Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 20Б.11-мм

Интеграция микросервисов в систему логов в формате OpenTelemetry

Бакаев Евгений Владимирович

Отчёт по производственной практике

Научный руководитель:
ст. преп. С.Ю. Сартасов

Консультант:
руководитель групп разработки DataCamp и технологий индексации,
ООО «Яндекс.Технологии», Хвастунов А. П.

Санкт-Петербург
2023

Оглавление

Введение **3**

1. Постановка задачи **5**

2. Обзор инструментов для поставки и агрегации логов **6**

2.1. OpenTelemetry 6

2.2. Unified Agent и Logshatter 8

2.3. ClickHouse 8

2.4. Grafana и Yandex Monitoring 8

3. Ход работы **10**

3.1. Сбор форматов логов внутри DataCamp 10

3.2. Описание способов поставки логов 11

3.3. Переход на новую систему логов 12

3.3.1. Настройка Unified Agent 12

3.3.2. Подключение логов `multi_offers_tasks` 13

3.3.3. Подключение остальных микросервисов 13

Заключение **15**

Список литературы **16**

Введение

Десятки миллионов пользователей используют множество сервисов компании Яндекс ежемесячно. DataCamp – система офферных данных e-com партнеров и платформа интеграции между компонентами Яндекс.Маркета, Товарной вертикали, Яндекс.Директа и Яндекс.Еды. Система выполняет получение офферов от B2B партнеров, подготовку офферов для публикации и поставку в B2C компоненты Яндекс.Маркета. Для круглосуточного доступа к сервису и своевременного решения инцидентов требуются дежурные, а так же инструменты для быстрого реагирования и отлаживания: мониторинги, оповещения и логи. С помощью этих инструментов можно следить за деградацией производительности сервиса и локализовать проблему.

Логирование является важной частью любых информационных систем. В системе, состоящей из множества микросервисов, с помощью логов можно отслеживать статус сервиса, полноту данных, следить за производительностью и использовать для отладки. Сохранять логи можно нескольми способами: писать в файл, терминал, отправлять поверх HTTP или gRPC. Логи, без возможности их анализировать и искать необходимые записи, не несут в себе никакой ценности, поэтому необходимо иметь инструмент для их чтения, визуализации и фильтрации. Для чтения лога из файла достаточно иметь доступ по ssh, но такой подход имеет множество проблем, например, с ростом количества микросервисов – растет и количество ручной работы, а также разные микросервисы могут писать логи в разных форматах. Помимо этого, с увеличением нагрузки на микросервис растет и объем логов, в которых нужно найти необходимую информацию.

Для решения описанных выше проблем в Яндексе был создан сервис «Логвьюшница», в который можно отправлять логи в согласованном формате, а потом агрегировать по нужным ключам и временным промежуткам на вебстранице и смотреть на графики, построенные с помощью Grafana [4]. В бекенде Логвьюшницы используется ClickHouse [1] – высокопроизводительная аналитическая СУБД с открытым исход-

ным кодом. Система позволяет отвечать на аналитические запросы по обновляемым в реальном времени данным и способна масштабироваться до десятков триллионов записей и петабайтов данных [2].

Поставку логов в Логвьюшницу осуществляет Unified Agent [9], который запускается в контейнере, читает необходимый лог, применяет необходимые фильтры и в зависимости от выбранного плагина пишет либо в файл, либо в Logbroker(сервис для передачи упорядоченных потоков данных), либо напрямую в backend Логвьюшницы.

Backend Логвьюшницы принимает логи в виде сериализованного protobuf в OpenTelemetry формате или в виде json, мимикрирующий под protobuf.

Работа состоит из обзора способов поставки логов внутри Яндекса, инструментов для их агрегации и визуалиции, а также пайплайн для перехода с одной системы поставки на другую.

1. Постановка задачи

Цель данной работы – реализовать прямую поставку логов в ClickHouse для всех микросервисов внутри DataCamp. Для достижения цели были поставлены следующие задачи:

- Выполнить обзор существующих инструментов для поставки и агрегации логов
- Выполнить обзор всех форматов логов, использующихся в DataCamp
- Поддержать поставку логов для нового микросервиса
- Перенести логи с системы поставки через Logbroker на прямую
- Протестировать доставку логов

2. Обзор инструментов для поставки и агрегации логов

2.1. OpenTelemetry

OpenTelemetry [5] – это проект с открытым исходным кодом, который обеспечивает сбор, обработку и анализ данных телеметрии для современных приложений. Он предоставляет единый набор библиотек, агентов и инструментов для сбора и отправки метрик, трассировок и других телеметрических данных в различных форматах. Не является официальным стандартом, но используется во многих известных компаниях: Google, Amazon, Yandex.

LogRecord – сообщение в моделе данных логов OpenTelemetry. Поля LogRecord включают в себя:

- Время события – содержит метку времени, когда произошло данное событие
- Уровень – определяет важность события, например: “info”, “warn”, “error”
- Тело сообщения – данные, которые были залогированы приложением
- Теги – дополнительные метаданные, связанные с событием, например, название сервиса, название приложения, уникальный идентификатор процесса и другие

OpenTelemetry Collector позволяет получать, обрабатывать и экспортировать данные телеметрии и избавляет от необходимости собственной реализации сборщиков логов. Эскпортировать данные может быть полезно для дальнейшего анализа и визуализации данных. Существует два основных способа работы с Collector.

1. Поставка логов в Collector напрямую из приложения с помощью Open Telemetry Protocol (Рис. 1). Open Telemetry Protocol (OTLP)

описывает механизм кодирования, транспортировки и доставки телеметрических данных между источниками телеметрии, может работать поверх gRPC и HTTP 1.1.

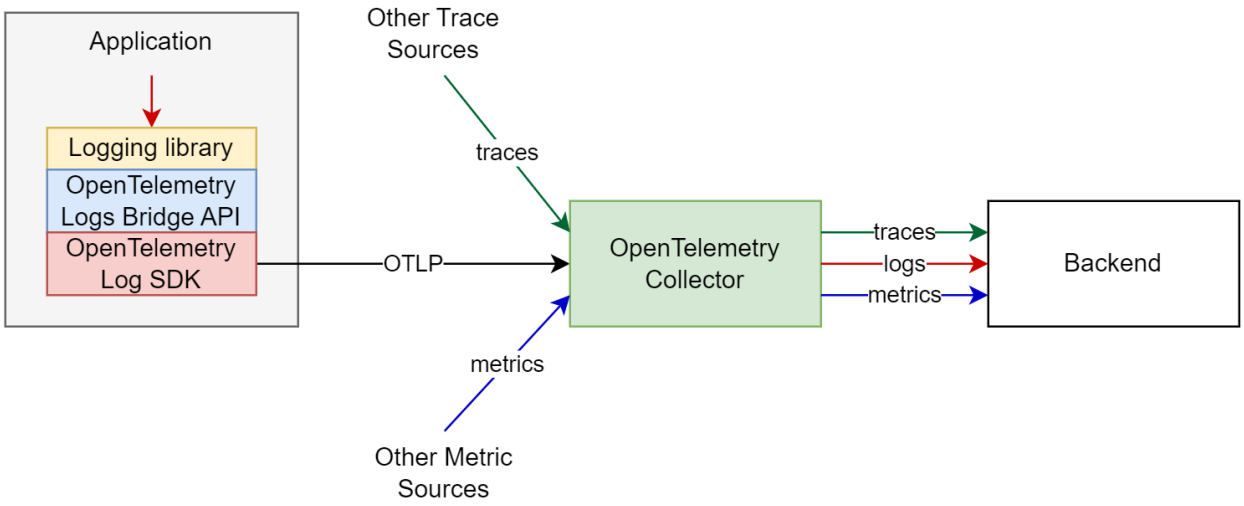

Рис. 1: Схема прямой поставки логов [7]

2. Поставка логов в Collector через стороннее приложение, например FluentBit [3], которое читает и пересылает логи, записанные основным приложением. (Рис. 2). В данной работе вместо FluentBit используется внутренняя разработка Яндекса – Unified Agent.

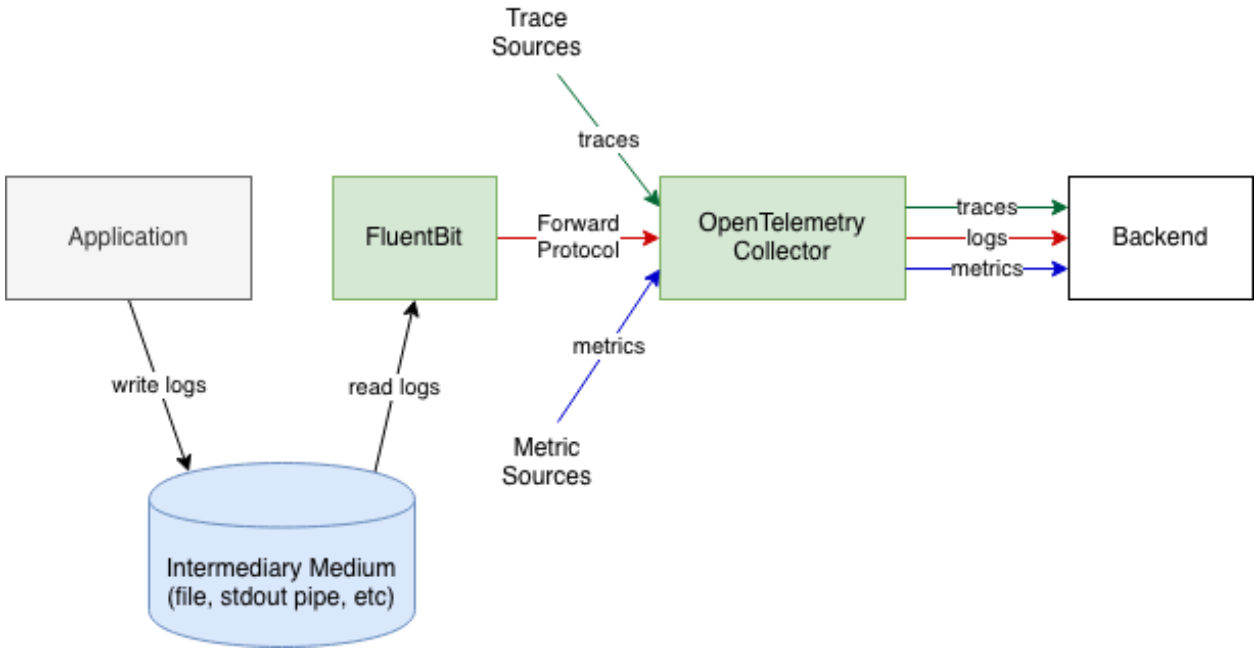

Рис. 2: Схема поставки логов с помощью стороннего приложения [8]

Лучшей практикой [6] для новых приложений считается отправка логов и трассировок напрямую из приложения в Collector.

2.2. Unified Agent и Logshatter

Unified Agent (UA) – это агент, специализирующийся на передаче потоков данных. Примерами таких потоков данных могут быть логи, метрики, данные трассировок и любые другие данные, которые можно представить в виде потока сообщений.

UA поддерживает отправку логов в OpenTelemetry Collector, Logbroker, сбор метрик и отправку их в распределенную и высокодоступную систему количественного мониторинга Yandex Monitoring.

Logshatter – распределенный сервис для преобразования сообщений из Logbroker и отправки сообщений батчами в ClickHouse.

2.3. ClickHouse

ClickHouse - столбцовая система управления базами данных для онлайн обработки аналитических запросов (OLAP). Логи, которые приходят в Collector отправляются в ClickHouse для агрегации и дальнейшей визуализации в Логвьюшнице.

2.4. Grafana и Yandex Monitoring

Grafana – это платформа с открытым исходным кодом для аналитики и визуализации данных, которая позволяет пользователям создавать графики, диаграммы и отчеты на основе различных источников данных. Она широко используется для мониторинга и анализа производительности систем, приложений и сервисов. Grafana предоставляет возможность работы с различными типами данных, такими как временные ряды, события и сигналы.

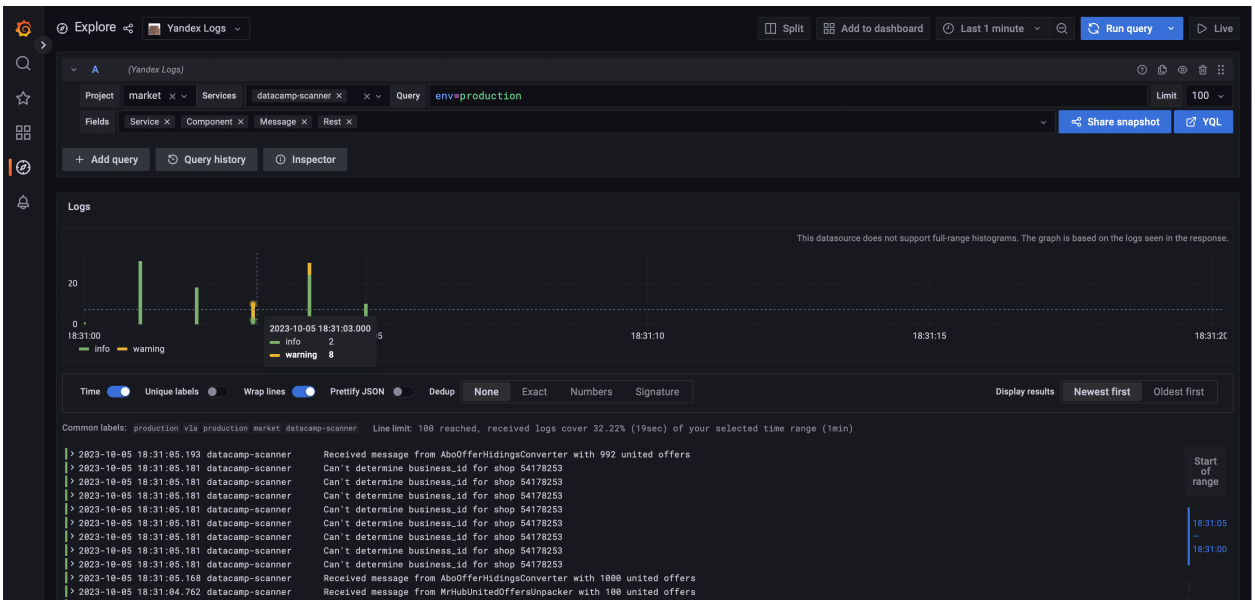

Рис. 3: Интерфейс Логвьюшницы с логами DataCamp

Yandex Monitoring – это распределенная и высокодоступная система количественного мониторинга: сбор, агрегация, хранение и визуализация метрик (временных рядов) с возможностью алертинга через Email, SMS, телефонный звонок, Telegram, Yandex.Messenger.

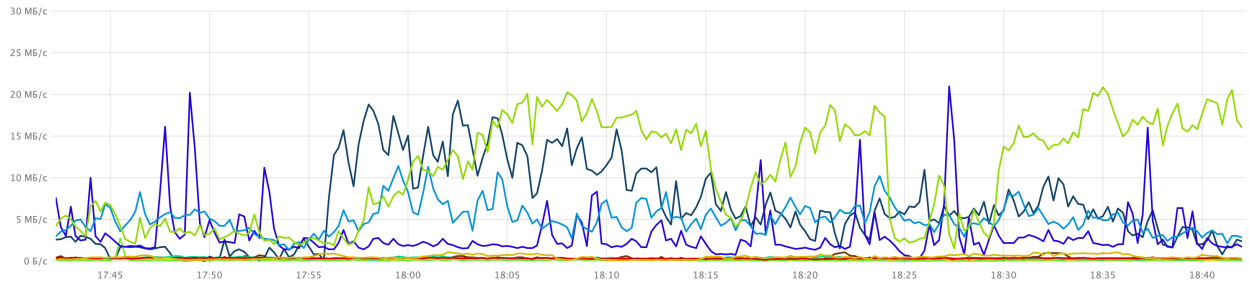

Рис. 4: Интерфейс Monitoring по объему отправляемых логов DataCamp в секунду

3. Ход работы

3.1. Сбор форматов логов внутри DataCamp

Внутри сервисов DataCamp существует четыре вида логов: tskv, BigRT, Cpp и Py. BigRT-логи – это логи из библиотеки для написания сервисов по потоковой обработке данных.

- tskv (Tab-Separated Key-Value). Строка лога начинается с "tskv" и состоит из пар ключ-значение, разделенных \t

  tskv\ttimestamp=2023-08-02T00:11:30.168189\tlevel=INFO\tmodule=SomeModule (some.cpp:123 some message)\ttask_id=7FB98A227C00\tthread_id=0x00007FB98F9FA700\ttext=some message\ttotal_time=0.007699trace_id=7814cfdc98c440ae92c89ded02724cff\tspan_id=f31c5886c76cc8fc\tparent_id=\tlink=711ebaaf35774908a6f92a38f47c3f5b

- BigRT. Строка лога состоит из времени записи, уровня логирования, категории, названия потока и еще нескольких меток, разделенных \t

  2023-09-04 15:22:29,578979\tI\tBigRT\tSomeMessage\tthread_name\tSomeLabel\tAnotherLabel

- Cpp. Строка лога состоит из уровня логирования, времени записи, источника записи и сообщения, разделленных пробелом

  INFO: 2023-09-04 00:00:27.974 +0300 someFile.cpp:12 Some Message

- Py. Строка лога состоит из того же набора меток, что и Cpp-логи, но в другом формате

  2023-09-04 15:19:05,289 INFO [Module.File:123] some message

3.2. Описание способов поставки логов

Перед началом работы уже была настроена поставка логов DataCamp, используя следующую архитектуру:

1. Приложение пишет свои логи на диск
2. Unified Agent читает и отправляет их в очередь Logbroker
3. Logshatter читает логи из очереди, парсит и отправляет в ClickHouse

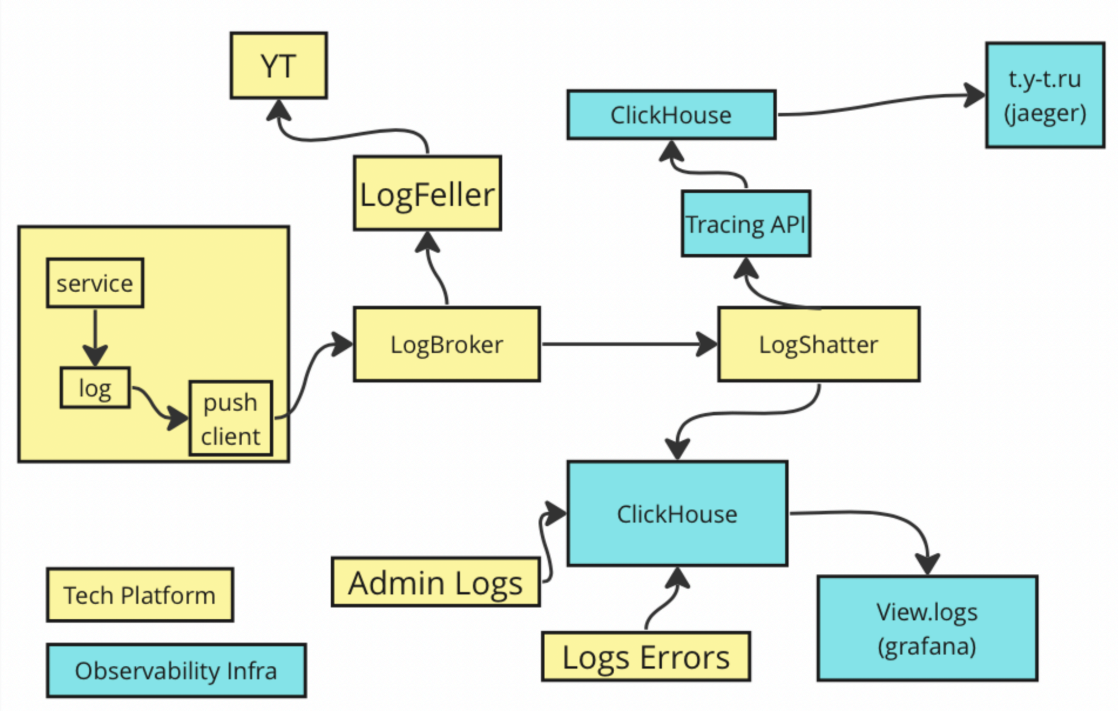

Рис. 5: Поставка логов через Logbroker

Основным минусами такой архитектуры являются время поставки логов (лишняя запись в очередь и ее вычитывание), проблемы с масштабируемостью и полные потери логов в случае инцидентов с LogBroker.

Развитием этой архитектуры стал отказ от промежуточной записи в очередь и переход к единому формату сообщений OpenTelemetry.

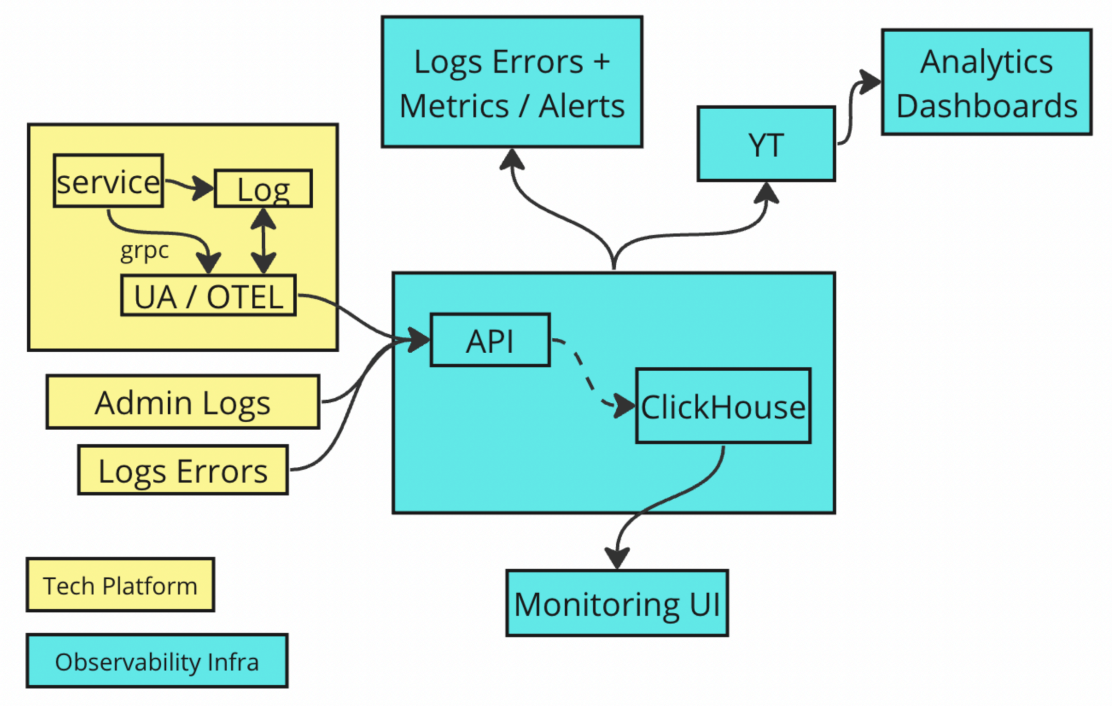

Рис. 6: Прямая поставка логов

Внутри Маркета DataCamp стал одним из первых сервисов, переведенных на новую систему логов.

3.3. Переход на новую систему логов

3.3.1. Настройка Unified Agent

Все сервисы DataCamp разворачиваются с помощью Yandex Deploy на сотнях хостах в трех окружениях – testing, prestable и production. На каждом хосте запущено по крайней мере два контейнера: `application_box` с основным приложением и `infra_box` с другими приложениями, например, для отправки логов с хоста. Ресурс Unified Agent поставляется в каждый `infra_box` по версии, которая указывается для каждого сервиса и его окружения. "Из коробки" Deploy не поддерживает обновление ресурсов на сервисах одновременно, поэтому был написан python скрипт, который обновляет UA до версии с поддержкой плагина для отправки в OpenTelemetry формате.

Для подключения UA с отправкой в Collector необходимо настроить yaml-конфиг нужного сервиса: указать файл с логами, парсер, формат

отправки (сериализованный protobuf/json) в Collector, а затем доставить его в `infra_box`.

3.3.2. Подключение логов `multi_offers_tasks`

`multi_offers_tasks` – микросервис, помогающий реализовывать операции над множеством офферов. На момент запуска сервиса логи (cpp и tskv) не были подключены ни к какой системе поставки логов, а просто хранились на хосте.

Из-за того, что для cpp логов готового парсера не было, а реализованный парсер для tskv-логов на Golang не использовал OpenTelemetry классы (LogRecord, ExportLogsServiceRequest), а лишь мимикрировал под них с помощью json (что негативно влияет на время доставки в Collector и на рефакторинг в будущем), было принято решение реализовать парсер на C++ для всех типов логов внутри DataCamp.

Парсер логов работает внутри `infra_box` и конфигурируется с помощью Unified Agent:

1. UA читает строку из файла, отправляет в парсер
2. Парсер преобразует строку в сериализованный ExportLogsServiceRequest и пишет в stdout
3. UA читает stdout и отправляет в Collector

После проверки доступности логов `multi_offers_tasks` в testing, логи были включены для prestable и production.

3.3.3. Подключение остальных микросервисов

Для подключения остальных сервисов были проделаны те же действия: реализованы недостающие парсеры (для Py и BigRT логов), настроены конфиги и протестирована доставка логов в каждом окружении (testing, prestable, production). Для облегчения перехода остальных сервисов Маркета на новую систему была написана недостающая документация.

После полного перехода на прямую поставку логов в ClickHouse, время доставки улучшилось почти в 2 раза (95-й перцентиль), а также была устранена проблема с потерей логов.

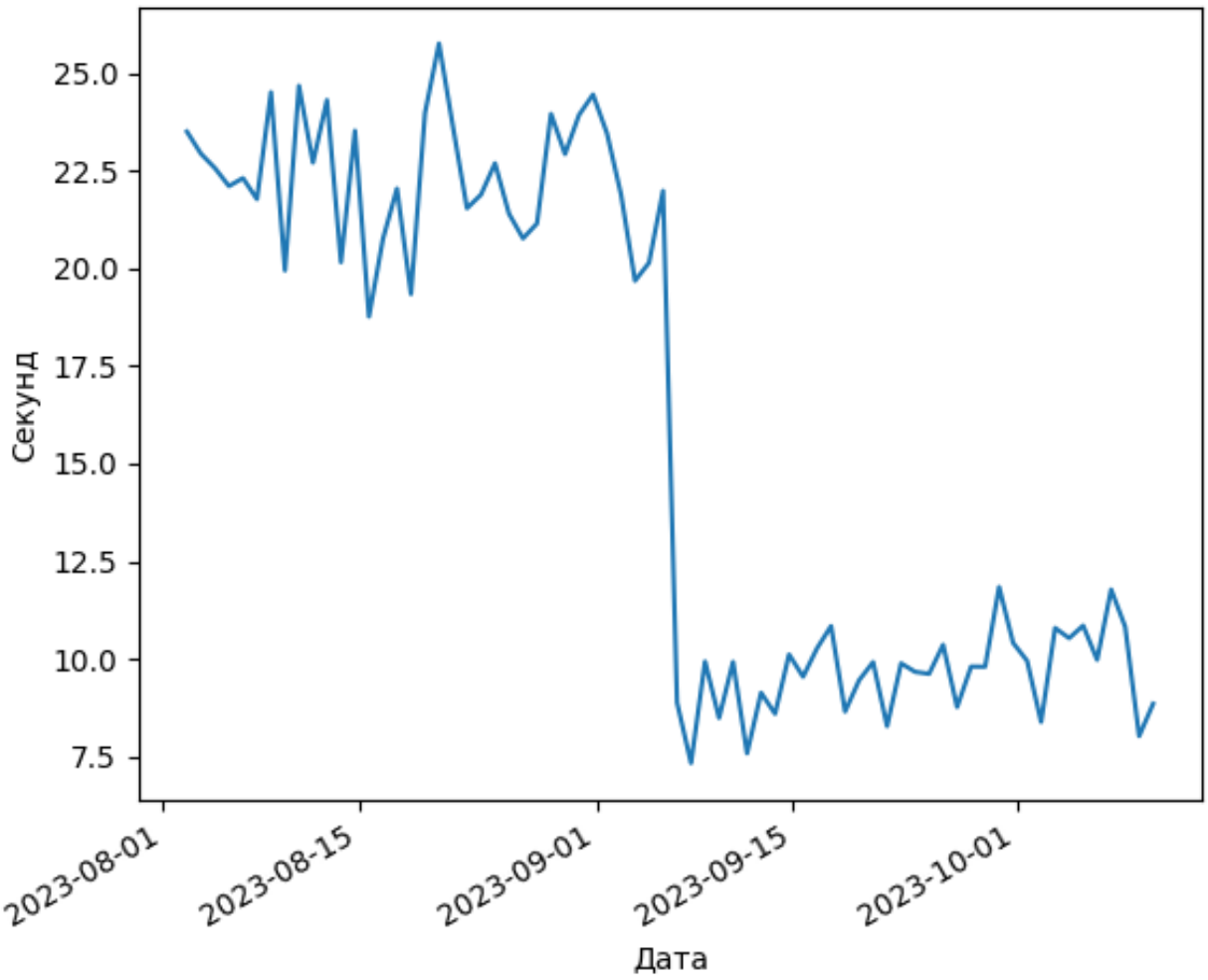

Рис. 7: 95-перцентиль среднего времени доставки логов по всем сервисам DataCamp

Заключение

В ходе работы были выполнены все поставленные задачи:

- Выполнен обзор существующих инструментов для поставки и агрегации логов
- Выполнен обзор всех форматов логов, использующихся в DataCamp
- Для всех сервисов DataCamp выполнен переход с системы поставки логов через Logbroker на прямую поставку
- Протестирована доставка логов для всех сервисов

Список литературы

[1] ClickHouse. — https://clickhouse.com/. — (дата обращения: 05.10.2023).

[2] ClickHouse for developers. — https://yandex.ru/dev/clickhouse/. — (дата обращения: 05.10.2023).

[3] FluentBit. — https://github.com/fluent/fluent-bit. — (дата обращения: 05.10.2023).

[4] Grafana. — https://grafana.com/. — (дата обращения: 05.10.2023).

[5] OpenTelemetry. — https://opentelemetry.io/. — (дата обращения: 05.10.2023).

[6] OpenTelemetry for new first-party application. — https://opentelemetry.io/docs/specs/otel/logs/#new-first-party-application-logs. — (дата обращения: 05.10.2023).

[7] OpenTelemetry logs direct to Collector. — https://opentelemetry.io/docs/specs/otel/logs/#direct-to-collector. — (дата обращения: 05.10.2023).

[8] OpenTelemetry logs via file. — https://opentelemetry.io/docs/specs/otel/logs/#via-file-or-stdout-logs. — (дата обращения: 05.10.2023).

[9] Unified Agent. — https://cloud.yandex.ru/docs/monitoring/concepts/data-collection/unified-agent/. — (дата обращения: 05.10.2023).