SETUP

Нетрадиционный (not gay) PostgreSQL: хранение бинарных данных в БД

Хорошие, плохие и ужасные стороны и борьба за эффективность

Александр Чистяков,
главный инженер Git in Sky,
2014

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Давайте познакомимся

- Меня зовут Саша
- Я работаю главным инженером в Git in Sky
- Когда программа заболевает, ее приносят ко мне
- Я пытаюсь поставить диагноз и назначить лечение до того, как программа умрет
- Чаще всего это удается

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Теперь ваша очередь

- Как ваше здоровье?
- Принимаете PostgreSQL?
- Злоупотребляете веб-разработкой?
- Пишете на PHP?
- Какие другие вредные привычки имеете?

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Больной, задержите дыхание!

- http://slideshare.net/profyclub_ru/08-6
- ^ карточка пациента (передана из детской поликлиники)
- Жалобы на головную боль у специалистов отдела эксплуатации
- То есть, у нас, а мы не любим, когда болит голова

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Итак, что нам известно?

- Заказчик — конструктор сайтов, http://www.setup.ru
- Пользовательский контент хранится в базе данных (угадайте, какая СУБД?)
- Для работы с большими файлами используется large objects API
- Приложение на Perl под Apache + mod_perl

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

История болезни

- Количество файлов: было 6 млн, стало 207 млн (85, если не учитывать версионность)
- Размер индексов: был 2Gb, стал десятки Gb
- Скорость синхронизации упала с 100 файлов/сек до примерно 30 файлов/сек
- Объем базы данных на дисках на момент начала лечения: 6Tb (сейчас уже 7Tb)

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Может, просто переписать это всё?

- Исходно у нас не было достаточно знаний о том, как работает текущая реализация
- Но было бизнес-требование обеспечить атомарную публикацию групп файлов
- Где транзакции — там и СУБД
- Или другое транзакционное хранилище
- Много вы знаете NoSQL хранилищ с поддержкой транзакций на несколько строк?

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Объекты предметной области (и таблицы)

- Таблица domains — имена доменов
- Таблица content — метаинформация о файле (время последнего изменения и путь)
- Таблица stat — сами бинарные данные и их sha-1 хэш для дедупликации
- Таблица deleted — признак того, что файл удален
- Все четыре таблицы связаны между собой

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Пользовательские сценарии

- Публикация и синхронизация файлов:
  - Сайт публикуется всегда на одну и ту же ноду
  - Кастомный синхронизатор медленно обновляет все остальные ноды
- Отдача статического контента:
  - Отдаем а) последнюю, б) неудаленную версию

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

На что жалуетесь, больной?

- Файлы отдаются недостаточно быстро (50 миллисекунд в лучшем случае)
- Публикация, а, особенно, синхронизация работают медленно
- Железо справляется недостаточно хорошо, судя по графикам основных параметров
- Да, все это происходит в Hetzner

PGDAY'14 RUSSIA

SETUP

Солидный хостинг для солидных господ

- Было: RAID0 из 2*3Tb SATA, 16Gb RAM, 128Gb SSD — для pg_temp и nginx, сортировка в PostgreSQL и буферизация в nginx работают быстро
- Стало: RAID10 из 4*4Tb SATA, 48Gb RAM, SSD не дают ни за какие деньги, хотя место в корпусе физически еще есть (добро пожаловать в Hetzner!)
- С точки зрения производительности стало хуже, чем было, но надо как-то жить с этим, на 6Tb база уже не поместится

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Как поставить диагноз?

- Как это делается обычно:
  - slow queries log
  - pgFouine или pgBadger, генерация отчетов раз в период
  - Анализ отчетов, анализ планов долгих запросов

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Если подумать, все еще проще

- pgFouine и pgBadger все равно не справятся с логом — запросов слишком много
- Количество разных запросов ограничено, так как система очень проста, два самых популярных при отдаче и синхронизации - “найти неудаленный файл” и “найти, что синхронизировать”
- Используют views, тормозят, их и нужно лечить

PGDAY'14 RUSSIA

План запроса отдачи файла

```
                                   QUERY PLAN
---------------------------------------------------------------------------------------------------------------
 Limit  (cost=137.50..137.51 rows=1 width=102)
   ->  Sort  (cost=137.50..137.51 rows=1 width=102)
         Sort Key: ((list_latest.sha1)::character(40))
         ->  Subquery Scan on list_latest  (cost=0.00..137.49 rows=1 width=102)
               ->  Nested Loop Anti Join  (cost=0.00..137.47 rows=1 width=164)
                     ->  Nested Loop  (cost=0.00..126.33 rows=1 width=164)
                           ->  Nested Loop  (cost=0.00..110.03 rows=1 width=105)
                                 ->  Index Scan using i_domains_name on domains d  (cost=0.00..8.61 rows=1 width=37)
                                       Index Cond: ((name)::text = 'meitanspb.ru'::text)
                                 ->  Index Scan using i_stat_domain_name on stat s  (cost=0.00..101.41 rows=1 width=92)
                                       Index Cond: ((domain = d.id) AND ((name)::text = '/katalog/kapsulnaya-kosmetika/index.html'::text))
                                       Filter: (d.ptime = ptime)
                           ->  Index Scan using content_pkey on content c  (cost=0.00..16.29 rows=1 width=75)
                                 Index Cond: (id = s.content)
                     ->  Index Only Scan using deleted_pkey on deleted e  (cost=0.00..5.44 rows=1 width=8)
                           Index Cond: (id = s.id)
(16 rows)
```

- Не так плохо, как обычно бывает, но и не так хорошо, как может быть

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Как будем лечить?

- Традиционный способ — материализация view
- У нас PostgreSQL 9.2, там нет materialized views
- Но их можно эмулировать с помощью триггеров!
- Глава из книги “Enterprise Rails” с примером находится на второй странице в Google по запросу “postgresql materialized views triggers”
- Так и сделаем!

PGDAY'14 RUSSIA

SETUP

Применять по рецепту врача

- “Поверх” обычного view делается таблица с такими же полями, как у view
- Она работает как кэш — записи в ней заводятся по запросу
- Сначала ищем в ней, потом в исходном view, если не нашлось в ней (и не забываем найденное класть в нее)
- Записи в таблице-кэше инвалидируются триггерами на всех таблицах-участниках исходного view
- Вместо инвалидации можно делать сразу апдейт кэша

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Нужен хороший термометр

- pgFouine и pgBadger не подходят — ресурсоемки, медленны, долго ждать результат, в лог медленных запросов могут попасть не все нужные запросы
- Расширение pg_stat_statements
- Позволяет накапливать и анализировать статистику в реальном времени

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Как пользоваться pg_stat_statements?

- CREATE EXTENSION pg_stat_statements;
- SELECT
  (total_time / 1000 / 60) as total_minutes,
  (total_time/calls) as average_time,
  calls, query
  FROM pg_stat_statements
  ORDER BY total_minutes/average_time desc;

PGDAY'14 RUSSIA

SETUP

Что будет видно на приборах?

```
    total_minutes     |     average_time     |  calls   |
                       query
----------------------+----------------------+----------+-----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------
     8441.05738103645 |     15.5697532365578 | 32528675 | SELECT * FROM file_fetch($1,$2)
     2890.76438703287 |     5.23092619103136 | 33157773 | SELECT          llp.chunk
                                                                                                                                                          +
                      |                      |          |          ,llp.largeobj
                                                                                                                                                          +
                      |                      |          |          ,llp.size::BIGINT
                                                                                                                                                          +
                      |                      |          |          ,llp.revision
                                                                                                                                                          +
                      |                      |          |          ,llp.sha1::CHAR(40)
                                                                                                                                                          +
                      |                      |          |          ,(EXTRACT(EPOCH FROM llp.mtime))::BIGINT AS mtime
                                                                                                                                                          +
                      |                      |          |      FROM list_latest_proper llp
                                                                                                                                                          +
                      |                      |          |      WHERE llp.domain = domname
                                                                                                                                                          +
                      |                      |          |        AND llp.name   = filename
                                                                                                                                                          +
                      |                      |          |      LIMIT 1
     971.570193190344 |     4.66234262162769 | 12503202 | SELECT           s.id AS id from stat s, domains d where s.domain=d.id and d.name=domname and h_int(s.name)=h_
int(filename) and s.name=filename and s.ptime = d.ptime
     724.934005300585 |     6.51757204753482 |  6673657 | SELECT * from file_stat($1,$2,$3)
     608.306674250064 |     13.3626863233699 |  2731367 | SELECT date_part('epoch'::text, s.rtime)::bigint AS mtime
                                                                                                                                                          +
                      |                      |          |          ,NULL::BIGINT AS size
                                                                                                                                                          +
                      |                      |          |          ,encode(c.sha1, 'hex'::text)::CHAR(40) AS sha1
                                                                                                                                                          +
                      |                      |          |          ,date_part('epoch'::text, s.ptime) AS revision
                                                                                                                                                          +
                      |                      |          |           ,CASE
                                                                                                                                                          +
                      |                      |          |             WHEN s.ptime = d.ptime THEN true
                                                                                                                                                          +
                      |                      |          |             ELSE false
u  5:binstore  6:su-                                                                                                                  "fe7.setup.ru" 09:25 12-Apr-14
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Стало ли нам хорошо?

- Принцип работает — таблица кэширует
- Процент непопаданий слишком высок — 30-40%
- После суток ожидания он не изменился — где-то должна быть ошибка!

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Знание — сила!

- Шаблон “посмотреть в кэширующей таблице, а потом во view” не очень хорошо работает для веба
- Если запрашиваемого контента нет вообще (404), то кэш для таких запросов работать не будет — мы нагружаем view лишней работой
- Надо быстро определять, есть ли у нас вообще запрашиваемый файл

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Стало ли лучше теперь?

- Ночью — 15 мс на SQL в среднем
- Днем — 40-50 мс на SQL в среднем
- Железо, несмотря на улучшения, все равно работает на грани возможного
- Проблема еще в том, что логику приложения я меняю ночью, а результат надо смотреть днем

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Термометра мало, нужна МРТ-установка

- Нас интересует время отдачи контента, а не просто время SQL-запроса
- Его нужно измерять на эппсервере
- Варианты: Zabbix, Graphite/StatsD
- http://goo.gl/x6lf1S
- ^ Ansible playbook для установки Graphite и StatsD

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Zabbix убивает!

- Никогда не используйте Zabbix!
- Нет времени объяснять, просто не делайте этого

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Как устроен Graphite/StatsD стек

- Dashboard (сначала я пользовался стандартным от Graphite)
- Веб-сервис отдачи графиков (на Python/Django)
- Коллектор с RRD-like хранилищем, которое называется Whisper (тоже на Python)
- Агрегатор/препроцессор с UDP-интерфейсом (собственно, StatsD)

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Имплементации StatsD-сервера

- Исходно — Node.JS
- Есть на C, Perl, Ruby, Python, Go, ...
- Сначала я взял Python

```
root@fe5 ~ strace -p 14628
Process 14628 attached - interrupt to quit
futex(0x1075000, FUTEX_WAIT_PRIVATE, 0, NULL) = 0
futex(0x1075000, FUTEX_WAIT_PRIVATE, 0, NULL^C <unfinished ...>
```

- Потом был Perl, сейчас я перехожу на Go (меньше памяти, быстрее)

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Вернемся к нашему пациенту

- 70-120 миллисекунд в среднем и 150+ миллисекунд — upper 90%
- Как быть?
- Мы же про СУБД говорим, давайте построим индексы!
- Как мы это будем делать?
- “Один запрос — один индекс на таблицу”

PGDAY’14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Скальпель, спирт, огурец!

- Для самого частого запроса был построен индекс на все три столбца, на которые наложены условия в WHERE
- После этого пациент чуть не умер! :)
- Вскрытие показало, что размер нового индекса — 18Gb, и он просто не помещается в память в нужном для нормальной работы объеме

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Дефибриллятор!

- Одно из полей в индексе - varchar
- Превращаем varchar в int:
- http://stackoverflow.com/a/9812029/601572
- Да, у меня однажды был клиент, который не любил хранимые процедуры и триггеры
- Его бизнес успешно умер, отчасти, именно поэтому

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Принимать по рецепту врача

- Вот что было по ссылке:
- create function h_int(text) returns int as $$

  select ('x'||substr(md5($1),1,8))::bit(32)::int;

  $$ language sql;

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Читаем план запроса

- SET enable_bitmapscan=false; <= старые добрые nested loops

```
SELECT something
FROM stat s JOIN domains d ON d.id = s.domain JOIN content c ON c.id = s.content
LEFT JOIN deleted e ON e.id = s.id
WHERE d.name   = domname
 AND h_int(s.name) = h_int(filename)     <= работает новый маленький индекс
 AND s.name     = filename
 AND date_part('epoch'::text, s.ptime) = filerev
```

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Здоров ли пациент?

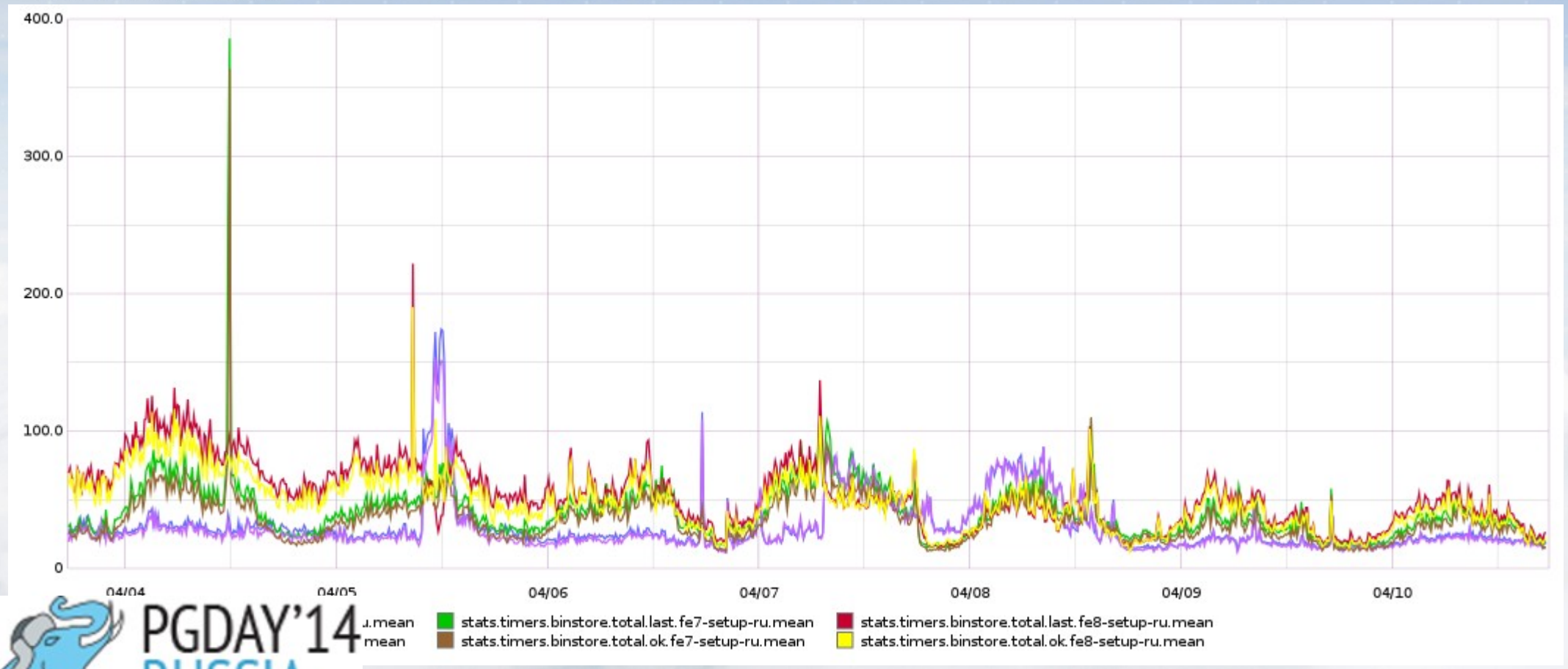
- Размер индекса: 18Gb => 8Gb
- Среднее время выполнения SQL запроса: стало 20-25 мс
- Среднее время отдачи контента — 40-50 мс
- Upper 90% - стало 100 мс

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Температура больного (Graphite)

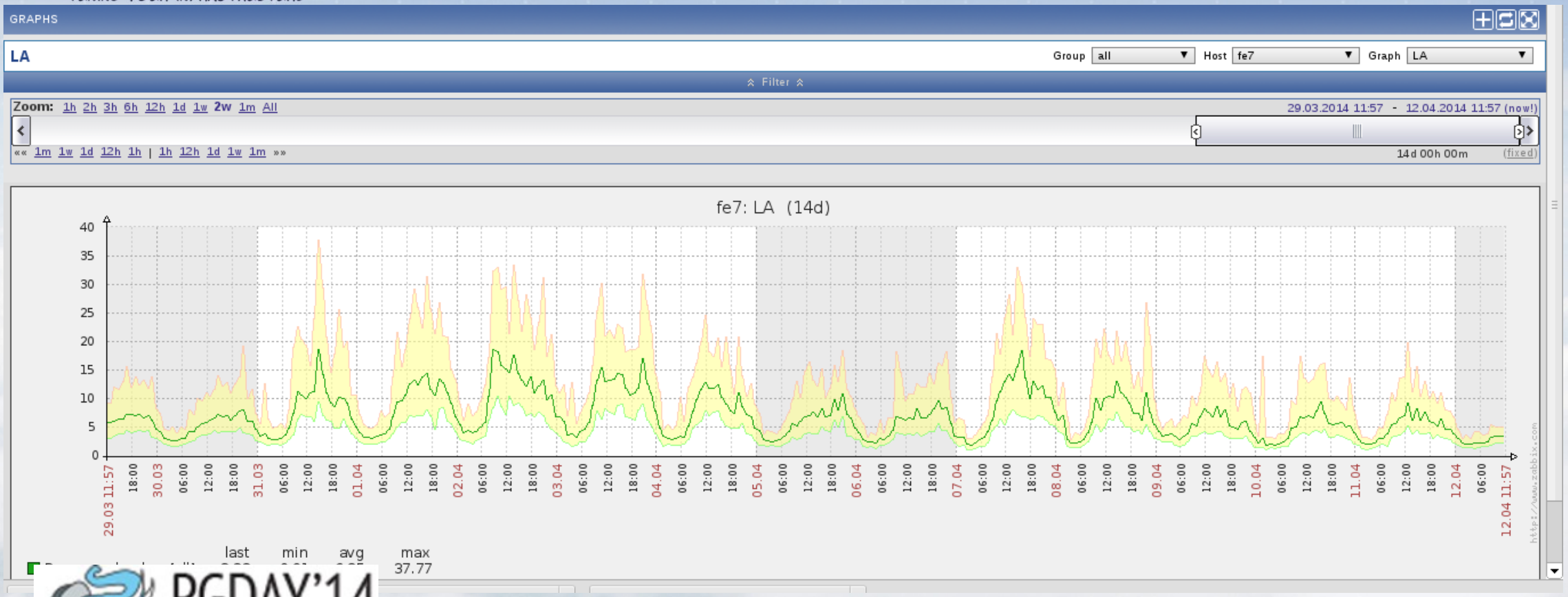


PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Температура больного (Zabbix)



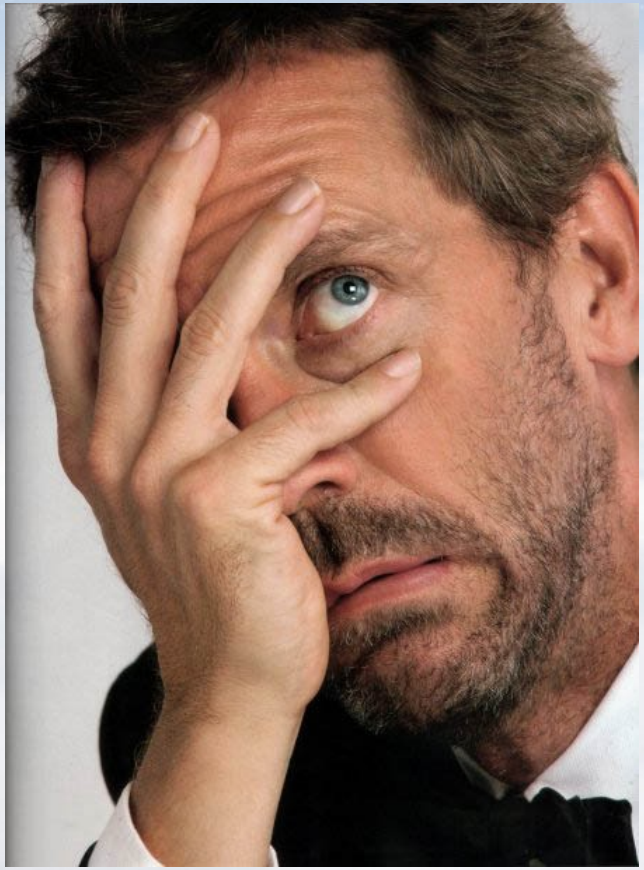
PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Переходим к ужасному

- PL/pgSQL — это ужасно, особенно, в моем исполнении
- Процедура отдачи файла занимала две трети экрана, после всех оптимизаций стала занимать два экрана
- Естественно, я допустил в ней ошибку



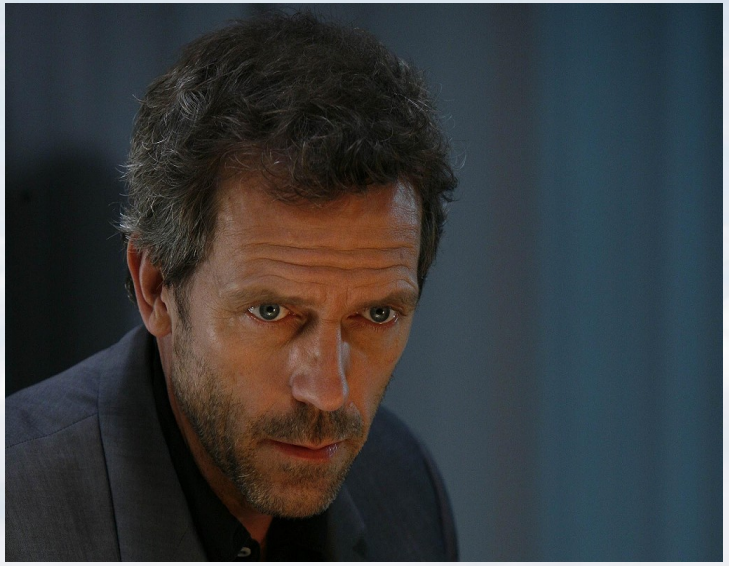
PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Разрезать и зашить заново

- При проверке существования файла я получал id файла и решил ускорить бы обращение ко view (по PK)
- Оказалось, мне возвращался целый массив ключей
- Поэтому после оптимизации клиенты стали видеть старый контент
- Исправил это — убрал нагрузку с машин!



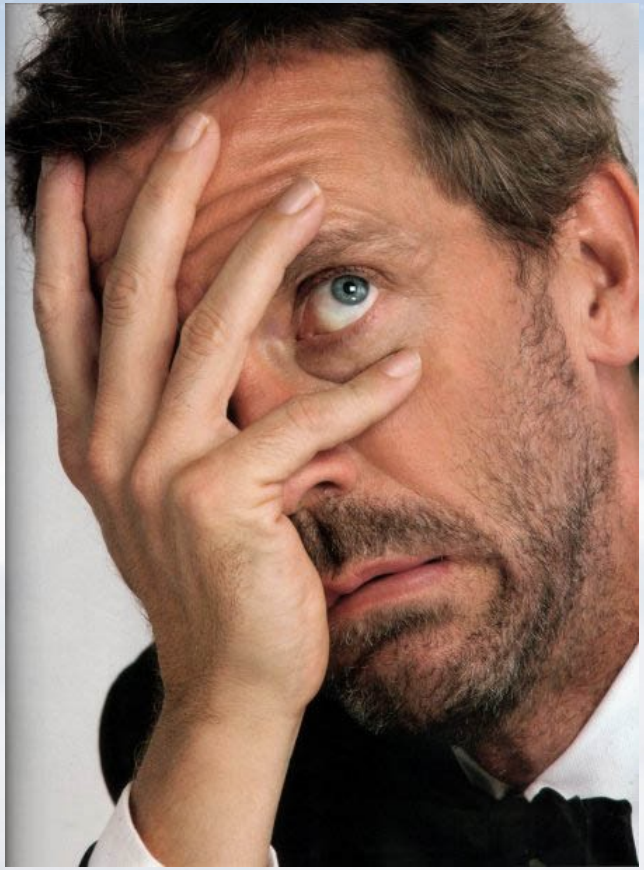
PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Еще немного про PL/pgSQL

- Я так и не смог найти, как сконструировать программно множество из нуля строк (всегда получалось из одной почему-то), поэтому для получения такого множества просто завел специальную пустую таблицу с нужным списком полей
- Кстати, из PL/Perl я мог бы слать метрики в StatsD, но Perl я тоже что-то не очень…



PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Новые болезни

- Как я уже говорил, база растет, а максимум доступного нам места — чуть меньше 8Tb, поэтому нужен был выход
- Теперь я уже знал логику приложения, и обнаружил, что large objects никогда не переписываются, а всегда записываются заново с новым id
- Это же object storage!

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Как не надо читать большую таблицу

- Осталось просто переложить большие объекты в другое место, но...

```
binstore=# explain SELECT * FROM list_all_proper WHERE largeobj IS NOT NULL AND deleted IS false order by id offset 100 LIMIT 100;
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Limit  (cost=153.55..307.09 rows=100 width=141)
   ->  Index Scan using list_all_dirty_id on list_all_dirty  (cost=0.00..2623962.47 rows=1708897 width=141)
         Filter: ((dirty IS FALSE) AND (largeobj IS NOT NULL) AND (deleted IS FALSE))
(3 rows)

binstore=# explain SELECT * FROM list_all_proper WHERE largeobj IS NOT NULL AND deleted IS false order by id offset 7000000 LIMIT 100;
                                               QUERY PLAN
-------------------------------------------------------------------------------------------------
 Limit  (cost=911540.57..911540.57 rows=1 width=141)
   ->  Sort  (cost=907268.32..911540.57 rows=1708897 width=141)
         Sort Key: list_all_dirty.id
         ->  Seq Scan on list_all_dirty  (cost=0.00..607696.89 rows=1708897 width=141)
               Filter: ((dirty IS FALSE) AND (largeobj IS NOT NULL) AND (deleted IS FALSE))
(5 rows)

binstore=#
```

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Как надо читать большую таблицу

- MVCC хранилище не знает, делая index scan, какие из строк живы, поэтому очень болезненно относится к большим значениям OFFSET — лучше накладывайте условия на значения самого ключа, а не на порядковый номер записи

```
binstore=# explain SELECT * FROM list_all_proper WHERE largeobj IS NOT NULL AND deleted IS false AND id > 7000000 order by id LIMIT 100;
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Limit  (cost=0.00..154.82 rows=100 width=141)
   ->  Index Scan using list_all_dirty_id on list_all_dirty  (cost=0.00..2645695.44 rows=1708897 width=141)
         Index Cond: (id > 7000000)
         Filter: ((dirty IS FALSE) AND (largeobj IS NOT NULL) AND (deleted IS FALSE))
(4 rows)

binstore=#
```

PGDAY'14 RUSSIA

Git in Sky
TUNING YOUR INFRASTRUCTURE

SETUP

Выводы:

- Вылечить программу гораздо проще, чем человека (я рассказал об очень простых вещах, правда ведь?)
- PostgreSQL лучше, чем MySQL
- Чем лучше? Чем MySQL!

PGDAY'14 RUSSIA

SETUP

Спасибо за внимание!

Пожалуйста, ваши вопросы.

С вами был Александр Чистяков,
главный инженер Git in Sky

alex@gitinsky.com
http://gitinsky.com
http://meetup.com/DevOps-40