МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ
VIII Международной молодежной научной
конференции
«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ,
ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 26–30 мая 2021 г.

Под общей редакцией И.С. Шмырина

Томск
Издательство Томского государственного университета
2021

Заключение

В результате проделанной работы было спроектировано и реализовано приложение для имитационного моделирования ресурсной системы массового обслуживания $M/M/\infty$, функционирование которой зависит от состояния случайной среды. Приложение успешно используется для научных исследований специалистами по теории массового обслуживания Томского государственного университета.

ЛИТЕРАТУРА

1. Назаров А.А., Баймеева Г.В. Исследование системы $M/M/\infty$ в полумарковской случайной среде // Известия вузов. Физика. – 2015. – Т. 58. – № 11/2. – С. 198–203.
2. Полин Е.П., Моисеева С.П., Рожкова С.В. Асимптотический анализ неоднородной системы массового обслуживания $M/M/\infty$ в марковской случайной среде // Вестн. Том. гос. ун-та. УВТиИ. – 2019. – № 47. – С. 75–83.
3. Чернышова Е.Н., Галилейская А.А., Лисовская Е.Ю. Суммарный объем занятого ресурса в системе обработки информации с реализацией хранения резервной копии сообщений // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: материалы Всероссийской конференции с международным участием, Москва, 13-17 апреля 2020 г. – М.: Изд-во РУДН, 2020. – С. 59–61.
4. Лисовская Е.Ю., Моисеев А.Н., Моисеева С.П., Пагано М. Моделирование процессов обработки данных физических экспериментов в виде немарковской многоресурсной системы массового обслуживания // Известия вузов. Физика. – 2018. – Т. 61. – № 12. – С. 39–46.

DOI: 10.17223/978-5-907442-42-9-2021-24

ИМИТАЦИОННАЯ МОДЕЛЬ ТАНДЕМНОЙ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ С ПОВТОРНЫМИ ВЫЗОВАМИ

Морозова М.А., Благинин А.Л.

Томский государственный университет

morozova_mariya_a@mail.ru, alex-b.l@yandex.ru

Введение

В рамках дипломной работы мы проводим исследование числа заявок на орбите тандемной системы с повторными вызовами и двумя последовательно обслуживающими приборами методом асимптотического анализа. Для того, чтобы определить область применимости полученного решения, было решено построить имитационную модель исследуемой системы. Метод имитационного моделирования [1] представляет из себя компьютерный эксперимент [2], который осуществляется на базе программного продукта, воссоздающего структуру и принципы работы исследуемой системы. Данный подход позволяет получить статистический материал, необходимый для построения эмпирического распределения вероятностей, которое можно сравнить с результатами, полученными аналитически.

Целью преддипломной практики является реализация программы имитационной модели тандемной RQ-системы с двумя последовательно обслуживающими приборами для нахождения эмпирического распределения вероятностей числа заявок на орбите.

Для достижения цели были поставлены следующие задачи:

- определить методику проведения имитационного моделирования; спроектировать архитектуру имитационной модели исследуемой системы;
- определить набор инструментов для реализации;
- реализовать спроектированную имитационную модель; оценить точность результатов имитационного моделирования.

В первом разделе содержится описание математической модели и постановка задачи, во втором сказано о методике проведения имитационного моделирования, в третьем – основные структурные элементы программы имитационной модели, в четвертом говорится о реализации имитационной модели, в пятом – о точности результатов имитационного моделирования.

1. Описание математической модели и постановка задачи

Прежде чем перейти к проектированию, опишем исходную модель (рис. 1). Мы исследуем систему с повторными вызовами и двумя последовательно обслуживающими приборами, на вход которой поступает простейший поток заявок с интенсивностью λ . Если заявка входящего потока обнаруживает первый прибор свободным, то она встает на прибор и обслуживается в течении экспоненциально распределенного времени с параметром μ_1 , после чего обращается ко второму прибору. Если второй прибор свободен, заявка встает на него и обслуживается в течении экспоненциально распределенного времени с параметром μ_2 . Если заявка входящего потока застает первый прибор занятым, она мгновенно отправляется на орбиту, где после случайной экспоненциально-распределенной задержки с параметром σ снова пытается встать на обслуживание на первый прибор. Если заявка после обслуживания на первом приборе застает второй прибор занятым, она мгновенно отправляется на орбиту, где после случайной экспоненциально-распределенной задержки с параметром σ снова пытается встать на обслуживание на первый прибор.

Рис. 1. Тандемная RQ-система

Ставится задача нахождения распределения вероятностей числа заявок на орбите.

2. Методика проведения имитационного моделирования

В основу метода имитационного моделирования было решено положить дискретно-событийный подход [3], который активно применяется для моделей массового обслуживания. В процессе функционирования в системе происходят события, которые переводят ее в разные состояния, и дискретно-событийный подход позволяет обойти моделирование интервалов времени, в которые состояние системы остается неизменным. Это способствует ускорению получения результатов, что немаловажно для проведения исследований.

К событиям в исследуемой системе относятся:

- 1) поступление заявки из входящего потока;
- 2) завершение обслуживания заявки на первом приборе,
- 3) завершение обслуживания заявки на втором приборе;
- 4) обращение заявки с орбиты к первому прибору.

Таким образом, в процессе моделирования модельное время постоянно корректируется, сдвигаясь после каждого события к моменту следующего, т.е., если обозначить T_1 – момент времени поступления заявки из входящего потока, T_2 – момент времени завершения на первом приборе, T_3 – момент времени завершения на втором приборе, T_4 – момент времени обращения заявки с орбиты к первому прибору, T – модельное время, то

$$T = \min(T_1, T_2, T_3, T_4). \quad (1)$$

Каждое из перечисленных событий предполагает выполнение определенных действий:

1. Поступление заявки из входящего потока: система пытается отправить заявку на обслуживание первым прибором. Если он свободен, то забирает данную заявку и определяет время события T_2 ; если прибор занят, то он возвращает предложенную заявку, которую система направляет на орбиту, определяющую событие T_4 для этой заявки.

2. Завершение обслуживания заявки на первом приборе: система пытается отправить заявку на обслуживание вторым прибором. Если он свободен, то забирает данную заявку и определяет время события T_3 ; если прибор занят, то он возвращает предложенную заявку, которую система направляет на орбиту, определяющую событие T_4 для этой заявки.
 3. Завершение обслуживания заявки на втором приборе: заявка покидает систему.
 4. Обращение заявки с орбиты к первому прибору: аналогично п.1.
- Завершение процесса моделирования происходит тогда, когда в системе произойдет заданное число событий.

3. Основные структурные элементы программы имитационной модели

Проектирование имитационной модели осуществлено на основе объектно-ориентированного подхода [4] с помощью языка моделирования UML [5]. В данной главе описаны основные структурные элементы программы, изображенные на диаграмме классов (рис. 2).

Рис. 2. Объектно-ориентированная модель базовых элементов автономного модуля для исследуемой тандемной RQ-системы

Начнем рассмотрение архитектуры с класса заявки (Claim). Он имеет единственный атрибут (MomentOfTime), который содержит момент времени, в который с этой заявкой должно произойти одно из событий T_1 , T_2 , T_3 , T_4 .

Далее о реализации входящего потока. Поскольку в дальнейших исследованиях мы планируем рассматривать и другие (помимо простейшего) входящие потоки, было решено реализовать интерфейс IStream с методом Send(), который занимается созданием следующей заявки, которая поступит в систему, что характерно для любого потока.

У простейшего потока (SimpleStream) параметром конструктора является его интенсивность (intensity), на базе которой создается заявка, которая должна следующей поступить в систему (nextSend). Аналогичные генерации следующей заявки происходят в методе Send() после каждого фактического поступления заявки из потока.

У класса прибора (Node) есть два атрибута: интенсивность обслуживания (intensity) и обслуживаемая заявка (nowServing). Метод Availability(...) получает в качестве параметра заявку, пытающуюся попасть на прибор. Если прибор занят, заявка возвращается, а если свободен, то заявка встает на прибор и обслуживается (метод Serve()). По завершению обслуживания метод Serve() возвращает обслуженную заявку, а атрибут nowServing обнуляется.

В классе орбиты (Orbit) есть два атрибута: интенсивность повторных вызовов (intensity), которая является параметром конструктора орбиты, и коллекция заявок, находящихся на орбите. В методе Push(...) реализовано поступление заявки на орбиту, а в методе Send(), соответственно, уход с нее.

Связующим для вышеперечисленных классов является класс системы (Sys), который содержит в себе входной поток событий (input), два последовательно работающих прибора (Node1, Node2) и орбиту (Orbit), а также логику работы, исследуемой СМО (метод Process()).

Статический класс Utility предназначен для моделирования экспоненциального распределения. За отслеживание изменений в системе и сбор статистики отвечает статический класс Model. Свойство time – это условное «время», которое прошло с начала запуска моделирования или, другими словами, текущее время моделирования. Условным мы называем его по причине того, что оно не соответствует реальному, поскольку мы используем преимущества дискретно-событийного подхода, о котором говорили ранее при выводе (1).

С помощью метода PopClosestClaim() модель определяет, какое событие должно произойти следующим (заявка с минимальным значением поля MomentOfTime), а в методе NextMoment() производится переход к нему (согласно (1)) и происходит запуск основного процесса системы.

Далее, у класса Model есть свойства eventsInvoked и eventLimit. Первое является счетчиком для уже прошедших событий, второе содержит значение числа событий, при достижении которого моделирование завершается. Массив stats используется для накопления статистических данных о числе заявок на орбите в процессе моделирования.

Метод GetStatistics() вызывается после завершения моделирования и формирует массив, представляющий из себя искомое распределение вероятностей числа заявок на орбите.

В результате была спроектирована архитектура автономного модуля, она объединяет в себе все классы, о которых сказано в данном и предыдущем разделах.

4. Реализация имитационной модели

Имитационная модель реализована как десктопное приложение с пользовательским интерфейсом Windows Forms.

Для разработки был выбран язык C# в силу его объектной ориентированности, высокой популярности, автоматической «сборки мусора». Платформа реализации – Microsoft .NET Framework (версия 4.6.1) с огромной базой библиотек и шаблонов.

Среда разработки (интегрированная) – Visual Studio Community 2017.

На рис. 3 показано главное окно программы, где пользователь может задать параметры системы: интенсивность экспоненциального обслуживания приборов (Mu1 и Mu2), интенсивность экспоненциального обращения заявок с орбиты (Sigma), интенсивность входящего потока. Также пользователю предлагается задать число событий, при достижении которого процесс моделирования завершится (Event Limit).

Рис. 3. Интерфейс приложения

При нажатии кнопки Start запускается процесс моделирования и показывается соответствующее окно (рис. 4). Процесс можно прервать, нажав кнопку Cancel.

Рис. 4. Окно, свидетельствующее о процессе моделирования

Результатом работы программы является график распределения вероятностей числа заявок на орбите, а также в левом нижнем углу дополнительно посчитано математическое ожидание (Expected value) и дисперсия (Dispersion) для полученного распределения (рис. 5).

Рис. 5. Пример работы программы

Также в программе предусмотрено сохранение полученного распределения в файл формата .csv. Такие файлы удобно использовать в среде Mathcad для работы с полученными результатами. Чтобы сохранить распределение, нужно в верхней части главного окна нажать на кнопку File, где будет предложено сохранить файл.

5. Точность результатов имитационного моделирования

Прежде чем использовать имитационную модель, реализованную в рамках преддипломной практики, и говорить об области применимости полученных аналитически асимптотических результатов, следует сказать о надежности результатов имитационного моделирования, на базе которых мы оценим решение, полученное методом асимптотического анализа.

Как известно из закона больших чисел, чем больше повторений эксперимента, тем меньше отклонение эмпирического распределения вероятностей от «эталонного» результата, т.е. напрашивается вывод о том, что результатам имитационного моделирования можно доверять при объеме выборки, стремящемся к бесконечности. Однако на практике мы не можем работать с бесконечными выборками, поэтому встал вопрос о том, каким лучше задавать число моделируемых событий (Event Limit).

Чтобы количественно определять погрешность получаемых распределений, было решено применять расстояние Колмогорова, описываемое формулой

$$d = \max_{0 \leq i < \infty} \left| \sum_{v=0}^i (P1_i - P2_i) \right|,$$

где $P1_i$ и $P2_i$ – эмпирические вероятности числа заявок на орбите, полученные в результате работы имитационной модели.

Экспериментально было выявлено, что выборки объемом около $10^6 \times M$ (где M – среднее число заявок на орбите) гарантируют стабильность получаемых распределений: расстояние Колмогорова для них не превышает значения 0.005.

Стабильность работы имитационной модели свидетельствует о ее точности и, как следствие, возможности применимости ее для оценки асимптотических результатов.

Заключение

В результате работы была спроектирована и реализована имитационная модель тандемной системы с повторными вызовами и двумя последовательно обслуживающими приборами для нахождения распределения вероятностей числа заявок на орбите. Программа представляет себя десктопное приложение с пользовательским интерфейсом Windows Forms, написанное на языке программирования C#. В основу моделирования был положен дискретно-событийный подход, который позволяет значительно ускорить время сбора статистических данных, необходимых для получения искомого распределения.

Точность результатов моделирования определена по критерию расстояния Колмогорова, которое не превышает значения 0.005 при объеме выборки, превышающем математическое ожидание в 10^6 раз.

ЛИТЕРАТУРА

1. Лоу А., Кельтон В. Имитационное моделирование. 3-е изд. – СПб.: Питер, 2004. – 848 с.
2. Бахвалов Л.А. Моделирование систем: учеб. пособие для вузов. – М.: Изд-во Московского гос. горного ун-та, 2006. – 295 с.
3. Robinson S. Simulation: the practice of model development and use. – Hoboken: Wiley, 2004. – 336 p.
4. Буч Г., Максимчук Р.А., Энгл М.У. и др. Объектно-ориентированный анализ и проектирование с примерами приложений. пер. Д. Ключин. 3-е изд. – М.: Вильямс, 2010. – 720 с.
5. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. – СПб.: Питер, 2007. – 544 с.