PostgresPro

Snapshot Standby с использованием BTRFS

Виктор Васильев
архитектор решений

email: v.vasilev@postgrespro.ru

Про что доклад?

- **Как создать тестовый стенд с большим объемом данных?**

Про что доклад?

- **Как создать тестовый стенд с большим объемом данных?**
- **Что для этого можно использовать?**

Про что доклад?

PostgresPro

- **Как создать тестовый стенд с большим объемом данных?**
- **Что для этого можно использовать?**
- **Быстро – за миллисекунды!**

Про стенд

- Сертифицированное ПО
- Нет виртуализации
- Реплика БД
- Постоянно восстанавливается из архива
- Нужно актуальное состояние БД
- Все время изменяется каталог PGDATA

Традиционный способ

- Каждый раз нужно
  - Копировать каталог PGDATA
  - Подниматься из РК
  - Или создавать реплику

- А что если, у вас база - 10TB?
- Все это будет медленно (часы)

- И у нас уже есть реплика в актуальном состоянии!

Моментальные снимки

- **На уровне гипервизора**
  - создать снапшоты ВМ
  - на основе снапшотов можно делать новые ВМ
  - много новых ВМ
    - не очень хочется в условиях ограниченных ресурсов

Моментальные снимки

- **На уровне гипервизора**
  - создать снапшоты ВМ
  - на основе снапшотов можно делать новые ВМ
  - много новых ВМ
    - не очень хочется в условиях ограниченных ресурсов
- **Но у нас нет гипервизора!**
- **У нас сертифицированная ОС :)**

Моментальные снимки

- На уровне ФС
  - LVM
  - ZFS
  - BTRFS

- Про перформанс мы не будем говорить
- Кому интересно – доклад Томаса Вондры на PGConf.EU 2023

Почему не LVM?

- Сложная настройка
  - Physical Volume (PV)
  - Volume Group (VG)
  - Logical Volume (LV)
- Монтирование снимков
- Резервируется место при создании снимка
- Нельзя делать снимок из снимка

Почему не ZFS?

- Не входит в ядро Linux
- Нужно подключать внешний репозиторий
- К снимку нет прямого доступа
  - нужно делать клон
- Нельзя удалить родительский снимок, пока не удалены все его клоны

BTRFS

PostgresPro

BTRFS (англ. B-Tree Filesystem) – файловая система для операционных систем Linux

- Основана на структурах B-деревьях
- Работает по принципу CoW (англ. Copy on Write)
- Входит в ядро Linux с версии 2.6.29
- Много багов исправлено, особенно в последних версиях

Утилиты BTRFS

PostgresPro

С дистрибутивом поставляются:

- ***mkfs.btrfs*** - *для* создания BTRFS на блочном устройстве
- ***btrfs*** - для управления и использования остальной функциональности

Дополнительно:

- **btrfs-convert** - преобразование из ext2/3/4 в btrfs
- **btrfstune** - тюнинг файловой системы
- …

Возможности BTRFS

- **Подтома (Subvolume)** - создание собственных независимых иерархий файлов/каталогов и пространств имен номеров индексных дескрипторов
- **Снимки (Snapshots)** - создание мгновенных снимков подтомов
- **Сжатие данных** на уровне файловой системы
- … и другие

Настройка BTRFS

- Создать раздел на блочном устройстве
- Отформатировать раздел в BTRFS
- Подмонтировать в каталог (например в /data)

Команды BTRFS для Snapshot Standby

PostgresPro

Посмотреть файловые системы BTRFS

```
btrfs filesystem show
```

Посмотреть информацию об точке монтирования

```
btrfs filesystem df $PATH
```

Посмотреть список подтомов

```
btrfs subvolume list -qtu $PATH
```

Посмотреть список снимков

```
btrfs subvolume list -qtus $PATH
```

Команды BTRFS для Snapshot Standby

PostgresPro

Создать подтом

```
btrfs subvolume create $PATH
```

Создать снимок (-r – только на чтение)

```
btrfs subvolume snapshot -r $PATH $SNAP_PATH
```

Разрешить редактирование снимка

```
btrfs property set -ts $SNAP_PATH ro false
```

Удалить подтом

```
btrfs subvolume delete $PATH
```

Схема стенда

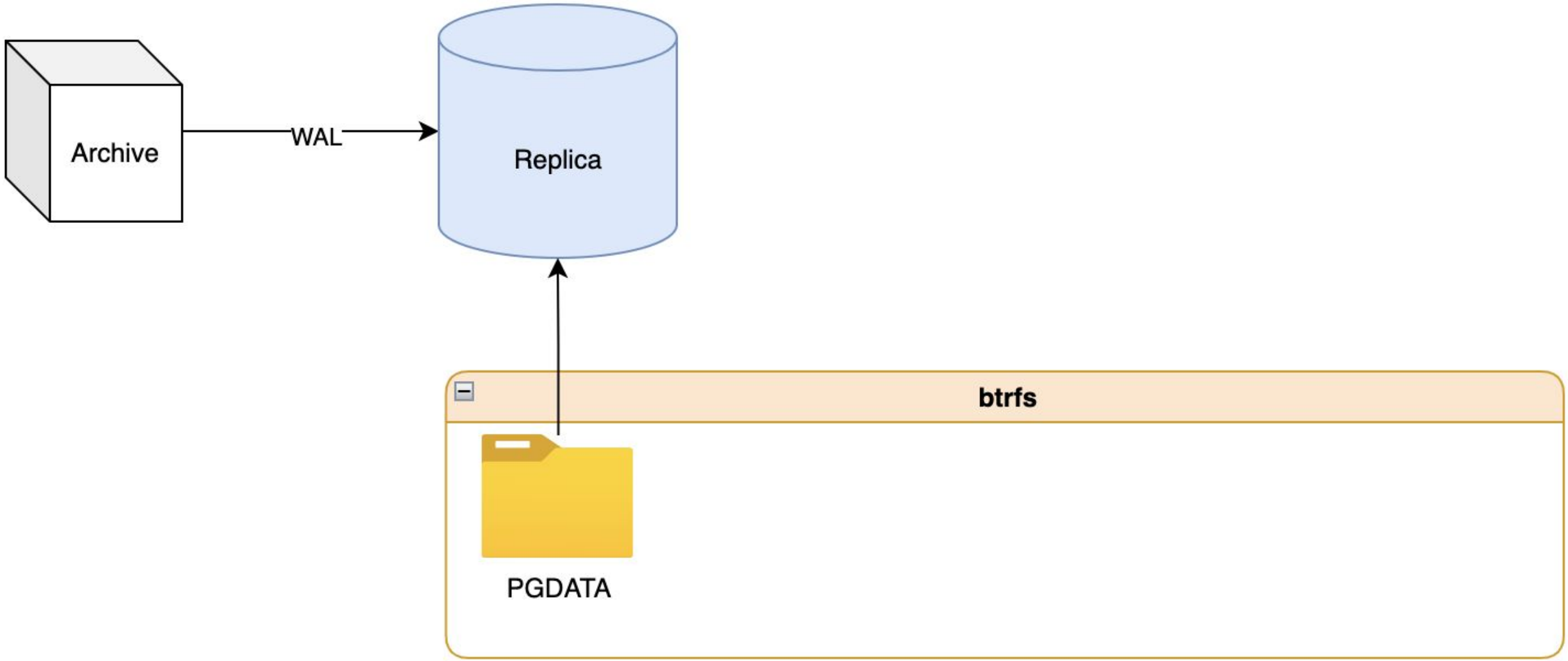


Схема стенда

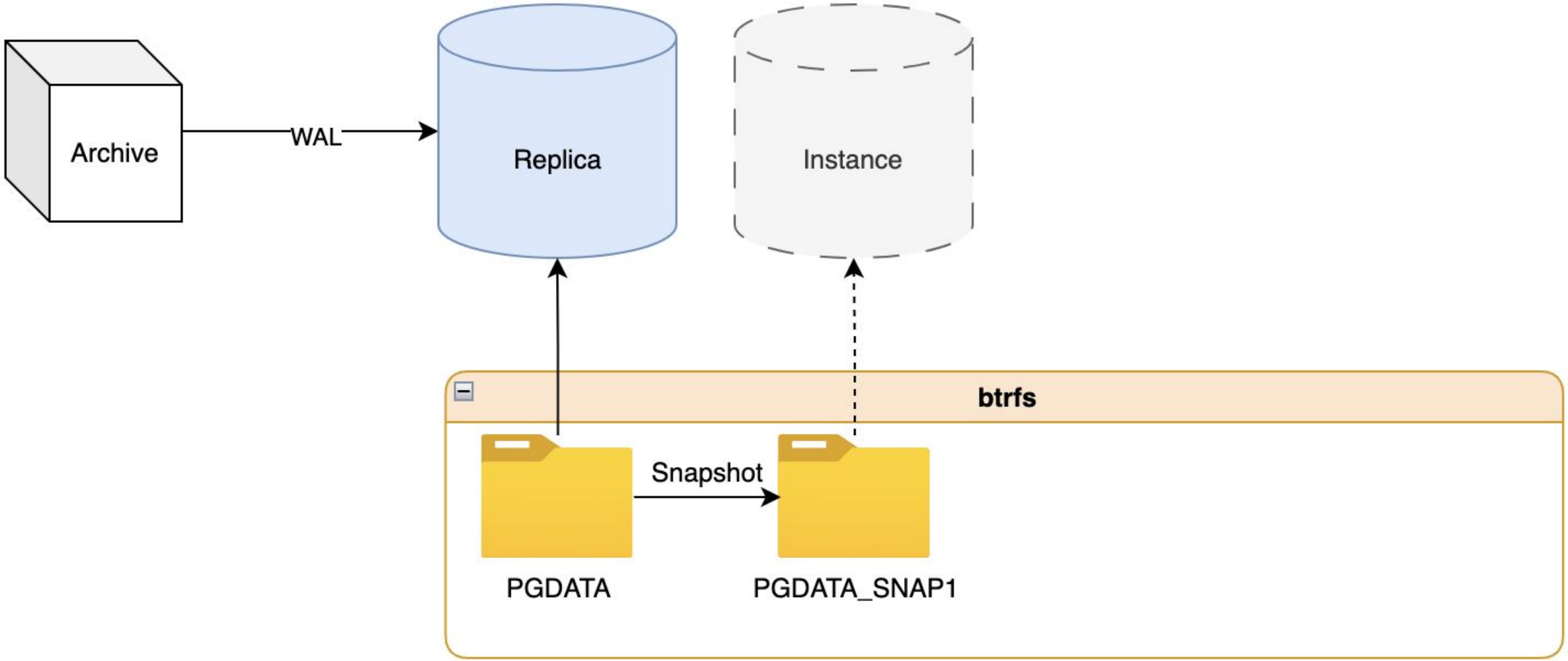
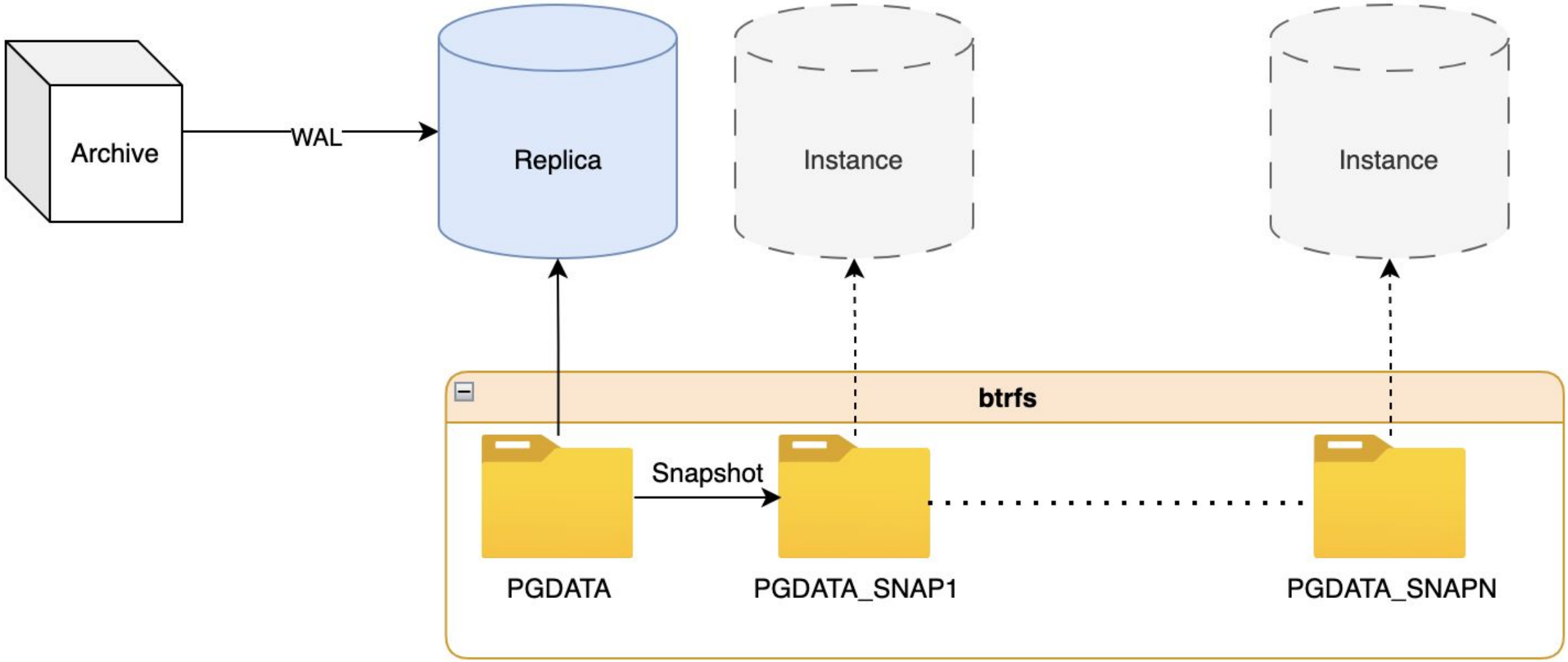


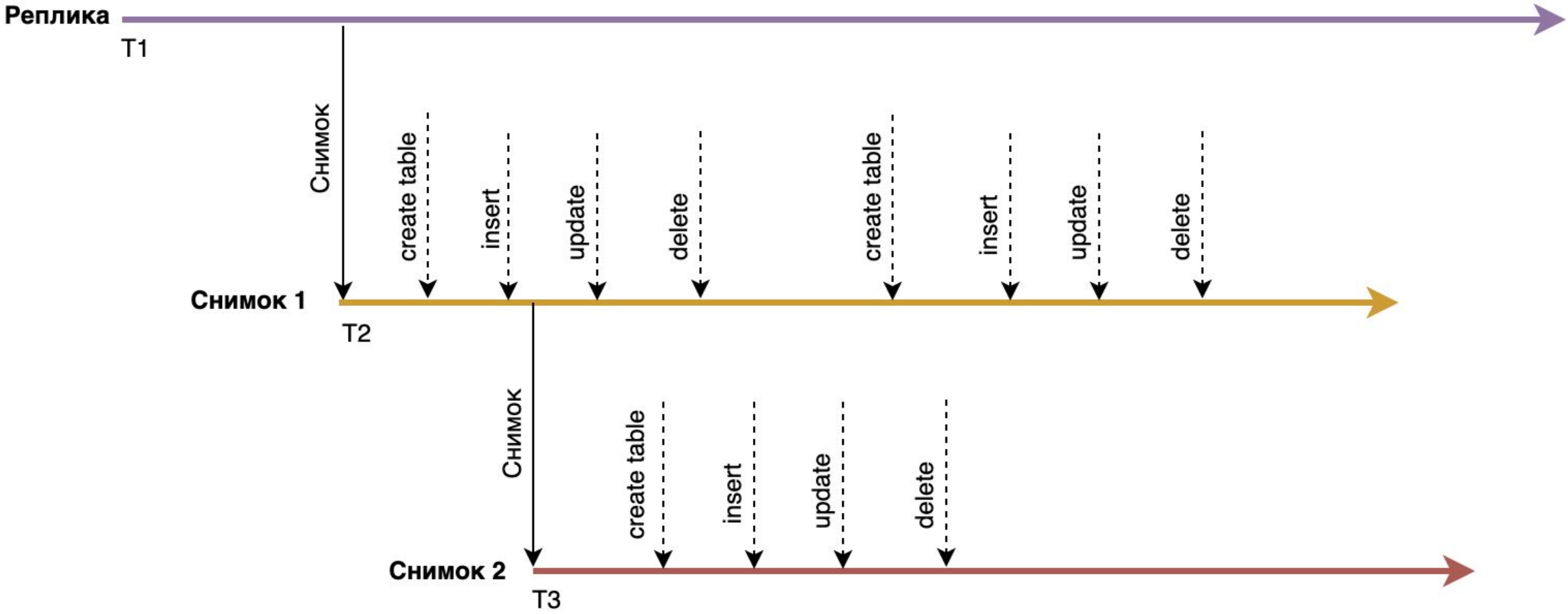
Схема стенда



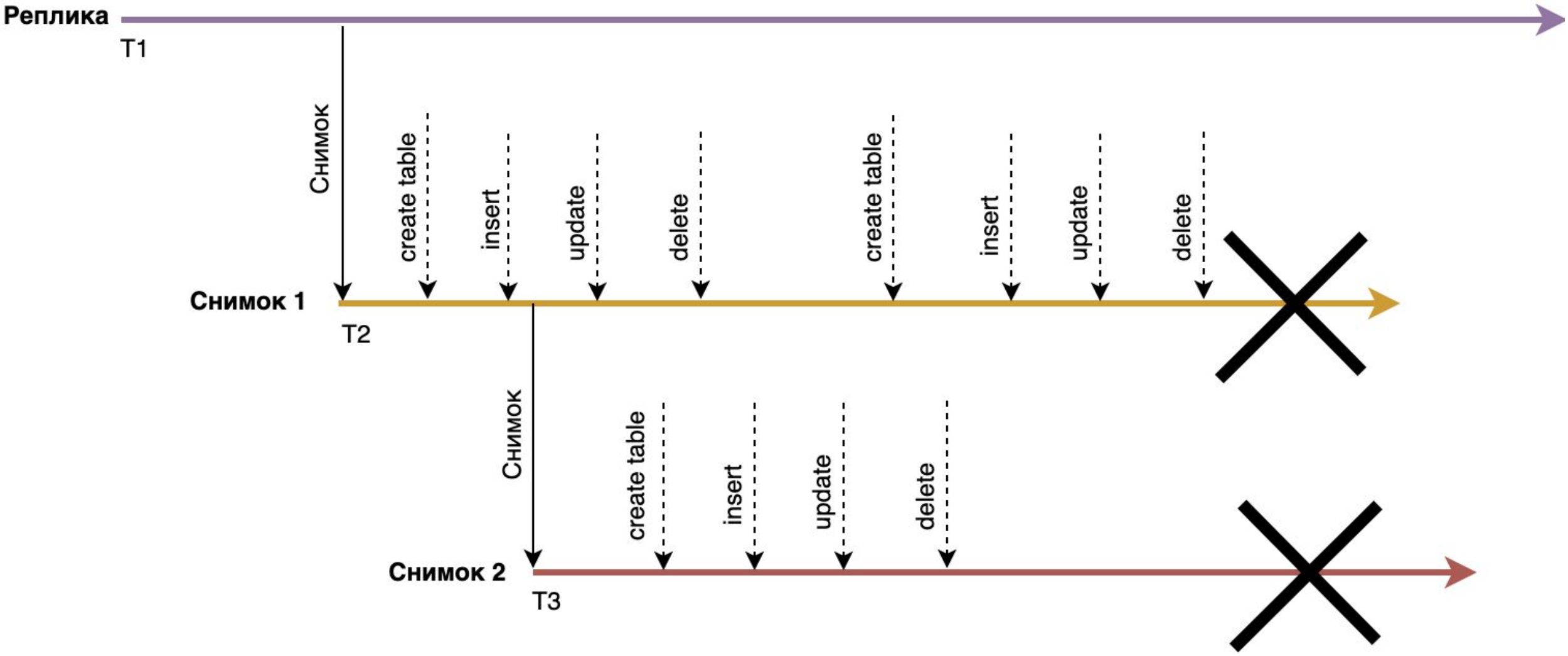
Сценарий работы со стендами

- Создадим каскад мгновенных снимков каталога PGDATA
- Запустим экземпляры СУБД на основе каждого снимка в режиме на запись
- Поработаем с экземплярами СУБД
- Удалим снимки

Визуализация создания стендов



Визуализация удаления стендов



Просмотр информации раздела btrfs

PostgresPro

Посмотрим разделы BTRFS

```
postgres@node2:/$ df -HT -t btrfs
Filesystem     Type   Size  Used Avail Use% Mounted on
/dev/vdb1      btrfs   65G   30G   32G  53% /data
```

Посмотрим структуру каталога */data*

```
postgres@node2:/$ du -hs /data/*
30G     /data/data_test
```

Создание снимка из реплики

PostgresPro

Создать снимок snap1

```
btrfs subvolume snapshot /data/data_test /data/data_snap1
```

Просмотр информации о подтомах

PostgresPro

Посмотрим структуру каталога */data*

```
postgres@node2:/$ du -hs /data/*
30G     /data/data_test
30G     /data/snap1
```

Посмотрим список снимков и подтомов

```
postgres@node2:/data$ sudo btrfs subvolume list -qtu /data
ID      gen     top level       parent_uuid     uuid            path
--      ---     ---------       -----------     ----            ----
257     2452    5               -               c814faa8-...    data_test
359     2452    5               c814faa8-...    0256a33d-...    snap1
```

Запуск экземпляра СУБД из снимка

PostgresPro

Перед запуском экземпляра СУБД на основе снимка snap1

- Редактируем postgresql.conf:
  - Изменяем порт
  - Убираем команды в *archive_command/restore_comand*
- Удаляем файл *standby.signal*
- Удаляем файл *postmaster.pid*

Запускаем экземпляра СУБД на основе снимка snap1

- Можем работать с БД

Создание снимка из снимка

PostgresPro

Создадим снимок snap2 из снимка snap1

```
btrfs subvolume snapshot /data/data_snap1 /data/data_snap2
```

Просмотр информации о подтомах

PostgresPro

Посмотрим структуру каталога */data*

```
postgres@node2:/$ du -hs /data/*
30G     /data/data_test
30G     /data/snap1
30G     /data/snap2
```

ls -l /data/snap2

Посмотрим список снимков и подтомов

```
postgres@node2:/data$ sudo btrfs subvolume list -qtu /data
ID      gen     top level       parent_uuid     uuid            path
--      ---     ---------       -----------     ----            ----
257     2452    5               -               c814faa8-...    data_test
359     2452    5               c814faa8-...    0256a33d-...    snap1
361     2452    5               0256a33d-...    8f67601d-...    snap2
```

Запуск экземпляра СУБД из снимка

PostgresPro

Перед запуском экземпляра СУБД на основе снимка snap2

- Изменяем порт
- Удаляем файл *postmaster.pid*

Запускаем экземпляра СУБД на основе снимка snap2

- Можем работать с БД

Просмотр информации раздела btrfs

Посмотрим разделы BTRFS

```
postgres@node2:/$ df -HT -t btrfs
Filesystem     Type   Size  Used Avail Use% Mounted on
/dev/vdb1      btrfs   65G   30G   32G  53% /data
```

Удаление стендов СУБД

PostgresPro

Удаление снимков

- Останавливаем экземпляры СУБД на основе снимков

- Удаляем снимки в любом порядке

```
btrfs subvolume delete /data/data_snap2
btrfs subvolume delete /data/data_snap1
```

Просмотр информации раздела btrfs

Посмотрим разделы BTRFS

```
postgres@node2:/$ df -HT -t btrfs
Filesystem     Type   Size  Used Avail Use% Mounted on
/dev/vdb1      btrfs   65G   30G   32G  53% /data
```

Посмотрим структуру каталога */data*

```
postgres@node2:/$ du -hs /data/*
30G     /data/data_test
```

Посмотрим список снимков и подтомов

```
postgres@node2:/data$ sudo btrfs subvolume list -qtu /data
ID      gen     top level       parent_uuid   uuid            path
--      ---     ---------       -----------   ----            ----
257     2452    5               -             c814faa8-...    data_test
```

Итоги

- BTRFS легко настраивается
- Для создания копии экземпляра PostgreSQL требуется всего **5 команды:**
  - Создать snapshot
  - Редактировать postgresql.conf (опционально)
  - Удалить standby.signal (опционально)
  - Удалить postmaster.pid
  - Запустить экземпляр

P.S. снимки != резервная копия

Итоги

- BTRFS входит в ядро Unix-подобных операционных систем
- Есть во всех дистрибутивах из коробки
- Не требует дополнительного монтирования
- Позволяет делать мгновенные снимки
- Позволяет создавать снимки из снимков
- Изменять данные в любом снимке
- Возможен запуск экземпляра СУБД на любом снимке ФС
- Обеспечивает мгновенное удаление снимков

Ссылки

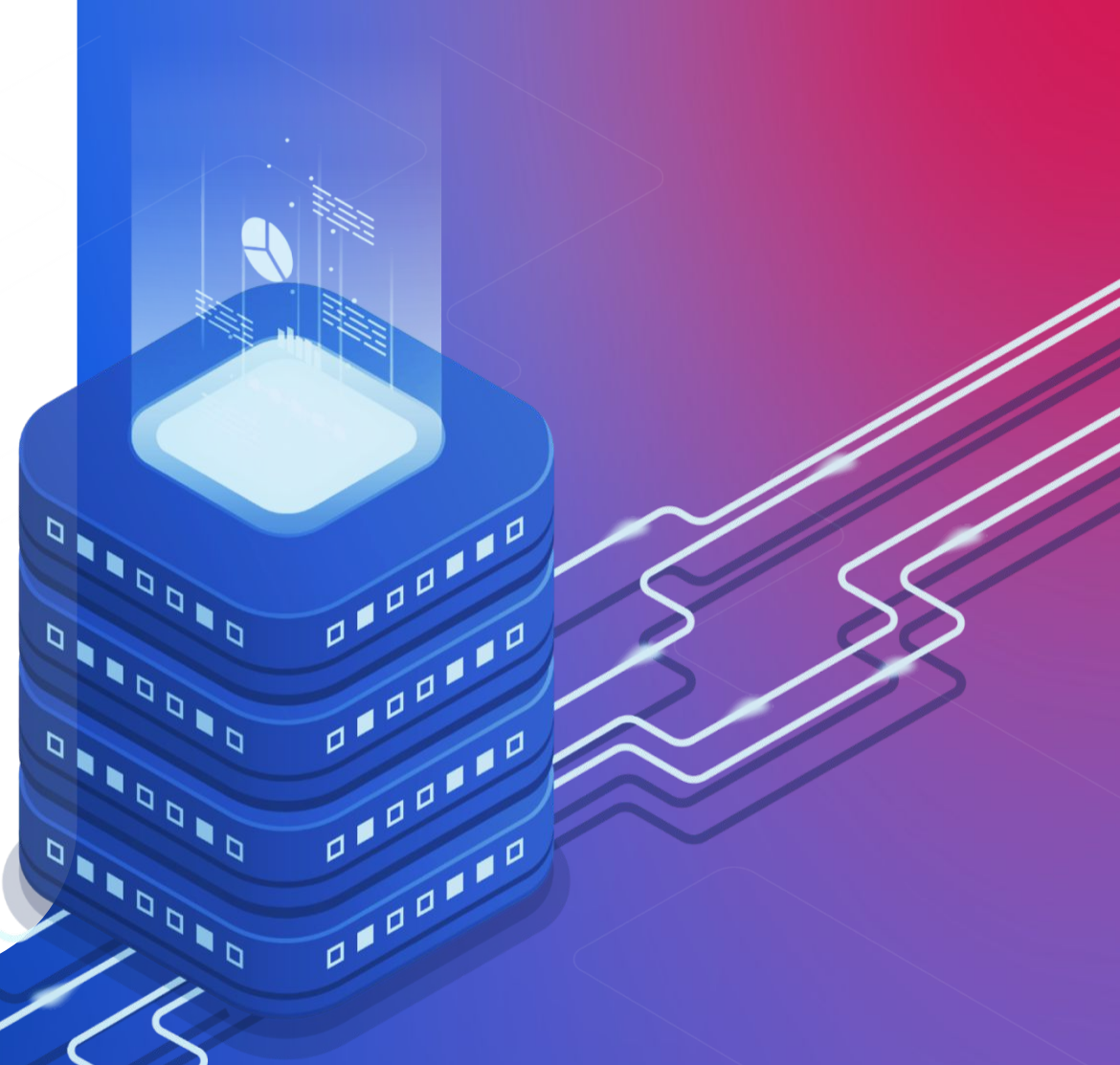
PostgresPro

Доклад Томаса Вондры на PGConf.EU 2023:

- https://www.postgresql.eu/events/pgconfeu2023/schedule/session/4670-postgres-vs-linux-filesystems

PostgresPro

Спасибо за внимание!



PostgresPro

Q&A