PostgreSQL-Consulting

data egret

Давайте отключим автовакуум!?

PGConf.Russia 2018, Moscow

Alexey Lesovsky

alexey.lesovsky@dataegret.com

01 **Вакуум это источник бед и несчастий!**

02 **Или вакуум это средство от бед и несчастий?**

Как приготовить вакуум.

03

01

Сплошные проблемы от вашего вакуума

01 Коллеги, что там с базой?!?

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          27.28    0.11    1.91    30.07    0.00   40.62

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sdc               0.00   126.00 1586.00   95.00   452.51    12.52   258.76     1.25   40.42   38.40    2.02  41.27  95.60
sdb               0.00   196.00    0.00 1034.00     0.00   108.48    86.79     1.04   15.03   14.10    0.93  13.98  34.50
sda               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
```

01 Приложение подтормаживает...

- Запросы отвечают медленно.
- Диски загружены на 100%.

01 И запросы отваливаются...

```
ERROR:  canceling statement due to conflict with recovery
DETAIL:  User query might have needed to see row versions that must be removed.
STATEMENT:  SELECT p.name AS product, p.category AS category, price, LAG (price, 1) OVER (
ERROR:  canceling statement due to conflict with recovery
DETAIL:  User query might have needed to see row versions that must be removed.
STATEMENT:  SELECT p.name AS product, p.category AS category, price, LAG (price, 1) OVER (
ERROR:  canceling statement due to conflict with recovery
DETAIL:  User query might have needed to see row versions that must be removed.
```


01 А что с репликой??

```
LOG:  started streaming WAL from primary at 1/71000000 on timeline 1
FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000100000071 has already been removed
LOG:  started streaming WAL from primary at 1/71000000 on timeline 1
FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000100000071 has already been removed
LOG:  started streaming WAL from primary at 1/71000000 on timeline 1
FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000100000071 has already been removed
LOG:  started streaming WAL from primary at 1/71000000 on timeline 1
FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000100000071 has already been removed
LOG:  started streaming WAL from primary at 1/71000000 on timeline 1
FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000100000071 has already been removed
LOG:  started streaming WAL from primary at 1/71000000 on timeline 1
FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000100000071 has already been removed
```

01 Кажется, вакуум все положил

```
Total DISK READ :     433.54 M/s | Total DISK WRITE :     113.21 M/s
Actual DISK READ:     427.97 M/s | Actual DISK WRITE:     110.10 M/s
   PID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN      IO>    COMMAND

 95432 idle postgres      2.20 G    122.10 M  0.00 %  22.10 % postgres: autovacuum worker process   asia_engine
 87669 idle postgres      1.80 G     96.49 M  0.00 %  18.56 % postgres: autovacuum worker process   asia_engine
122509 idle postgres      1.61 G     80.01 M  0.00 %  17.73 % postgres: autovacuum worker process   asia_engine
  2197 be/3 root          0.00 B      0.00 B  0.00 %  15.01 % [jbd2/sdb1-8]
 62816 idle postgres      2.48 G    134.15 M  0.00 %   9.12 % postgres: autovacuum worker process   asia_engine
 81627 be/4 postgres      0.00 B    816.09 M  0.00 %   8.10 % postgres: wal writer process
 92626 idle postgres      1.81 G     48.22 M  0.00 %   5.56 % postgres: autovacuum worker process   asia_engine
109172 idle postgres    799.50 M     13.84 M  0.00 %   4.83 % postgres: autovacuum worker process   asia_engine
 87818 idle postgres   1114.20 M     67.20 M  0.00 %   2.92 % postgres: autovacuum worker process   asia_engine
105261 idle postgres    325.00 M     48.51 M  0.00 %   0.73 % postgres: autovacuum worker process   asia_engine
111821 idle postgres    401.00 M     55.90 M  0.00 %   0.41 % postgres: autovacuum worker process   asia_engine
  5936 be/4 postgres     31.00 K      8.00 K  0.00 %   0.03 % postgres: asia_api asia_engine [local] idle
 12428 be/4 postgres      0.00 B    122.00 K  0.00 %   0.00 % postgres: logger process
```


Таки, что делать?

01 Решение!?

autovacuum = off

01 Вроде всё хорошо, но что-то не так

- Про статистику планировщика можно забыть.

01 Вроде всё хорошо, но что-то не так

- Про статистику планировщика можно забыть.
- Таблицы и индексы начнут пухнуть.

01 Вроде всё хорошо, но что-то не так

- Про статистику планировщика можно забыть.
- Таблицы и индексы начнут пухнуть.
- Неэффективное использование shared buffers.

01 Вроде всё хорошо, но что-то не так

- Про статистику планировщика можно забыть.
- Таблицы и индексы начнут пухнуть.
- Неэффективное использование shared buffers.
- Снижение общей производительности.

01 Оказывается, что всё не очень хорошо

Практический тест и как воспроизвести – https://goo.gl/Tql87l

- До: 3565.5 tps, 0.839 ms, 3% от shared_buffers.
- После: 172.8 tps, 17.373 ms, 21% от shared_buffers.

02

Вакуум?
Ну да,
я что-то
слышал
про него

02 Зачем нужен вакуум?

Все знают что такое MVCC?

02 Зачем нужен вакуум?

MVCC – Multi-Version Concurrency Control:

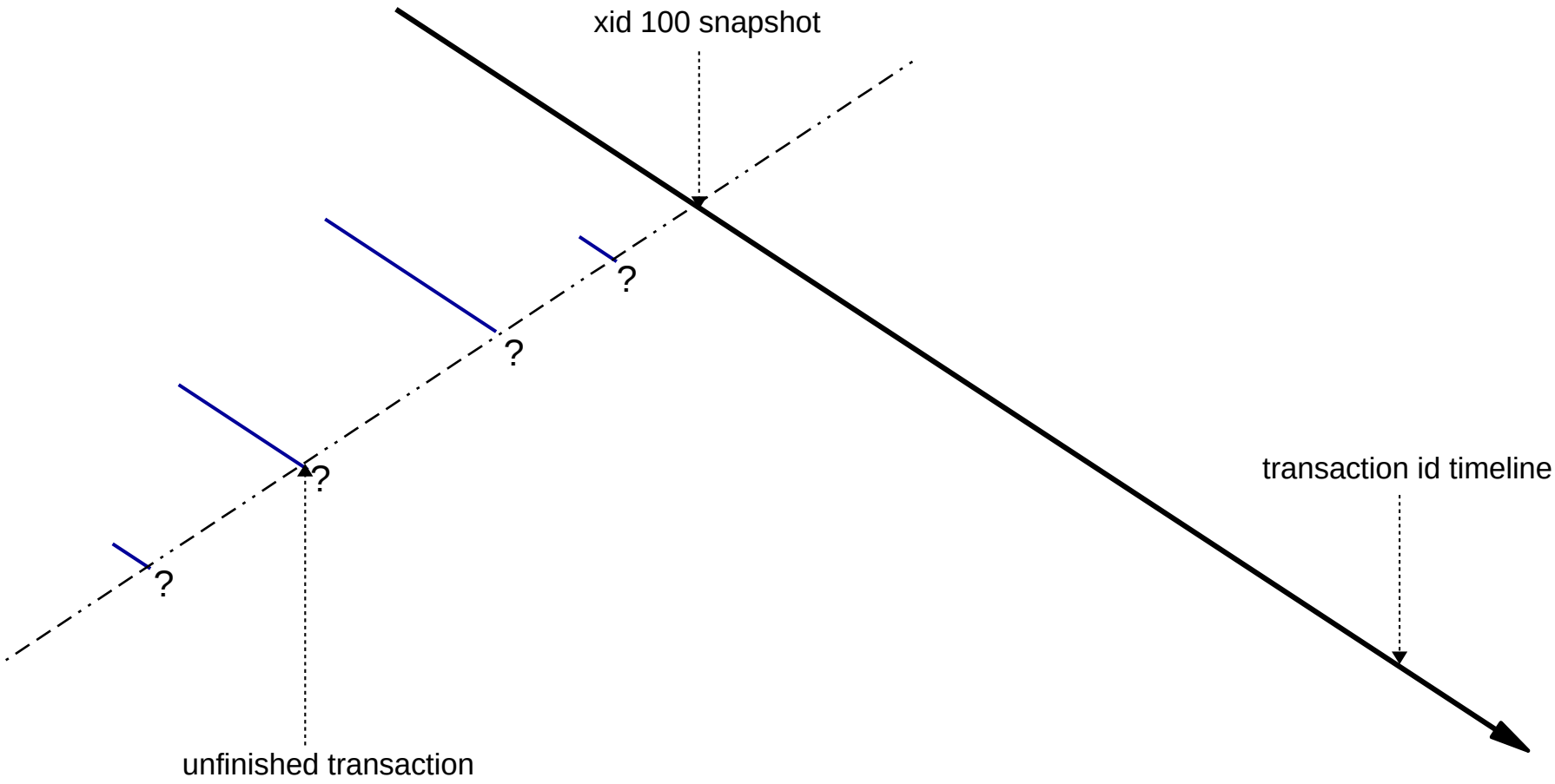
- Хорошая производительность при конкурентном доступе.
- При высокой активности на чтение и запись.
- Читатели не блокируют читателей; Писатели не блокируют писателей.

02 Зачем нужен вакуум?

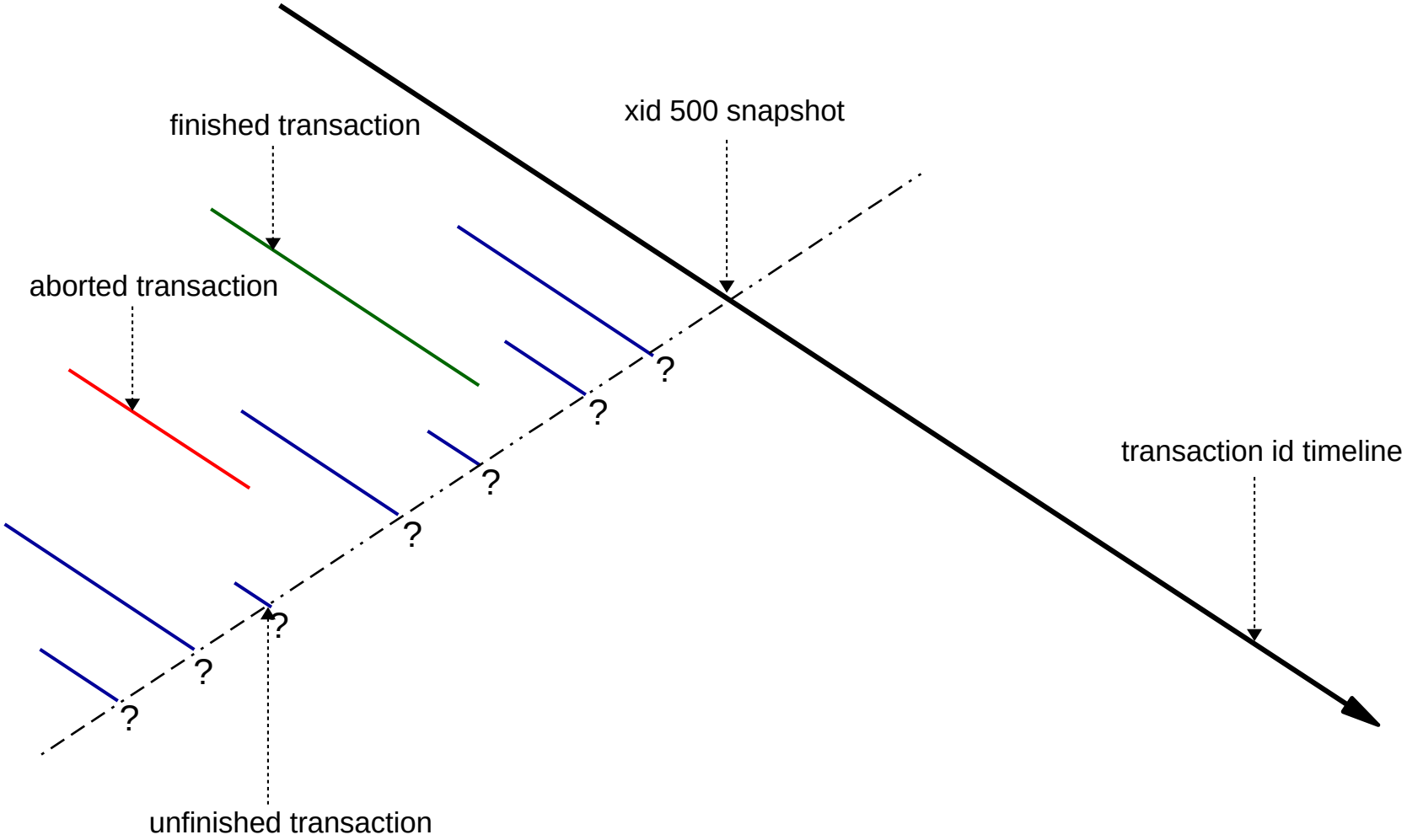
MVCC – Multi-Version Concurrency Control:

- Хорошая производительность при конкурентном доступе.
- При высокой активности на чтение и запись.
- Читатели не блокируют читателей; Писатели не блокируют писателей.
- *Почти ;)*

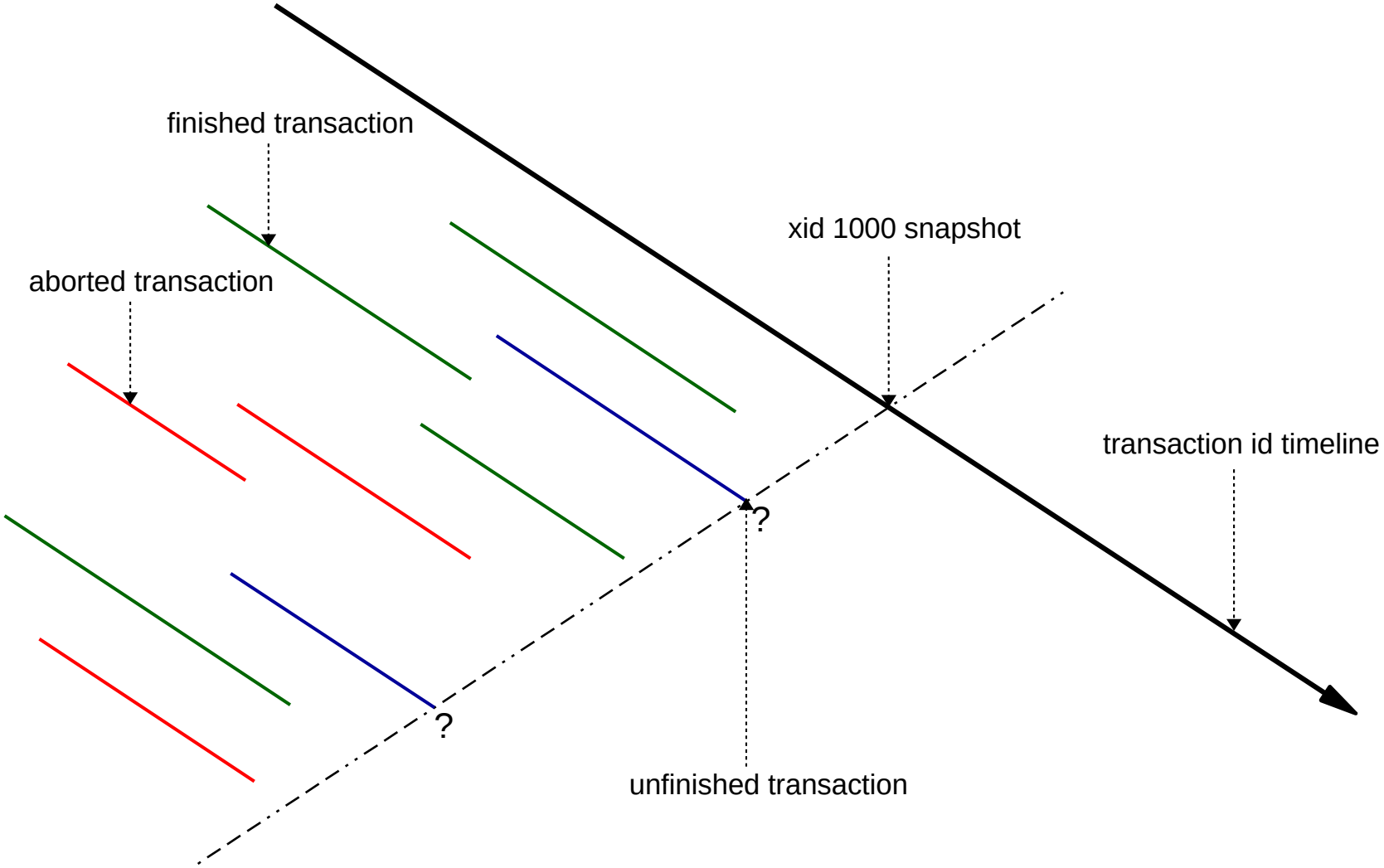
02 MVCC


xid 100 snapshot
?
?
?
?
transaction id timeline
unfinished transaction

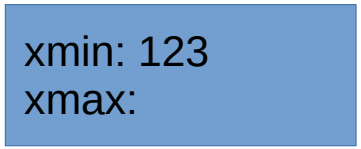
02 MVCC


finished transaction
xid 500 snapshot
aborted transaction
?
?
?
?
?
?
transaction id timeline
unfinished transaction

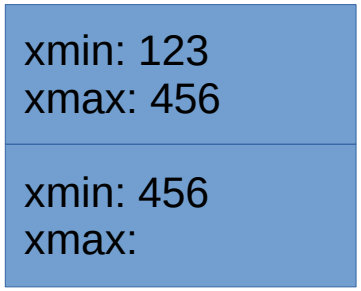
02 MVCC


finished transaction
aborted transaction
xid 1000 snapshot
transaction id timeline
?
?
unfinished transaction

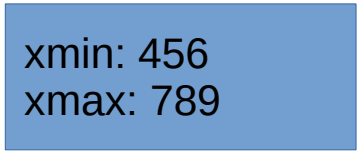
02 MVCC



INSERT строки транзакцией №123

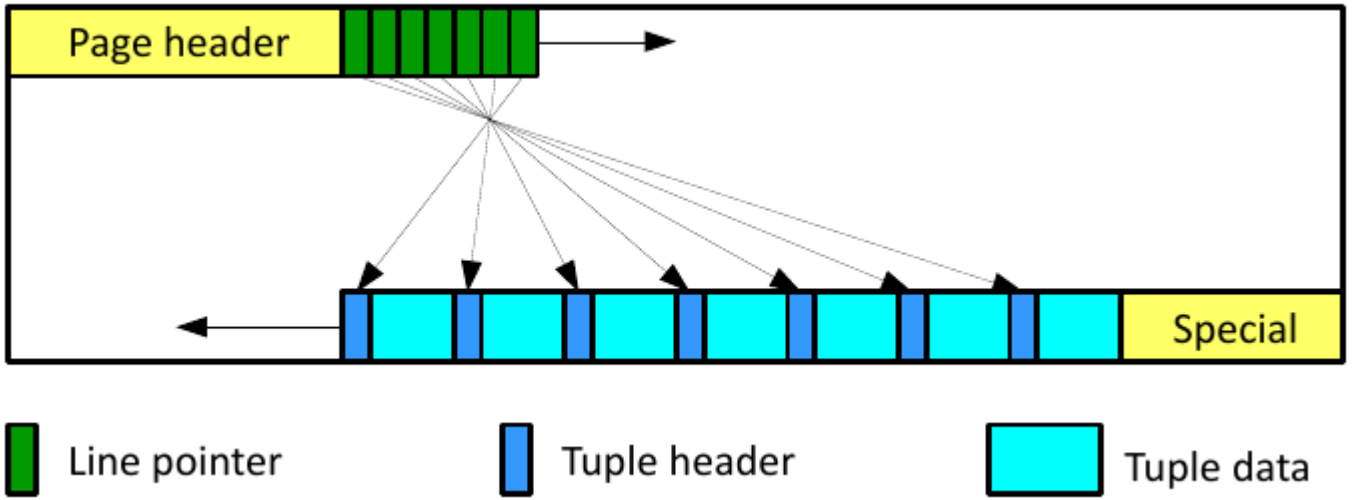


UPDATE строки транзакцией №456

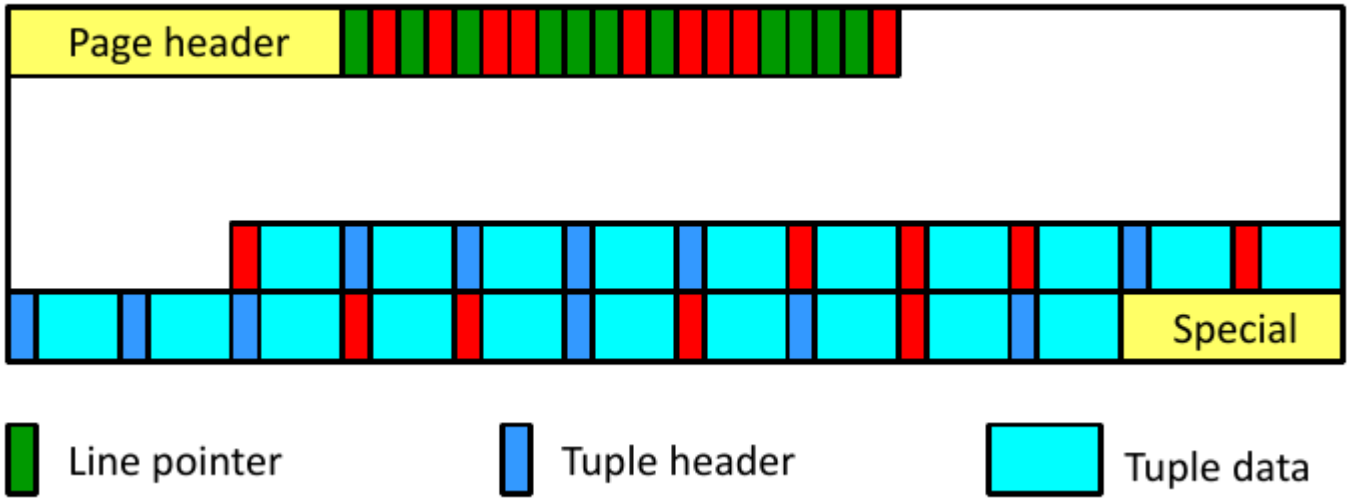


DELETE строки транзакцией №789

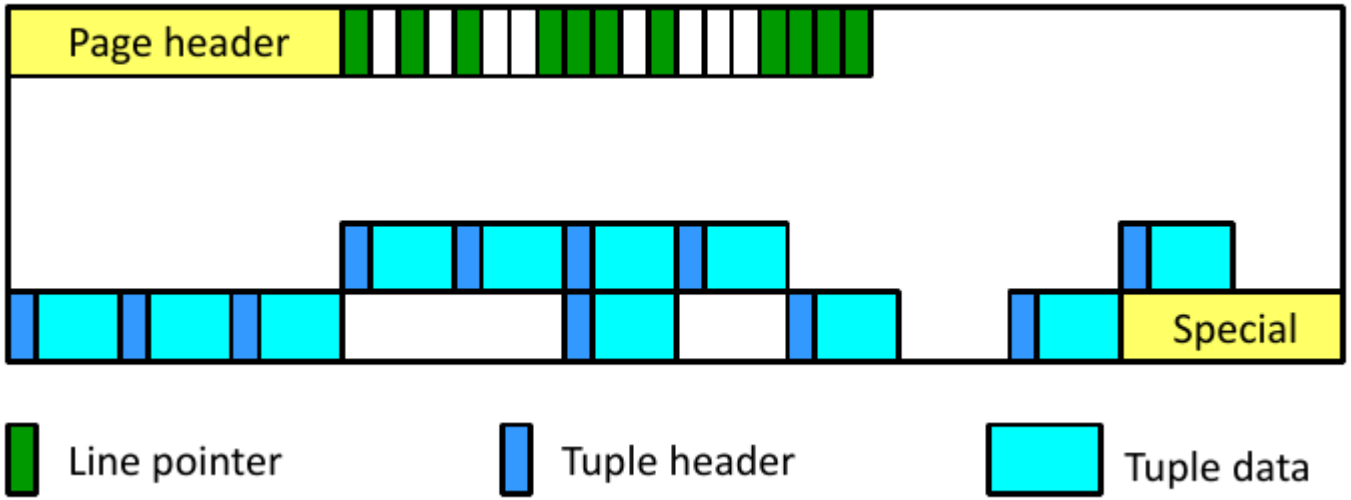
02 Как обстоят дела на уровне страницы



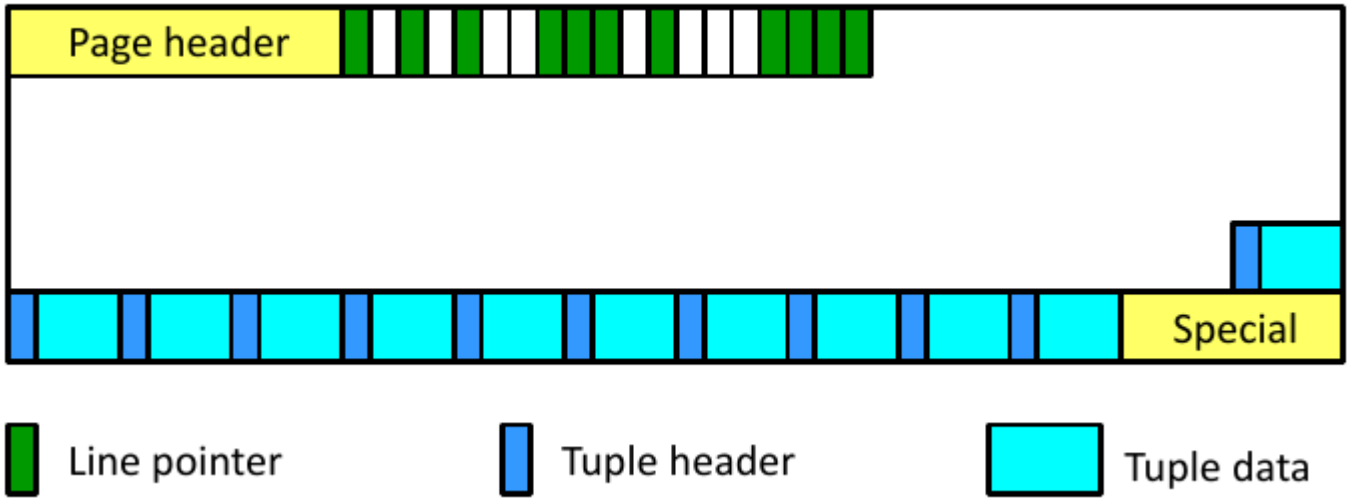
02 Как обстоят дела на уровне страницы



02 Как обстоят дела на уровне страницы



02 Как обстоят дела на уровне страницы



02 Еще раз о главном

- Сохранение общей производительности.
- Эффективное использование shared buffers.
- Минимизация *«bloat»* эффекта.
- И конечно собирается статистика планировщика.

02 Как обстоят дела с вакуумом

- Autovacuum это **фоновая** задача/штука/процесс:
  - Включен по-умолчанию, ограничен в количестве.
  - Запускается с некоторым интервалом.
  - Также собирает статистику для планировщика.

02 Как обстоят дела с вакуумом

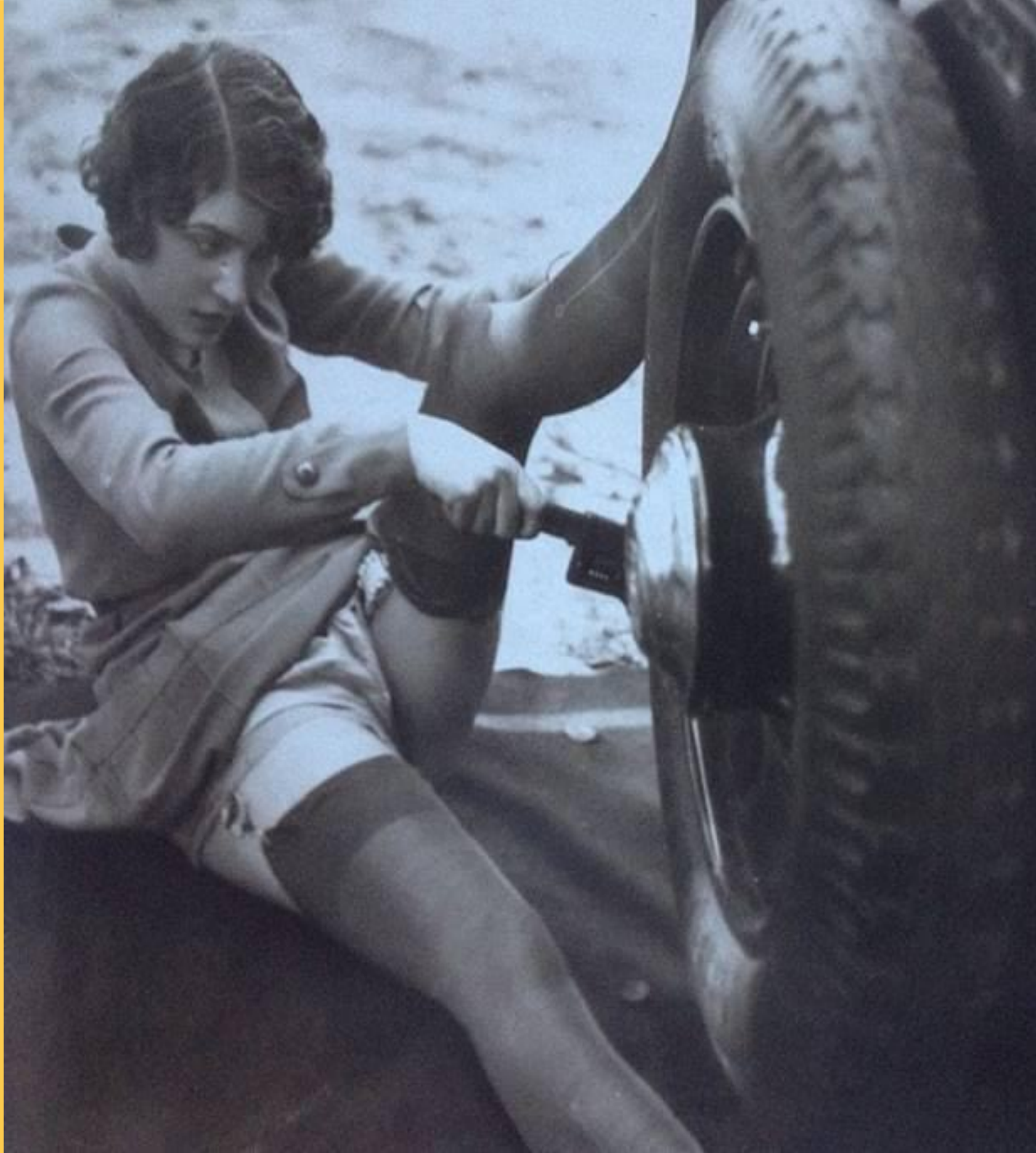
- Обрабатываются базы/таблицы/индексы по списку:
  - Первыми идут базы, где есть риск *wraparound*.
  - Далее – те базы что давно не обрабатывались.
  - Таблицы где накопилось много *«мертвых»* строк.

02 Как обстоят дела с вакуумом

- Настройки по-умолчанию – никуда не годятся.
- Лучше всего, дела с вакуумом обстоят начиная с 9.6

03

Ну ок, давайте затюним вакуум



03 С чего начать?

- Во-первых (auto)vacuum он всегда cost-based:
  - *vacuum_cost_limit*
  - *vacuum_cost_delay*
  - *vacuum_cost_page_hit*
  - *vacuum_cost_page_miss*
  - *vacuum_cost_page_dirty*

03 С чего начать?

- Во-вторых, рабочих может быть много
  - *autovacuum_max_workers*
  - *autovacuum_naptime*
  - *vacuum_cost_limit* делится между всеми активными воркерами

03 С чего начать?

- В-третьих, запуск вакуума зависит от количества «*мертвых*» строк
  - *autovacuum_vacuum_threshold*
  - *autovacuum_vacuum_scale_factor*

03 С чего начать?

- В-третьих, запуск вакуума зависит от количества «*мертвых*» строк
  - *autovacuum_vacuum_threshold*
  - *autovacuum_vacuum_scale_factor*

n_dead_tup > (reltuples * scale_factor) + threshold

03 Что использовать?

- Scale factor vs. Threshold

03 Разные диски?

- HDD – да, еще встречается.

03 Разные диски?

- HDD – да, еще встречается.
- SSD – и даже их производительности иногда не хватает.

03 Разные диски?

- HDD – да, еще встречается.
- SSD – и даже их производительности иногда не хватает.
- NVME – зачем вы здесь?

03 Может есть универсальное правило?

- Общее правило – регулируем *delay* и *limit*.

03 Пример настройки для SSD

```
vacuum_cost_delay = 0
vacuum_cost_page_hit = 0
vacuum_cost_page_miss = 5
vacuum_cost_page_dirty = 5
vacuum_cost_limit = 200
--
autovacuum_max_workers = 10
autovacuum_naptime = 1s
autovacuum_vacuum_threshold = 50
autovacuum_analyze_threshold = 50
autovacuum_vacuum_scale_factor = 0.05
autovacuum_analyze_scale_factor = 0.05
autovacuum_vacuum_cost_delay = 5ms
autovacuum_vacuum_cost_limit = -1
```


03 О чем еще стоит помнить

- *Storage parameters* – когда глобальные настройки не подходят:
  - ALTER TABLESPACE my_tblspc SET (storage_parameter);
  - ALTER TABLE my_table SET (storage_parameter);
  - ALTER INDEX my_index SET (storage_parameter);

03 Нетрадиционная медицина

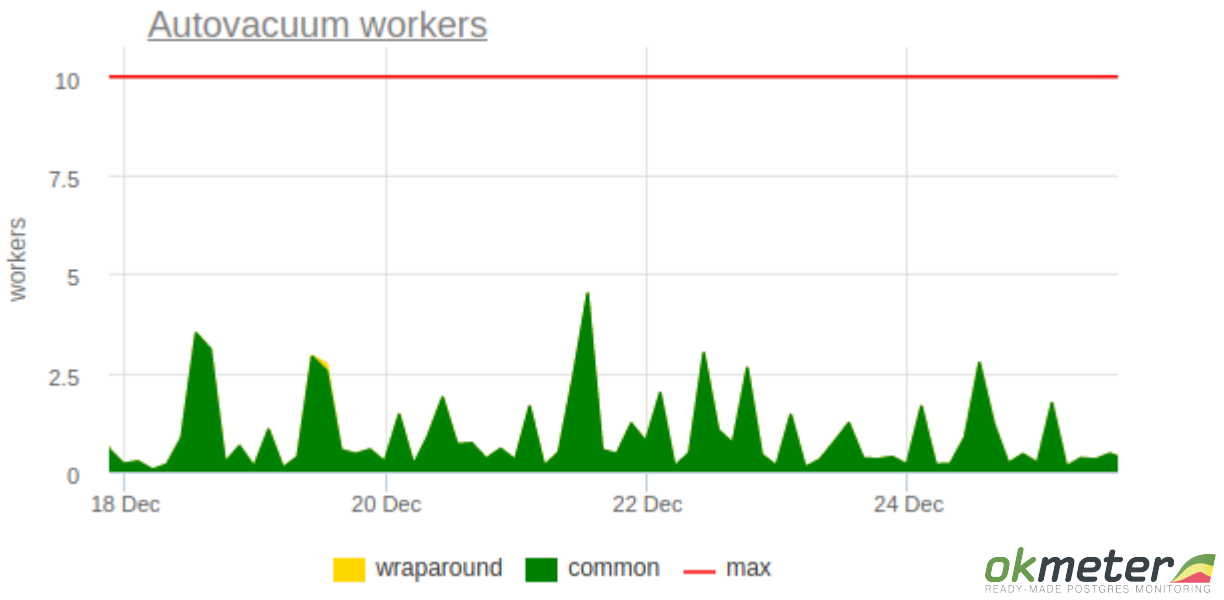
- pgcompacttable – долгий, легкий, безопасный.
- pg_repack – быстрый, простой, надежный, но иногда небезопасный.

Пара слов про мониторинг…

03 Как мониторить вакуум

- pg_stat_activity – должно быть в любом мониторинге.
  - Количество и тип воркеров.
  - Длительность работы воркеров.

03 Как мониторить вакуум



03 Как мониторить вакуум

pg_stat_progress_vacuum – когда нужно посмотреть детали.

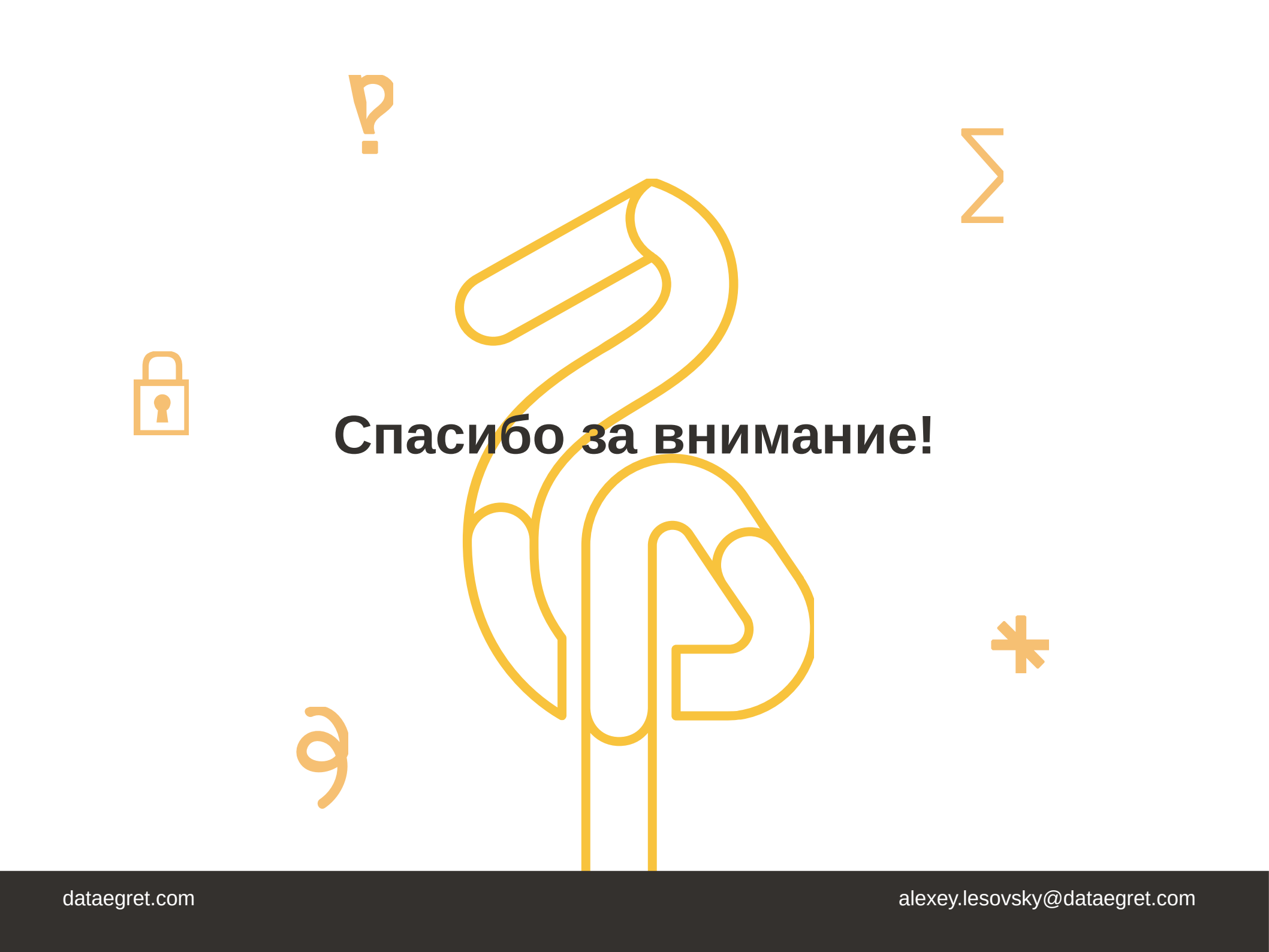
03 Как мониторить вакуум

https://github.com/lesovsky/uber-scripts/blob/master/postgresql/sql/vacuum_activity.sql

```
-[ RECORD 1 ]-------+------------------
pid                 | 104701
duration            | 03:21:51.330818
waiting             | f
mode                | regular
database            | analytics
table               | events
phase               | vacuuming indexes
table_size          | 1188 GB
total_size          | 1682 GB
scanned             | 601 GB
vacuumed            | 571 GB
scanned_pct         | 50.0
vacuumed_pct        | 48.0
index_vacuum_count  | 6
dead_tup_pct        | 100.0
```


03 Что в итоге?

- Вакуум это не сложно.
- **Вакуум отключать нельзя.**
- Вакуум это хорошо.



Спасибо за внимание!

alexey.lesovsky@dataegret.com