

9. Arning A., Agrawal R., Raghavan Pr. A Linear Method for Deviation Detection in Large Databases // Proceedings of KDD'1996. 1996, pp. 164-169.
10. Антипов С.Г., Фомина М.В. Проблема обнаружения аномалий в наборах временных рядов // Программные продукты и системы. 2012. № 2. с.78-82.
11. Marina Fomina, Sergey Antipov, Vadim Vagin Methods and algorithms of anomaly searching in collections of time series // Proceedings of the first International Scientific Conference Intelligent Information Technologies for Industry (IITI'16), Vol.1, pp.63-73 // Series Advances in Intelligent Systems and Computing. –Springer Verlag, Vol. 450. 2016.
12. Naoki Saito. Local feature extraction and its application using a library of bases. // PhD thesis. Yale University, December. 1994.
13. Pham D.T., Chan A.B. Control Chart Pattern Recognition using a New Type of Self Organizing Neural Network // Proc. Of Institution of Mechanical Engineering. Vol. 212. No. 1. 1998. Pp. 115-127.
14. Bruno B., Mastrogiovanni F., Sgorbissa A., Vernazza T., Zaccaria R. Analysis of human behavior recognition algorithms based on acceleration data // IEEE Int. Conf. on Robotics and Automation (ICRA). 2013. pp. 1602-1607.
15. UCI Repository of Machine Learning Datasets. // <http://archive.ics.uci.edu/ml/>.
16. Chen Yanping, Keogh Eamonn, Hu Bing et al. The UCR Time Series Classification Archive–2015. July. // www.cs.ucr.edu/~eamonn/time_series_data
17. Olszewski R. Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data: Ph.D thesis // School of Computer Science. – Carnegie Mellon University, Pittsburgh, 2001. 125 p.

Methods of diagnostics of dynamic objects based on the analysis of time series

Sergej Genadjevich Antipov,

Vadim Nikolaevich Vagin, Dr. of Tech. Sciences, prof., Department. PM, National Research University "MEI"

Marina Vladimirovna Fomina, Cand. Tech. Sciences, Assoc. Cafe. BT, National Research University "MEI"

The paper deals with the problem of detecting anomalies in sets of time series. The statement of the problem is given. An algorithm is proposed for the case when the training set contains sets of time series of several classes. The results of software simulation for real problems are given. Comparison with the results shown by a number of well-known algorithms is given.

Keywords: time series, inductive concept formation, search for exceptions, classification

УДК 681.3.068

ОЦЕНКА КАЧЕСТВА WEB-СЕРВИСОВ

Вера Львовна Волушкова, канд. техн. наук, доцент,
e-mail: w2lvera@gmail.com,
Тверской государственный университет,
<http://university.tversu.ru>

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.

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

Для поставщиков сервисов важным аспектом технологии Web-сервисов является качество обслуживания (Quality of Service, QoS). Предложенная модель оценки качества сервисов, позволяет 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. Решение проблемы сокращения этого роста с помощью группирующей случайной величины рассматривается далее.

Показатели QoS для Web-сервисов можно разделить на пять категорий: быстродействие, ресурсоемкость, функциональная надежность, свойства транзакционности, безопасность.

Различные языки конструирования сервисов предлагают разные конструкции для управления ходом выполнения атомарных сервисов. В работе [1] приведено 5 базовых структурных конструкций, каждая из которых организует сервисы-компоненты уникальным образом. Будем рассматривать модель качества, которая учитывает эти конструкции и приведена в [2]. 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 [3].

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 .

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

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.

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

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

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

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

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

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

При расчёте полученной схемы растёт 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.

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 введём 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 ,

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

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. Введём 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 расчёта 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 даёт в результате область определения в m^2 элементов. Перед следующим сложением нужно произвести сокращение до m .

В работе [4] приведены результаты сравнения двух алгоритмов. Отклонение от теоретической функции для рекурсивного алгоритма меньше, чем у «жадного».

Была проведена также серия тестов для получения зависимости точности алгоритмов от желаемого числа элементов в области определения результирующей величины. По результатам тестирования жадный алгоритм даёт всегда большее отклонение от теоретического закона распределения. С ростом числа элементов в области определения оба алгоритма становятся точнее.

Как и предполагалось в теоретическом расчёте 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.

Проверка модели проводилась на примере обработки заказа для он-лайн магазина. В качестве параметра QoS использовалось только время отклика T , т.к. его можно легко измерить. Для сравнения с рассчитанным по модели временем используется реально измеренное время отклика составного сервиса.

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 (GetFromVendor).

Гистограмма времени отклика (Рис.1) 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). Получено следующее:

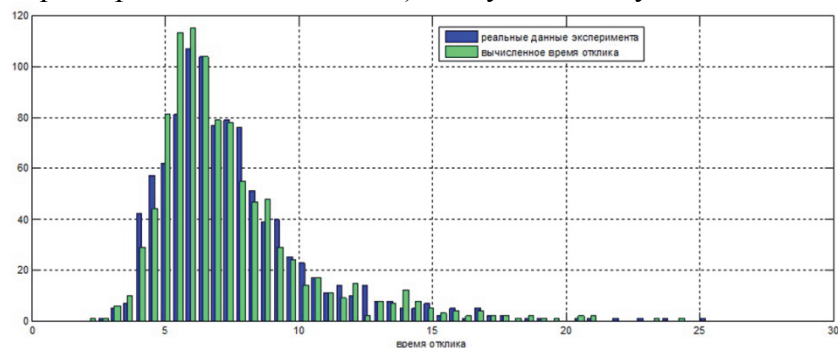


Рис. 1. Гистограмма времени отклика - реальные данные и вычисленное время

При расчёте без сокращения области определения (Рис.2) на каждом шаге расчета время работы алгоритма существенно возрастает, а изменение отклонения от практически измеренного результата не заметно.

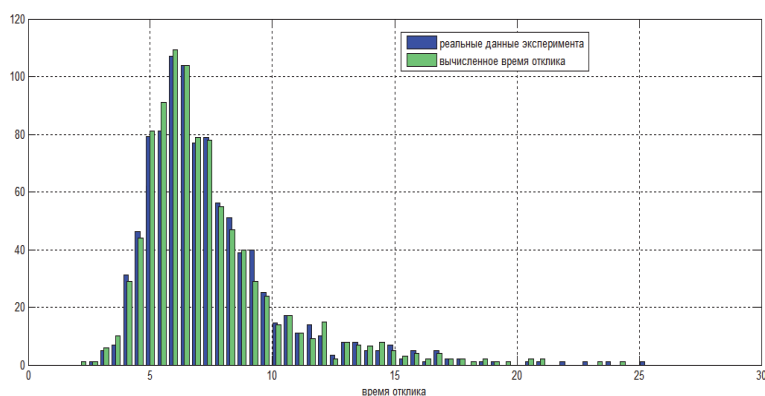


Рис. 2. Время отклика без сокращения области определения

Далее на графике (Рис.3) показано отклонение практически полученной функции плотности для времени отклика и посчитанной по схеме предлагаемой модели без применения алгоритмов сокращения области определения.

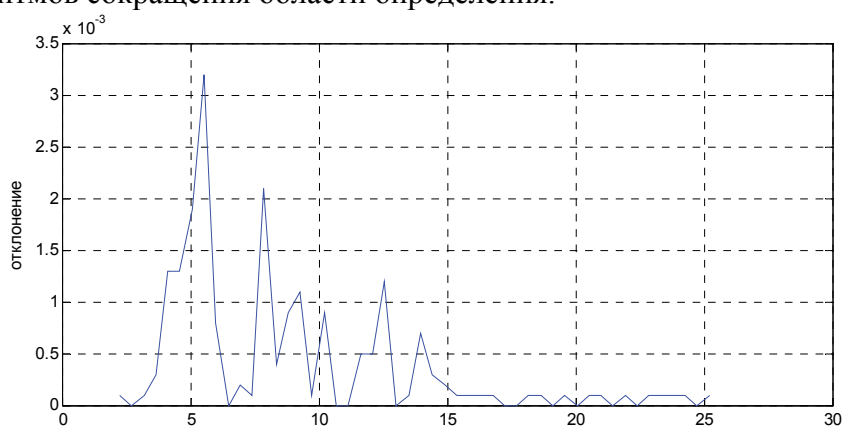


Рис. 3. Разница в точности оценки времени отклика реального сервиса и вычисленного моделью без применения алгоритмов сокращения области определения

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

Таким образом, 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.

Были произведены 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.

Автор считает, что в данной работе новыми являются следующие положения и результаты:

1. Модель качества, которая основана на вероятностном подходе и рассматривает 5 базовых структурных конструкций, каждая из которых организует сервисы-компоненты уникальным образом.

2. Решение проблемы моделирования (рост области определения результирующей случайной величины) с помощью группирующей случайной величины.

3. Программный продукт, который позволяет рассчитать качество композитного Web-сервиса.

Литература:

1. 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. 363 p.

2. Волюшкова В.Л., Школенко Е.Ю. Модель оценки качества Web –сервисов // Информационные технологии в науке, образовании, телекоммуникации и бизнесе: Материалы международной конференции. IT+SE`2013, 2013. С. 213-216.

3. Hwang S.-Y., Tang J. Consulting past exceptions to facilitate workflow exception handling // Decision Support Systems (DSS). 2004. Vol. 37(1). P. 49–69.

4. Волюшкова В.Л., Школенко Е.Ю. Алгоритмическая реализация модели качества web-сервисов // Наукоедение. 2014. № 1 (20). С. 60-65.

Web services' quality

Vera L'vovna Volushkova, Candidate of Technical Sciences, Associate Professor, Tver State University

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.

Key words: web-service; composite web-service; quality of web-service; quality of service attribute; response time; reliability; optimization; recursive algorithm; "greedy" algorithm; Web service composition.

УДК 004.89:[004.05:351]

**ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ В РАЗРАБОТКЕ
АРХИТЕКТУРЫ ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА**

*Роман Валерьевич Ерженин, канд. экон. наук, генеральный директор,
e-mail: rerzhenin@gmail.com,
Ооо «научно-практический центр госучет»,
<http://www.guchet.ru>*

Использование интеллектуальных технологий для поддержки разработки методологий архитектурных построений может повысить эффективность использования и проектирования информационных систем управления в секторе государственного управления и обеспечить их рациональную стратегию развития.

Ключевые слова: интеллектуальные технологии, архитектура электронного правительства, архитектура информационных систем.

Введение

В современной России начало становления электронного правительства (э-правительства) связывают с выполнением Федеральной целевой программы (ФЦП) «Электронная Россия 2002—2010 гг.» В 2004 году, в рамках ее реализации компания