МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ

IV Международной молодежной научной конференции

«МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ, ТЕХНИЧЕСКИХ И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 20–21 мая 2016 г.

Под общей редакцией
кандидата технических наук И.С. Шмырина

Томск
Издательский Дом Томского государственного университета
2016

СЕКЦИЯ III.
ПРИКЛАДНОЙ ВЕРОЯТНОСТНЫЙ АНАЛИЗ

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ДВУХФАЗНОЙ RQ-СИСТЕМЫ

А.А. Анисимова, А.Н. Моисеев
Томский государственный университет
siberienne94@yandex.ru, moiseev.tsu@gmail.com

Введение

В последнее время все большую роль стали играть компьютерные и телекоммуникационные системы, которые подходят под определение систем массового обслуживания [1]: они предназначены для удовлетворения массовых запросов на выполнение каких-либо услуг. Классические модели систем массового обслуживания подразумевают, что заявка, застав обслуживающий прибор занятым, встает в очередь или покидает систему. В данной статье рассматривается модель, в которой заявка в такой ситуации уходит в источник повторных вызовов, откуда через случайные моменты времени пытается вновь встать на обслуживание. Такие модели описываются в виде систем массового обслуживания с повторными вызовами или RQ-систем (Retrial queueing system) [2].

Одной из важнейших характеристик RQ-систем является распределение вероятностей числа заявок в источнике повторных вызовов. К сожалению, получить аналитическое решение относительно данного распределения слишком сложно, поэтому удобнее разработать программную модель, имитирующую поведение системы.

1. Описание модели

Рассмотрим двухфазную RQ-систему (рис. 1).

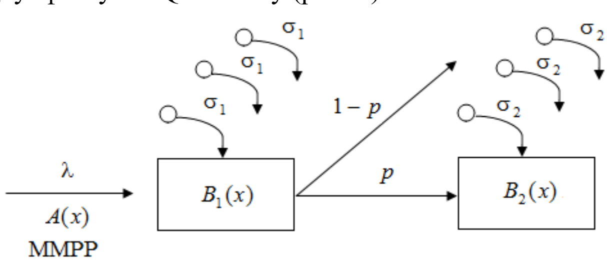

Рис. 1. Модель двухфазной RQ-системы

В данной работе рассматриваются модели с входящими потоками заявок трех видов: простейшим, рекуррентным или MMPP. Поступив в систему, заявка вначале проходит первую фазу, которая содержит прибор, обслуживающий заявку по заданному закону распределения $B_1(x)$, и источник повторных вызовов (ИПВ). Если заявка застает прибор занятым, она отправляется в ИПВ, откуда через случайные моменты времени, имеющие экспоненциальное распределение с параметром σ_1, предпринимает попытки вновь обратиться за обслуживанием. После обслуживания на первой фазе заявка с вероятностью p отправляется на вторую фазу, которая содержит прибор с другим распределением времени обслуживания $B_2(x)$ и ИПВ с параметром времени задержки σ_2. С вероятностью $(1-p)$ заявка считается обработанной и покидает систему.

2. Имитационная модель

Имитационная модель данной системы строится на основе дискретно-событийного подхода, учитывающего, что система изменяет свои характеристики не непрерывно, а в определённые моменты времени, которые называются событиями. Основными типами событий являются:

– поступление заявки в систему;
– поступление заявки из ИПВ;
– завершение обслуживания заявки на приборе и ее передача на другой прибор, либо уход из системы;
– завершение моделирования.

Поступление заявки в систему моделируется в соответствии с выбранным пользователем типом потока. Для простейшего потока интенсивности λ генерируются независимые экспоненциально распределенные интервалы между событиями потока: $\tau_1 = t_1 - t_0$, $\tau_2 = t_2 - t_1, \ldots, \tau_i = t_i - t_{i-1}, \ldots$ по формуле $\tau_i = -\frac{\ln(\alpha)}{\lambda}$, где α — случайная величина, равномерно распределенная на отрезке $(0;1)$. Таким образом, каждое последующее событие потока генерируется по формуле: $t_i = t_{i-1} - \frac{\ln(\alpha)}{\lambda}$, $t_0 = 0$ [3].

MMPP-поток задается матрицей инфинитезимальных характеристик $\mathbf{Q}$ с элементами $\|q_{ij}\|_{n\times n}$ и вектором $\mathbf{\Lambda} = \{\lambda_1, \ldots, \lambda_n\}$, содержащим значения интенсивности потока при каждом состоянии управляющей цепи Маркова. Для определения начального состояния вычисляется стационарное распределение вероятностей состояний цепи. Согласно этому распределению определяется начальное состояние *i*. По формуле $T_i = \frac{\ln(\alpha)}{q_{ii}}$ рассчитывается время пребывания цепи Маркова в *i*-ом состоянии, в течение которого генерируются события простейшего потока с интенсивностью λ_i. По истечении этого времени цепь Маркова должна перейти в следующее состояние. Для его нахождения вычисляются переходные вероятности $p_{ij} = -\frac{q_{ij}}{q_{ii}}$, $i \neq j$, в соответствии с которыми определяется новая интенсивность наступления событий [3].

В рекуррентном потоке интервалы τ_i, $i \geq 2$, независимы и одинаково распределены с функцией распределения $A(x)$, а интервал τ_1 имеет распределение $A_1(x)$. Так как для получения адекватных результатов моделирования приходится генерировать очень большое число заявок во входящем потоке (1 млн. и более), этим условием можно пренебречь и моделировать все интервалы, используя функцию $A(x)$.

3. Приложение для имитационного моделирования

Приложение позволяет выбрать один из трех указанных выше типов входного потока заявок, причем для рекуррентного потока реализованы следующие распределения:

– двухпараметрическое гамма-распределение;
– двухпараметрическое бета-распределение;
– равномерное в заданном интервале;
– детерминированное.

Распределение времени обслуживания на каждой фазе также выбирает пользователь. Оно может быть экспоненциальным или иметь одно из вышеперечисленных распределений. Распределение времени между повторными вызовами имеет экспоненциальное распределение с заданным пользователем параметром.

Архитектура приложения реализована на основе объектной модели [4]. Для реализации дискретно-событийного механизма моделирования введен специальный класс Event, предназначенный для обработки событий внутри модели. Каждое событие связано с определенной заявкой. Класс Call реализует объект заявки, а процесс ее обработки заключается в генерации и последующей передаче элементам системы в зависимости от обрабатываемого события. Связь этих классов показана на рис. 2.

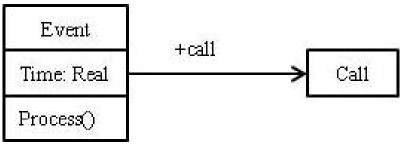

Рис.2. Модель событий

На рис. 3 представлены основные классы предметной области. Класс QueueSimulationModel отвечает за процесс моделирования. Он имеет поле Timer, реализующее таймер модельного времени, который после обработки очередного события сдвигается на момент следующего, и операцию DoStep(), выполняющую один шаг моделирования.

Все события заносятся в отсортированный по времени журнал событий (класс Journal), предназначенный для дискретно-событийного управления модельным временем.

При такой организации один шаг моделирования реализуется следующим образом: модель извлекает из журнала ближайшее событие (операция ReturnEvent()) и определяет его тип, в зависимости от которого переадресует обработку события одному из обрабатывающих элементов.

В общем случае, такие элементы имеют следующие операции:

– GenerateEvent() генерирует события в зависимости от заданного типа входящего потока или распределения времени обслуживания/времени между повторными вызовами;
– Accept() принимает на обслуживание заявку, соответствующую данному событию;
– ProcessEvent() производит обработку события.

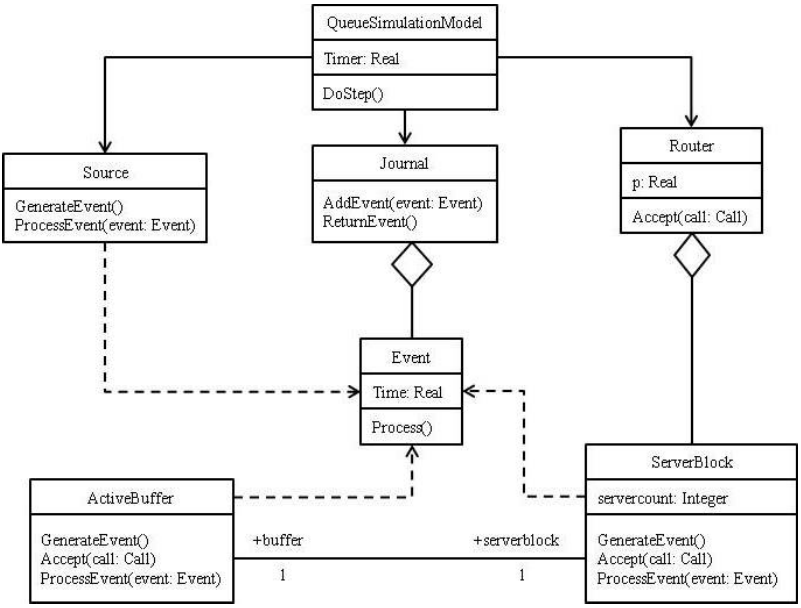

Рис.3. Иерархия элементов модели

Класс Source (источник заявок) генерирует входящий поток событий, а их обработка заключается в передаче соответствующей заявки маршрутизатору Router. Класс Router направляет заявку на очередную фазу, в зависимости от того, была ли она уже обработана, а также с учетом вероятности *p* продолжения обслуживания на второй фазе.

Класс ServerBlock реализует блок обслуживающих приборов, соответствующих одной фазе. После того, как заявка начала обслуживание, счетчик serverCount увеличивается на единицу (по завершении обслуживания уменьшается). Если заявка обращается за обслуживанием в тот момент, когда serverCount равен 1, она передается в ИПВ, реализованный посредством класса ActiveBuffer, откуда предпринимает дальнейшие попытки обслужиться.

На рис. 4 представлено главное окно программы. Пользователь задает число заявок, которые должны быть сгенерированы во входящем потоке, распределение входного потока, распределение времени обслуживания и параметры экспоненциального распределения времени ожидания в ИПВ на первой и второй фазе, а также вероятность, с которой заявка может покинуть систему после окончания обслуживания на первой фазе.

Результатом моделирования является двумерное распределение числа заявок в ИПВ первой и второй фазы, который сохраняется в файле. Также программа выводит на экран маргинальные распределения вероятностей числа заявок в ИПВ каждой фазы в виде полигона частот и сохраняет их в виде рядов распределений. Здесь же можно вывести полигон частот для выбранного распределения, чтобы провести сравнение с результатами имитационного моделирования. Программа позволяет рассчитывать вектор математических ожиданий и матрицу ковариаций этих распределений.

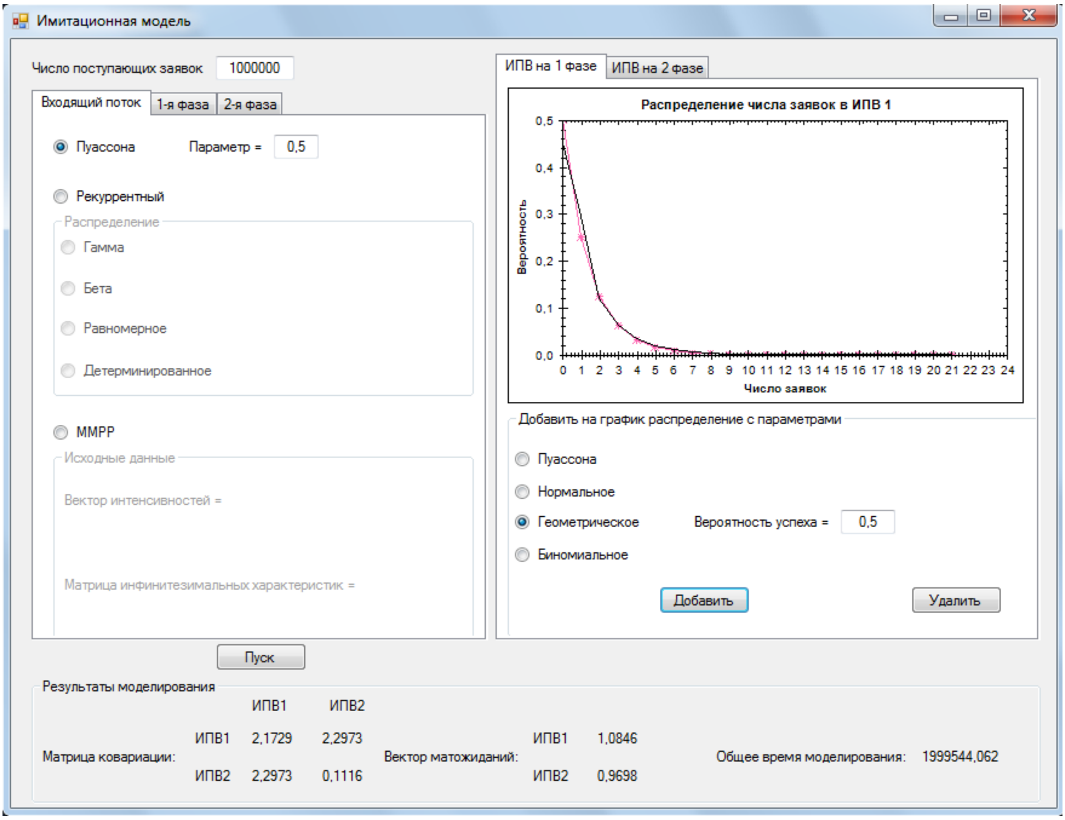

Рис.4. Главное окно программы

Заключение

В работе представлено описание имитационной модели и приложения для моделирования двухфазной RQ-системы. Разработанное приложение позволяет получить основные вероятностные характеристики для сравнения с аналитическими результатами, полученными для данной системы.

ЛИТЕРАТУРА

1. *Гнеденко Б. В.* Введение в теорию массового обслуживания. – 2-е изд., перераб. и доп. / Б.В. Гнеденко, И.Н. Коваленко. – М. : Наука. Гл. ред. физ.-мат. лит., 1987. – 336 с.

2. *Artalejo J.R., Gomez-Corral A.* Retrial Queueing Systems: A Computational Approach. – Berlin: Springer, 2008. – 267 p.

3. *Даммер Д.Д.* Имитационное моделирование: учебно-методическое пособие / Д.Д. Даммер, Н.Ю. Марголис, С.А. Цой. – Томск : ТГУ, 2010 – 32 с.

4. *Моисеев А.Н., Синяков М.В.* Разработка объектно-ориентированной модели системы имитационного моделирования процессов массового обслуживания // Вестник Том. гос. ун-т. Управление, вычислительная техника и информатика. 2010, №1, С. 89–93.

ИССЛЕДОВАНИЕ СУММАРНОГО ОБЪЁМА ТРЕБОВАНИЙ В БЕСКОНЕЧНОЛИНЕЙНОЙ СМО С РЕКУРРЕНТНЫМ ВХОДЯЩИМ ПОТОКОМ

В.А. Колбасова, Е.Ю. Лисовская
Томский государственный университет
vera-kolbasova@mail.ru

Настоящая работа посвящена проблеме определения характеристик систем массового обслуживания (СМО) требований случайной длины, представляющих собой модели реальных информационных систем с ограниченным объёмом памяти, в которой хранится информация об обрабатываемых сообщениях. Более подробно задачи, возникающие при проектировании реальных информационных систем, моделируемых с помощью указанных СМО, а также примеры решения таких задач и классификация моделей рассмотрены в [1].

Рассмотрим систему массового обслуживания с неограниченным числом приборов, на вход которой поступает рекуррентный поток заявок, заданный функцией распределения вероятностей $A(x)$ длин интервалов между наступлением событий. Продолжительности обслуживания заявок стохастически независимы, одинаково распределены и имеют экспоненциальное распределение с параметром μ. Заявка, поступившая в систему, занимает любой из свободных приборов, завершив обслуживание, она покидает систему. Будем предполагать, что каждое требование характеризуется некоторым объёмом $\xi_i > 0$. Введем обозначение $V(t) = \sum_{i=0}^{i(t)} \xi_i$ — суммарный объём заявок, находящихся в системе. Объёмы различных требований представляют собой независимые случайные величины с функцией распределения вида $F(x) = P\{\xi < x\}$.

Ставится задача определения основных характеристик суммарного объёма заявок в системе.

Пусть $i(t)$ — число заявок в системе, иначе говоря, число приборов, занятых в момент времени *t*. Для суммарного объёма находящихся в системе требований запишем характеристическую функцию в виде

$$H(u) = M\left\{e^{ju\sum_{i=1}^{i(t)}\xi_i}\right\} = M\left\{e^{u\sum_{i=1}^{i(t)}\xi_i} \mid P(i(t)=i)\right\} =$$