

doi: 10.17586/2226-1494-2023-23-3-547-552

УДК 004.75

## Метод линейной аппроксимации временных параметров в эластичных системах

Илья Геннадьевич Мартынчук<sup>1</sup>✉, Сергей Александрович Жмылёв<sup>2</sup><sup>1</sup> Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация<sup>2</sup> ООО «ГК Ядро», Москва, 125373, Российская Федерация<sup>1</sup> [mt4.ilja@gmail.com](mailto:mt4.ilja@gmail.com)✉, <https://orcid.org/0000-0003-2409-9677><sup>2</sup> [s.zhmylev@yadro.com](mailto:s.zhmylev@yadro.com), <https://orcid.org/0000-0001-8916-5474>

### Аннотация

**Введение.** В современных эластичных системах важной задачей является прогнозирование изменений в нагрузочной процессах. Оценка скорости изменения нагрузки помогает заблаговременно адаптировать структуру системы для сохранения качества обслуживания пользователей. В современных решениях мало внимания уделено анализу скорости изменения нагрузки, которая напрямую влияет на то, насколько заранее необходимо включать или выключать узлы из вычислительного процесса. В большинстве случаев данные интервалы запуска задаются заранее установленными эмпирическими значениями. Для определения скорости изменения нагрузочного процесса достаточно решить задачу линейной аппроксимации на промежутке возрастания или убывания функции нагрузки от времени. Существующие методы линейной аппроксимации не удовлетворяют всем требованиям, предъявляемым к окружениям эластичных систем. В связи с этим необходима разработка собственного метода аппроксимации. **Метод.** Упрощенный метод линейной аппроксимации основан на расчете центра масс исходного набора данных, а также среднего относительного отклонения упорядоченных точек по оси ординат друг от друга. Новизна предложенного метода заключается в одновременном константном потреблении памяти и отсутствии операций с квадратичными зависимостями. Это позволит учесть все требования, предъявляемые к функционирующим в окружениях эластичных систем методам. **Основные результаты.** Разработан генератор точек в двумерной плоскости для получения набора упорядоченных точек, разбросанных относительно заданной прямой. Генератор позволяет оценить точность предлагаемого метода аппроксимации относительно других методов путем расчета среднего результирующего отклонения сгенерированных точек от заданной прямой. С доверительной вероятностью 0,95 показано, что при максимальном количестве точек в исходном наборе данных, равном 10 000, снижение времени выполнения аппроксимации благодаря разработанному методу достигает 23 %. Значение среднего отклонения для разработанного метода и метода наименьших квадратов, полученное в рамках экспериментов, одинаково. **Обсуждение.** Полученные результаты могут быть применены в сервисах автоматического масштабирования эластичных систем с целью снижения времени выполнения прогнозов скорости изменения нагрузочных процессов. Разработанный метод, в отличие от метода наименьших квадратов, лишен недостатка, связанного с операциями с квадратичными зависимостями, что обеспечивает возможность более широкого его применения в условиях ограниченности разрядной сетки некоторых архитектур.

### Ключевые слова

линейная аппроксимация, эластичные системы, автоматическое масштабирование, метод наименьших квадратов, метод достижения консенсуса на основе случайных выборок, метод главных компонент

**Ссылка для цитирования:** Мартынчук И.Г., Жмылёв С.А. Метод линейной аппроксимации временных параметров в эластичных системах // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 3. С. 547–552. doi: 10.17586/2226-1494-2023-23-3-547-552

## Time parameters linear approximation method in elastic systems

Ilya G. Martynchuk<sup>1</sup>, Sergei A. Zhmylev<sup>2</sup><sup>1</sup> ITMO University, Saint Petersburg, 197101, Russian Federation<sup>2</sup> GK YADRO LLC, Moscow, 125373, Russian Federation<sup>1</sup> mt4.ilja@gmail.com, <https://orcid.org/0000-0003-2409-9677><sup>2</sup> s.zhmylev@yadro.com, <https://orcid.org/0000-0001-8916-5474>**Abstract**

In modern elastic systems, an important task is to predict changes in load processes. Estimating the load change rate helps to adapt the system structure in advance to maintain the quality of user experience. In modern solutions, little attention is paid to the analysis of the load change rate which directly affects how far in advance it is necessary to turn nodes on or off from the computing process. In most cases, these trigger intervals are set to pre-set static values. In order to determine the load process change rate, it is sufficient to solve the linear approximation problem over the interval of increase or decrease in the load function over time. The existing methods of linear approximation do not satisfy all the requirements for the elastic systems environments, which necessitates the development of own approximation method. A simplified linear approximation method ZFLAM is based on the calculation of the center of the initial data set mass as well as the average relative deviation of the ordered points along the ordinate axis from each other. The novelty of the proposed method lies in the simultaneous constant consumption of memory and the absence of operations with quadratic dependencies, which makes it possible to satisfy all the requirements for methods operating in elastic system environments. A two-dimensional plane point generator has been developed which makes it possible to obtain a set of ordered points scattered relative to a given line. The developed generator makes it possible to evaluate the accuracy of the proposed approximation method relative to other methods by calculating the average resulting deviation of the generated points from a given straight line. It was revealed that with a confidence probability of 0.95, with the maximum number of points in the original data set equal to 10,000, the reduction in the approximation execution time due to the developed method reaches 23 %. It was determined that with a confidence probability of 0.95, the value of the mean deviation for both methods in the framework of the experiments is the same. The obtained results can be applied in the elastic systems automatic scaling services in order to reduce the execution time of load processes change rate forecasts. The developed method, in contrast to the least squares method, is free from the disadvantage associated with operations with quadratic dependencies, which makes it possible to use it more widely in the conditions of limited bit grid of some architectures.

**Keywords**

linear approximation, elastic systems, automatic scaling, least squares, random sample consensus, principal component analysis

**For citation:** Martynchuk I.G., Zhmylev S.A. Time parameters linear approximation method in elastic systems. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 3, pp. 547–552 (in Russian). doi: 10.17586/2226-1494-2023-23-3-547-552

**Введение**

Современные вычислительные системы работают в условиях изменяющихся во времени нагрузочных процессов [1]. Причины данных изменений имеют различную природу: суточная активность пользователей, особенности автоматизируемых системой бизнес-процессов, порядок функционирования смежных систем и т. д. Для сохранения приемлемого уровня качества предоставления услуг вычислительные системы должны обладать возможностью изменения своей структуры в зависимости от характера обрабатываемой нагрузки. Данная возможность достигается разработкой и внедрением в систему процесса автоматического масштабирования, который анализирует протекающие в системе нагрузочные процессы. А также путем добавления или исключения вычислительных узлов при адаптации структуры системы для обработки необходимого количества запросов. Вычислительные системы с реализованным процессом автоматического масштабирования называются эластичными системами [2].

В современных эластичных системах процессы автоматического масштабирования включают в себя механизмы прогнозирования нагрузочных процессов [3]. Данные механизмы позволяют проактивно реагировать на изменения в потоках пользовательских запросов, основываясь на данных о характере нагрузочных про-

цессов в прошлом. Например, заранее запуская необходимое количество вычислительных узлов при росте нагрузки или вывода из вычислительного процесса узлы при ее снижении [4].

Однако в современных решениях мало внимания уделено анализу скорости изменения нагрузки, которая напрямую влияет на то, насколько заранее необходимо включать или исключать узлы из вычислительного процесса. В открытых источниках не удалось найти решений, позволяющих оценивать скорость изменения нагрузочного процесса для обеспечения возможности своевременного изменения структуры эластичной системы. В большинстве случаев данные интервалы запуска задаются заранее установленными эмпирическими значениями [5].

**Постановка задачи**

Пусть имеется некоторая спрогнозированная зависимость интенсивности входящих в систему запросов  $\lambda$  от времени (рис. 1). Рассмотрим промежуток возрастания функции  $\lambda(t)$ , начиная с момента времени  $t_1$  и заканчивая  $t_5$ . Для определения скорости изменения нагрузочного процесса достаточно решить задачу линейной аппроксимации на промежутке возрастания функции  $\lambda(t)$ . Полученная аппроксимирующая прямая образует с осью  $x$  некоторый угол  $\alpha$ , который и опре-

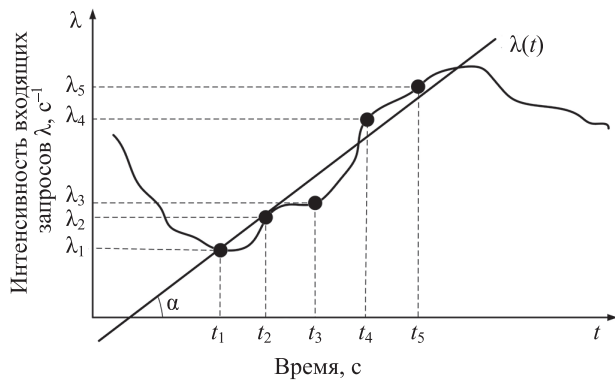


Рис. 1. Задача аппроксимации при прогнозировании скорости изменения нагрузочного процесса

Fig. 1. Approximation problem in predicting the load process change rate

деляет величину скорости возрастания интенсивности запросов к системе на рассматриваемом промежутке времени.

Для решения задачи аппроксимации выберем метод, наиболее подходящий для разнообразных окружений эластичных систем. С учетом особенностей функционирования таких систем сформулируем ряд требований к рассматриваемым методам аппроксимации.

Метод должен работать с упорядоченным по времени набором данных. Временные параметры в эластичных системах поступают к узлам управления от систем мониторинга и телеметрии, которые в большинстве случаев отправляют данные через равные промежутки времени [6].

Метод должен обладать линейной алгоритмической сложностью, а также линейной зависимостью объема потребляемой памяти от количества входных данных. В окружениях эластичных систем часто функционирует большое количество различных приложений [6]. Так как задача прогнозирования скорости изменения нагрузочных процессов возложена на управляющий узел системы, то при выполнении данной задачи необходимо сохранить требования к отказоустойчивости данного узла. Повышенная алгоритмическая сложность и потребление памяти в процессах прогнозирования может негативно сказаться на стабильности функционирования системы в целом [7–9].

Метод должен работать в условиях ограничений, накладываемых некоторыми архитектурами. Например, эластичные системы, работающие в окружениях интернета вещей и встраиваемых систем, имеют в своем распоряжении меньшую разрядную сетку представления типов данных, чем системы, работающие в окружениях облачных систем общего назначения.

В результате анализа предметной области были выбраны наиболее популярные методы, решающие задачи линейной аппроксимации упорядоченного набора данных [10]: метод наименьших квадратов (МНК), метод достижения консенсуса на основе случайных выборок (Random Sample Consensus, RANSAC), метод главных компонент (Principal Component Analysis, PCA).

**Метод наименьших квадратов.** Наиболее популярный метод решения задач линейной аппроксимации

[11]. Суть метода заключается в нахождении коэффициентов прямой, при которых сумма квадратов отклонений аппроксимируемых значений от соответствующих значений этой прямой является минимальной. Решим данную задачу, определив минимальные значения следующей функции:

$$F(k, b) = \sum_{i=1}^n (y_i - (kx_i + b))^2, \quad (1)$$

где  $k, b$  — коэффициенты линейной зависимости (угол наклона и смещение соответственно);  $n$  — величина исходной выборки;  $y, x$  — координаты исходной выборки.

Найдем коэффициенты  $a$  и  $b$ , при которых функция (1) принимает наименьшее значение. Для этого решим систему уравнений из частных производных данной функции по соответствующим коэффициентам линейной зависимости:

$$\begin{cases} a = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2} \\ b = \frac{\sum_{i=1}^n y_i - a \sum_{i=1}^n x_i}{n} \end{cases}$$

**Метод достижения консенсуса на основе случайных выборок.** В случае двумерного пространства используем метод RANSAC, который заключается в случайном выборе двух опорных точек из исходного набора данных и оценке степени удаленности всех точек от получившейся прямой [12]. Точки, лежащие в пределах заданного расстояния, называются невыбросами. В результате выберем прямую, имеющую наибольшее количество невыбросов в окрестности, заданной определенным заранее граничным расстоянием. Выбор опорных точек происходит итеративно. Оценка количества итераций имеет вид:

$$k = \frac{\lg(1 - p)}{\lg(1 - \omega^n)},$$

где  $p$  — вероятность существования опорных точек без выбросов;  $\omega$  — доля невыбросов;  $n$  — количество точек в исходном наборе данных.

**Метод главных компонент (PCA).** В задачах линейной регрессии применяется метод PCA, который получил широкое распространение, благодаря возможности поиска линейных подпространств, где суммарная дисперсия проекции исходных данных минимальна [13]. Цель метода — поиск собственных векторов и соответствующих собственных значений. В случае линейной аппроксимации в двумерном пространстве достаточно найти два ортогональных собственных вектора. Если выбрать вектор с наибольшим собственным значением, то для нахождения искомой прямой достаточно провести линию в направлении данного вектора через «центр масс» исходного набора данных. В случае вектора с наименьшим собственным значением необходимо провести прямую через центр масс перпендикулярно направлению данного вектора.

По результатам анализа методов выявлено, что методы RANSAC и PCA в своих реализациях требуют проведения операций с матрицами, которые сами по себе обладают нелинейной алгоритмической сложностью. Кроме того, данные матрицы необходимо полностью проинициализировать в начале исполнения алгоритма, что при больших наборах данных ведет к повышенному объему потребления памяти. МНК, в свою очередь, лишен данных недостатков. Однако квадратичные зависимости могут привести к переполнениям в условиях ограниченной разрядной сетки конкретной архитектуры. В связи с этим возникла необходимость разработки метода аппроксимации, удовлетворяющего всем ранее заявленным требованиям.

**Описание предлагаемого метода**

Пусть имеется некоторый массив точек, упорядоченный по оси  $x$ . Каждая точка имеет координаты  $x$  и  $y$ :  $(x_i; y_i)$ . Угол наклона аппроксимирующей прямой можно оценить путем расчета среднего относительного отклонения упорядоченных точек ( $\Delta$ ) по оси  $y$  друг от друга:

$$\Delta = \frac{\sum_{i=2}^N (y_i - y_{i-1})}{x_n - x_1}, \tag{2}$$

где  $N$  — количество точек в исходном наборе;  $x_n$  и  $x_1$  — координаты крайней правой и левой точек по оси  $x$ . Отметим, что разница координат по оси  $y$  берется не по модулю.

Чтобы определить первую точку, через которую будет проведена аппроксимирующая прямая, необходимо определить координаты  $x_c$  и  $y_c$  «центра масс» упорядоченного исходного набора точек:

$$\begin{cases} x_c = \frac{\sum_{i=1}^N x_i}{N} \\ y_c = \frac{\sum_{i=1}^N y_i}{N} \end{cases}. \tag{3}$$

Используя полученную величину среднего отклонения (2), получим координаты  $x_b$  и  $y_b$  второй точки, лежащей на искомой прямой:

$$\begin{cases} x_b = x_c + 1 \\ y_b = y_c + \Delta \end{cases}. \tag{4}$$

Уравнение результирующей аппроксимирующей прямой  $y = kx + b$  сформируем с помощью известных формул для нахождения угла наклона прямой  $k$  и смещения  $b$ :

$$\begin{cases} k = \frac{y_b - y_c}{x_b - x_c} \\ b = y_b - \frac{x_b(y_b - y_c)}{x_b - x_c} \end{cases}. \tag{5}$$

Пусть дан упорядоченный массив точек, имеющих следующие координаты: (1; 1); (2; 3); (3; 2); (4; 3); (5; 3);

(6; 5); (7; 4). Рассчитаем среднее относительное отклонение точек (2):  $\Delta = 0,5$ . Определим координаты «центра масс» набора данных в соответствии с системой уравнений (3):  $x_c = 4; y_c = 3$ , а координаты второй точки, лежащей на искомой прямой с (4):  $x_b = 5; y_b = 3,5$ . С помощью полученных координат двух точек и системы уравнений (5) рассчитаем угол наклона  $k$  и смещение  $b$  искомой прямой:  $k = 0,5; b = 1$ . В результате получим уравнение аппроксимирующей прямой:

$$y = kx + b = 0,5x + 1.$$

**Постановка эксперимента**

Для проведения экспериментов и оценки точности рассматриваемых алгоритмов на языке Perl разработан генератор исходного набора данных, позволяющий получить на основе уравнения некоторой прямой набор упорядоченных по оси  $x$  точек, разбросанных относительно заданной прямой  $y'$  (рис. 2).

Использование генератора случайных чисел является причиной возможной ситуации, когда среднее отклонение  $\Delta_{cp}$  всех сгенерированных точек становится отличным от нуля, тем самым несколько смещая прямую  $y'$  и образуя прямую  $y''$ . Таким образом, точность аппроксимирующего метода оценим относительно прямой  $y''$ .

Отметим, что изначально для оценки точности коэффициентов прямой  $y''$  нет, так как на вход алгоритмам подается только набор сгенерированных упорядоченных точек. Для решения данной задачи разработанный генератор также рассчитывает среднее результирующее отклонение сгенерированных точек от заданной прямой  $y'$ . Таким образом, прямая  $y''$ , относительно которой необходимо оценивать точность рассматриваемых методов, численно представлена в виде:

$$\Delta_{cp} = \frac{\sum_{i=1}^N \Delta_i}{N}.$$

Рассчитаем параметр  $\Delta_{cp}$  для каждого конкретного метода и оценим относительную точность рассматриваемых методов аппроксимации.

Из-за несоответствия методов RANSAC и PCA предъявляемым ограничениям по быстродействию и потреблению памяти целесообразно сравнивать разработанный метод только с МНК.

В процессе проведения экспериментов сгенерируем множество синтетических наборов данных. Сравним реализации методов на языке C<sup>1</sup>. Разработанный метод и МНК используют итеративный подход к расчету промежуточных значений, считывая на каждой итерации координаты очередной точки. Такой подход обеспечивает константное, близкое друг к другу потребление памяти в обеих реализациях и позволяет исключить его из показателей для сравнения. Время исполнения

<sup>1</sup> Github zhmylove/Math-LinearApprox. ZFLAM – fast linear approximation of 2D sequential points [Электронный ресурс]. Режим доступа: <https://github.com/zhmylove/Math-LinearApprox> (дата обращения: 08.12.2022).

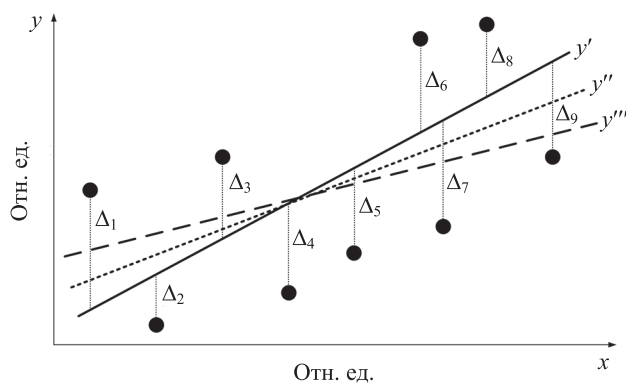


Рис. 2. Оценка точности метода относительно сгенерированной прямой

Fig. 2. Estimation of the method accuracy with respect to the generated straight line

на процессоре замерим только для алгоритмической части каждой реализации, так как в этом случае оно не включает в себя временные затраты на операции ввода/вывода.

Выполним аппроксимацию точек в исходном наборе данных в диапазоне от 10 до 10 000 точек. Для каждого количества точек проведем 100 000 экспериментов для снижения влияния генератора случайных чисел на результат и повышения точности итоговой оценки.

Эксперименты осуществим на тестовом стенде с установленным процессором AMD Ryzen 7 2700X. Запуск реализаций выполним в окружении WSL2 под управлением операционной системы Windows 10.

### Результаты

По завершению экспериментов выявлено, что с доверительной вероятностью 0,95, начиная с 50 точек в исходном наборе данных, разработанный метод затрачивает на расчет меньше процессорного времени, чем МНК (рис. 3). Снижение времени выполнения растет линейно и достигает 23 % при количестве точек в исходном наборе данных равном 10 000.

Определено, что с доверительной вероятностью 0,95 значение среднего отклонения для обоих методов в рамках экспериментов одинаково, что свидетельствует об одинаковой их точности.

Выбранный диапазон количества точек для аппроксимации в экспериментах покрывает большинство реальных наборов данных, который должен поступать от системы мониторинга для оценки скорости изменения нагрузки. Например, если обратиться к графику суммарного транзитного трафика через узлы MSK-IX в день и в неделю, то можно заметить, что рост нагрузки

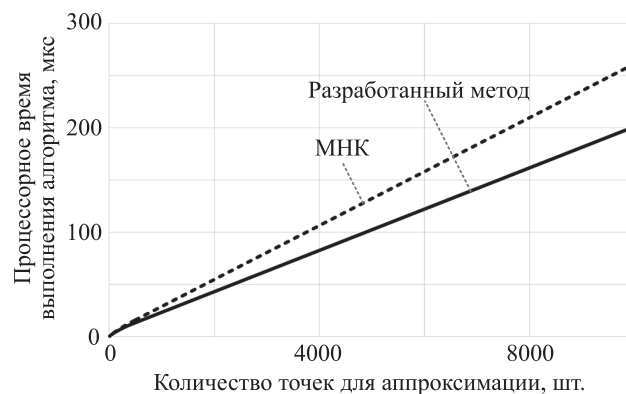


Рис. 3. Сравнение результатов разработанного метода и метода наименьших квадратов

Fig. 3. Comparison of the developed method results and LSM

наблюдается в среднем начиная с 04:00 и заканчивая 20:00<sup>1</sup>. В ежедневном графике шаг по времени равен 5 мин. Таким образом, суммарное количество точек для аппроксимации равно 32. Однако данный график является агрегированным по множеству систем. В каждой конкретной системе могут быть свои особенности роста и снижения нагрузки и временного шага получения данных с мониторинга. Например, при том же периоде роста, данные от мониторинга могут поступать каждые 10 с, что в итоге дает на выходе 5760 точек для аппроксимации. Такая частота опроса широко распространена в окружении интернета вещей, когда необходимо снимать показания с аналоговых датчиков или сенсоров с некоторой степенью агрегации по времени [14].

### Заключение

Предложенный метод линейной аппроксимации временных параметров в эластичных системах позволяет добиться снижения времени выполнения аппроксимации по сравнению с методом наименьших квадратов с сохранением точности результата. Также разработанный метод, в отличие от метода наименьших квадратов, лишен недостатка, связанного с операциями с квадратичными зависимостями, что обеспечивает возможность более широкого его применения в условиях ограниченности разрядной сетки некоторых архитектур.

<sup>1</sup> Акционерное общество «Центр взаимодействия компьютерных сетей «МСК-IX» (АО «ЦВКС «МСК-IX»). Суммарный трафик участников, передаваемый через MSK-IX [Электронный ресурс]. Режим доступа: <https://www.msk-ix.ru/traffic/> (дата обращения: 08.12.2022).

## Литература

1. Aliev T.I., Rebezova M.I., Russ A.A. Statistical methods for monitoring travel agencies in the settlement system // Automatic Control and Computer Sciences. 2015. V. 49. N 6. P. 321–327. <https://doi.org/10.3103/s0146411615060024>
2. Lorido-Botran T., Miguel-Alonso J., Lozano J.A. A review of auto-scaling techniques for elastic applications in cloud environments // Journal of Grid Computing. 2014. V. 12. N 4. P. 559–592. <https://doi.org/10.1007/s10723-014-9314-7>
3. Nikraves A.Y., Ajila S.A., Lung C.H. Towards an autonomic auto-scaling prediction system for cloud resource provisioning // Proc. of the IEEE/ACM 10<sup>th</sup> International Symposium on Software Engineering for Adaptive and Self-Managing Systems. 2015. P. 35–45. <https://doi.org/10.1109/seams.2015.22>
4. Moore L.R., Bean K., Ellahi T. Transforming reactive auto-scaling into proactive auto-scaling // Proc. of the 3<sup>rd</sup> International Workshop on Cloud Data and Platforms (CloudDP '13). 2013. P. 7–12. <https://doi.org/10.1145/2460756.2460758>
5. Aceto G., Botta A., de Donato W., Pescapè A. Cloud monitoring: A survey // Computer Networks. 2013. V. 57. N 9. P. 2093–2115. <https://doi.org/10.1016/j.comnet.2013.04.001>
6. Portnoy M. Virtualization Essentials. John Wiley & Sons, 2012. 286 p.
7. Bogatyrev V.A., Bogatyrev A.V. Functional reliability of a real-time redundant computational process in cluster architecture systems // Automatic Control and Computer Sciences. 2015. V. 49. N 1. P. 46–56. <https://doi.org/10.3103/S0146411615010022>
8. Bogatyrev V.A., Parshutina S.A., Poptcova N.A., Bogatyrev A.V. Efficiency of redundant service with destruction of expired and irrelevant request copies in real-time clusters // Communications in Computer and Information Science. 2016. V. 678. P. 337–348. [https://doi.org/10.1007/978-3-319-51917-3\\_30](https://doi.org/10.1007/978-3-319-51917-3_30)
9. Богатырев В.А., Богатырев С.В. Резервированная передача данных через агрегированные каналы в сети реального времени // Известия вузов. Приборостроение. 2016. Т. 59. № 9. С. 735–740. <https://doi.org/10.17586/0021-3454-2016-59-9-735-740>
10. Remondino F. Detectors and descriptors for photogrammetric applications // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2006. V. 36. N 3. P. 49–54.
11. Markovsky I., Van Huffel S. Overview of total least-squares methods // Signal Processing. 2007. V. 87. N 10. P. 2283–2302. <https://doi.org/10.1016/j.sigpro.2007.04.004>
12. Fischler M.A., Bolles R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography // Communications of the ACM. 1981. V. 24. N 6. P. 381–395. <https://doi.org/10.1145/358669.358692>
13. Abdi H., Williams L.J. Principal component analysis // Wiley Interdisciplinary Reviews: Computational Statistics. 2010. V. 2. N 4. P. 433–459. <https://doi.org/10.1002/wics.101>
14. Othman M.M., El-Mousa A. Internet of Things & Cloud Computing Internet of Things as a service approach // Proc. of the 11<sup>th</sup> International Conference on Information and Communication Systems (ICICS). 2020. P. 318–323. <https://doi.org/10.1109/icics49469.2020.239503>

## Авторы

**Мартынюк Илья Геннадьевич** — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, [sc 57208319866](https://orcid.org/0000-0003-2409-9677), <https://orcid.org/0000-0003-2409-9677>, [mt4.ilja@gmail.com](mailto:mt4.ilja@gmail.com)

**Жмылёв Сергей Александрович** — кандидат технических наук, инженер-консультант, ООО «ГК Ядро», Москва, 125373, Российская Федерация, [sc 57208416951](https://orcid.org/0000-0001-8916-5474), <https://orcid.org/0000-0001-8916-5474>, [s.zhmylev@yadro.com](mailto:s.zhmylev@yadro.com)

Статья поступила в редакцию 30.12.2022  
Одобрена после рецензирования 25.04.2023  
Принята к печати 25.05.2023

## References

1. Aliev T.I., Rebezova M.I., Russ A.A. Statistical methods for monitoring travel agencies in the settlement system. Automatic Control and Computer Sciences, 2015, vol. 49, no. 6, pp. 321–327. <https://doi.org/10.3103/s0146411615060024>
2. Lorido-Botran T., Miguel-Alonso J., Lozano J.A. A review of auto-scaling techniques for elastic applications in cloud environments. Journal of Grid Computing, 2014, vol. 12, no. 4, pp. 559–592. <https://doi.org/10.1007/s10723-014-9314-7>
3. Nikraves A.Y., Ajila S.A., Lung C.H. Towards an autonomic auto-scaling prediction system for cloud resource provisioning. Proc. of the IEEE/ACM 10<sup>th</sup> International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2015, pp. 35–45. <https://doi.org/10.1109/seams.2015.22>
4. Moore L.R., Bean K., Ellahi T. Transforming reactive auto-scaling into proactive auto-scaling. Proc. of the 3<sup>rd</sup> International Workshop on Cloud Data and Platforms (CloudDP '13), 2013, pp. 7–12. <https://doi.org/10.1145/2460756.2460758>
5. Aceto G., Botta A., de Donato W., Pescapè A. Cloud monitoring: A survey. Computer Networks, 2013, vol. 57, no. 9, pp. 2093–2115. <https://doi.org/10.1016/j.comnet.2013.04.001>
6. Portnoy M. Virtualization Essentials. John Wiley & Sons, 2012, 286 p.
7. Bogatyrev V.A., Bogatyrev A.V. Functional reliability of a real-time redundant computational process in cluster architecture systems. Automatic Control and Computer Sciences, 2015, vol. 49, no. 1, pp. 46–56. <https://doi.org/10.3103/S0146411615010022>
8. Bogatyrev V.A., Parshutina S.A., Poptcova N.A., Bogatyrev A.V. Efficiency of redundant service with destruction of expired and irrelevant request copies in real-time clusters. Communications in Computer and Information Science, 2016, vol. 678, pp. 337–348. [https://doi.org/10.1007/978-3-319-51917-3\\_30](https://doi.org/10.1007/978-3-319-51917-3_30)
9. Bogatyrev V.A., Bogatyrev S.V. Redundant data transmission using aggregated channels in real-time network. Journal of Instrument Engineering, 2016, vol. 59, no. 9, pp. 735–740. (in Russian). <https://doi.org/10.17586/0021-3454-2016-59-9-735-740>
10. Remondino F. Detectors and descriptors for photogrammetric applications. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2006, vol. 36, no. 3, pp. 49–54.
11. Markovsky I., Van Huffel S. Overview of total least-squares methods. Signal Processing, 2007, vol. 87, no. 10, pp. 2283–2302. <https://doi.org/10.1016/j.sigpro.2007.04.004>
12. Fischler M.A., Bolles R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 1981, vol. 24, no. 6, pp. 381–395. <https://doi.org/10.1145/358669.358692>
13. Abdi H., Williams L.J. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, vol. 2, no. 4, pp. 433–459. <https://doi.org/10.1002/wics.101>
14. Othman M.M., El-Mousa A. Internet of Things & Cloud Computing Internet of Things as a service approach. Proc. of the 11<sup>th</sup> International Conference on Information and Communication Systems (ICICS), 2020, pp. 318–323. <https://doi.org/10.1109/icics49469.2020.239503>

## Authors

**Ilya G. Martynchuk** — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, [sc 57208319866](https://orcid.org/0000-0003-2409-9677), <https://orcid.org/0000-0003-2409-9677>, [mt4.ilja@gmail.com](mailto:mt4.ilja@gmail.com)

**Sergei A. Zhmylev** — PhD, Engineering consultant, GK YADRO LLC, Moscow, 125373, Russian Federation, [sc 57208416951](https://orcid.org/0000-0001-8916-5474), <https://orcid.org/0000-0001-8916-5474>, [s.zhmylev@yadro.com](mailto:s.zhmylev@yadro.com)

Received 30.12.2022  
Approved after reviewing 25.04.2023  
Accepted 25.05.2023



Работа доступна по лицензии  
Creative Commons  
«Attribution-NonCommercial»