



Резервное копирование, валидация и верификация PostgreSQL

Смолкин Григорий
pgconf Siberia 2018

Что такое бэкап

- ~~1. Логическая копия данных~~
2. Физическая копия данных

Зачем нужен бэкап

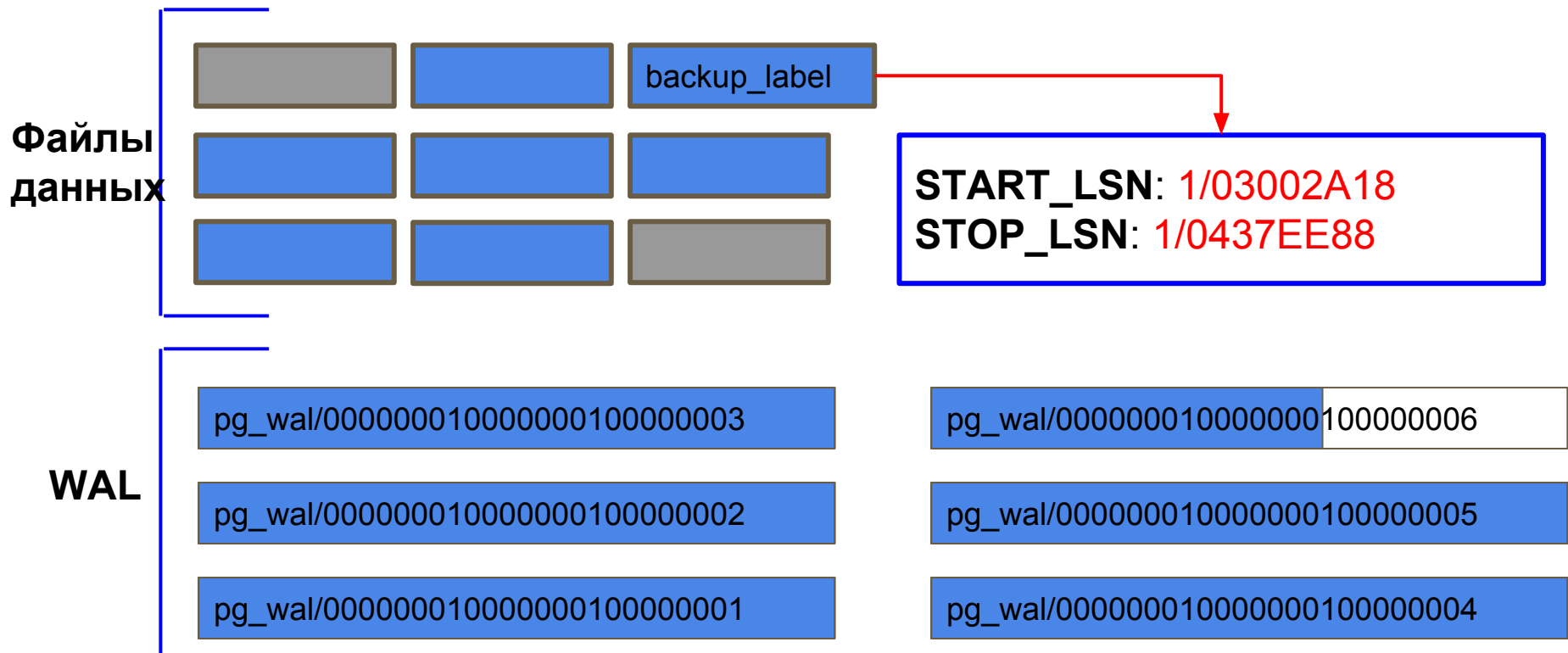
1. Защита от потери данных:

- Отказ оборудования
- Человеческий фактор
- Ошибки приложения
- Баги СУБД(!!!), glibc, OS, драйвера, прошивки

2. Перенос данных:

- Реплика
- Копия, например, dev-окружение

Как устроен бэкап в общем случае





PG_PROBACKUP

- Текущая stable версия: 2.0.24
- PostgreSQL \geq 9.5
- Linux support
- Windows support beta(!)
- https://github.com/postgrespro/pg_probackup
- C
- PostgreSQL license

PG_PROBACKUP zen of default

- не навреди
- не обмани
- будь предсказуем
- будь пессимистичен
- целостность первична



PG_PROBACKUP недостатки

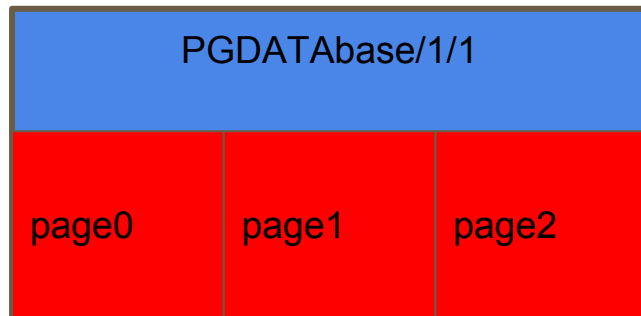
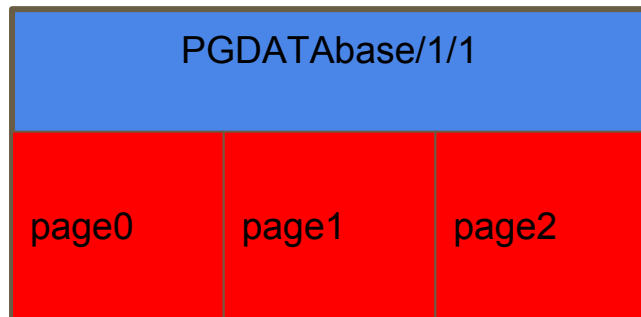
- удаленный бэкап отсутствует(beta)
- шифрование отсутствует

PG_PROBACKUP фичи

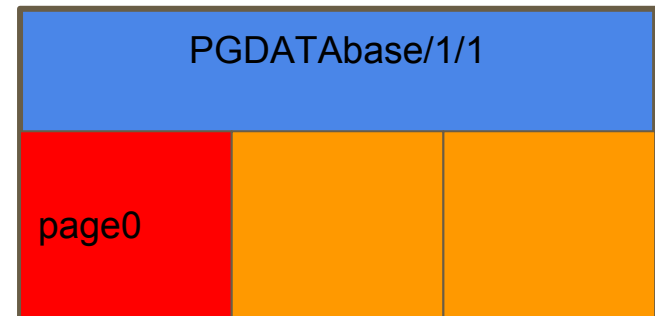
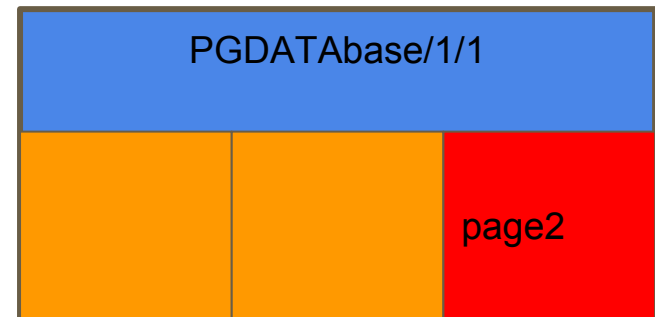
- инкрементальность на уровне блока (8kB). 3 режима инкрементального бэкапа: **DELTA, PAGE, PTRACK**
- многопоточность
- компрессия (zlib)
- политики хранения бэкапов (время, кол-во)
- физическая валидация блоков
- бэкап с реплики
- слияние бэкапов (**merge**).

DELTA

FULL

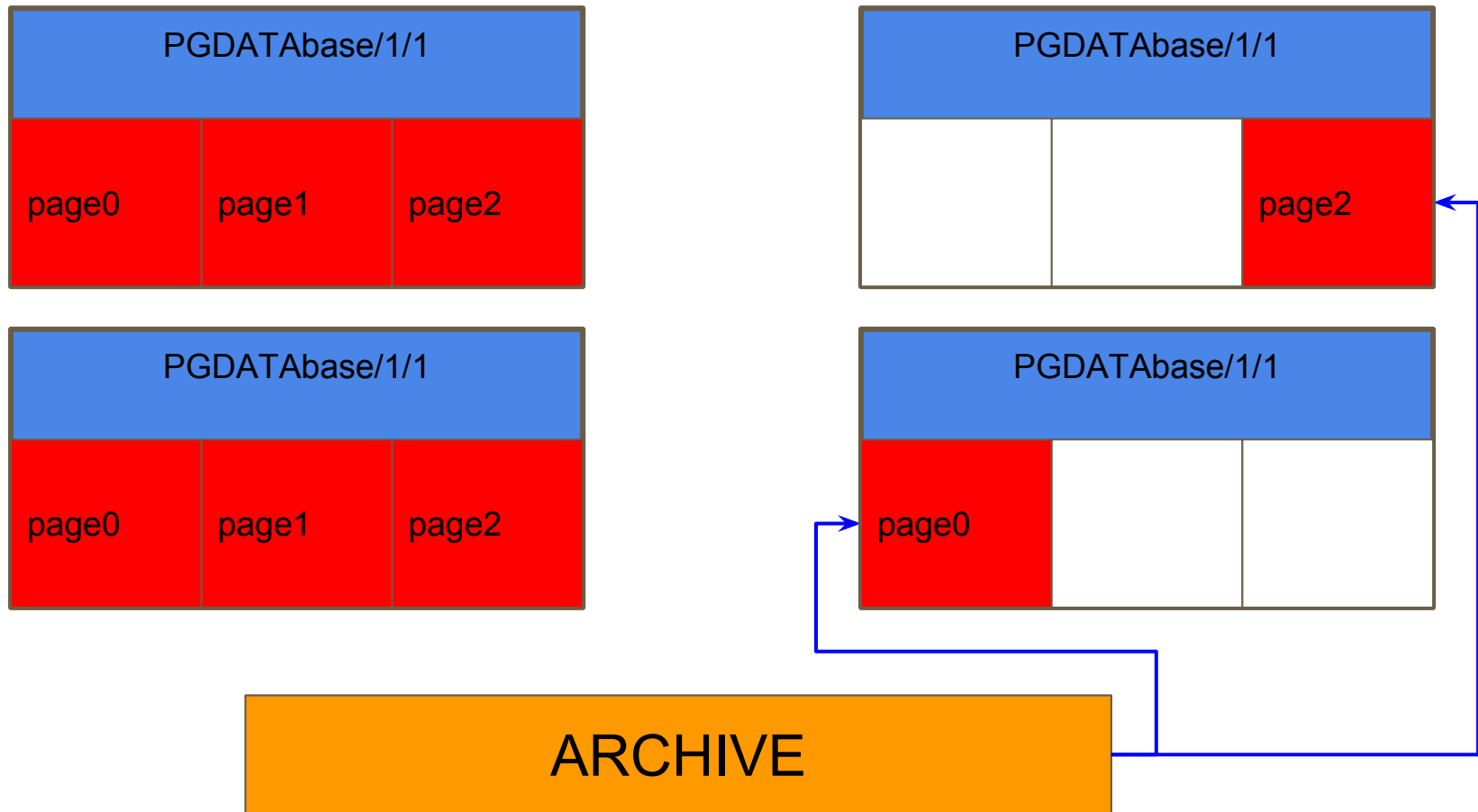


DELTA



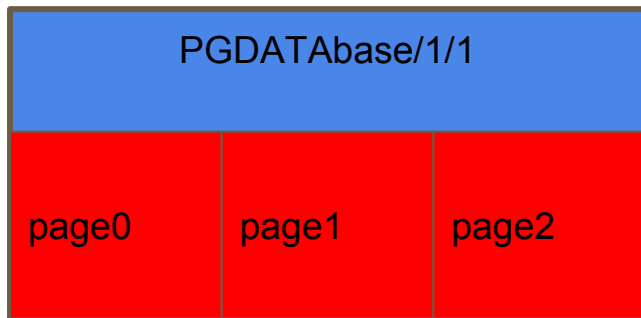
- I/O нагрузка на чтение аналогична FULL

PAGE

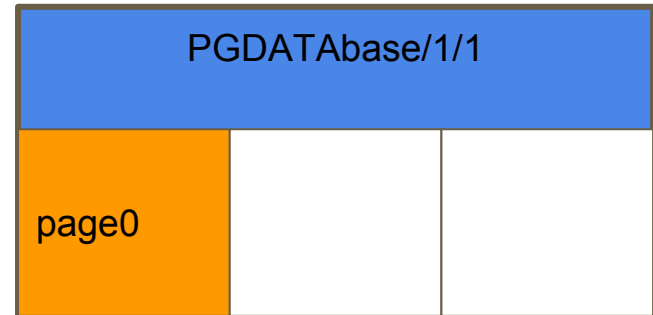
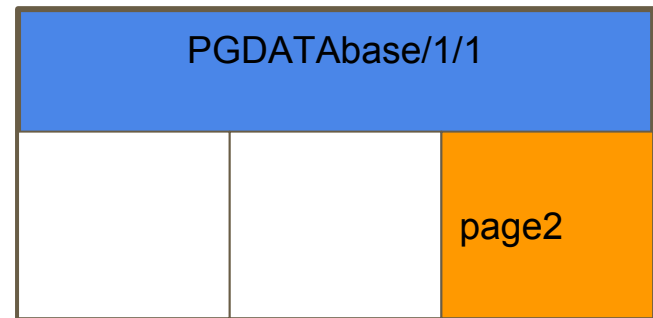


MERGE

FULL A

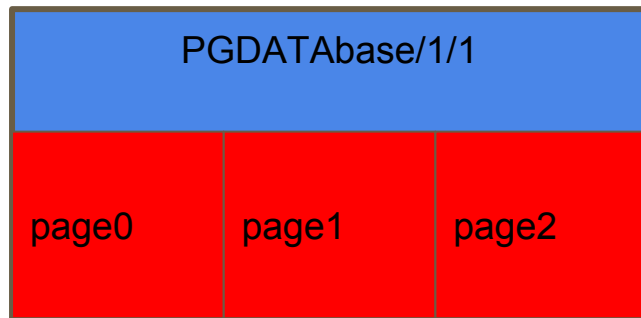


PAGE B



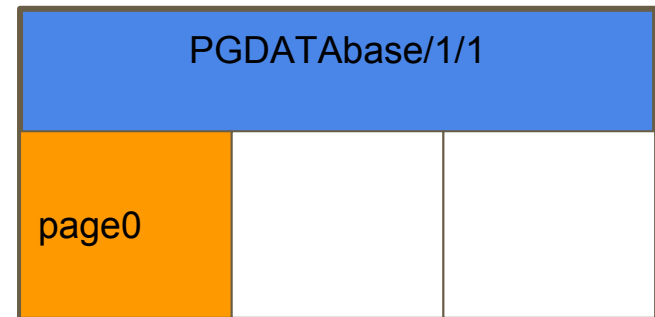
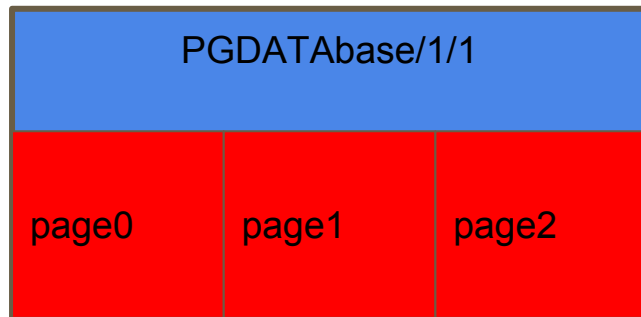
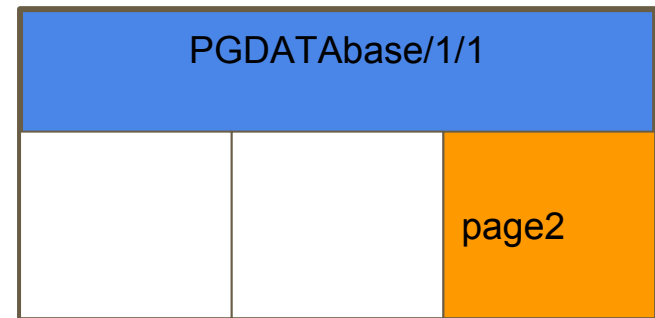
MERGE

FULL A



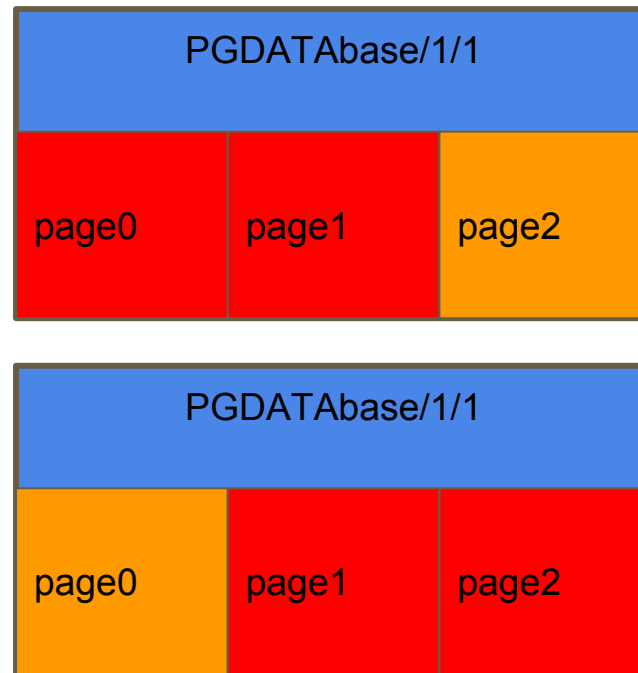
MERGE

PAGE B



MERGE

FULL B



- Один раз сделав полный бэкап, можно “тащить” его за собой, вмерживая в него инкременты



Главная фишка PG_PROBACKUP

- Защищаем пользователя от “нюансов” PostgreSQL
- Защищаем пользователя от багов PostgreSQL
- Паранойя



Нюансы и баги PostgreSQL

- Недостижение точки восстановления при PITR не считается ошибкой!
- Зануленный блок считается валидным
- Блоки из будущего в 9.6.10, 9.5.14, 10.5. Коммит **8d68ee6**
- физическая целостность != логическая целостность

Физическая валидация

1. Проверка чексуммы
2. Проверка значений в заголовках страниц на “нормальность”:
 - читаемость заголовка
 - `pd_lower/pd_lower` в разумных пределах
 - etc
3. Дешево
 - Физической валидации недостаточно!

Физическая валидация

Как обнаружить?

1. Валидный блок из будущего/прошлого/другой БД.
2. Зануленный блок.
3. Нарушение связности между таблицей и индексом.
4. Нарушение связности между таблицами.

Логическая валидация.

1. Проверка логической консистенности между таблицей и индексом
2. Проверка логической консистенности между таблицей и таблицей(FK)

```
pg_dumpall > /dev/null
```

AMCHECK



1. <https://github.com/petergeoghegan/amcheck>
2. <https://www.postgresql.org/docs/10/amcheck.html>
3. Отдельный пакет amcheck в PGDG репозитории.

Как работает amcheck

1. **bt_index_check(index regclass, heapallindexed)**
 2. **bt_index_parent_check(index regclass, heapallindexed)**
- В планах интеграция pg_probackup с amcheck

Итого

1. Бэкап считается валидным, он прошел **физическую и логическую** валидации.
2. Логическая валидация подразумевает **восстановление** из бэкапа.
3. Для логической валидации желательно использовать **amcheck**



Вопросы?