

МЕТОДЫ ОБРАБОТКИ ОТСУТСТВУЮЩИХ ЗНАЧЕНИЙ С ПОДСТАНОВКОЙ НУЛЯ В PROMQL И METRICSQL

Казначеева Наталия Леонидовна

канд. экон. наук, проф., Сибирский государственный университет телекоммуникаций и информатики, РФ, г. Новосибирск

Леньшина Дарья Сергеевна

студент, Сибирский государственный университет телекоммуникаций и информатики, РФ, г. Новосибирск

Кравцова Анна Александровна

студент, Сибирский государственный университет телекоммуникаций и информатики, РФ, г. Новосибирск

METHODS FOR HANDLING MISSING VALUES WITH ZERO SUBSTITUTION IN PROMQL AND METRICSQL

Kaznacheeva Natalia Leonidovna

Candidate of economic sciences, professor, Siberian State University of Telecommunications and Informatics, Russia, Novosibirsk

Lenshina Daria Sergeevna

Student, Siberian State University of Telecommunications and Informatics, Russia, Novosibirsk

Kravtsova Anna Aleksandrovna

Student, Siberian State University of Telecommunications and Informatics, Russia, Novosibirsk

Аннотация. В статье рассматриваются методы обработки отсутствующих данных в результатах метрик Prometheus и VictoriaMetrics. Определяются технические и логические особенности интерпретации «no data», предлагаются практические подходы через языки запросов PromQL и MetricsQL. Помимо этого, уделяется внимание преимуществам и недостаткам функций, используемых в решении проблем. Приводятся примеры запросов с подстановкой нулевых значений. Описываются рекомендации по проектированию дашбордов и настройке алертов с учетом особенностей системы.

Abstract. The article discusses methods for processing missing data in the results of Prometheus and VictoriaMetrics metrics. Technical and logical features of interpreting "no data" are defined, and practical approaches are proposed through the PromQL and MetricsQL query languages.

Ключевые слова: Prometheus, PromQL, VictoriaMetrics, MetricsQL, мониторинг, метрики, алерты.

Keywords: Prometheus, PromQL, VictoriaMetrics, MetricsQL, monitoring, metrics, alerts.

В системах с развитой микросервисной архитектурой мониторинг и анализ метрик становятся необходимыми для отслеживания состояния системы и её поведения. Метрики являются числовыми показателями, отражающими работу сервиса. Все данные собираются в реальном времени, затем анализируются и используются для выявления аномалий, построения графиков и настройки оповещений (алертов).

Современные системы мониторинга основываются на метриках с применением специализированных СУБД, таких как Prometheus и VictoriaMetrics. С помощью метрик можно количественно отслеживать поведение системы: количество HTTP-запросов, среднее время ответа, количество ошибок, загрузку CPU и прочее. Благодаря собранным данным инженерам удастся оперативно реагировать на проблемы, а бизнес-стороне принимать решения на основе фактических показателей работы системы.

Одна из наиболее частых практических проблем при работе с метриками – отсутствие данных в результатах запросов. Например, подобная ситуация может возникнуть, когда сервис не отправляет значение по определённой метке. В таком случае система мониторинга интерпретирует значение как «no data», но с точки зрения бизнес-логики это может означать лишь отсутствие активности. В инструменте визуализации Grafana такие ситуации могут проявиться как пробо на графике, или – некорректная работа алертов. В связи с этим очень важно обрабатывать значение 0 в логических обоснованных случаях [1].

PromQL является основным языком запросов в системе мониторинга Prometheus. Для подстановки нуля вместо отсутствующих данных в ответах запроса часто используется выражение «OR on() vector(0)». Его работа заключается в следующем: в случае, когда отсутствуют результаты основного выражения, система вернет ноль, подставленный из искусственно созданного временного ряда. При этом имеются ограничения: корректная работа будет достигнута только при запросах к одиночным временным рядам. При наличии множества лейблов (job, status, instance) возможно появление побочных эффектов, включая генерацию дополнительных рядов с неполными или пустыми метками.

MetricsQL – это язык запросов, используемый в системе VictoriaMetrics. В отличие от PromQL, он предоставляет более элегантный механизм обработки отсутствующих значений с помощью оператора «default 0», который, в свою очередь, автоматически подставляет ноль вместо отсутствующих данных, но при этом сохраняется исходная структура меток. Данный подход является особенно эффективным при построении многомерных графиков и таблиц, содержащих обширные множества лейблов. Однако, при этом следует учитывать, что MetricsQL не является полной реализацией PromQL. Поскольку имеются отличия в реализации агрегирующих функций, механизмах фильтрации и правилах обработки значений NaN - степень совместимости между языками оценивается ориентировочно в пределах 72-74%.

На практике работа с отсутствующими данными в системе мониторинга зачастую оказывается более значимой, чем это может показаться на первый взгляд. Так, в одной из крупных платформ интернет-торговли, реализованной на основе микросервисной архитектуры, было зафиксировано нестабильное срабатывание алертов, отслеживающих ошибки при оплате. При анализе выяснилось, что периоды, когда платёжный сервис не регистрировал события, соответствующая метрика не поступала в систему мониторинга. Визуализатор Grafana отображал такие участки «пустыми» (no data), в то время как система оповещений интерпретировала это как отсутствие проблем. Фактически, сервис пребывал в нерабочем состоянии, но это оставалось незамеченным. Решением данной проблемы стало использование конструкции «OR on() vector(0)» в PromQL – удалось явно интерпретировать отсутствие значений как нулевую активность и алерты стали работать корректно [2].

Пример 1. Запрос PromQL с подстановкой нуля:

```
sum(rate(http_requests_total{job="app",status="500"}[5m])) OR on(job, status) vector(0)
```

Помимо этого, подобное поведение наблюдается в визуализации с использованием Grafana: при отсутствии данных линия графика обрывается или отображается как NaN – это является серьезной проблемой и дезинформирует, особенно, если пользователь работает с загруженными дашбордами в режиме реального времени. Для устранения данной проблемы необходимо подставить нулевое значение на уровне самого запроса, но ни в коем случае не пытаться убирать пробелы в графике «визуально» средствами отображения.

В языке MetricsQL, используемом в VictoriaMetrics, аналогичная задача решается более удобным способом, используя оператор «default». Пример запроса:

Пример 2. Запрос PromQL с подстановкой нуля:

```
sum(rate(http_requests_total{status="500"}[5m])) default 0
```

Благодаря данному выражению эффективно сохраняется структура временных рядов, даже в ситуации, когда данные отсутствуют при отдельных сочетаниях лейблов (job, status или instance). Данная структура важна при построении таблиц, многомерных визуализациях или дашбордах, где полное представление данных критично для анализа [3].

Помимо этого, корректная интерпретация отсутствующих значений играет ключевую роль в отчётности и автоматизированной аналитике. Так, например, при формировании метрик доступности (SLA [5]), критичными являются даже кратковременные разрывы в данных. Неправильная интерпретация отсутствия метрик как нормального состояния может привести к снижению чувствительности мониторинга и риску нарушения обязательств по SLA [4].

Подход к обработке отсутствующих значений в мониторинговых системах не должен сводиться к техническому обходному решению, поскольку он должен быть частью системной стратегии, где важна не только визуализация, но и их корректная интерпретация в прикладном контексте.

Следует учитывать, что отсутствие метрики не всегда указывает на сбой, поскольку в ряде случаев оно отражает нормальное поведение системы. Так, при отсутствии HTTP-запросов с кодом 500 метрика `http_requests_total{status="500"}` не появляется – это не считается ошибкой. Но, в то же время, исчезновение метрики `up`, которая отвечает за состояние экспортёра, в свою очередь свидетельствует о проблеме недоступности узла и требует немедленного вмешательства. Необдуманная подстановка нулевого значения вместо отсутствующего, без логичного анализа, может исказить реальную ситуацию.

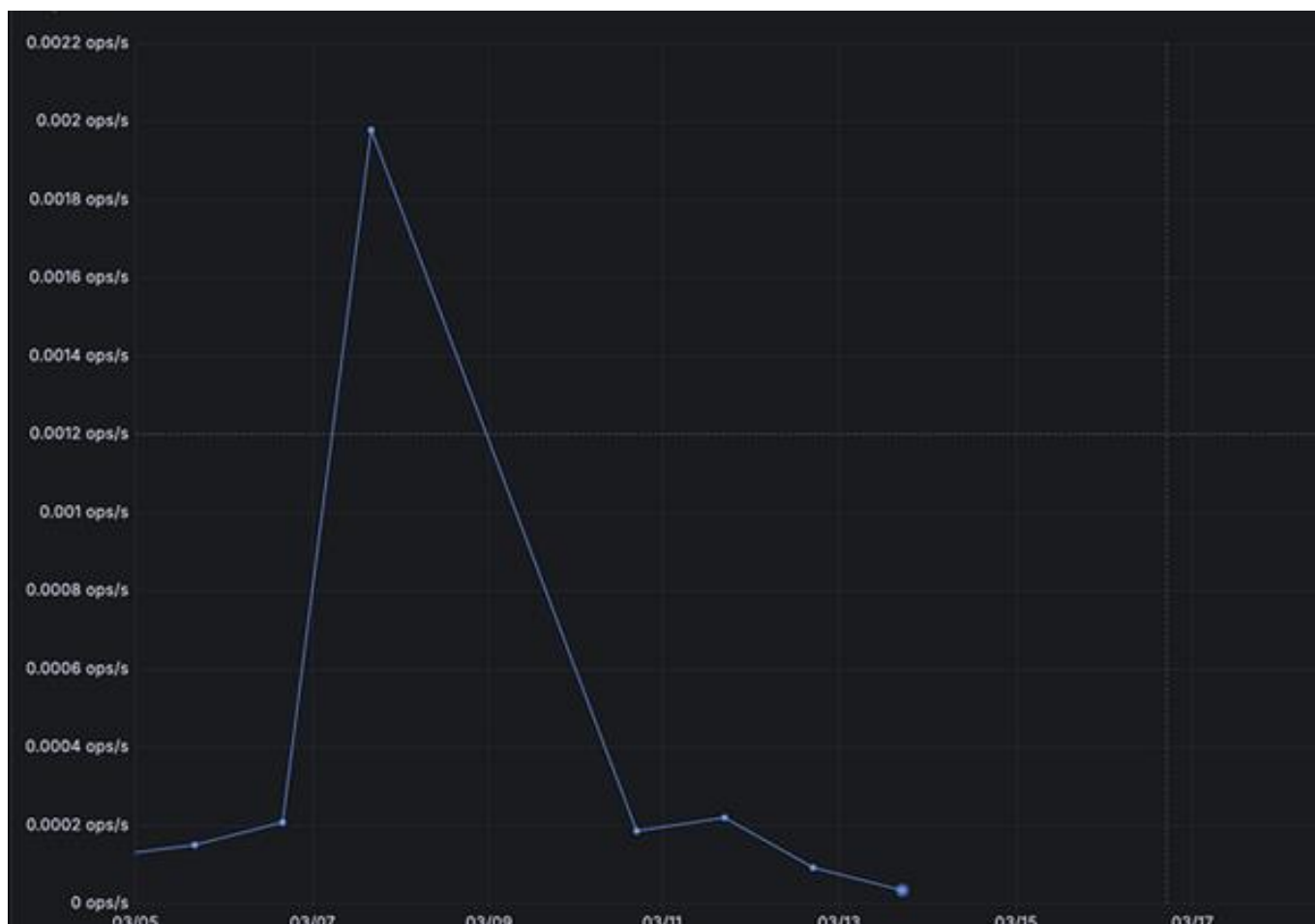


Рисунок 1. Различие между "no data" и нулевыми значениями в графике мониторинга

На графике имеются сегменты, где отсутствуют точки данных, и сегменты, где визуализируется серия с нулевыми значениями – эти участки отображаются как нулевой уровень и воспринимаются как отсутствие активности, а не ошибка.

При проектировании и визуализации алертов большое внимание стоит уделять контексту, поскольку автоматическая подстановка нуля может быть оправдана для графиков, отражающих количественную активность, но в то же время не следует допускать этого в критичных дашбордах. Подобные ошибки особенно чувствительны в отчётах по SLA и в бизнес-метриках, где даже кратковременное снижение значений оказывает влияние на ключевые показатели.

Таким образом, корректная обработка отсутствующих данных является важным для архитектуры системы и логики метрик. Эффективный мониторинг – это не только сбор и отображение информации, но и осмысленная интерпретация данных. Благодаря грамотной обработке отсутствующих значений мониторинговые системы сохраняют устойчивость к ошибкам, обеспечивают достоверность визуализации и снижают риск пропуска критических событий. Применение функций PromQL и MetricsQL требует предварительного внимательного изучения метрик и их бизнес-значению в конкретной системе [5].

Список литературы:

1. Антиферов Н. PromQL: как вернуть 0 вместо «no data» [Электронный ресурс] // Medium. – 2020. – Режим доступа: <https://nklya.medium.com/promql-how-to-return-0-instead-of-no-data-9e49f7cc80d> (дата обращения 23.07.2025)

2. Volz J. Dealing with Missing Time Series in Prometheus [Электронный ресурс] // PromLabs. – 13 сентября 2023. – Режим доступа: https://promlabs.com/blog/2023/09/13/dealing-with-missing-time-series-in-prometheus/?utm_source=chatgpt.com (дата обращения 25.07.2025)
3. Prometheus сообщество. Trigger an alarm when no data received or no more data received [Электронный ресурс] // Discuss Prometheus. – 6 ноября 2023. – Режим доступа: <https://discuss.prometheus.io/t/trigger-an-alarm-when-no-data-received-or-no-more-data-received/1989/1> (дата обращения 24.07.2025)
4. Хавроненко Р. VictoriaMetrics: соответствие PromQL [Электронный ресурс] // Medium. – 2021. – Режим доступа: <https://medium.com/@romanhavronenko/victoriametrics-promql-compliance-d4318203f51e> (дата обращения 23.07.2025)
5. Service Level Agreement – соглашение об уровне обслуживания, заключаемое между поставщиком услуг и клиентом.