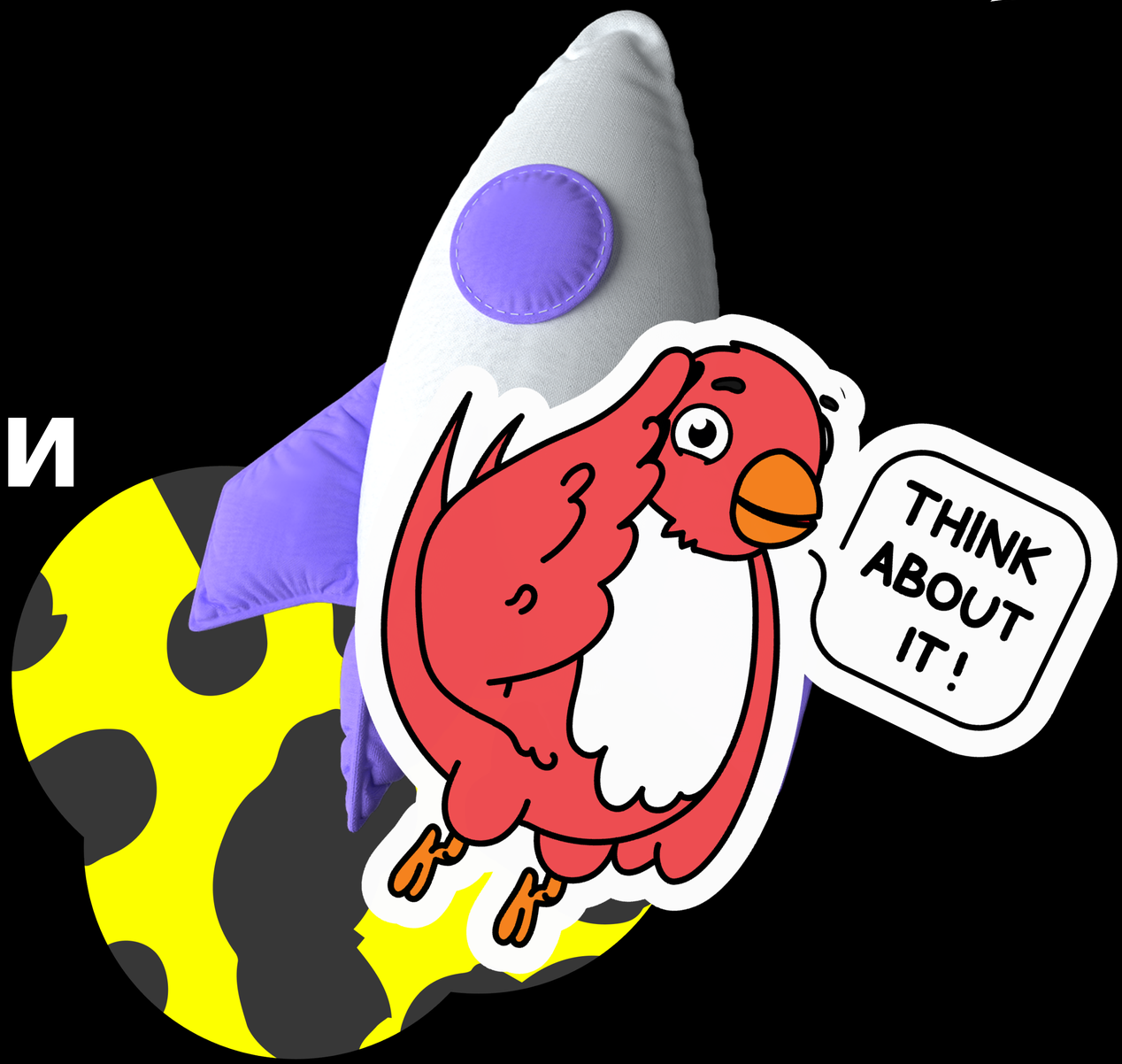
avito.tech Москва 2024

Как мы Apache Kafka на Redpanda меняли



Роман Ананьев NoSQL Engineer

Роман Ананьев

NoSQL Engineer в юните DBA

- больше трех лет в Авито;
- развиваем NoSQL-технологии;
- работаю с Kafka и прочими брокерами сообщений с 2015 года;
- рад ответить на вопросы tg: @rum_cola.

План доклада

01

Оглянемся назад

02

Оглянулись, что не понравилось?

03

Осмотримся вокруг

04

Подход к кандидатам

05

Основные тесты

06

Замена и что она нам дала

Оглянемся назад

Kafka Federation

1

Суть проекта

01

Использовать отдельные кластеры Kafka на запись и один — на чтение

02

Настроить репликацию из каждого кластера на запись в кластер на чтение

03

Каждый кластер на запись работает локально в своем ДЦ

04

Набор топиков идентичен для всех кластеров

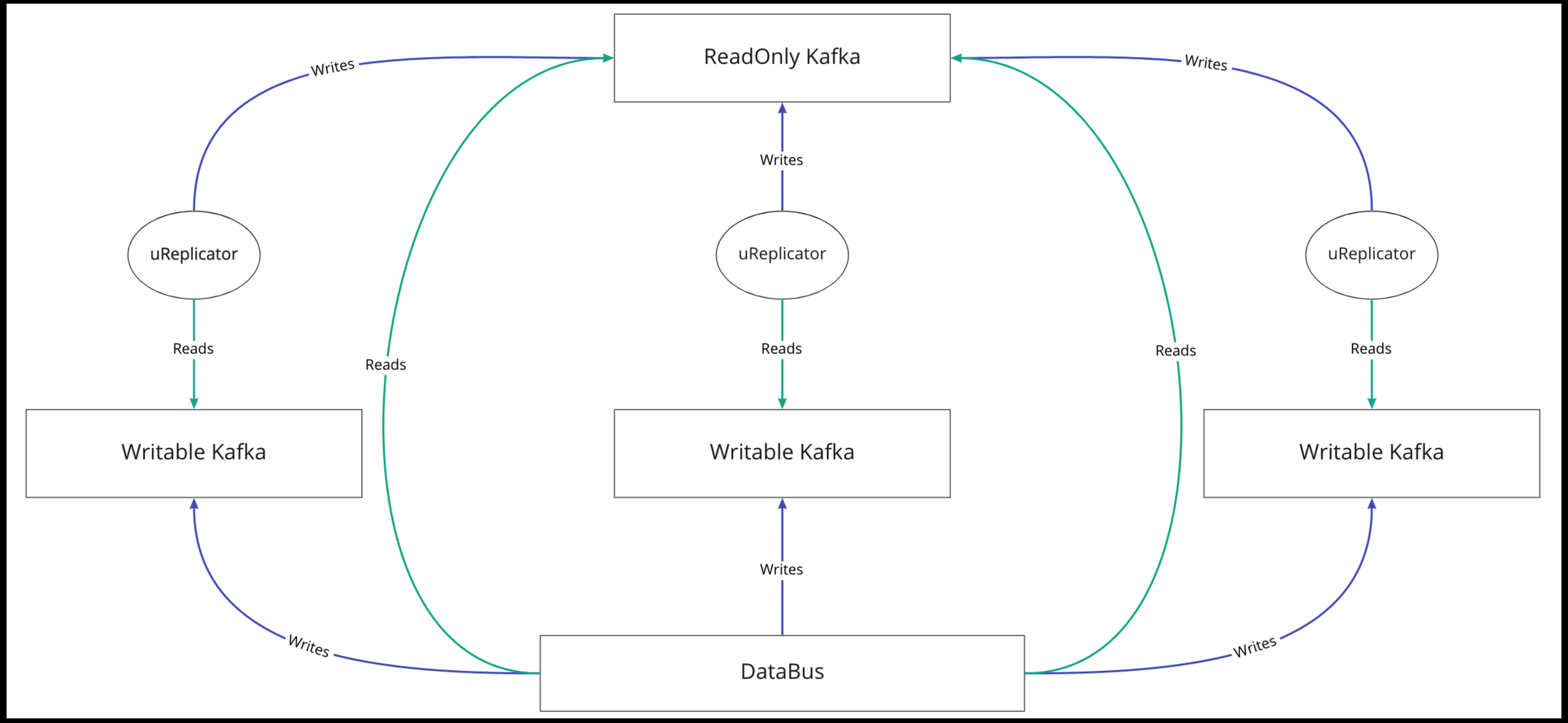
05

При отключении одного из ДЦ, оставшиеся принимают весь трафик

06

При отключении ДЦ нет влияние на сам produce и его latency

Kafka Federation - устройство



Моя статья про это на Хабре

Оглянулись, что не понравилось?

Сложность

2

Не обновляемость и ресурсы

01

Архаичность

- последний commit в Ureplicator — от 2021 года;
- привязка к Zookeeper.

02

Ресурсы Ureplicator

- Containers (x3): 27;
- - Manager: 2;
- - Controller: 3;
- - Worker: 22;
- - CPU: 2;
- - RAM: 1GB.

03

Ресурсы Kafka

- Topics per cluster (x4) - 2000;
- Partitions per cluster (x4) - 50000;
- Brokers per cluster WO (x3) - 6;
- Brokers per cluster RO - 21.

Осмотримся вокруг

В поисках простоты

3

Что хотели бы увидеть

01

Ищем простоту

- эксплуатации;
- обслуживания;
- мониторинга;
- состава кластера.

02

Кодовая база клиентов

- поддержка Kafka API;
- клиент на GO.

03

Ресурсы

- лучшая утилизация серверов;
- плавная масштабируемость вверх и вниз;
- поддержка работы в K8S by design.

Что увидели

- NATS Jetstream
- Apache Pulsar
- Redpanda
- Redis (KeyDB)
- YaDB Topics

Подход к кандидатам

Взяли и просеяли

4

Те, кто просеялся

NATS Jetstream — низкая надежность и высокий latency produce\consume

Apache Pulsar — низкая надежность, наличие почти не исправляемых багов и вялое коммьюнити

Redpanda — и осталась она одна

Redis (KeyDB) — очереди еще ладно, но классический брокер сообщений крайне непредсказуем и сложен

YaDB Topics — это интересный вариант, но пока молодой и с начальной поддержкой Kafka API

Основные тесты

И только Redpanda

5

Вот и потестили

01

Требовательна к железу и настройкам ОС

02

Горизонтально масштабируется вверх и вниз

03

Latency на уровне Apache Pulsar и Kafka

04

Всего 1 компонент и работа в K8S

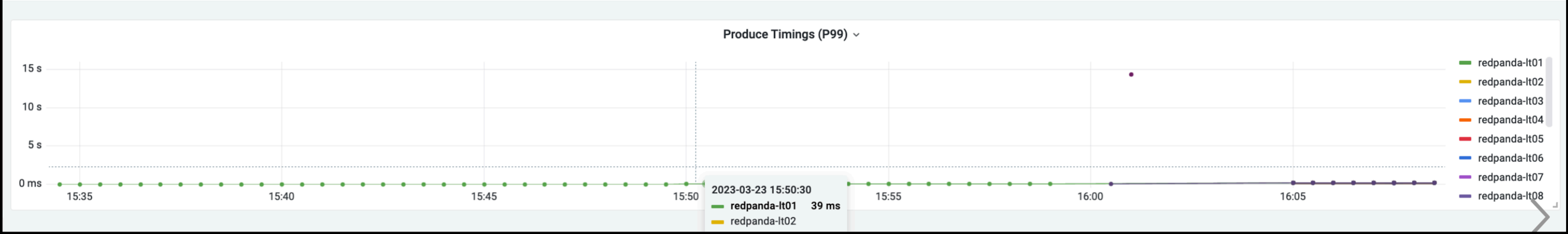
05

Поддержка Kafka API и prometheus-метрик

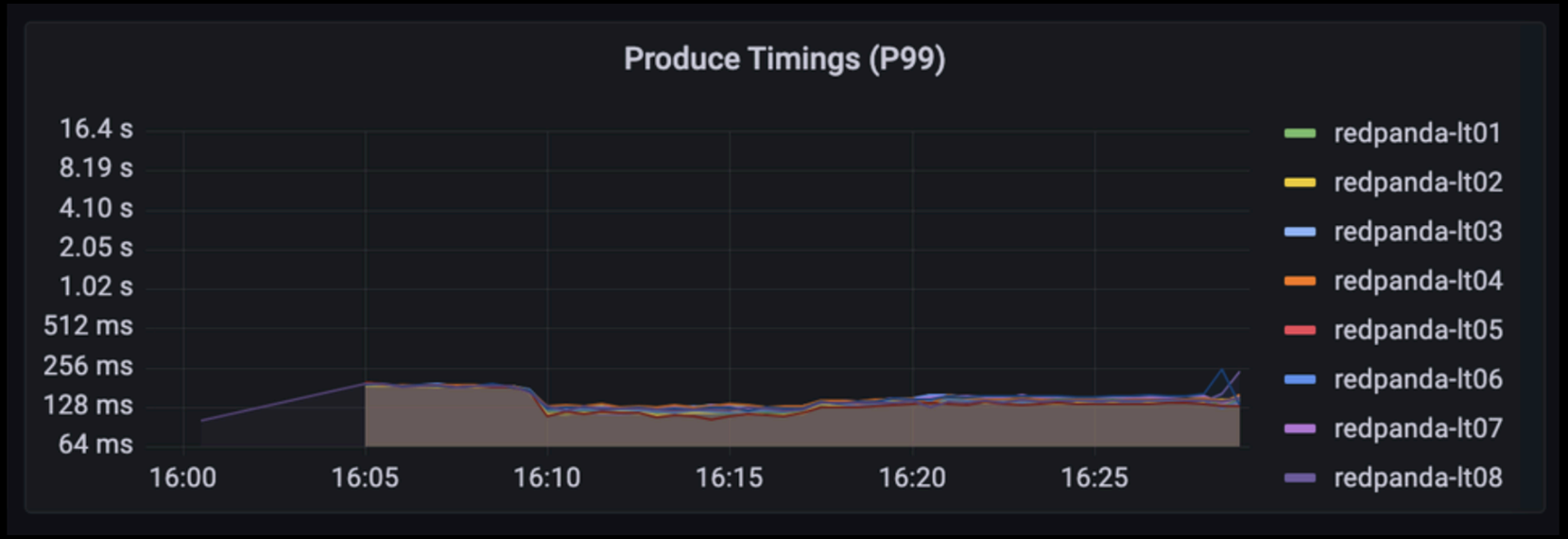
06

Отличное время сходимости кластера после отказа брокера

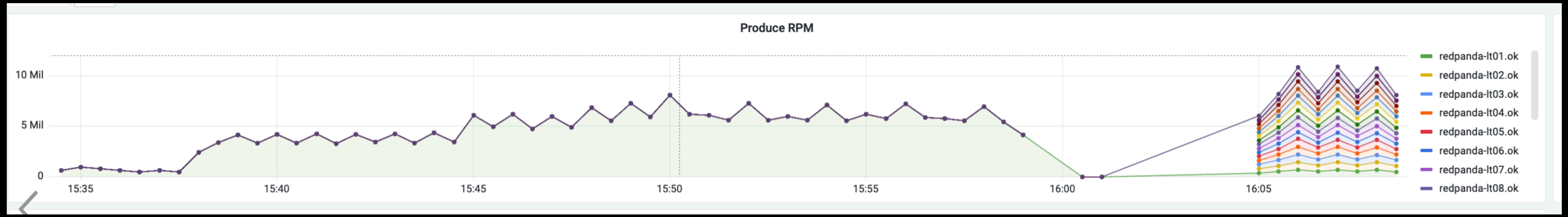
Latency на produce (HDD)



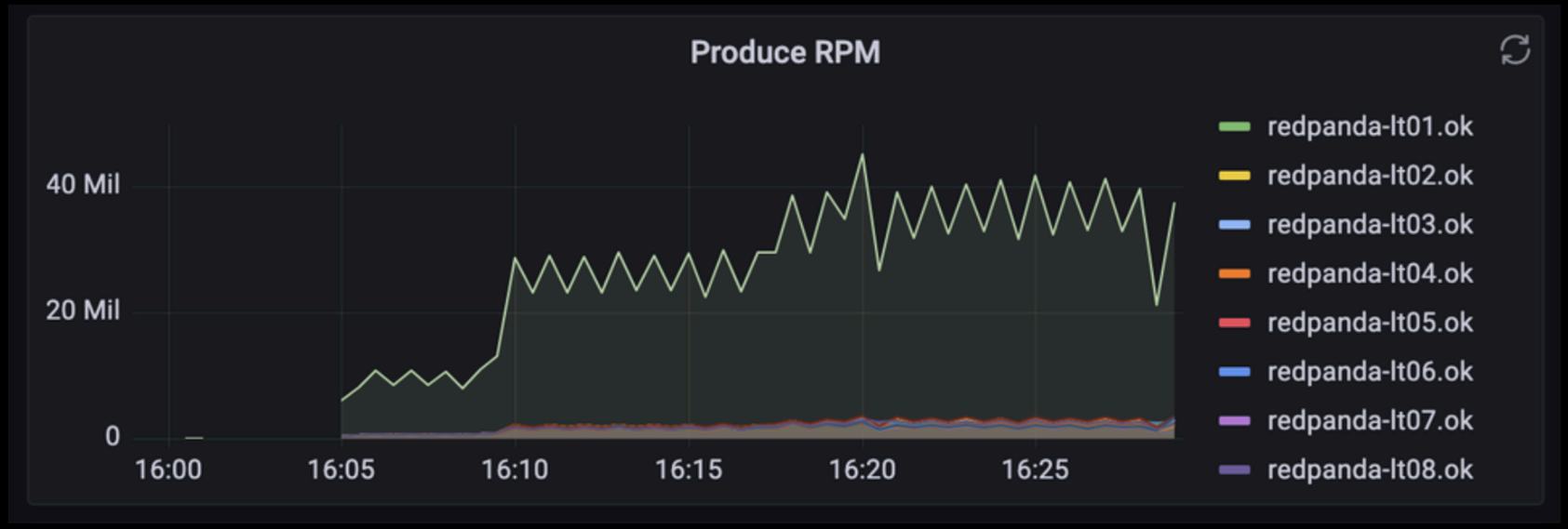
Latency на produce (SSD)



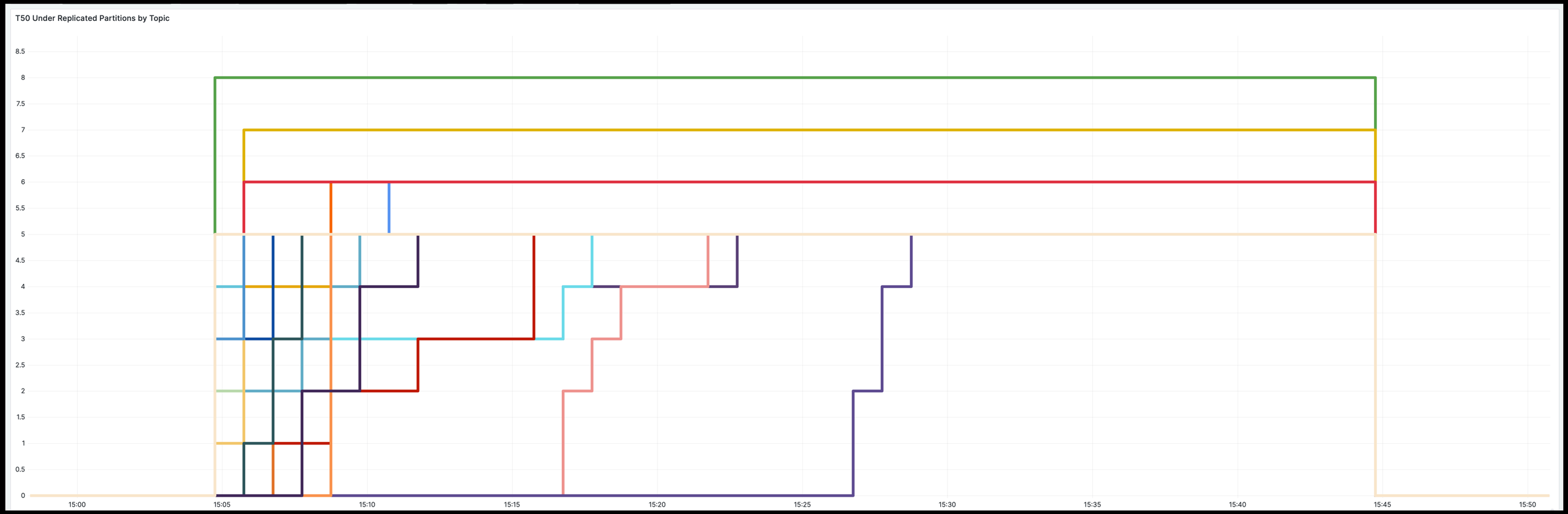
RPM на produce (HDD)



RPM на produce (SSD)



Время сходимости кластера (5M produce RPM и 5 брокеров)



Теперь чуть поближе правую часть

Замена произошла

Что мы выиграли?

6

(Не)поправимые улучшения

01

Ресурсы

Было:

- 39 prod брокеров;

Стало:

- 15 prod брокеров.

02

Доступность

Было:

- время сходимости кластера около 10 минут;

Стало:

- время сходимости кластера около 1 минуты.

03

Упрощение

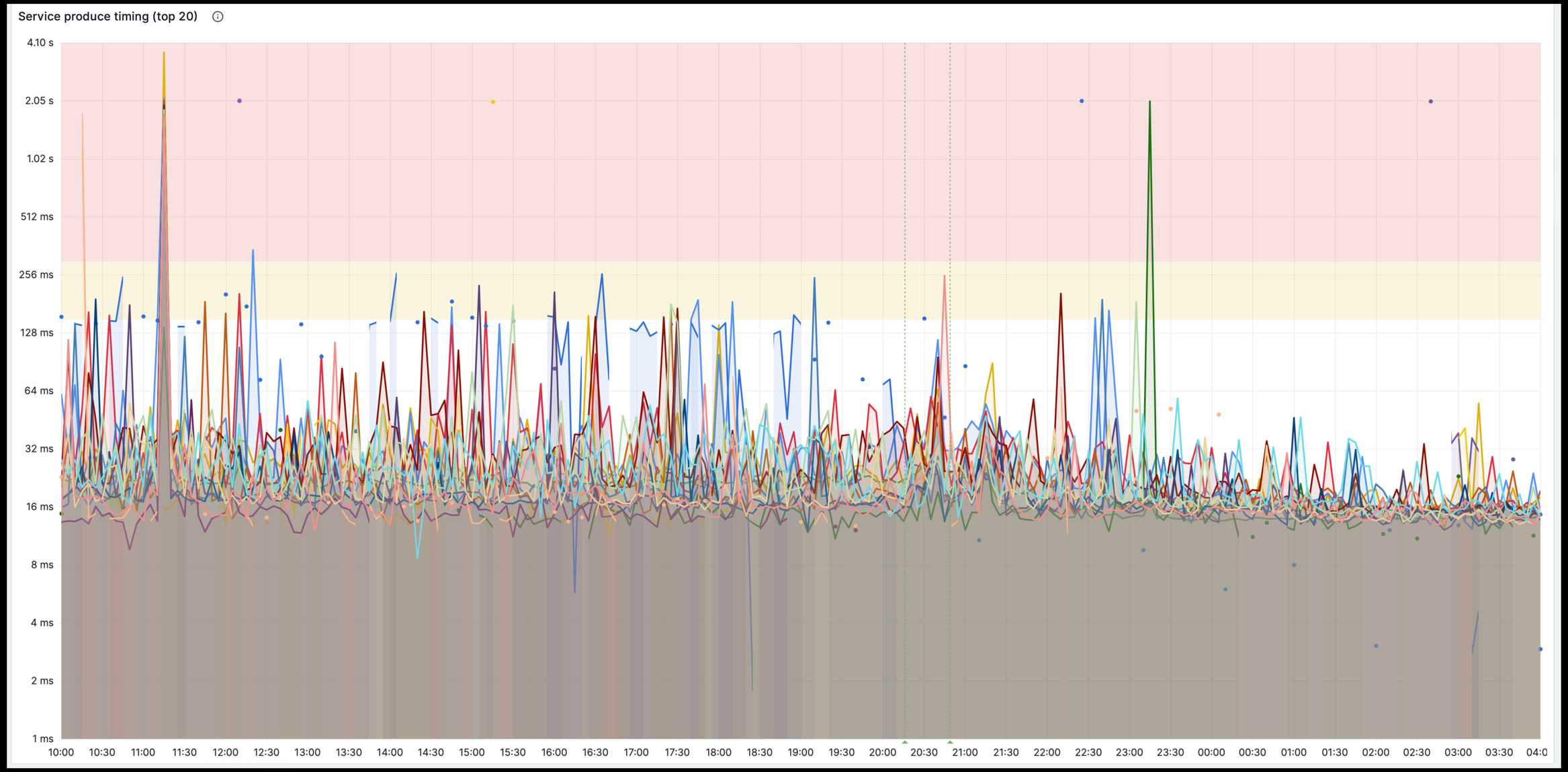
Было:

- федерация из 4-х prod Kafka кластеров;
- 3-х компонентность системы;

Стало:

- федерация из 3-х prod кластеров;
- 1 компонент системы.

Kafka/Redpanda - produce latency



Итоги и выводы

Приключение удалось

0

Основные мысли

01

Может иногда полностью заменить Kafka

02

Лучше утилизирует железо, но и более требовательна к нему

03

Легковеснее и менее компонентна

04

Нужно периодически оглядываться назад и осматриваться вокруг

05

Тесты и время

06

Упрощать — новый хит сезона

Вопросы