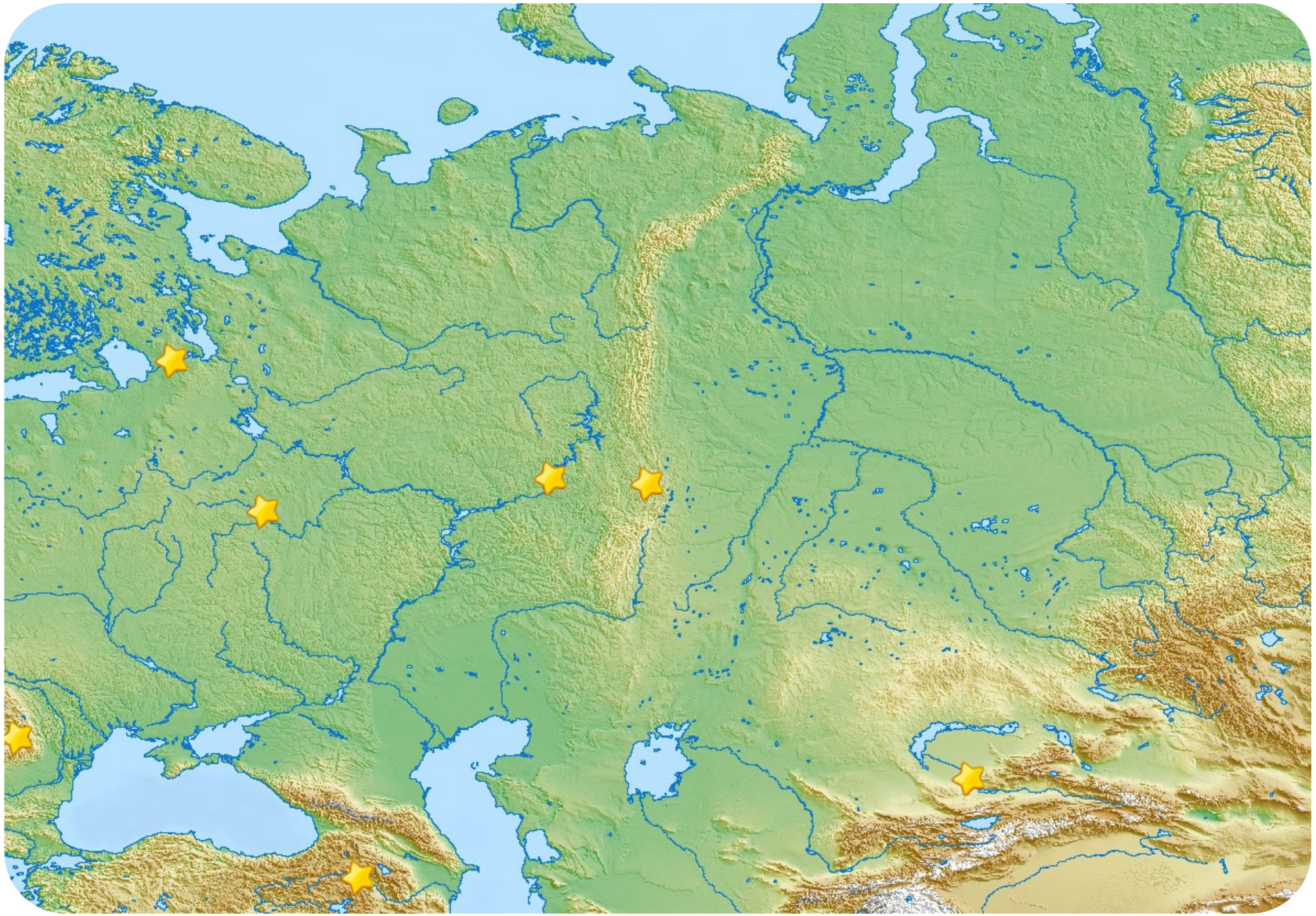
Yandex Cloud

Планы выполнения в pg_stat_statements

Леонид Борчук,
Team Lead Greenplum,
Yandex Cloud

Наша команда



О чём сегодня поговорим

1. Про планы выполнения
2. Какие есть расширения
3. Pg_stat_query_plans
4. Сравниваем с pg_stat_statements

1. Про планы выполнения
2. Какие есть расширения
3. Pg_stat_query_plans
4. Сравниваем
   с pg_stat_statements

Запросов много — группируем их

```
bench=# SELECT
  query,
  calls,
  total_exec_time,
  rows,
  100.0 * shared_blks_hit / nullif(shared_blks_hit + shared_blks_read, 0) AS hit_percent
FROM pg_stat_statements
ORDER BY total_exec_time DESC LIMIT 5;

-[ RECORD 1 ]---+-----------------------------------------------------------
query           | UPDATE pgbench_branches SET bbalance = bbalance + $1 WHERE bid = $2
calls           | 3000
total_exec_time | 25565.855387
rows            | 3000
hit_percent     | 100.0000000000000000
```

Похожие запросы, но разные планы выполнения

```
# create table ids as select 1 as id from generate_series(1, 9999);
# insert into ids values (2);
# create index ids_i1 on ids(id);
```

```
Index Only Scan using ids_i1 on ids
(cost=0.29..2.30 rows=1 width=4) (actual
time=0.035..0.036 rows=1 loops=1)
  Index Cond: (id = 2)
  Heap Fetches: 1
Planning Time: 0.158 ms
Execution Time: 0.100 ms
```

```
Seq Scan on ids  (cost=0.00..170.00
rows=9′999 width=4) (actual
time=0.012..0.835 rows=9′999 loops=1)
  Filter: (id = 1)
  Rows Removed by Filter: 1
Planning Time: 0.063 ms
Execution Time: 1.195 ms
```

Зачем знать план выполнения

- План нужен для оптимизации производительности
- Удобно не смешивать статистику разных планов выполнения
- Удобно видеть конкретные запросы и конкретные планы выполнения
- Всё должно продолжать работать быстро!

1. Про планы выполнения
2. Какие есть расширения
3. Pg_stat_query_plans
4. Сравниваем
   с pg_stat_statements

Pg_stat_plans

github.com/2ndQuadrant/pg_stat_plans

Author: Peter Geoghegan

Based on an idea by Peter Geoghegan and Simon Riggs

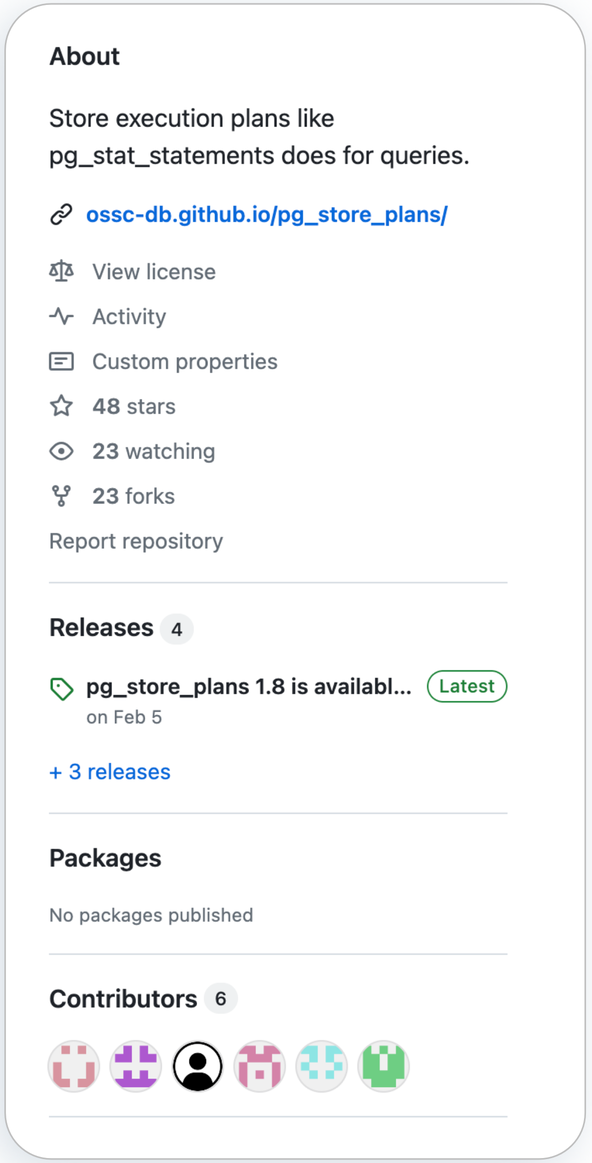
Introduction

pg_stat_plans is a variant of the standard Postgres contrib module pg_stat_statements. It differentiates between and assigns execution costs to plans, rather than queries. This makes it particularly suitable for monitoring query planner regressions. However, it is also suitable as a general-purpose tool for understanding execution costs at both the plan and query granularity

Не обновлялся 11 лет

Pg_store_plans

github.com/ossc-db/pg_store_plans



Pg_stat_monitor

github.com/percona/pg_stat_monitor

- Time Interval Grouping
- Multi-Dimensional Grouping
- Capture Actual Parameters in the Queries
- Query Plan
- Tables Access Statistics for a Statement
- Histogram

```
716                     plan_info.plan_len = (rv < PLAN_TEXT_LEN) ? rv : PLAN_TEXT_LEN - 1;
717                     plan_info.planid = pgsm_hash_string(plan_info.plan_text, plan_info.plan_len);
718                     plan_ptr = &plan_info;
```

1. Про планы выполнения
2. Какие есть расширения
3. Pg_stat_query_plans
4. Сравниваем
   с pg_stat_statements

Pg_stat_query_plans

github.com/postgredients/pg_stat_query_plans

Pg_stat_query_plans

- Сохраняет план выполнения
- Генерирует шаблонный план выполнения, вычисляет plan_id по нему
- Сохраняет запрос с конкретными значениями предикатов
- Сохраняет статистику выполнения с точностью до plan_id
- Хранит тексты в памяти в сжатом (pgz) виде
- Работает для PG11–PG16

Вычисление plan_id

1. Сохраняем план выполнения запроса в виде текста
2. Парсим вывод core_yylex
3. Заменяем все вхождения ?CONST на символы $<num>
4. От получившегося плана выполнения считаем hash

```
UPDATE pgbench_tellers SET tbalance = tbalance + $1 WHERE tid = $2

Update on public.pgbench_tellers
 -> Bitmap Heap Scan on public.pgbench_tellers
      Output: (tbalance + $1::integer), ctid
      Recheck Cond: (pgbench_tellers.tid = $2)
      -> Bitmap Index Scan on pgbench_tellers_pkey
           Index Cond: (pgbench_tellers.tid = $3) +
```

Больше 4000 кластеров

ID запроса	Хост	User	База данных	Query	Графики	Calls	Total time
ca0d11c1c458ab6a	rc1b-ujdxubgfal9fpsx3	user1	db1	UPDATE pgbench_branches SET bbalance = bb...		360,000	7,683,267.26
ID плана выполнения	**Plan rows/width**	**Total cost**	**Startup cost**	**Normalized plan**			
e944987da4d943a4	0 строк/0 Б	0	0	Update on public.pgbench_branches -> Index Sca...		179,928	3,854,069.763
f8fd4b8b9de3db2c	0 строк/0 Б	0	0	Update on public.pgbench_branches -> Index Sc...		180,072	3,829,197.496

Query df7f85f9e7f43648

```
SELECT schema_name FROM information_schema.schemata
```

Показать полностью | Explain analize

Граф плана | Текст плана | JSON

Hash Join 0ms 0%
Inner join

Seq Scan <1ms 0%
Most expensive
Stats | I/O & Buffers | Misc | Info

Total Cost	1.13
* Actual Cost	1.13
* Actual Duration (ms)	0
Plan Rows	13
Schema	pg_catalog
Plan Width	4

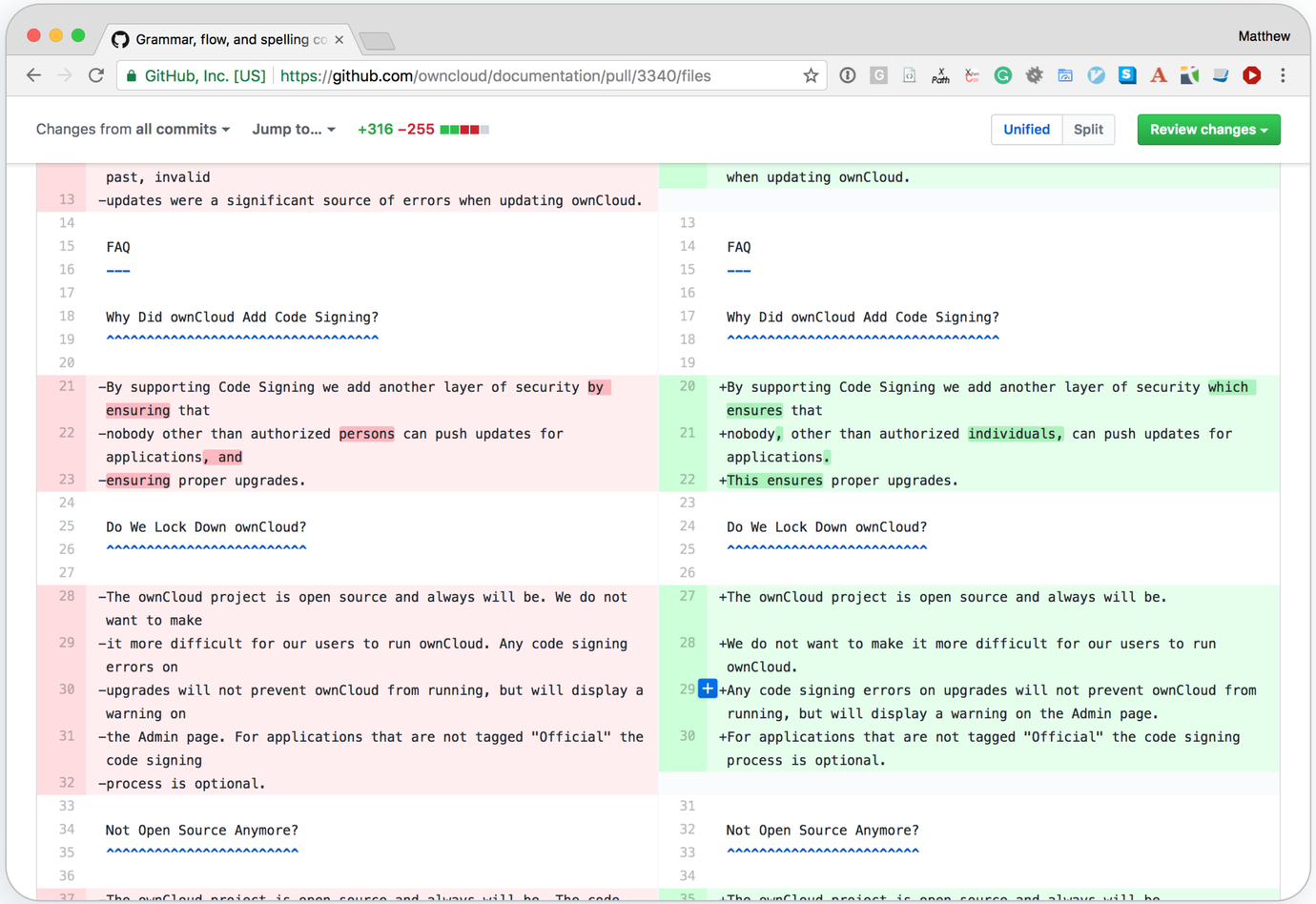
Hash 0ms 0%
Stats | I/O & Buffers | Misc | Info

Startup Cost	1.06
Total Cost	1.06
* Actual Duration (ms)	NaN
Plan Rows	2
Plan Width	68

Seq Scan <1ms 0%
Expensive
Stats | I/O & Buffers | Misc | Info

Total Cost	1.06
* Actual Cost	1.06
* Actual Duration (ms)	0
Plan Rows	2
Schema	pg_catalog
Plan Width	68
Filter	(pg_has_role(n.nspowner, 'USAGE'::text) OR has_schem

Как всё устроено



PostgreSQL

ClickHouse

PostgreSQL

Excel

gadget.wikisort.org/camera/ru

1. Про планы выполнения
2. Какие есть расширения
3. Pg_stat_query_plans
4. Сравниваем
   с pg_stat_statements

#1. Парсинг не влияет на performance

- Pg_stat_statements
  +15% к времени выполнения

  В тестах pgbench
- Pg_stat_query_plans
  +30% к времени выполнения

  В тестах pgbench

- Незаметен на многих нагрузках
- Но видели ожидания pg_stat_query_plans в pg_stat_activity

#2. Планы выполнения меняются

count(distinct queryid, planid) / count(distinct queryid) = 1,05

```
UPDATE pgbench_tellers SET tbalance = tbalance + $1 WHERE tid = $2

 Update on public.pgbench_tellers
   ->  Index Scan using pgbench_tellers_pkey on public.pgbench_tellers
         Output: (tbalance + $1::integer), ctid
         Index Cond: (pgbench_tellers.tid = $2)

 Update on public.pgbench_tellers
   ->  Bitmap Heap Scan on public.pgbench_tellers
         Output: (tbalance + $1::integer), ctid
         Recheck Cond: (pgbench_tellers.tid = $2)
         ->  Bitmap Index Scan on pgbench_tellers_pkey
               Index Cond: (pgbench_tellers.tid = $3)
```

#3. Длинный текст запроса — исключение

- Запросы с текстом длиной более 1 МБ — 0,008%
- Запросы с текстом длиной более 100 МБ — есть. Обычно это заливка данных

- Есть БД, где файл с текстами разрастается
- Тексты запросов очень хорошо пакуются (5x–7x)

#4. Сохранение текста запроса на диске влияет на performance

- 300 МБ достаточно для хранения текстов запросов и планов
- В современных системах 300 МБ — не проблема
- Текст сохраняется при появлении нового запроса
- Редкое событие и почти не влияет на performance

#5. Неблокирующий вызов хука — нужен!

```
464         if (lock_iteration_count == 0)
465             LWLockAcquire(pgqp->lock, LW_EXCLUSIVE);
466         else
467             for (int i = 0; i < lock_iteration_count; i++)
468             {
469                     if (LWLockConditionalAcquire(pgqp->lock, LW_EXCLUSIVE))
470                             break;
471                     if (i > lock_iteration_count / 2)
472                             pg_usleep(i * lock_backoff_timeout);
473                     if (i == lock_iteration_count - 1)
474                     {
475                             elog(DEBUG1, "could not enter query information to pgqp_ya");
476                             return;
477                     }
478             }
```

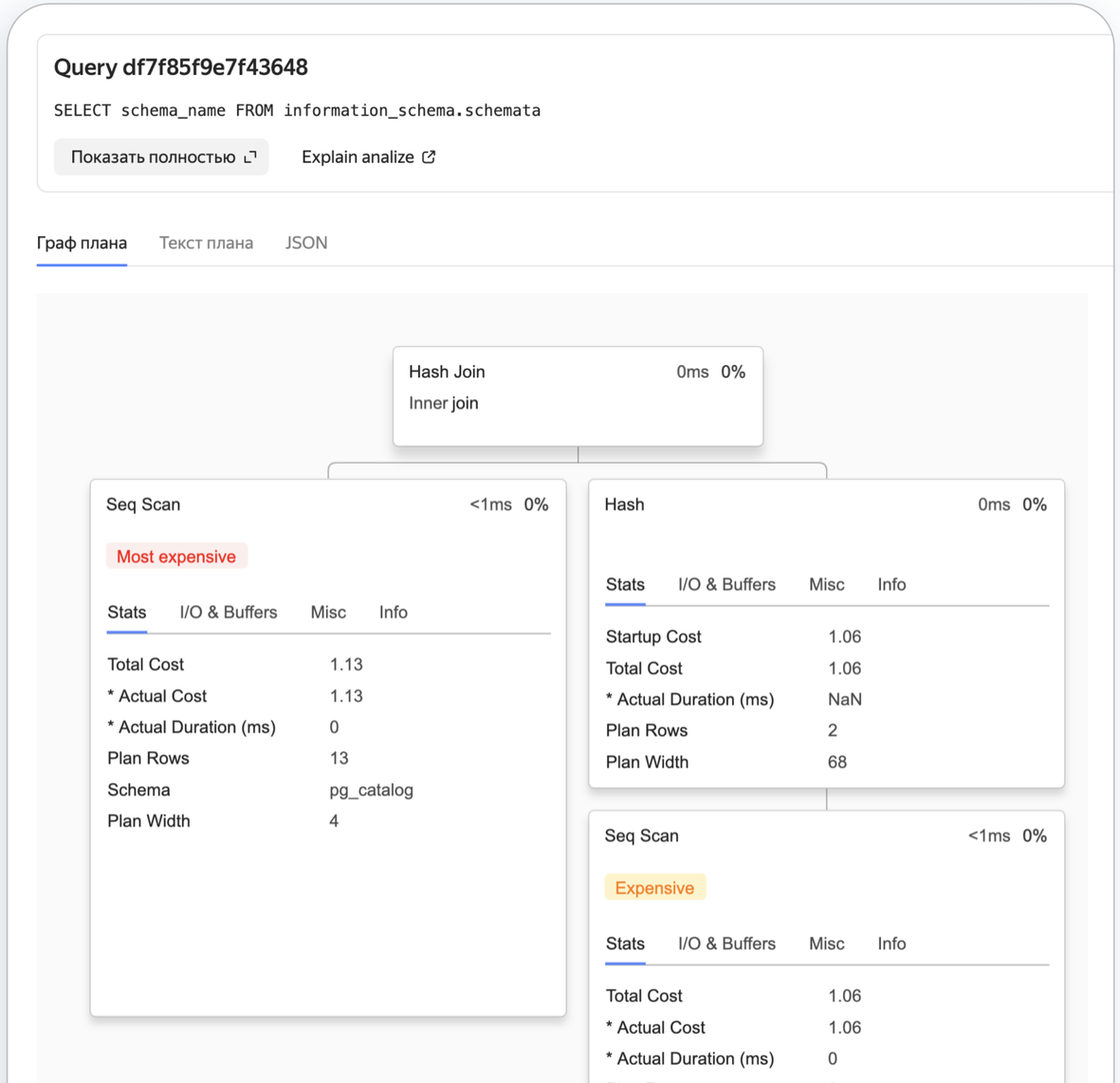
#6. Pg_stat_statements можно заменить

Слишком усложняет код

- Две hash-таблицы
- Сложная процедура удаления запросов и планов
- Нужны счётчики для подсчёта ссылок на запросы
- Сложная процедура удаления текстов запросов и планов

Множество скриптов
и утилит умеют использовать
pg_stat_statements

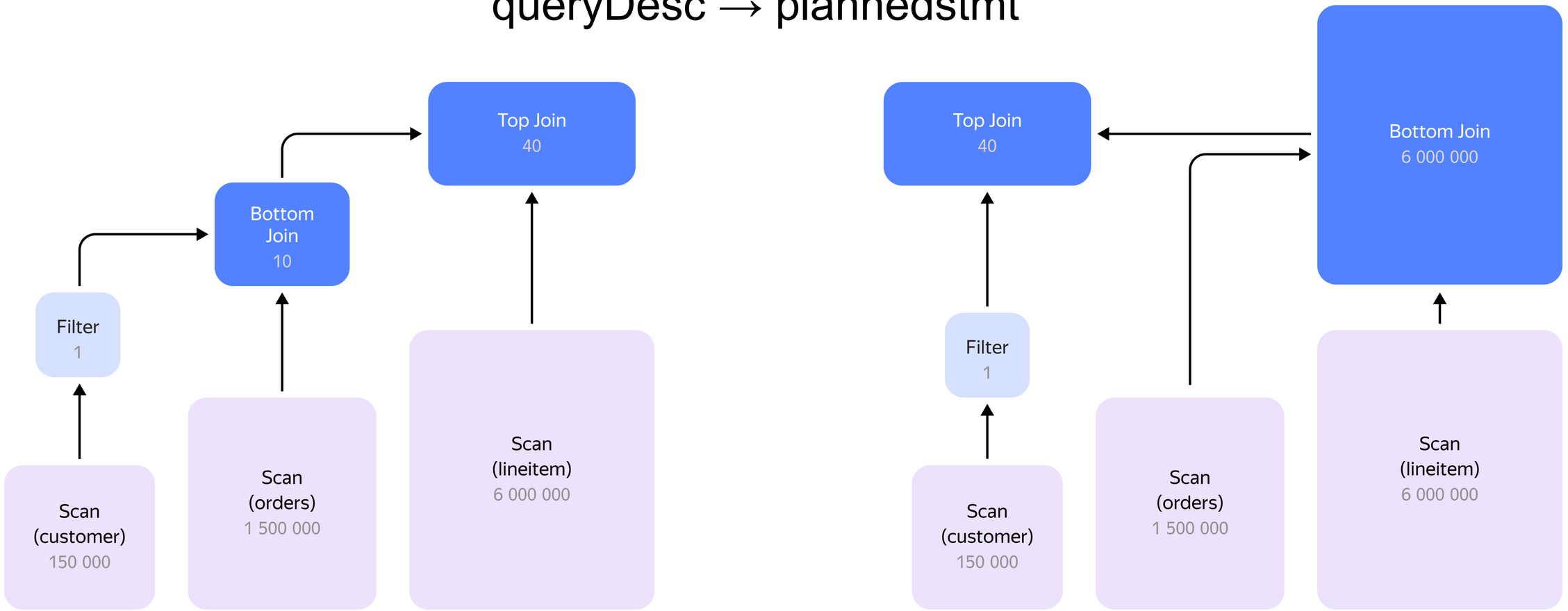
#7. Точный текст запроса нужен



Показываем
первый текст

#8. Plan_id можно вычислять без парсинга

queryDesc → plannedstmt



Нужно вычислять
plan_id
по аналогии
с query_id

1

Разные способы
группировки
не нужны,
для этого есть
Excel и ClickHouse

2

Лучше
не заменять,
а дополнять
pg_stat_statements

3

Готов ответить на ваши вопросы

Леонид Борчук,
Team Lead Greenplum,
Yandex Cloud