Слепые ощупывают слона

Александр Чистяков, главный инженер Git in Sky

16.07.2015

PGDay, Санкт-Петербург

Git in Sky
TUNING YOUR INFRASTRUCTURE

Давайте познакомимся

- Меня зовут Саша
- Я работаю в компании Git in Sky
- I have an elephant
- Вы, я так понимаю, временно нигде не работаете

Git in Sky
TUNING YOUR INFRASTRUCTURE

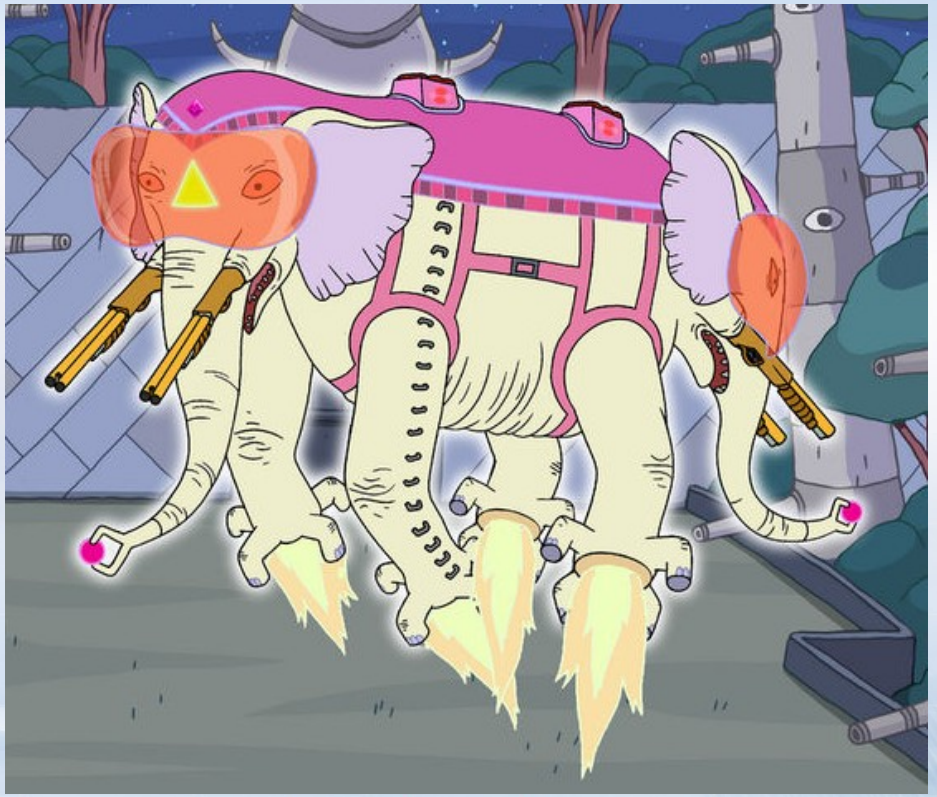
Что делать?

- Возьмем PostgreSQL
- Выдвинем какие-нибудь гипотезы
- Облучим PostgreSQL пучком быстрых запросов
- Проверим гипотезы

Git in Sky
TUNING YOUR INFRASTRUCTURE

Гипотеза о чудесах

- Высоко в горах Старшие эльфы делают секретную ОС, которая превосходит Linux во всём
- FreeBSD жива!
- ZFS лучше всех



Git in Sky
TUNING YOUR INFRASTRUCTURE

Дарвиновская гипотеза

- Ядро 3.16 лучше, чем 2.6.32*
- PostgreSQL 9.4 лучше, чем 9.0
- ext4 лучше, чем ext2

* 2.6.32 отличается от 2.6.32 всем (спасибо RH)

Git in Sky
TUNING YOUR INFRASTRUCTURE

Гипотеза скептика

- Докладчик – лох какой-то
- 9.4 и 9.0 работают с одинаковой скоростью на простых нагрузках
- Ядро Linux давно остановилось в развитии
- Эльфов не бывает

Git in Sky
TUNING YOUR INFRASTRUCTURE

Инженерная гипотеза

- Мы упремся в диск
- Мы упремся в процессор
- Мы упремся в блокировки внутри кода PostgreSQL
- Мы упремся в блокировки внутри ядра

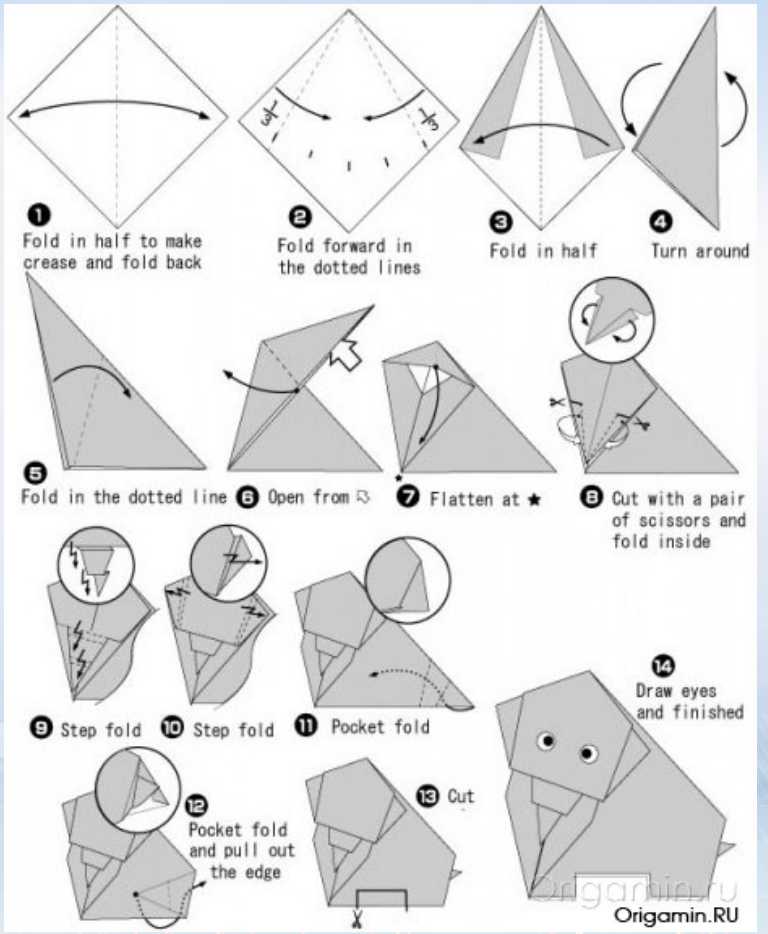
Git in Sky
TUNING YOUR INFRASTRUCTURE

Как все устроено

- Основная тестовая машина (1):
- AMD Phenom(tm) II X4 965 Processor

  32Gb RAM

  1Tb SATA drive, 128Gb SSD drive
- Виртуализация KVM:
- 8Gb RAM, 4 ядра
- pgbench



Git in Sky
TUNING YOUR INFRASTRUCTURE

640Kb should be enough

- Вспомогательная тестовая машина (2):
- Intel Xeon CPU E5-1650 v2 @ 3.50GHz

  128Gb RAM

  4*2Tb SATA drives
- Ubuntu 14.04, PostgreSQL 9.4

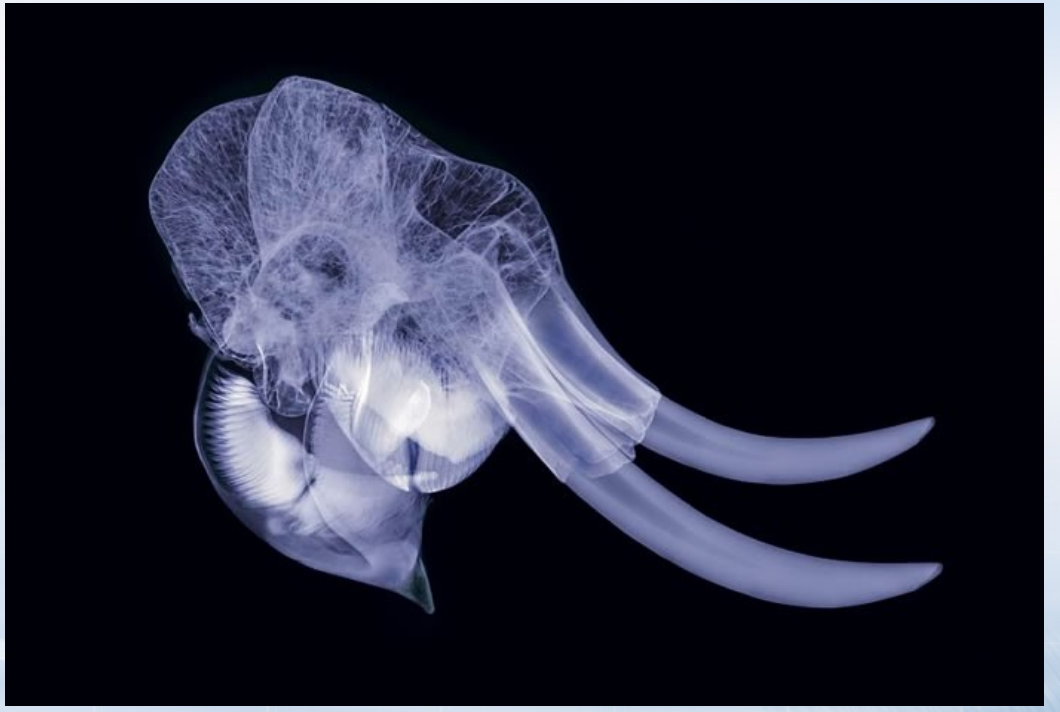
Git in Sky
TUNING YOUR INFRASTRUCTURE

Начнем с конца

- Машина 2, PostgreSQL 9.4
- pgbench -i -s 1000 –foreign-keys \
  pgbench

```
         relation          | total_size
---------------------------+------------
 public.pgbench_accounts   | 15 GB
 public.pgbench_tellers    | 712 kB
 public.pgbench_branches   | 112 kB
 public.pgbench_history    | 0 bytes
(4 rows)
```

- pgbench -t 300000 -r pgbench



Git in Sky
TUNING YOUR INFRASTRUCTURE

Первые результаты

- Машина 2, PostgreSQL 9.4, XFS, какой-то тюнинг конфига

```
postgres@fe10:~$ time pgbench -t 300000 -r pgbench
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1000
query mode: simple
number of clients: 1
number of threads: 1
number of transactions per client: 300000
number of transactions actually processed: 300000/300000
latency average: 0.000 ms
tps = 553.008716 (including connections establishing)
tps = 553.011174 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    9m2.663s
user    0m8.771s
sys     0m9.315s
postgres@fe10:~$
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам

```
0.000975        \set nbranches 1 * :scale
0.000248        \set ntellers 10 * :scale
0.000250        \set naccounts 100000 * :scale
0.000304        \setrandom aid 1 :naccounts
0.000281        \setrandom bid 1 :nbranches
0.000274        \setrandom tid 1 :ntellers
0.000282        \setrandom delta -5000 5000
0.015939        BEGIN;
1.267644        UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
0.065004        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.113583        UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
0.193360        UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
0.123228        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :aid,
0.024045        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Кое-что интересное

- Инженер был прав во всем!

Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util
sda	0.00	0.00	0.00	585.00	0.00	4816.50	16.47	76.97	124.33	0.00	124.33	1.26	74.00
sdb	0.00	5.00	0.00	575.00	0.00	4720.50	16.42	101.28	153.76	0.00	153.76	1.32	76.00
sdc	0.00	0.00	0.00	527.00	0.00	4336.50	16.46	103.35	167.39	0.00	167.39	1.40	74.00
sdd	0.00	5.00	0.00	530.00	0.00	4356.00	16.44	223.71	1322.69	0.00	1322.69	1.89	100.00
md0	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
md1	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
md2	0.00	0.00	0.00	4437.00	0.00	36728.00	16.56	0.00	0.00	0.00	0.00	0.00	0.00
dm-0	0.00	0.00	0.00	5.00	0.00	20.00	8.00	1.64	80.80	0.00	80.80	106.40	53.20
dm-1	0.00	0.00	0.00	4431.00	0.00	36704.00	16.57	2691.39	923.69	0.00	923.69	0.23	100.00

- (Это мы уперлись в диск)

Git in Sky
TUNING YOUR INFRASTRUCTURE

ВАЗ 2101

- Машина 1, VM с CentOS 5.11 (2.6.18), ext4, PostgreSQL 9.4
- Никаких изменений в дефолтном конфиге
- А ЗРЯ

```
< 10:55:37.423 MSK >HINT:  Consider increasing the configuration parameter "checkpoint_segments".
< 10:55:58.319 MSK >LOG:  checkpoints are occurring too frequently (21 seconds apart)
< 10:55:58.319 MSK >HINT:  Consider increasing the configuration parameter "checkpoint_segments".
< 10:56:18.478 MSK >LOG:  checkpoints are occurring too frequently (20 seconds apart)
< 10:56:18.478 MSK >HINT:  Consider increasing the configuration parameter "checkpoint_segments".
< 10:56:39.168 MSK >LOG:  checkpoints are occurring too frequently (21 seconds apart)
< 10:56:39.168 MSK >HINT:  Consider increasing the configuration parameter "checkpoint_segments".
< 10:56:59.609 MSK >LOG:  checkpoints are occurring too frequently (20 seconds apart)
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Закопайте стюардессу

- Ждал полчаса – не дождался, а поэтому

```
# - Checkpoints -

checkpoint_segments = 64
checkpoint_timeout = 1h
checkpoint_completion_target = 0.9
#checkpoint_warning = 30s
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Ура!

- Вместо 300000 транзакций поставил 100000

```
-bash-3.2$ time /usr/pgsql-9.4/bin/pgbench -t 100000 -r pgbench
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1000
query mode: simple
number of clients: 1
number of threads: 1
number of transactions per client: 100000
number of transactions actually processed: 100000/100000
latency average: 0.000 ms
tps = 292.664370 (including connections establishing)
tps = 292.670092 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m41.752s
user    0m8.442s
sys     0m21.044s
-bash-3.2$
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Напоминаю: CentOS 5, 9.4

- Разбивка по запросам

```
0.003861        \set nbranches 1 * :scale
0.001488        \set ntellers 10 * :scale
0.001446        \set naccounts 100000 * :scale
0.001672        \setrandom aid 1 :naccounts
0.001530        \setrandom bid 1 :nbranches
0.001511        \setrandom tid 1 :ntellers
0.001476        \setrandom delta -5000 5000
0.098533        BEGIN;
1.111907        UPDATE pgbench_accounts SET abalance = abalance + :delta W
0.216766        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.199617        UPDATE pgbench_tellers SET tbalance = tbalance + :delta WH
0.185879        UPDATE pgbench_branches SET bbalance = bbalance + :delta W
0.247080        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime)
1.325411        END;
```

- Последняя строчка отличается, почему?

Git in Sky
TUNING YOUR INFRASTRUCTURE

Попробуем схитрить

- Остановим виртуалку
- Настройку cache у виртуального диска сделаем writeback

```
latency average: 0.000 ms
tps = 317.396297 (including connections establishing)
tps = 317.405007 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m15.198s
user    0m7.037s
sys     0m12.978s
```

- Производительность подросла, посмотрим запросы

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам, writeback

- Лучше, но на машине 2 было еще лучше!

```
0.004244        \set nbranches 1 * :scale
0.001298        \set ntellers 10 * :scale
0.001241        \set naccounts 100000 * :scale
0.001432        \setrandom aid 1 :naccounts
0.001325        \setrandom bid 1 :nbranches
0.001299        \setrandom tid 1 :ntellers
0.001293        \setrandom delta -5000 5000
0.061993        BEGIN;
1.837629        UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
0.171002        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.191474        UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
0.156578        UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
0.208654        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid,
0.493458        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Вернем почти все как было

- Но теперь сделаем synchronous_commit=off
- Транзакций стало чуть больше:

```
latency average: 0.000 ms
tps = 298.844151 (including connections establishing)
tps = 298.853310 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m34.735s
user    0m6.751s
sys     0m20.898s
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам

- Понятно, почему END занимал так мало времени на машине 2

```
0.004109        \set nbranches 1 * :scale
0.001434        \set ntellers 10 * :scale
0.001395        \set naccounts 100000 * :scale
0.001617        \setrandom aid 1 :naccounts
0.001472        \setrandom bid 1 :nbranches
0.001425        \setrandom tid 1 :ntellers
0.001414        \setrandom delta -5000 5000
0.112964        BEGIN;
2.128962        UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid
0.221168        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.218633        UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid =
0.192673        UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid
0.249464        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (
0.192055        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Переместимся во времени

- Машина 1, VM с CentOS 6.6 (2.6.32), ext4, PostgreSQL 9.4
- Синхронный коммит пока оставляем, чекпойнты тюним
- ОЙ... пришлость сделать 30000 транзакций, а не 100000

```
latency average: 0.000 ms
tps = 98.727124 (including connections establishing)
tps = 98.729898 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m4.071s
user    0m5.629s
sys     0m3.542s
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Найдем виновника

```
0.010459        \set nbranches 1 * :scale
0.002300        \set ntellers 10 * :scale
0.001941        \set naccounts 100000 * :scale
0.002332        \setrandom aid 1 :naccounts
0.002097        \setrandom bid 1 :nbranches
0.002109        \setrandom tid 1 :ntellers
0.001969        \setrandom delta -5000 5000
0.094950        BEGIN;
1.039033        UPDATE pgbench_accounts SET abalance = abalance + :delta
0.314614        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.357472        UPDATE pgbench_tellers SET tbalance = tbalance + :delta
0.294467        UPDATE pgbench_branches SET bbalance = bbalance + :delta
0.395799        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime
7.574487        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Ладно, асинхронный коммит

- O_o Это было быстро! Вернул 100000 транзакций
- Похоже, мы имеем дело с регрессией производительности, отключение синхронного коммита подходит не всем

```
latency average: 0.000 ms
tps = 304.930437 (including connections establishing)
tps = 304.938215 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m28.035s
user    0m10.971s
sys     0m6.206s
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам

- Коммит работает с той же скоростью, как и на машине 2

```
0.004555        \set nbranches 1 * :scale
0.001421        \set ntellers 10 * :scale
0.001210        \set naccounts 100000 * :scale
0.001465        \setrandom aid 1 :naccounts
0.001346        \setrandom bid 1 :nbranches
0.001356        \setrandom tid 1 :ntellers
0.001256        \setrandom delta -5000 5000
0.041743        BEGIN;
2.303960        UPDATE pgbench_accounts SET abalance = abalance + :de
0.194856        SELECT abalance FROM pgbench_accounts WHERE aid = :ai
0.208123        UPDATE pgbench_tellers SET tbalance = tbalance + :del
0.174085        UPDATE pgbench_branches SET bbalance = bbalance + :de
0.254110        INSERT INTO pgbench_history (tid, bid, aid, delta, mt
0.068245        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Едем дальше

- Машина 1, VM с CentOS 7 (3.10.0), ext4, PostgreSQL 9.4
- Синхронный коммит пока оставляем, чекпойнты тюним
- Регрессия никуда не делась

```
latency average: 0.000 ms
tps = 82.064426 (including connections establishing)
tps = 82.066873 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    6m5.687s
user    0m4.536s
sys     0m2.610s
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Расклад все тот же

```
0.008346        \set nbranches 1 * :scale
0.001767        \set ntellers 10 * :scale
0.001406        \set naccounts 100000 * :scale
0.001962        \setrandom aid 1 :naccounts
0.001726        \setrandom bid 1 :nbranches
0.001575        \setrandom tid 1 :ntellers
0.001515        \setrandom delta -5000 5000
0.075091        BEGIN;
0.875060        UPDATE pgbench_accounts SET abalance = abalance + :delta
0.267433        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.319371        UPDATE pgbench_tellers SET tbalance = tbalance + :delta
0.262085        UPDATE pgbench_branches SET bbalance = bbalance + :delta
0.351341        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime
9.995110        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Лечение все то же

- Асинхронный коммит

```
0.003158        \set nbranches 1 * :scale
0.000867        \set ntellers 10 * :scale
0.000700        \set naccounts 100000 * :scale
0.000891        \setrandom aid 1 :naccounts
0.000801        \setrandom bid 1 :nbranches
0.000767        \setrandom tid 1 :ntellers
0.000688        \setrandom delta -5000 5000
0.026738        BEGIN;
3.452714        UPDATE pgbench_accounts SET abalance = abalance + :delta
0.133026        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.134629        UPDATE pgbench_tellers SET tbalance = tbalance + :delta
0.117364        UPDATE pgbench_branches SET bbalance = bbalance + :delta
0.177184        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime
0.081838        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Попробуем другой фломастер

- Машина 1, VM с FreeBSD 10.1, UFS (w/o softupdates), 9.4
- Синхронный коммит пока оставляем, чекпойнты тюним
- Результат предсказуем – у нас нет журнала на UFS

```
number of clients: 1
number of threads: 1
duration: 300 s
number of transactions actually processed: 89612
latency average: 3.336 ms
latency stddev: 2.088 ms
tps = 298.664109 (including connections establishing)
tps = 298.672114 (excluding connections establishing)
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам

- Без журнала каждая операция быстрее, чем на Linux

```
0.013770        \set nbranches 1 * :scale
0.010256        \set ntellers 10 * :scale
0.010234        \set naccounts 100000 * :scale
0.010389        \setrandom aid 1 :naccounts
0.010167        \setrandom bid 1 :nbranches
0.010145        \setrandom tid 1 :ntellers
0.010175        \setrandom delta -5000 5000
0.171228        BEGIN;
0.780820        UPDATE pgbench_accounts SET abalance = abalance + :delta
0.309149        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.337585        UPDATE pgbench_tellers SET tbalance = tbalance + :delta W
0.313711        UPDATE pgbench_branches SET bbalance = bbalance + :delta
0.365830        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime)
0.855024        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Включим journaled soft-updates

- Машина 1, VM с FreeBSD 10.1, UFS (newfs -U -j), 9.4
- Синхронный коммит пока оставляем, чекпойнты тюним
- Результат все еще предсказуем – теперь журнал есть :)

```
number of clients: 1
number of threads: 1
duration: 300 s
number of transactions actually processed: 67288
latency average: 4.445 ms
latency stddev: 7.249 ms
tps = 224.290075 (including connections establishing)
tps = 224.294151 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m0.139s
user    0m10.888s
sys     0m39.808s
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам

- Естественно, больше всех пострадал COMMIT

```
0.014469        \set nbranches 1 * :scale
0.010636        \set ntellers 10 * :scale
0.010594        \set naccounts 100000 * :scale
0.010692        \setrandom aid 1 :naccounts
0.010565        \setrandom bid 1 :nbranches
0.010476        \setrandom tid 1 :ntellers
0.010572        \setrandom delta -5000 5000
0.183203        BEGIN;
0.815119        UPDATE pgbench_accounts SET abalance = abalance + :delta
0.317819        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.352610        UPDATE pgbench_tellers SET tbalance = tbalance + :delta
0.323195        UPDATE pgbench_branches SET bbalance = bbalance + :delta
0.375061        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime
1.868362        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Окей, асинхронный коммит

- И Linux остается позади, у нас 335 tps и

```
0.019054        \set nbranches 1 * :scale
0.014712        \set ntellers 10 * :scale
0.018756        \set naccounts 100000 * :scale
0.020278        \setrandom aid 1 :naccounts
0.017959        \setrandom bid 1 :nbranches
0.015152        \setrandom tid 1 :ntellers
0.013578        \setrandom delta -5000 5000
0.260180        BEGIN;
0.757913        UPDATE pgbench_accounts SET abalance = abalance + :delta WHEF
0.333951        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.341788        UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE
0.328006        UPDATE pgbench_branches SET bbalance = bbalance + :delta WHEF
0.391262        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VAL
0.258933        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Постойте, постойте

- Мы видим, что во FreeBSD в случае асинхронного COMMIT
  - COMMIT занимает больше времени
  - UPDATE занимает меньше времени
  - Стандартное отклонение времени на операцию, работающую с диском, меньше
- Можем ли мы так в Linux?
  - Планировщик IO? Для virtio дисков он и так none

Git in Sky
TUNING YOUR INFRASTRUCTURE

Постойте, постойте

- Но есть же планировщик на хосте?
  - Но он влияет на все виртуальные машины одинаково
- Опция монтирования data=writeback (“метаданные прежде данных”)
- Попробовал – не помогло, результат тот же

Git in Sky
TUNING YOUR INFRASTRUCTURE

То, ради чего все затевалось

- Машина 1, VM с FreeBSD 10.1, ZFS (с тюнингом), 9.4
- Синхронный коммит можно сразу убрать*, чекпойнты тюним
- Тюнинг ZFS (и его видимый результат):

```
[root@pgday-fbsd101 ~]# zfs get all tank/postgres | grep local
tank/postgres  recordsize            8K                     local
tank/postgres  compression           on                     local
tank/postgres  sync                  disabled               local
[root@pgday-fbsd101 ~]# zfs get all tank/postgres | grep compressr
tank/postgres  compressratio         5.11x                  -
tank/postgres  refcompressratio      5.11x                  -
[root@pgday-fbsd101 ~]#
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

Вы думали, в сказку попали?

- Неутешительный результат

```
number of clients: 1
number of threads: 1
duration: 300 s
number of transactions actually processed: 85033
latency average: 3.514 ms
latency stddev: 13.387 ms
tps = 283.408908 (including connections establishing)
tps = 283.415113 (excluding connections establishing)
statement latencies in milliseconds:
        ...

real    5m0.090s
user    0m12.477s
sys     0m51.534s
```

- Логично – за CoW надо платить

Git in Sky
TUNING YOUR INFRASTRUCTURE

Разбивка по запросам для ZFS

- UPDATE опять вырвался вперед (виновник – CoW?)

```
0.014161        \set nbranches 1 * :scale
0.010506        \set ntellers 10 * :scale
0.010593        \set naccounts 100000 * :scale
0.010750        \setrandom aid 1 :naccounts
0.011128        \setrandom bid 1 :nbranches
0.011333        \setrandom tid 1 :ntellers
0.010653        \setrandom delta -5000 5000
0.165508        BEGIN;
1.557351        UPDATE pgbench_accounts SET abalance = abalance + :delta
0.315840        SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
0.344360        UPDATE pgbench_tellers SET tbalance = tbalance + :delta W
0.317568        UPDATE pgbench_branches SET bbalance = bbalance + :delta
0.364976        INSERT INTO pgbench_history (tid, bid, aid, delta, mtime)
0.237523        END;
```

Git in Sky
TUNING YOUR INFRASTRUCTURE

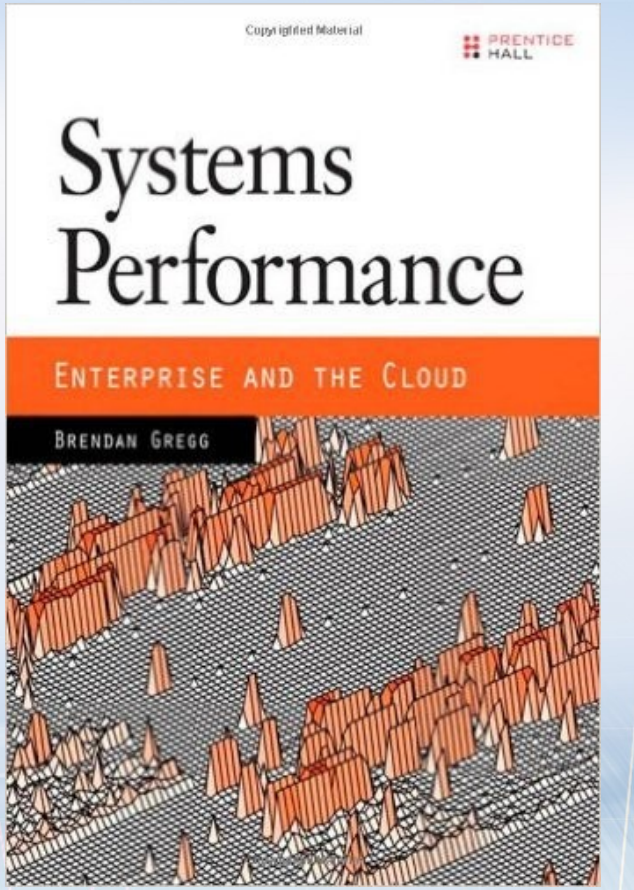
Возьмем другие фломастеры

- DragonFly BSD – нет паравиртуальных драйверов диска
- OmniOS – нет паравиртуальных драйверов диска
- Сравнивать эмуляцию IDE или SATA с virtio как-то не очень правильно
- Мы пытались поставить DragonFly BSD на удаленную машину, но консоль перестала отзываться на нажатия клавиш

Git in Sky
TUNING YOUR INFRASTRUCTURE

Список исп. литературы

- Brendan Gregg “Systems Performance: Enterprise and the Cloud”
- Robert Pirsig “Zen And The Art Of Motorcycle Maintenance”



Git in Sky
TUNING YOUR INFRASTRUCTURE

Выводы

- FreeBSD жива! (технически, умолчания в newfs – это ой)
- ZFS лучше всех (это такой анекдот*)
- ~~Других чудес у меня для вас нет~~ – хахаха, а вот и есть!
- Не чудеса:
  - Не используйте дефолтый конфиг (тюньте саму СУБД)
  - Пользуйтесь средствами вверенной вам ОС (Это моя дисковая подсистема, таких много, но эта – моя...)

Git in Sky
TUNING YOUR INFRASTRUCTURE

Спасибо за внимание!

- Пожалуйста, ваши вопросы?
- С вами был
- Александр Чистяков, главный инженер, Git in Sky
- http://gitinsky.com
- alex@gitinsky.com
- http://meetup.com/DevOps-40