

Моделирование немарковской СМО с коррелирующим входным потоком заявок

09, сентябрь 2014

Орлова А. А.

УДК: 519.872

Россия, МГТУ им. Баумана

orlova.anastasia@list.ru

Введение

Известно, что при исследовании систем массового обслуживания в основном рассматриваются системы с ожиданием и системы с потерями. Наблюдается также возрастающий интерес к системам с источником повторных вызовов, так называемых RQ-системах (Retrial Queuing System) [1]-[3]. Однако, в большинстве работ практически все внимание уделяется именно самой СМО, полагая что входящий поток является марковским и чаще всего – простейшим. В реальных системах входящий поток далеко не всегда является марковским и тем более простейшим. В данной статье рассматривается система с коррелирующим потоком заявок, описываются различные модели поведения системы при обработке неоднородного потока заявок и предлагаются результаты моделирования в разработанной подсистеме имитационного моделирования.

1. Постановка задачи

Моделирование немарковских системы массового обслуживания с неоднородным входным потоком заявок является весьма актуальной задачей. Рассмотрим информационную систему (далее ИС), которая представляет собой интеграционную платформу, обеспечивающую взаимодействие отдельных информационных систем в составе большой распределенной системы обработки информации.

В основе работы данной ИС лежит обработка разнородного потока заявок, связанных между собой. Обрабатываемая заявка может ссылаться на заявку, которая еще только находится в очереди на обслуживание или получила отказ на обслуживание, что приводит к приостановке обработки текущей заявки. При этом количество заявок, на которые можно ссылаться, не ограничено. Заявка, обработка которой была приостановлена, повторно поступает на обработку тогда, когда все заявки, на которые она ссылалась, успешно были обработаны.

Время обслуживания заявки в ИС без учета связей с другими заявками в общем случае подчиняется нормальному закону распределения. В случае если заявки становятся взаимосвязанными, время обслуживания в ИС перестает подчиняться какому-либо закону

распределения и является случайной величиной. Кроме того, в ИС при обработке заявки проверяется наличие связанных заявок, т.е. в системе присутствует механизм памяти, что, так же как и отсутствие определенного закона распределения для времени обработки заявки, является одной из характеристик немарковской системы.

Для подобных немарковских систем достаточно трудно, а порой и невозможно построить аналитическую модель системы, в связи с чем необходимо построить имитационную модель для того, чтобы проанализировать процесс обработки коррелирующего потока в зависимости от того, каким образом заявка обрабатывается в системе. Необходимо так же исследовать возможные варианты обработки заявки для случая отсутствия связанных заявок.

2. Варианты использования

Существующие на данный момент системы имитационного моделирования [4]-[6] поддерживают анализ марковских моделей, поскольку моделирование немарковской системы в общем случае требует значительных аппаратных ресурсов и, как правило, отдельных алгоритмов функционирования. Поэтому, для моделирования обработки заявок в рассматриваемой ИС разработана подсистема имитационного моделирования основанная на классическом дискретно-событийном методе протяжки модельного времени, особенностью которой является работа с несколькими вариантами обработки входного потока заявок.

Реализованы следующие модели поведения (здесь и далее под моделью поведения понимается способ обработки заявок в обслуживающем аппарате, проверяющем наличие связанных объектов у заявки):

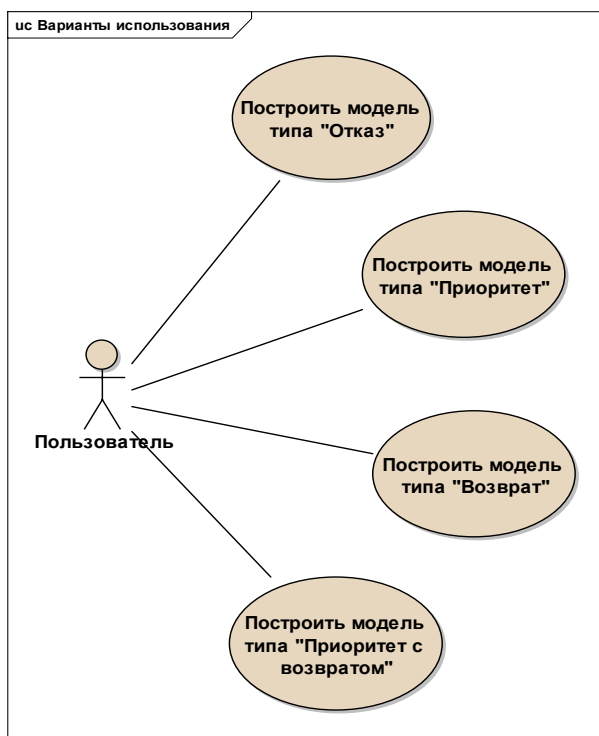


Рис. 1. Варианты использования системы моделирования для построения модели

Рассмотрим подробнее каждую модель поведения.

2.1 Модель «Отказ»

Наиболее простым вариантом обработки заявок является модель поведения типа «Отказ». Обработка заявки, для которой не найдено связанных заявок, завершается с ошибкой. Для данного варианта характерны высокий процент ошибок и достаточно высокая скорость обработки потока заявок.

2.2 Модель «Приоритет»

Современные системы промежуточного ПО, предоставляющие механизмы передачи сообщений (Message-Oriented Middleware), поддерживают приоритеты сообщений. Кроме того данная возможность реализуется так же в POSIX MQ – механизме, поддерживаемом во всех UNIX-системах. Основная идея данной модели состоит в назначении приоритетов сообщениям входного потока. Расстановка приоритетов должна осуществляться на основании зависимостей заявок различных типов между собой. Например, известно, что заявки типа 1 всегда зависят от заявок типа 2, благодаря чему приоритет заявок 2 должен быть всегда выше приоритета заявок типа 1. Обработка заявок из входной очереди должна вестись с учетом приоритета заявки: даже если заявка пришла позже всех, но у нее наивысший приоритет, то она будет обработана первой. Если для обрабатываемой заявки отсутствуют связанные заявки, то она считается ошибочной.

2.3 Модель «Возврат»

Данная модель предполагает достаточно простой вариант борьбы с ошибками – если заявка не может быть обработана, то она возвращается в конец очереди. Главным недостатком такого подхода является то, что для ряда заявок резко возрастает время обслуживания, что может быть критично при наличии соответствующих требований к системе. Однако, такой подход в теории должен сводить процент ошибок к нулю.

2.4 Модель «Приоритет с возвратом»

Данный вид модели представляет собой комбинированный способ обработки заявок. Он сочетает в себе варианты «Возврат» и «Приоритет». Согласно такой модели сообщения выбираются из очереди по приоритетам, а в случае отсутствия связанных объектов в ИС-получателе возвращаются во входную очередь. В данном варианте так же возможно указание максимально допустимого количества возвратов сообщения в очередь.

3. Формализация объекта «Заявка» и входного потока

Заявка в ИС имеет идентификатор и тип. Связь заявки с другими заявками аналогично информационным объектам в ИС указывается посредством указания множества идентификаторов связанных заявок.

Так же для заявки должен указываться ее приоритет. Приоритет заявки в дальнейшем будет использован при разрешении ситуаций отсутствия связанных объектов.

Для целей моделирования и сбора статистики требуется указывать время генерации заявки (аналогичное поле есть у сообщений в ИС), а так же время ее жизни.

Таким образом, объект «заявка» может быть представлен следующим образом:

$$I = (Id, C, P, Q, T_0, T),$$

где

Id – идентификатор заявки;

C – тип заявки;

P – приоритет;

Q – множество идентификаторов Id связанных заявок;

T_0 – время генерации заявки;

T – время жизни заявки.

Входной поток заявок представляет собой неупорядоченный набор заявок, поступающих на вход обрабатывающему узлу системы моделирования.

Поток заявок характеризуется следующими параметрами $F = (M, N)$, где:

M – множество заявок I ;

N – количество заявок.

4. Функциональные требования к программному обеспечению имитационного моделирования немарковских СМО

В процессе моделирования реализуются обработка коррелирующего потока заявок, ссылающихся друг на друга. При поступлении новой заявки система (ее конкретный элемент, обслуживающий аппарат) проверяет, проходили ли связанные заявки в системе, если нет, то заявка должна быть перемещена в начало очереди. Очевидно, что в таком случае должна быть реализован некоторый запоминающий аппарат в системе, позволяющий сохранять связи между заявками, что является характерной чертой немарковских систем.

Так же на систему накладывается требование на количество обрабатываемых заявок в один момент времени – 1. Основанием для такого ограничения в программной реализации является то, что сама система нами рассматривается как один обслуживающий аппарат, декомпозированный в программной реализации на N обслуживающих.

Для реализации данного ограничения необходимо реализовать своеобразные шлюзы на первом и последнем обслуживающих аппаратах, останавливающих и возобновляющих обработку заявок из входной очереди.

Требования к обслуживающему аппарату:

- Для начального аппарата прекращать обслуживание заявок из очереди пока не будет получен сигнал от последнего аппарата или аппарата, завершившего обработку заявки;
- Обслуживающий аппарат должен поддерживать работу с несколькими очередями;
- Обслуживающий аппарат должен поддерживать обработку нескольких типов заявок;
- Аппарат, завершающий обработку заявки, должен отправлять сигнал первому аппарату для возобновления приема заявок.
- Сигнал о возобновлении приема заявок должен представлять собой заявку специального типа.

5. Генерация входного потока

Генерация входного потока, заявки которого коррелируют между собой, основывается на анализе графа связей заявок системы между собой.

На рисунке 2 представлены три типа заявок и связи между ними:

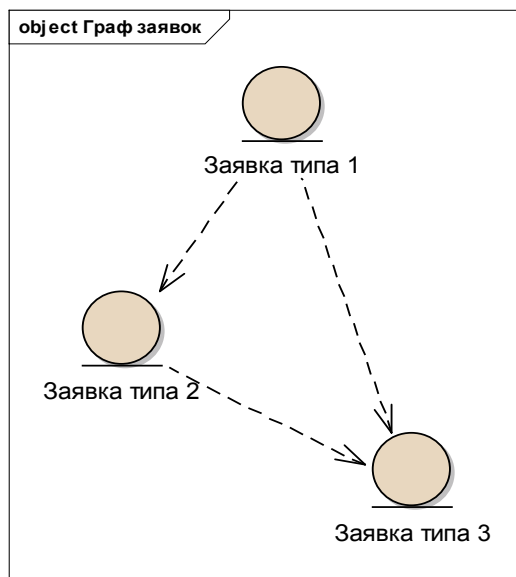


Рис. 2. Заявки различных типов и их взаимосвязи

Алгоритм генерации входного потока выглядит следующими образом:

1. Сгенерировать стартовую заявку.
2. Стартовая заявка выбирается произвольным образом. Стартовая заявка становится текущей.
3. Для текущей заявки для каждого из возможных видов связей сгенерировать связанную заявку и назначить связь.
4. Для каждой связанной заявки повторить рекурсивно п.2.
5. Поместить во входную очередь сгенерированные заявки.
6. Повторить шаги 1-3 K раз (настраиваемый параметр).
7. Перемешать входную очередь.

Однако у данного алгоритма есть один недостаток – если перемешивать входную очередь равномерно, то разброс связанных заявок может оказаться очень большим, что не соответствует входному потоку в реальных системах. Поэтому перемешивать заявки предлагается следующим алгоритмом:

1. Обозначим размер набора сгенерированных за 1 шаг заявок – N .
2. Задать начальные значения интервала $[I_{start}; I_{end}]$ - $I_{start} = 1$; $I_{end} = M*N$, где M – параметр обозначающий, сколько наборов заявок будет перемешиваться за одну итерацию; K (общее число наборов заявок) должно нацело делиться на $M/2$.
3. Выбрать из входной очереди заявки $[I_{start}; I_{end}]$ и перемешать их.
4. Увеличить $I_{start} = I_{start} + M*N/2$; $I_{end} = I_{end} + M*N/2$;

5. Повторить шаги 3-4 до тех пор, пока $I_{\text{end}} < K \cdot N$;

Отдельно нужно рассмотреть приоритезацию входного потока. Для случая, представленного на рисунке 2, приоритеты назначаются просто: «Заявка типа 1» – наивысший, «Заявка типа 2» – средний, «Заявка типа 3» – низший.

Назначение приоритетов в общем случае выполняется по алгоритму нахождения кратчайшего пути по графу. Для этого нужно необходимо выбрать начальное состояние или «стартовую» заявку.

Замечание: Приоритеация используется в модели «Приоритет» и от нее зависит порядок выборки заявок из входной очереди. На практике граф связей может содержать рекурсивную связь – заявка одного типа ссылается на заявку такого же типа:

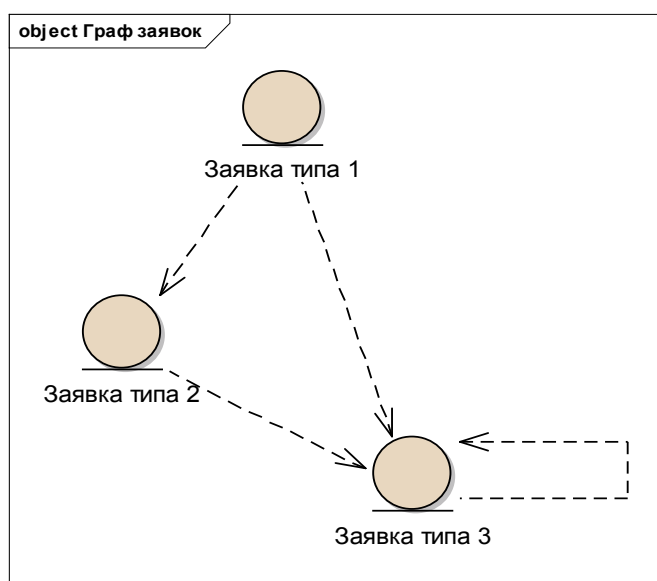


Рис. 3. Заявки нескольких типов и их взаимосвязи, включающие рекурсивный тип связи

Приоритеты в данном случае будут расставлены так же, но модель поведения «Приоритет» для такого графа связей может не дать 0% ошибок, т.к. заявки, соответствующие связным заявкам могут быть так же перемешаны. Таким образом, главным условием отсутствия ошибок для модели «Приоритет» является отсутствие циклов в графе связей заявок.

6. Реализация

Согласно приведенных функциональных требований, а так же общих функциональных требований, не рассматриваемых в данной статье, разработана система имитационного моделирования, позволяющая реализовать перечисленные в п.2 модели поведения системы при обработке заявок.

Целью проведения экспериментов с различными моделями поведения системы является анализ выходных параметров моделирования для каждого типа модели поведения.

Основными параметрами, на основе которых выполняется сравнение и анализ, являются:

- процент ошибок;
- минимальное время обслуживания заявки;
- среднее время обслуживания заявки;
- максимальное время обслуживания заявки;
- максимальное количество возвратов заявки в очередь;
- среднее количество возвратов заявки в очередь;
- максимальная длина очереди;
- время моделирования.

Таблица 1. Порядок построения моделей обработки заявок в ИС

Этап	Описание этапа	Комментарий
Построение модели.	На данном этапе в системе имитационного моделирования собирается модель, задаются начальные и конечные обслуживающие аппараты, выбираются аппараты, проверяющие наличие связей.	Данный этап является общим для всех моделей поведения.
Настройка типов заявок	Задаются возможные типы заявок и возможные связи между ними	Общий этап для всех моделей поведения
Задание приоритетов	Задаются приоритеты типов заявок автоматически или вручную	Данный этап необходим для проведения эксперимента с моделью поведения «Приоритет»
Настройка проверяющих аппаратов	Задается необходимая модель поведения	Вариативный пункт плана
Запуск симуляции	Запускается симуляция модели с указанием объема входного потока	Параллельно с процессом симуляции собирается необходимая статистика
Анализ результатов	Визуализация собранной статистики и подведение итогов	

7. Результаты

Рассмотрена реальная ИС, обрабатывающая неоднородный входной поток заявок, взаимосвязанных между собой. В результате анализа состава входного потока ИС, построен граф связей заявок (см. рисунок 4).

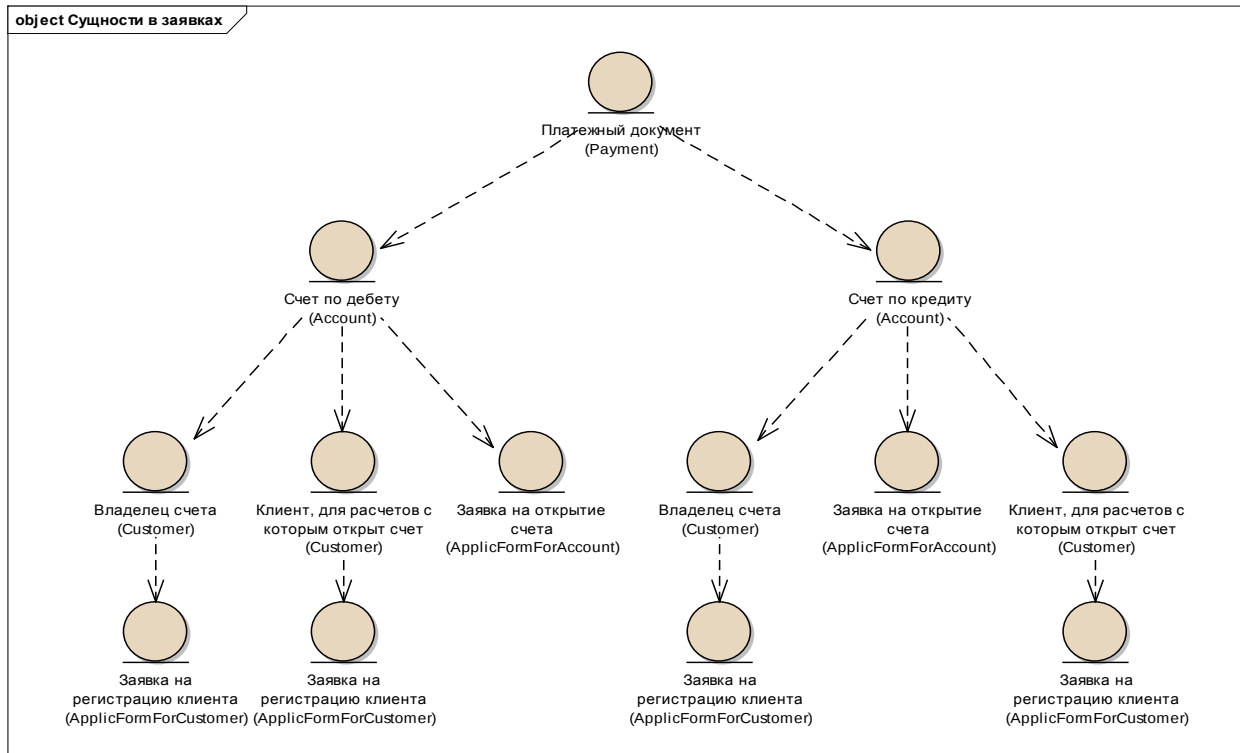


Рис. 4. Граф связей заявок для реальной ИС

Результаты экспериментов приведены далее в таблице 2.

Как видно по сводной таблице результатов моделирования наиболее оптимальными вариантами с точки зрения отсутствия ошибок при обработке являются модели поведения «Возврат» и «Задержка». Оба этих варианта не являются самыми быстрыми (по времени моделирования), однако, они самые надежные (по проценту ошибок). В случае если приоритетом является скорость обработки, то оптимальным вариантом будет модель «Приоритет». Среднее время обработки незначительно больше модели «Отказ», максимальное время обработки также сравнимо, однако процент ошибок значительно меньше.

Таблица 2. Результаты экспериментов

	Процент ошибок	Мин. время	Среднее время	Макс. время	Макс. количество возвратов	Среднее количество возвратов	Максимум в очереди	Время моделирования	Среднее время генерации 1 заявки	Среднее время обработки 1 заявки
Отказ	54	4	117	235	0	0	31	735	5	10
Приоритет	3	5	242	860	0	0	49	990		
Возврат	0	20	525	1185	9	1	69	1445		
Приоритет с возвратом	0	10	272	945	3	0	52	1020		
Отказ	54	5	8	12	0	0	1	815	8	10
Приоритет	54	5	8	12	0	0	1	815		
Возврат	2	21	352	939	10	1	47	1435		
Приоритет с возвратом	0	10	152	985	6	0	26	1065		
Отказ	54	5	7	10	0	0	1	1010	10	10
Приоритет	54	5	7	10	0	0	1	1010		
Возврат	2	20	251	765	10	1	33	1440		
Приоритет с возвратом	0	10	122	1100	6	0	18	1180		
Отказ	54	5	7	10	0	0	1	1210	12	10
Приоритет	54	5	7	10	0	0	1	1210		
Возврат	2	22	166	515	10	1	20	1470		
Приоритет с возвратом	0	10	88	1225	6	0	11	1310		
Отказ	54	5	7	10	0	0	1	2010	20	10
Приоритет	54	5	7	10	0	0	1	2010		
Возврат	23	10	23	40	10	3	2	2020		
Приоритет с возвратом	23	10	31	135	10	3	3	2020		

Заключение

В данной статье были рассмотрены различные способы обработки коррелирующего потока заявок в немарковской СМО, сформулированы требования к программному обеспечению имитационного моделирования немарковских СМО, а так же приведены результаты анализа рассмотренных способов обработки входного потока для модели, основанной на реальной системе. Сделаны выводы о применимости каждого способа обработки входного потока.

Список литературы

1. Назаров А. А., Моисеева Е. А. Исследование RQ-системы $MMPP|M|1$ методом Асимптотического анализа в условии большой нагрузки // Вестник Томского государственного университета. Сер. Управление, вычислительная техника и информатика. 2013. №4. С. 84-94.
2. Назаров А. А., Семенова И. А. Исследование RQ-систем методом асимптотических семиинвариантов // Известия Томского Политехнического Университета. Сер. Управление. Вычислительная техника и информатика. 2010. №3. С. 85-96.
3. Назаров А. А., Моисеева С. П., Морозова А. С. Исследования СМО с повторным обращением и неограниченным числом обслуживающих приборов методом предельной декомпозиции // Вычислительные технологии. 2008. Т. 13, Специальный выпуск. С. 272-277.
4. Обзор — Инструмент имитационного моделирования AnyLogic. Режим доступа: <http://www.anylogic.ru/overview> (дата обращения 15.05.2014).
5. Arena Simulation Software by Rockwell Automation Home. Режим доступа: http://www.arenasimulation.com/Arena_Home.aspx (дата обращения 15.05.2014).
6. GPSS World - General Information. Режим доступа: http://www.minutemansoftware.com/general_information.htm (дата обращения 15.05.2014).
7. Рудаков И. В., Шляева А. В. Моделирование входных потоков данных для Стохастических моделей дискретных систем // Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. 2008. №2. С. 65-72.
8. Кельтон В. Дэвид, Лоу М. Аверилл. Имитационное моделирование. СПб.: Питер, 2004. 846с.
9. Лоскутов А. Ю., Михайлов А. С. Основы теории сложных систем. Ижевск: Институт компьютерных исследований, 2007. 620 с.