

## Программная поддержка моделей надежности

© 2010 г. Я. ШТРИК, д-р философии  
(Дебреценский Университет, Дебрецен, Венгрия),  
Д. В. ЕФРОСИНИН, канд. физ.-мат. наук  
(Институт математики, Университет Иоганна Кеплера, Линц, Австрия)

### АНАЛИЗ НАДЕЖНОСТИ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С ПОВТОРНЫМИ ЗАЯВКАМИ И КОНЕЧНЫМ ИСТОЧНИКОМ ТРЕБОВАНИЙ ПРИ ПОМОЩИ ИНСТРУМЕНТАЛЬНЫХ ПРОГРАММНЫХ СРЕДСТВ<sup>1</sup>

Рассматривается роль инструментальных программных средств для моделирования поведения систем массового обслуживания. Представлены 4 крупных центра разработки инструментальных программных средств и показано, как каждый отдельный программный продукт может быть применим для исследования надежности систем массового обслуживания с конечным источником повторных требований в стационарном режиме. Приведены несколько численных примеров, демонстрирующих зависимость интенсивностей отказа и ремонта обслуживающего прибора от среднего времени отклика источников требований.

#### 1. Введение

Аргументом для использования методов оценки показателей эффективности инструментальных программных средств в анализе надежности сложных систем всегда был тот факт, что такие системы не могут быть эффективно спроектированы или модифицированы без помощи некоторой прогнозирующей модели, точно так же как и в других инженерно-технических областях. Чтобы эти методы имели практическое применение, при оценивании показателей производительности используют инструментальные программные средства, которые делают методы оценки показателей эффективности доступными для специалистов по анализу отказов, а это, в свою очередь, зависит от используемых точных методов, которые включают в себя аналитические методы, стохастические модели и имитационное моделирование. Разрабатывается множество математических методов расчета показателей производительности на базе марковских моделей вознаграждения, которые составляют основу для почти всех моделей и программных средств оценки показателей производительности.

Целью данной статьи является представление некоторых инструментальных программных средств и демонстрация того, как каждый конкретный программный продукт может быть применен для исследования надежности систем массового обслуживания (СМО) с конечным источником повторных вызовов в стационарном режиме. Приведены несколько численных примеров, демонстрирующих зависимость интенсивностей отказа и ремонта обслуживающего прибора от среднего времени отклика источников требований.

<sup>1</sup> Работа выполнена при частичной поддержке Венгерского фонда научных исследований № К60698/2005 и Организации Австро-Венгерского научного сотрудничества № 72öи6.

Статья организована следующим образом. В разделе 2 представлены некоторые новейшие инструментальные программные средства анализа производительности сложных систем. В разделе 3 дано полное описание математической модели с помощью соответствующей цепи Маркова. Далее выводятся основные показатели производительности и надежности систем, которые могут быть получены с помощью инструментального программного обеспечения MOSEL. В разделе 4 приводятся несколько численных примеров и даются некоторые комментарии. Статья завершается заключением.

## 2. Некоторые новейшие программные средства моделирования

Далее кратко представлены 4 крупных центра разработки инструментальных программных средств.

### *2.1. Программные средства, разработанные на факультете информатики Университета Дортмунда, Германия*

В этом традиционно известном центре разработан ряд пакетов программ, размещенных на сайте: <http://ls4-www.informatik.uni-dortmund.de/tools.html>. Параллельно с методологической работой в центре непрерывно разрабатывались и применялись инструментальные программные средства для оценки характеристик производительности и качества работы сложных систем. Предназначением этих программных средств является обеспечение способов для описания модели, близкого к исходной спецификации системы, при этом скрывая от пользователя детали методик анализа. Такие инструментальные программные средства автоматически приводят спецификацию модели в соответствие с анализируемой моделью.

### *2.2. Пакет Möbius*

Möbius<sup>TM</sup> является пакетом программных средств для моделирования поведения сложных систем. Это один из основных научно-исследовательских проектов исследовательской группы PERFORM объединенной научной лаборатории Иллинойского университета в Урбана и Шампейне, США. В программном продукте Möbius интегрировано множество новаторских моделей моделирования и инновационных и мощных методов решения. Этот пакет доступен по адресу в Интернете: <http://www.mobius.uiuc.edu/>.

### *2.3. Пакет MOSEL*

Инструментальные программные средства обычно имеют свой собственный графический или текстовый язык спецификации, который зависит, главным образом, от лежащего в основе метода моделирования. Пакет MOSEL (MOdeling, Specification and Evaluation Language, язык моделирования, спецификации и вычисления), разработанный в Эрлангенском университете, Германия, основан на следующей идее. Вместо того, чтобы создавать ее один инструментальный пакет со всеми компонентами, необходимыми для описания системы, формирования фазового пространства, определения случайного процесса и получения численного решения, разработчики программного продукта сфокусировались на компоненте строгого описания системы и сделали возможным подключать к пакету MOSEL и использовать в нем потенциал различных уже существующих и хорошо проверенных пакетов. Для использования существующих инструментальных программных средств для анализа систем программная среда пакета MOSEL снабжена набором трансляторов, которые

преобразуют спецификации модели, полученную в пакете MOSEL, в различные (в зависимости от используемых существующих инструментальных пакетов) описания системы [1].

MOSEL-2 предоставляет пользователю средства, с помощью которых можно непосредственно определить множество интересных характеристик производительности и надежности и представить их графически. Вся информация о программном продукте MOSEL может быть найдена по адресу: <http://www4.informatik.uni-erlangen.de/Projects/MOSEL/>.

#### 2.4. Пакет PRISM

В основе пакета PRISM (PRobabIiStic Model checker, программа проверки вероятностной модели) лежит высокоуровневый язык моделирования для описания систем, проявляющих вероятностное поведение, с помощью моделей, основанных как на теории цепей Маркова с непрерывным временем, так и на теории цепей Маркова с дискретным временем и марковских процессах принятия решений. Для описания свойств системы пакет PRISM использует непрерывную стохастическую логику (CSL) для цепей Маркова с непрерывным временем и вероятностную логику деревьев вычислений (PCTL) для цепей Маркова с дискретным временем и марковских процессов принятия решений. Обе логики являются расширением логики деревьев вычислений (CTL), темпоральной логики, которая используется в различных классических программах проверки модели для определения свойств системы. Пакет PRISM доступен для скачивания по адресу <http://www.prismmodelchecker.org>.

### 3. Анализ надежности СМО с конечным источником повторных вызовов

Чтобы продемонстрировать пример применения пакета MOSEL, проанализируем поведение системы массового обслуживания с повторными вызовами в следующих предположениях. Рассмотрим СМО с одним обслуживающим прибором, в которой имеется  $K$ ,  $1 < K < \infty$  однородных источников первичных вызовов. Обслуживающий прибор может находиться в трех состояниях: свободен, занят, в отказе. Если обслуживающий прибор свободен, он может обслуживать поступающие от источников вызовы. Каждый из источников вызовов может находиться в трех состояниях: свободен, посылает повторные вызовы и находится в обслуживании. Если источник требований свободен в момент  $t$ , то он может с вероятностью  $\lambda dt + o(dt)$  сгенерировать за временной интервал  $(t, t + dt)$  один первичный вызов. Если обслуживающий прибор свободен в момент поступления вызова, тогда вызов начинает обслуживаться немедленно, а соответствующий источник вызовов переходит в состояние “занят”. С вероятностью  $\mu dt + o(dt)$  обслуживание завершается за время  $(t, t + dt)$ , если обслуживающий прибор доступен. Если обслуживающий прибор занят в момент поступления вызова, то источник вызовов начинает генерировать пуассоновский поток повторных вызовов с интенсивностью  $\nu$  до тех пор, пока обслуживающий прибор не освободится. После обслуживания источник освобождается и может генерировать новый первичный вызов, обслуживающий прибор также становится незанятым и может обслуживать новый вызов. За временной интервал  $(t, t + dt)$  обслуживающий прибор может отказать с вероятностью  $\delta dt + o(dt)$ , если он незанят, и с вероятностью  $\gamma dt + o(dt)$ , если занят. Если  $\delta = 0$ ,  $\gamma > 0$  или  $\delta = \gamma > 0$ , то можно говорить об *активных и независимых отказах*, соответственно. Если обслуживающий прибор выходит из строя, находясь в состоянии занятости, то он после починки либо *продолжает обслуживать* вызов, на котором обслуживание прервалось, либо прерванный на обслуживание вызов *направляется на так называемую орбиту*. Время ремонта обслуживающего прибора распределено экспоненциально с конечным средним  $1/\tau$ . Если обслуживающий прибор вышел из строя, то можно рассматривать

два различных случая. А именно, первый случай – случай заблокированных источников, когда все операции блокируются, т.е. прекращается генерация как первичных, так и повторных вызовов. Второй случай – случай неблокированных (умных) источников, когда прекращается только обслуживание заявок, а все остальные операции продолжают выполняться (первичные и повторные вызовы могут генерироваться). Предполагается, что все длительности в рамках данной модели взаимно независимы. Более подробную информацию о системах подобного типа можно найти в работах [2–4].

Основная цель настоящей статьи – продемонстрировать, как в стационарном режиме показатели работоспособности, такие как среднее число вызовов на орбите или в обслуживании, среднее время ожидания обслуживания, зависят от интенсивностей отказа и ремонта обслуживающего прибора. Для достижения этой цели используется пакет MOSEL, с помощью которого формулируется и решается задача.

Состояние системы в момент времени  $t$  может быть описано процессом  $X(t) = (Y(t); C(t); N(t))$ , где  $Y(t) = 0$ , если обслуживающий прибор находится в рабочем состоянии,  $Y(t) = 1$ , если он вышел из строя,  $C(t) = 0$ , если обслуживающий прибор свободен,  $C(t) = 1$  в противном случае,  $N(t)$  – число источников повторных вызовов в момент  $t$ . В силу экспоненциальности распределений задействованных в модели случайных величин, этот процесс является цепью Маркова с конечным пространством состояний. Поскольку фазовое пространство процесса  $(X(t), t \geq 0)$  конечно, процесс эргодический для всех допустимых значений интенсивностей, задействованных в построении модели, следовательно, далее будем полагать, что система находится в стационарном режиме.

Определим стационарные вероятности:

$$P(q; r; j) = \lim_{t \rightarrow \infty} P(Y(t) = q, C(t) = r, N(t) = j),$$

$$q = 0, 1, \quad r = 0, 1, \quad j = 0, \dots, K^*,$$

где

$$K^* = \begin{cases} K - 1 & \text{для случая заблокированных источников,} \\ K - r & \text{для случая неблокированных источников.} \end{cases}$$

Преимущество использования инструментальных программных средств заключается в том, что при этом не придется сталкиваться с вычислительными трудностями при нахождении стационарных вероятностей. Нужно лишь сосредоточиться на постановке задачи с помощью некоторого языка описания (обычно это сети Петри), и тогда весь процесс решения становится автоматическим. В данном случае описывается множество состояний, и задаются интенсивности переходов цепи Маркова с непрерывным временем и конечным фазовым пространством и с помощью так называемого решателя (решающей программы) вычисляется стационарное распределение. При решении практических задач возникает проблема быстрого роста числа состояний системы, что вызывает вычислительные трудности из-за большого числа неизвестных. Следует отметить, что эта проблема не имеет решения в аналитическом виде. Однако, как всегда при нахождении стационарных вероятностей, может быть применен алгоритмический подход, см. [3]. Это предполагает дополнительную работу по программированию и графическому представлению результатов, которая в данном случае осуществляется с помощью встроенного модуля, поэтому её описание опускается.

С помощью стационарных вероятностей состояний основные показатели производительности вычисляются следующим образом:

- Загрузка обслуживающего прибора

$$U_S = \sum_{j=0}^{K-1} P(0, 1, j).$$

- Загрузка ремонтного устройства

$$U_R = \sum_{r=0}^1 \sum_{j=0}^{K^*} P(1, r, j).$$

- Коэффициент готовности обслуживающего прибора

$$A_S = \sum_{r=0}^1 \sum_{j=0}^{K^*} P(0, r, j) = 1 - U_R.$$

- Среднее число вызовов на орбите или в обслуживании

$$M = E[N(\infty) + C(\infty)] = \sum_{q=0}^1 \sum_{r=0}^1 \sum_{j=0}^{K^*} jP(q, r, j) + \sum_{q=0}^1 \sum_{j=0}^{K-1} P(q, 1, j).$$

- Загрузка источников вызовов

$$U_{SO} = \begin{cases} \frac{E[K - C(\infty) - N(\infty); Y(\infty) = 0]}{K} & \text{в случае блокировок,} \\ \frac{K - M}{K} & \text{при отсутствии блокировок.} \end{cases}$$

- Суммарная загрузка

$$U_O = U_S + KU_{SO} + U_R.$$

- Средняя интенсивность поступления первичных вызовов

$$\bar{\lambda} = \begin{cases} \lambda E[K - C(\infty) - N(\infty); Y(\infty) = 0] & \text{в случае блокировок,} \\ \lambda E[K - C(\infty) - N(\infty)] & \text{при отсутствии блокировок.} \end{cases}$$

- Среднее время ожидания

$$E[T] = M/\bar{\lambda}.$$

#### 4. Численные примеры

Для примера был использован пакет SPNP, который способен производить расчеты для моделей с числом источников до 126. В этом случае время счета на компьютере с процессором с тактовой частотой 1,1 ГГц и оперативной памятью объемом 512 Мб составило приблизительно 1 секунду.

На рис. 1, 2 изображены графики зависимости среднего времени ожидания и среднего числа вызовов на орбите или на обслуживании от интенсивности отказов обслуживающего прибора для надежных и ненадежных систем массового обслуживания с повторными вызовами. На рис. 3, 4 изображены графики зависимости тех же величин от интенсивности ремонта обслуживающего прибора. В таблице записаны входные параметры системы.

Из рисунка 1 видно, что в случае, когда вызов возвращается на орбиту при поломке обслуживающего прибора, время ожидания обслуживания источников всегда

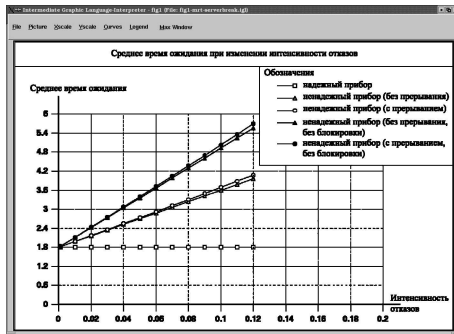


Рис. 1. Зависимость среднего времени ожидания  $E[T]$  от интенсивности отказов.

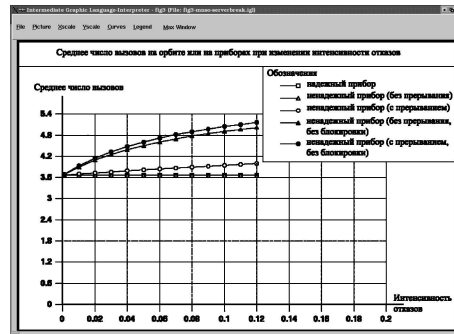


Рис. 2. Зависимость среднего числа вызовов  $M$  от интенсивности отказов.

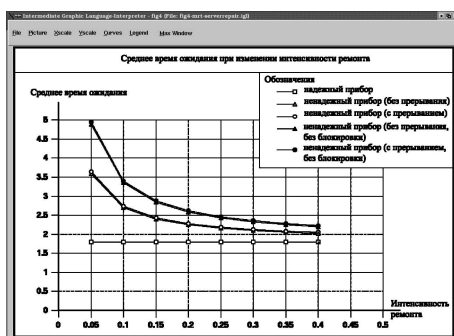


Рис. 3. Зависимость среднего времени ожидания  $E[T]$  от скорости ремонта.

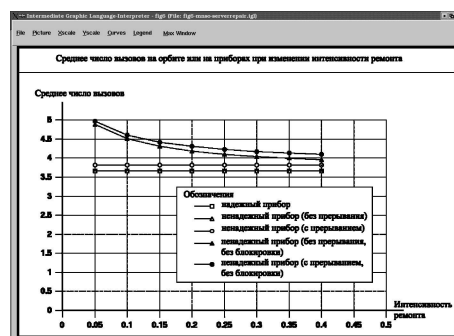


Рис. 4. Зависимость среднего числа вызовов  $M$  от скорости ремонта.

будет расти. Хотя различие незначительно, но оно увеличивается с ростом интенсивности отказов. Почти линейное увеличение  $E[T]$  может быть объяснено следующим образом. В случае блокировки источников вызовов выход из строя обслуживающего прибора блокирует все операции в системе и время ожидания есть сумма времени простоя обслуживающего прибора, времени обслуживания и времени генерации повторного вызова (которое не меняется в течение простоя обслуживающего прибора). Таким образом, отказ обслуживающего прибора влияет на эту числовую характеристику линейно. Единственное отличие случая “умных” источников состоит в том, что источники посылают повторные вызовы во время простоя сервера, поэтому это время простоя не учитывается при суммировании.

Из рис. 2 видно, что среднее число вызовов на орбите или в обслуживании не зависит от интенсивности отказов обслуживающего прибора в непрерывном, блоки-

Входные параметры системы

	K	$\lambda$	$\mu$	$\nu$	$\delta, \gamma$	$\tau$
Рис. 1	6	0,8	4	0,5	ось x	0,1
Рис. 2	6	0,1	0,5	0,05	ось x	0,1
Рис. 3	6	0,8	4	0,5	0,05	ось x
Рис. 4	6	0,1	0,5	0,05	0,05	ось x

рованном случае, что совпадает со случаем надежного обслуживающего прибора. Это объяснимо тем, что во время и после простоя обслуживающего прибора число заявок в этих состояниях остается неизменным. Почти линейный рост среднего числа вызовов при увеличении интенсивности отказов обслуживающего прибора в дискретном, заблокированном случае может быть объяснен тем, что чем чаще происходит выход из строя обслуживающего прибора, тем чаще обслуживающий прибор оказывается незанятым после ремонта до тех пор, пока источник не пошлет повторный вызов.

Из рис. 3 видно, что если заявка возвращается на орбиту после выхода из строя обслуживающего прибора, то время ожидания обслуживания источников вызовов будет расти, как на рис. 1. Разница здесь также невелика, и, как и следовало ожидать, кривые сходятся к прямой, описывающей случай надежного обслуживающего прибора.

На рис. 4 видно, что среднее число заявок на орбите или в обслуживании не зависит от интенсивности ремонта обслуживающего прибора в непрерывном, заблокированном случае, что совпадает со случаем надежного обслуживающего прибора (см. рис. 2).

## 5. Заключение

В работе была исследована однородная система массового обслуживания с конечными источниками повторных вызовов с новой, ранее не изучавшейся особенностью—ненадежным обслуживающим прибором. Для постановки и решения задачи использовался инструментальный программный пакет MOSEL, были получены основные характеристики производительности и надежности, дан графический анализ полученных величин. Для демонстрации зависимости между длительностями простоя и ремонта обслуживающего прибора и средними длительностями ожидания обслуживания требований и средним числом вызовов в системе были произведены некоторые численные подсчеты.

## СПИСОК ЛИТЕРАТУРЫ

1. *Begain K., Bolch G., and Herold H.* Practical performance modeling, application of the MOSEL language// Boston: Kluwer Academic Publisher, 2001.
2. *Almási B., Roszik J., and Sztrik J.* Homogeneous finite-source retrial queues with server subject to breakdowns and repairs// Math. Comput. Modeling. 2005. V. 42. P. 673–682.
3. *Artalejo J.R., Gomez-Corral A.* Retrial Queueing Systems. Berlin: Springer, 2008.
4. *Falin G.I., Templeton J.G.C.* Retrial queues. London: Chapman and Hall, 1997.

*Статья представлена к публикации членом редколлегии Б.Г. Воликом.*

Поступила в редакцию 20.08.2009