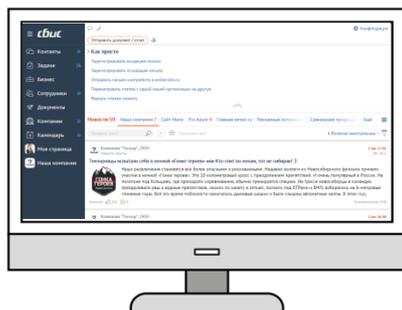


Массовая оптимизация запросов PostgreSQL – explain.sbis.ru

«Тензор» – это СБИС МИЛЛИОН КЛИЕНТОВ

- 100+ проектов
- 10 центров разработки
- более 1000 сотрудников в них



СБИС – сеть деловых ком. X
Надежный | https://sbis.ru

СБИС Сеть деловых коммуникаций Поддержка 24/7 | Тарифы | Контакты

Множество возможностей в рамках одной системы

<p>Электронный документооборот Электронная подпись и обмен документами между компаниями, внутри компании и с обыкновенными людьми</p>	<p>Отчетность через интернет Подготовка, проверка и сдача отчетности во все госорганы, сверка расчетов с бюджетом, переписка с инспекторами</p>	<p>Все о компаниях и владельцах Реквизиты, владельцы, финансовое состояние, стоимость бизнеса и другие самые актуальные сведения о всех компаниях в России</p>	<p>Поиск и анализ закупок Актуальные данные и аналитика со всех торговых площадок, оценка шансов на победу, всё для работы тендерного отдела</p>
<p>Онлайн-кассы и ОФД Сервис оператора фискальных данных, модернизация и онлайн-регистрация ККТ, анализ продаж, мониторинг торговых точек</p>	<p>Точка продаж Автоматизация магазинов и сферы услуг с поддержкой ОФД и ЕГАИС, оснащение кассовых мест «под ключ», четкая логистика и легкий бухгалтер</p>	<p>Заказы и поставки (EDI) Обмен заказами, прайсами, документами и данными о товарах между торговыми сетями и поставщиками</p>	<p>Корпоративная социальная сеть Единое пространство и удобные инструменты для совместной работы и коммуникаций</p>
<p>Управление бизнес-процессами Электронные согласования, лобье</p>	<p>Видеокommunikации Видеозвонки между коллегами</p>	<p>Управление персоналом Учет рабочего времени, KPI,</p>	<p>ещё 9 сервисов</p>

СБИС – data-centric application

Активно используем PostgreSQL

- ~**400ТВ** «рабочих» данных
- «в продакшене» **с 2008 года**
- уже более **250 серверов**

СБИС – data-centric application

SQL – декларативный язык

○ вы описываете, **что** хотите получить

○ СУБД лучше «знает», **как** это сделать:

какие индексы использовать, в каком порядке

соединять таблицы, как накладывать условия, ...

СБИС – data-centric application

SQL – декларативный язык

○ некоторые СУБД принимают «подсказки»

○ PostgreSQL – **нет**, но...

всегда готов рассказать, **как конкретно** он
выполняет ваш запрос

СБИС – data-centric application

Классика: «Почему тут выполнялось долго?»

- неэффективный алгоритм
- неактуальная статистика
- «затык» по ресурсам (CPU, disk, RAM)
- блокировки – для DML-запросов

СБИС – data-centric application

Классика: «Почему тут выполнялось долго?»

- неэффективный алгоритм
- неактуальная статистика
- «затык» по ресурсам (CPU, disk, RAM)

«Нам нужен план!»

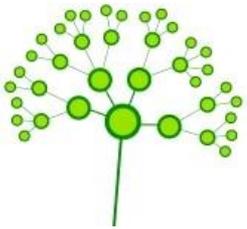


Получение плана

Получение плана

План запроса – **дерево** в текстовом виде

○ каждый элемент – одна из операций



получение данных, построение битовых карт, обработка данных, операция над множествами, соединение, вложенный запрос

○ выполнение плана – обход дерева

Получение плана

EXPLAIN (ANALYZE, BUFFERS) SELECT ...

- <https://postgrespro.ru/docs/postgrespro/9.6/using-explain>
- подходит только для локальной отладки

Получение плана

Модуль **auto_explain**

- <https://postgrespro.ru/docs/postgresql/9.6/auto-explain>
- анализирует все запросы **дольше XXXms**
- фиксирует для них планы выполнения
- пишет все это в лог сервера

Получение плана

Модуль `auto_explain`

```
2017-09-05 17:06:25.638 MSK [57829:113/3890139] [profiles] 10.76.101.19(58818) sbis3mon : sbis3mon-ppc1.unix.tensor.ru [14899] LOG: duration: 11.007 ms plan:
Query Text: SELECT
    extract(epoch FROM now())::integer ts
    , sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_numscans(oid) END) seq
    , sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_tuples_returned(oid) END) seq_tup_read
    , sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_tuples_hot_updated(oid) END) hot
    , sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_live_tuples(oid) END) live
    , sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_dead_tuples(oid) END) dead
    , sum(CASE WHEN relkind = 'i' THEN pg_stat_get_numscans(oid) END) idx
    , sum(CASE WHEN relkind = 'i' THEN pg_stat_get_tuples_fetched(oid) END) idx_tup_fetch
FROM
    pg_class
WHERE
    relkind IN ('r', 't', 'm', 'i');

Aggregate  (cost=58.77..58.79 rows=1 width=228) (actual time=10.998..10.998 rows=1 loops=1)
  Buffers: shared hit=24
-> Seq Scan on pg_class  (cost=0.00..32.73 rows=443 width=5) (actual time=0.009..0.217 rows=443 loops=1)
   Filter: (relkind = ANY ('{r,t,m,i}':"char"[]))
   Rows Removed by Filter: 139
   Buffers: shared hit=24
```

Получение плана

Модуль `auto_explain`

```
2017-09-05 17:06:25.638 MSK [57829:113/3890139] [profiles] 10.76.101.19(58818) sbis3mon : sbis3mon-ppc1.unix.tensor.ru [14899] LOG: duration: 11.007 ms plan:
Query Text: SELECT
  extract(epoch FROM now())::integer ts
, sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_numscans(oid) END) seq
, sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_tuples_returned(oid) END) seq_tup_read
, sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_tuples_hot_updated(oid) END) hot
, sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_live_tuples(oid) END) live
, sum(CASE WHEN relkind IN ('r', 't', 'm') THEN pg_stat_get_dead_tuples(oid) END) dead
, sum(CASE WHEN relkind = 'i' THEN pg_stat_get_numscans(oid) END) idx
, sum(CASE WHEN relkind = 'i' THEN pg_stat_get_tuples_fetched(oid) END) idx_tup_fetch
FROM
  pg_class
WHERE
  relkind IN ('r', 't', 'm', 'i');

Aggregate (cost=58.77..58.79 rows=1 width=228) (actual time=10.998..10.998 rows=1 loops=1)
  Buffers: shared hit=24
-> Seq Scan on pg_class (cost=0.00..32.73 rows=443 width=5) (actual time=0.009..0.217 rows=443 loops=1)
  Filter: (relkind = ANY ('{r,t,m,i}':"char"[]))
  Rows Removed by Filter: 139
  Buffers: shared hit=24
```



Получение плана

Логи и план текстом – **ненаглядно**:

- в узле – **сумма по ресурсам** поддеревя
- время необходимо **умножать на loops**
- ... так кто же «самое слабое звено»?

Получение плана

Логи и план текстом – **ненаглядно**:

○ в узле – **сумма по ресурсам** поддеревя

○ время необходимо **умножать на loops**

○ ... так кто же «самое слабое звено»?

«Понимание плана – это искусство, и чтобы овладеть им, нужен определённый опыт, ...»

Получение плана

Логи и план текстом – **ненаглядно**:

- в узле – **сумма по ресурсам** поддеревя
- время необходимо **умножать на loops**
- ... так кто же «самое слабое звено»?

Нужна **хорошая** визуализация!



Визуализация плана

Визуализация плана

explain.depesz.com

#	exclusive	inclusive	rows x	rows	loops	node
1.	10.781	10.998	↑ 1.0	1	1	→ Aggregate (cost=58.77..58.79 rows=1 width=228) (actual time=10.998..10.998 rows=1 loops=1) Buffers: shared hit=24
2.	0.217	0.217	↑ 1.0	443	1	★ → Seq Scan on pg_class (cost=0.00..32.73 rows=443 width=5) (actual time=0.009..0.217 rows=443 loops=1) Filter: (relkind = ANY ('{r,t,m,i}'::"char"[])) Rows Removed by Filter: 139 Buffers: shared hit=24

Визуализация плана

explain.depesz.com – pro

- «собственное» время каждого узла
- отклонение от статистики по rows
- количество повторов каждого узла
- архив планов (обмен ссылками)

Визуализация плана

explain.depesz.com – contra

- требует copy&paste планов из лога
- нет анализа ресурсов (buffers)
- код на Perl, нет развития
- ошибки анализа CTE/InitPlan :(

Визуализация плана

explain.depesz.com – ошибки анализа CTE

#	exclusive	inclusive	rows x	rows	loops	node
1.	6.406	? 6.406	↓ 1.0	12,213	1	→ <u>CTE Scan</u> on ci (cost=485.67..725.01 rows=11,967 width=197) (actual time=0.006..6.406 rows=12,213 loops=1) Buffers: shared hit=366
2.						CTE ci
3.	1.389	1.389	↓ 1.0	12,213	1	→ <u>Seq Scan</u> on pg_class (cost=0.00..485.67 rows=11,967 width=219) (actual time=0.004..1.389 rows=12,213 loops=1) Buffers: shared hit=366

Визуализация плана

explain.sbis.ru

- ура! мы пишем свое!
- Node.JS / Express / Twitter Bootstrap / D3.js
- прототип за 2 недели

Визуализация плана

explain.sbis.ru

- собственный парсер плана
- корректный анализ CTE Scan
- анализ распределения ресурсов (buffers)
- наглядность, подсветка синтаксиса

Визуализация плана

explain.sbis.ru – полный план

explain	диаграмма	план	оригинал	для ошибки	статистика	
#	node, ms	tree, ms	rows	ratio	node	sh.ht
	6.406	12 213			итоговые результаты	366
0	5.017	6.406	12 213	----	CTE Scan on cl (cost=485.67..725.01 rows=11967 width=197) (actual time=0.006..6.406 rows=12213 loops=1) Buffers: shared hit=366	
1					CTE cl	
2	1.389	1.389	12 213	----	-> Seq Scan on pg_class (cost=0.00..485.67 rows=11967 width=219) (actual time=0.004..1.389 rows=12213 loops=1) Buffers: shared hit=366	366

Визуализация плана

explain.sbis.ru – сокращенный план (шаблон)

explain	диаграмма	план	оригинал	для ошибки	статистика	
#	node, ms	tree, ms	rows	ratio	node	sh.ht
	6.406	12	213		итоговые результаты	366
0	5.017	6.406	12	213	---- CTE Scan on cl	
1					CTE cl	
2	1.389	1.389	12	213	---- -> Seq Scan on pg_class	366

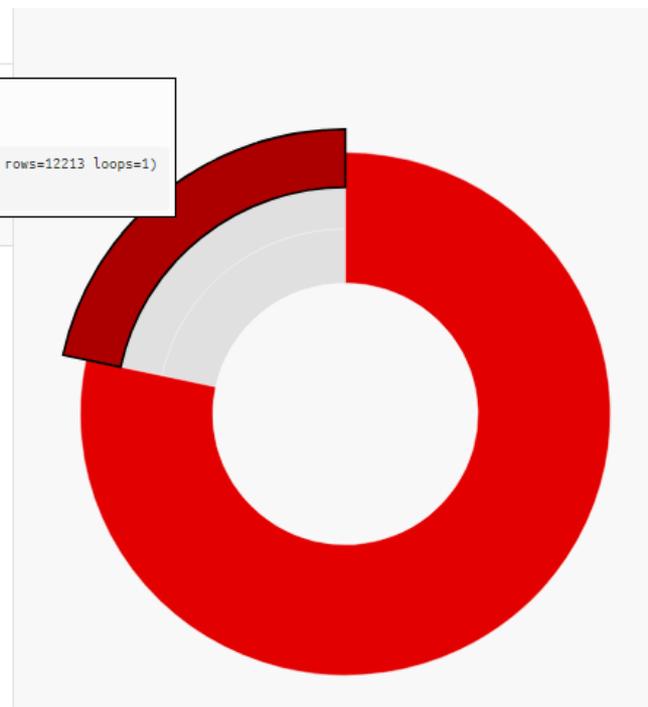
Визуализация плана

explain.sbis.ru – распределение времени

explain диаграмма план оригинал

#	node, ms	tree, ms	rows	ratio	
		6.406	12 213		итоговые результаты
0	5.017	6.406	12 213	----	CTE Scan on cl
1					CTE cl
2	1.389	1.389	12 213	---->	-> Seq Scan on pg_class

#2 **1.389**ms (21.7%), rows=12213, loops=1
Buffers узла (366): shared hit=366
Seq Scan on pg_class (cost=0.00..485.67 rows=11967 width=219) (actual time=0.004..1.389 rows=12213 loops=1)
Buffers: shared hit=366

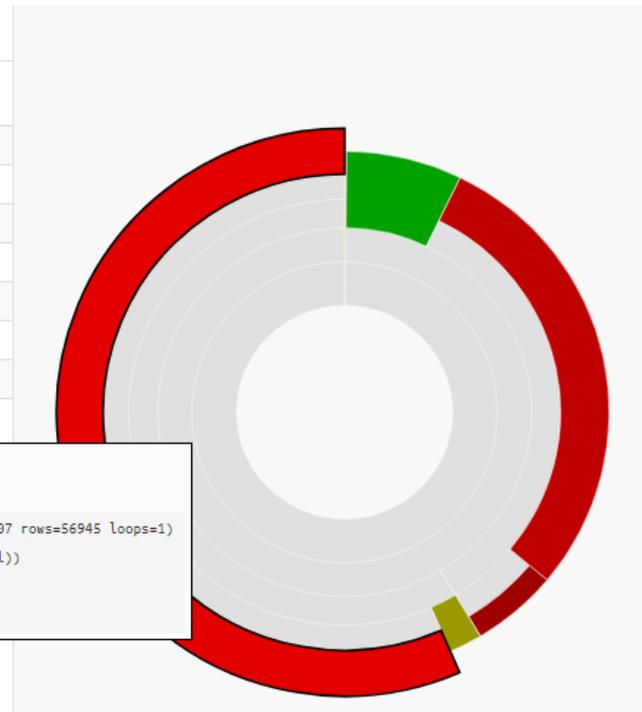


Визуализация плана

explain.sbis.ru – распределение времени

explain	диаграмма	план					
#	node, ms	tree, ms	rows	ratio	RRbF	👁	node
		651.914	1 000		3 723 969		итоговые результаты
0	0.086	651.914	1 000	----			Limit
1	0.691	651.828	1 000	1.7↑			-> Sort
2	46.168	651.137	1 945	1.2↓			-> Hash Join
3	188.476	223.124	324 690	----			-> Bitmap Heap Scan on "Пользователь" cli
4	34.648	34.648	325 094	----			-> Bitmap Index Scan on "iPolzovatel-VneshniyIdentifikator"
5	13.238	381.845	56 945	2.7↓			-> Hash
6	368.607	368.607	56 945	2.7↓	3 723 969	➔	-> Seq Scan on "ПользовательРасширение" cli_ext

```
#6 368.607ms (56.5%), rows=56945, loops=1
Buffers узла (35050): shared hit=35050
Seq Scan on "ПользовательРасширение" cli_ext (cost=0.00..110649.52 rows=21125 width=20) (actual time=2.132..368.607 rows=56945 loops=1)
Filter: (("РазмерДанныхФакт" > '536870912'::bigint) AND ((now() - "ВремяОценкиРазмераДанных") < '1 mon'::interval))
Rows Removed by Filter: 3723969
Buffers: shared hit=35050
```



Визуализация плана

explain.sbis.ru – «грабли»

- проблемы округления

$$0.001\text{ms} \times (\text{loops}=1000) = 0.95\text{ms} \text{ .. } 1.05\text{ms}$$

- распределение ресурсов CTE/InitPlan

+4 недели отладки :(

Визуализация плана

explain.sbis.ru – «грабли»

```
WITH c1 AS (  
    TABLE pg_class  
)  
(TABLE c1 LIMIT 1)  
    UNION ALL  
(TABLE c1 LIMIT 1 OFFSET 100);
```

Визуализация плана

explain.sbis.ru – «грабли»

explain	диаграмма	план	оригинал	для ошибки	статистика	
#	node, ms	tree, ms	rows	ratio	node	sh.ht
		0.066	2		итговые результаты	3
0	0.001	0.066	2	----	Append (cost=1077.37..1079.43 rows=2 width=195) (actual time=0.007..0.066 rows=2 loops=1) Buffers: shared hit=3	
1					CTE cl	
2	0.017	0.017	101	48.9↑	-> Seq Scan on pg_class (cost=0.00..1077.37 rows=4937 width=651) (actual time=0.005..0.017 rows=101 loops=1) Buffers: shared hit=3	3
3	0.000	0.006	1	----	-> Limit (cost=0.00..0.02 rows=1 width=195) (actual time=0.006..0.006 rows=1 loops=1) Buffers: shared hit=1	
4	0.000	0.006	1	4 937.0↑	-> CTE Scan on cl (cost=0.00..98.74 rows=4937 width=195) (actual time=0.006..0.006 rows=1 loops=1) Buffers: shared hit=1	
5	0.008	0.059	1	----	-> Limit (cost=2.00..2.02 rows=1 width=195) (actual time=0.058..0.059 rows=1 loops=1) Buffers: shared hit=2	
6	0.040	0.051	101	48.9↑	-> CTE Scan on cl cl_1 (cost=0.00..98.74 rows=4937 width=195) (actual time=0.000..0.051 rows=101 loops=1) Buffers: shared hit=2	

Визуализация плана

explain.sbis.ru – дерево выполнения

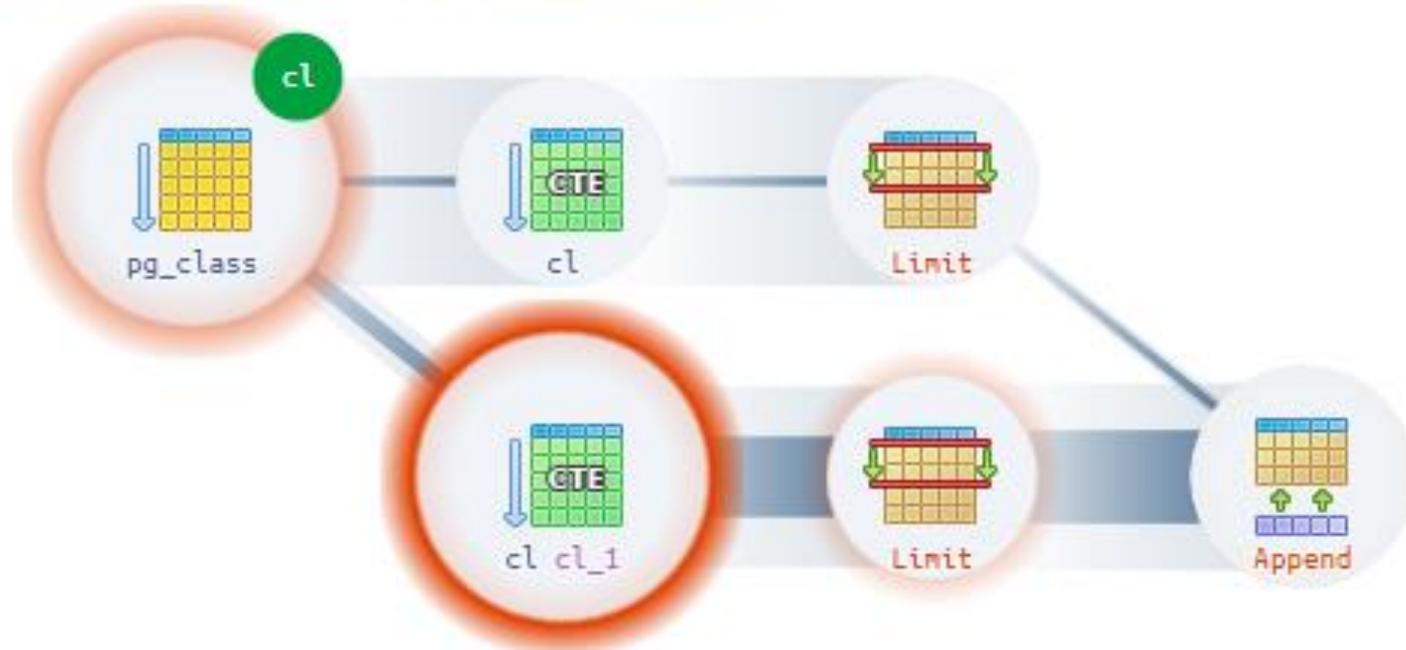
explain

диаграмма

план

оригинал

по затраченному времени | по использованию данных

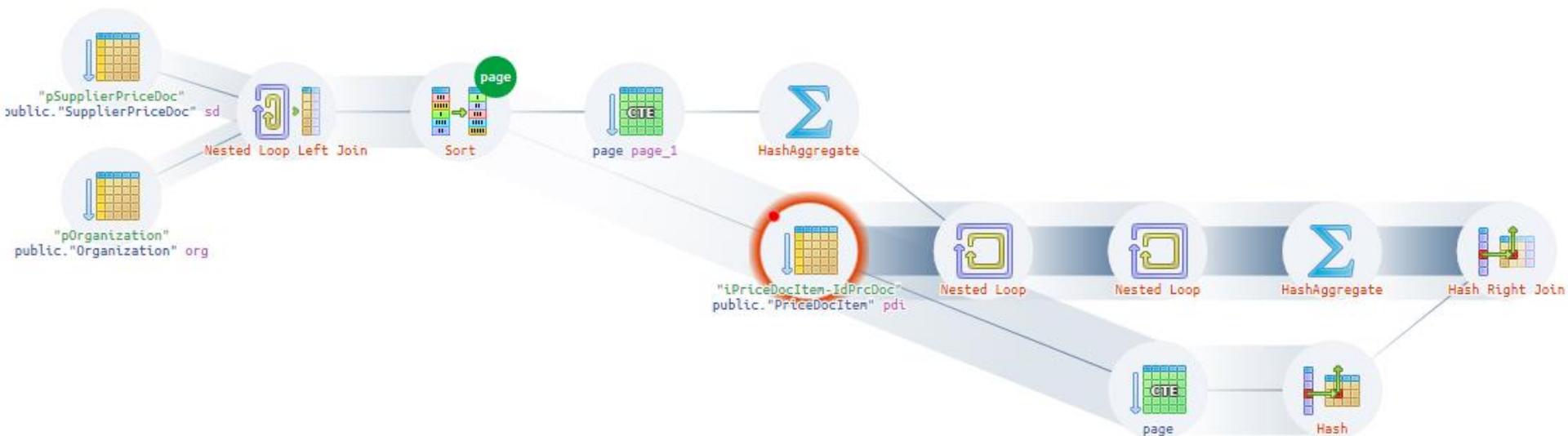


Визуализация плана

explain.sbis.ru – дерево выполнения

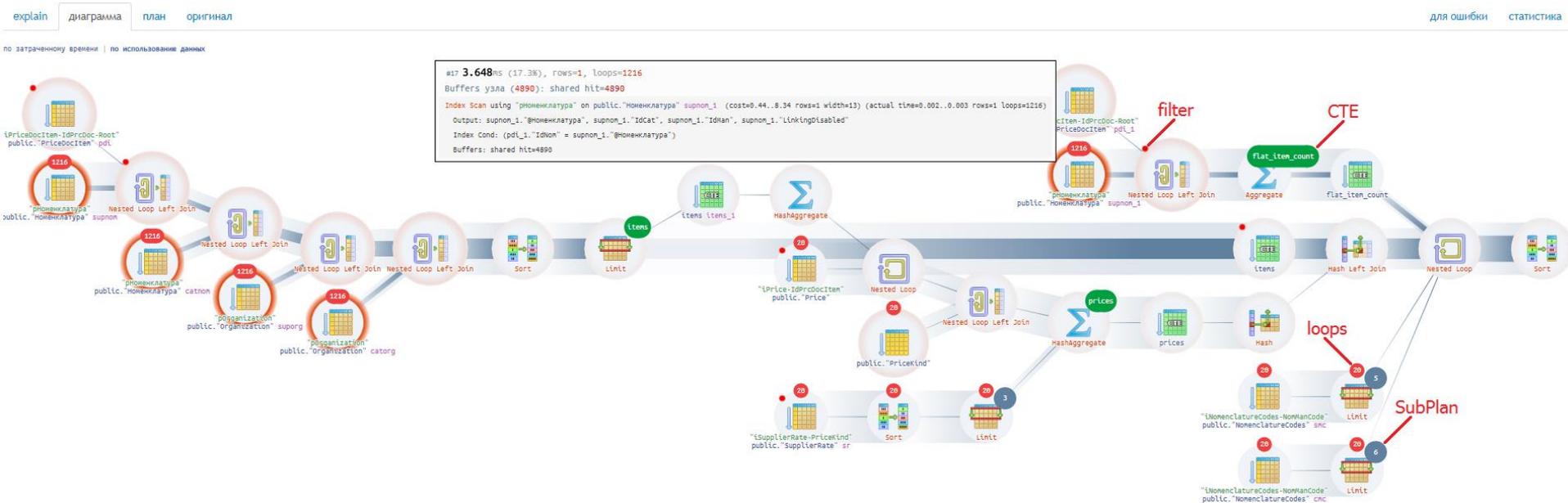
[explain](#) | [диаграмма](#) | [план](#) | [оригинал](#)

по затраченному времени | по использованию данных



Визуализация плана

explain.sbis.ru – дерево выполнения

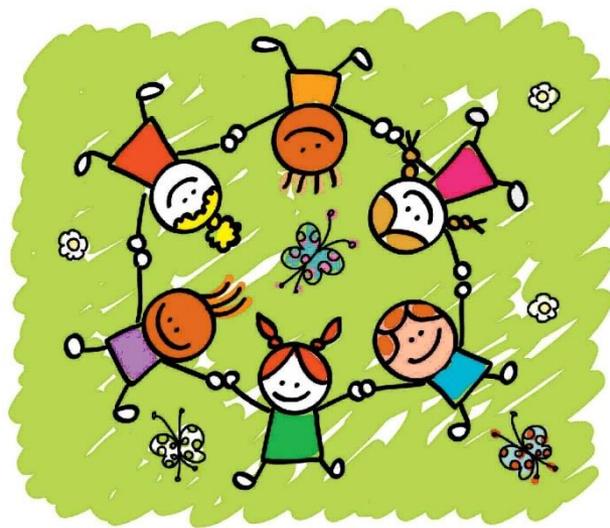


Визуализация плана

explain.sbis.ru

○ «Теперь, Нео, ты знаешь кунг-фу»





Консолидация логов

Консолидация логов

pg_stat_statements

- <https://postgrespro.ru/docs/postgrespro/9.6/pgstatstatements>
- разные QueryId для разных схем
- нет планов, только запросы
- нет «фактов», только агрегаты

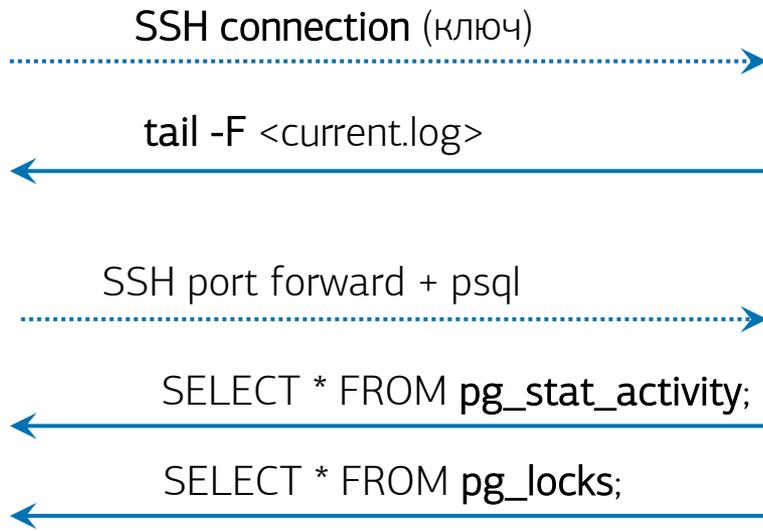
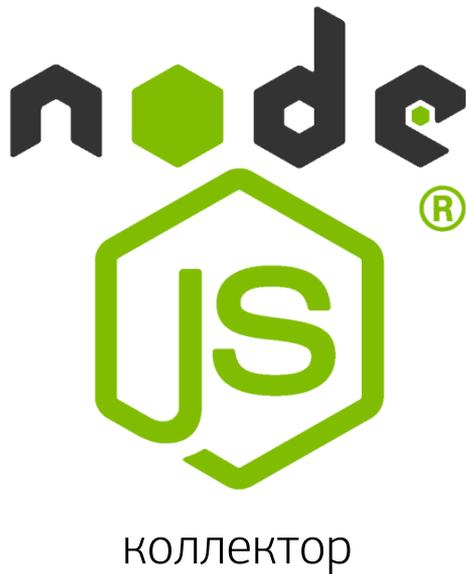
Консолидация логов

«Копипаста» – плохо

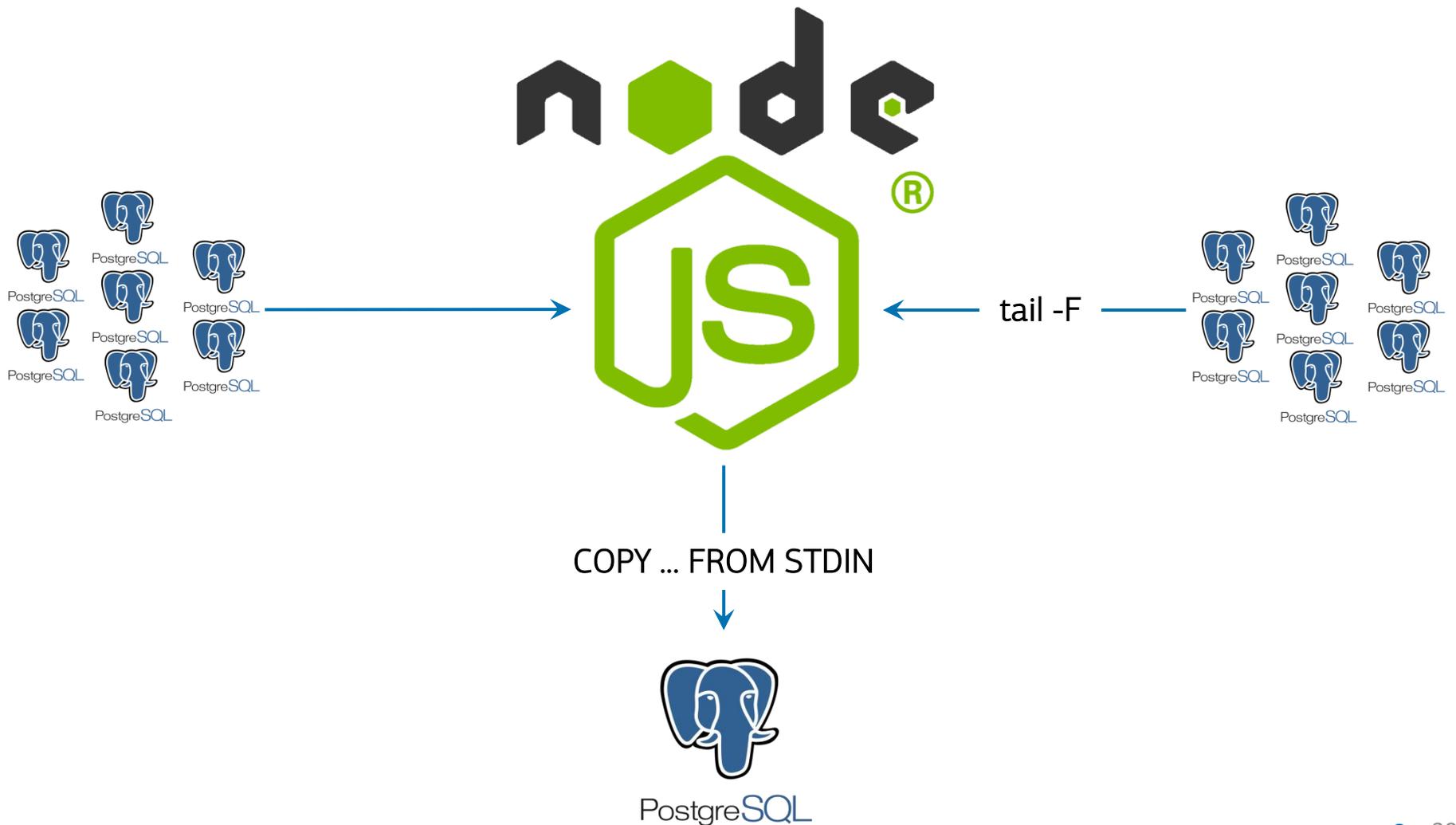
- 100+ серверов
- 1000+ разработчиков



Консолидация логов



Консолидация логов



Консолидация логов

100+ серверов, 50Kqps, 100-150GB/день

○ секционирование по дням

○ очень-очень быстрый «поточковый» COPY

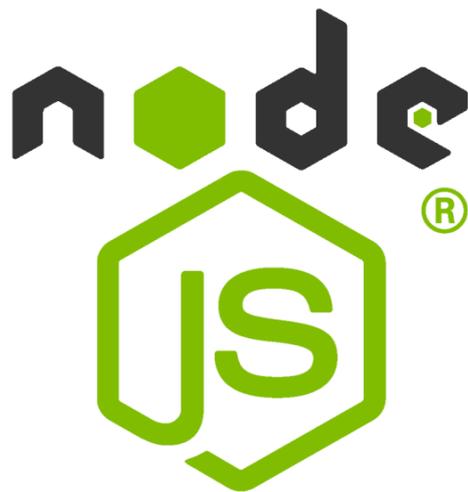
○ отказались от триггеров (почти)

Консолидация логов

Отказались от триггеров

- нет ссылочной целостности (нет FK)
- агрегация и хэширование на коллекторе
- каждой таблице – «свой» COPY-поток

Консолидация логов



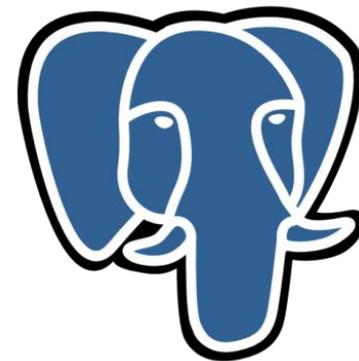
коллектор

COPY `plan` FROM STDIN

COPY `query` FROM STDIN

COPY `error` FROM STDIN

COPY `planagg` FROM STDIN



PostgreSQL

Консолидация логов

«Потоковый» COPY

○ всегда открыт COPY-канал/пул на таблицу

○ «переоткрывается» раз в XXXms

○ отправляем в канал **сразу** при получении

никакой дополнительной буферизации, да-да

Консолидация логов

«Потоковый» COPY

- таблицы-словари

триггер BEFORE INSERT

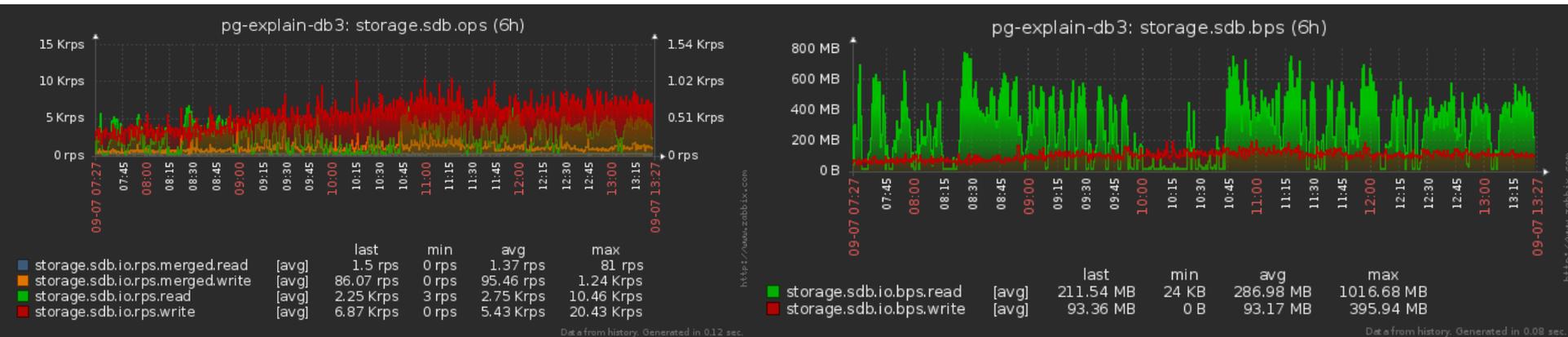
9.5+: INSERT ... ON CONFLICT DO NOTHING

Консолидация логов

«Потоковый» COPY

○ тогда: 4K write ops -> 1K write ops (в 4 раза!)

○ сейчас: 6K write ops, ~100MB/s, 10TB/3мес





Понимаем проблемы

Понимаем проблемы

100+ серверов, 50Kqps, 100-150GB/день

○ **МИЛЛИОНЫ** планов за сутки

Понимаем проблемы

100+ серверов, 50Kqps, 100-150GB/день

○ **МИЛЛИОНЫ** планов за сутки



Понимаем проблемы

100+ серверов, 50Kqps, 100-150GB/день

○ **МИЛЛИОНЫ** планов за сутки



Понимаем проблемы

100+ серверов, 50Kqps, 100-150GB/день

- кто? откуда этот запрос
- где? что за сервер, база
- как? в чем проблема в плане

Понимаем проблемы

«Хозяин» у каждого запроса

- SET application_name = ' <host>:<method>'

- страдаем от ограничения в 63 байта (name)

Понимаем проблемы

«Хозяин» у каждого запроса

- `log_line_prefix = ' %m [%p:%v] [%d] %r %a'`

- <https://postgrespro.ru/docs/postgrespro/9.6/runtime-config-logging>

```
2017-09-05 17:06:25.638 MSK [57829:113/3890139] [profiles] 10.76.101.19(58818) sbis3mon : sbis3mon-ppc1.unix.tensor.ru [14899] LOG:
```

Понимаем проблемы

«Хозяин» у каждого запроса

explain	диаграмма	план	для ошибки	статистика	контекст
Timestamp: 2017-09-08 14:52:41.174	UUID: ce6c9ccb-287c-4aff-83f4-7c209488d62c				
Host: csr-history2-db1.unix.tensor.ru	Application: csr-history-app1:10081/history : History.List				
Database: history1	User: ----				
PID: 16974	vXID: 23/33981073				
Client: 127.0.0.1:55858	tXID: ----				
LOG	WITH	utility_table AS(SELECT UNNEST (ARRAY[41183468, 40374310, 39867908, 39619463, 39476492, 38917139, 3909228	4 910.938		

Понимаем проблемы

Модель анализа

○ экземпляр PostgreSQL (хост:порт), день

○ **шаблон**, приложение/метод, узел плана

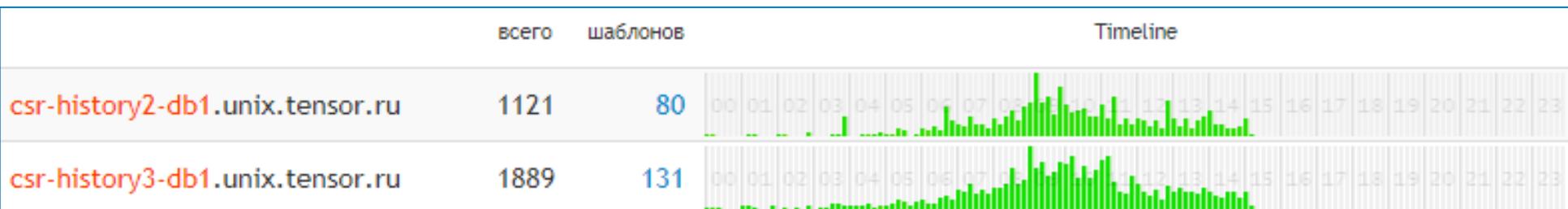
explain	диаграмма	план	оригинал	для ошибки	статистика	
#	node, ms	tree, ms	rows	ratio	node	sh.ht
	6.406	12	213		итоговые результаты	366
0	5.017	6.406	12 213	----	CTE Scan on cl	
1					CTE cl	
2	1.389	1.389	12 213	----	-> Seq Scan on pg_class	366

Понимаем проблемы

От планов – к шаблонам

○ меньше объектов для анализа

○ вычленение общих паттернов поведения



Понимаем проблемы

Разрезы анализа планов

- количество фактов по шаблону/методу
- суммарное и среднее время
- количество ресурсов (buffers hit/read)
- таймлайны

Понимаем проблемы

по шаблонам по приложениям по узлам плана по длительности

Шаблон / метод(ы)	app	Кол-во	sum, mc	avg, mc	buf:mem	buf:dsk	%	last	Timeline
c8e7ef0c-397a-99b0-0844-7639196a061b	History.List	523	1 394 521.292	2 666.389	5 010 459	316 849	5.9	14:52:41	
75967c0e-e00c-561f-fd91-a077f8b4b83c	History.HandleHistory	196	272 312.940	1 389.352	42 677	47	0.1	14:05:40	

```
Update on "ИсторияЭкземплярОбъекта" instance
-> Nested Loop
  -> Subquery Scan on instance_update
    -> LockRows
      InitPlan 1 (returns $0)
        -> Limit
          -> Index Scan using "ИсторияОбъект-Объект" on "ИсторияОбъект"
        -> Sort
          -> Index Scan using "ИсторияЭкземплярОбъекта-Экземпляр" on "ИсторияЭкземплярОбъекта"
      -> Index Scan using "ИсторияЭкземплярОбъекта" on "ИсторияЭкземплярОбъекта" instance
```

b160f01b-e4e1-7068-8780-262cf8986c4d	History.List	37	61 223.497	1 654.689	205 552	12 447	5.7	12:56:40	
a5910953-6a31-e236-bd0a-ba772d18f8b1	History.HandleHistory	43	53 989.315	1 255.565	5 545	175	3.1	13:53:49	

```
CTE Scan on next_event
CTE next_event
-> Result
CTE object_id_insert
-> Insert on "ИсторияОбъект" "ИсторияОбъект_1"
  -> Subquery Scan on "+SELECT+"
    -> Result
      InitPlan 2 (returns $1)
        -> Index Only Scan using "ИсторияОбъект-Объект" on "ИсторияОбъект"
CTE object_id_select
-> Limit
  -> Index Scan using "ИсторияОбъект-Объект" on "ИсторияОбъект" "ИсторияОбъект_2"
CTE object
-> Hash Full Join
  -> CTE Scan on object_id_insert
  -> Hash
    -> CTE Scan on object_id_select
CTE instance_insert_id
-> Insert on "ИсторияЭкземплярОбъекта" "ИсторияЭкземплярОбъекта_1"
  InitPlan 6 (returns $6)
    -> CTE Scan on next_event next_event_1
    -> CTE Scan on object
      SubPlan 7
        -> Limit
          -> Index Scan using "ИсторияЭкземплярОбъекта-Экземпляр" on "ИсторияЭкземплярОбъекта"
CTE instance_select_id
-> Limit
  InitPlan 9 (returns $10)
    -> CTE Scan on object object_1
    -> Index Scan using "ИсторияЭкземплярОбъекта-Экземпляр" on "ИсторияЭкземплярОбъекта" "ИсторияЭкземплярОбъекта_2"
CTE instance
```

Понимаем проблемы

Метод / шаблон(ы)	ptr	Кол-во	sum, мс	avg, мс	buf:mem	buf:dsk	%	last	Timeline
History.List	75	878	2 067 254.483	2 354.504	7 803 874	423 873	5.2	14:52:41	
History.HandleHistory	4	242	331 096.412	1 368.167	48 654	255	0.5	14:05:40	
75967c0e-e00c-561f-fd91-a077fab4b83c		196	272 312.940	1 389.352	42 677	47	0.1	14:05:40	
a5910953-6a31-e236-bd0a-ba772d18f8b1		43	53 989.315	1 255.565	5 545	175	3.1	13:53:49	
e8c26590-ad32-517c-3d58-4c165e4ef6d7		2	2 588.827	1 294.414	270	12	4.3	09:33:22	
3c1b7e84-63cd-b651-848e-17a6be43de7d			2 205.330	2 205.330	162	21	11.5	09:02:04	
5c168372-4fdf-6c25-2439-b9bce59758fc			1 251.113	1 251.113	128	38	22.9	17:12:01	
fd3fda54-8870-220a-21f1-68bbdfef1358			1 227.091	1 227.091	17	33	66.0	17:12:01	
History.HistoryCommonM		2	2 075.775	1 037.888	73	6	7.6	08:54:26	

Понимаем проблемы

Разрезы анализа узлов по объектам БД

- { Seq | Index [Only] | Bitmap (Index|Heap) } Scan
- количество фактов/шаблонов по узлу
- loops, rows, RRbF (суммарно и в среднем)
- время и объем ресурсов

Понимаем проблемы

по шаблонам		по приложениям		по узлам плана		по длительности											
Тип узла	таблица	индекс	ptr	кол-во	loops	loops, avg	rows	rows, avg	RRbF	RRbF, avg	last	Timeline					
Index Scan	ИсторияЭкземплярОбъекта	iИсторияЭкземплярОбъекта-ObjectChanges	72baee62-bf95-eeb3-23ff-2e4fd39a2982	11	11	1	965 844	87 884			12:26:32						
Index Scan	КонтекстИсполнения	iКонтекстИсполнения-id		57	21 180	25	21 174				14:52:41						
Bitmap Heap Scan	ИсторияДействие			64	809	21 131	26	21 131	1		14:52:41						
Bitmap Index Scan	ИсторияДействие	riИсторияДействие		64	809	21 131	26	21 131	1		14:52:41						
Index Only Scan	ИсторияЭкземплярОбъекта	iИсторияЭкземплярОбъекта-InstanceInfo		49	780	21 018	26	21 018	1		14:52:41						
Index Scan	ИсторияЭкземплярОбъекта	iIstoriyaEkzemplyarObekta-Ekzemplyar		49	1 055	66 531	63	8 401			14:52:41						
Seq Scan	ИсторияОбъект			12	26	26	1	7 029	270	3 505	134	14:17:44					
Index Scan	ИсторияОбъект	iИсторияОбъект-Объект		60	1 037	1 037	1	1 037	1		14:52:41						
Index Scan	ИсторияСобытие.\$2017-09-08	iИсторияСобытие.\$2017-09-08-last_changed		8	18	27	1	761	28		14:41:43						
	a0142691-aea6-ef2d-5af2-72ac0710cdb0			7	7	1	2 291	327			17:07:05						
	49719919-4b95-f044-b801-490e7cd2c2c5			3	3	1	82	27			15:54:02						
	2e49565b-233f-5fd6-4638-e6aa7a87980f			4	8	2	6				15:22:04						

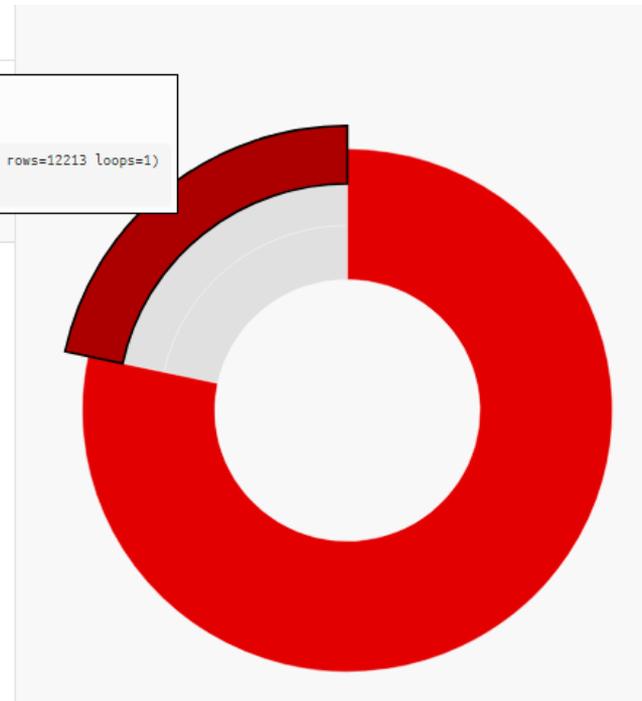
Понимаем проблемы

Разрезы анализа динамических узлов

○ { CTE | Function | Subquery | Values | WorkTable } Scan

#	node, ms	tree, ms	rows	ratio	
		6.406	12 213		итоговые результаты
0	5.017	6.406	12 213	----	CTE Scan on cl
1					CTE cl
2	1.389	1.389	12 213	---->	-> Seq Scan on pg_class

#2 1.389ms (21.7%), rows=12213, loops=1
Buffers узла (366): shared hit=366
Seq Scan on pg_class (cost=0.00..485.67 rows=11967 width=219) (actual time=0.004..1.389 rows=12213 loops=1)
Buffers: shared hit=366



Понимаем проблемы

Подсказываем решения



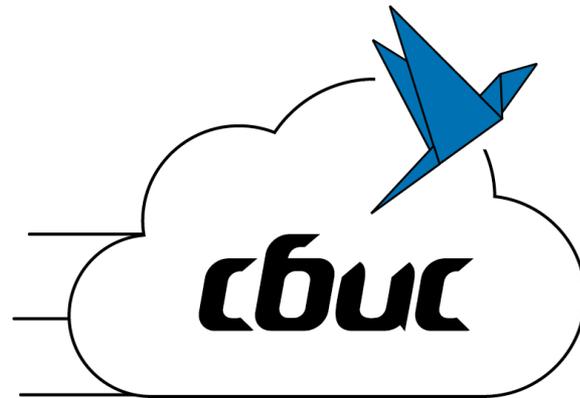
```
-> Limit
  -> Sort
    -> Index Only Scan using "iДокумент-ТипДокументаЛицо1" on "Документ" "Документ_1"
```

Рекомендации:

- **создайте индекс**, включающий поля, используемые для сортировки



... и устраняем причины



Спасибо за внимание!

Боровиков Кирилл

kilor@tensor.ru

sbis.ru / tensor.ru