

УДК 004.042

DOI: [10.26102/2310-6018/2023.42.3.027](https://doi.org/10.26102/2310-6018/2023.42.3.027)

## Архитектура программного модуля сбора и обработки временных рядов зонтичной системы мониторинга ИТ-инфраструктуры

А.С. Каменев✉, Ю.С. Сахаров

*Государственный университет «Дубна», Дубна, Российская Федерация*

**Резюме.** Статья посвящена проблеме сбора временных рядов зонтичной системы мониторинга ИТ-инфраструктуры с последующей обработкой полученных данных в режиме реального времени. Актуальность исследования обусловлена возрастающим интересом к системам подобного класса со стороны крупных предприятий и организаций с высокой степенью цифровизации производственных процессов. В свою очередь, организация процесса сбора такой информации обусловлена рядом особенностей: во-первых, программные модули должны проектироваться с учетом значительной нагрузки (сбор и обработка порядка 10 млн значений временных рядов в минуту); во-вторых, сбор данных зачастую осуществляется не с конечных устройств, а из других систем мониторинга. Также требуется учитывать современное состояние ИТ-инфраструктуры, характеризующееся своим динамизмом, вызванным развитием и повсеместным внедрением технологий аппаратной виртуализации, контейнеризации приложений и автоматизированного управления конфигурациями. На основе сравнения подходов к сбору и обработке временных рядов, реализованных в различных средствах мониторинга, в работе делается вывод о перспективности применения и развития в зонтичных системах мониторинга Prometheus-подхода. Авторы предлагают свой вариант адаптации и развития данного подхода. Отличительными чертами предложенного варианта являются мультистатусная модель порогов с временем жизни, а также опосредованная установка связей между объектами в ресурсно-сервисной модели и собранной метрической информацией, что позволяет реализовать требуемую предприятиями функциональность сбора и обработки временных рядов для зонтичной системы мониторинга в условиях высокой нагрузки и динамизма современной ИТ-инфраструктуры. В завершении приводятся результаты предварительных испытаний разработанного программного модуля, а также делается заключение о возможности использования предложенного авторами подхода для реализации функции контроля степени покрытия мониторинга объектов. В настоящее время описанный вариант архитектуры используется в коммерческом программном продукте «MONQ» и проходит апробацию в ряде ключевых российских предприятий.

**Ключевые слова:** система мониторинга, временные ряды, ИТ-сервис, ресурсно-сервисная модель, система управления услугами, AIOps, big data.

**Для цитирования:** Каменев А.С. Сахаров Ю.С. Архитектура программного модуля сбора и обработки временных рядов зонтичной системы мониторинга ИТ-инфраструктуры. *Моделирование, оптимизация и информационные технологии.* 2023;11(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=1409> DOI: 10.26102/2310-6018/2023.42.3.027

## AIOps system software module architecture for collecting and processing IT infrastructure time series

A.S. Kamenev✉, Y.S. Sakharov

*Dubna State University, Dubna, the Russian Federation*

**Abstract.** The article examines the problem of collecting time series data by AIOps system for monitoring the IT infrastructure with subsequent processing of the received data in real time. The

relevance of the study is due to the growing interest in systems of this class on the part of large enterprises and organizations with a high degree of production process digitalization. In its turn, the organization of the process of collecting such information is conditioned by a number of features: firstly, software modules must be designed taking into account a significant load (collection and processing of about 10 million metrics per minute); secondly, end devices are not often used to collect data, other monitoring systems are employed instead. It is also required to consider the current state of the IT infrastructure characterized by its dynamism caused by the development and widespread implementation of hardware virtualization technologies, application containerization and automated configuration management. Based on a comparison of approaches to the collection and processing of time series data implemented in various monitoring tools, the paper concludes that the application and development of the Prometheus approach in AIOps monitoring systems is promising. The authors offer their own version of the adaptation and development of this approach. Distinctive features of the proposed option are a multi-status model of thresholds with a lifetime as well as the indirect establishment of links between objects in the resource-service model and the collected metric information, which helps to implement the functionality required by enterprises for collecting and processing metrics for an AIOps monitoring system under the conditions of high load and dynamism of modern IT infrastructure. In conclusion, the results of the developed software module preliminary testing are presented, and the possibility of using the approach proposed by the authors to implement the function of controlling the degree of monitoring object coverage is underscored. Currently, the described version of the architecture is used in the commercial software product "MONQ" and is being tested in several key Russian enterprises.

**Keywords:** monitoring system, time series, IT service, resource-service model, service management system, AIOps, big data.

**For citation:** Kamenev A.S., Sakharov Y.S. AIOps system software module architecture for collecting and processing IT infrastructure time series. *Modeling, Optimization and Information Technology*. 2023;11(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=1409> DOI: 10.26102/2310-6018/2023.42.3.027 (In Russ.).

## Введение

Растущая гетерогенность и сложность ИТ-инфраструктуры неминуемо ведет к росту разнообразия средств мониторинга и контроля. Системы сбора и обработки логов, классические системы инфраструктурного мониторинга, системы мониторинга производительности приложений (application performance monitoring), средства роботизированного тестирования приложений, продукты для трассировки взаимосвязанных программных компонент, средства автоматической инвентаризации (автообнаружение), SIEM-системы (системы анализа и предотвращения угроз информационной безопасности), платформы визуализации – вот неполный перечень видов одновременно использующихся инструментов мониторинга ИТ-инфраструктуры на крупном предприятии. Столь популярный в последние годы продуктовый подход и проектное управление в организациях еще более усложняют ландшафт: зачастую каждая продуктовая команда использует свой отдельный экземпляр инструмента мониторинга. Все это приводит к необходимости использования систем зонтичного мониторинга (AIOps), для которых источником данных выступают другие инструменты мониторинга (системы мониторинга 1-го уровня). Внедрение зонтичного мониторинга позволяет обеспечить централизованную политику управления ИТ-инфраструктурой при сохранении разнообразия и продуктового подхода на предприятии [1].

Многие системы зонтичного мониторинга строятся на основе обработки получаемых из систем мониторинга 1-го уровня аварийных сообщений и различных логов (журналов). Обработка временных рядов предполагается на уровне систем мониторинга 1-го уровня, которые на их основе генерируют аварийные сообщения и уже их передают в зонтичный мониторинг. Представителями этого подхода можно назвать

программные продукты Splunk, BigPanda, Moogsoft, ELK, PagerDuty и MONQ, а также ряд других.

Но сбор одних только событий и логов имеет ключевой недостаток: не позволяет в рамках всего предприятия обеспечить единую политику сбора временных рядов ИТ-инфраструктуры. Например, используя данный подход, невозможно реализовать такой важный функционал как централизованный контроль степени покрытия инструментами мониторинга ИТ-сервисов. Также предприятия, внедряя технологии «больших данных» (big data) и «глубинного анализа данных» (data mining), стремятся организовать единое хранилище всей первичной информации: не только логов (текстовых сообщений), но и временных рядов [2]. В связи с этим в последние несколько лет в технических требованиях российских крупных организаций все чаще стало появляться требование реализовать в продуктах зонтичного мониторинга функционал сбора и обработки временных рядов. По данным внутреннего исследования компании «Монк Диджитал Лаб» 17 из 23 поступивших за 2022 год в том или ином виде технических требований от крупных российских предприятий содержали в себе данное требование.

В самом общем виде задача разработки архитектуры модуля сбора и обработки временных рядов может быть представлена в виде следующих взаимосвязанных частных задач:

1. Выявление особенностей и специальных требований, вытекающих из области применения зонтичных систем мониторинга.
2. Организация процессов сбора и хранения значений временных рядов.
3. Генерация событий из временных рядов.
4. Установление связи между временными рядами и конфигурационными единицами (объекты мониторинга) базы данных управления конфигураций ИТ-сервисов предприятия (CMDB).
5. Обеспечение поддержки принятия решения на основе обрабатываемых данных.

Далее будут рассмотрены результаты решения обозначенных задач и полученная нами в итоге архитектура программного модуля сбора и обработки временных рядов с рядом отличий, позволяющих использовать ее непосредственно для расширения функциональных возможностей систем зонтичного мониторинга. Предложенная архитектура была реализована для коммерческого программного продукта «MONQ» (версия 7.6.0 от февраля 2023 г.) и в текущий момент внедряется на ряде ключевых российских предприятий.

### **Особенности сбора временных рядов системами зонтичного мониторинга**

Сбор временных рядов системами зонтичного мониторинга по сравнению с классическими инструментами обладает рядом особенностей.

Во-первых, сбор данных необходимо осуществлять не с конечных устройств, а с других независимых систем мониторинга. Это требование вызвано следующими причинами: 1) предприятия на практике уже имеют настроенные и отлаженные на протяжении многих лет инструменты мониторинга, собирающие временные ряды с конечных устройств, поэтому перенос точек сбора повлечет удорожание и сдвиг сроков проекта внедрения; 2) конечные устройства и объекты (например, отраслевые АСУТП) могут обладать своей спецификой и специально предназначенными для них средствами мониторинга; 3) сохранение продуктового подхода, когда в рамках одной организации команды с разной степенью зрелости процессов могут выбирать свой набор инструментов. Данное требование приводит к тому, что значения временных рядов поступают не десятками тысяч малых информационных потоков (от каждого устройства

свой информационный поток), а несколькими десятками высоконагруженных потоков, без предварительного разбиения и идентификации по конечным объектам.

Во-вторых, программный модуль сбора временных рядов должен проектироваться с учетом высокой нагрузки. Например, усредненные параметры на проектах внедрения программного продукта «MONQ» составляют: сбор и обработку до 10 млн значений временных рядов в минуту из 20 источников с распределением по 500 тыс. конфигурационных единиц (далее KE) из базы данных управления конфигурацией ИТ-сервисов предприятия (CMDB). Очевидно, что при таких нагрузках проектируемая архитектура должна обладать свойством горизонтального масштабирования.

В-третьих, в случае внедрения проектируемого модуля в событийно-ориентированную систему зонтичного мониторинга, изначально рассчитанную на обработку текстовых сообщений, необходимо сочетать его с существующими сервисами, а также с интегральным информационным слоем по объектам мониторинга.

В-четвертых, следует учитывать динамизм современной ИТ-инфраструктуры предприятия. Использование технологий аппаратной виртуализации, контейнеризации приложений и автоматизированного управления конфигурациями приводит к ситуации, когда объект мониторинга существует порядка 30–60 минут. Система зонтичного мониторинга за этот срок должна успеть обнаружить объект, занести его в свою базу данных, соотнести с ним поступающие значения временных рядов и после исчезновения объекта сохранить информацию о его работе без накопления «мусорных» сущностей и данных, замедляющих работу системы [3].

Кратко сформулируем основные требования к модулю сбора и обработки временных рядов, вытекающие из области применения зонтичных систем мониторинга:

1. Поддержка горизонтального масштабирования и высокого коэффициента сжатия данных для обеспечения сбора и обработки десятков миллионов значений временных рядов в минуту.

2. Получение значений временных рядов в виде небольшого количества разнородных высоконагруженных информационных потоков.

3. Обеспечение единого интегрального информационного слоя для операторов вне зависимости от вида исходной информации: временного ряда или текстового сообщения.

4. Работа в условиях современной динамической ИТ-инфраструктуры, предполагающей короткое время жизни объектов мониторинга (порядка нескольких часов).

### **Хранение и сбор временных рядов**

Хранение данных временных рядов требует специфического системного программного обеспечения, функционал которого должен отличаться как от реляционных СУБД, так и от штатных решений NoSQL [4, 5], зачастую используемых для сбора и обработки текстовых сообщений (Elasticsearch, Splunk). В последние годы в качестве такого специализированного системного ПО выступает класс СУБД-ВР (TSDB).

Системное ПО хранения временных рядов для применения в решениях зонтичного мониторинга должно обладать следующими характеристиками:

- 1) высоким коэффициентом компрессии данных и низкой нагрузкой на дисковую подсистему;
- 2) горизонтальной масштабируемостью;

3) наличием кластерных отказоустойчивых вариантов реализации с горячим резервированием;

4) возможностью проводить математические операции над выборками временных рядов непосредственно в запросе;

5) встроенной поддержкой интеллектуального анализа данных временных рядов: поиском аномалий, функцией прогноза, восстановлением пропущенных значений и т. п.

В случае разработки зонтичной системы мониторинга для российского рынка используемое системное ПО должно соответствовать правилам формирования и ведения единого реестра российского ПО.

На момент разработки архитектуры модуля по данным рейтинга портала DB-Engines.com насчитывалось порядка сорока различных TSDB-решений, из которых наиболее популярными и обладающими необходимыми характеристиками являлись: InfluxDB, Prometheus, TimescaleDB, OpenTSDB и VictoriaMetrics. Согласно опубликованным результатам тестов производительности [6, 7, 8] Victoria Metrics обладает наилучшим коэффициентом компрессии данных и одним из лучших показателей использования оперативной памяти, уступая по данному показателю только TimescaleDB. Но TimescaleDB обладает значительно худшим по сравнению с VictoriaMetrics (в 8,6 раз) сублинейным пределом масштабируемости по приему данных на одной ноде. В случае приема большого объема (больше 2 млн значений временных рядов в секунду) превосходство TimescaleDB по использованию памяти нивелируется необходимостью построения кластера. Таким образом, в качестве центрального элемента нашего модуля приема и обработки временных рядов было выбрано системное ПО VictoriaMetrics.

### Обработка временных рядов

Помимо хранения от системы зонтичного мониторинга как системы поддержки принятия решения требуется обработка временных рядов, происходящая в режиме реального времени. Основными результатами такой обработки должны стать, во-первых, информационные сообщения о состоянии временного ряда, преобразующие количественные характеристики в качественные, во-вторых, установленные связи между временным рядом и объектом мониторинга (конфигурационной единицей) [9]. Для выбора за основу наиболее подходящего метода для обработки временных рядов в зонтичной системе мониторинга разберем два противоположных по своей сути подхода, которые используются в популярных инструментах мониторинга Zabbix и Prometheus. В качестве основного показателя предлагаем взять показатель RPS – количество запросов к хранилищу временных рядов в сек.

По данным портала аналитического агентства Gartner [10] Zabbix занимает 1-ю позицию среди наиболее популярных средств мониторинга ИТ-инфраструктуры с открытым кодом (и 3-ю в общем рейтинге). Zabbix – классическая универсальная система ИТ-мониторинга, предназначенная в первую очередь для мониторинга временных рядов с оборудования и операционных систем, и уже во вторую очередь – для мониторинга логов и приложений. В Zabbix для каждого временного ряда существует специальная сущность – элемент данных, привязанная к хосту (аналог конфигурационной единицы), для генерации аварийных сообщений используется сущность, также связанная с хостом, – триггер, содержащий правило обработки временного ряда, запускающее с определенной периодичностью и изменяющее статус триггера.

Стоит отметить, что данный подход удобен и прост для установления связи между конкретным объектом мониторинга и временным рядом, но в случае применения

данного подхода к зонтичной системе мониторинга, когда данные поступают не от объекта мониторинга, а от промежуточной системы мониторинга в произвольном формате необходимо будет для каждого временного ряда объекта мониторинга устанавливать правила фильтрации для выделения в принятых «сырых» данных искомого временного ряда. Для обеспечения функции оценки степени покрытия мониторингом в режиме реального времени, таким образом, потребуется постоянно с некоторой периодичностью опрашивать хранилище на предмет наличия «свежих» значений всех временных рядов.

Рассматриваемый подход предполагает, что в общем случае каждый триггер соответствует аварийному состоянию конкретного объекта (хоста). Таким образом, в системе мониторинга, использующий данный подход, должно быть одновременно активно столько правил обработки временных рядов, сколько возможно аварийных сообщений / состояний объектов мониторинга, а следовательно, столько же запросов к СУБД в единицу времени. В итоге можно составить следующую формулу (1) оценки нагрузки на хранилище (запросов в секунду) при применении Zabbix-подхода:

$$RPS_{TSDB} = RPS_{match} + RPS_{trigger} = CI(\overline{TS} * F_{match} + \overline{Tr} * F_{trigger}), \quad (1)$$

где  $RPS_{match}$  – нагрузка, генерируемая функцией выделения временных рядов и установления связи (запрос в сек.);  $RPS_{trigger}$  – нагрузка, генерируемая функцией расчета триггеров;  $CI$  – количество КЕ (объектов мониторинга),  $\overline{TS}$  – среднее количество временных рядов, характеризующих одну КЕ,  $F_{match}$  – необходимая частота проверки наличия данных по временным рядам,  $\overline{Tr}$  – среднее количество триггеров у одной КЕ,  $F_{trigger}$  – частота проверки выполнения условия триггера.

Применяя это формулу к усредненной инсталляции зонтичного мониторинга Monq, получаем:

$$RPS_{TSDB} = 5 * 10^5(100 * 0,003 + 100 * 0,017) = 10^6 \text{ запросов в сек.}$$

Prometheus – инструмент мониторинга, разрабатываемый некоммерческим консорциумом The Linux Foundation и, в первую очередь, предназначенный для сбора и обработки временных рядов контейнеризированных приложений [11]. Одной из ключевых особенностей Prometheus является отсутствие специальной выделенной сущности для хранения значений временного ряда («элемент данных» в Zabbix). Временной ряд формируется в момент запроса к БД из «сырых» данных, представленных в виде значения, временной метки и набора лейблов. Причем одним запросом можно сформировать несколько однотипных временных рядов и, соответственно, рассчитать пороговые значения для них.

Таким образом, Prometheus-подход позволяет реализовать пакетную обработку, объединяя в одном правиле, а значит и запросе, обработку временных рядов сразу для нескольких типовых объектов. Формула оценки нагрузки на хранилище (2) будет иметь следующий, вид:

$$RPS_{TSDB} = RPS_{match} + RPS_{trigger} = CI_{type}(\overline{TS} * F_{match} + \overline{Tr} * F_{trigger}) \quad (2)$$

где  $CI_{type}$  – количество типов<sup>1</sup> объектов мониторинга.

Применяя это формулу к усредненной инсталляции зонтичного мониторинга Monq, получаем:

$$RPS_{TSDB} = 10^4(100 * 0,003 + 100 * 0,017) = 2 * 10^4 \text{ запросов в сек.}$$

<sup>1</sup> Под типами понимаются группы объектов, объединенные единым шаблоном мониторинга.

Как видно, использование Prometheus-подхода, благодаря группировке правил для однотипных объектов мониторинга с последующей пакетной обработкой, позволяет кратно сократить нагрузку. Еще одним достоинством Prometheus-подхода является отсутствие необходимости создания таких специальных сущностей для каждого объекта мониторинга как триггеры, что значительно упрощает работу в условиях динамической ИТ-среды. Таким образом, можно сделать вывод о перспективности применения этого подхода для создания обработчика временных рядов в системах зонтичного мониторинга.

### Адаптация Prometheus-подхода

В свою очередь, для использования в системах зонтичного мониторинга Prometheus-подхода необходимо решить набор взаимосвязанных специфических проблем:

1. Организация связи между данными временных рядов с объектами мониторинга, хранящимися в CMDB.

2. Проблема «незавершенных аварийных ситуаций». Возникает в случаях прекращения поступления данных временного ряда при уже открытой аварии, что ведет к накоплению незавершенных аварийных ситуаций и создает информационный шум в системе мониторинга. Особенно остро стоит в зонтичных системах мониторинга, неконтролирующих сбор данных с конечных объектов мониторинга.

3. Обеспечение историчности: сохранение информации об аварийных ситуациях в условиях многократно изменяющейся топологии и конфигурации объектов мониторинга.

4. Поддержка функционала централизованного контроля покрытия объектов инструментами мониторинга (регулярность и полнота данных временных рядов о состоянии объектов).

Для решения обозначенных проблем нами была предложена и реализована концепция развития и адаптации Prometheus-подхода, включающая в себя следующие основные пункты:

– результатом обработки временного ряда является управление жизненным циклом нового специального объекта: порога. Порог по своей сути представляет собой понятную человеку смысловую «дискретизацию» временного ряда<sup>2</sup> (уровень критичности);

– порог, в отличие от триггера, – «короткоживущий» объект: у него есть время начала и время завершения, заново открывать уже завершённые пороги нельзя. Это требование позволяет сочетать необходимый динамизм с историчностью без накопления мусорных объектов, замедляющих систему;

– пороги, созданные по одному и тому же правилу и временному ряду, называются идентичными. Одновременно может быть открытым только один идентичный порог. В случае если происходит переход временного ряда от одного уровня критичности к другому: текущий порог закрывается, и открывается новый с другим уровнем критичности. Для обеспечения механизма идентичности каждому порогу присваивается хэш-идентификатор, формирующийся из правила и меток временного ряда;

– для решения проблемы «незавершенных проблем» у каждого порога есть срок жизни (TTL). После истечения этого времени порог закрывается. Если при пересчете

---

<sup>2</sup> В MONQ используется следующая шкала: зеленый (OK), голубой (Info), желтый (Warning), оранжевый (Major), красный (Critical), бордовый (Fatal).

правила происходит подтверждение порога, то срок жизни порога устанавливается заново;

– после создания порог при помощи специальных сценариев привязывается к объектам ресурсно-сервисной модели. Таким образом, связь временного ряда с объектом мониторинга осуществляется опосредовано через порог;

– правило порога всегда содержит минимально один уровень. Вследствие чего, в случае наличия соответствующего правила обеспечивается возможность контроля своевременности поступления временных рядов: если данные временного ряда поступают, то в системе всегда будет порог; в случае прекращения поступления данных, прекращают генерироваться сообщения подтверждения порога и по истечению срока жизни порог закрывается, отсутствие порога свидетельствует о прекращении поступления данных.

Отдельно стоит отметить, что изложенная концепция развития и адаптации Prometheus-подхода позволяет объединить в одной операции обращения к хранилищу как расчет правил, так и контроль поступления данных временных рядов, что дополнительно сокращает нагрузку на 15 %.

### Заключение

Изложенные в статье результаты исследований были заложены в основу разработанного модуля сбора и обработки временных рядов программного продукта «MONQ». Функциональная схема основного тракта сбора обработки данных с разработанным модулем представлена на Рисунке 1.

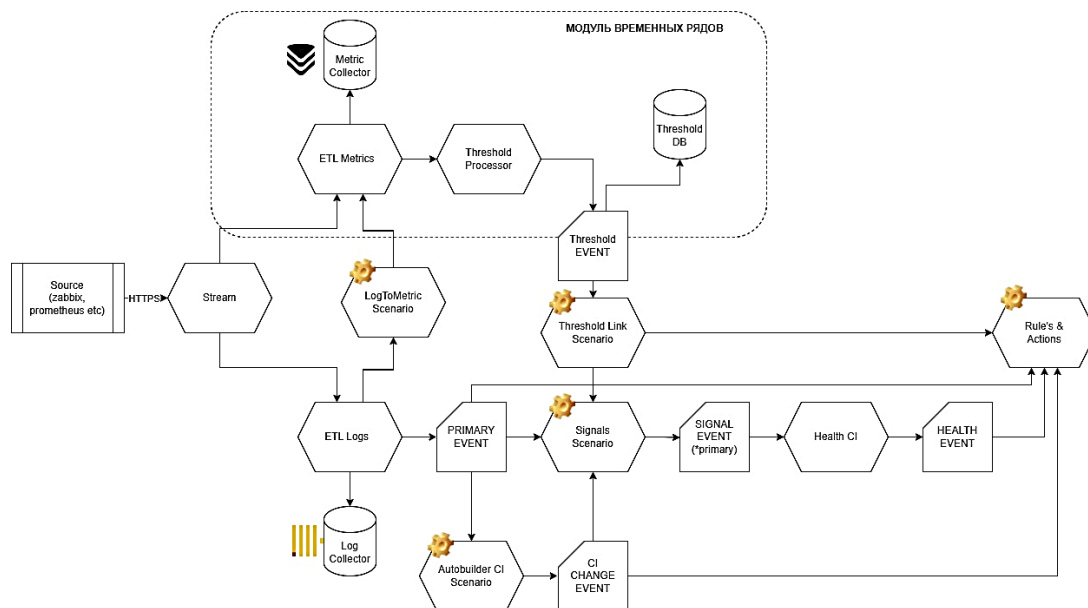


Рисунок 1 – Функциональная схема основного тракта сбора и обработки событий ПО «MONQ»  
Figure 1 – Functional diagram of the main pipeline for collecting and processing events of MONQ software

Временные ряды, как и события, поступают через блок потоков данных (Stream), управляющего сбором данных. Для приема метрических данных модуль временных рядов предоставляет точки API (для каждого формата данных временных рядов предусматривается своя точка API). Для отправки данных требуется указать ключ потока данных в заголовке или параметрах запроса. Поступившие на точки API данные преобразовываются в блоке ETL Metrics: в зависимости от точки приема, ETL блок



преобразовывает данные в формат Prometheus BrotoBuf, добавляет служебные метки и сохраняет их в хранилище временных рядов (СУБД «Victoria Metrics») через механизм RemoteWrite.

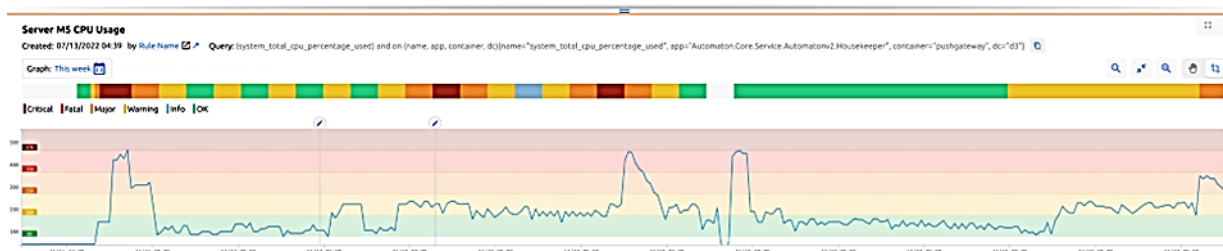


Рисунок 2 – Визуализация последовательности идентичных порогов в пользовательском интерфейсе ПО «MONQ»

Figure 2 – Visualization of a sequence of identical thresholds in the user interface of MONQ software

Блок расчета порогов (Threshold Processor) с заданной в правилах формирования порогов периодичностью отправляет запросы к хранилищу временных рядов и генерирует на выходе сообщения об открытии, подтверждении или закрытии порогов. Данные события сохраняются в БД порогов (используется реляционная СУБД PostgreSQL). Параллельно посредством общей шины асинхронного межмикросервисного обмена сообщениями (Rabbit MQ) события поступают на маршрутный узел привязки порогов к конфигурационным единицам ресурсно-сервисной модели (Threshold Link Scenario) встроенного движка low-code автоматизации, где по определенным программным сценариям в сгенерированных порогах проставляется метка об их связи с объектами мониторинга. Далее уже обогащенные пороги поступают в остальной тракт программного продукта, где коррелируются между собой и лог-событиями (Signal Scenario), а также запускают сценарии автоматизации бизнес-процессов (Rule's & Action). Примеры пользовательского интерфейса приведены на Рисунках 2 и 3.

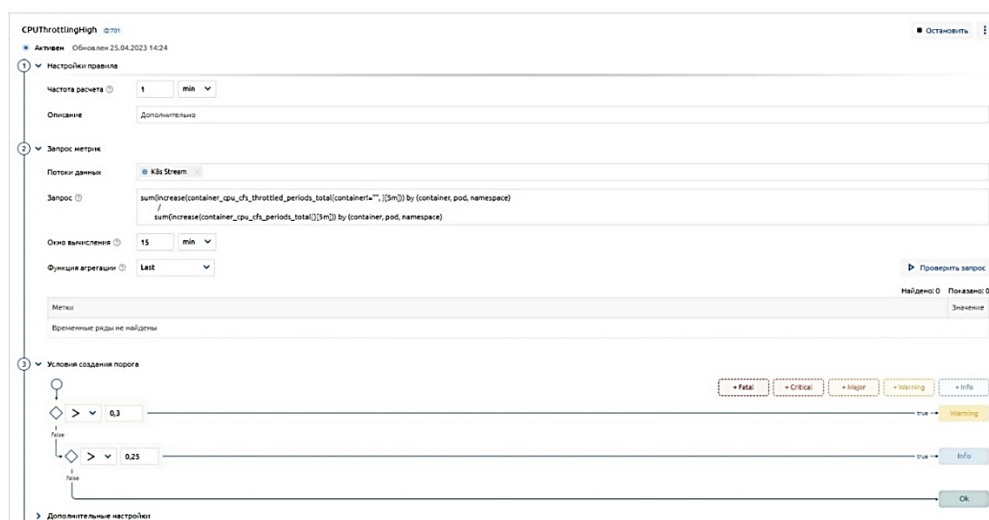


Рисунок 3 – Создание правила расчета порогов в ПО «MONQ»

Figure 3 – Creating a rule for calculating thresholds using MONQ software

Полученные результаты позволили обеспечить требуемый функционал сбора и обработки временных рядов в существующей системе зонтичного мониторинга,

основанной на событиях. Основными достоинствами разработанной архитектуры являются:

- гибкость. Позволяет реализовать сколь угодно сложные сценарии установления связи между временными рядами и объектами мониторинга;
- историчность. Связи между зарегистрированными порогами и объектами мониторинга сохраняются даже в случае удаления последних;
- массовая обработка и сниженная нагрузка на хранилище временных рядов. Благодаря применению адаптированного Prometheus-подхода одним запросом в базу данных можно получать результат по множеству сходных временных рядов. Также одним запросом объединяются функции установления порога и связи между временным рядом и объектом мониторинга;
- пригодность для современного динамического ИТ-окружения.

В настоящее время апробация результатов проходит на ряде ключевых российских предприятий, как в пилотном, так и промышленном режимах. В частности, разработанные решения применяются для обслуживания ИТ-инфраструктуры крупнейшего российского розничного ритейлера – X5 Group [12].

### СПИСОК ИСТОЧНИКОВ

1. Большаков М.А., Михайлов Г.В. Средства мониторинга ИТ-инфраструктуры ГВЦ ОАО "РЖД". *Научные разработки: евразийский регион: международная научная конференция теоретических и прикладных разработок, 20 мая 2019, Москва*. Уфа: Инфинити; 2019. С. 225–230.
2. Горшков С. Три шага к дата-центричной архитектуре. *Открытые системы. СУБД*. 2019;4:26.
3. Лазарева Н.Б. Оптимальный выбор системы мониторинга для различных типов ИТ-инфраструктур. *Инженерный вестник Дона*. 2022;4:60–69.
4. Иванова Е.В., Цымблер М.Л. Обзор современных систем обработки временных рядов. *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2020;9(4):79–97.
5. Намиот Д.Е. Базы данных временных рядов в системах "Интернета вещей". *Прикладная информатика*. 2017;12(2):79–87.
6. Valialkin A. High-cardinality TSDB benchmarks: VictoriaMetrics vs TimescaleDB vs InfluxDB. *Medium*. 2018. URL: <https://valyala.medium.com/high-cardinality-tsdb-benchmarks-victoriametrics-vs-timescaledb-vs-influxdb-13e6ee64dd6b> [дата обращения: 10.04.2023].
7. Valialkin A. Measuring vertical scalability for time series databases in Google Cloud. *Medium*. 2019. URL: <https://valyala.medium.com/measuring-vertical-scalability-for-time-series-databases-in-google-cloud-92550d78d8ae> [дата обращения: 10.04.2023].
8. Valialkin A. Prometheus vs VictoriaMetrics benchmark on node\_exporter metrics. *Medium*. 2020. URL: <https://valyala.medium.com/prometheus-vs-victoriametrics-benchmark-on-node-exporter-metrics-4ca29c75590f> [дата обращения: 10.04.2023].
9. Дубровин М.Г. Концепция проактивного мониторинга и управления объектами ИТ-инфраструктуры. *ИТНОУ: Информационные технологии в науке, образовании и управлении*. 2020;1:44–49.
10. Gartner Ranking & Review of IT-infrastructure monitoring tools. URL: <https://www.gartner.com/reviews/market/it-infrastructure-monitoring-tools> [дата обращения: 10.04.2023].
11. Brazil B. *Prometheus: Up & Running: Infrastructure and Application Performance Monitoring*. O'Reilly Media; 2018. 386 p.

12. X5 Group запускает платформу мониторинга ИТ-ландшафта на базе ИИ. *Сетевое издание CNews*. 2023 URL: [https://www.cnews.ru/news/line/2023-02-13\\_x5\\_group\\_zapuskaet\\_platformu](https://www.cnews.ru/news/line/2023-02-13_x5_group_zapuskaet_platformu) [дата обращения: 10.04.2023].

## REFERENCES

1. Bolshakov M.A., Mikhailov G.V. Monitoring tools for the IT infrastructure of the Main Computing Center of Russian Railways. *Scientific developments: the Eurasian region: international scientific conference of theoretical and applied developments, 20 May 2019, Moscow*. Ufa, Infinity; 2019. p. 225–230. (In Russ.).
2. Gorshkov S. Three steps to data-centric architecture. *Otkrytye sistemy. SUBD = Open systems. DBMS*. 2019;(4):26. (In Russ.).
3. Lazareva N.B. The optimal choice of monitoring system for various types of IT infrastructures. *Inzhenernyj vestnik Dona = Engineering journal of Don*. 2022;4:60–69. (In Russ.).
4. Ivanova E.V., Tsymler M.L. Review of modern time series processing systems. *Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika = Vestnik YURGU. Series: Computational Mathematics and Informatics*. 2020;9(4):79–97. (In Russ.).
5. Namiot D.E. Time series databases in the "Internet of Things" systems. *Prikladnaya informatika = Journal of Applied Informatics*. 2017;12(2):79–87. (In Russ.).
6. Valialkin A. High-cardinality TSDB benchmarks: VictoriaMetrics vs TimescaleDB vs InfluxDB. *Medium*. 2018. URL: <https://valyala.medium.com/high-cardinality-tsdb-benchmarks-victoriametrics-vs-timescaledb-vs-influxdb-13e6ee64dd6b> [accessed on 04.10.2023].
7. Valialkin A. Measuring vertical scalability for time series databases in Google Cloud. *Medium*. 2019. URL: <https://valyala.medium.com/measuring-vertical-scalability-for-time-series-databases-in-google-cloud-92550d78d8ae> [accessed on 04.10.2023].
8. Valialkin A. Prometheus vs VictoriaMetrics benchmark on node\_exporter metrics. *Medium*. 2020. URL: <https://valyala.medium.com/prometheus-vs-victoriametrics-benchmark-on-node-exporter-metrics-4ca29c75590f> [accessed on 04.10.2023].
9. Dubrovin M.G. The concept of proactive monitoring and management of IT infrastructure objects. *ITNOU: Informatsionnye tekhnologii v nauke, obrazovanii i upravlenii*. 2020;(1):44–49. (In Russ.).
10. Gartner Ranking & Review of IT infrastructure monitoring tools. URL: <https://www.gartner.com/reviews/market/it-infrastructure-monitoring-tools> [accessed on 04.10.2023].
11. Brazil B. *Prometheus: Up & Running: Infrastructure and Application Performance Monitoring*. O'Reilly Media; 2018. 386 p.
12. X5 Group launches AI-powered IT landscape monitoring platform. *Internet Media CNews*. 2023 URL: [https://www.cnews.ru/news/line/2023-02-13\\_x5\\_group\\_zapuskaet\\_platformu](https://www.cnews.ru/news/line/2023-02-13_x5_group_zapuskaet_platformu) [accessed on 04.10.2023]. (In Russ.).

## ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Каменев Алексей Сергеевич**, старший преподаватель кафедры проектирования электроники для установок «мегасайнс», Инженерно-физический институт, Государственный Университет «Дубна», Технический директор, ООО «Монк Диджитал Лаб», Дубна, Российская Федерация.

**Aleksey Sergeevich Kamenev**, Senior Lecturer at the Department of Electronics Design for Mega science Projects, Engineering Physics Institute, Dubna State University, Chief Technology Officer, Monq Digital Lab LLC, Dubna, the Russian Federation.

*e-mail:* [akamdragon@yandex.ru](mailto:akamdragon@yandex.ru)  
ORCID: [0000-0002-8028-4562](https://orcid.org/0000-0002-8028-4562)

**Сахаров Юрий Серафимович**, доктор технических наук, профессор, заведующий кафедрой проектирования электроники для установок «мегасайенс», Инженерно-физический институт, Государственный Университет «Дубна», Дубна, Российская Федерация.

*e-mail:* [sakharovu@yandex.ru](mailto:sakharovu@yandex.ru)

**Yuri Serafimovich Sakharov**, Doctor of Technical Sciences, Head of the Department of Electronics Design for Mega Science Projects, Engineering Physics Institute, Dubna State University, Dubna, the Russian Federation.

*Статья поступила в редакцию 03.08.2023; одобрена после рецензирования 29.08.2023; принята к публикации 27.09.2023.*

*The article was submitted 03.08.2023; approved after reviewing 29.08.2023; accepted for publication 27.09.2023.*