

Яндекс

Яндекс

Резервное копирование

с WAL-G

Андрей Бородин

В предыдущих сериях

Владимир Бородин: Как перестать беспокоиться и спать спокойно

› clck.ru/CHXZ4

Евгений Дюков: «Двухколёсные» бэкапы

› clck.ru/CHXiK

Простой способ переноса данных

pg_dump



pg_dump



pg_dump — extract a PostgreSQL database into a script file or other archive file

pg_dump



- › Логическая копия ваших данных
- › Не зависит от версии Pg
- › Можете нарезать свою базу кусочками

Стандартный инструмент в PgAdmin

Резервная копия базы данных "postgres"

Имя файла

Формат: Настраиваемый

Степень сжатия

Кодировка

Количество заданий

Имя роли

Файл | Параметры дампа #1 | Параметры дампа #2 | Объекты | Сообщения

Резервная копия | Отменить

Настройки 0

Резервная копия базы данных "postgres"

Секции

Pre-data

Данные

Post-data

Тип объектов

Только данные

Только схема

Blobs

Не сохранять

Владелец

Привилегии

Табличное пространство

Нежурналируемые данные таблиц

Файл | Параметры дампа #1 | Параметры дампа #2 | Объекты | Сообщения

Резервная копия | Отменить

Настройки 1

Резервная копия базы данных "postgres"

Запросы

Добавить CREATE DATABASE

Добавить DROP DATABASE

Insert с указанием колонок

Использовать команды Insert

Выключить

Триггер

\$ кватирование

Разное

Использовать SET SESSION AUTHORIZATION

С OID

Подробные сообщения

Использовать двойные кавычки для идентификаторов

Файл | Параметры дампа #1 | Параметры дампа #2 | Объекты | Сообщения

Резервная копия | Отменить

Настройки 2

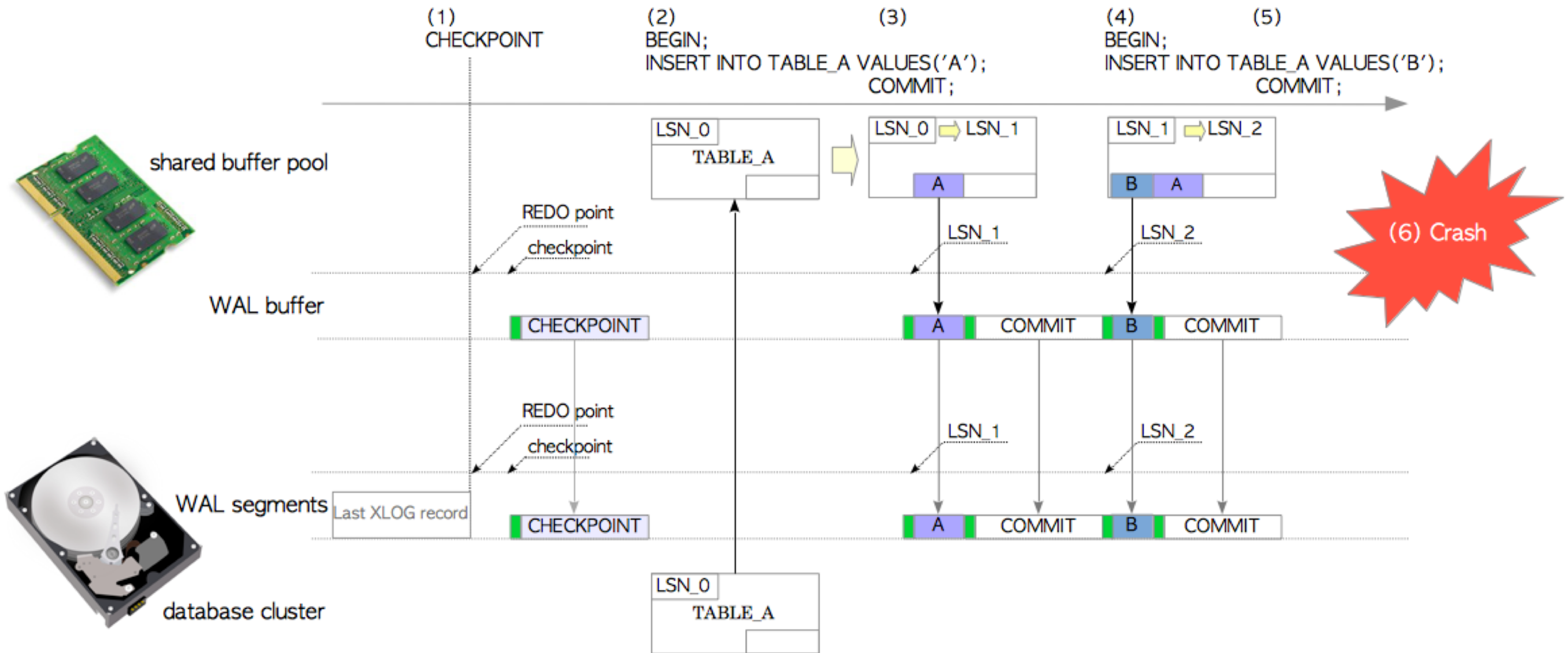


Если всё серьёзно

Point in time recovery



Write Ahead Log **TODO**



pg_start_backup API

- › Checkpoint
- › Full page writes = on
- › Записывает LSN начала бэкапа

 Не консистентная копия базы данных!

pg_basebackup

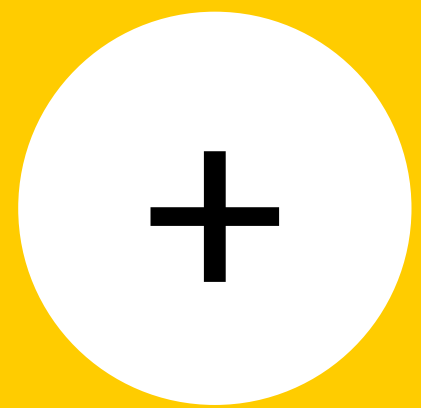
■ Самый простой инструмент создания резервной копии



Barmman

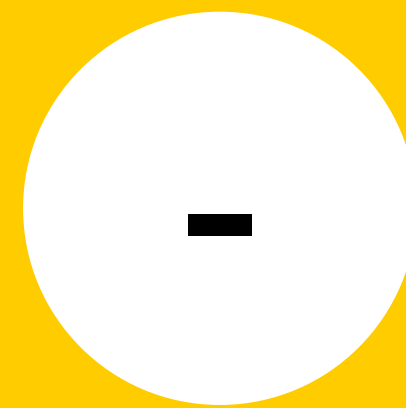
Backup and recovery
manager for PostgreSQL

Barman



Плюсы

- › Много док и юзеров
- › Один сервер на много баз



Минусы

- › Сложен
- › Должен быть на отдельном железе
- › Ещё один объект мониторинга



Простой инструмент для решения сложной задачи

WAL-E





Команды WAL-E

- › WAL-PUSH
- › WAL-FETCH
- › BACKUP-PUSH
- › BACKUP-FETCH

Вообще, WAL-E умеет много всего

- › BACKUP-LIST

- › DELETE

- Поддерживает AWS S3, Azure Blob, Google Storage, Swift, FS

WAL-E

- › Написан на Python
- › Активно использует конвейер Unix
- › Не очень производительный
- › Фокус на простоте настройки

| Куда развивать WAL-E?

Фич-то и так много

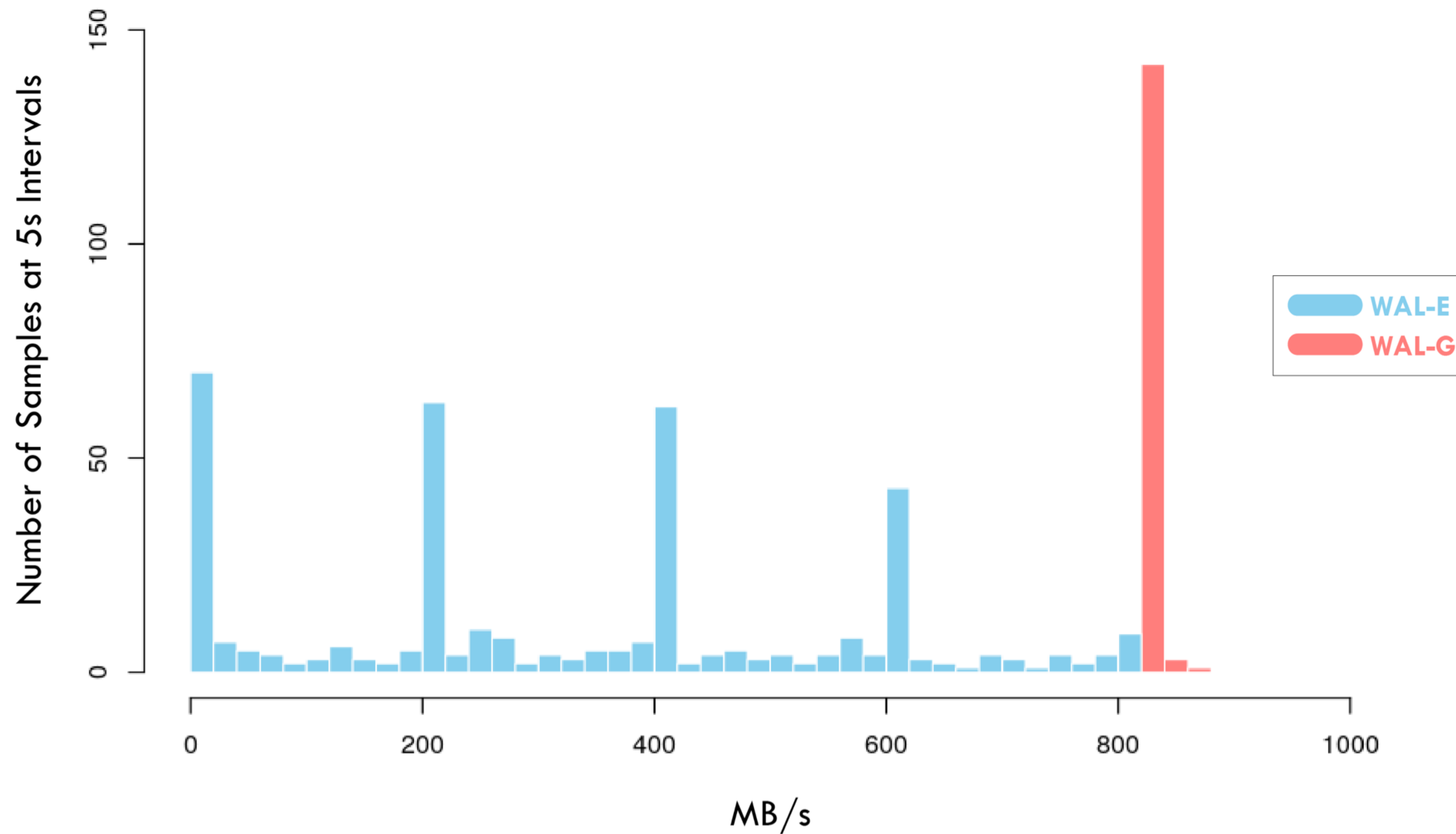
WAL-G



WAL-G

- Переписан на Go
- С отказом от идеологии Unix
- Совместными усилиями Katie Le и Daniel Farina (Citus Data)

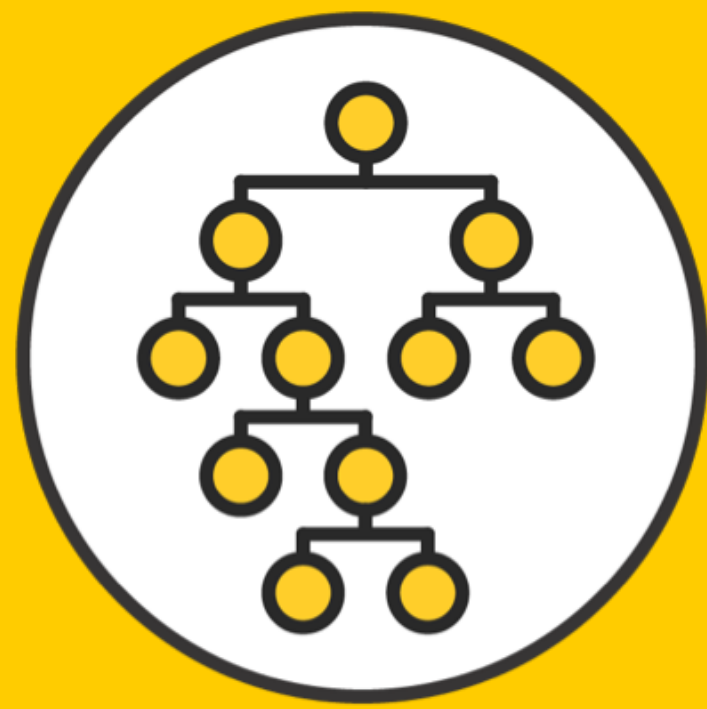
Histogram of Throughput



WAL-G

	Avg throughput	Std	Med
WAL-E	323 Mb/s	236	307 Mb/s
WAL-G	838 Mb/s	4.2	838 Mb/s

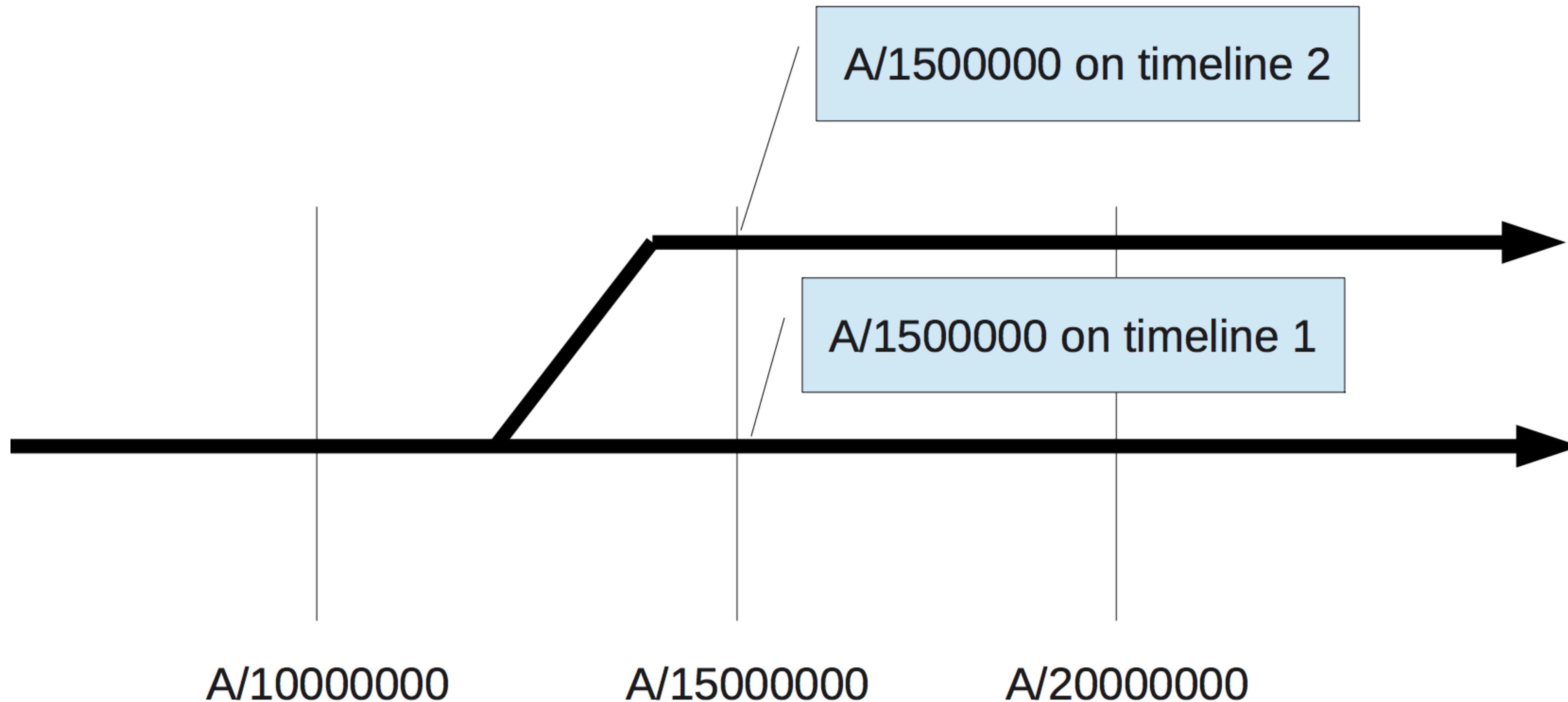
Бэкапы с реплик



Разгружают мастер
Такие же актуальные

› Но есть одна тонкость...

PostgreSQL Timeline



Дельта-копии



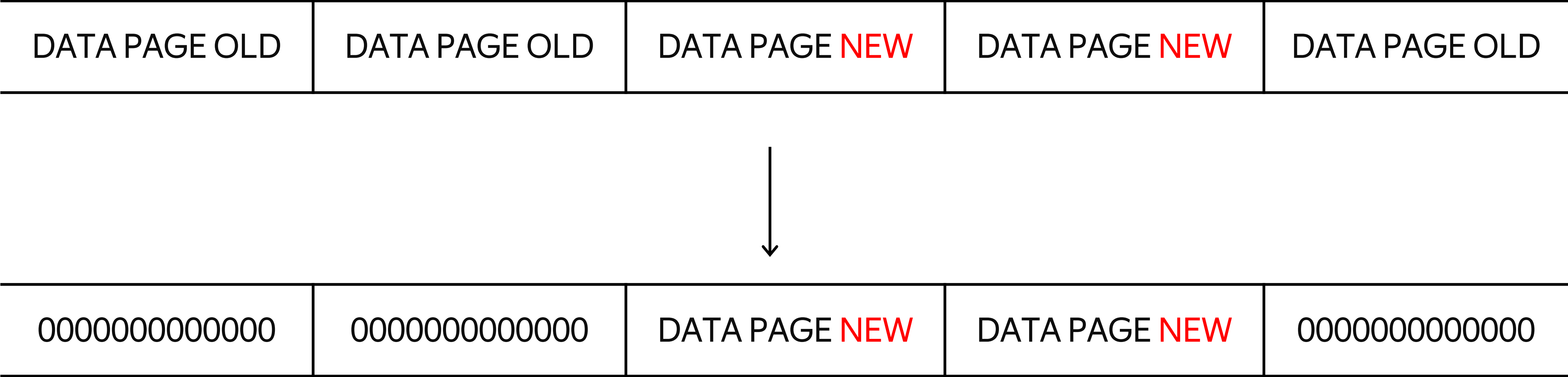
- › На основе ModTime
- › На основе LSN

API дельта-копий

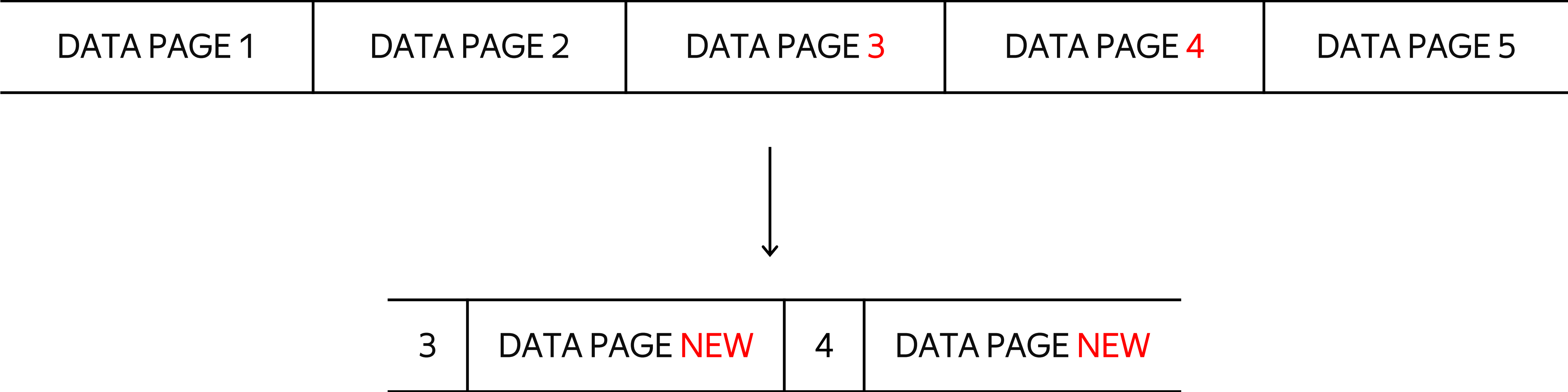
WALG_DELTA_MAX_STEPS=6

WALG_DELTA_ORIGIN=[LATEST| LATEST_FULL]

Формат дельта-копий



Формат дельта-копий



Формат дельта-копий

DATA PAGE 1	DATA PAGE 2	DATA PAGE 3	DATA PAGE 4	DATA PAGE 5
-------------	-------------	-------------	-------------	-------------



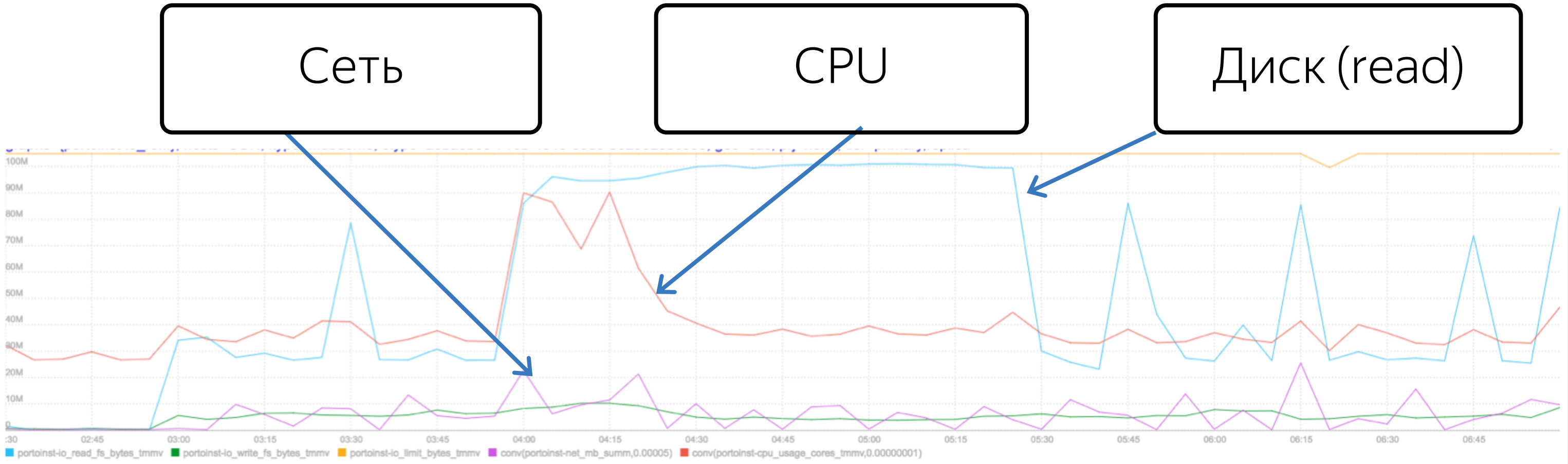
3	4	DATA PAGE NEW	DATA PAGE NEW
---	---	---------------	---------------

Оптимизация наката файлов

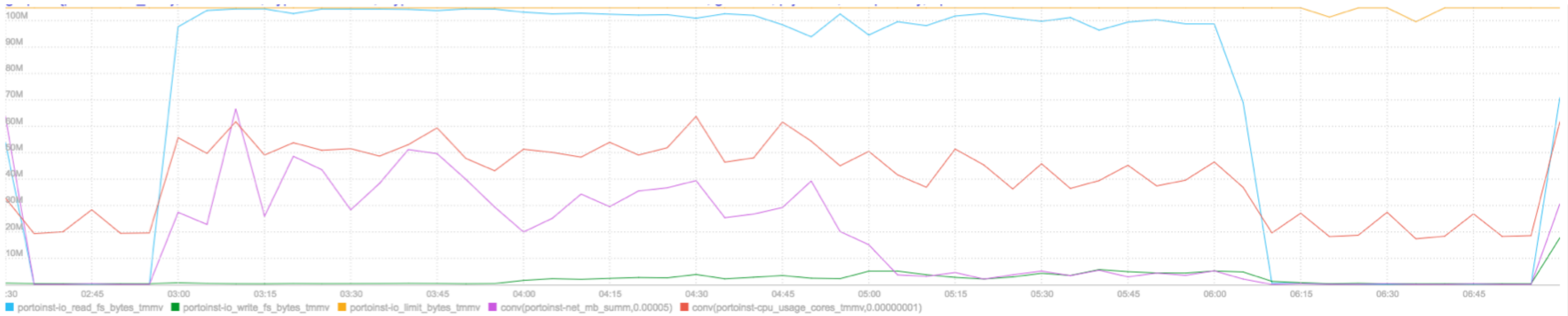
Base	Delta 1	Delta 2	Delta 3	Delta 4
File X	Δ file X	Δ file X	File X	Δ file X

Графики

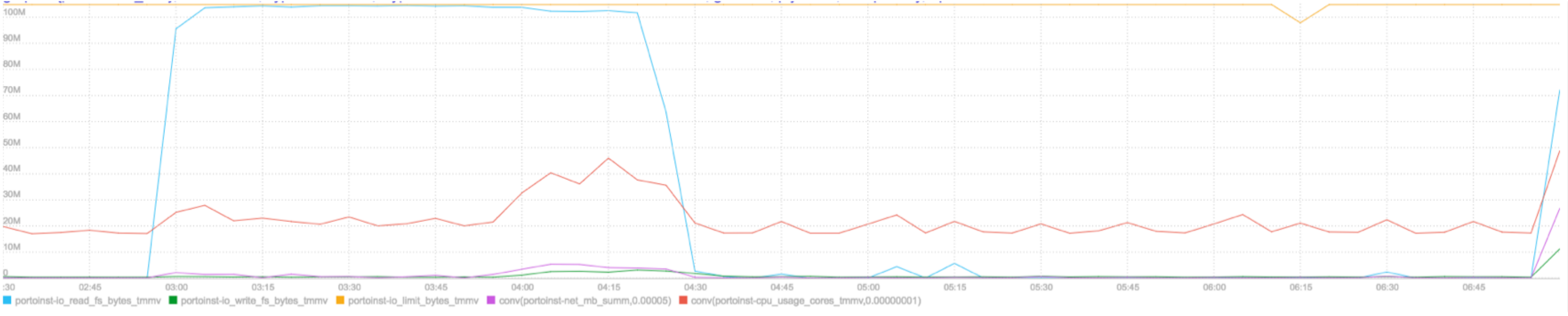
WAL-E



WAL-G



WAL-G Delta



Prefetch



Служебная команда

› WAL-PREFETCH

Удаление



- › DELETE BEFORE X [название]
- › DELETE RETAIN Y [количество]

Нельзя удалить всё

ЛИСТИНГИ



› BACKUP-LIST

```

root@: ~ # su postgres
postgres@:/root$ cd ~
postgres@:~$ /usr/bin/envdir /etc/wal-g/envdir /usr/bin/timeout 600 /usr/bin/wal-
g backup-list
BUCKET:
SERVER: wal-e/b8b0554d-09a3-4338-bac9-18757dca1d8f/906
name      last_modified    wal_segment_backup_start
base_00000001000000000000000033_00000040  2017-12-05T22:00:34Z    00000001000000000000000033_00000040
base_00000001000000000000000035_00000040  2017-12-06T22:00:32Z    00000001000000000000000035_00000040
base_00000001000000000000000037_00000040  2017-12-07T22:00:31Z    00000001000000000000000037_00000040
base_00000001000000000000000039_00000040  2017-12-08T22:00:33Z    00000001000000000000000039_00000040
base_0000000100000000000000003B_00000040  2017-12-09T22:00:32Z    0000000100000000000000003B_00000040
base_0000000100000000000000003D_00000040  2017-12-10T22:00:34Z    0000000100000000000000003D_00000040
base_00000001000000000000000041    2017-12-11T22:00:15Z    00000001000000000000000041
postgres@:~$

```

WAL-G killer feature





Патч для Barman с инкрементальными бэкапами
github.com/2ndquadrant-it/barman/issues/21

› Мёржим 2 года

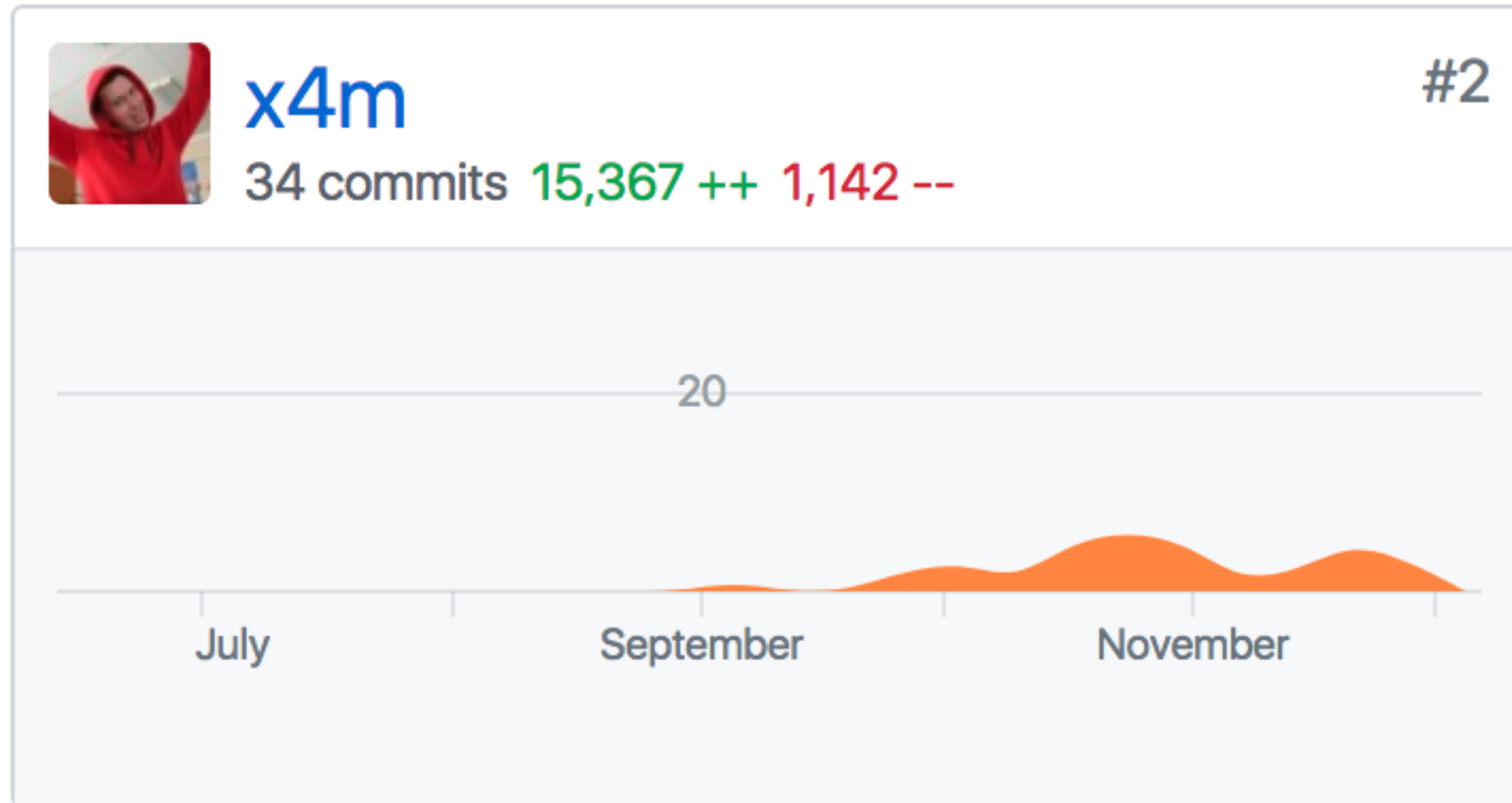


Патч для WAL-E

github.com/wal-e/wal-e/issues/271

› Мёржили полгода

С WAL-G всё проще



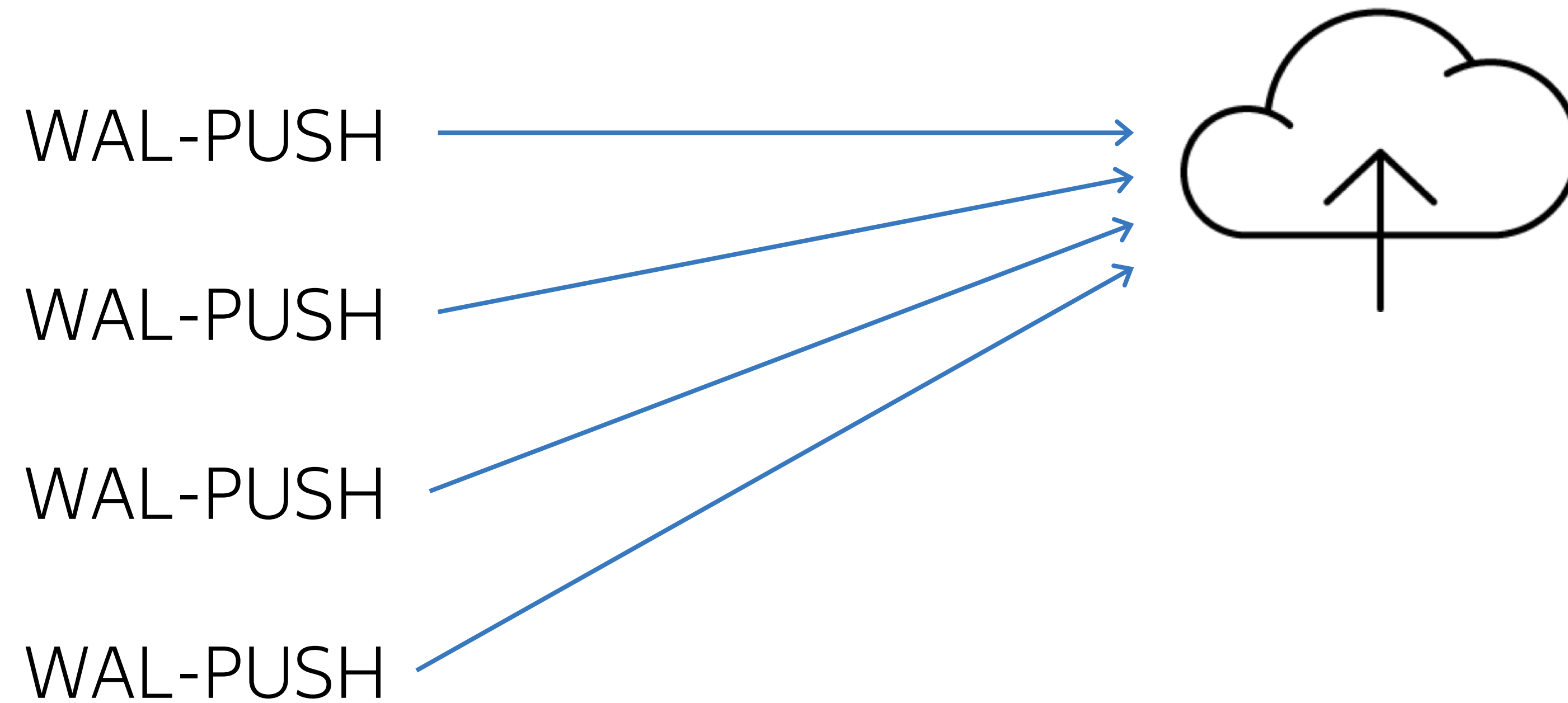
Что ещё будет?



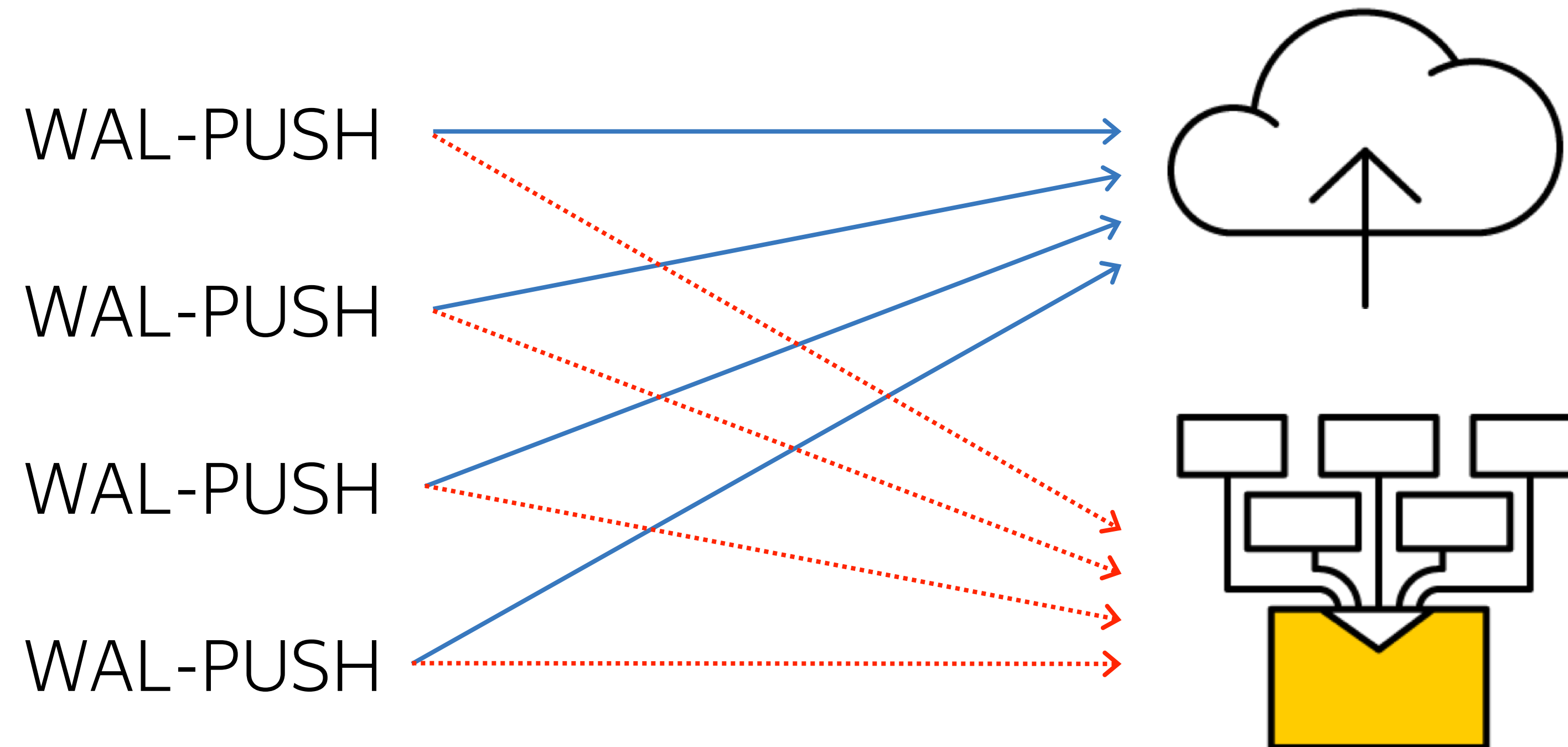
Другие фи́чи

- › Больше криптографии
- › Больше платформ хранения
- › Не только PostgreSQL?

Сканирование WAL



Сканирование WAL

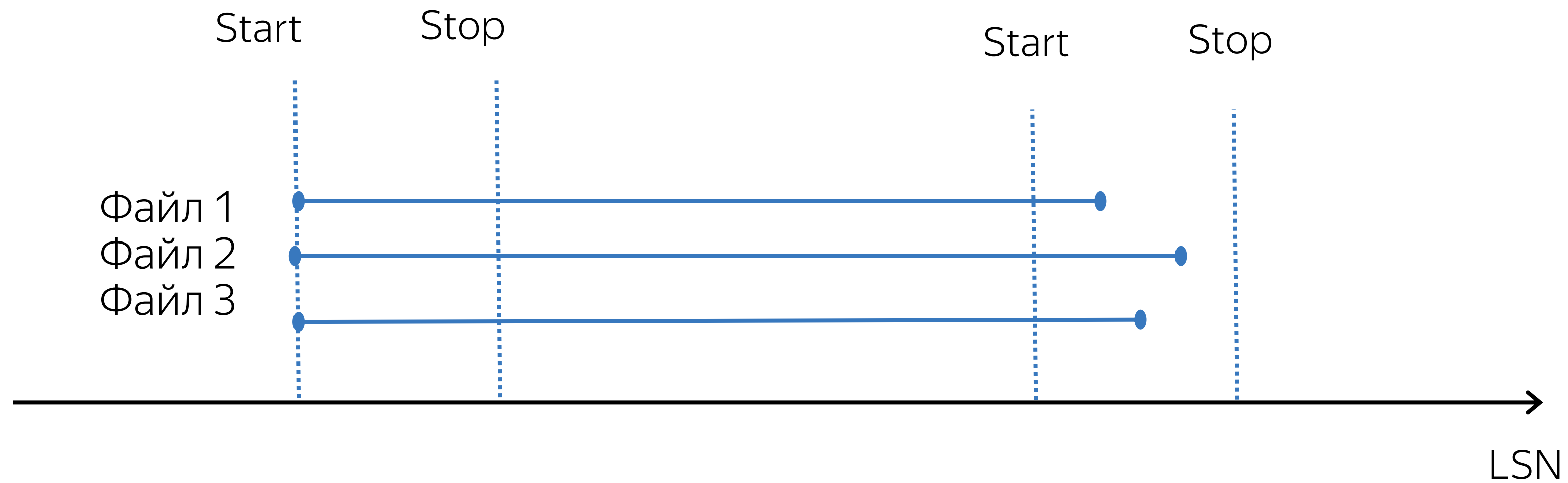


PTRACK

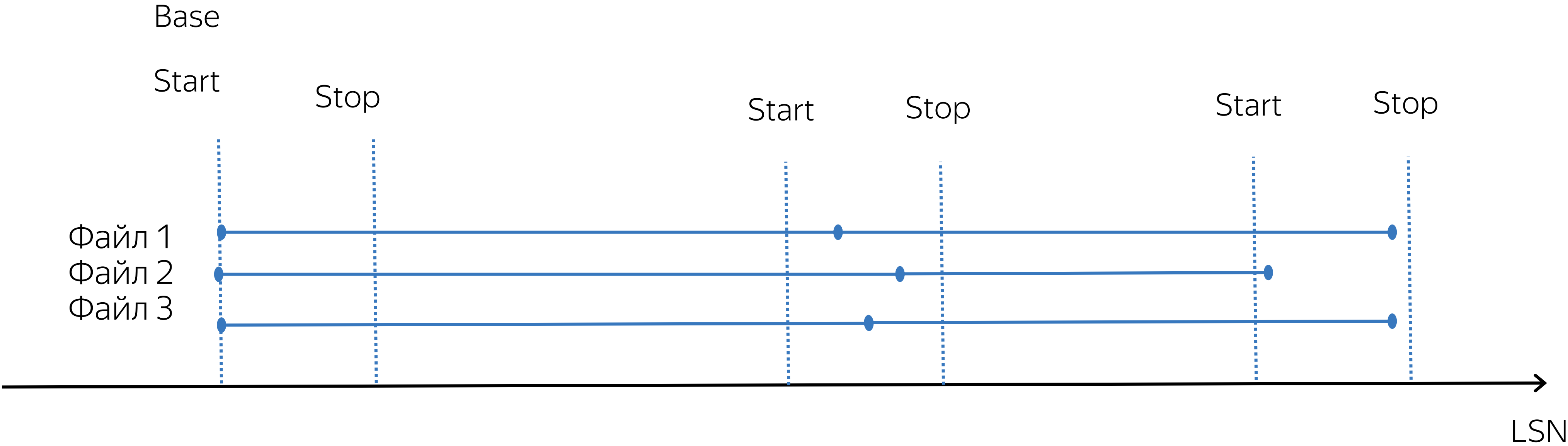


API для получения page-level разницы между состояниями БД

LSN-based дельты



РТРАСК дельты



Множественные бэкапы от одной базы

WALG_DELTA_ORIGIN=[LATEST| LATEST_FULL]

Hooks to track changed pages for backup purposes

Status	2017-09: Returned with feedback
Authors	Andrey Borodin (x4m)

Хотелось бы в pg_basebackup

Больше хранилищ

- › Архивация WAL в облако

Дельты

- › Часть кода уже есть в pg_rewind

Хотелось бы в pg_basebackup

- Больше хранилищ

- › Архивация WAL в облако

- Дельты

- › Часть кода уже есть в pg_rewind

- Сжатие, параллелизм, шифрование, троттлинг ресурсов, листинг, верификация, операции с дельтами

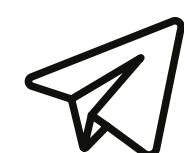
Жду вопросов 😊

Андрей Бородин

Разработчик



amborodin@acm.org



[x4mmm](https://t.me/x4mmm)