Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЁТ

По лабораторной работе №5

«Исследование сетевого подхода к моделированию»

Выполнил:

Ст. гр. ИС/б-20-1-о

Хроменко Д.А.

Проверил:

Хохлов В.В.

Севастополь

2022

**5.1 Цель работы**

Исследование способов построения простейших моделей непрерывных систем с помощью методов аналитического и имитационного моделирования. Изучения технологии системно-динамического имитационного моделирования в среде AnyLogic.

**5.2 Ход работы**

Был нарисован сетевой граф вычислительной системы, заданной по варианту.

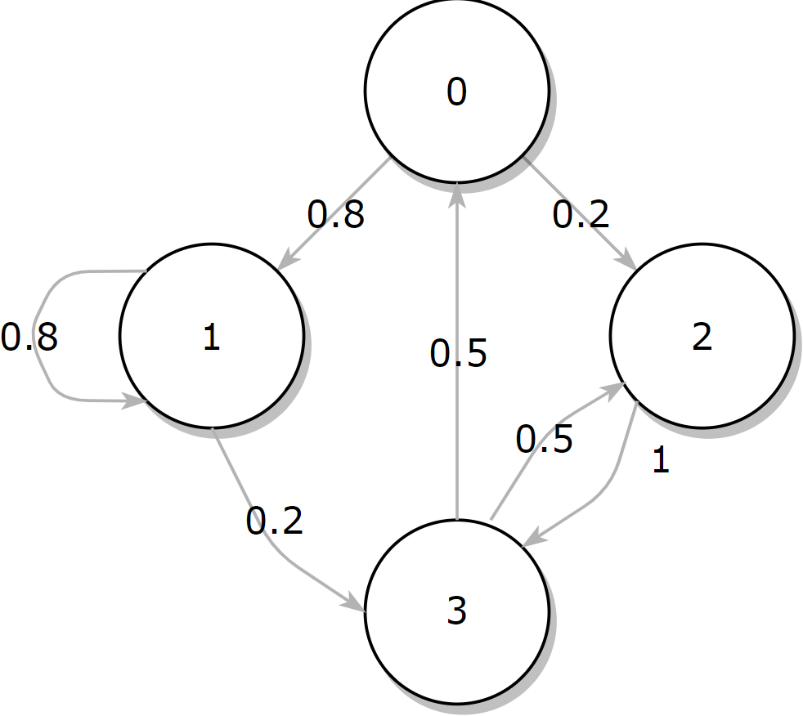


Рисунок 1 – Граф сетевой модели

Аналитически было проверено, существует ли перегрузка сети для заданной модели.



Рисунок 2 – Аналитическая оценка СеМО

Была построена имитационная модель сети. В модели был также организован сбор статистики для подтверждения аналитических расчетов. Был организован эксперимент с подбором параметров (количества приборов в СМО, интенсивности входящего потока заявок).

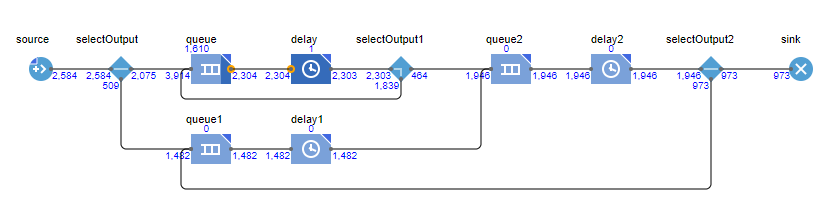


Рисунок 3 – Имитационная оценка СМО

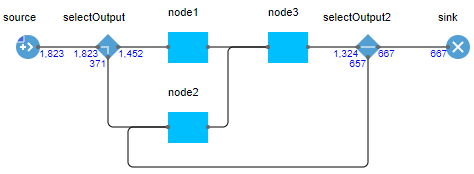


Рисунок 4 – Имитационная модель СеМО

**Выводы**

В ходе выполнения данной лабораторной работы были исследованы характеристики одноканальной системы массового обслуживания, использовав аналитический и имитационный методы моделирования; изучены особенности работы, отладки с помощью пакета моделирования Anylogic.

**Ответы на контрольные вопросы**

1. Сеть массового обслуживания (СеМО) – совокупность взаимосвязанных СМО, в среде которых циркулируют заявки. Основными элементами СеМО являются узлы и источники заявок.

Узел сети представляет собой систему массового обслуживания.

Источник – генератор заявок, поступающих в сеть и требующих определенных этапов обслуживания в узлах сети.

Граф СеМО – ориентированный граф, вершины которого соответствуют узлам СеМО, а дуги отображают переходы заявок между узлами.

Переходы заявок между узлами СеМО, в общем случае, могут быть заданы в виде вероятностей передач.

Путь движения заявок в СеМО называется маршрутом.

2. Для описания линейных разомкнутых и замкнутых однородных экспоненциальных СеМО необходимо задать следующие параметры:

• число узлов в сети *n*;

• число обслуживающих приборов в узлах сети *K1, ..., Kn*;

• матрицу вероятностей передач *P = [pij | i, j = 0, 1, ..., n]*;

• интенсивность λ0 источника заявок, поступающих в РСеМО, или число заявок M, циркулирующих в ЗСеМО;

• средние длительности обслуживания заявок в узлах сети *b1, ..., bn*.

Условие отсутствия перегрузок в разомкнутой СеМО предполагает отсутствие перегрузок в каждом из узлов сети. В замкнутой СеМО перегрузки не возникают.

Характеристики СеМО делятся на узловые и сетевые.

Состав узловых характеристик СеМО, работающей в стационарном режиме, такой же, как и для СМО. На основе узловых характеристик рассчитываются средние значения сетевых характеристик СеМО:

• суммарная нагрузка *Y* и загрузка *R*;

• среднее суммарное число заявок *L* во всех очередях сети;

• среднее суммарное число заявок *M* в разомкнутой сети (во всех узлах);

• среднее время ожидания *W* и пребывания заявок *U* в сети;

• *αj =λj / λ0* - коэффициент передачи для узла *j*, показывающий среднее число попаданий заявки в узел j за время ее нахождения в сети;

• производительность *λ0* замкнутой СеМО.

Для неоднородной СеМО перечисленные характеристики определяются как для каждого класса в отдельности, так и для объединенного (суммарного) потока заявок.

3. С помощью экспериментов задаются конфигурационные настройки модели. AnyLogic поддерживает несколько типов экспериментов, каждый из которых соответствует своей задаче моделирования.

AnyLogic поддерживает следующие типы экспериментов: Простой эксперимент, Варьирование параметров, Оптимизация, Сравнение "прогонов", Монте-Карло, Анализ чувствительности, Калибровка, Обучение ИИ, Нестандартный.

4. Разработанная в AnyLogic модель полностью отображается в Java код, и будучи связанной с исполняющим модулем AnyLogic (также написанном на Java) и опционально с Java оптимизатором, становится абсолютно независимым самостоятельным Java приложением. Это делает модели AnyLogic кроссплатформенными: они могут быть запущены в любой поддерживающей Java среде.

5. Агентное моделирование включает в себя клеточные автоматы, элементы теории игр, сложных систем, мультиагентных систем и эволюционного программирования, методы Монте-Карло, использует случайные числа. В агентном моделировании сначала устанавливаются параметры активных объектов – агентов, и определяется их поведение. Затем устанавливаются связи между агентами, задается окружающая среда и запускается моделирование. Индивидуальные действия каждого из агентов образуют глобальное поведение моделируемой системы.

6. Чтобы добавить синхронизацию в агентную модель:

1) Откройте панель Свойства для агента, задающего среду для других агентов (например, Main).

2) Откройте секцию свойств Пространство и сеть.

3) Установите флажок Выполнять шаги. По умолчанию шаг длится 1 единицу модельного времени, но его длительность может быть изменена в поле Длительность шага (в единицах мод. времени).

4) В полях Действие перед выполнением шага и Действие после выполнения шага напишите код, который будет выполняться соответственно до и после того, как все агенты выполнят свои шаги, если таковые будут.

5) В секции Действия агента задайте действие Перед выполнением шага и Действие на шаге (однако эти поля можно оставить пустыми).

7. В разработанной агентной модели был использован класс Agent.

8. В разработанной агентной модели реализованы функции нахождения загрузки, нагрузки и обнаружения перегрузки сети массового обслуживания. Эти функции осуществлены при помощи элементов, имитирующих события: Queue, Delay, SelectOutput.

9. Поддерживаются следующие типы функций сбора статистики: Кол-во, Сумма, Среднее, Мин. и Макс.

10. Объект сбора статистики Статистика вычисляет основную статистическую информацию (среднее значение, минимум, максимум и т.д.) для последовательности измеренных значений. Объект работает по-разному в зависимости от того, считаются ли собираемые данные непрерывными или дискретными.

При работе с непрерывными данными, измерения имеют длительность во времени, то есть значение остается актуальным в течение какого-то времени, и изменяется только в дискретные моменты времени (например, длина очереди). В объект непрерывной статистики можно добавлять только измерения с возрастающими значениями временных меток. Методы вычисления среднего значения и т.д. подразумевают, что последнее добавленное значение было актуально до времени вызова метода. Собранная объектом статистика в этом случае доступна посредством программного интерфейса класса StatisticsContinuous.

При работе с дискретными данными, измерения не имеют длительности во времени (например, цена продуктов или длительность пребывания пациентов). Они считаются изолированными, дискретными точками во времени, так что среднее значение является просто суммой измерений, поделенной на их общее количество. Собранная объектом статистика в этом случае доступна посредством программного интерфейса класса StatisticsDiscrete.