



PG BootCamp Russia 2025 Ekaterinburg  
[PGBootCamp.ru](http://PGBootCamp.ru)

# Xid, отладка, 2 фикса.

Разработка и отладка системы 64-битной  
идентификации транзакций

Евгений Воропаев, старший разработчик «Тантор Лабс»

# Немного обо мне

- › Кандидат физ.-мат. наук (физика конденсированного состояния вещества).
- › Занимаюсь разработкой Postgres с 2023 г.
- › Вклад в сообщество:
  - Xid64 для PG18dev
  - Bootstrap SLRU (рефакторинг, Ready for committer)
- › Хобби: альпинизм, гитара, IT-стартапы.



Автореферат диссертации



Xid64 для PG18dev

Профессионально занимаюсь разработкой ПАК с 2006 года.

- 2006-2007 - охранно-пожарные системы
- 2006 -2010 - комплексы оптической спектрометрии
- **2010 - 2015 - противоугонные системы “Microlock”**
- 2014 - 2018 - АСУТП и РК
- 2018 - 2020 - интернет вещей в освещении
- **2020 - 2021 - автоматический спортивный хронометраж “Торок”**
- 2021 - 2022 - контроллеры доступа “Apollo”
- 2022 - 2023 - система виртуализации “Брест”
- с 2023 - разработка СУБД Postgres/Tantor



# План на следующие полчаса

О чём поговорим. Теория.

1. Термины
2. 32-битная система идентификации транзакций
3. 64-битная система идентификации транзакций

Чему научимся. Практика.

1. «Пришла беда откуда ЖДАЛИ»  
Включение расширенных тестов при репликации
2. «Ну, всё понятно! Но что конкретно?»  
Проблемы консистентности страниц при PRUNE/FREEZE
3. «Сапожник без сапог?»  
Отладка без отладчика
4. «Вскрытие покажет!»  
Анализ багов
5. «Staying alive!»  
Ремонт

# Термины

Счётчик  
транзакций

Системы  
идентификации  
транзакций

64-битный  
идентификатор  
транзакций

FullTransactionID

TransactionID

ShortTransactionID

Идентификатор  
транзакций

Xid64

Xid32

32-битный  
идентификатор  
транзакций

# Термины

## Система идентификации транзакций

это набор

- алгоритмов,
- типов переменных
- и самих переменных,

реализующий

- уникальную идентификацию транзакций
- и функционал определения порядка следования транзакций во времени.

## Счётчик транзакций

алгоритм, реализующий функционал получения следующего свободного идентификатора транзакций.

## Идентификатор транзакции

число, однозначно идентифицирующее транзакцию. Также этот термин используем для переменной, хранящей это число.

# Идентификатор транзакции

В оригинальном Postgres используются два типа идентификаторов транзакций:

## **TransactionID**

идентификатор транзакции, 32 - битный

## **FullTransactionID**

полный идентификатор транзакции, 64-битный

**Какой тип применяется при определении порядка следования транзакций?**

**Какой тип используется в кортежах?**

**Что инкрементируется в счётчике транзакций?**

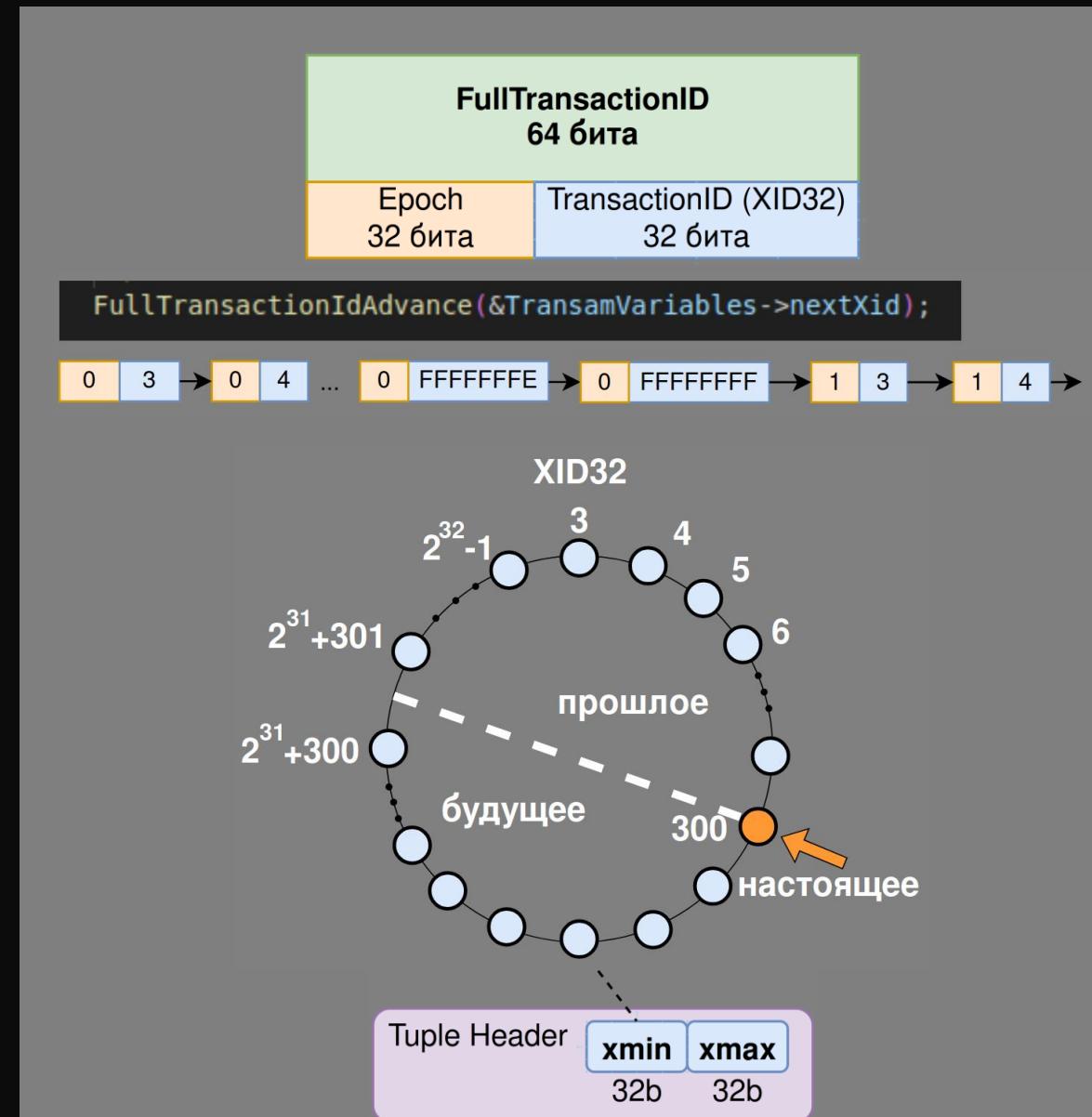
# Счётчик транзакций VS Идентификатор транзакции

**TransactionID (XID32)**  
идентификатор транзакции. 32-битный

**FullTransactionID**  
полный идентификатор транзакции  
64-битный

**Состав счётчика транзакций**  
1) переменная-счётчик (64б)  
2) вспомогательные переменные  
3) алгоритм инкремента

**Где хранится/используется Xid32?**  
- в самой транзакции  
- в кортежах!



# Wraparound

Причины:

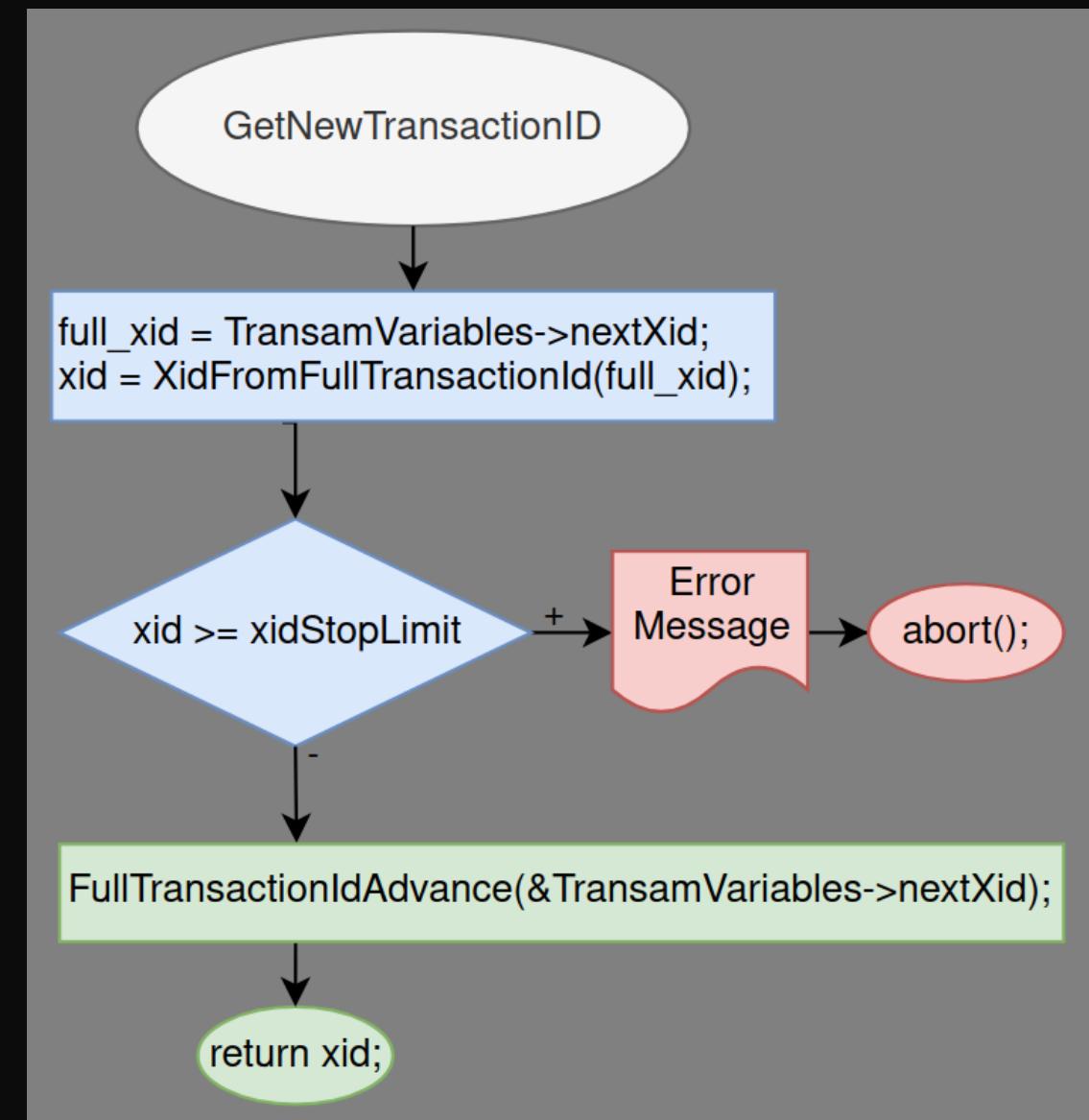
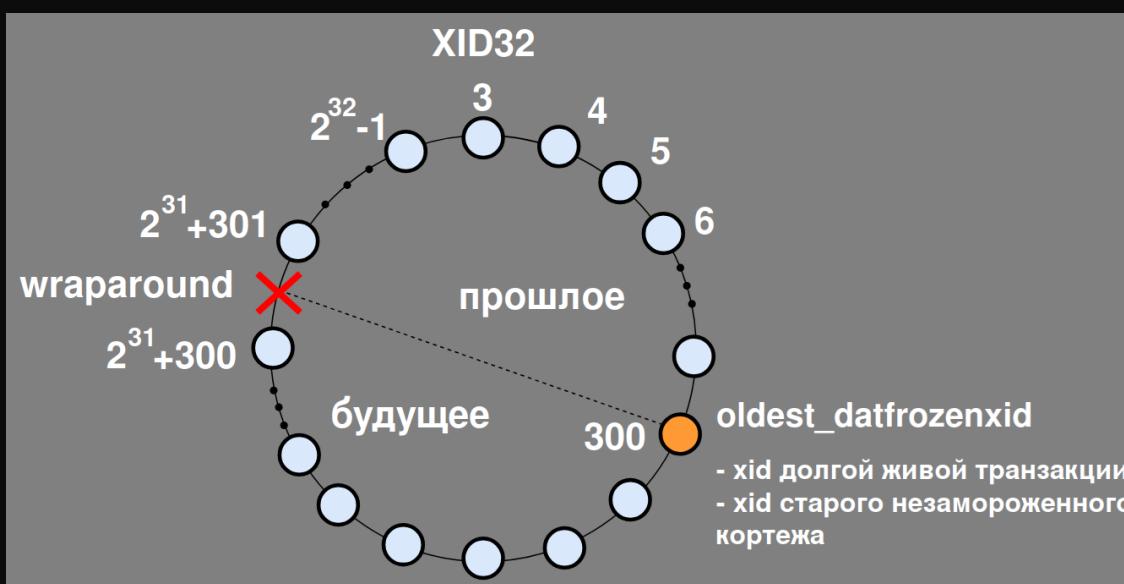
- долгие транзакции
- необработанные VACUUM-ом кортежи

Итог - остановка кластера.

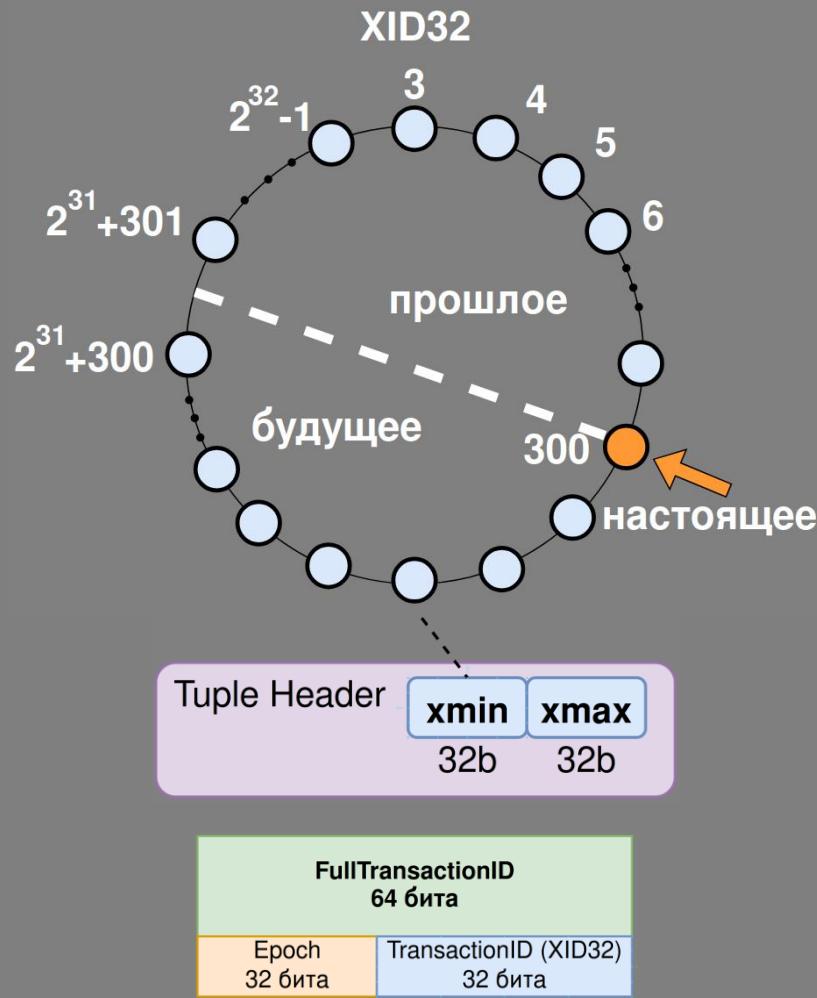
Оценка периодичности проблемы:

Xid32:  $2^{32}-3 = 4\ 294\ 967\ 293$  идентификаторов транзакций

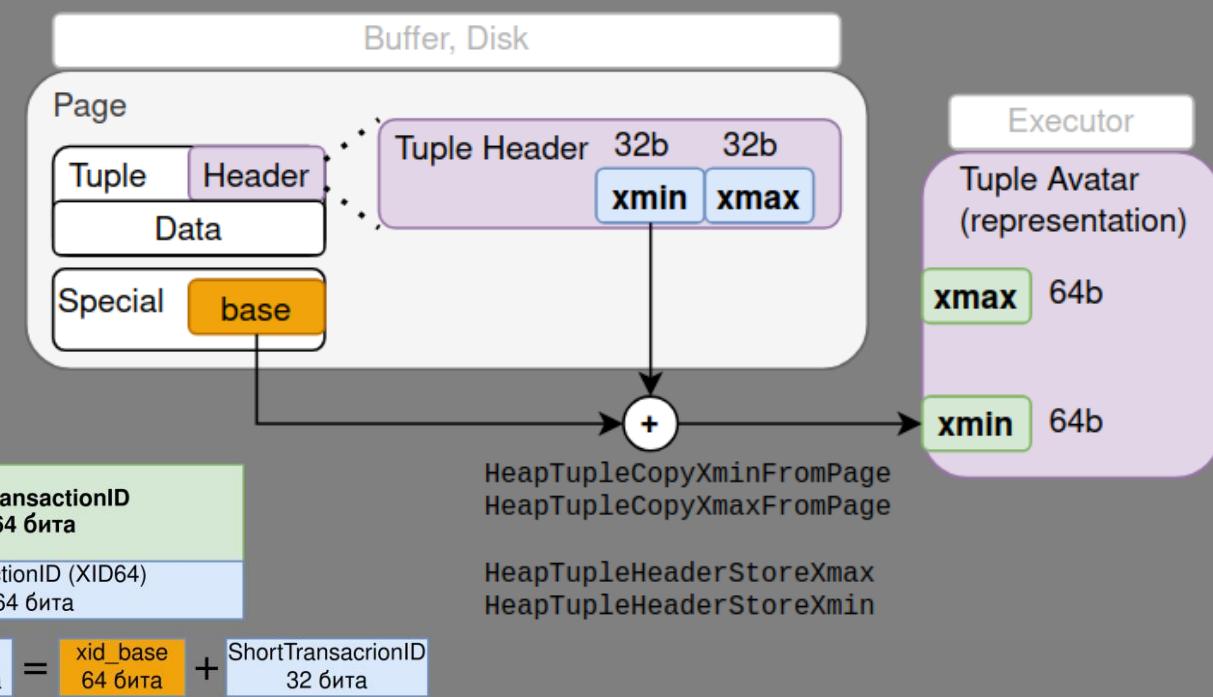
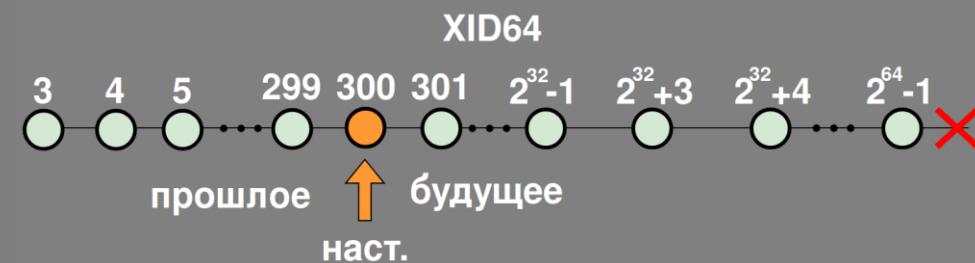
20 000 транз/с => 2,5 сут => Wraparound



# Xid32



# Xid64



# Доработка патча xid64 для текущего мастера

**Дано:**

- 1) патч версии 52, совместимый с master-веткой где-то между PG16 и PG17.
- 2) master-ветка версии PG18dev

3) фрагменты, вынесенные в отдельные патчи GUC64, MXIDOFF64 и др.

**Задача:**

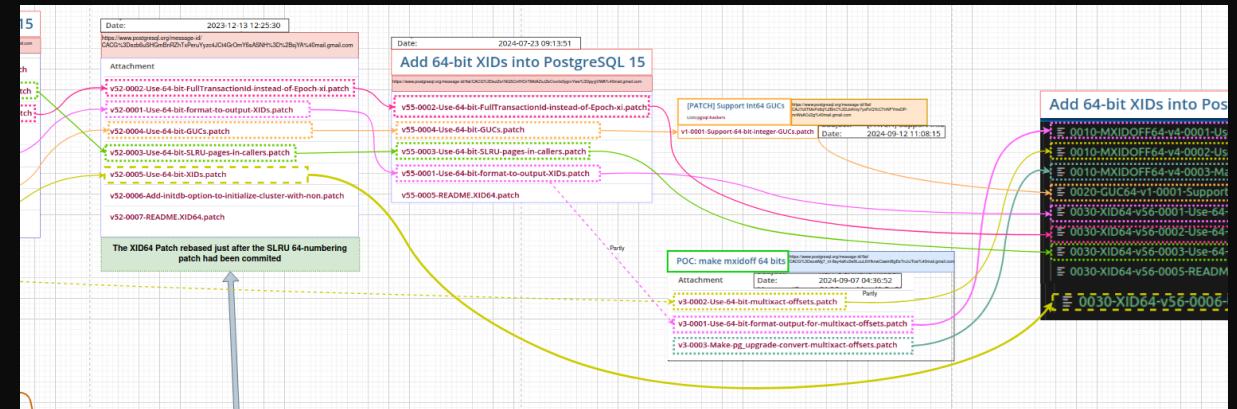
реализовать XID64 для текущего PG18dev

**Сложности (формальные):**

- запутанное генеалогическое древо патча;
- существенные изменения кодовой базы PG.

**Сложности (функциональные):**

- **изменение WAL-записей prune и freeze;**
- баги версии 52;
- и мн. др.



```
evoro:lin-ev:~/tantor/dv/pg18xid64PS(ps/pg18xid64)$ git log
commit f83d709760d84253b539ee454ce9ce1ceb9ad9ac
Author: Heikki Linnakangas <heikki.linnakangas@iki.fi>
Date: Mon Mar 25 14:59:58 2024 +0200

    Merge prune, freeze and vacuum WAL record formats
```

```
evoro:lin-ev:~/tantor/dv/pg18xid64PS(ps/pg18xid64)$ git log
commit 6dbb490261a6170a3fc3e326c6983ad63e795047
Author: Heikki Linnakangas <heikki.linnakangas@iki.fi>
Date: Wed Apr 3 19:32:28 2024 +0300

    Combine freezing and pruning steps in VACUUM
```

# Доработка патча xid64 для текущего мастера

1. пошагово применяем патч за патчем (11 патчей)
2. большие патчи разделяем на мелкие фрагменты и также пошагово применяем (>200 файлов)
  - устранием формальные несоответствия кода
  - сложные изменения базы пытаемся проанализировать и адаптировать вдумчиво (= закладываем новые баги)
3. собираем, устранием ошибки сборки
4. тестируем
5. если есть ошибки в тестах, отлаживаем, устранием ошибки, возвращаемся в пункт 3
6. видим сообщение “All test successful!”
7. не видим сообщение “Error”
8. Радуемся! 😊
9. Вспоминаем, что не включили “wal\_check\_consistency”  


```
v60-0001-Use-64-bit-format-output-for-mu
v60-0002-Use-64-bit-multixact-offsets.pa
v60-0003-Make-pg_upgrade-convert-multixa
v60-0004-Get-rid-of-MultiXactMemberFreeze
v60-0005-Support-64-bit-integer-GUCs.pat
v60-0006-Use-64-bit-format-to-output-XID
v60-0007-Use-64-bit-FullTransactionId-in
v60-0008-Use-64-bit-SLRU-pages-in-caller
v60-0009-Use-64-bit-XIDs.patch
v60-0010-Add-initdb-option-to-initialize
v60-0011-README.XID64.patch
v60-0012-Fixed-the-stale-xid64-problem-a
v60-0013-Fixed-the-error-that-occurred-w
v60-0014-Removed-the-management-of-repair
v60-0015-Fixed-the-error-An-unsaved-xmax
```

```
246 files changed, 8696 insertions(+),
```

```
All tests successful.
```

# Готовимся к практике

На вашем компьютере НАСТРОЕНО окружение для сборки и  
тестирования текущего master (PG18dev)

[https://github.com/e-voro/pg18xid64v60\\_ekat\\_practice](https://github.com/e-voro/pg18xid64v60_ekat_practice)

git clone [https://github.com/e-voro/pg18xid64v60\\_ekat\\_practice](https://github.com/e-voro/pg18xid64v60_ekat_practice)



Практика. Материалы

# Из жизни страниц

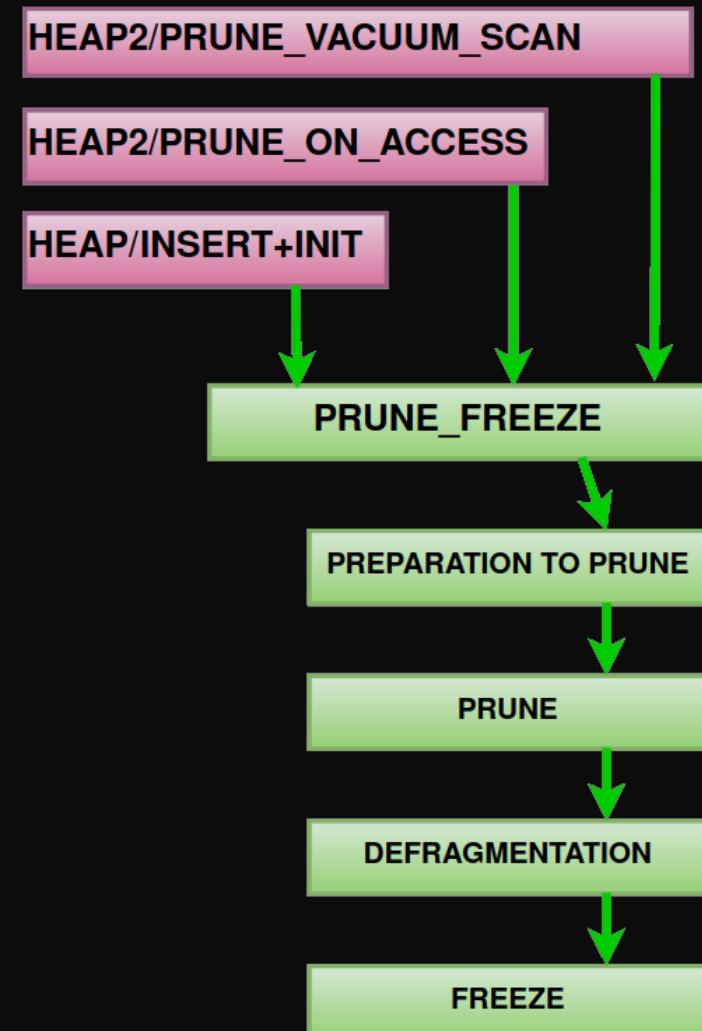
Розовым - операции уровня Resource Manager

Зелёным - низкоуровневые операции

PRUNE - умерщвление, удаление, перенаправление кортежей

DEFRAGMENTATION - восстановление фрагментации страницы

FREEZE - замораживание кортежей



## Практика. Начало

**cd pg18xid64v60\_ekat\_practice**

**git checkout ABC**

Имеем патч Xid64, доработанный для PG18dev, содержащий исправления ряда багов.

Запускаем тест 027\_stream\_regress

**./tst0.sh**

“All test successful!” 😊

```
evoro:lin-ev:~/tantor/dv/pg18xid64v60_ekat_practice(pg18xid64v60_ekat_practice)$ ./tst0.sh
=====
BALOO: STARTED TESTING. NO WAL consistency checking.
=====
++ make -C src/test/recovery/ check PROVE_TESTS=t/027_stream_regress.pl
...
# +++
# +++ tap check in src/test/recovery +++
t/027_stream_regress.pl .. ok
All tests successful.
Files=1, Tests=9, 42 wallclock secs ( 0.01 usr  0.00 sys +  2.07 cusr  1.53 csys =  3.61 CPU)
Result: PASS
...
=====
BALOO: FINISHED TESTING. NO WAL consistency checking.
=====
```

## Практика. Баг 0.

Запускаем тот же тест 027\_stream\_regress, но с проверкой  
консистентности.

**./tst1.sh**

```
make -C "src/test/recovery/" \
    check PROVE_TESTS="t/027_stream_regress.pl" \
    PG_TEST_EXTRA="wal_consistency_checking"
```

**Ctrl+C**

Имеем **неконсистентное состояние страницы на стороне реплики**.

Был бы Postgres –  
приключения  
на него найдутся!

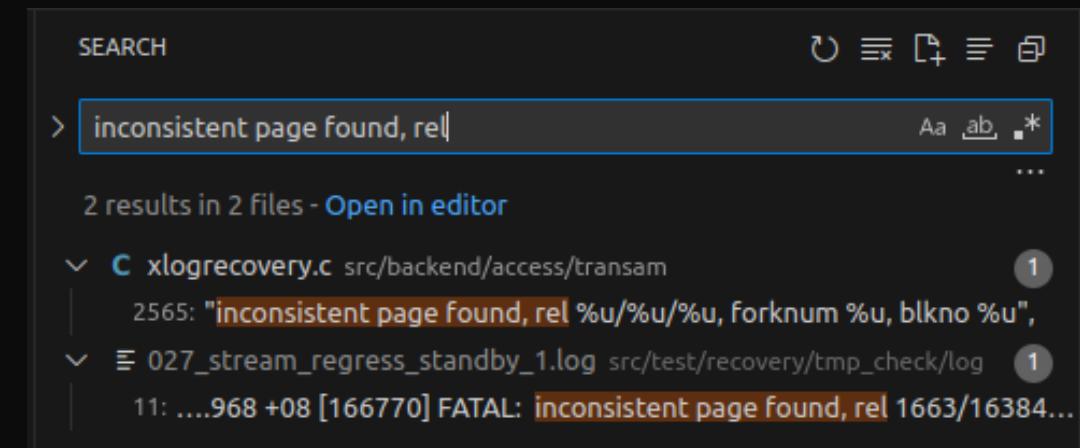
```
LOG: started streaming WAL from primary at 0/3000000 on timeline 1
FATAL: inconsistent page found, rel 1663/16384/1259, forknum 0, blkno 0
CONTEXT: WAL redo at 0/3490DF0 for Heap/INSERT+INIT: off: 0, flags: 0x00;
LOG: startup process (PID 166770) exited with exit code 1
```

## Баг 0. Поиск источника сообщения

Текстовым поиском выясняем место в коде, откуда мы получили сообщение об ошибке.

Это функция **verifyBackupConsistency**

Откуда звук?



```
src > backend > access > transam > C xlogrecovery.c > verifyBackupPageConsistency(XLogReaderState *)
2470 {
2484 {
2561     /* Time to compare the primary and replay images. */
2562     if (memcmp(replay_image_masked, primary_image_masked, BLCKSZ) != 0)
2563     {
2564         elog(FATAL,
2565             "inconsistent page found, rel %u/%u/%u, forknum %u, blkno %u",
2566             rlocator.spc0id, rlocator.db0id, rlocator.relNumber,
2567             forknum, blkno);
2568     }
2569 }
2570 }
```

## Баг 0. Логгеры

primary\_image ≠ replay\_image  
Но что именно не равно?

Добавляем логгеры.

**git checkout DEF**

Запускаем сборку и тест.

**./bld.sh**

**./tst1.sh**

Останавливаем тест.

**Ctrl+C**

Извлекаем лог страницы из логов  
standby-сервера.

**./pagelog\_standb.sh**

Вскрытие  
покажет!

```
static inline void baloo_log_page_X(RelFileLocator locator, Page page, const char* page_name)
{
#define BALOO_X_SPC 1663U
#define BALOO_X_DB 16384U
#define BALOO_X_REL 1259U

    if(locator.spcOid == BALOO_X_SPC
        && locator.dbOid == BALOO_X_DB
        && locator.relNumber == BALOO_X_REL)
    {
        StringInfo str;
        str = makeStringInfo();
        appendStringInfo(str,
            "BALOO: STARTED LOG PAGE %u/%u/%u. Page name: %s =====\n",
            BALOO_X_SPC, BALOO_X_DB, BALOO_X_REL, page_name);

        baloo_log_page(str, page);
        baloo_log_backtrace(str);

        appendStringInfo(str,
            "BALOO: FINISHED LOG PAGE %u/%u/%u. Page name: %s =====",
            BALOO_X_SPC, BALOO_X_DB, BALOO_X_REL, page_name);

        elog(WARNING, "%s", str->data);

        destroyStringInfo(str);
    }
}
```

```
evoro:lin-ev:~/tantor/dv/pg18xid64v60_ekat_practice[774d1d398a6]$ ./pagelog_standb.sh
=====
BALOO: STARTED EXTRACTING PAGES FROM PRIMARY LOGS
=====
++ ./pagelogextractor.pl src/test/recovery/tmp_check/log/027_stream_regress_standby_1.log
Extract a page N=1 with pagename=primary_image_UNMASKED. File: ./pagelog/20250405-163357138-N00001_primary_image_UNMASKED
Extract a page N=2 with pagename=replay_image_UNMASKED. File: ./pagelog/20250405-163357139-N00002_replay_image_UNMASKED
Extract a page N=3 with pagename=primary_image_MASKED. File: ./pagelog/20250405-163357141-N00003_primary_image_MASKED
Extract a page N=4 with pagename=replay_image_MASKED. File: ./pagelog/20250405-163357143-N00004_replay_image_MASKED
++ set +x
=====
BALOO: FINISHED EXTRACTING PAGES FROM PRIMARY LOGS
```

# Баг 0. Сравниваем страницы

meld pagelog/1primary\_image\_UNMASKED pagelog/2replay\_image\_UNMASKED

Вскрытие покажет!

```

1 2025-04-04 14:53:05.519 +08 [67553] WARNING: BALOO: STARTED LOG PAGE 1663/16384/1259. Page name: p →
  primary_image_UNMASKED =====
2 → 00 00 00 00 00 B0 44 31 03 69 C0 01 00 1C 00 50 00 F0 1F 05 20 00 00 00 00 50 80 70 01 00 00 00 00 00
3 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5 → 01 00 22 00 01 08 20 FF FF FF 7F 00 00 00 00 00 02 40 00 00 63 68 61 72 5F 74 62 6C 00 00 00 00
6 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8 → 0A 00 00 00 02 00 00 00 02 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 → 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
11 → FF FF 7F 00 00 00 00 00 00 07 0C 00 00 00 00 70 67 5F 65 78 74 6E 73 69 6F 6E 00 00 00 00 00 00 00
12 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
14 → 07 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
15 → 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16 → 01 00 00 00 1D 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17 → B4 0E 00 00 70 67 5F 74 73 5F 74 65 6D 70 6C 61 74 65 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```

1 2025-04-04 14:53:05.524 +08 [67553] WARNING: BALOO: STARTED LOG PAGE 1663/16384/1259. Page name: r →
  replay_image_UNMASKED =====
2 → 00 00 00 00 C8 2E 49 03 00 00 00 00 1C 00 38 1F F0 1F 05 20 00 00 00 00 38 9F 70 01 00 00 00 00
3 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
11 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
14 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
15 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18 → 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```

259 → Header of the page ===== ←
260 → .... checksum= 49257, flags= 1, lower= 28, upper= 80, ← →
261 → .... special= 8176, pagesize_version= 8197, pd_prune_xid= 0, ←
262 → .... pd_lsn= 0/0-33144B0 ← →
263 → Special of the page ===== ←
264 → .... xid_base= 0, multi_base= 0 ←
265 → Line pointers array image: ←
266 → 50 80 70 01 ← ←
267 → === Started logging items === ←
268 → .... maxoff=1 ←
269 → ItemId.info: ←
270 → .... offnum=1, IsUsed=1, IsDead=0, IsRedirected=0 ←
271 → .... len=184, offset=80 ← →
272 → TupleHeader: ←
273 → .... choice = { .xmin=753, .xmax=0, .cid=0 }, ←
274 → .... ctid = { .blkid { .bihi=0, .bilo=0 }, .posid=1 }, ←
275 → .... infomask2=0022, infomask=0801, t_hoff=32, ←
276 → ← ←

```

```

259 → Header of the page ===== ←
260 → .... checksum= 0, flags= 0, lower= 28, upper= 7992, ← →
261 → .... special= 8176, pagesize_version= 8197, pd_prune_xid= 0, ←
262 → .... pd_lsn= 0/0-3492EC8 ← →
263 → Special of the page ===== ←
264 → .... xid_base= 0, multi_base= 0 ←
265 → Line pointers array image: ←
266 → 38 9F 70 01 ← ←
267 → === Started logging items === ←
268 → .... maxoff=1 ←
269 → ItemId.info: ←
270 → .... offnum=1, IsUsed=1, IsDead=0, IsRedirected=0 ←
271 → .... len=184, offset=7992 ← →
272 → TupleHeader: ←
273 → .... choice = { .xmin=753, .xmax=0, .cid=0 }, ←
274 → .... ctid = { .blkid { .bihi=0, .bilo=0 }, .posid=1 }, ←
275 → .... infomask2=0022, infomask=0801, t_hoff=32, ←
276 → ← ←

```

# Баг 0. Сравниваем код функций, производящих страницы

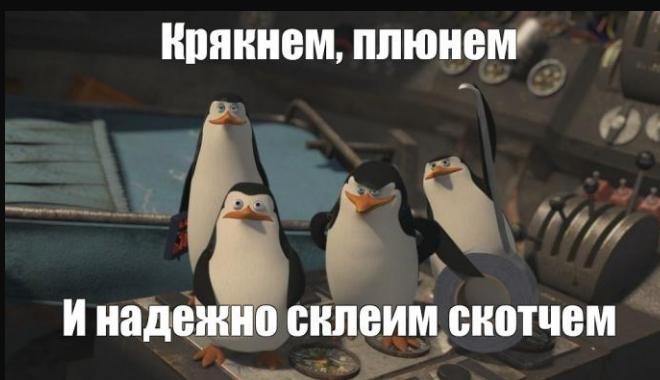
Вскрытие  
покажет!

DO (PRIMARY)		REDO (SECONDARY)	
2138   /* You, 2 days ago against locale variation...		428   /*	
2139 void heap_insert(Relation relation, HeapTuple tup, CommandId cid,		429 static void	
2140        int options, BulkInsertState bstate)		430 heap_xlog_insert(XLogReaderState *record)	
2141 {		431 {	
2142        /*		432       XLogRecPtr lsn = record->EndRecPtr;	
2158        heaptup = heap_prepare_insert(relation, tup, cid, options);		448       XLogRedoAction action;	
2159        /*		449       bool isinit = (XLogRecGetInfo(record) & XLOG_HEAP_INIT_PAGE) != 0;	
2160        *		450       Pointer rec_data = (Pointer) XLogRecGetData(record);	
2161        * Find buffer to insert this tuple into. If the page is all visible,		451       transactionId xid_base = InvalidTransactionId;	
2162        if (RelationNeedsWAL(relation))		452       TransactionId multi_base = InvalidTransactionId;	
2219            /*		483        /*	
2220            * buffer references from XLogInsert.		484            * If we inserted the first and only tuple on the page, re-initialize the	
2221            */		485            * page from scratch.	
2222            if (ItemPointerGetOffsetNumber(&(heaptup->t_self)) == FirstOffsetNumber &&		486            */	
2223              PageGetMaxOffsetNumber(page) == FirstOffsetNumber)		487        if (isinit)	
2224            {		488        {	
2225              info  = XLOG_HEAP_INIT_PAGE;		489              buffer = XLogInitBufferForRedo(record, 0);	
2226              bufflags  = REGBUF_WILL_INIT;		490              page = BufferGetPage(buffer);	
2227            }     You, 2 days ago * psql: Make test robust against locale variation...		491	
2228 }		492              if (xlrec->flags & XLH_INSERT_ON_TOAST_RELATION)	
2191 START_CRIT_SECTION();	<b>Запись тапла на существующую страницу с мусором</b>	544       HeapTupleHeaderSetCmin(htup, FirstCommandId);	
2192		545       htup->t_ctid = target_tid;	
2193 RelationPutHeapTuple(relation, buffer, heaptup,		546	
2194            (options & HEAP_INSERT_SPECULATIVE) != 0);		547       if (PageAddItem(page, (Item) htup, newlen, xlrec->offnum,	
2195		548              true, true) == InvalidOffsetNumber)	
2196 if (PageIsAllVisible(BufferGetPage(buffer)))		549       elog(PANIC, "failed to add tuple");	

## Баг 0. Заплатка

Добавляем логику,  
определяющую, была ли  
страница чистой до  
вставки тапла.

**git checkout GHI**



### AFTER THE FIX

```
2139 void
2140 heap_insert(Relation relation, HeapTuple tup, CommandId cid,
2141               int options, BulkInsertState bistate)
2142 {
2143     TransactionId xid = GetCurrentTransactionId();
2144     HeapTuple heaptup;
2145     Buffer buffer;
2146     Buffer vmbuffer = InvalidBuffer;
2147     bool all_visible_cleared = false;
2148     PageHeader pageheader;
2149     bool were_not_tuples; You, 2 hours ago • GHI. Bug0 is fixed. A cond
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196     were_not_tuples = pageheader->pd_special == (uint32) pageheader->pd_upper;
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4069
4070
4071
4072
407
```

## Баг 1. Это prune

**git checkout GHI**  
**./bld.sh**

Запускаем тест.

**./tst1.sh**

Останавливаем тест.

**Ctrl+C**

Извлекаем лог

страницы из логов  
standby-сервера.

**./pagelog\_standb.sh**

```
2025-04-05 16:48:15.225 +08 [276638] FATAL: inconsistent page found, rel  
1663/16384/1259, forknum 0, blkno 19  
2025-04-05 16:48:15.225 +08 [276638] CONTEXT: WAL redo at 0/7E2A378 for  
Heap2/PRUNE_ON_ACCESS: snapshotConflictHorizon: 782, isCatalogRel: F,  
nplans: 0, nredirected: 4, ndead: 0, nunused: 1, redirected: [16->19,  
25->30, 29->33, 34->37], unused: [28]; blkref #0: rel 1663/16384/1259, blk  
19 FPW
```

```
evoro:lin-ev:~/tantor/dv/pg18xid64PS(ps/pg18xid64)$ git log  
commit f83d709760d84253b539ee454ce9celceb9ad9ac  
Author: Heikki Linnakangas <heikki.linnakangas@iki.fi>  
Date: Mon Mar 25 14:59:58 2024 +0200  
  
Merge [prune, freeze and vacuum WAL record formats]
```

**Это prune!**  
**Ну, всё понятно!**  
**Но что конкретно?**

## Баг 1. «Вскрытие покажет». Анализ

2025-04-05 16:48:15.223 +08 [276638] WARNING: BALOO: STARTED LOG PAGE 1663/16384/1259. Page name: primary\_image MASKED =====

2 00 00 00 00 00 00 00 00 00 00 00 C4 00 C0 03 F0 1F 05 20 00 00 00 00 38 9F 70 01 04 00 01 00  
3 08 9A 70 01 48 9E DA 01 07 00 01 00 50 99 70 01 58 9D DA 01 09 00 01 00 A0 9C 70 01 0C 00 01 00  
4 98 98 70 01 B0 9B DA 01 0F 00 01 00 E0 97 70 01 C0 9A DA 01 13 00 01 00 70 96 70 01 B8 95 70 01  
83 00 01 6E 00 00 00 00 00 13 03 00 00 00 00 00 00 00 62 00 00 00 62 00 00 00  
84 00 00 00 00 00 00 13 00 25 00 22 40 21 05 20 FF FF FF 7F 00 00 00 00 32 40 00 00 73 74 75 64 → ← 84  
85 65 6E 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
112 0C 03 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
113 21 00 22 40 21 05 20 FF FF FF 7F 00 00 00 00 00 2D 40 00 00 65 6D 70 00 00 00 00 00 00 00 00 00 → ← 113  
114 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
117 00 00 00 00 00 00 00 00 70 72 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
118 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
119 FF FF 7F 00 00 00 00 00 00 28 40 00 00 70 65 72 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
135 0D 03 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
136 1C 00 22 40 21 05 20 FF FF FF 7F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
137 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
274 === Started logging items====  
275 maxoff=43  
276 ItemId info:  
277 offnum=1, IsUsed=1, IsDead=0, IsRedirected=0  
278 len=184, offset=7992  
279 TupleHeader:  
280 choice = { xmin=3, xmax=0, cid=0 },  
281 ctid = { blkid { bihi=0, bilo=19 }, posid=1 },  
282 infomask2=0022, infomask=2301, t\_hoff=32,  
  
evoro:lin-ev:~/tantor/dv/pg18xid64v60\_ekat\_practice[4a1f305687d]\$ grep -c "20 FF" ./pagelog/20250405-164815223-N00003\_primary\_image\_MASKED  
38  
evoro:lin-ev:~/tantor/dv/pg18xid64v60\_ekat\_practice[4a1f305687d]\$ grep -c "20 FF" ./pagelog/20250405-164815225-N00004\_replay\_image\_MASKED  
38  
  
evoro:lin-ev:~/tantor/dv/pg18xid64v60\_ekat\_practice[4a1f305687d]\$ grep -c "t\_hoff=" ./pagelog/20250405-164815225-N00004\_replay\_image\_MASKED  
33

# Баг 1. Prune на стороне DO

## **DO (PRIMARY)**

```
361 void heap_page_prune_and_freeze(Relation relation, Buffer buffer,
362                                     GlobalState *globalstate)
363 {
364     C pruneheap.c X ...
365
366     src > backend > access > heap > C pruneheap.c > heap_page_prune_opt(Relation, Buffer)
367
368     371         offnum = OffsetNumberPrev(offnum))
369
370     372         prstate.htsv[offnum] = heap_prune_satisfies_vacuum(prstate,
371
371             373                 buffer);
372
373             374
374             375                 if (!HeapTupleHeaderIsHeapOnly(htup))
375                 prstate.root_items[prstate.nroot_items++] = offnum;
376             else
377                 prstate.heaponly_items[prstate.nheaponly_items++] = offnum;
378             }
379
380             381
381             382             /*
382                 * If checksums are enabled, heap_prune_satisfies_vacuum() may !
383                 * emit an FPI to be emitted.
384                 */
385
386             387             hint_bit_fpi = fpi_before != pgWalUsage.wal_fpi;
387
388             389
389             390             /*
390                 * Process HOT chains.
391                 */
392
393             394             /*
394                 * We added the code to handle changes from toast items.
395                 */
396
397             398             if (do_prune || do_freeze)
398             {
399                 400                     /* Apply the planned item changes and repair page fragmentation.
400                     if (do_prune)
401                     {
402                         403                             heap_page_prune_execute(buffer, false,
403
404                                         prstate.redirected, prstate.nred,
405                                         prstate.nowdead, prstate.ndead,
406                                         prstate.nowunused, prstate.nunus,
407                                         repairFragmentation,
408                                         IsToastRelation(relation));
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
667
668
669
669
670
671
672
673
674
675
675
676
677
677
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1650
1651
1651
1652
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1660
1661
1661
1662
1662
1663
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1670
1671
1671
1672
1672
1673
1673
1674
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1680
1681
1681
1682
1682
1683
1683
1684
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1690
1691
1691
1692
1692
1693
1693
1694
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1700
1701
1701
1702
1702
1703
1703
1704
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1710
1711
1711
1712
1712
1713
1713
1714
1714
1715
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1720
1721
1721
1722
1722
1723
1723
1724
1724
1725
1725
1726
1726
1727
1727
1728
1728
1729
1729
1730
1730
1731
1731
1732
1732
1733
1733
1734
1734
1735
1735
1736
1736
1737
1737
1738
1738
1739
1739
1740
1740
1741
1741
1742
1742
1743
1743
1744
1744
1745
1745
1746
1746
1747
1747
1748
1748
1749
1749
1750
1750
1751
1751
1752
1752
1753
1753
1754
1754
1755
1755
1756
1756
1757
1757
1758
1758
1759
1759
1760
1760
1761
1761
1762
1762
1763
1763
1764
1764
1765
1765
1766
1766
1767
1767
1768
1768
1769
1769
1770
1770
1771
1771
1772
1772
1773
1773
1774
1774
1775
1775
1776
1776
1777
1777
1778
1778
1779
1779
1780
1780
1781
1781
1782
1782
1783
1783
1784
1784
1785
1785
1786
1786
1787
1787
1788
1788
1789
1789
1790
1790
1791
1791
1792
1792
1793
1793
1794
1794
1795
1795
1796
1796
1797
1797
1798
1798
1799
1799
1800
1800
1801
1801
1802
1802
1803
1803
1804
1804
1805
1805
1806
1806
1807
1807
1808
1808
1809
1809
1810
1
```

# Баг 1. Prune на стороне REDO

## REDO (SECONDARY)

```
ckend > access > heap > C heapam_xlog.c > heap_xlog_prune_freeze(XLogReaderState *)
    heap_xlog_prune_freeze(XLogReaderState *record)
        if (action == BLK_NEEDS_REDO)

            heap_xlog_deserialize_prune_and_freeze(dataptr, xlrec.flags,
                &nplans, &plans, &frz_offsets,
                &nredirected, &redirected,
                &ndead, &nowdead,
                &nunused, &nowunused);

            /*
             * Update all line pointers per the record, and repair fragmentation
             * if needed.
             */
            if (nredirected > 0 || ndead > 0 || nunused > 0)
                heap_page_prune_execute_buffer(
                    (xlrec.flags & XLHP_CLEANUP_LOCK) == 0,
                    redirected, nredirected,
                    nowdead, ndead,
                    nowunused, nunused,
                    (xlrec.flags & XLHP_REPAIR_FRAGMENTATION) != 0,
                    (xlrec.flags & XLHP_ON_TOAST_RELATION) != 0);
```

2

## Баг 1. На что повлиял патч?

**Xid64 - патч**

```
C pruneheap.c X
> C pruneheap.c > heap_page_prune_execute(Buffer, bool, OffsetNumber *, int, OffsetNumber *, int
1591     void
1592     heap_page_prune_execute(Buffer buffer, bool lp_truncate_only,
1593                             OffsetNumber *redirected, int nredirected,
1594                             OffsetNumber *nowdead, int ndead,
1595                             OffsetNumber *nowunused, int nunused,
1596                             bool repairFragmentation,
1597                             bool is_toast)
1598 }

C pruneheap.c X
src > backend > access > heap > C pruneheap.c > heap_page_prune_execute(Buffer, bool, OffsetNumber *, int, OffsetNumber *, int, bool is_toast)
1597     else
1598         /*
1599 */
1745     if (repairFragmentation)
1746         PageRepairFragmentation(page, is_toast);
1747     /*
1748     * Now that the page has been modified, assert that redirected
1749     * still point to valid targets.
1750     */
1751     page_verify_redirects(page);
1752 }
1753 You, 3 days ago • psql: Make test robust against locale
1754 }
```

**PG18dev**

```
C pruneheap.c 5 X
src > backend > access > heap > C pruneheap.c > heap_page_prune_execute(Buffer, bool, OffsetNumber *, int, OffsetNumber *, int, bool is_toast)
1550     /*
1551     * The buffer... If it is SEC, all ordinary EXCLUSIVE LOCK SURFACES.
1552     */
1559 */
1560 void
1561 heap_page_prune_execute(Buffer buffer, bool lp_truncate_only,
1562                         OffsetNumber *redirected, int nredirected,
1563                         OffsetNumber *nowdead, int ndead,
1564                         OffsetNumber *nowunused, int nunused)
1565 {

C pruneheap.c 5 X
src > backend > access > heap > C pruneheap.c > ...
1564
1565     else
1566     {
1567         /*
1568         * Finally, repair any fragmentation, and update the page's
1569         * about whether it has free pointers.
1570         */
1571         PageRepairFragmentation(page);
1572     }
1573
1574     /*
1575     * Now that the page has been modified, assert that redirected
1576     * still point to valid targets.
1577     */
1578     page_verify_redirects(page);
1579 }
```

## Баг 2.

**git checkout JKL**  
**./bld.sh**

Запускаем тест.

**./tst1.sh**

```
2025-04-05 19:26:02.102 +08 [297359] FATAL: inconsistent page found, rel 1663/16384/2611,  
forknum 0, blkno 0  
2025-04-05 19:26:02.102 +08 [297359] CONTEXT: WAL redo at 0/7D0E3758 for Heap2/  
PRUNE_VACUUM_SCAN: snapshotConflictHorizon: 6905, isCatalogRel: F, nplans: 1, nredirected: 0,  
ndead: 6, runused: 0, plans: [{ xmax: 0, infomask: 2816, infomask2: 4, ntuples: 33, offsets:  
[1, 2, 3, 4, 5, 6, 66, 67, 69, 70, 71, 72, 75, 76, 77, 78, 104, 105, 106, 107, 108, 109, 194,  
195, 196, 197, 198, 199, 200, 202, 203, 260, 261] }, dead: [204, 205, 211, 212, 240, 241];  
blkref #0: rel 1663/16384/2611 blk 0 FPW
```

```
evoro:lin-ev:~/tantor/dv/pg18xid64PS(ps/pg18xid64)$ git log  
commit f83d709760d84253b539ee454ce9celceb9ad9ac  
Author: Heikki Linnakangas <heikki.linnakangas@iki.fi>  
Date: Mon Mar 25 14:59:58 2024 +0200
```

Merge **prune**, **freeze** and **vacuum** **WAL record formats**

## Баг 2.

### Адаптируем логи

**git checkout PQR**

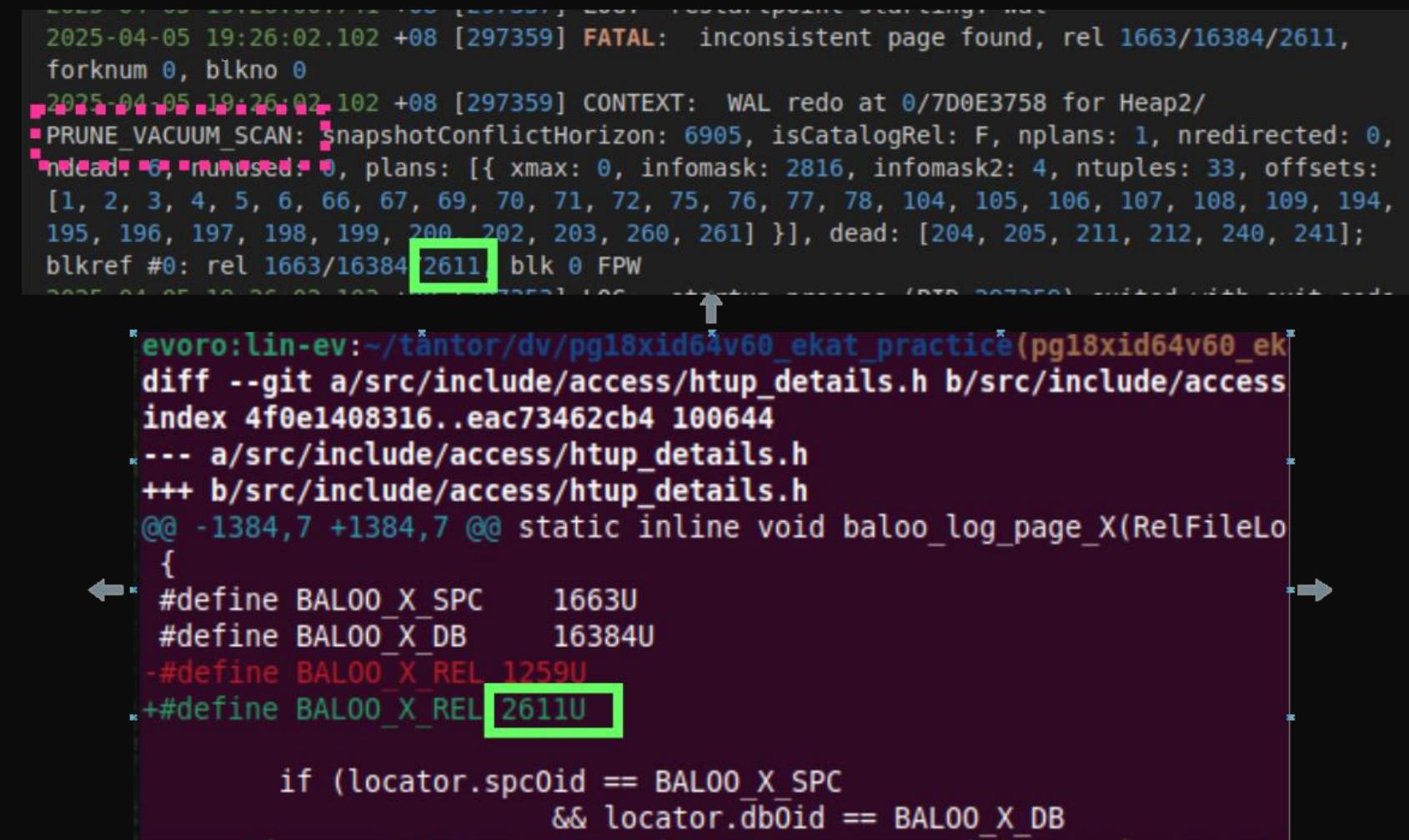
**./bld.sh**

Запускаем тест.

**./tst1.sh**

Извлекаем лог  
страницы из логов  
standby-сервера.

**./pagelog\_standb.sh**



The terminal window displays two main sections of text:

**PostgreSQL Log Output:**

```
2025-04-05 19:26:02.102 +08 [297359] FATAL: inconsistent page found, rel 1663/16384/2611,  
forknum 0, blkno 0  
2025-04-05 19:26:02.102 +08 [297359] CONTEXT: WAL redo at 0/7D0E3758 for Heap2/  
PRUNE_VACUUM_SCAN: snapshotConflictHorizon: 6905, isCatalogRel: F, nplans: 1, nredirected: 0,  
ndead: 6, nunused: 0, plans: [{ xmax: 0, infomask: 2816, infomask2: 4, ntuples: 33, offsets:  
[1, 2, 3, 4, 5, 6, 66, 67, 69, 70, 71, 72, 75, 76, 77, 78, 104, 105, 106, 107, 108, 109, 194,  
195, 196, 197, 198, 199, 200, 202, 203, 260, 261] }, dead: [204, 205, 211, 212, 240, 241];  
blkref #0: rel 1663/16384/2611 blk 0 FPW
```

**Source Code Diff:**

```
* evoro:lin-ev:~/tantor/dv/pg18xid64v60_ekat_practice(pg18xid64v60_ek  
diff --git a/src/include/access/htup_details.h b/src/include/access  
index 4f0e1408316..eac73462cb4 100644  
--- a/src/include/access/htup_details.h  
+++ b/src/include/access/htup_details.h  
@@ -1384,7 +1384,7 @@ static inline void baloo_log_page_X(RelFileLo  
{  
← * #define BALOO_X_SPC 1663U  
← * #define BALOO_X_DB 16384U  
-#define BALOO_X_REL 1259U  
+#define BALOO_X_REL 2611U  
  
if (locator.spcOid == BALOO_X_SPC  
&& locator.dbOid == BALOO_X_DB
```

The code diff highlights changes made to the `BALOO_X_REL` macro definition from 1259U to 2611U. The PostgreSQL log entry shows a page inconsistency found at offset 2611.

## Баг 2. Поиск образа страницы.

The diagram illustrates the process of identifying a page image in a PostgreSQL page log. It starts with a screenshot of a terminal window showing a page header dump. A specific value, `pd_lsn = 0/0-7C77E1D8`, is highlighted in green and linked via a red arrow to a command-line search for the same value in a page log file. This leads to another terminal window where the command `meld /pagelog/20250405-224843053-N00016_D0_after_prune_before_freeze ./pagelog/20250405-224843057-N00017_D0_after_freeze` is run. The resulting meld window shows two side-by-side binary files. A green box highlights the difference at line 1194, specifically the `xmax` field. Two green boxes at the bottom identify the left side as "Страна DO. Перед freeze" and the right side as "Страна DO. После freeze".

```
Header of the page =====
checksum= 39937, flags= 0, lower= 1188, upper= 6856,
special= 8176, pagesize_version= 8197, pd_prune_xid= 0, →
pd_lsn= 0/0-7C77E1D8
Special of the page =====
xid_base= 779, multi_base= 0

evoro:lin-ev:~/tantor/dv/pg18xid64v60_ekat_practice(pg18xid64v60_ekat_practice)$ grep '7C77E1D8' ./pagelog/*
./pagelog/20250405-224843053-N00016_D0_after_prune_before_freeze: pd_lsn= 0/0-7C77E1D8
./pagelog/20250405-224843057-N00017_D0_after_freeze: pd_lsn= 0/0-7C77E1D8
./pagelog/20250405-224843100-N00001_primary_image_UNMASKED: pd_lsn= 0/0-7C77E1D8

(practice)$ meld /pagelog/20250405-224843053-N00016_D0_after_prune_before_freeze ./pagelog/20250405-224843057-N00017_D0_after_freeze

20250405-224843053-N00016_D0_after_prune_before_freeze
1187     ItemId info:
1188         offnum=253, IsUsed=1, IsDead=1, IsRedirected=0
1189         len=0, offset=0
1190     ItemId info:
1191         offnum=254, IsUsed=1, IsDead=0, IsRedirected=0
1192         len=37, offset=6896
1193     TupleHeader:
1194         choice = { xmin=4177, xmax=4187, cid=0 }, →
1195         ctid = { blkid { bihi=0, bilo=0 }, posid=254 },
1196         infomask2=2004, infomask=0900, t_hoff=24, →
1197         ...
1198     ...
1199     Сторона DO. Перед freeze     IsRedirected=0
1200     TupleHeader:
1201         choice = { xmin=4182, xmax=4187, cid=1 }, →
1202         ctid = { blkid { bihi=0, bilo=0 }, posid=255 },
1203         infomask2=2004, infomask=0900, t_hoff=24, →
1204         ...

20250405-224843057-N00017_D0_after_freeze
1187     ItemId info:
1188         offnum=253, IsUsed=1, IsDead=1, IsRedirected=0
1189         len=0, offset=0
1190     ItemId info:
1191         offnum=254, IsUsed=1, IsDead=0, IsRedirected=0
1192         len=37, offset=6896
1193     TupleHeader:
1194         choice = { xmin=4177, xmax=0, cid=0 }, ←
1195         ctid = { blkid { bihi=0, bilo=0 }, posid=254 },
1196         infomask2=0004, infomask=0B00, t_hoff=24, ←
1197         ...
1198     ...
1199     Сторона DO. После freeze     IsRedirected=0
1200     TupleHeader:
1201         choice = { xmin=4182, xmax=0, cid=1 }, ←
1202         ctid = { blkid { bihi=0, bilo=0 }, posid=255 },
1203         infomask2=0004, infomask=0B00, t_hoff=24, ←
```

## Баг 2. Анализ.

20250405-224843053-N00016\_DO\_after\_prune\_before\_freeze

```
1185     len=0, offset=0
1187     ItemId info:
1188         offnum=253, IsUsed=1, IsDead=1, IsRedirected=0
1189         len=0, offset=0
1190     ItemId info:
1191         offnum=254, IsUsed=1, IsDead=0, IsRedirected=0
1192         len=37, offset=6896
1193     TupleHeader:
1194         choice = { xmin=4177, xmax=4187, cid=0 }, →
1195         ctid = { blkid { bihi=0, bilo=0 }, posid=254 }, →
1196         infomask2=2004, infomask=0900, t_hoff=24, →
1197     TupleHeader:
1198         choice = { xmin=4182, xmax=4187, cid=1 }, →
1199         ctid = { blkid { bihi=0, bilo=0 }, posid=255 }, →
1200         infomask2=2004, infomask=0900, t_hoff=24, →
1201     TupleHeader:
1202         choice = { xmin=4182, xmax=4187, cid=1 }, →
1203         ctid = { blkid { bihi=0, bilo=0 }, posid=255 }, →
```

Страна DO. Перед freeze

1185 len=0, offset=0
1187 ItemId info:
1188 offnum=253, IsUsed=1, IsDead=0, IsRedirected=0
1189 len=0, offset=0
1190 ItemId info:
1191 offnum=254, IsUsed=1, IsDead=0, IsRedirected=0
1192 len=37, offset=6896
1193 TupleHeader:
1194 choice = { xmin=4177, xmax=0, cid=0 }, ←
1195 ctid = { blkid { bihi=0, bilo=0 }, posid=254 }, ←
1196 infomask2=0004, infomask=0B00, t\_hoff=24, ←
1197 TupleHeader:
1198 offnum=255, IsUsed=1, IsDead=0, IsRedirected=0
1199 len=37, offset=6896
1200 TupleHeader:
1201 choice = { xmin=4182, xmax=0, cid=1 }, ←
1202 ctid = { blkid { bihi=0, bilo=0 }, posid=255 }, ←
1203 infomask2=0004, infomask=0B00, t\_hoff=24,

Страна DO. После freeze

20250405-224843106-N00003\_primary\_image\_MASKED

```
7     ItemId info:
8         offnum=253, IsUsed=1, IsDead=1, IsRedirected=0
9         len=0, offset=0
10    ItemId info:
11        offnum=254, IsUsed=1, IsDead=0, IsRedirected=0
12        len=37, offset=6896
13    TupleHeader:
14        choice = { xmin=4177, xmax=0, cid=0 }, →
15        ctid = { blkid { bihi=0, bilo=0 }, posid=254 }, →
16        infomask2=0004, infomask=0300, t_hoff=24,
17    ItemId info:
18        choice = { xmin=4182, xmax=0, cid=0 }, →
19        ctid = { blkid { bihi=0, bilo=0 }, posid=255 }, →
20        infomask2=0004, infomask=0300, t_hoff=24,
```

Страна DO. После freeze

20250405-224843108-N00004\_replay\_image\_MASKED

```
1187     ItemId info:
1188         offnum=253, IsUsed=1, IsDead=1, IsRedirected=0
1189         len=0, offset=0
1190     ItemId info:
1191         offnum=254, IsUsed=1, IsDead=0, IsRedirected=0
1192         len=37, offset=6896
1193     TupleHeader:
1194         choice = { xmin=4177, xmax=4187, cid=0 }, →
1195         ctid = { blkid { bihi=0, bilo=0 }, posid=254 }, →
1196         infomask2=0004, infomask=0300, t_hoff=24,
1197     ItemId info:
1198         offnum=255, IsUsed=1, IsDead=0, IsRedirected=0
1199         len=37, offset=6896
1200     TupleHeader:
1201         choice = { xmin=4182, xmax=4187, cid=0 }, →
1202         ctid = { blkid { bihi=0, bilo=0 }, posid=255 }, →
1203         infomask2=0004, infomask=0300, t_hoff=24,
```

Страна REDO. После freeze

## Баг 2. Анализ.

**Страна DO**

```

C pruneheap.c x
src > backend > access > heap > C pruneheap.c > heap_page_prune_and_freeze(Relation, Buffer, GlobalVisState *, int, VacuumCutoffs *, Prun...
301 void
362 heap_page_prune_and_freeze(Relation relation, Buffer buffer,
363                             GlobalVisState *vistest,
371                                     bool repairFragmentation)
805     if (do_prune || do_freeze)
817         memcpy(page_aft_prune_bef_freeze, BufferGetPage(buffer), BLCKSZ); DO. Перед freeze
818     if (do_freeze)
820         heap_freeze_prepared_tuples(relation, buffer, prstate.frozen, prstate.nfrozen);
821         memcpy(page_aft_freeze, BufferGetPage(buffer), BLCKSZ); Страна DO. После freeze
823
824
C heapam.c x
src > backend > access > heap > C heapam.c > heap_freeze_prepared_tuples(Relation, Buffer, HeapTupleFreeze *, int)
7589 heap_freeze_prepared_tuples(Relation rel, Buffer buffer, HeapTupleFreeze *tuples, int ntuples)
7590
7599     htup = (HeapTupleHeader) PageGetItemId(page, itemid); You, 3 days ago + psql: Make
7600     heap_execute_freeze_tuple_page(page, htup, frz,
7601                                     IsToastRelation(rel));
7602
C heapam.h x
tmp_install > home > evoro > tantor > dv > pg18xid64v60_ekat_practice > inst > include > postgresql > server > access > C heapam.h > heap...
14 #ifdef HEAPAM_H
478 static inline void
479 heap_execute_freeze_tuple_page(Page page, HeapTupleHeader tuple,
480                                 HeapTupleFreeze *frz, bool is_toast)
481 {
482     HeapTupleData htup;
483
484     htup.t_data = tuple;
485     heap_execute_freeze_tuple(&htup, frz);
486
487     HeapTupleHeaderStoreXmax(page, &htup, is_toast);
488 }

```

**Страна REDO**

```

C heapam_xlog.c x
src > backend > access > heap > C heapam_xlog.c > heap_xlog_prune_freeze(XLogReaderState *)
30 heap_xlog_prune_freeze(XLogReaderState *record)
79 if (action == BLK_NEEDS_REDO)
111
112
113
114
115
116
117
118
119
114 /* Freeze tuples */
115 for (int p = 0; p < nplans; p++)
116 {
117     HeapTupleFreeze frz;
118
119
120     /*
121     * Convert freeze plan representation from WAL record into
122     * per-tuple format used by heap_execute_freeze_tuple
123     */
124     frz.xmax = plans[p].xmax;
125     frz.t_infomask2 = plans[p].t_infomask2;
126     frz.t_infomask = plans[p].t_infomask;
127     frz.frzflags = plans[p].frzflags;
128     frz.offset = InvalidOffsetNumber; /* unused, but be tidy */
129
130     for (int i = 0; i < plans[p].ntuples; i++)
131     {
132         OffsetNumber offset = *(frz_offsets++);
133         ItemId lp;
134         HeapTupleData tp;
135
136         lp = PageGetItemId(page, offset);
137         tp.t_data = (HeapTupleHeader) PageGetItem(page, lp);
138         tp.t_len = ItemIdGetLength(lp);
139         HeapTupleCopyXidsFromPage(tp, page);
140         heap_execute_freeze_tuple(lp, tp);
141     }
142
143
144     /* There should be no more data */
145     Assert((char *) frz_offsets == dataptr + datalen);
146
147

```

## Баг 2. Ремонт. “Staying alive!”

**git checkout VWX**  
**./bld.sh**

Запускаем тест.  
**./tst1.sh**

```
# +++ tap check in src/test/recovery +++
t/027_stream_regress.pl ... ok
All tests successful.
Files=1, Tests=9, 215 wallclock secs ( 0.02 usr  0.00 sys +
Result: PASS
make: Leaving directory '/home/evoro/tantor/dv/pg18xid64v60'
++ set +x
=====
BALOO: FINISHED TESTING. WITH WAL consistency checking.
=====
```



# Партнёры и соучастники

(в доработке патча xid64 для  
PG17, PG18dev)

**Евгений Воропаев (Тантор)**  
Адаптация кода. Доработки.  
Ремонт основной массы багов. Доклад.

**Иван Кушнаренко (Тантор)**  
Fixed the “stale xid64” problem at the  
heapam\_tuple\_satisfies\_snapshot function.

**Эдуард Степанов (Тантор)**  
Fixed the error that occurred during parsing an XLog  
record in the DecodeUpdate function when the  
XLOG\_HEAPINITPAGE flag is set.

**Сергей Соловьёв (Тантор)**  
Participation in fixing the bug of the  
XMAX\_COMMITTED hint bit.

**Олег Гуров (Тантор)**  
Руководство, техническое наставничество,  
волшебный пендель.

**Андрей Бородин (Yandex)**  
Техническое наставничество.



PG BootCamp Russia 2025 Ekaterinburg  
[PGBootCamp.ru](http://PGBootCamp.ru)

# Спасибо за внимание!



[www.tantorlabs.ru](http://www.tantorlabs.ru)