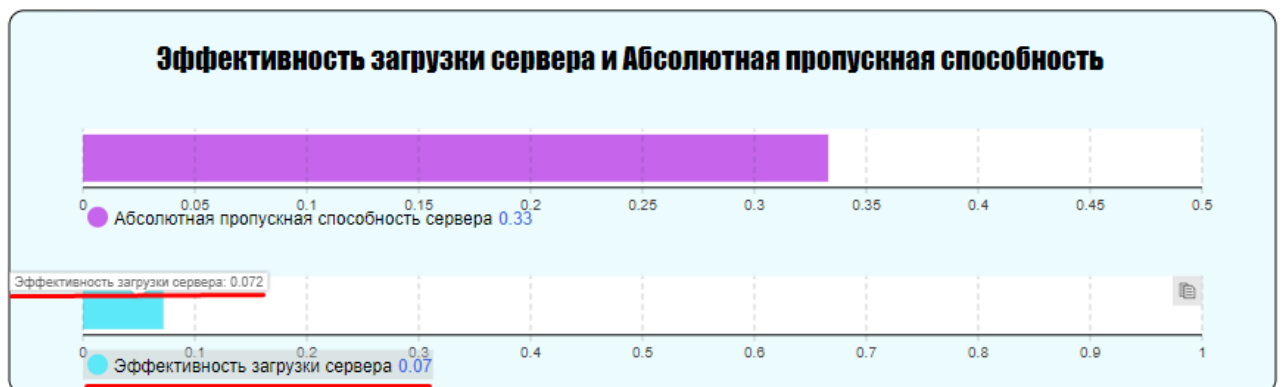


Рисунок 14 – Диаграмма эффективности загрузки сервера

- эффективность ТС равна $\frac{476 \text{ сек} + 180 \text{ сек}}{707 \text{ сек}} = 0.928$;

На рисунке 15 приведена диаграмма эффективности ТС, полученная с помощью модели. Вычисленное моделью значение равно полученному теоретическому значению. Входные параметры для каждой ТС одинаковы, поэтому значения эффективностей ТС совпадают.



Рисунок 15 – Диаграмма эффективностей загрузки ТС

– абсолютная пропускная способность сервера равна $\frac{1 \text{ команда}}{3 \text{ сек}} = 0.333 \frac{\text{команда}}{\text{сек}}$;

На рисунке 16 приведена диаграмма абсолютной пропускной способности сервера. Вычисленное моделью значение абсолютной пропускной способности равно полученному теоретическому значению.



Рисунок 16 – Диаграмма абсолютной пропускной способности сервера

– производительность ТС равна $\frac{1 \text{ заданий}}{707 \text{ сек}} \approx 0.0014 \frac{\text{заданий}}{\text{сек}}$;

На рисунке 17 приведена диаграмма производительности ТС, полученная с помощью модели. Вычисленное моделью значение равно полученному теоретическому значению. Входные параметры для каждой ТС одинаковы, поэтому значения производительностей ТС совпадают.



Рисунок 17 – Диаграмма производительностей ТС

– время ожидания результата оператором равно $17 \cdot 3 = 51$ сек.

На рисунке 18 приведена диаграмма времени ожидания результата оператором, полученная с помощью модели. Вычисленное моделью значение равно полученному теоретическому значению.



Рисунок 18 – Диаграмма времени ожидания результата оператором

На основании предоставленных результатов работы программы для детерминированных параметров можем прийти к выводу, что работа модели является адекватной.

8 Статистическая оценка характеристик системы

8.1 Пропускная способность сервера

Нахождение пропускной способности сервера, или среднего количества обрабатываемых сервером заданий в единицу времени, происходит по формуле

$$\text{производительность_сервера} = \frac{\text{обработанные_команды}}{\text{время_работы_сервера}}, \text{ где}$$

обработанные_команды – количество команд, обработанных сервером,

время_работы_сервера – текущее значение модельного времени работы сервера.

На рисунке 19 приведен результат вычисления данной выходной характеристики моделью.

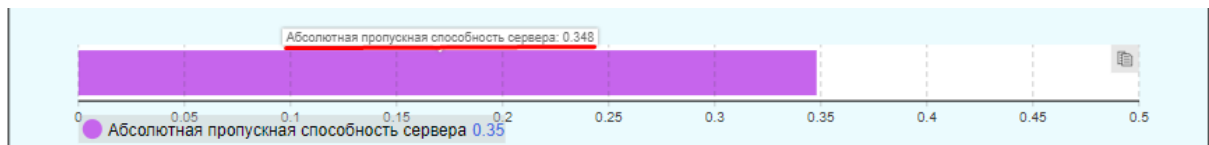


Рисунок 19 – Диаграмма абсолютной пропускной способности сервера

Для входных параметров, соответствующих заданию, значение $\text{производительность_сервера} = 0.348 \frac{\text{команд}}{\text{сек}}$.

8.2 Эффективность работы сервера

Эффективность загрузки сервера вычисляется в блоке Распределитель по формуле: $\text{Эффективность_загрузки_сервера} = \frac{\text{время_работы_сервера}}{\text{time}}$, где

время работы сервера – текущее значение времени работы сервера,

time – текущее значение модельного времени.

На рисунке 20 приведен результат вычисления эффективности загрузки сервера.

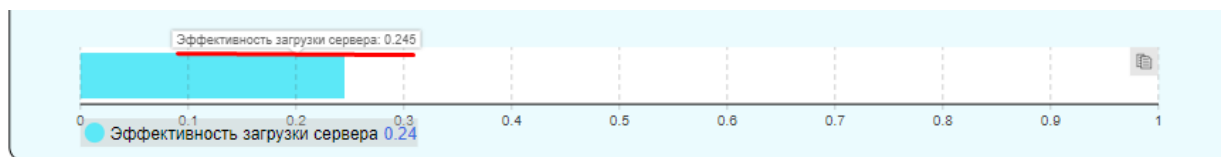


Рисунок 20 – Диаграмма эффективности загрузки сервера

Для входных параметров, соответствующих заданию, значение $\text{эффективность_загрузки_сервера} = 0.341$.

8.3 Производительность работы i -ой ТС

Производительность i -ой ТС CTS_i , $i \in [1,3]$, или количество составленных заданий в единицу времени, вычисляется в блоке CMD по формуле:

$$CTS_i = \frac{\text{сформ}_i}{\text{time}}, \text{ где}$$

сформировать_i – количество заданий, созданных i -ой ТС,

time – текущее значение модельного времени.

На рисунке 21 приведены результаты вычисления моделью производительностей ТС.



Рисунок 21 – Диаграммы производительностей ТС

Для входных параметров, соответствующих заданию, значениям производительностей ТС соответствуют:

- $CTS_1 = 0.002 \frac{\text{заданий}}{\text{сек}}$;
- $CTS_2 = 0.002 \frac{\text{заданий}}{\text{сек}}$;
- $CTS_3 = 0.002 \frac{\text{заданий}}{\text{сек}}$.

8.4 Эффективность работы i -ой ТС

Эффективность работы i -ой ТС QTS_i , $i \in [1,3]$, или отношение времен формирования задания и анализа результата оператором i -ой ТС к общему времени модельного цикла, вычисляется по формуле: $QTS_i = \frac{t_{cmd} + t_{res}}{t_{OP}}$, где

t_{OP} – параметр агента Агент_ТС, хранящий общее время цикла,

t_{cmd} – параметр агента Агент_ТС, хранящий время набора задания,

t_{res} – параметр агента Агент_ТС, хранящий время анализа результатов работы оператором.

На рисунке 22 приведена диаграмма эффективностей ТС.



Рисунок 22 – Диаграмма эффективностей ТС

Для входных параметров, соответствующих заданию, значениям эффективностей ТС (безразмерная величина) соответствуют:

- $QTS_1 = 0.915$;
- $QTS_2 = 0.917$;
- $QTS_3 = 0.914$.

8.5 Время ожидания результата выполнения i -ой ТС

Время ожидания результата выполнения задания i -ой ТС t_wait , $i \in [1,3]$, вычисляется в Анализе путём вычисления времени входа в Сервер и выхода из него. Таким образом мы получаем время нахождения транзакции на сервере, или время ожидания результата оператором.

Строится диаграммы времён ожидания результатов операторами ТС, представленная на рисунке 23.



Рисунок 23 – Гистограмма времени ожидания результата операторами ТС

Для входных параметров, соответствующих заданию, значениям времени ожидания результата операторами ТС соответствуют:

- $t_wait_1 = 48.26$ сек;
- $t_wait_2 = 47.44$ сек;
- $t_wait_3 = 46.85$ сек.

8.6 Время ожидания заданием i -ой ТС начала выполнения

Время ожидания заданием i -ой ТС начала выполнения сервером вычисляется на основании параметра t_{trans} агента Агент_ТС, хранящего время нахождения транзакции в серверном накопителе.

При входе в блок Накопитель параметру агента t_{trans} присваивается текущее время системы с помощью функции:

`agent.t_trans = time()`.

При выходе агента из блока Накопитель параметру агента t_{trans} присваивается значение:

`agent.t_trans = time() – agent.t_trans.`

Таким образом, параметр $time_{trans}$ на выходе содержит время нахождения транзакции в серверном накопителе. Затем в зависимости от номера ТС, с которого была выпущена транзакция, пополняются данные для диаграмм i -ой ТС, отображающей время ожидания терминальными станциями начала обработки транзакции сервером.

На основании этих данных строились диаграммы, представленные на рисунке 24.



Рисунок 22 – Диаграммы времени ожидания ТС начала обслуживания транзакции

Для входных параметров, соответствующих заданию, значениям времени ожидания ТС начала обслуживания транзакции соответствуют:

- $t_{trans_1} = 4.35$ сек;
- $t_{trans_2} = 2.78$ сек;
- $t_{trans_3} = 3.83$ сек.

9 Эксперименты с моделью и оптимизация параметров системы

В ходе работы с моделью был создан оптимизационный эксперимент, основной задачей которого является подбор оптимальных параметров производительности сервера, обеспечивающих его максимальную загрузку, а операторам любой ТС работу, при которой вероятность того, что среднее время ожидания оператором любой терминальной станции результата обработки задания не превысит 5 мин не должна быть ниже 0.9. В качестве целевой функции, которую будем максимизировать, возьмем эффективность загрузки сервера:

$$root. \text{эффективность_загрузки_сервера} \rightarrow \max$$

Также были прописаны требования, при которых среднее время ожидания оператором любой терминальной станции результата обработки задания не превысит 5 мин не должна быть ниже 0.9.

$$\text{вероятность_1} \geq 0.9,$$

$$\text{вероятность_2} \geq 0.9,$$

$$\text{вероятность_3} \geq 0.9.$$

В качестве параметров оптимизационного эксперимента были выбран параметр распределения времени обработки сервером одной команды.

В ходе эксперимента оптимизируется медиана равномерного распределения, заданная в границах от 14 до 18, как непрерывный параметр.

Рассмотрим более подробно оптимизационный эксперимент с одним параметрами.

Свойства оптимизационного эксперимента, в частности, вид целевой функции, используемые параметры, модельное время и заданные требования представлены на рисунке 25 и на рисунке 26.

На рисунке 27 приведены результаты работы оптимизационного эксперимента.

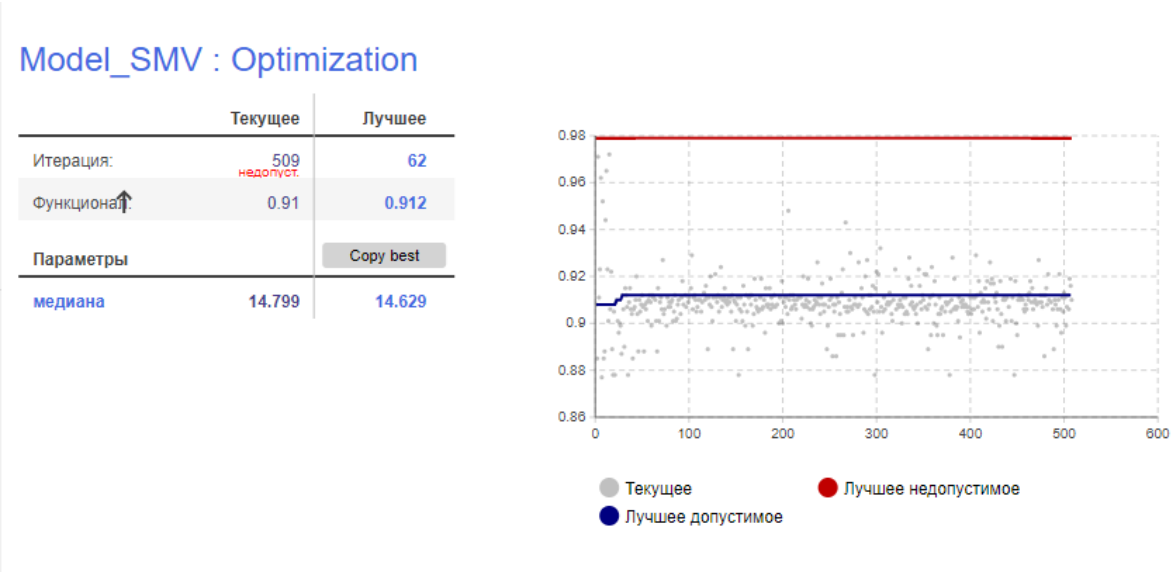


Рисунок 27 – Результат выполнения оптимизационного эксперимента

По результатам работы эксперимента оптимальное значение медианы составляет 14.629, тогда время обработки сервером одной команды должно распределяться по закону $uniform(12.435, 16.823)$.

На основании полученных оптимальных параметров проведём простой эксперимент. На рисунках 28–32 представлены результаты его работы.

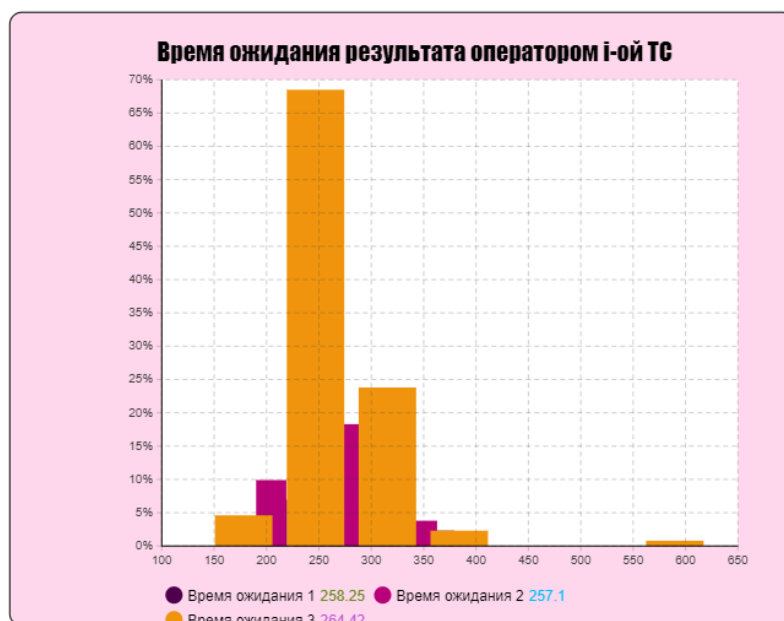


Рисунок 28 – Диаграмма времени ожидания результата операторами ТС при оптимальных параметрах

По диаграмме видно, что среднее время ожидания результата операторами ТС составило: $\frac{258.25+257.1+264.42}{3} \approx 259.92$ сек, что не превышает 5 минут. А это соответствует требованиям работы в режиме on-line.



Рисунок 29 – Диаграмма времени ожидания обслуживания транзакции при оптимальных параметрах

Время ожидания обслуживания транзакции в среднем равняется:

$$\frac{24.26+23.21+31.29}{3} = 26.25 \text{ секунд.}$$


Рисунок 30 – Диаграммы оценочных характеристик сервера при оптимальных параметрах

При оптимальных параметрах обеспечивается высокая загрузка сервера равная 91%, а абсолютная пропускная способность равна $0.068 \frac{\text{команд}}{\text{сек}}$.



Рисунок 31 – Диаграммы производительностей ТС при оптимальных параметрах

Производительность ТС при оптимальных параметрах равна $0.001 \frac{\text{заданий}}{\text{сек}}$.



Рисунок 32 – Диаграммы эффективностей ТС при оптимальных параметрах

Эффективность ТС при оптимальных параметрах в среднем равна $\frac{0.7+0.68+0.69}{3} = 0.69$ или 69%.

10 Итоговые результаты работы

В процессе выполнения курсовой работы был проведен анализ семантической модели многотерминальной вычислительной системы и была разработана математическая модель, которая представляет собой непрерывную событийно-стохастическую модель.

Учитывая, что потоки обслуживания в системе не являются простейшими, невозможно построить математическую модель в виде системы дифференциальных уравнений Колмогорова для определения вероятностей состояний. Поэтому на основе математической модели была создана имитационная модель многотерминальной вычислительной системы в системе моделирования AnyLogic.

С использованием определенных детерминированных параметров, описанных ниже, были рассчитаны выходные характеристики системы:

- количество команд в задании – 17;
- время формирования оператором одной команды – 28 с;
- время обработки сервером одной команды – 2.8 с;
- время анализа результата обработки оператором – 180 с.

В ходе эксперимента значения подтвердились, доказав адекватность модели. В таблице 3 представлены полученные значения.

Таблица 3 – Сравнение теоретических и практических значений

| Характеристики | Теоретические значения | Значения, полученные в ходе эксперимента |
|--------------------------------|------------------------|--|
| 1 | 2 | 3 |
| Эффективность загрузки сервера | 0.072 | 0.072 |
| Эффективности ТС | 0.928 | 0.928 |
| Пропускная способность сервера | 0.333 | 0.333 |

Продолжение таблицы 3

| 1 | 2 | 3 |
|--------------------------------------|--------|-----------------|
| Производительности ТС | 0.0014 | ≈ 0.002 |
| Время ожидания результата оператором | 51 | 51 |

В ходе работы были проведены оценки выходных характеристик заданной системы.

При заданных параметрах результаты вычисления выходных характеристик следующие:

- абсолютная пропускная способность сервера составила: $0.348 \frac{\text{команд}}{\text{сек}}$;
- эффективность загрузки сервера составила: 0.341;
- производительности ТС в среднем составили: $\frac{0.002 \cdot 3}{3} = 0.002 \frac{\text{заданий}}{\text{сек}}$;
- эффективности работы ТС в среднем составили: $\frac{0.915+0.917+0.914}{3} \approx 0.915$;
- среднее время ожидания операторами ТС результатов выполнения составило: $\frac{48.26+47.44+46.85}{3} \approx 47.517$ сек;
- среднее время ожидания каждой из ТС начала выполнения сервером составило: $\frac{4.35+2.78+3.83}{3} \approx 3.65$ сек.

Также были найдены оптимальные параметры обработки заданий в режиме on-line.

При оптимальных параметрах результаты вычисления выходных характеристик следующие:

- среднее время ожидания операторами ТС результатов выполнения составило 259.92 сек;
- среднее время ожидания транзакцией каждой из ТС начала выполнения сервером составило 26.25 секундам;
- эффективность загрузки сервера составила 91%;

- абсолютная пропускная способность сервера составила $0.068 \frac{\text{команд}}{\text{сек}}$;
- производительности ТС в среднем составили $0.001 \frac{\text{заданий}}{\text{сек}}$;
- эффективности работы ТС в среднем составили 69%.

11 Список использованных источников

- 1 Стандарт организации. Комплексная система управления качеством деятельности вуза. СТО СГАУ 02068410-004-2007 [Электронный ресурс] // Самарский университет: [сайт]. Общие требования к учебным текстовым документам. – Самара: Изд-во Самарского университета, 2011. – 29 с.
- 2 Порядок выполнения и защиты курсовых работ [Электронный ресурс] // Самарский университет: [сайт]. Метод. указания / сост.: Н.А. Дубровина, А.Г. Лукин, Ю.И. Ряжева. – Самара: Изд-во Самарского университета, 2018. – 34 с.
- 3 Справочная система AnyLogic [Электронный ресурс]. URL: <https://help.anylogic.ru/index.jsp> (дата обращения: 15.05.2021).