

Очистка



Проблема мертвых кортежей

- По мере закрытия транзакций необходимость в использованных ими снимках отпадает
 - Все версии строк, относящиеся к этим снимкам – «**мертвые версии**», необходимо удалить и освободить место
 - Мертвые кортежи могут снизить производительность БД, поскольку они продолжают занимать место и приводят к раздуванию файлов
- Для решения данной проблемы используется:
 - **VACUUM** - для удаления мертвых кортежей
 - **ANALYZE** - для обновления статистики по таблице, чтобы оптимизатор мог выбрать оптимальный план выполнения для оператора SQL
- В PostgreSQL за автоматическое выполнение как очистки, так и анализа таблиц отвечает Autovacuum
 - работает параллельно с другими процессами и ничего не блокирует
 - в файлах данных появляются «дыры», которые могут быть использованы для новых строчек
 - **сами файлы не уменьшаются!**

<https://habr.com/ru/company/tensor/blog/481866/>



Задачи Автоочистки

- Удаление мертвых кортежей (dead tuples)
 - очищает ненужные версии строк на табличных страницах (пропуская страницы, уже отмеченные в карте видимости)
 - очищает индексные записи, ссылающиеся на очищенные версии строк
 - освобождает указатели
 - обновляет статистику по таблице
- Замораживание старых txid
 - При необходимости замораживаются старые txid кортежей
 - Обновление информации, связанной с txid, в системном каталоге (pg_database и pg_class)
 - Удаляются старые файлы статусов транзакций (**xact**), поскольку все транзакции, хранящиеся в этих файлах, могут рассматриваться как замороженные txids во всем кластере базы данных
- Другие
 - Обновление **FSM** и **VM** обработанных таблиц
 - Обновление статистических данных (**pg_stat_all_tables** и т. д.)



Stats Collector

- Для корректной работы Автоочистки необходима статистическая информация
- Фоновый процесс **Stats Collector** - отслеживает информацию об изменении табличных данных и активности

```
track_activities = on  
track_counts = on
```

- **track_activities** – включает stats collector
- **track_counts** - сбор статистики по текущему состоянию всех таблиц и индексов (какие активности по таблицам и индексам в последнее время осуществлялись)



Автоочистка

- Выполняется периодически
 - для таблиц с определенным количеством изменений
 - в том числе для toast-таблиц
- Процесс **autovacuum launcher**
 - постоянно запущен
 - использует информацию, собранную **Stats Collector** для определения списка таблиц-кандидатов для автоочистки
 - планирует запуск рабочих процессов
- Процессы **autovacuum worker**
 - запускаются процессом **postmaster** по просьбе **autovacuum launcher**
 - подключаются к заданной БД, перебирают и очищают таблицы

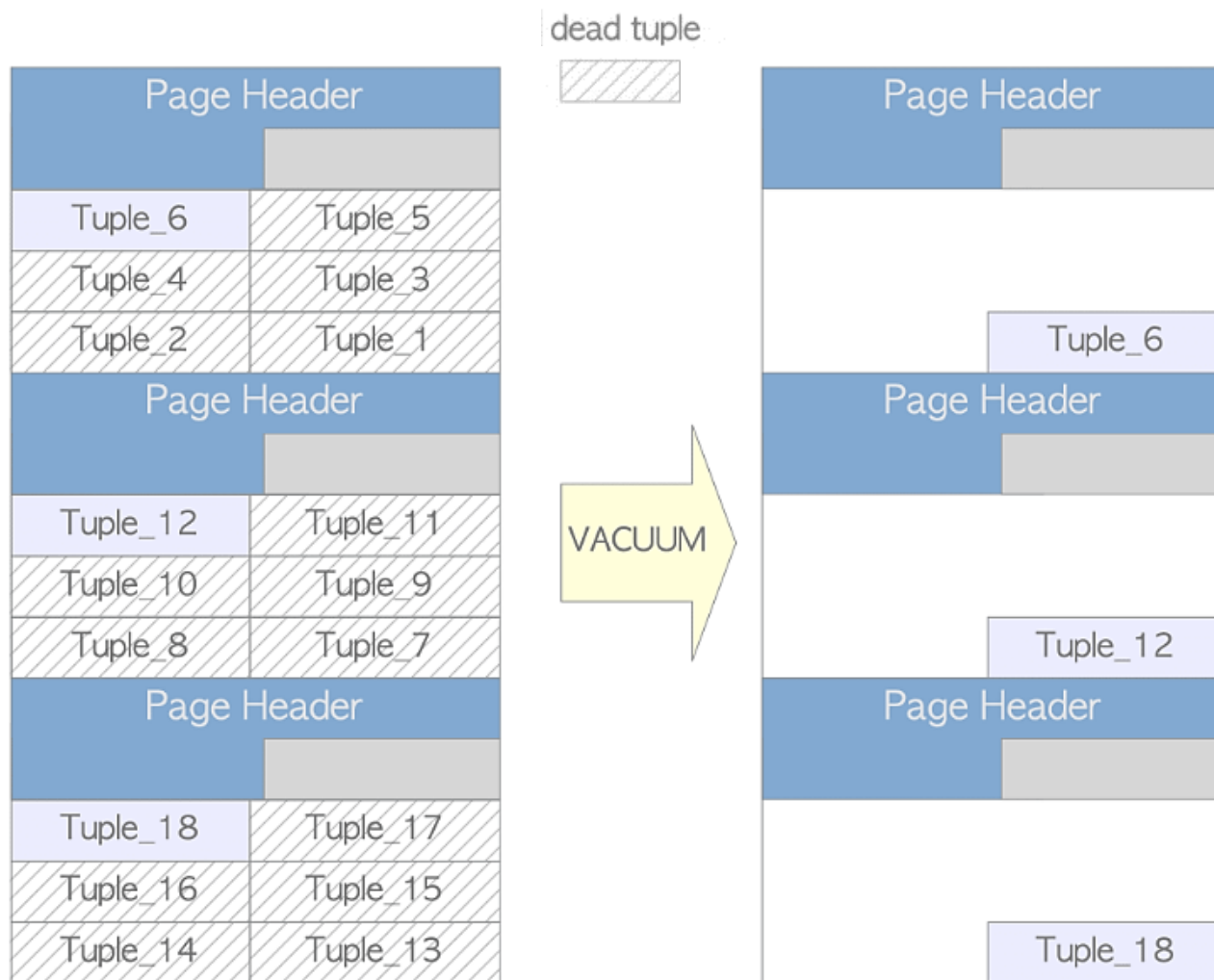


Мониторинг Autovacuum worker

- Параметр **log_autovacuum_min_duration** - задаёт время (в миллисекундах) выполнения действия автоочистки, при превышении которого информация об этом действии записывается в журнал сообщений сервера
- Представление **pg_stat_progress_vacuum** – содержит по одной строке для каждого обслуживающего процесса (включая рабочие процессы автоочистки)
- Представление **pg_stat_all_tables** – содержит по одной строке на каждую таблицу в текущей базе данных со статистикой по обращениям к этой таблице (в т.ч. процессов автоочистки)



Процесс очистки (autovacuum, vacuum)



- **VACUUM** не может уменьшить размер таблицы, даже если удалить много мертвых кортежей
- Только в случае полного освобождения страниц в **конце файла** – они возвращаются файловой системе
- Это приводит к пустой трате места на диске и отрицательно влияет на производительность базы данных
- **Например**, в приведенном примере при чтении трех кортежей в таблице необходимо загрузить с диска три страницы



dead tuple



- **VACUUM FULL** перестраивает всю таблицу и освобождает место на диске
- При использовании команды **VACUUM FULL** следует учитывать два момента:
 - **Никто не может получить доступ** (чтение/запись) к таблице во время обработки **VACUUM FULL**
 - Временно используется не более чем вдвое больше дискового пространства таблицы; поэтому необходимо проверять наличие свободного пространства на диске



Когда необходимо выполнить VACUUM FULL?

- К сожалению, не существует наилучшей практики, когда необходимо выполнять **VACUUM FULL**
- Для принятия решения следует руководствоваться информацией о свободном пространстве для конкретной таблицы
 - Средний коэффициент свободного пространства для таблицы:

```
CREATE EXTENSION pg_freespacemap;  
SELECT count(*) as "количество страниц",  
       pg_size_pretty(cast(avg(avail) as bigint)) as "Ср.размер св.пространства",  
       round(100 * avg(avail)/8192 ,2) as "Ср.коэф. св.пространства"  
FROM pg_freespace('Sales"."Orders");
```

- Коэффициент свободного пространства на каждой странице указанной таблицы:

```
SELECT *, round(100 * avail/8192 ,2) as "коэф.свободного пространства"  
FROM pg_freespace('Sales"."Orders");
```



Карта заморозки

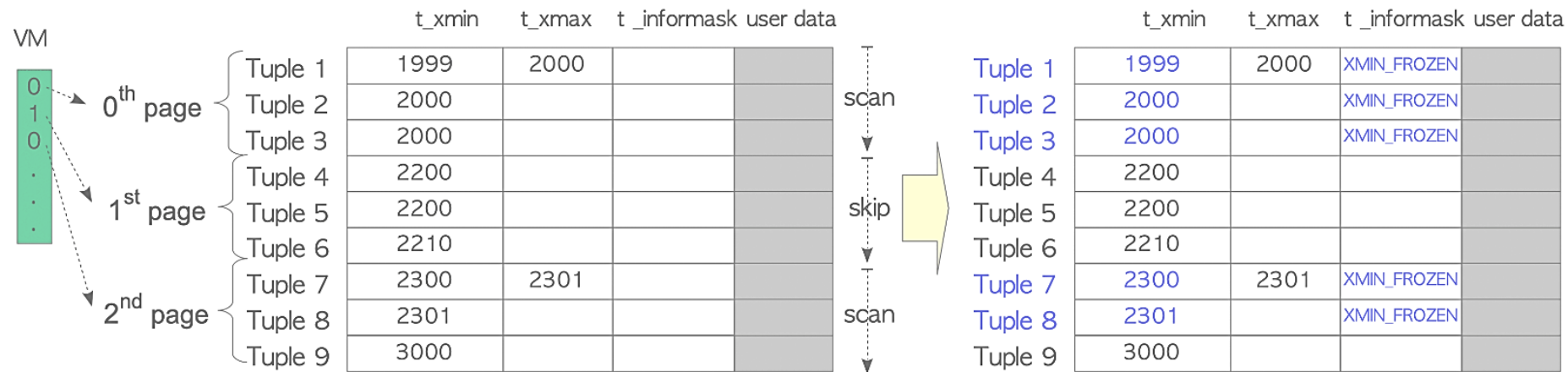
- Идентификаторы транзакций имеют ограниченный размер (32 бита)
- Кластер, который работает в течение длительного времени (**более 4 миллиардов транзакций**), может столкнуться с проблемой циклического переноса идентификаторов транзакций:
 - При достижении максимально-возможного значения счетчик XID обнуляется
 - Транзакции, которые были в прошлом, оказываются в будущем — а это значит, что их вывод становится невидимым
 - Это может привести к потере данных — данные будут недоступны из-за некорректного снимка
- **Решение:**
 - Помечать строки, вставленные транзакцией в далеком прошлом, **как замороженные**, чтобы они наверняка были видны для всех текущих и будущих транзакций
 - Это необходимо выполнять для каждой таблицы в каждой БД по крайней мере один раз каждые **2 миллиарда транзакций**



Процесс заморозки (Lazy Mode)

- При выполнении команды **VACUUM**
 - определяется **OldestXmin** – **txid** самой старой среди текущих транзакций
 - вычисляется **freezeLimit_txid**
 - замораживаются кортежи, **t_xmin** которых меньше, чем **freezeLimit_txid**
- **Vacuum_Freeze_min_age** – параметр конфигурации (по умолчанию 50 000 000)
- «Ленивый режим» не может полностью заморозить кортежи, поскольку он может пропускать страницы

OldestXmin = 50,002,500
freezeLimit_txid = 2500 (= OldestXmin - vacuum_freeze_min_age)





Процесс заморозки (Eager Mode)

- Сканирует все страницы, за исключением тех, которые содержат только замороженные кортежи:
 - обновляет соответствующие системные каталоги
 - по возможности удаляет ненужные файлы и страницы
- Активный режим выполняется, когда выполняется следующее условие

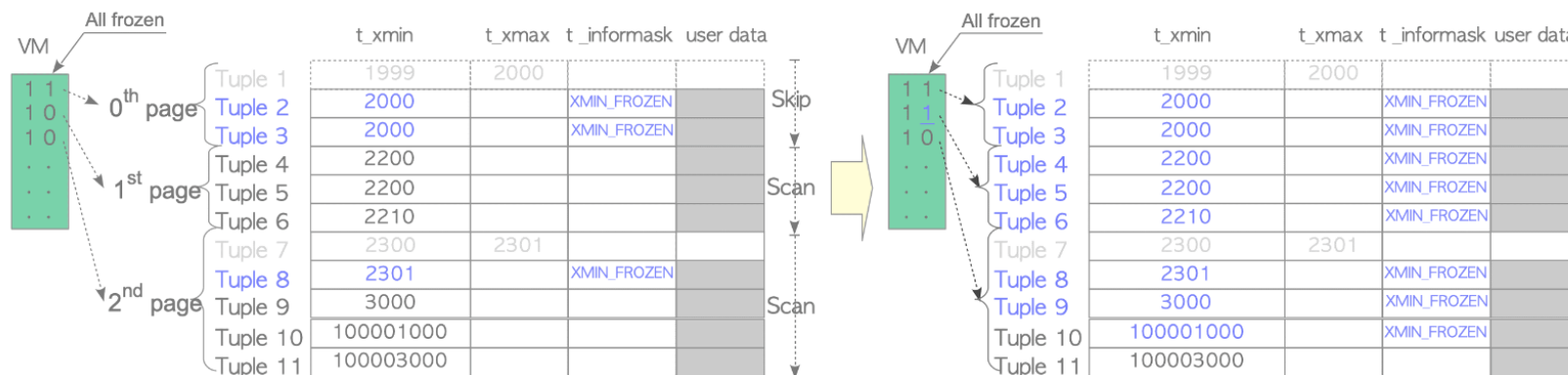
`pg_database.datfrozenxid < (OldestXmin - vacuum_freeze_table_age)`

- **`pg_database.datfrozenxid`** — самый старый замороженный **txid** для каждой БД (**`relfrozenxid`**)
- **`vacuum_freeze_table_age`** — параметр конфигурации (по умолчанию 150 000 000)

OldestXmin = 150,002,000
freezeLimit_txid = 100,002,000

`pg_class.relfrozenxid(Table_1) = 1821`

`pg_class.relfrozenxid(Table_1) = 100,002,000`



Столбец **`pg_class.relfrozenxid`** содержит последний зафиксированный **xid** соответствующей таблицы