DOI 10.37882/2223-2966.2025.02.07

АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМЫ МОНИТОРИНГА И ОПОВЕЩЕНИЙ

ANALYSIS OF THE EFFICIENCY OF MONITORING AND NOTIFICATION SYSTEMS

B. Goryachkin
L. Perlin

Summary. *Problem statement*. Modern monitoring and notification systems face a growing volume of collected metrics. The number of devices, and consequently the amount of data in such systems, has increased significantly, especially with the rise of the Internet of Things (IoT). The number of monitored objects has grown exponentially. Operators in these systems must track an increasing number of potential notifications, which increases the cognitive load on the operator. In the current context, a monitoring system must not only process large volumes of data quickly but also present it in an intuitive and comprehensible form to reduce the user's cognitive burden.

Purpose. The primary goal of this research is to analyze the processes within monitoring and notification systems, including information flows, load, performance, timing, and frequency of processes, with attention to the cognitive load on human operators and the characteristics of the information delivered to them.

Results. Information flows in monitoring systems and their subprocesses have been analyzed. A test environment was built to simulate the increasing number of monitored objects and notifications. The experiments demonstrated that sending notifications exclusively upon state changes reduces redundancy in notifications and aligns them with the information entropy curve, representing the amount of information. Adjusting notification conditions toward less equiprobable triggers significantly reduces the number of necessary notifications. Introducing a waiting period for condition checks reduces the operator's cognitive load but increases the system's reaction time. As the number of monitored objects grows, optimizing the data collection frequency minimizes overall system delays in detecting and sending notifications while reducing the operator's workload associated with processing and extrapolating outdated alerts.

Practical significance. The described criteria and methods for ensuring effective settings have practical significance for configuring the frequency of monitoring system subprocesses, designing notification conditions, and implementing mechanisms for their logging and delivery. These findings are applicable to monitoring systems in industrial and IT infrastructure contexts.

Keywords: metric, monitoring, notification, monitoring system, notification system, cognitive load, information entropy, monitored object.

Горячкин Борис Сергеевич
кандидат технических наук, доцент;
Московский государственный технический университет им. Н.Э. Баумана
bsgor@mail.ru
Перлин Леонид Вадимович
Московский государственный технический университет им. Н.Э. Баумана
perlinlv@student.bmstu.ru

Аннотация. *Постановка проблемы*. Современные системы мониторинга и оповещений сталкиваются с ростом объема собираемых метрик. Количество устройств и как следствие данных в системах мониторинга увеличивается, с появлением интернета вещей количество объектов мониторинга кратно увеличилось. Оператору в данной системе нужно отслеживать всё большее количество возможных оповещений, что увеличивает когнитивную нагрузку на оператора. В современных условиях система мониторинга должна не только быстро обрабатывать большие данные, но и предоставлять их в понятной форме, снижая когнитивную нагрузку на пользователя.

Цель. Основной целью исследования является проанализировать процессы системы мониторинга и оповещений, информационные потоки, нагрузку, производительность, время и частоту процессов с вниманием к когнитивной нагрузке на человека-оператора, характеристикам информации, поставляемой оператору.

Результаты. Разобраны информационные потоки системы мониторинга, представлены подпроцессы. Собран тестовый стенд, на котором смоделировано увеличение количества объектов мониторинга, оповещения. В ходе эксперимента показано как отправка уведомлений исключительно об изменении состояния оповещения позволяет снизить избыточность отправляемых уведомлений, приближает уведомления к графику информационной энтропии — количеству информации, корректирование условий срабатывания оповещений в сторону менее равновероятностных приводит к значительному снижению числа необходимых уведомлений. Период ожидания для проверки условий уменьшает когнитивную нагрузку оператора, но увеличивает время реакции системы. При увеличении количества объектов мониторинга изменение частоты сбора данных позволяет повысить эффективность системы, снизить общую задержку системы на обнаружение и отправку оповещения, а также нагрузку на оператора, связанную с взаимодействием и экстраполяцией устаревших оповещений.

Практическая значимость. Описанные критерии и методы для обеспечения более эффективных настроек представляют практическую ценность при конфигурации частоты выполнения подпроцессов системы мониторинга, проектировании условий оповещений, механизмов их фиксации и отправки, что применимо для систем мониторинга промышленности, IT-инфраструктуры.

Ключевые слова: метрика, мониторинг, оповещение, когнитивная нагрузка, информационная энтропия, объект мониторинга.

Введение

Рост количества собираемых метрик в современных системах мониторинга и оповещений ставит задачу обработки большого количества информации. Современные системы мониторинга являются примером эргатической системы, в которой оператор вынужден постоянно взаимодействовать с поступающими оповещениями, которые требуют его реакции, а иногда — оперативного принятия решений. Согласно рекомендациям стандарта управления оповещениями ISA18.2[1] оператор может эффективно обработать только одно оповещение за 10 минут. Оператору в данной системе нужно отслеживать всё большее количество возможных оповещений, что увеличивает когнитивную нагрузку на оператора. Чтобы минимизировать вероятность ошибок и повысить эффективность работы оператора, система мониторинга должна не только быстро обрабатывать данные, но и предоставлять их в понятной форме, снижая когнитивную нагрузку на пользователя.

Основной целью исследования является проанализировать процессы системы мониторинга и оповещений, информационные потоки, нагрузку, производительность, время и частоту процессов с вниманием к когнитивной нагрузке на человека-оператора, характеристикам информации, поставляемой оператору.

Система мониторинга позволяет человеку постоянно отслеживать явления и процессы для последующего анализа. Сегодня люди используют современные автоматизированные системы мониторинга. Состояние наблюдаемых процессов описывается набором данных о состоянии объектов наблюдения — метрик. Количество устройств и как следствие данных в системах мониторинга увеличивается, с появлением интернета вещей количество устройств, собирающих информацию, кратно увеличилось [2]. Все чаще используют датчики интернета вещей для сбора критичных для производств показателей [3]. Каждый датчик интернета вещей IoT в рамках производства может являться объектом мониторинга. Также объектом мониторинга может являться часть IT-инфраструктуры веб-приложений, метрики могут собираться с клиентских устройств. Объект мониторинга хранит данные о своем текущем состоянии. Далее он передает данные на сервер, который хранит показатели этого объекта и фиксирует время сбора показателей, что позволяет формировать временные ряды и отслеживать необходимые процессы в динамике.

Для подобной эргатической системы важно учитывать основные свойства и характеристики информации [4], которая поступает человеку-оператору. Количество информации является важнейшей характеристикой информации, используемой в инженерной психологии [5]. Задачей технических устройств является сжатие или агрегация собранной информации, уменьшение неопределенности системы. Данную меру можно выразить с помощью информационной энтропии Шэннона [6].

Схема и информационные потоки системы мониторинга

На рис. 1 представлена схема с примером инфраструктуры системы мониторинга. Общий процесс мониторинга можно разделить на следующие периодические процессы:

1. Обнаружение объектов мониторинга — сборщик данных получает список объектов мониторинга для сбора [7].
2. Сбор данных от объектов наблюдения — сборщик данных получает метрики, описывающие состояние объектов мониторинга с помощью запросов
3. Сборщик данных производит запись полученных показателей в СУБД временных рядов [8], фиксируя время сбора, для последующего анализа данных с привязкой ко времени.
4. Экспортер данных отправляет PromQL запросы для получения информации за период времени для последующей проверки данных показателей на предмет соответствия условиям нештатной ситуации
5. Система оповещений проверяет новые значения показателей и при выполнении условий нештатного состояния формирует и отправляет оповещения

Все подпроцессы в данной системе пронумерованы на схеме (рис. 1).

Получение сборщиком данных информации об актуальных объектах мониторинга

В данном процессе сборщик данных опрашивает систему обнаружения [7] об объектах мониторинга, данные с которых необходимо собрать. Сбор данных происходит с определенной частотой, отправляется HTTP GET запрос и сборщик данных получает список источников в виде массива сетевых адресов. Данный процесс происходит с определенной частотой $\nu_{\text{обнаружения}}$, например 1 раз в 15 секунд. В данном процессе определяется количество объектов мониторинга $N_{\text{объектов мониторинга}}$, далее сборщик данных в последующем будет опрашивать объекты мониторинга исходя из этого списка (рис. 2).

Сбор показателей объектов мониторинга

Сборщик данных отправляет запросы, чтобы собрать данные о текущем состоянии наблюдаемых объектов. Сборщик посылает $N_{\text{объектов мониторинга}}$ запросов для получения состояния каждого из объектов. Выполнение данного процесса происходит с настраиваемой частотой

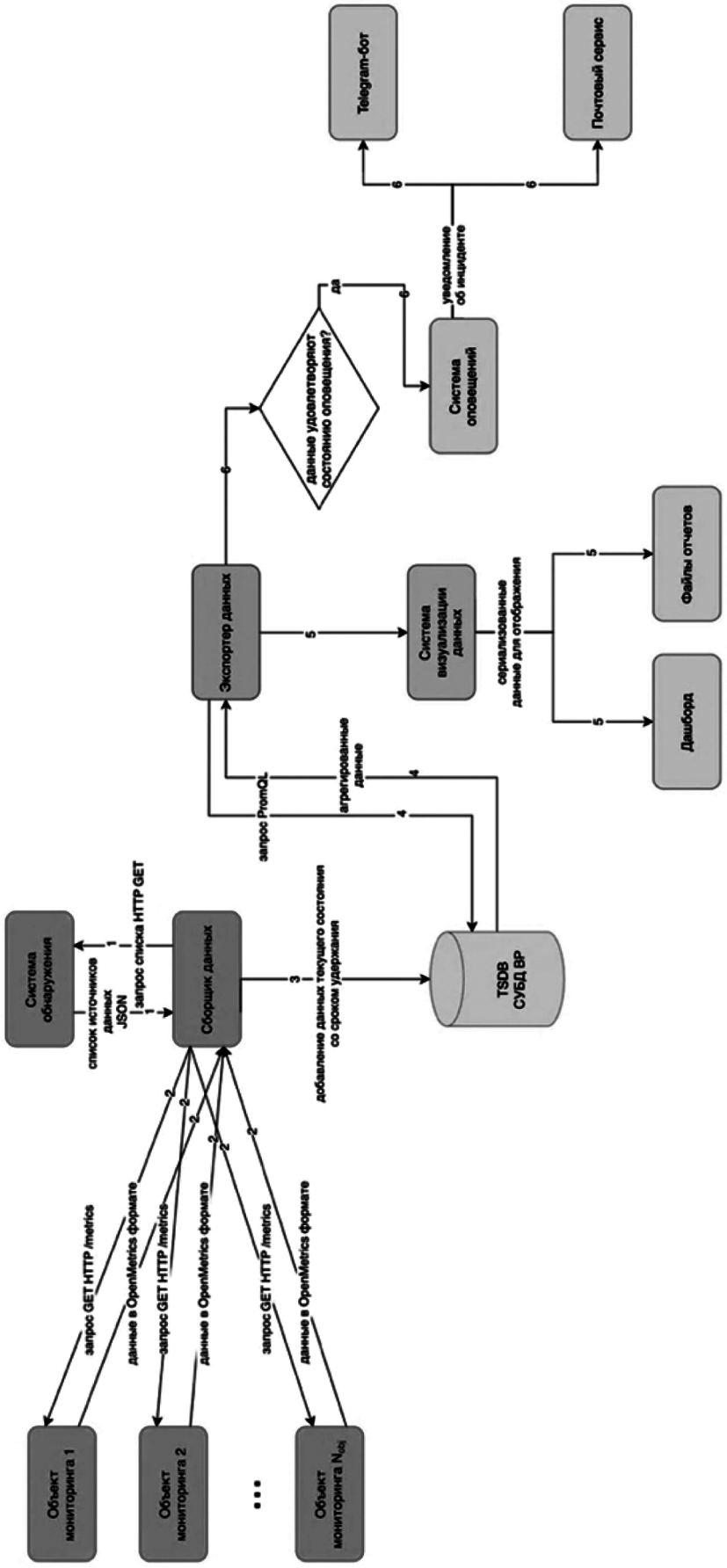


Рис. 1. Общая схема инфраструктуры системы мониторинга и оповещений

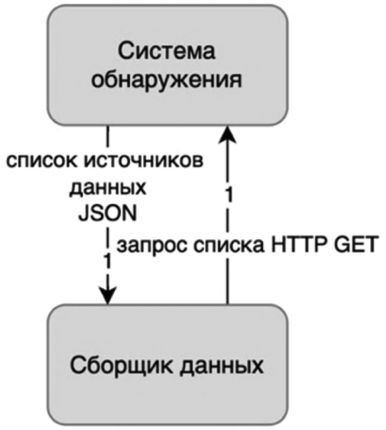


Рис. 2. Схема подпроцесса получения сборщиком данных информации об объектах мониторинга

$\nu_{сбора}$. В ответ каждый из источников отправляет данные о текущем состоянии в формате OpenMetrics [12] (рис. 3).

Количество метрик в системе

Для примера был выбран один из популярных стандартов формата метрик — OpenMetrics [12], объекты мониторинга передают информацию о своем состоянии в этом формате. Формат OpenMetrics подразумевает события. У каждого объекта есть конечное количество базовых событий, у каждого события есть метки, позволяющие более точно классифицировать событие. У каждой метки есть конечное количество возможных состояний (рис. 4).

В таком случае для одного объекта мониторинга общее количество метрик для одного объекта мониторинга можно посчитать как:

$$N_{метрик} = \sum_{i=1}^{N_{базовых\ событий}} \prod_{j=1}^{N_{меток\ i}} N_{состояний}(i,j) \quad (1)$$

где: $N_{метрик}$ — количество метрик объекта мониторинга; $N_{базовых\ событий}$ — количество возможных базовых событий объекта; $N_{меток\ i}$ — количество меток события *i;* $N_{состояний}$ — количество состояний метки *j* события *i*;

Получив конечное количество метрик для одного объекта наблюдения, можно посчитать объем данных для одного объекта наблюдения:

$$V_{данных} = \sum_{i=1}^{N_{метрик}} \left(V_{названия} + N_{меток_i} \cdot V_{метки} + V_{значения} \right) \quad (2)$$

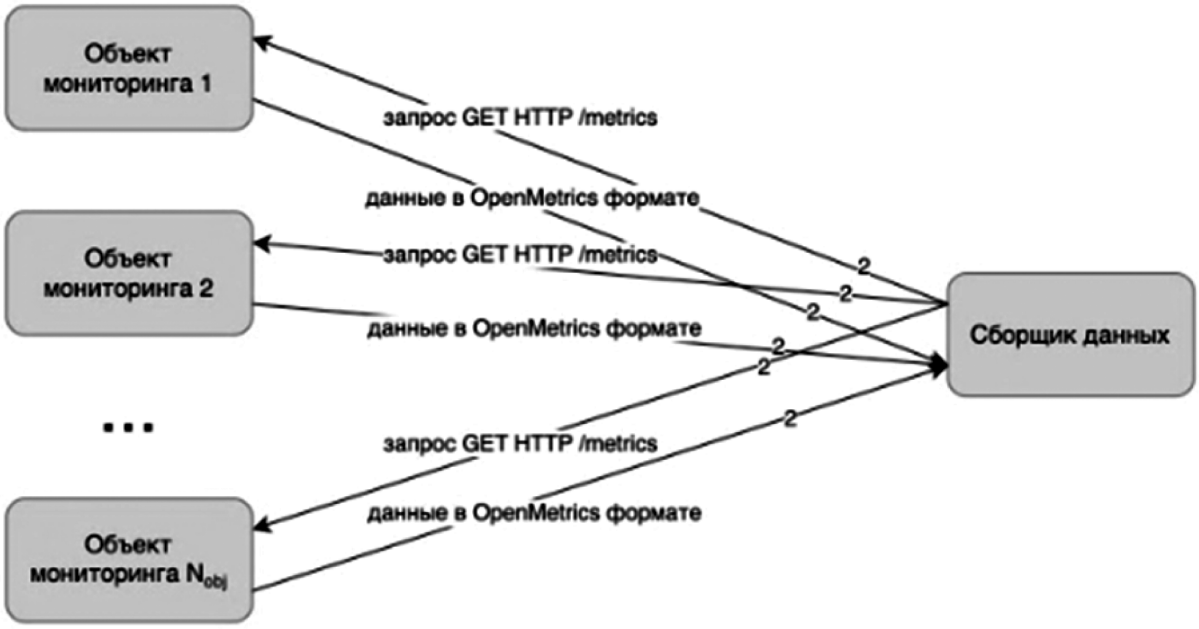


Рис. 3. Схема подпроцесса сбора показателей

```
acme_http_router_request_seconds_sum{path="/api/v1",method="GET"} 9036.32
acme_http_router_request_seconds_count{path="/api/v1",method="GET"} 807283.0
acme_http_router_request_seconds_created{path="/api/v1",method="GET"} 1605281325.0
acme_http_router_request_seconds_sum{path="/api/v2",method="POST"} 479.3
acme_http_router_request_seconds_count{path="/api/v2",method="POST"} 34.0
acme_http_router_request_seconds_created{path="/api/v2",method="POST"} 1605281325.0
```

Рис. 4. Пример данных в формате OpenMetrics

где: $V_{данных}$ — объем данных состояния одного объекта мониторинга, сериализованных в OpenMetrics формат в битах; $N_{метрик}$ — количество метрик объекта мониторинга; $V_{названия}$ — объем данных названия метрики в битах; $N_{меток\ i}$ — количество меток метрики i; $V_{метки}$ — объем данных названия метки в битах; $V_{значения}$ — объем данных закрепленного за меткой значения в битах;

Запись текущих показателей в СУБД ВР

В рассматриваемой системе используется СУБД временных рядов (TSDBMS), которая является специализированной СУБД для хранения временных рядов, каждая запись в такой базе данных связана с временной меткой. Примерами современных СУБД ВР являются Prometheus [12], Graphite [13], они оптимизированы для работы с данными временных рядов [10]. В случае с временными рядами схема данных известна заранее, итог сохраненных данных за период наблюдения есть временной ряд для каждой метрики.

Это отражается и в запросах на выборку данных, каждая из которых зависит от временных показателей, для более эффективной обработки запросов СУБД ВР строит индексы, основанные на временных метках.

Данные добавляются с большой частотой, например каждую секунду. То есть СУБД должна часто дополнять существующие ряды, однако данные приходят в упорядоченном порядке, это позволяет отказаться от постоянной передачи времени сбора, в описанном выше примере входных данных (рис. 4) видно, что данных о времени сбора нет, время добавляется автоматически, что упрощает добавление новых записей.

Особенностью также является срок удержания данных — данные автоматически хранятся определенное время, например: 15 дней в случае Prometheus. Это позволяет автоматически удалять старые неактуальные данные.

Также для СУБД ВР характерен процесс компактации — процесс объединения мелких сегментов временных рядов в более крупные, например, объединение 10 записей, сделанных каждую секунду, в одну запись, которая отражает усреднённое или агрегированное значение за 10 секунд. Эти особенности делают TSDBMS идеальным выбором для систем мониторинга, где требуется быстрая запись, чтение и анализ временных рядов.

После сбора метрик с конечных точек всех объектов наблюдения сборщик данных записывает данные в СУБД ВР, добавляются сведения о новых значениях отслеживаемых показателей. СУБД ВР сохраняет их в WAL для обеспечения надежности данных. Этот журнал временно хранит изменения, позволяя восстановить данные в случае сбоя. Затем, периодически выполняется процесс компактации, который перемещает данные из WAL в более эффективные структуры хранения на диске, оптимизируя пространство и обеспечивая быстрый доступ к метрикам. В системах мониторинга данные о метриках фиксируются как дискретные значения с временными метками, что означает запись моментальных значений метрик в определённые моменты времени, а не потоков непрерывных данных (рис. 5). Каждый временной ряд позволяет анализировать изменения значений метрик во времени.

Рис. 5. Схема подпроцесса записи данных в СУБД ВР

Агрегация данных

Данный процесс обращается к данным, записанным в итоге прохождения процесса 3. Данный процесс происходит с настраиваемой частотой опроса $\nu_{опроса}$

Экспортер данных отправляет запросы PromQL [15] для получения агрегированных данных, такой запрос представляет собой комбинацию нескольких метрик и несколько функций для их объединения.

Функции принимают метрики как входные данные и возвращают изменённые или обработанные результаты, например максимальное количество запросов за 5 минут.

Важным аспектом этих функций является работа с временными рядами данных, что подразумевает, что селекция данных осуществляется на основе временных интервалов. Тогда количество строк к обработке в запросе учитывая настраиваемую частоту сбора данных $\nu_{сбора}$.

$$N_{\text{строк}} = \nu_{\text{сбора метрики}} * T_{\text{наблюдения}} \quad (3)$$

где: $N_{\text{строк}}$ — количество строк, обрабатываемых в запросе; $\nu_{\text{сбора метрики}}$ — частота сбора метрики объекта мониторинга сборщиком данных в Гц; $T_{\text{наблюдения}}$ — временной интервал, заданный в запросе в сек.;

Назад СУБД возвращает агрегированные данные в формате JSON. В ответ на запрос СУБД отправляет данные в агрегированном формате за заданный параметрами запроса период (рис. 6).

Проверка условий и отправка оповещений

Полученные от сборщика агрегированные данные проверяются на соответствие установленным условиями для оповещения. Условия проверяются с конфигурируемой периодичностью, которую можно обозначить как $T_{\text{проверки}}$ и соответствующей частотой $\nu_{\text{проверки}}$. Данными для проверки условий являются последние данные, полученные из СУБД ВР.

Факт того, что данные удовлетворяют состоянию оповещения, добавляет неопределенность в систему, появляется информационная энтропия в связи с вероятностью срабатывания уведомления. Далее система оповещений отправляет уведомление выбранным средством оповещений (рис. 7).

Оценка времени процессов системы мониторинга

Учитывая информационные потоки системы мониторинга, можно оценить количество запросов и данных проходящих в подпроцессах, для обнаружения объектов мониторинга при их увеличении системе все равно необходимо отправить только один запрос, количество запросов необходимых для агрегации данных и проверки условий равно количеству заданных условий для оповещений, также в этих подпроцессах соединение устанавливается между двумя устройствами, что позволяет отправлять несколько запросов в рамках одного постоянного HTTP-соединения начиная с протокола HTTP 1.1 [16], что также уменьшает необходимое время прохождения данного количества запросов.

Рис. 6. Схема подпроцесса получения данных из СУБД ВР

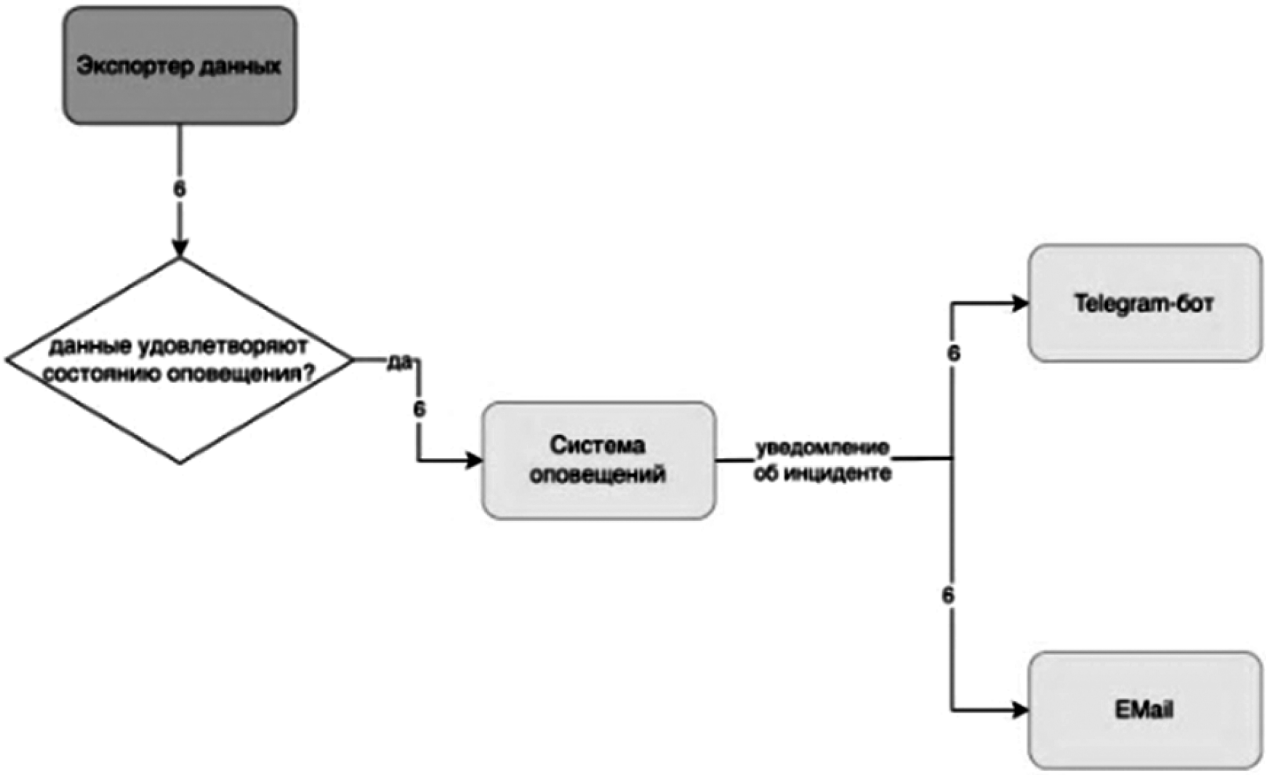


Рис. 7. Схема процесса проверки инцидентов и отправки оповещений

В случае со сбором данных от объектов мониторинга сборщик данных вынужден создавать новое соединение с каждым из объектов мониторинга, количество которых может быть крайне большим. Самым нагруженным процессом из всех является сбор данных от объектов мониторинга, здесь собираются все сырые данные и формируются для записи и для дальнейшей обработки. Сборщик посылает $N_{\text{объектов мониторинга}}$ запросов для получения состояния каждого из объектов. Учитывая объем данных, количество метрик объекта мониторинга, количество объектов мониторинга и технические ограничения сервера, можно оценить время сбора данных от объектов мониторинга используя (1), (2):

$$T_{\text{сбора}} = \sum_{i=1}^{N_{\text{объектов}}} \left(\begin{array}{l} T_{\text{задержки сети}_i} + \\ +V_{\text{данных}_i} \left(\dfrac{1}{S_{\text{сети}_i}} + \dfrac{1}{S_{\text{десериализации}}} \right) \end{array} \right) \quad (4)$$

где $N_{\text{объектов}}$ — количество объектов мониторинга; $T_{\text{задержки сети}_i}$ — задержка сети до объекта мониторинга i в сек.; $S_{\text{десериализации}}$ — скорость десериализации данных из OpenMetrics формата сборщиком данных в битах/с [17]; $S_{\text{сети}_i}$ — скорость соединения в битах/с; $V_{\text{данных}_i}$ — объем данных состояния объекта мониторинга i в битах;

Информационная энтропия в системах мониторинга и оповещений

Система мониторинга отслеживает множество показателей и генерирует уведомления при отклонениях от нормы. Эти события могут происходить с разной вероятностью. Энтропия Шеннона [9] измеряет неопределённость в распределении вероятностей событий, помогая понять, насколько предсказуемы или случайны состояния источника. Если вероятность выхода метрики за пределы нормы высока, то неопределённость (энтропия) будет низкой. Если же состояния источника менее предсказуемы, энтропия будет высокой.

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i \quad (5)$$

где p_i — вероятность состояния; N — количество возможных состояний; H — энтропия системы, представляющая количество информации в битах.

В случае с энтропией двух и более независимых источников, энтропия объединения равна сумме энтропий этих источников [9], при этом для каждого источника собственные вероятности $\sum_{i=1}^{N} p_i = 1$

$$H_{\text{общее}}(X;Y;Z;\ldots) = \\ = H(X) + H(Y) + H(Z) + \ldots = \sum_{i=1}^{n} H_i \quad (6)$$

где $H_{\text{общее}}$ — общая энтропия системы в битах; H_i — энтропия независимого источника i в битах

После сбора всех изначальных показателей от объектов мониторинга, энтропию можно выразить, используя формулы количества метрик, приведенные ранее. Каждая из метрик содержит свое состояние со своими вероятностями нахождения в этом состоянии, состоянием метрики является комбинация меток и их значений. Если у метрики есть две метки, каждая из которых может быть в одном из двух состояний, то всего по у метрики есть 4 возможных состояния. Тогда энтропия одной метрики:

$$H_{\text{метрики}}(x) = -\sum_{i=1}^{N_{\text{состояний}}} P(i)\log_2\left(P(i)\right) \quad (7)$$

где p_i — вероятность нахождения метрики в состоянии i; $N_{\text{состояний}}$ — количество возможных состояний метрики; $H_{\text{метрики}}$ — энтропия метрики, представляющая количество информации в битах.

В случае всех метрик системы энтропия в этом случае показывает общую неопределённость в данных до агрегации метрик, когда каждая метрика и её состояния считаются отдельно. Учитывая, что события независимы, то исходя из следствий (6) теоремы Шеннона [9] общая энтропия будет равна сумме энтропий метрик, каждая метрика является источником энтропии, то есть чем больше независимых источников энтропии — метрик, тем выше общая энтропия системы в момент сбора сырых данных. Учитывая (6) и (7), энтропия состояния системы:

$$H_{\text{состояния}}(X) = \sum_{i=1}^{M_{\text{метрик}}} H_{\text{метрики i}}\left(P(i)\right) = \\ = -\sum_{i=1}^{M_{\text{метрик}}} \sum_{j=1}^{N_{\text{состояний i}}} P(i,j)\log_2\left(P(i,j)\right) \quad (8)$$

где p_i — вероятность состояния; $M_{\text{метрик}}$— количество метрик всей системы. $N_{\text{состояний i}}$— количество возможных состояний; $H_{\text{состояния}}$ — энтропия системы, представляющая количество информации в битах.

Информационную энтропию оповещений, то есть количество информации, обрабатываемой оператором, можно оценить, зная количество проверяемых условий и вероятности их срабатывания. Каждое оповещений представляет собой независимое событие. После агрегации проверяются условия оповещений с частотой проверки $\nu_{\text{проверки}}$, с использованием актуальных на момент проверки значений, полученных в ходе подпроцесса экспорта данных, то есть результаты PromQL запросов при использовании Prometheus, итогом проверки может быть либо выполнение, либо невыполнение условия, то

есть два исхода для каждого из событий, количество информации для одного оповещения можно выразить как:

$$H_{\text{оповещения}}(x) = -\sum_{j=1}^{2} P(j)\log_2\left(P(j)\right) \quad (9)$$

Имея $N_{\text{условий}}$ событий, учитывая (6) и (9):

$$H_{\text{оповещений}}(x) = \sum_{i=1}^{N_{\text{оповещений}}} H_{\text{оповещения i}}\left(P(i)\right) = \\ = -\sum_{i=1}^{N_{\text{оповещений}}} \sum_{j=1}^{2} P(i,j)\log_2\left(P(i,j)\right) \quad (10)$$

где p_i — вероятность состояния оповещения; $N_{\text{оповещений}}$ — количество оповещения; $H_{\text{оповещений}}$ — энтропия оповещений системы, представляющая количество информации в битах.

В таком случае уменьшение неопределенности системой мониторинга и оповещений можно определить как разность между изначальной неопределенностью системы и количеством информации оповещений, поставляемых оператору:

$$\Delta H(X,Y) = H_{\text{состояния}}(X) - H_{\text{оповещений}}(Y) \quad (11)$$

Посмотрим количество энтропии одного условия оповещения используя формулу (8) с вероятностью P_1 — срабатывания уведомления и вероятностью P_2, эти вероятности комплиментарны так как условие оповещение либо выполняется, либо не выполняется. В левой части графика при низком, P_1 — энтропия минимальна, если бы оператор стабильно получал оповещения о текущем состоянии, они повторялись бы, и система почти всегда сообщала бы о некритическом состоянии. В правой части графика при высоком P_1 энтропия также минимальная, система стабильно в неисправном состоянии. Максимум количества информации оповещение достигает в том случае, когда вероятности P_1 и P_2 равны. Итогом каждой проверки подобного условия являются разные сообщения в каждой отдельной проверке, система постоянно переходит из критического в некритическое состояние и оператору тяжелее отслеживать подобные оповещения. Таким образом наиболее когнитивно нагружающими оповещениями являются оповещения, создаваемые условиями проверки с более близкими вероятностями срабатывания и несрабатывания условия. (рис. 8)

Рис. 8. Энтропия проверки одного условия оповещения

Период ожидания и состояния оповещений

Таким образом при составлении условий оповещений необходимо стремиться к наиболее не равновероятностным исходам проверки. Механизмом снижения количества информации может быть добавление периода ожидания $T_{\text{ожидания}}$, который отправлял бы оповещение, только при выполнении условия на протяжении всего интервала оценки для исключения одноразовых превышений порогового значения. В таком случае необходимое количество срабатывающих проверок $N_{\text{проверок}}$ при известной частоте проверки условия $\nu_{\text{проверки}}$:

$$N_{\text{проверок}} = \nu_{\text{проверки}} \cdot T_{\text{ожидания}} \quad (12)$$

где: $N_{\text{проверок}}$ — количество необходимых успешных подряд проверок; $\nu_{\text{проверки}}$ — частота проверки оповещения в Гц; $T_{\text{ожидания}}$ — период ожидания в сек.

Когда система оповещений выполняет запросы к источникам данных и сравнивает данные с условиями срабатывания, если условия выполняются, инцидент регистрируется как «сработавшее оповещение». Однако до этого данные должны пройти через стадию, называемую «ожидание», когда метрики начинают удовлетворять условиям, но оповещение еще не срабатывает (рис. 9).

Период ожидания — это время, в течение которого условие должно быть выполнено непрерывно. Если условие сохраняется в течение этого периода, система переходит к отправке уведомления. То есть для срабатывания одного условия, проверяемого каждую секунду с временем оценки 5 секунд, будет необходимо 5 сработавших проверок для отправки уведомления, что снижает вероятность срабатывания уведомления и, как следствие, необходимое количество информации для передачи оповещения, поставляемого оператору. Необходимо учитывать, что добавление периода ожидания увеличивает минимальное время, необходимое для отправки на время ожидания.

Снизить количество необходимых уведомлений можно, отправляя уведомления оператору только в случаях прохождения периода проверки, то есть подтверждения критического состояния показателя и нормализации состояния — возвращения состояния в нормальное состояние, что в свою очередь снижает избыточность повторяемой информации.

Моделирование объектов наблюдения системы мониторинга и тестовый стенд

Для моделирования объекта наблюдения было создано серверное приложение REST API [18] на Golang [20] с методами для добавления и удаления показателей в процессе эксперимента.

Был создан набор со 100000 метрик. В проведении эксперимента используется варьируемое количество объектов наблюдения с различным количеством событий для каждого объекта, при этом каждый показатель может иметь разное количество меток, формирующих метрику и описывающих более детально текущее состояние. Выбранная инфраструктура позволяет контролировать количество обрабатываемых метрик (рис. 10).

Для моделирования подобной системы для вероятностей нахождения показателя в определенном состоянии используется распределение Пуассона и интенсивность возникновения событий. Это распределение хорошо подходит для моделирования в системе мониторинга, где важно учитывать частоту событий, например возникновения ошибок, запросов, изменений показателей и подсчитывать их количество [19].

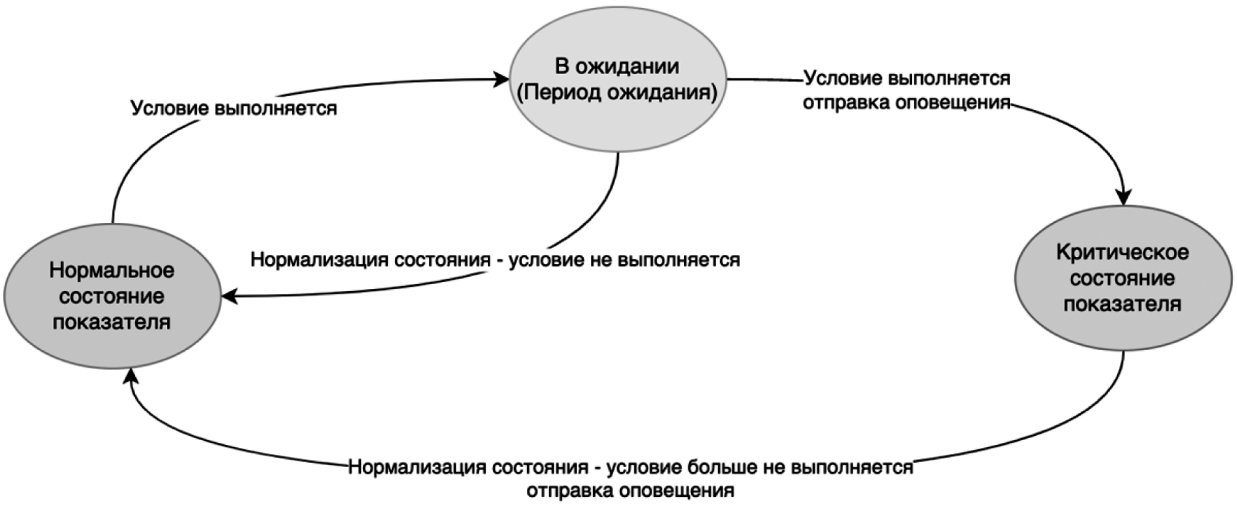


Рис. 9. Граф состояний оповещения при использовании периода ожидания

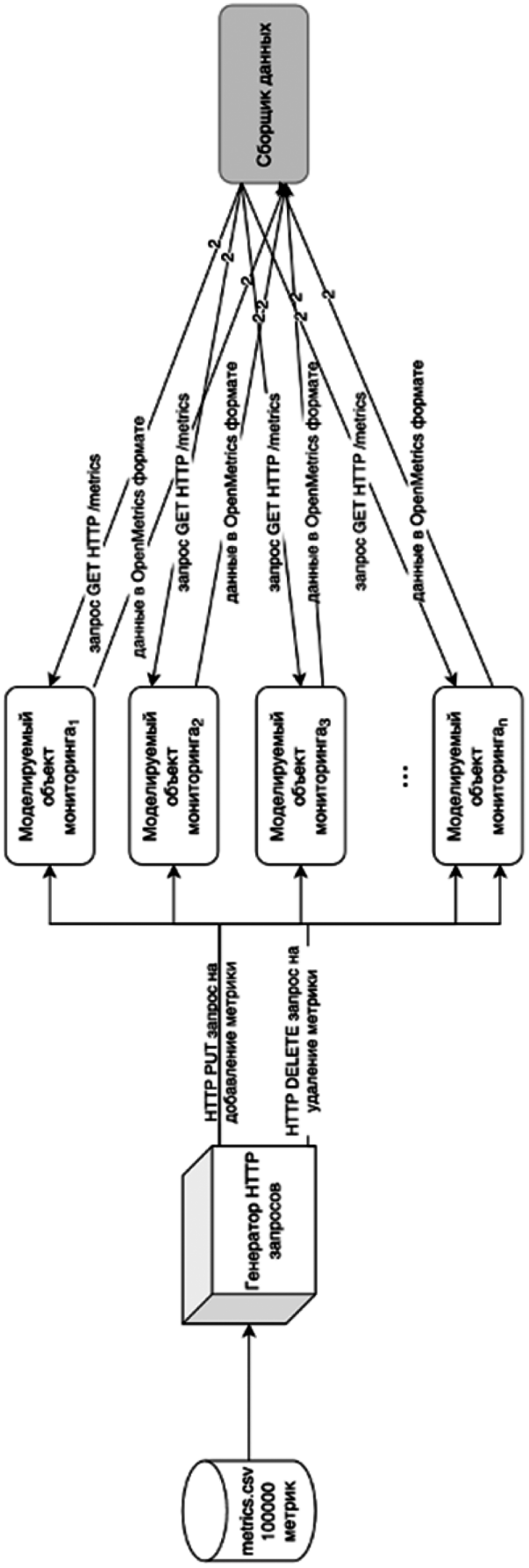


Рис. 10. Схема тестового стенда

Таким образом можно зная интенсивность возникновения события получить вероятности и подсчитать информационную энтропию системы и сопоставить ее с количеством уведомлений для проверки.

Экспериментальное исследование времени сбора в зависимости от количества метрик

В данном эксперименте использовался описанный выше тестовый стенд. Частота сбора $\nu_{сбора}$= 0,05 Гц, $T_{сбора} = 20c$. При увеличении количества метрик (рис. 11б) увеличивается время сбора (рис. 11а), а при остановке добавления новых показателей время сбора выходит на плато.

Это объясняется тем, что при росте количества объектов мониторинга, базовых событий и, как следствие, общего количества метрик, увеличивается объем передаваемых данных $V_{данных_i}$, из формулы для расчета $T_{сбора}$ (4). При 100000 показателей при выходе на плато среднее время опроса всех объектов мониторинга составило 2,3 секунды. Максимальное фактическое значение $T_{сбора}$ составило 2,8 секунд. Используя данную информацию, следует уменьшить настраиваемое $T_{сбора}$.

Экспериментальное исследование количества оповещений при различных правилах отправки

На описанном выше тестовом стенде были заданы оповещения с различными вероятностями нахождения в критическом состоянии (прохождения условия оповещения). Для демонстрации снижения количества информации и информационной нагрузки на человека-оператора было выбрано количество оповещений. На тестовым стенде было смоделировано пять условий проверки состояния с различными пороговыми значениями и соответствующими вероятностями определения состояния оповещения: очень часто, часто, равновероятностно, редко, очень редко. Условия оповещений проверялись один раз за одну минуту на протяжении шести часов проведения эксперимента. Для отправки уведомлений использовался описанный выше метод отправки исключительно изменений состояний, то есть в каждом из случаев уведомление сообщает оператору о переходе в критическое состояние, либо нормализацию состояния оповещения. Результаты представлены в табл. 1.

Таблица 1.

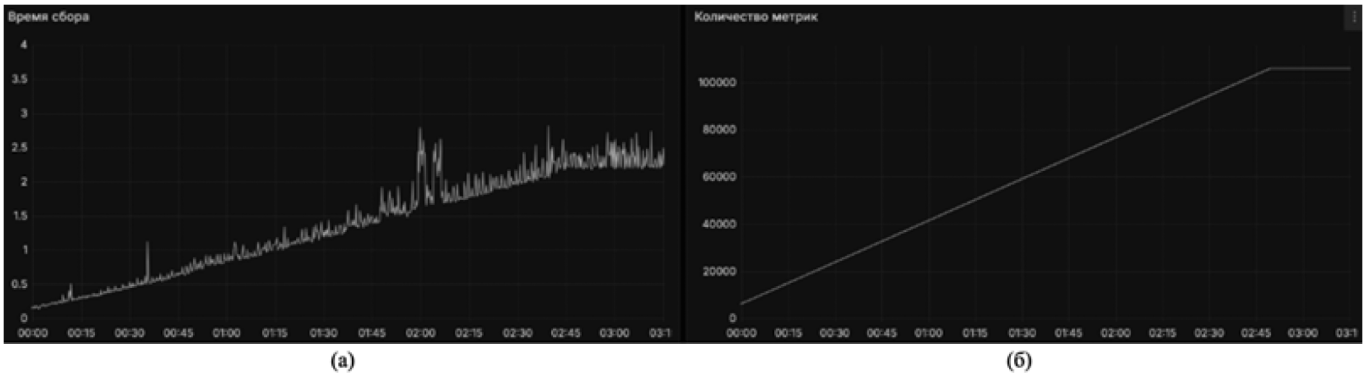
Результаты сравнения условий проверок с различными вероятностями срабатывания при использовании отправки уведомлений о изменении состояний оповещений

Вероятность срабатывания (в %)	Инф. энтропия	Количество зафиксированных критических состояний	Количество зафиксированных некритических состояний	Количество уведомлений о изменении состояния оповещения
1 %	0,08 бит	1	359	2
99 %	0,08 бит	354	6	4
10 %	0,46 бит	37	323	63
90 %	0,46 бит	321	39	71
50 %	1 бит	164	196	280
$H_{оповещений}$ при одном цикле проверки (10)	2,08 бит			

Оценка полученных результатов

Исходя из полученных практически в ходе эксперимента с зависимостью времени сбора данных от количества метрик результатов можно установить, что при увеличении числа метрик в системе наблюдается линейный рост времени сбора данных. Это связано с увеличением объема передаваемых данных $V_{данных}$, что также подтверждается теоретическими расчётами по формуле $T_{сбора}$ (4).

Для оценки практических результатов, полученных в ходе экспериментального исследования количества



(а) (б)

Рис. 11. а) Временной ряд фактического времени сбора б) Временной ряд количества собираемых метрик

оповещений при различных правилах отправки, сопоставим данные таблицы 1 с полученным ранее теоретическим графиком энтропии проверки одного условия. Отобразим столбчатую диаграмму зависимости количества уведомлений от вероятностей срабатывания по итогам измеренных значений, рассчитанную энтропию оповещений (рис. 12).

При наложении графика энтропии проверки одного условия видно, что при отправке оповещения в случае изменения состояния оповещения, количество оповещений при разных вероятностях повторяет график энтропии, столбцы с количеством уведомлений симметричны относительно 50 % вероятности. Собранные данные показывают, что наименьшее количество уведомлений отправляется при вероятностях срабатывания условия 99 % и 1 %. Наибольшее количество уведомлений связано с условиями, имеющими вероятность срабатывания 50 %, что сопровождается чрезвычайно высокой энтропией. Учитывая, что энтропия выражает количество информации, это можно интерпретировать как то, что именно в таком случае система создает наименьшее количество избыточных уведомлений и передает фактическое количество информации о происходящих событиях.

При отправке при каждом фиксации критического состояния максимальное количество оповещений наблюдалось бы при 99 % вероятности, так как было зафиксировано 354 критических состояний, однако если посчитать часть энтропии события появления критического оповещения при данной вероятности, то это будет всего $-0,99\log_2 0,99 = 0,01\text{бит}$, сообщение о критическом состоянии почти не несет информации, переходя к оповещениям — оповещения каждый раз сообщали бы одинаковую информацию, в данном случае наиболее важным событием, несущем не избыточную информацию, были бы оповещения о редких (1 % срабатывания) некритических состоянии с энтропией события $-0,01\log_2 0,01 = 0,07\text{бит}$ в данном случае. Однако если бы система отправляла бы сообщения при каждой фиксации некритического состояния, которых было зафиксировано 6, также наблюдалось бы как минимум одно повторное избыточное оповещения, учитывая, что изменений состояний было 4, как минимум один раз система два раза находилась в некритическом состоянии подряд, что можно было бы передать одним оповещением.

При вероятности срабатывания 90 % условие может выполняться несколько раз подряд, в таком случае за несколько периодов создаётся только одно уведомление, так как состояние не меняется, что объясняет несоответствие между количеством критических состояний, выходов из них и количеством отправленных уведомлений. Для равновероятных условий (50 %) число уведомлений превышает их суммарное количество для всех остальных условий. Ужесточение условий проверки в случае вероятностей при переходе от 90 % до 99 % позволило сократить количество уведомлений более чем в 10 раз.

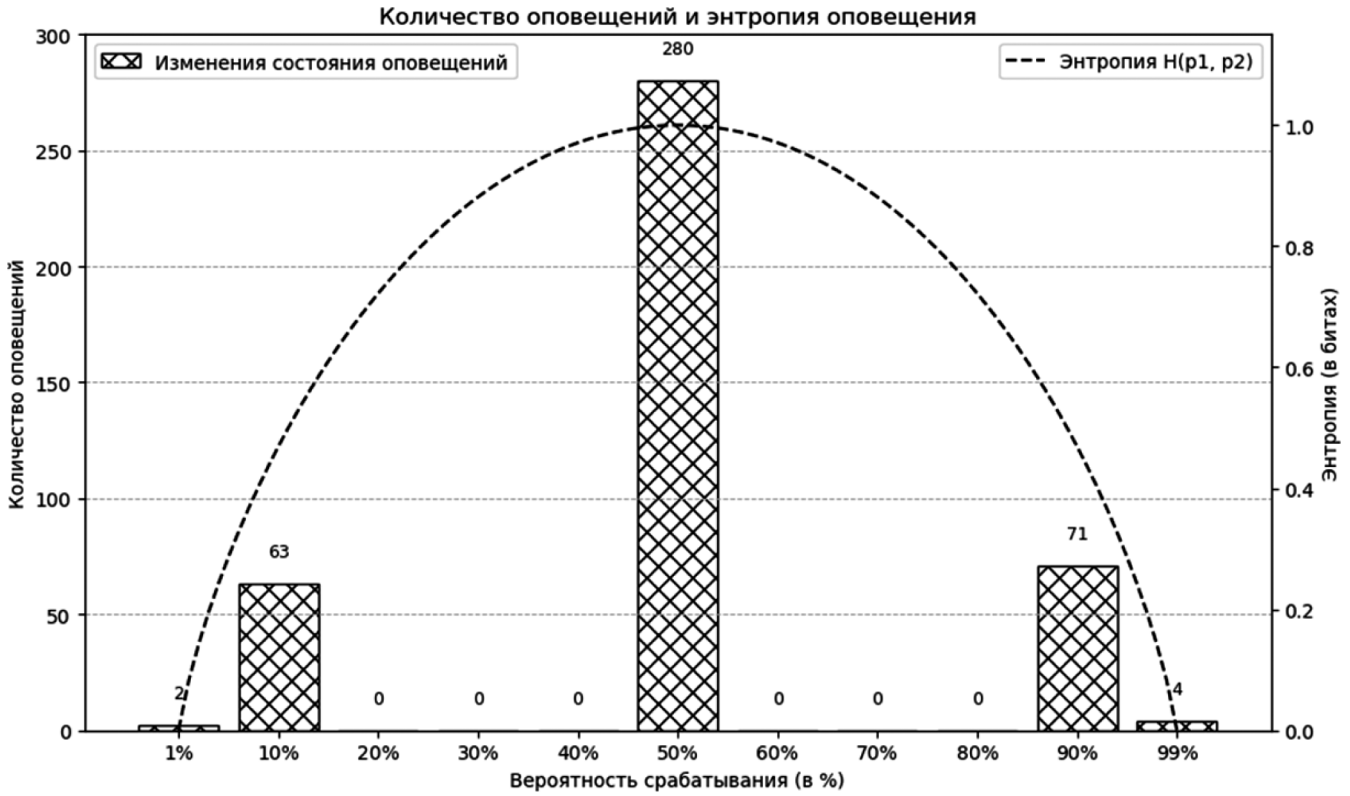


Рис. 12. Столбчатая диаграмма количества отправленных оповещений в ходе эксперимента

Заключение

В ходе исследования были рассмотрены информационные потоки и процессы системы мониторинга и оповещений, произведена теоретическая оценка количества метрик в системе, их влияние на время сбора, оценка когнитивной нагрузки на человека-оператора с использованием информационной энтропии.

Был создан тестовый стенд, на котором были проведены экспериментальные исследования, результаты показывают, что использование механизма состояния оповещений снижает избыточность, а в совокупности с использованием менее равновероятностных условий проверки, позволяет системе более эффективно уменьшать когнитивную нагрузку на человека-оператора и повышать общую эффективность системы. Использование периода ожидания для проверки условий также позволяет снизить возможную информационную энтропию, однако компромиссом является увеличение времени фиксации критического состояния, это может быть полезно в случае менее срочных оповещений.

При увеличении количества метрик узким местом, влияющим на время, становится периодический опрос объектов мониторинга, конфигурация частоты $\nu_{\text{сбора}}$ с использованием рассмотренного в ходе исследования $T_{\text{сбора}}$ позволяет повысить эффективность системы, снижая общую задержку системы на обнаружение и отправку оповещения.

Описанные методы, критерии для повышения эффективности представляют практическую ценность при конфигурации частоты выполнения подпроцессов системы мониторинга, проектировании условий оповещений, механизмов их фиксации и отправки.

ЛИТЕРАТУРА

1. International Society of Automation. ISA 18.2-2016 Management of Alarm Systems for the Process Industries. ISA, 2016.
2. Iot-Analytics.com. Statistics on the growth of IoT devices. URL: https://iot-analytics.com/number-connected-iot-devices/. (Дата обращения: 24.10.2024.)
3. Manyika J., «Big data: The next frontier for innovation, competition, and productivity.» McKinsey Global Institute Report, 2011.] URL: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/big-data-the-next-frontier-for-innovation (дата обращения: 24.10.2024)
4. Ефимов А.Н. Информация: ценность, старение, рассеяние. М.: Знание, 1978.
5. Душков Б.А. Основы инженерной психологии: учебник для студентов вузов / Б.А. Душков, А.В. Королев, Б.А. Смирнов. — Москва: Академический проект, 2020. — 574 с. — ISBN 978-5-8291-2717-6.
6. Lucchese A., Mossa G., Mummolo G., Sisto F.P. (2020). A Shannon entropy graph-based model to evaluate the operator mental workload involved in procedure-guided tasks. Proceedings of the 32nd European Modeling & Simulation Symposium (EMSS 2020), pp. 103–111. DOI: https://doi.org/10.46354/i3m.2020.emss.014
7. Service Discovery в распределенных системах на примере Consul. / Александр Сигачев URL: https://habr.com/ru/articles/487706/ (дата обращения: 25.10.2024)
8. Mueen Abdullah, Keogh Eamonn, Zhu Qiang, Cash Sydney, Westover Brandon (2009). "Exact Discovery of Time Series Motifs". University of California, Riverside. 2009: 473–484. doi:10.1137/1.9781611972795.41
9. Shannon C. (1948) A mathematical theory of communication. Bell System Technical Journal 27 379–423.
10. Time Series Compression Algorithms Explained. / Timescale. URL: https://www.timescale.com/blog/time-series-compression-algorithms-explained/ (дата обращения: 10.10.2024)
11. Сбор и обработка больших данных в системах мониторинга информационно-телекоммуникационных сетей средствами технологий Hadoop. / Авторы: Будко Н.П., Васильев Н.В., Груздев А.А. URL: https://cyberleninka.ru/article/n/sbor-i-obrabotka-bolshih-dannyh-v-sistemah-monitoringa-informatsionno-telekommunikatsionnyh-setey-sredstvami-tehnologii-hadoop/viewer (Дата обращения: 10.10.2024)
12. OpenMetrics. / OpenMetrics Community. URL: https://openmetrics.io (Дата обращения: 10.10.2024)
13. Prometheus. / Prometheus Community. URL: https://prometheus.io (Дата обращения: 10.10.2024)
14. Graphite. / Graphite Community. URL: https://graphiteapp.org (Дата обращения: 24.10.2024)
15. Basics of Querying in Prometheus. / Prometheus Community. URL: https://prometheus.io/docs/prometheus/latest/querying/basics/ (дата обращения: 10.10.2024)
16. HTTP Persistent Connection Establishment, Management, and Termination. / TCP/IP Guide. URL: http://www.tcpipguide.com/free/t_HTTPPersistentConnectionEstablishmentManagementand.htm (Дата обращения: 10.10.2024)
17. Sonic. / ByteDance. URL: https://github.com/bytedance/sonic (дата обращения: 10.10.2024)
18. Representational State Transfer (REST) Architectural Style. / Roy T. Fielding. URL: https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (Дата обращения: 10.10.2024)
19. A First Course in Probability. / Sheldon Ross. URL: https://www.math.wm.edu/~leemis/2004iie.pdf (Дата обращения: 25.10.2024)
20. Go Programming Language. / Go Dev. URL: https://go.dev (Дата обращения: 10.10.2024)

Журнал «Современная наука: актуальные проблемы теории и практики»