

УДК 004.4.242
05.13.18 Математическое моделирование,
численные методы и комплексы программ

Волушкова Вера Львовна
ФГБОУ ВПО «Тверской государственный университет»
Россия, Тверь¹
Доцент, кандидат технических наук
E-Mail: W2L@pisem.net

Школенко Екатерина Юрьевна
ООО «Люксофт Профешнл»
Россия, Санкт-Петербург
Программист
Магистр технических наук
E-Mail: katya.shkolenko@gmail.com

Алгоритмическая реализация модели качества web-сервисов

Аннотация: Объединение отдельных Web-сервисов в потоки предполагает, что приложение может оптимально выбрать поставщиков отдельных сервисов в соответствии с требованиями качества. В вероятностной среде оценка качества составного Web-сервиса не всегда может напрямую вычисляться по оценкам качества его составных частей. Модель оценки качества сервисов учитывает особенности каждого атомарного Web-сервиса. Модель качества основана на вероятностном подходе и рассматривает 5 базовых структурных конструкций, каждая из которых организует сервисы-компоненты уникальным образом. При алгоритмической реализации модели возникают трудности с ростом области определения результирующей случайной величины. Авторы предлагают сократить этот рост с помощью группирующей случайной величины. Погрешность группировки позволяет оценить насколько полученная группирующая случайная величина отличается от исходной. Эта погрешность должна быть минимизирована, т.е. встает задача поиска оптимальной группирующей случайной величины. Для решения данной задачи в статье рассмотрены два алгоритма – рекурсивный и «жадный». На основе ряда тестов было принято решение о применении «жадного» алгоритма для получения группирующей случайной величины. В работе описаны эксперименты по вычислению времени отклика реального web-сервиса по обработке заказа для интернет-магазина. Результат эксперимента позволяет сделать вывод о том, что оценка качества композитного Web-сервиса, полученная с помощью предложенной методики, гарантирует, что сервис работает на рассчитанном уровне качества.

Ключевые слова: Web-сервис; композитный Web-сервис; качество Web-сервиса; показатели качества; время отклика; надежность; оптимизация; рекурсивный алгоритм; «жадный» алгоритм; организация сервис-компонентов.

Идентификационный номер статьи в журнале 85TVN114

¹ 170004, г.Тверь, Садовый переулок, 35, ауд. 310а

Vera Volushkova
Tver State University
Russia, Tver
E-Mail: W2L@pisem.net

Ekaterina Skolenko
ООО «Luxoft Production»
Russia, Saint-Petersburg
E-Mail: katya.shkolenko@gmail.com

Programming Approach for Modeling of web services' quality

The Abstract: Joining web services into web-service chains supposes that applications can choose separate web-services' providers according to requirements of quality in the best way. In the terms of probability the measure of quality of some composite web service cannot always be simply calculated using measures of quality of its parts. The described model of quality of composite web service takes into account the characteristics of each of the atomic web services. The model of quality is based on stochastic approach. The model of quality considers 5 basic service compositions. In case of algorithmic implementation of model there are difficulties with growth of definition range of a resultant random variable. In this paper we suggest reducing this growth by grouping random variable. An error of grouping allows evaluating the difference between the grouping random variable and the initial one. The problem is to find the best grouping random variable. We consider the recursive and "greedy" algorithms to solve this problem. As a result of some tests the decision to apply the "greedy" algorithm to get grouping random variable was made. We describe the experiments on computation of response time of a real web service for an e-commerce shop. The result of experiment shows that the quality of the composite Web service, got by means of the offered technique, guarantees that service will work at the calculated quality level.

Keywords: web service; composite web service; quality of web service; quality of service attribute; response time; reliability; optimization; recursive algorithm; "greedy" algorithm; web service composition.

Identification number of article 85TVN114

Web-сервисы - технология, позволяющая приложениям взаимодействовать друг с другом независимо от языка программирования, на котором они написаны, а также от платформы, на которой они развернуты. Web-сервис идентифицируется строкой URI и имеет интерфейсы, определенные на языке XML. Описание сервиса может быть найдено другими программными системами, которые могут взаимодействовать с ним посредством сообщений, основанных на XML, и передаваемых с помощью интернет протоколов[11].

Для поставщиков сервисов важным аспектом технологии Web-сервисов является качество обслуживания (Quality of Service, QoS). QoS является критически важным фактором при выборе Web-сервисов в качестве составляющих для сложных бизнес-процессов. Поэтому аспекты надежности, доступности и другие аспекты QoS фиксируются в стандартах, в частности, в стандарте Business Process Execution Language for Web-Services (BPEL4WS)[7].

Качество Web-сервисов (QoS) дает возможность клиентам и поставщикам услуг специфицировать запросы и предложения с заданными свойствами и показателями QoS, обеспечивает эффективное обнаружение и выборку требуемых Web-сервисов, позволяет поставщикам услуг публиковать и обновлять предложения с разными аспектами QoS, поддерживает эффективный вызов Web-сервиса с требуемым QoS[10].

В настоящее время выбор Web-сервисов, основанный на QoS, становится все более популярным. Большинство работ [8, 1, 6] касается статического случая, полагая, что каждый сервис имеет заранее определенные показатели QoS. Преимуществом такого подхода является упрощение выводов. Например, допустим, что сервис W состоит из последовательно вызываемых сервисов W_1 , W_2 и W_3 с временами отклика 2, 3 и 4 соответственно. Очевидно, что время отклика сервиса W равно 9. Такой подход имеет существенные ограничения в применимости. Время отклика Web-сервиса зависит от таких неопределенных причин как число поступивших запросов к сервису и текущая загруженность сервера, на котором запускается сервис. Очевидно, что доступность и надежность тоже связаны с неопределенностью. Таким образом, параметры QoS в сущности вероятностны.

Более сложным случаем является оптимальный выбор сервиса, как части составного. Можно применять нечеткую логику при выборе Web-сервиса, удовлетворяющего параметрам качества, для выполнения отдельного задания [8]. Для построения и анализа хода выполнения составного сервиса в [2] применяется метод критического пути и метод оценки и пересмотра планов, применяемые в управлении проектами для обнаружения узких мест в плане выполнения проекта. В работе [3] рассматривается вероятностный подход к оценке качества.

Показатели QoS для Web-сервисов можно разделить на пять категорий: быстродействие, ресурсоемкость, функциональная надежность, свойства транзакционности, безопасность. Эти качества реализуются на различных уровнях в приложении: уровне экземпляра класса сервиса, уровне класса сервиса и системном уровне.

При существующих подходах к моделированию качества Web-сервисов невозможно с достаточной точностью получить показатели качества для составного сервиса из QoS его составляющих. Точность получаемых показателей качества особенно важна при автоматическом выборе сервиса для составления еще более сложного приложения.

Целью работы является реализация модели оценки качества сервисов, позволяющей не только учесть особенности построения составных сервисов, но и связанную с Web-средой неопределенность в работе сервисов. Для этого показатели QoS как атомарных, так и композитных Web-сервисов считаются вероятностными. При алгоритмической реализации модели оценки качества возникают трудности с ростом области определения результирующей случайной величины. Решению проблемы сокращения этого роста с помощью группирующей случайной величины посвящена эта статья.

Рассмотрим вероятностную модель качества сервисов. Каждый показатель QoS Web-сервиса можно представить дискретной случайной величиной с конечной областью определения. Дискретные случайные величины для атомарных сервисов могут быть получены статистически и использоваться напрямую, без проверки каких-либо критериев согласия и приближения к непрерывным распределениям.

Рассмотрим время отклика простого сервиса, выполнение программы которого занимает постоянное время (рис.1). Для этого был разработан Web-сервис, складывающий 2 числа, которые приходят сервису как параметры. Сервис вызывался 1000 раз, чтобы получить гистограмму времени отклика (мс):

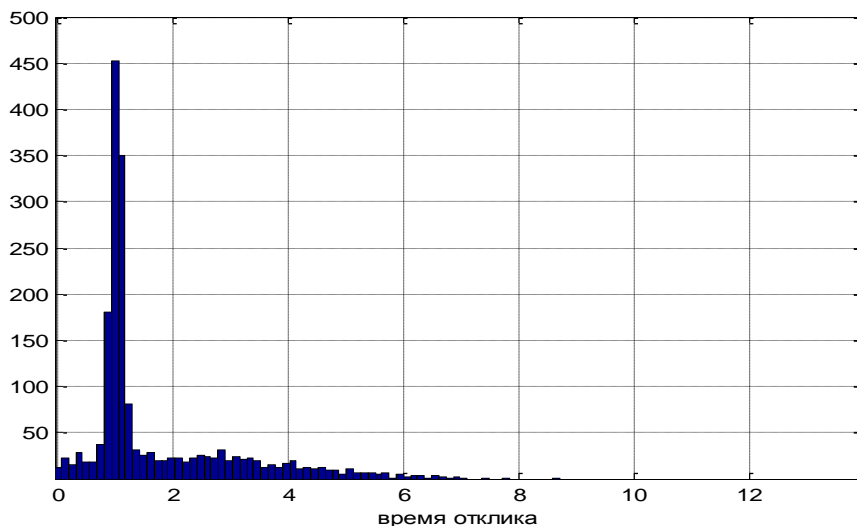


Рис. 1. Время отклика простого web-сервиса (разработано авторами)

Видно, что в большинстве случаев время отклика примерно 10^{-3} сек, но бывают отклонения. В данном случае вероятностная составляющая мала.

Для более сложных сервисов время отклика варьируется гораздо сильнее. Например, рассмотрим гистограмму (рис.2) времени отклика сервиса, показывающего табло аэропортов Аэрофлота.

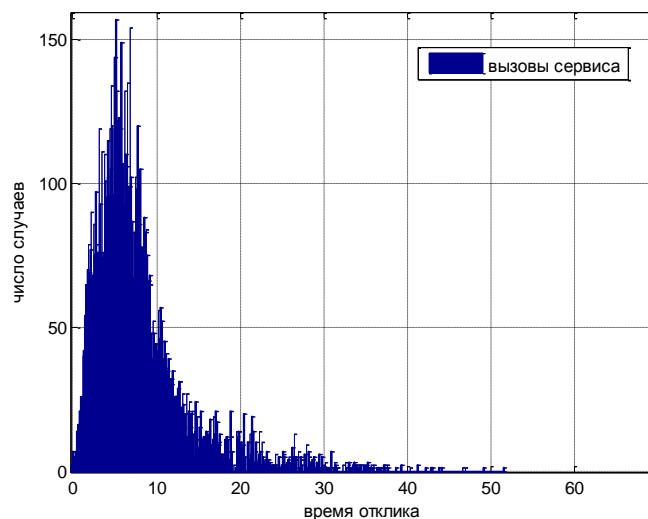


Рис. 2. Время отклика сервиса Аэрофлота (разработано авторами)

Аналогично другие показатели, например, доступность и надежность, тоже связаны с неопределенностью: загруженностью сети, работоспособностью систем, которые используют сервисы и т.д. Таким образом, каждый показатель QoS Web-сервиса можно представить дискретной случайной величиной с конечной областью определения. Дискретные случайные величины для атомарных сервисов могут быть получены статистически и использоваться напрямую, без проверки каких-либо критериев согласия и приближения к непрерывным распределениям.

В вероятностной среде QoS составного сервиса не всегда может напрямую получиться из QoS его составных частей. Например, рассмотрим два сервиса W_1 и W_2 , и поток W , в котором W_1 и W_2 запускаются параллельно.

Допустим, нужно получить среднее время отклика W . Будет неправильно считать средним временем выполнения W максимум из математических ожиданий времени выполнения W_1 и W_2 . Недостаточно учитывать только средние величины, как предполагается в [8]. Так же нереалистично предполагать, что параметры QoS имеют нормальное распределение. Как показано выше, случайная величина времени отклика может иметь более сложную структуру.

Естественно описывать надежность и стоимость Web-сервиса с помощью функций плотности распределения с областью определения, задаваемой множеством $\{0(\text{неудача}), 1(\text{успех})\}$ и множеством возможных стоимостей соответственно. Время отклика также можно смоделировать дискретной случайной величиной, если рассмотреть разбиение области определения ее функции плотности - интервала $(0; +\infty)$. Для трансформации времени отклика в дискретную величину может использоваться алгоритм разбиения на равные по ширине интервалы.

Можно рассчитать плотность распределения параметров QoS для составных Web-сервисов. Различные языки конструирования сервисов предлагают разные конструкции для управления ходом выполнения атомарных сервисов. В работе [7] приведено 5 базовых структурных конструкций, каждая из которых организует сервисы-компоненты уникальным образом.

Авторами предложена модель качества, которая учитывает эти конструкции [9]. В модели рассматривается только нормальное выполнение Web-сервисов и не учитываются исключительные ситуации, когда происходят ошибки. Ошибки и исключения в составных сервисах, а также вопросы транзакций подробно рассматриваются, например, в [5]. Рассмотрим предложенную модель качества.

Для каждого Web-сервиса считаем заданными 3 параметра QoS, а именно, время отклика, стоимость и надежность, которые обозначены $T(a)$, $C(a)$ и $R(a)$ соответственно.

Последовательность: w состоит из последовательности сервисов $(a_1, a_2, a_3, \dots, a_n)$.

$$C(W) = \sum_{i=1}^n C(a_i), T(W) = \sum_{i=1}^n T(a_i), R(W) = \sum_{i=1}^n R(a_i)$$

Параллельность: w состоит из сервисов $(a_1, a_2, a_3, \dots, a_n)$, выполняющихся одновременно с последующим объединением результатов их выполнения.

$$C(W) = \sum_{i=1}^n C(a_i), T(W) = \max_{1 \leq i \leq n} \{T(a_i)\}, R(W) = \prod_{i=1}^n R(a_i)$$

Исключающий выбор: w состоит из множества исключающих друг друга сервисов $(a_1, a_2, a_3, \dots, a_n)$, каждый из которых связан с вероятностью p_i , показывающей вероятность того, что выполнится a_i .

$$C(W) = \prod_{i=1}^n (C(a_i), p_i), \quad T(W) = \prod_{i=1}^n (T(a_i), p_i), \quad R(W) = \prod_{i=1}^n (R(a_i), p_i),$$

где $\prod_{i=1}^n (X_i, p_i)$ операция исключающего выбора одной величины из множества $\{X_i\}, i=1, \dots, n$ с вероятностями $p_i, i=1, \dots, n$, где p_i – вероятность того, что будет выбран величина X_i .

Дискриминатор: w состоит из параллельно выполняющихся сервисов $(a_1, a_2, a_3, \dots, a_n)$ без синхронизации.

$$C(W) = \sum_{i=1}^n C(a_i), \quad T(W) = \max_{1 \leq i \leq n} \{T(a_i)\}, \quad R(W) = 1 - \prod_{i=1}^n (1 - R(a_i)),$$

где $1 - R(a_i)$ обозначает обратную величину заданной $R(a_i)$.

В цикле сервис запускается несколько раз. Для цикла стоимость, время отклика и надежность считаются по формулам:

$$C(W) = \prod_{i=1}^n (C_a(l), f_{L(a)}(l)), \quad T(W) = \prod_{i=1}^n (T_a(l), f_{L(a)}(l)), \quad R(W) = \prod_{i=1}^n (R_a(l), f_{L(a)}(l)),$$

где $f_{L(a)}(l), 1 \leq l \leq c$ функция плотности числа итераций в цикле и

$$T_a(l) = \sum_{i=1}^l T(a_i), \quad C_a(l) = \sum_{i=1}^l C(a_i), \quad R_a(l) = \sum_{i=1}^l R(a_i).$$

При расчете полученной схемы растет область определения результирующей случайной величины. Например, при сложении k случайных величин, имеющих по n элементов в области определения, полученная случайная величина будет иметь в худшем случае порядка n^k элементов в области определения. Чтобы уменьшить это число после каждой операции до приемлемого размера можно группировать элементы. Если несколько последовательных элементов в области определения функции имеют одно и то же значение, они заменяются одним значением и рассчитывается групповая вероятность. Для этого вводится группирующая величина.

Пусть $Dom(X)$ область определения случайной величины X задана множеством $\{x_1, x_2, \dots, x_s\}$, где $x_i < x_{i+1}, 1 \leq i < s$ и функция плотности X обозначается как $f_X()$. Обозначим Y как группирующую случайную величину от X , если существует последовательность (j_1, j_2, \dots, j_m) элементов (x_1, x_2, \dots, x_s) , где $1 = j_1 < x_2 < \dots < j_m = j_s$ такая что

область определения $Y\{y_r : y_r = x_{j_r}, 1 \leq r \leq m\}$, и

плотность величины Y

$$f_Y(y_1) = f_X(x_1) = 0,$$

$$f_Y(y_r) = \sum_{k=j_{r-1}+1}^{j_r} f_X(x_k), 2 \leq r \leq m$$

Для измерения разницы между группирующей и изначальной величиной используется погрешность группировки. Введем величину погрешности группировки Y относительно X как среднеквадратичное отклонение:

$$group_error(Y, X) = \sum_{r=2}^m \sum_{k=j_{r-1}+1}^{j_r} \frac{f_X(x_k)}{f_Y(y_r)} \cdot (x_k - y_r)^2$$

Рассмотрим пример:

$$X : Dom(X) = \{0,1,2,3,4,5\}$$

$$Y : Dom(Y) = \{0,3,5,7\}$$

$$f_Y(y=0) = 0,$$

$$f_Y(y=3) = 0.3,$$

$$f_Y(y=5) = 0.3,$$

$$f_Y(y=7) = 0.4$$

$$f_X(x_i) = \begin{cases} 0, & i=1, \\ 0.1 & i=2,3,4,5, \\ 0.2 & i=6,7,8. \end{cases}$$

$$group_error(Y, X) = 2,5$$

Погрешность группировки позволяет оценить насколько полученная группирующая случайная величина отличается от исходной. Поэтому эта ошибка должна быть минимизирована, т.е. встает задача поиска оптимальной группирующей случайной величины, описываемая следующим образом.

Дана случайная величина X с s элементами в области определения и требуемое число элементов в области определения m . Задача заключается в нахождении случайной величины Y , в области определения которой m элементов, так чтобы минимизировалась погрешность группировки относительно X .

Для решения этой задачи рассмотрим два алгоритма: рекурсивный и «жадный».

Для построения рекурсии введем функцию $e(i, j, k)$ для обозначения оптимальной погрешности группировки элементов последовательности $(x_i, x_{i+1}, \dots, x_j)$ в k частей. Тогда:

$$e(i, j, k) = \begin{cases} \min_{i \leq a \leq j, 1 \leq b < k} \{e(i, a, b), e(a+1, j, k-b)\}, & \text{если } j-i+1 > k \text{ и } k > 1, \\ 0, & \text{если } j-i+1 = k, \\ error(i, j), & \text{если } k = 1 \end{cases}$$

Здесь $error(i, j)$ – погрешность группировки последовательности $(x_i, x_{i+1}, \dots, x_j)$ в один элемент x_j ,

$$error(i, j) = \sum_{i \leq k \leq j} \frac{f_X(x_k)}{\sum_{i \leq k' \leq j} f_X(x_{k'})} \cdot (x_k - x_j)^2$$

Временная сложность алгоритма $O(s^3 m^2)$. В случае если складываются две дискретные случайные величины с m элементами в области определения каждой из них (и эти множества не пересекаются) и требуется сократить область определения результирующей случайной величины снова до m , временная сложность алгоритма будет $O(m^8)$.

Можно подойти к задаче поиска оптимальной группирующей величины по-другому. Введем величину $pair_error(x_i, x_{i+1})$ – погрешность парной группировки соседних элементов x_i и x_{i+1} в один x_{i+1} .

$$pair_error(x_i, x_{i+1}) = \frac{f_X(x_i)}{f_X(x_i) + f_X(x_{i+1})} (x_i - x_{i+1})^2$$

Тогда можно выбрать элементы для группировки с минимальной погрешностью парной группировки, что используется в «жадном» алгоритме.

В цикле выполняется следующая последовательность:

1. Ищется пара соседних элементов x_i и x_{i+1} с наименьшей погрешностью парной группировки.
2. x_i и x_{i+1} заменяются на $x' = x_{i+1}$, пересчитывается $f_X(x') = f_X(x_i) + f_X(x_{i+1})$
3. пересчитываются погрешности парной группировки для полученной величины.

В каждой итерации шаг 1 и 3 требуют $O(\log(s))$ времени, шаг 2 выполняется за постоянное время. Общее число итераций равно $s - m$. Таким образом, временная сложность алгоритма $O(s \cdot \log(s))$.

Для расчета значений QoS n Web-сервисов, можно делать парные операции $n-1$ раз и применять каждый раз сокращение области определения описанными выше методами. Допустим, каждая случайная величина имеет область определения с m элементами. Сложение двух таких величин дает в результате область определения в m^2 элементов. Перед следующим сложением нужно произвести сокращение до m .

Проведем сравнение этих двух алгоритмов. Для сравнения возьмём простой случай. Для случайных величин X и Y , распределенных по нормальному закону с параметрами, приведенными в таблице, считаются теоретически параметры распределения величины $T=X+Y$.

Таблица 1

Параметры распределений

Величина	Математическое ожидание	Дисперсия
X	100	10
Y	90	20
T=X+Y	190	$10\sqrt{5}$

Для этих же величин X, Y считается сумма с применением алгоритмов сокращения области определения. Обозначим полученную с применением жадного алгоритма сокращения области определения – T_g , а полученную с помощью рекурсивного алгоритма случайную

величину – T_r . Для них посчитаем интегральные функции распределения, которые сравниваются с теоретическим законом распределения для T . График отклонений T_g и T_r приведен на рис. 3.

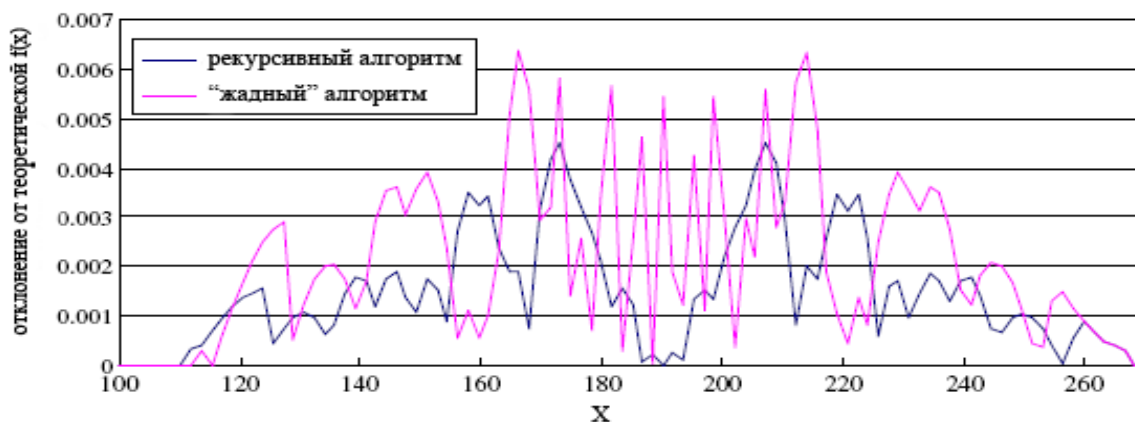


Рис. 3. Отклонение от теоретического распределения (разработано авторами)

По графику сложно сказать какой алгоритм лучше, но среднеквадратичное отклонение от теоретической функции для рекурсивного алгоритма равно 0.0014, тогда как для «жадного» оно составляет 0.0021.

Была проведена серия тестов для получения зависимости точности алгоритмов от желаемого числа элементов в области определения результирующей величины. Результаты можно видеть на рис. 4.

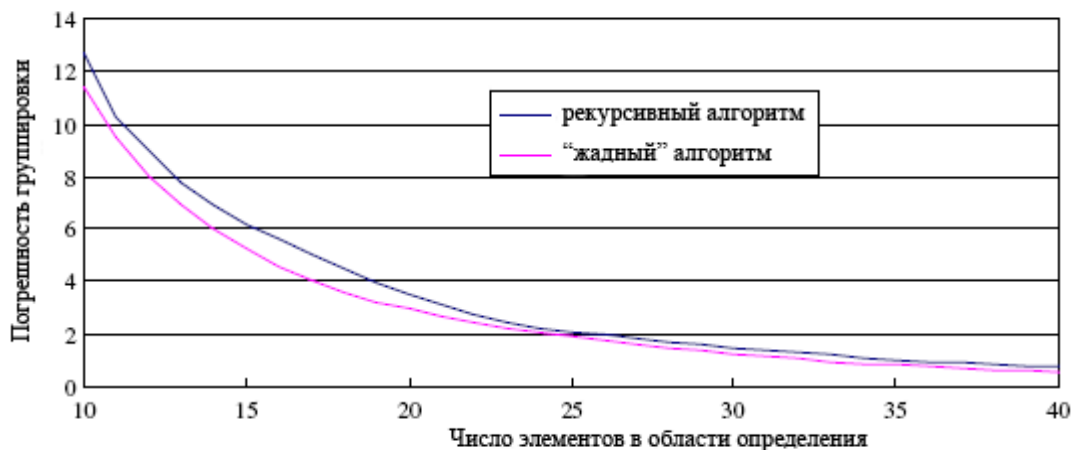


Рис. 4. Точность алгоритмов при различном числе элементов в области определения функции (разработано авторами)

По графику видно, что как и в предыдущем случае, жадный алгоритм дает всегда большее отклонение от теоретического закона распределения. С ростом числа элементов в области определения оба алгоритма становятся точнее.

Как и предполагалось в теоретическом расчете временной сложности алгоритмов, время работы намного быстрее возрастает для рекурсивного алгоритма, чем для «жадного» при увеличении размерности задачи. При этом разница в точности не велика, поэтому далее

для тестирования модели качества будет использоваться «жадный» алгоритм как более гибкий.

Проведем проверку модели. Будем использовать только время отклика T , т.к. его можно легко измерить. Для сравнения с рассчитанным по модели временем используется реально измеренное время отклика составного сервиса.

Представим типичную модель обработки заказа для он-лайн магазина. На самом высоком уровне в составном сервисе находится конструкция последовательности, которая состоит из сервисов OrderCreation, StockPaymentCheck и ShippingArrangement. Сервис StockPaymentCheck представляет собой параллельное выполнение сервисов PaymentCheck и StockCheck. PaymentCheck в свою очередь является конструкцией выбора, состоящей из сервисов PaypalValidation и CreditCardValidation, выбор между которыми определяется пользователем через запрос. StockCheck – это цикл вызовов сервиса GetItem, который выполняется для каждой записи в заказе. Если требуемое есть на складе, доставка заказывается оттуда (GetFromWarehouse), иначе отправляется заказ на доставку поставщику (GetFfromVendor).

Каждый элементарный сервис в составе композитного сервиса не делает ничего, кроме случайной задержки, выбираемой из массива, сгенерированного для каждого сервиса и загруженного в память заранее до начала эксперимента. Массивы формируются по определенным законам распределения, параметры которых представлены в таблице 2.

Таблица 2

Параметры распределения для случайных величин задержки

Название сервиса	Математическое ожидание	Дисперсия	Минимум	Максимум
OrderCreation	2	1	1	4
ShippingArrangement	3	1	1	5
PaypalValidation	2	0,8	1	4
CreditCardValidation	2	1,5	1	4
GetFromWarehouse	2	0,7	1	4
GetFfromVendor	5	2	1	8

Плотность распределения числа вызовов цикла с сервисом *GetItem* задана в таблице 3.

Таблица 3

Плотность распределения числа вызовов цикла с сервисом GetItem

Число вызовов сервиса GetItem	вероятность
1	0.1
2	0.2
3	0.4
4	0.2
5	0.1

Для реального web- сервиса эти параметры получаются статистически.

Гистограмма времени отклика (Рис. 5) строится на основе нескольких серий вызовов составного сервиса. Потом это же время считается по приведенным формулам с применением «жадного» алгоритма сокращения области определения (т.к. было показано, что этот алгоритм работает оптимально).

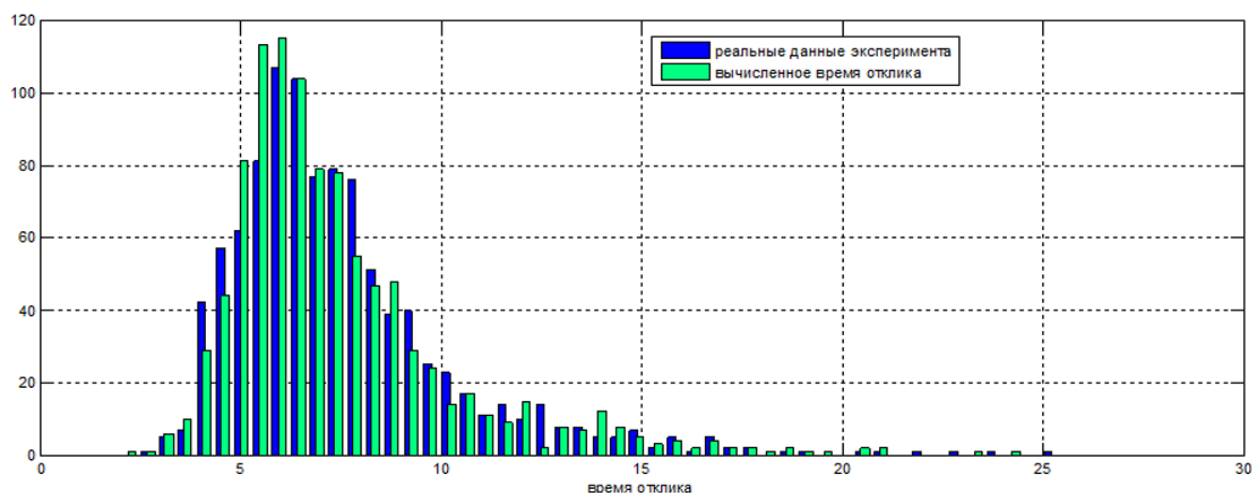


Рис. 5. Гистограмма времени отклика - реальные данные и вычисленное время (разработано авторами)

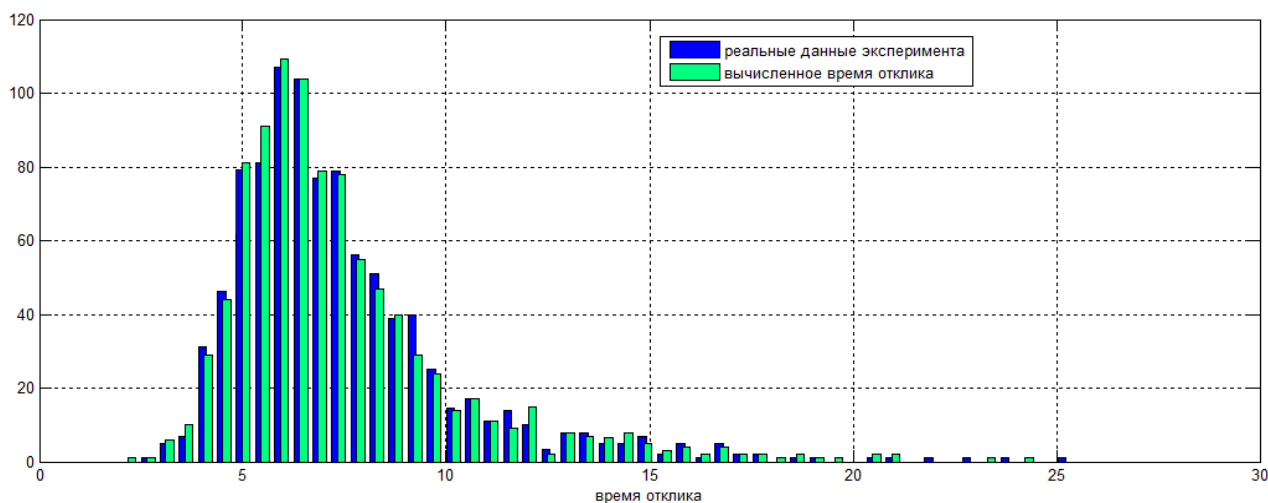


Рис. 6. Время отклика без сокращения области определения (разработано авторами)

При расчете без сокращения области определения (Рис. 6) на каждом шаге расчета время работы алгоритма существенно возрастает, а изменение отклонения от практически измеренного результата не заметно.

Далее на графике (Рис. 7) показано отклонение практически полученной функции плотности для времени отклика и посчитанной по схеме предлагаемой модели без применения алгоритмов сокращения области определения.

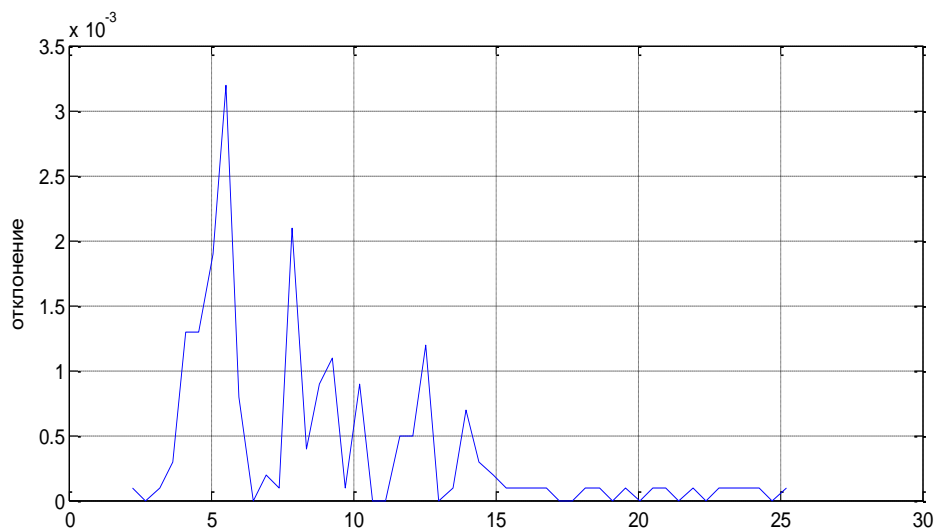


Рис. 7. Разница в точности оценки времени отклика реального сервиса и вычисленной моделью без применения алгоритмов сокращения области определения (разработано авторами)

Очевидно, большее время расчета и возросшие требования к ресурсам не соответствуют увеличению точности расчета. После проведенного эксперимента можно сказать, что расчет времени отклика web-сервиса по приведенным выше формулам с применением «жадного» алгоритма сокращения области определения дает приемлемую точность и может быть использован в модели оценки качества web-сервисов.

Задача поиска оптимальной группирующей случайной величины свелась к рассмотрению двух алгоритмов: рекурсивного и «жадного». Как и предполагалось, при увеличении размерности задачи время работы намного быстрее возрастает для рекурсивного алгоритма, чем для «жадного». При этом разница в точности не велика, поэтому далее для тестирования модели качества использовался «жадный» алгоритм как более гибкий.

Таким образом, при алгоритмической реализации модели оценки качества, возникающие трудности с ростом области определения результирующей случайной величины авторы предлагают решать с помощью группирующей случайной величины, поиск которой производить с использованием «жадного» алгоритма. Эффективность данного решения подтверждают эксперименты по вычислению времени отклика реального web-сервиса по обработке заказа для интернет-магазина. Результат эксперимента позволяет сделать вывод о том, что оценка качества композитного Web-сервиса, полученная с помощью предложенной методики, гарантирует, что сервис работает на рассчитанном уровне качества.

ЛИТЕРАТУРА

1. Cardoso J., Sheth A., Miller J., Arnold J., Kochut K. Quality of service for workflows and web service processes // Journal of Web Semantics, 2004. Vol. 1. P. 281–308.
2. El Haddad, J., M. Manouvrier, and M. Rukoz, TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition. IEEE Transactions on Services Computing, 2010. 3(1): p. 73-85
3. Gillmann M. et al. Workflow management with service quality guarantees //Information Sciences, 2002. Vol.1 P. 228-239
4. Hwang S.-Y. et al. A probabilistic approach to modeling and estimating the QoS of web-services-based workflows // Information Sciences, 2007. Vol. 177. P. 5484–5503
5. Hwang S.-Y., Tang J. Consulting past exceptions to facilitate workflow exception handling //Decision Support Systems (DSS), 2004 Vol. 37(1). P. 49–69.
6. Jaeger M.C., Rojec-Goldmann G., Muehl G. QoS aggregation for Web service composition using workflow patterns // Proceedings of IEEE 8th International Conference on Enterprise Distributed Object Computing (EDOC04), 2004. P. 149–159.
7. Keen M., Cavell J., Hill S., Kee C., Neave W., Rumph B., Tra H. BPEL4WS Business Processes with WebSphere Business Integration:Understanding, Modeling, Migrating.– International Business Machines Corporation. 2004.–363p.
8. Wang H.-C., Lee C.-S., Ho T.-H. Combining subjective and objective QoS factors for personalized Web service selection // Expert Systems with Applications, 2007. Vol. 2. P. 571–584.
9. Волушкова В.Л. Школенко Е.Ю. Модель оценки качества Web –сервисов// Вестник Московского университета имени С.Ю.Витте. Материалы ХLI международной конференции «Информационные технологии в науке, образовании, телекоммуникации и бизнесе IT+SE`2013», 2013.–с.213-216.
10. Кузнецов С. Как поймать в Сети злоумышленника? //Открытые системы, 2,2007, стр. 102-109.
11. Ньюкомер Э. Веб-сервисы. – СПб.:Питер, 2003.–256с.

Рецензент: Архипов Сергей Викторович, кандидат физико-математических наук, доцент, доцент кафедры математической статистики и системного анализа ФГБОУ «Тверской государственный университет».

REFERENCES

1. Cardoso J., Sheth A., Miller J., Arnold J., Kochut K. Quality of service for workflows and web service processes // *Journal of Web Semantics*, 2004. Vol. 1. P. 281–308.
2. El Haddad, J., M. Manouvrier, and M. Rukoz, TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition. *IEEE Transactions on Services Computing*, 2010. 3(1): p. 73-85
3. Gillmann M. et al. Workflow management with service quality guarantees // *Information Sciences*, 2002. Vol.1 P. 228-239
4. Hwang S.-Y. et al. A probabilistic approach to modeling and estimating the QoS of web-services-based workflows // *Information Sciences*, 2007. Vol. 177. P. 5484–5503
5. Hwang S.-Y., Tang J. Consulting past exceptions to facilitate workflow exception handling // *Decision Support Systems (DSS)*, 2004 Vol. 37(1). P. 49–69.
6. Jaeger M.C., Rojec-Goldmann G., Muehl G. QoS aggregation for Web service composition using workflow patterns // *Proceedings of IEEE 8th International Conference on Enterprise Distributed Object Computing (EDOC04)*, 2004. P. 149–159.
7. Keen M., Cavell J., Hill S., Kee C., Neave W., Rumph B., Tra H. BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating.– International Business Machines Corporation. 2004.–363p.
8. Wang H.-C., Lee C.-S., Ho T.-H. Combining subjective and objective QoS factors for personalized Web service selection // *Expert Systems with Applications*, 2007. Vol. 2. P. 571–584.
9. Volushkova V.L. Shkolenko E.Ju. Model' ocenki kachestva Web –servisov// *Vestnik Moskovskogo universiteta imeni S.Ju.Vitte. Materialy XLI mezhdunarodnoj konferencii «Informacionnye tehnologii v nauke, obrazovanii, telekommunikacii i biznese IT+SE`2013»*, 2013.–s.213-216.
10. Kuznecov S. Kak pojmat' v Seti zloumyshlennika? // *Otkrytye sistemy*, 2,2007, str. 102-109.
11. N'jukomer Je. *Veb-servisy*. – SPb.:Piter, 2003.–256s.