

*Ильичев Владимир Юрьевич, к.т.н., доцент кафедр «Тепловые двигатели и гидромашины» Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Калуга, Россия*

*Крысин К.Р., студент кафедры «Тепловые двигатели и гидромашины» Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Калуга, Россия*

## **ИСПОЛЬЗОВАНИЕ СИМУЛЯТОРА СЕТЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ С ПРИМЕНЕНИЕМ МОДУЛЯ QUEUEING-TOOL**

**Аннотация:** Представляемая работа посвящена описанию быстро развивающейся области исследований – моделированию систем массового обслуживания. Разработана методика использования модуля Queueing-tool для среды программирования Python, а также написания на её основе алгоритма и универсального программного кода, позволяющего производить построение систем (сетей) массового обслуживания, их расчёт для любого промежутка времени и формирование результатов, наглядно (визуально-графическим путём) отображающих их работу и характеристики. В статье приведены два примера использования данной методики, основанной на применении направленных сетевых графов, для разноплановых моделей, сильно отличающихся степенью сложности графического отображения результатов. Описаны дополнительные возможности и перспективы применения модуля Queueing-tool для разных типов перспективных исследований.

**Ключевые слова:** система массового обслуживания, сетевой граф, программирование, язык Python, библиотека NetworkX, модуль Queueing-tool.

**Annotation:** This paper describes a rapidly evolving field of research - modeling mass service systems. A methodology has been developed for using the Queueing-tool module for the Python programming environment, as well as writing an algorithm and universal program code based on it, which allows building mass service systems (networks), calculating them for any period of time and generating results that clearly (visually and graphically) display their work and characteristics. The article provides two examples of using this technique, based on the use of directional network graphs, for diverse models that greatly differ in the degree of complexity of graphical display of results. Additional possibilities and prospects for using the Queueing-tool module for different types of promising studies are described.

**Keywords:** mass service system, network graph, programming, Python language, NetworkX library, Queueing-tool module.

## **Введение**

Системы массового обслуживания [1] в настоящее время являются основой для моделирования процессов, происходящих в реальных сложных системах [2]. Можно привести следующие примеры систем массового обслуживания:

- распределение товаров в торговых супермаркетах;
- соединение компьютерных узлов или телефонных линий друг с другом;
- соединение химических элементов или живых клеток в более сложные структуры;
- демографические конгломерации;
- государственные структуры, отслеживающие выполнение тех или иных функций;
- военизированные формирования.

Перечисленные системы осуществляют обычно последовательную или параллельную обработку однотипных процессов. Так как современные системы

и осуществляемые с их помощью процессы являются достаточно сложными, для их исследования приходится использовать средства вычислительной техники и специальные программные продукты, одним из которых является описываемый в данной статье модуль Queueing-tool для языка Python [3]. С помощью этого модуля можно произвести исследование работы даже очень сложных систем массового обслуживания.

### **Материал и методы исследования**

Полноценную документацию по модулю Queueing-tool можно найти на сайте его разработчиков [4]. Она включает в себя все необходимые для исследования систем массового обслуживания средства, которые к тому же с течением времени развиваются с точек зрения повышения функциональности и быстродействия:

- описание применяемых методов организации баз данных;
- объяснение методов симуляции процессов в системах;
- средства визуального отображения, построенных с помощью теории графов сетей массового обслуживания;
- особые для данного модуля средства;
- встроенные средства генерации графов.

В документации приведена также характеристика понятий, используемых в теории массового обслуживания: агентов, очередей и осуществляемых ими функций. Рассмотрим кратко эти понятия [5].

Агентами в теории массового обслуживания называют объекты любого вида, перемещение которых по сети отслеживается. Агент с течением времени переходит из одного состояния (узла сети) в другое – это называется перемещением в очереди. Вероятность перемещения из каждого текущего состояния в каждое вероятное последующее называется функцией очереди (в современном лексиконе исследователей сетей принято понятие вероятности маршрутизации).

Целью данной статьи является подробная разработка методики, алгоритма и кода программы для работы с симулятором сетей, встроенным в

модуль `Queueing-tool`, предназначенный для применения совместно с современным бесплатным, универсальным и очень популярным языком программирования Python [6]. Интересно, что данный симулятор позволяет произвести графическое отображение любых типов сетей, используемых в теории массового обслуживания.

Рассмотрим порядок применения функций `Queueing-tool` (и не только этого модуля) в разработанном универсальном алгоритме, предназначенном для создания сетей массового обслуживания, задания их параметров, их расчёта для любого промежутка времени и представления результатов в наглядном графическом виде:

1. Импорт в программу Python самого модуля `Queueing_tool` наряду с модулем построения сетевых графов `NetworkX` [7].

2. Использование функции генерации графа сети: это может быть граф любого необходимого исследователю типа – со случайным выбором типа рёбер, случайный с выбранным типом узлов, не случайный по предварительно созданной матрице узлов и рёбер и т.д. При этом граф может быть направленным или ненаправленным [8] и разбит на несколько подграфов (сот).

3. Преобразование созданного графа в сеть очереди (с помощью функции `QueueNetwork` [9]).

4. Задание максимального количества агентов в очереди созданной сети.

5. Инициализация сети массового обслуживания с возможностью определения количества серверов очереди (каждый из которых будет управлять своей очередью).

6. Симуляция сети с заданным количеством серверов очереди.

7. С помощью специальных команд модуля `NetworkX` визуализация поведения созданной сети и в случае необходимости сохранение результатов в графический файл. При этом узлы и рёбра сети чаще всего создаются с помощью специального графического подмодуля `Graphviz` [10].

Создадим вначале для наглядности вышеописанных основных понятий в качестве примера несложную сеть массового обслуживания случайной

архитектуры, содержащую по одному началу и окончанию и состоящую из 10 узлов, по которой перемещаются 50 агентов, количество которых неизменно. Загрузка этих узлов, рассчитанная с помощью составленной программы на момент времени 100, приведена на рис. 1.

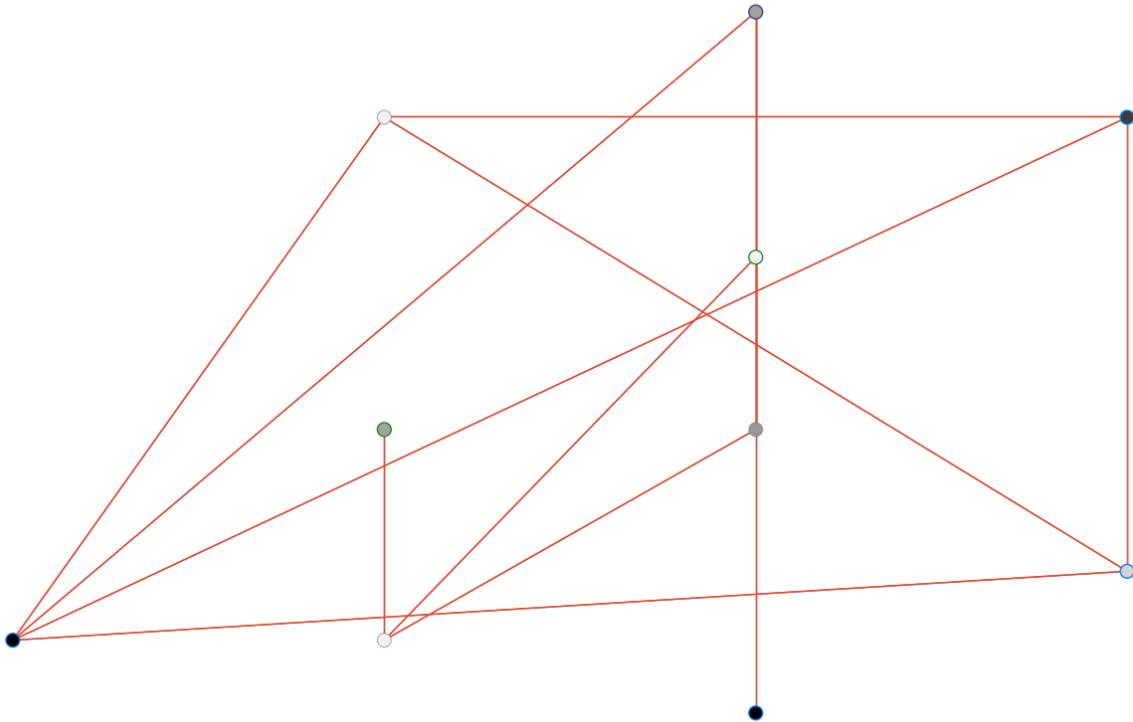


Рис. 1. Вычисленная с помощью программы загрузка сети массового обслуживания.

Чем темнее цвет узла на рисунке 1, тем более загруженным этот узел является. Наглядны также и созданные программой случайные связи между узлами.

### **Пример расчёта**

Далее рассмотрим другой, гораздо более сложный, случай применения разработанной методики исследования систем (сетей) массового обслуживания, в котором используется сеть, координаты узлов которой заранее определены в виде матрицы (это географические координаты части города), а агенты представляют из себя объекты – транспортные средства, перемещающиеся по местности (их координаты и тип в процессе работы программы постоянно пересчитываются и записываются в базу данных в формате xml [11]).

В данном случае в начале работы программы (п. 2 представленного выше алгоритма) координаты узлов считываются из файла, и в процессе симуляции сети в п. 6 образуется только три типа узлов, обозначаемых разным (указанным) цветом: красные узлы представляют из себя остановки агентов (транспортных средств) на светофорах, зеленые – стоянки в гаражах, синие – нахождение на дорогах. При этом в симуляции в качестве исходного параметра указывается также количество событий в системе, произошедших к моменту времени, на который приходится приводимая картина сети (и положение на ней транспорта). Оно задано равным 50000. При инициализации указываются 200 активно функционирующих объектов.

Полученная в результате симуляция показана на рис. 2.

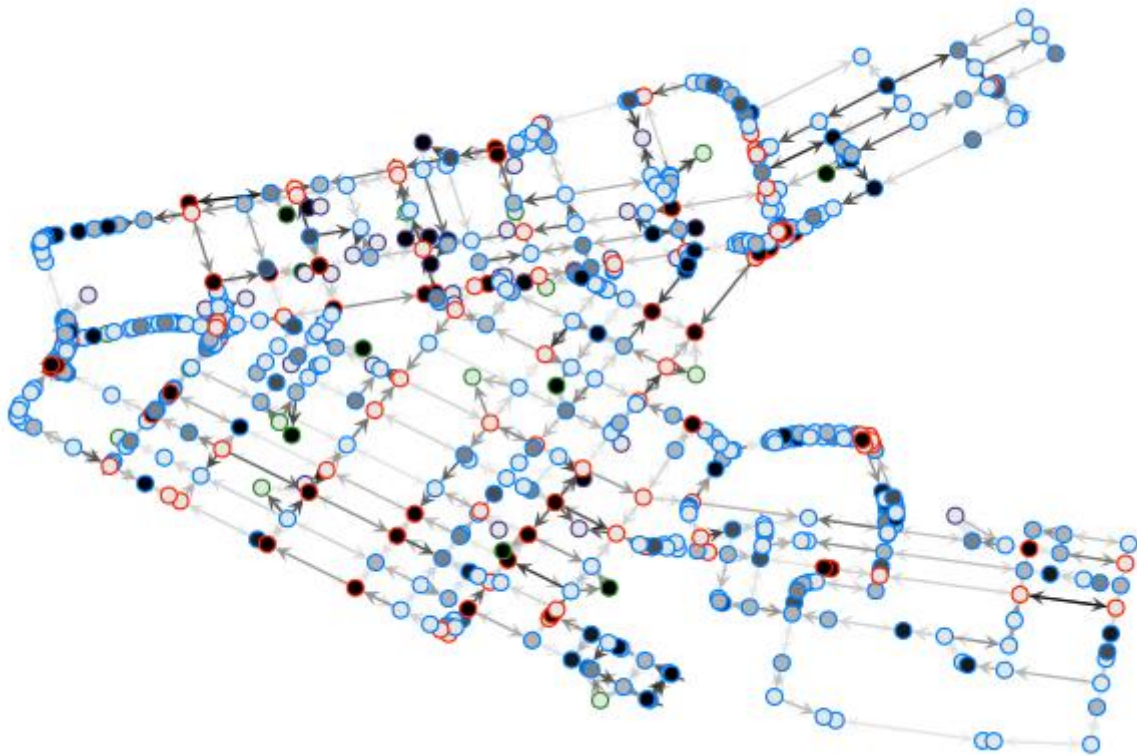


Рис. 2. Симуляция положения объектов в сети массового обслуживания.

Как видим, симуляция получилась намного сложнее, чем представленная на рис. 1 и визуализация результатов поэтому представлена несколько по-другому. Положения объектов на рис. 2 показаны чёрными кружками. Белые кружки показывают, что данное состояние узла системы свободно.

Граф в данном случае является направленным, так как важно направление движения транспортного средства. Также здесь мы можем видеть и отличие в интенсивности окраски стрелок, означающую сколько объектов находится «на данном отрезке пути».

В модуле `Queueing_tool` существует и возможность анимации [12] состояния сети массового обслуживания с помощью команды `animate`, но данная анимация происходит не в реальном времени, поэтому особого интереса при указанном исследовании не представляет.

### **Заключение**

Таким образом, цель данного исследования полностью достигнута: создана методика применения модуля создания и анализа систем массового обслуживания `Queueing-tool`, а также разобраны алгоритм её применения и программные коды.

На наглядных практических примерах показаны стадии использования созданной методики и описаны применяемые при этом команды, включая создание сетевой графической модели, правила представления параметров агентов, инициализации и симуляции сети. Приведены особенности определяемых характеристик систем массового обслуживания и способы их визуального отображения.

Также разобраны два разноплановых и отличных по сложности примера создания систем массового обслуживания, позволяющее читателю, заинтересованному данной технологией, достаточно легко создать графическую сеть и её параметры, наиболее подходящие для решения его собственной задачи получения характеристик процессов, происходящих при работе сложной системы любого рода.

Описанное исследование обладает особой важностью, так как при постоянном увеличении сложности строения и принципов работы систем наблюдается недостаток литературного материала, позволяющего фактически решать возникающие проблемы. Авторы пытались донести читателю мысль, что системы массового обслуживания в данном плане позволяют

смоделировать топологию практически любой сложной системы и получить её наглядные характеристики, отображаемые визуальными средствами, и даже произвести наблюдение за её работой путём создания анимации.

Благодаря всем вышеизложенным достоинствам, авторы настоятельно рекомендуют освоение модуля Queueing-tool для исследования систем массового обслуживания в среде программирования Python. Это может позволить вывести уровень проводимых исследований на качественно новый современный уровень.

### **Библиографический список:**

1. Viana Carvalho Cravid H., Zaryadov I.S., Milovanova T.A. Queueing systems with different types of renovation mechanism and thresholds as the mathematical models of active queue management mechanism. // Discrete and Continuous Models and Applied Computational Science. 2020. Т. 28. № 4. С.305-318.

2. Ilyichev V.Yu. Development of a program for Lorentz attractor research and its use. // The Complex Systems. - 2021. - № 1 (11). - С.58-64.

3. Скрипко А.А. Моделирование обработки трафика в телекоммуникационных системах. // В сборнике: 71-я Международная студенческая научно-техническая конференция. Материалы конференции. Астрахань, 2021. С. 408-409.

4. Queueing-tool. Module documentation. [Электронный ресурс]. URL:<https://queueing-tool.readthedocs.io/en/latest/modules.html> (Дата обращения 12.07.2022).

5. Игнатенко В.В Теория вероятностей и теория массового обслуживания в курсе высшей математики. // В сборнике: Инновационные технологии обучения физико-математическим и профессионально-техническим дисциплинам. Материалы XII Международной научно-практической конференции: в 2-х частях. Министерство образования Республики Беларусь,



Учреждение образования "Мозырский государственный педагогический университет имени И. П. Шамякина". 2020. С. 43-45.

6. Ильичев В.Ю. Создание параметрических конечно-элементных трехмерных объектов с использованием функций Python. // Системный администратор. 2021. № 5 (222). С. 82-85.

7. Ильичев В.Ю. Визуальное отображение сетевых графических структур с помощью библиотеки NetworkX. // E-Scio. 2022. № 4 (67). С. 705-713.

8. Сафронова М.Е. Разработка методики использования библиотеки GraphViz для создания направленных графов. // E-Scio. 2022. № 3 (66). С. 630-638.

9. Ильичев В.Ю., Крысин К.Р. Использование модуля DeepGraph для изображения графов отношений между объектами. // Заметки ученого. 2022. № 4. С. 26-30.

10. Ильичев В.Ю., Юрик Е.А. Обработка статистических данных методом глубокого обучения с использованием модуля Keras. // Научное обозрение. Технические науки. 2020. № 5. С. 16-20.

11. Михайлова Т.А. Исследование возможного усовершенствования xml-схемы, используемой для формирования XML-документа - межевого плана в форме электронного документа. // Кадастр недвижимости. 2019. № 2 (55). С. 109-114.

12. Гудыменко Е.Р. Понятие анимации. Функции и виды анимации. // Международный научный студенческий журнал. 2018. № 7. С. 59-63.