

# PG\_PROBACKUP 2.2

Смолкин Г.Н.

# Что такое PG\_PROBACKUP?

**HE Unix WAY**



# Что такое PG\_PROBACKUP?

- **C + PostgreSQL internal API**
  - ReceiveXlogStream
  - xlogreader
  - datapagemap
  - кроссплатформенные примитивы
  - etc

# Тесты: 62% кодовой базы

## pg\_probackup

Backup and recovery manager for PostgreSQL

backup

restore

archiving

postgresql

wall

recovery

incremental-backups

 Python

 26

 197

 39

 0

Updated yesterday

# Каталог резервных копий

ID	Recovery Time	Mode	WAL Mode	TLI	Time	Data	WAL	Zratio	Status
Q0RCQX	2019-11-10 17:44:48	FULL	STREAM	1/0	4m	761MB	16MB	11.00	OK
Q0RC8L	2019-11-10 17:32:18	FULL	STREAM	1/0	3m	8GB	64MB	1.00	OK

# WAL



# WAL and backups

-----b1-----b2----->

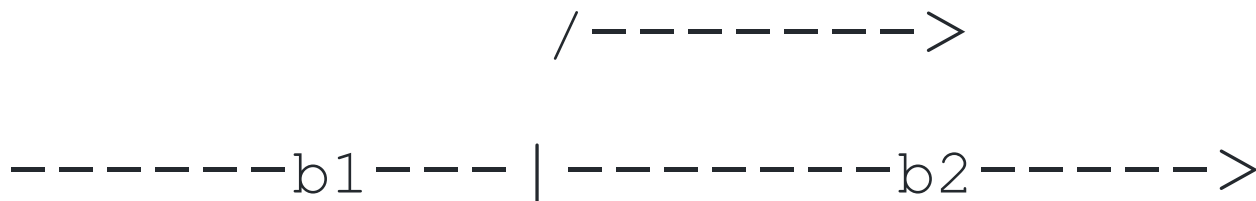
# WAL

/ DROP TABLE ... ОЙ

-----b1-----|-----b2----->



# PITR/standby promotion



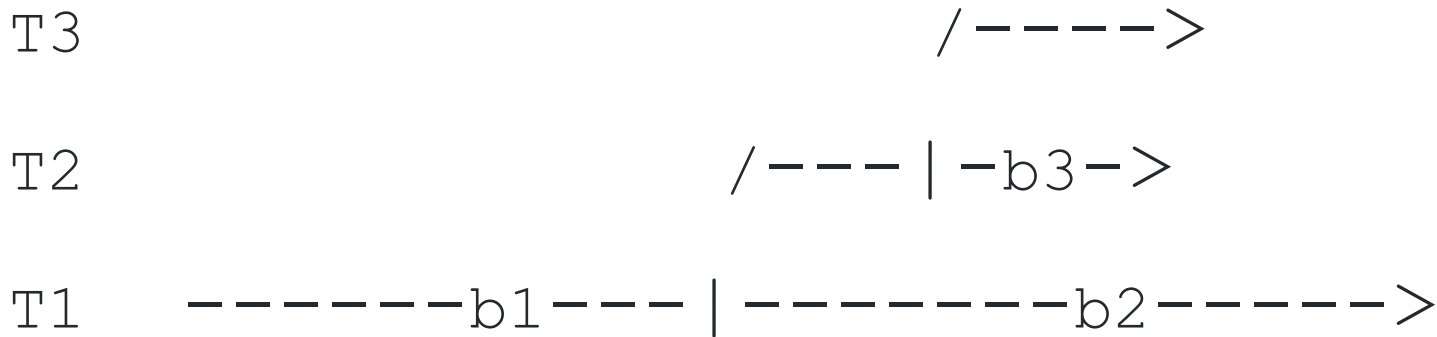
# Линии времени (timelines)



# WAL timelines



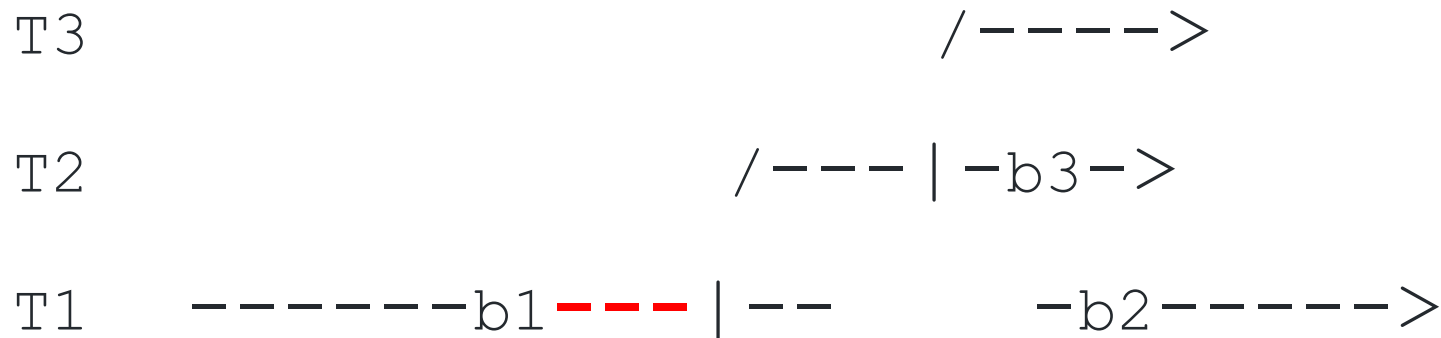
# WAL timelines



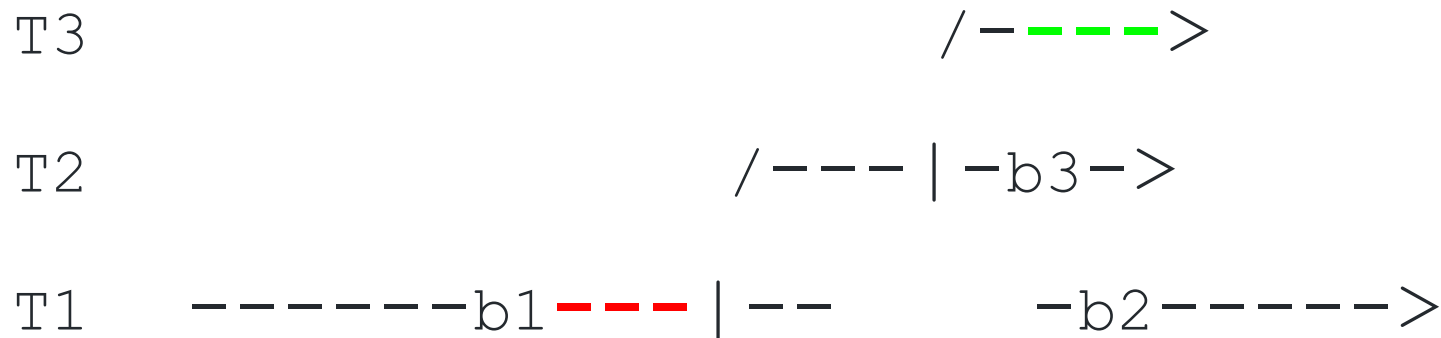
# Порча файлов



# Пропажа файлов



# “Чужие” файлы



# Каталог архива WAL

```
$ pg_probackup show --archive
```

```
ARCHIVE INSTANCE 'node'
```

```
=====
```

TLI	Parent	Switchpoint	Min Segno	Max Segno	Size	Zratio	N backups	Status
3	2	0/10DC3F38	00000000000000010	0000000000000001B	4911kB	30.02	0	DEGRADED
2	1	0/B000000	0000000000000000B	00000000000000011	4171kB	27.50	1	OK
1	0	0/0	00000000000000001	0000000000000000A	7526kB	21.77	2	OK

```
=====
```



# Каталог архива WAL

```
$ pg_probackup show --archive --format=json
```

```
"timelines": [  
  {  
    "tli": 2,  
    "parent-tli": 1,  
    "switchpoint": "0/16000000",  
    "min-segno": "00000000000000016",  
    "max-segno": "00000000000000019",  
    "n-segments": 3,  
    "size": 193376,  
    "zratio": 228.65,  
    "closest-backup-id": "Q0U6XU",  
    "status": "DEGRADED",  
    "lost-segments": [{"begin-segno": "00000000000000017", "end-segno": "00000000000000017"}],  
    "backups": []  
  },  
]
```

# Политика удержания WAL

b1-----b2-----b3-----b4-----b5----->

# Глубина хранения WAL (wal depth)

```
$ pg_probackup delete --delete-wal --wal-depth=2
```

b1            b2            b3            b4-----b5----->

```
INFO: On timeline 1 WAL segments between 0000000000000001 and 0000000000000014  
will be removed
```

# Глубина хранения WAL (wal depth)

```
$ pg_probackup delete --delete-wal --wal-depth=2
```

**t2**

/----->

**t1**

b1

b2

|

b3

b4

b5

>

```
LOG: Timeline 2 to stay reachable from timeline 1 protect from purge WAL  
interval between 0000000000000000B and 00000000000000010 on timeline 1
```

# Время жизни резервной копии (TTL)

- Исключение резервной копии из установленной политики удержания

```
=====
ID          Recovery Time          Mode WAL Mode TLI Time Data WAL Zratio Status
=====
Q0RC8L     2019-11-11 17:32:18     FULL ARCHIVE 1/0 3m 8GB 64MB 1.00 OK
Q0RWCY     2019-11-10 04:43:41     FULL ARCHIVE 1/0 3m 8GB 48MB 1.00 OK
```

# Васкуп TTL, интерфейс

## Установка :

```
$ pg_probackup set-backup -i Q0RWCY --ttl=30d
$ pg_probackup set-backup -i Q0RWCY --expire-time='2020-01-01 00:00:00'
$ pg_probackup backup --ttl=30d ...
```

## Выключение :

```
$ pg_probackup set-backup --ttl=0 ...
```

# Время жизни резервной копии (TTL)

```
$ pg_probackup backup --delete-expired --retention-redundancy=2 ...
```

```
=====
ID          Recovery Time          Mode  WAL Mode  TLI  Time  Data  WAL  Zratio  Status
=====
Q0U289  2019-11-12 08:45:41  FULL  ARCHIVE  1/0   3m   8GB  48MB   1.00  OK
Q0RC8L  2019-11-11 17:32:18  FULL  ARCHIVE  1/0   3m   8GB  64MB   1.00  OK
Q0RWCY  2019-11-10 04:43:41  FULL  ARCHIVE  1/0   3m   8GB  64MB   1.00  OK
```

# Время жизни резервной копии (TTL)

```
$ pg_probackup backup --delete-expired --retention-redundancy=2 ...
```

```
=====
ID          Recovery Time          Mode  WAL Mode  TLI  Time  Data  WAL  Zratio Status
=====
Q0U2AD     2019-11-13 09:47:09    FULL  ARCHIVE   3/0   3m   8GB  16MB  1.00  OK
Q0U289     2019-11-12 08:45:41    FULL  ARCHIVE   1/0   3m   8GB  48MB  1.00  OK
Q0RWCY     2019-11-10 04:43:41    FULL  ARCHIVE   1/0   3m   8GB  64MB  1.00  OK
=====
```



# Время жизни резервной копии (TTL)

```
$ pg_probackup show -i Q0RC8L | grep expire-time  
expire-time = '2020-01-01 00:00:00'
```

```
$ pg_probackup set-backup -i Q0RC8L --ttl=0  
INFO: Backup Q0RC8L is unpinned
```

# Выборочное восстановление

Name		Size
db1		100 MB
db2		500 MB
db3		5 GB
db4		50 GB
db5		300 GB

# Partial Restore

```
$ pg_probackup restore --db-include=db1 --db-include=db2
```

```
$ pg_probackup restore --db-exclude=db3 --db-exclude=db4 --db-exclude=db5
```

```
VERBOSE: Exclude file due to partial restore: "base/24600/1418"
```

# Как это работает?

1. Путь до файла данных файла содержит dbOid:

```
base/24600/24588
```

2. Маппинг dname к dbOid в каждой резервной копии:

```
{"dbOid":"13627", "datname":"postgres"}  
{"dbOid":"24598", "datname":"db1"}  
{"dbOid":"24599", "datname":"db2"}  
{"dbOid":"24600", "datname":"db3"}
```

3. Восстанавливаем ненужные файлы как файлы нулевого размера

# Как это работает?

- Если попытаться подключиться к db5:

```
FATAL: "base/24598" is not a valid data directory
```

```
File "base/24598/PG_VERSION" does not contain valid data.
```

```
You might need to initdb.
```

- Убираем за собой:

```
DROP DATABASE db3;
```

```
DROP DATABASE db4;
```

```
DROP DATABASE db5;
```

## Планы?

- Больше компрессии: zstd + lz4
- Новый формат хранения
- S3
- Шифрование
- Дифференциальная резервная копия
- Инкрементальное восстановление

# Прочее

- Github:

[https://github.com/postgrespro/pg\\_probackup](https://github.com/postgrespro/pg_probackup)

- Recovery target 'latest':

<https://commitfest.postgresql.org/26/2347/>

- Strict recovery:

<https://commitfest.postgresql.org/25/2284/>

**Вопросы?**