
Яндекс

Яндекс

Пул соединений в масштабе

Владимир Бородин
Администратор баз данных

Почему соединения стоит экономить?

Очевидные вещи

PostgreSQL использует процессную модель

- Один клиент == один процесс (backend)

Внутри backend’а кэшируется много всякого

- Содержимое системного каталога
- Информация о relation’ах и tablespace’ах
- Скомпилированный PL/pgSQL
- Разобранные планы запросов

Неочевидные вещи

ProcArray содержит по элементу на каждый backend

ProcArrayLock обеспечивает конкурентный доступ к ProcArray:

- `LWLockAcquire(ProcArrayLock, LW_SHARED)`
  при каждом создании MVCC snapshot'а
- `LWLockAcquire(ProcArrayLock, LW_EXCLUSIVE)`
  при каждом завершении транзакции и при создании сессии

Итого причины

- Повторное исполнение запросов в том же соединении работает быстрее, чем в новом
- Памяти много не бывает
- Чем больше соединений, тем медленнее работают все

Где можно экономить?

1. В приложении
2. Между приложением и БД
3. На стороне БД
4. Комбинации

Пул соединений в приложении

Есть готовые примитивы примерно во всех языках программирования

В любом случае полезен для поддержания соединений установленными

Плохо работает, когда много:

- серверов приложений
- серверов БД
- и того, и другого ¯_(ツ)_/¯

Пул соединений на стороне БД

Pgpool-II

- Умеет много всякого кроме pooling’а
- SPoF в реалиях нашей сети
- Только session pooling

PgBouncer

- Unix way tool
- Производительность

PgBouncer наше всё

Проблемы

Осложнение диагностики

```
miscdb01d/postgres M # SELECT client_addr, count(*)
FROM pg_stat_activity GROUP BY client_addr;
 client_addr | count
-------------+-------
 127.0.0.1   |   127
 ::1         |   136
(2 rows)

Time: 2.209 ms
miscdb01d/postgres M #
```

Осложнение диагностики

Сложно отлаживать:

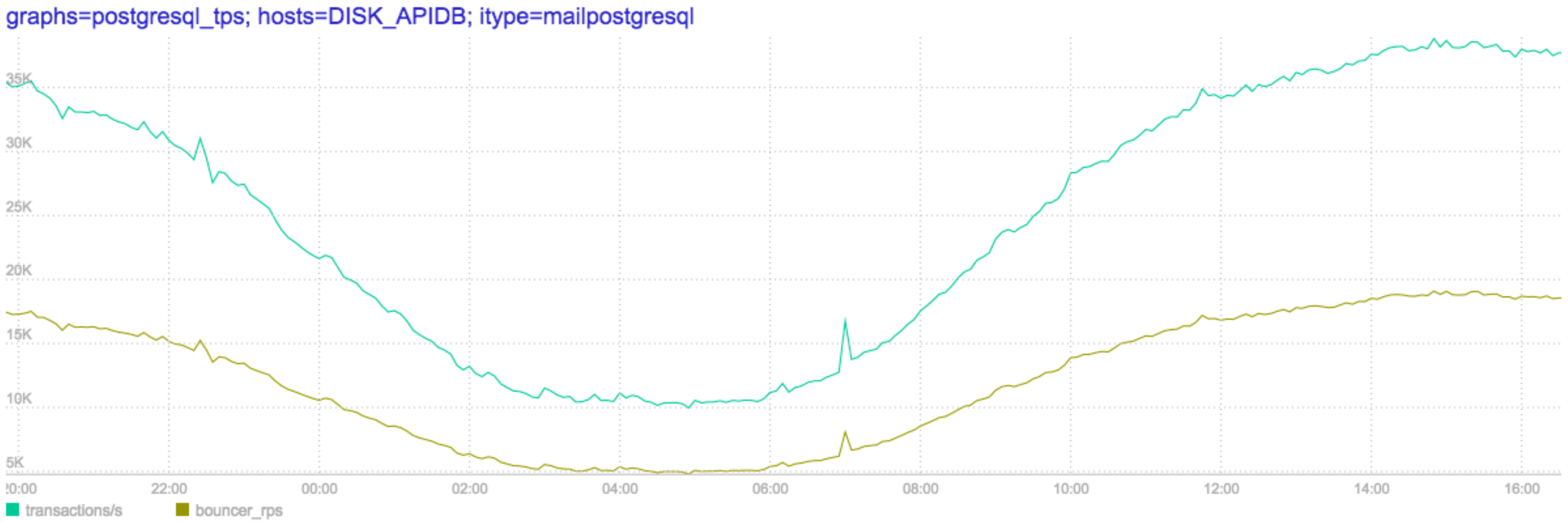
- сетевые проблемы
- ошибки клиентских драйверов

Нет возможности посмотреть происходящее в отдельно взятой сессии

application_name_add_host

```
miscdb01d/postgres M # SELECT client_addr, client_port, application_name
FROM pg_stat_activity LIMIT 1;
-[ RECORD 1 ]----+-----------------------------------------
client_addr      | 127.0.0.1
client_port      | 42051
application_name | app - [2a02:6b8:0:f12:225:90ff:fe94:155c]:50184

Time: 2.716 ms
miscdb01d/postgres M #
```

application_name_add_host



max_client_pool_conn

Нет возможности ограничить клиентские подключения для конкретных базы/пользователя

```
        key          | value
---------------------+-------
 max_client_conn     | 20000
 default_pool_size   | 500
 min_pool_size       | 0
 reserve_pool_size   | 0
```

max_client_pool_conn

Один клиент может открыть `max_client_conn` соединений
и все остальные перестанут обслуживаться

```
2017-03-13 10:36:11.671 28152 LOG C-0x1350dd0: (nodb)/
  (nouser)@[2a02:6b8:0:1a71::21a0]:55760 closing because: no more
  connections allowed (max_client_conn) (age=0)
2017-03-13 10:36:11.671 28152 WARNING C-0x1350dd0: (nodb)/
  (nouser)@[2a02:6b8:0:1a71::21a0]:55760 Pooler Error: no more connections
  allowed (max_client_conn)
```

max_client_pool_conn

Запатчили

```
          key            | value
-------------------------+-------
 max_client_conn         | 20000
 max_client_pool_conn    | 4000
 default_pool_size       | 500
 min_pool_size           | 0
 reserve_pool_size       | 0
```

Pgbouncer cannot connect to server

Нельзя средствами pgbouncer для одной базы пользователю XXX поставить лимит 200 серверных соединений, а пользователю YYY — 10

Зато это умеет PostgreSQL:

- ALTER ROLE XXX WITH CONNECTION LIMIT 200;
- ALTER ROLE YYY WITH CONNECTION LIMIT 10;

Pgbouncer cannot connect to server

2017-03-13 10:48:23.995 24408 ERROR S: login failed: FATAL: too many connections for role "YYY"

psycopg2.OperationalError: ERROR: pgbouncer cannot connect to server

```
>>> try:
...     conn = psycopg2.connect("port=6432 ...")
... except psycopg2.Error as e:
...     print(e.pgcode)
...
None
>>>
```

Обновление на 9.5

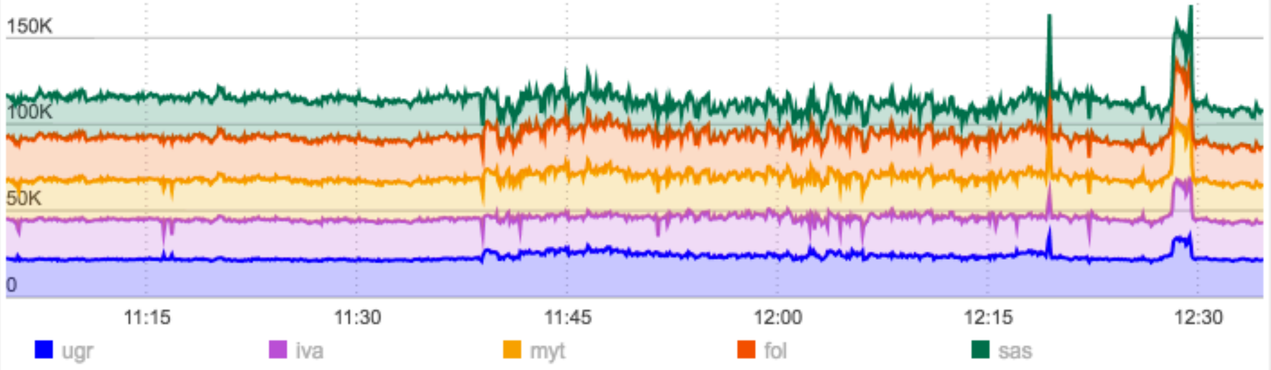
ОС	PostgreSQL	TPS	Среднее время, мс
RHEL 6	9.4	44 898	1,425
RHEL 6	9.5	26 199	2,443
RHEL 6	9.6	43 027	1,487
Ubuntu 14.04	9.4	45 971	1,392
Ubuntu 14.04	9.5	40 282	1,589
Ubuntu 14.04	9.6	45 410	1,409

clck.ru/AjXyE

Обновление на 9.5

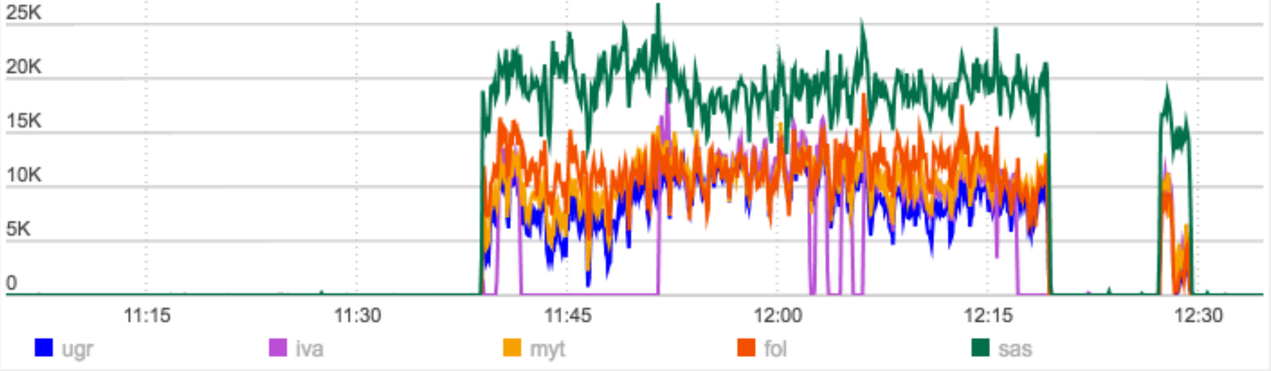
- clck.ru/AjXyE — детальное описание проблемы
- clck.ru/AjXya — решение

HTTP 20x codes by DC

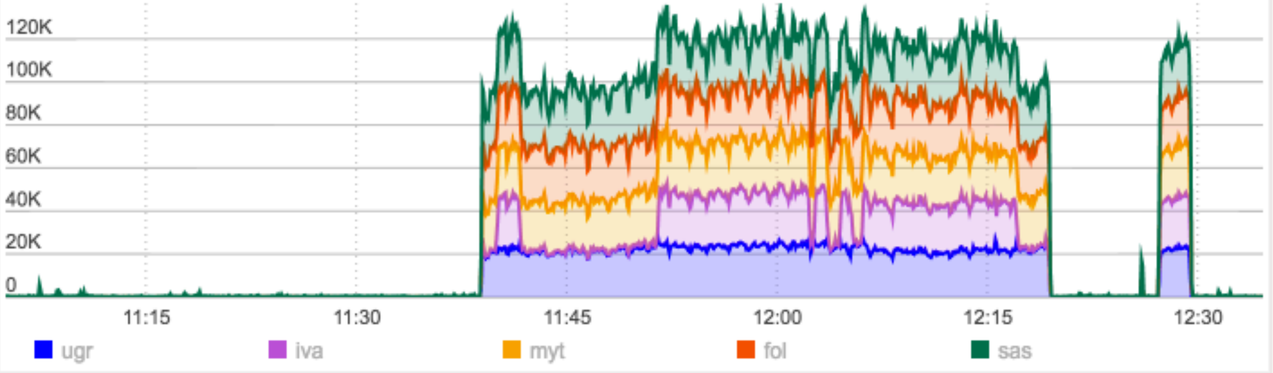


HTTP timings

HTTP 5xx errors by DC



HTTP slow queries by DC



Passport timings

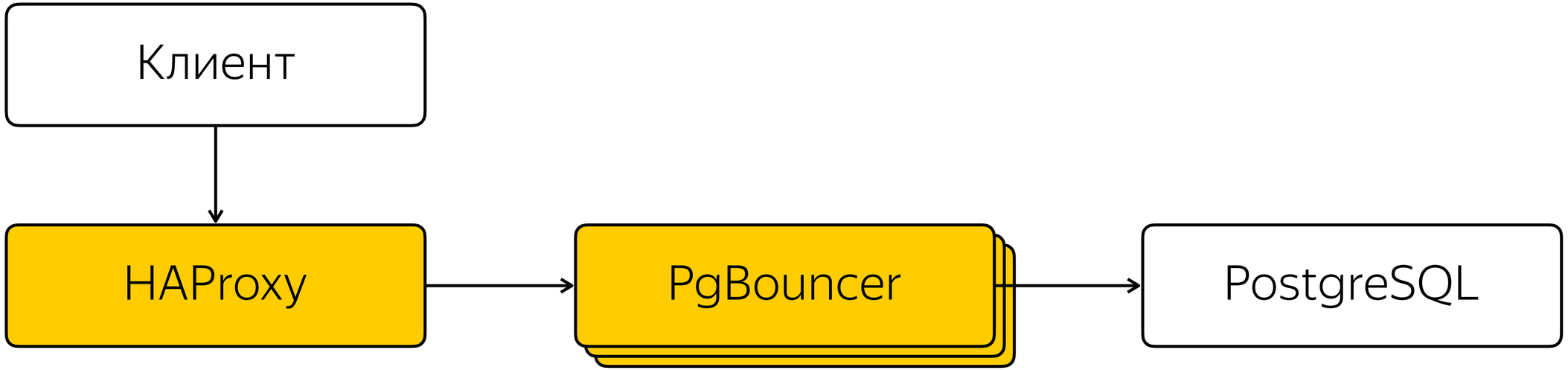
PostgreSQL timings

Производительность

```
  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 3548 pgbounce  10 -10 55344 16860   904 R 97.0  0.0 58h15:40 /usr/bin/pgbouncer -d -q /etc/pgbouncer/pgbouncer.ini
18561 postgres  20   0 66.1G  1804   712 S  7.0  0.0  5h51:48 postgres: wal writer process
21655 postgres  20   0 66.1G  3484  1876 S  5.0  0.0 50:00.07 postgres: wal sender process repl xivadb04d.mail.yandex.net(48473)
21688 postgres  20   0 66.1G  3484  1876 S  5.0  0.0 49:56.82 postgres: wal sender process repl xivadb04g.mail.yandex.net(36924)
26749 root      20   0 15968  1836  1048 R  3.0  0.0  0:02.06 htop
```

Нужно больше pgbouncer’ов

HAProxy



HAProxy

- Прозрачно для клиентов

- Сложность диагностики

  Снова незнание IP-адреса клиента

  Ещё одна движущаяся часть

- Спец. эффекты из-за незнания протокола

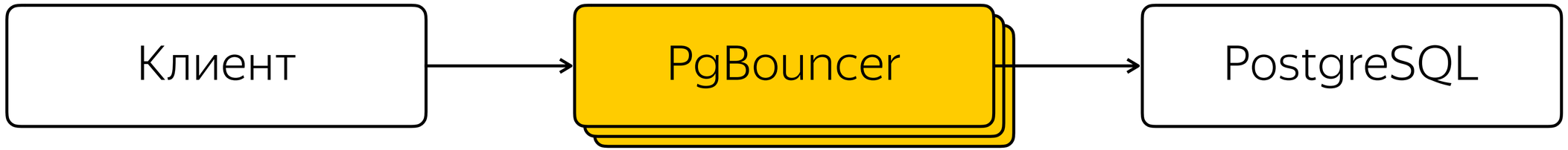
  Заканчивающиеся сокеты при недоступном PostgreSQL

SO_REUSEPORT

https://lwn.net/Articles/542629/

```
+   if (af != AF_UNIX && cf_listen_reuseport == 1) {
+       int val = 1;
+       errpos = "setsockopt";
+       res = setsockopt(sock, SOL_SOCKET, SO_REUSEPORT, &val, sizeof(val));
+       if (res < 0)
+           goto failed;
+   }
```

SO_REUSEPORT

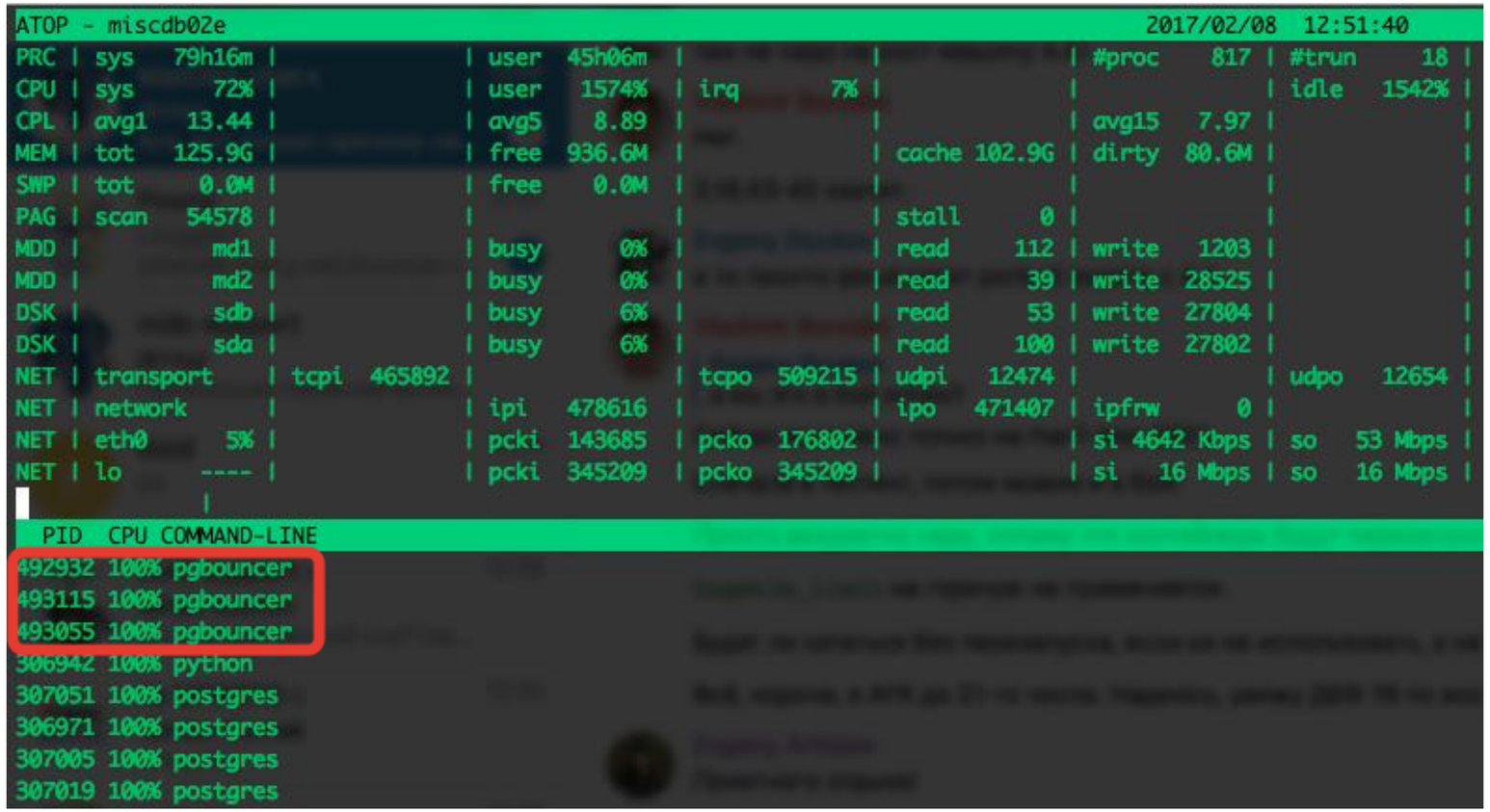


SO_REUSEPORT

- Прозрачно для клиентов
- Нет дополнительных движущихся частей

- Пропорциональное увеличение idle-соединений

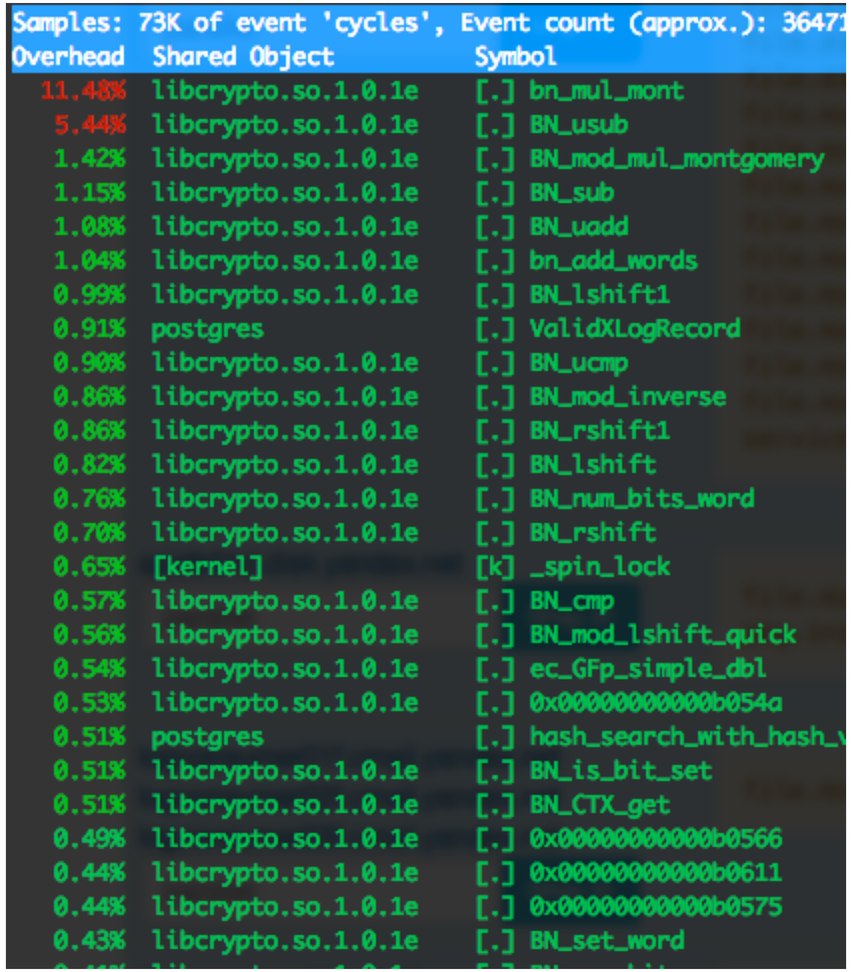
TLS


ATOP - miscdb02e 2017/02/08 12:51:40
PRC | sys 79h16m | user 45h06m | #proc 817 | #trun 18 |
CPU | sys 72% | user 1574% | irq 7% | idle 1542% |
CPL | avg1 13.44 | avg5 8.89 | avg15 7.97 |
MEM | tot 125.9G | free 936.6M | cache 102.9G | dirty 80.6M |
SWP | tot 0.0M | free 0.0M |
PAG | scan 54578 | stall 0 |
MDD | md1 | busy 0% | read 112 | write 1203 |
MDD | md2 | busy 0% | read 39 | write 28525 |
DSK | sdb | busy 6% | read 53 | write 27804 |
DSK | sda | busy 6% | read 100 | write 27802 |
NET | transport | tcpi 465892 | tcpo 509215 | udpi 12474 | udpo 12654 |
NET | network | ipi 478616 | ipo 471407 | ipfrw 0 |
NET | eth0 5% | pcki 143685 | pcko 176802 | si 4642 Kbps | so 53 Mbps |
NET | lo ---- | pcki 345209 | pcko 345209 | si 16 Mbps | so 16 Mbps |
PID CPU COMMAND-LINE
492932 100% pgbouncer
493115 100% pgbouncer
493055 100% pgbouncer
306942 100% python
307051 100% postgres
306971 100% postgres
307005 100% postgres
307019 100% postgres

TLS

```
$ pgbench -C -T 30 -j 300 -c 300 -S postgresql://127.0.0.1:6432/pgbench?sslmode=disable
<…>
latency average: 26.101 ms
tps = 11484.521542 (including connections establishing)

$ pgbench -C -T 30 -j 300 -c 300 -S postgresql://127.0.0.1:6432/pgbench?sslmode=require
<…>
latency average: 523.895 ms
tps = 566.809760 (including connections establishing)
```

TLS


Samples: 73K of event 'cycles', Event count (approx.): 3647
Overhead Shared Object Symbol
11.48% libcrypto.so.1.0.1e [.] bn_mul_mont
5.44% libcrypto.so.1.0.1e [.] BN_usub
1.42% libcrypto.so.1.0.1e [.] BN_mod_mul_montgomery
1.15% libcrypto.so.1.0.1e [.] BN_sub
1.08% libcrypto.so.1.0.1e [.] BN_uadd
1.04% libcrypto.so.1.0.1e [.] bn_add_words
0.99% libcrypto.so.1.0.1e [.] BN_lshift1
0.91% postgres [.] ValidXLogRecord
0.90% libcrypto.so.1.0.1e [.] BN_ucmp
0.86% libcrypto.so.1.0.1e [.] BN_mod_inverse
0.86% libcrypto.so.1.0.1e [.] BN_rshift1
0.82% libcrypto.so.1.0.1e [.] BN_lshift
0.76% libcrypto.so.1.0.1e [.] BN_num_bits_word
0.70% libcrypto.so.1.0.1e [.] BN_rshift
0.65% [kernel] [k] _spin_lock
0.57% libcrypto.so.1.0.1e [.] BN_cmp
0.56% libcrypto.so.1.0.1e [.] BN_mod_lshift_quick
0.54% libcrypto.so.1.0.1e [.] ec_GFp_simple_dbl
0.53% libcrypto.so.1.0.1e [.] 0x0000000000b054a
0.51% postgres [.] hash_search_with_hash_
0.51% libcrypto.so.1.0.1e [.] BN_is_bit_set
0.51% libcrypto.so.1.0.1e [.] BN_CTX_get
0.49% libcrypto.so.1.0.1e [.] 0x0000000000b0566
0.44% libcrypto.so.1.0.1e [.] 0x0000000000b0611
0.44% libcrypto.so.1.0.1e [.] 0x0000000000b0575
0.43% libcrypto.so.1.0.1e [.] BN_set_word

TLS

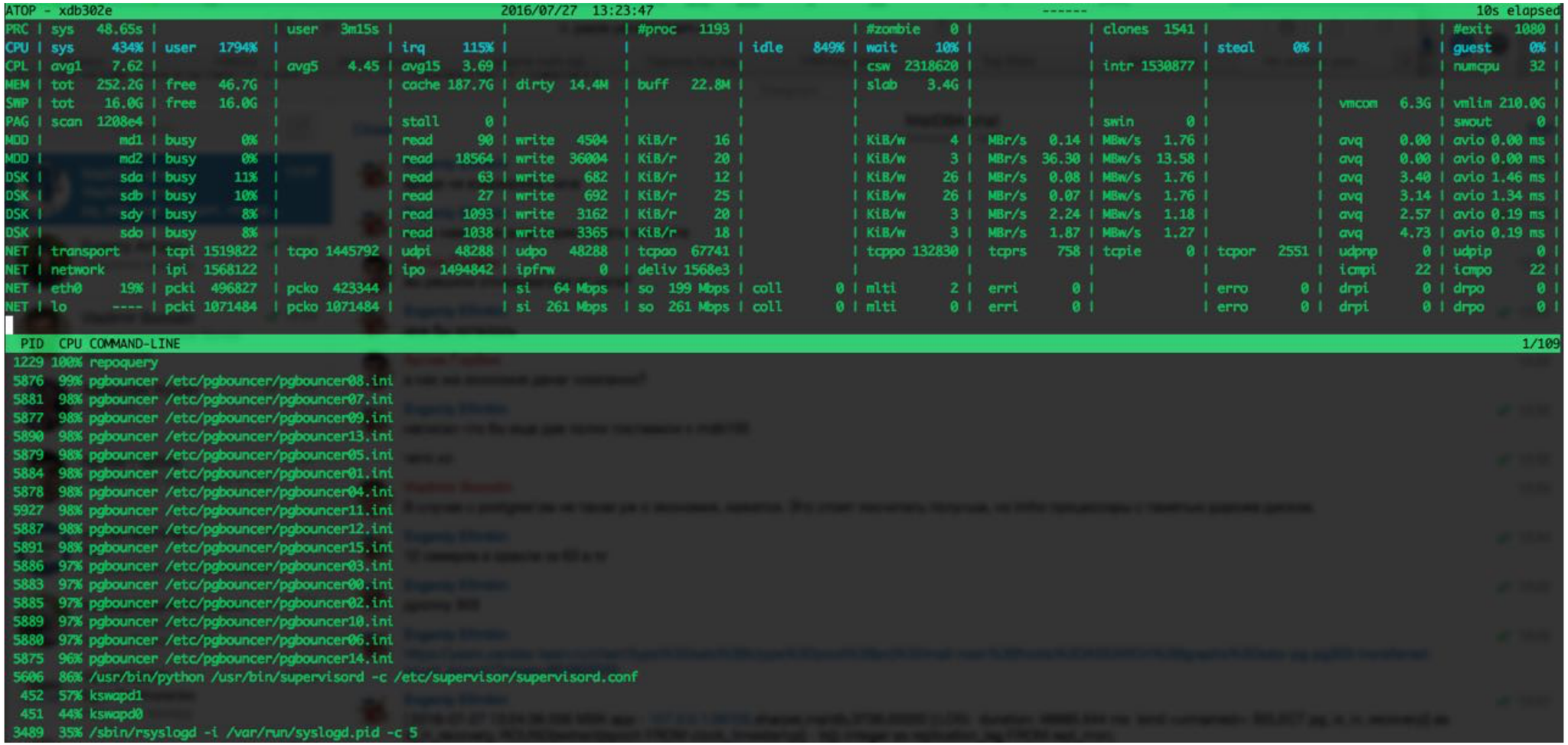
При открытии БД для нагрузки все клиенты примерно одновременно начинают открывать соединения

- Взрыв TLS hadshake’ов

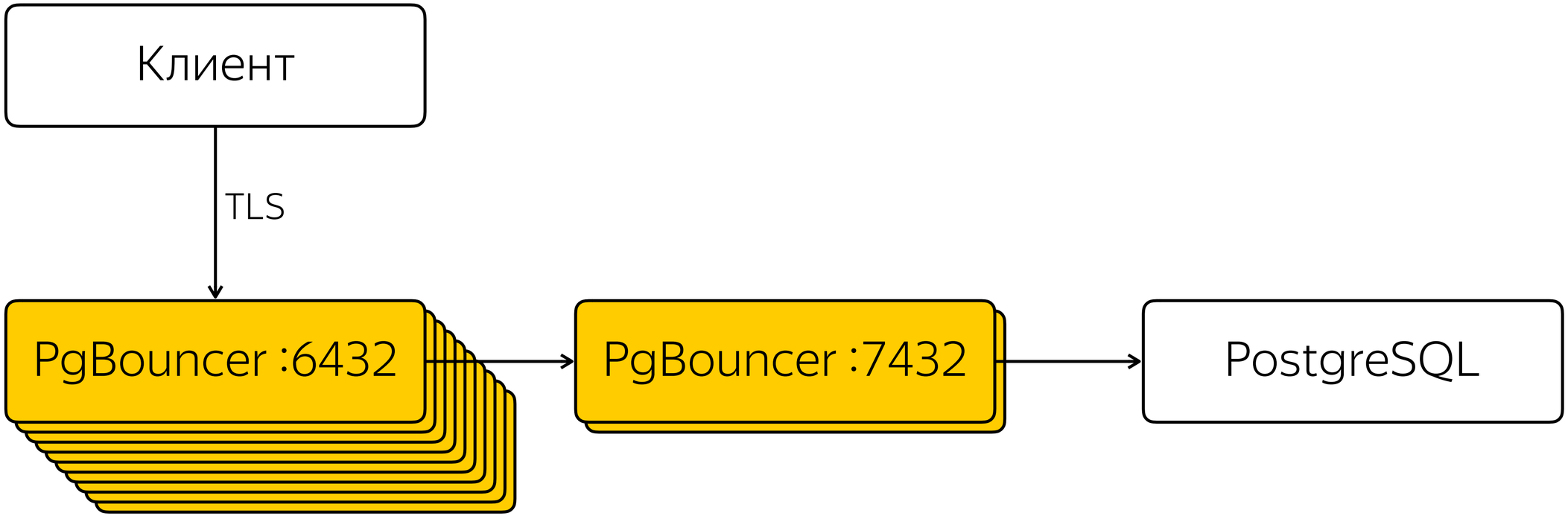
У некоторых клиентов очень маленький connect_timeout

- Клиенты не дожидаются и рвут соединения, а pgbouncer продолжает жечь процессор

TLS


ATOP - xdb302e 2016/07/27 13:23:47 ------ 10s elapsed
PRC | sys 48.65s | user 3m15s | #proc 1193 | #zombie 0 | clones 1541 | #exit 1080 |
CPU | sys 434% | user 1794% | irq 115% | idle 849% | wait 10% | steal 0% | guest 0% |
CPL | avg1 7.62 | avg5 4.45 | avg15 3.69 | csw 2318620 | intr 1530877 | numcpu 32 |
MEM | tot 252.2G | free 46.7G | cache 187.7G | dirty 14.4M | buff 22.8M | slab 3.4G | vmcom 6.3G | vmlim 210.0G |
SWP | tot 16.0G | free 16.0G | swin 0 | swout 0 |
PAG | scan 1208e4 | stall 0 |
MDD | md1 | busy 0% | read 90 | write 4504 | KiB/r 16 | KiB/w 4 | MBr/s 0.14 | MBw/s 1.76 | avq 0.00 | avio 0.00 ms |
MDD | md2 | busy 0% | read 18564 | write 36004 | KiB/r 20 | KiB/w 3 | MBr/s 36.30 | MBw/s 13.58 | avq 0.00 | avio 0.00 ms |
DSK | sda | busy 11% | read 63 | write 682 | KiB/r 12 | KiB/w 26 | MBr/s 0.08 | MBw/s 1.76 | avq 3.40 | avio 1.46 ms |
DSK | sdb | busy 10% | read 27 | write 692 | KiB/r 25 | KiB/w 26 | MBr/s 0.07 | MBw/s 1.76 | avq 3.14 | avio 1.34 ms |
DSK | sdy | busy 8% | read 1093 | write 3162 | KiB/r 20 | KiB/w 3 | MBr/s 2.24 | MBw/s 1.18 | avq 2.57 | avio 0.19 ms |
DSK | sdo | busy 8% | read 1038 | write 3365 | KiB/r 18 | KiB/w 3 | MBr/s 1.87 | MBw/s 1.27 | avq 4.73 | avio 0.19 ms |
NET | transport | tcpi 1519822 | tcpo 1445792 | udpi 48288 | udpo 48288 | tcpao 67741 | tcppo 132830 | tcprs 758 | tcpie 0 | tcpor 2551 | udpnp 0 | udpip 0 |
NET | network | ipi 1568122 | ipo 1494842 | ipfrw 0 | deliv 1568e3 | icmpi 22 | icmpo 22 |
NET | eth0 19% | pcki 496827 | pcko 423344 | si 64 Mbps | so 199 Mbps | coll 0 | mlti 2 | erri 0 | erro 0 | drpi 0 | drpo 0 |
NET | lo ---- | pcki 1071484 | pcko 1071484 | si 261 Mbps | so 261 Mbps | coll 0 | mlti 0 | erri 0 | erro 0 | drpi 0 | drpo 0 |
PID CPU COMMAND-LINE 1/109
1229 100% repoquery
5876 99% pgbouncer /etc/pgbouncer/pgbouncer08.ini
5881 98% pgbouncer /etc/pgbouncer/pgbouncer07.ini
5877 98% pgbouncer /etc/pgbouncer/pgbouncer09.ini
5890 98% pgbouncer /etc/pgbouncer/pgbouncer13.ini
5879 98% pgbouncer /etc/pgbouncer/pgbouncer05.ini
5884 98% pgbouncer /etc/pgbouncer/pgbouncer01.ini
5878 98% pgbouncer /etc/pgbouncer/pgbouncer04.ini
5927 98% pgbouncer /etc/pgbouncer/pgbouncer11.ini
5887 98% pgbouncer /etc/pgbouncer/pgbouncer12.ini
5891 98% pgbouncer /etc/pgbouncer/pgbouncer15.ini
5886 97% pgbouncer /etc/pgbouncer/pgbouncer03.ini
5883 97% pgbouncer /etc/pgbouncer/pgbouncer00.ini
5885 97% pgbouncer /etc/pgbouncer/pgbouncer02.ini
5889 97% pgbouncer /etc/pgbouncer/pgbouncer10.ini
5880 97% pgbouncer /etc/pgbouncer/pgbouncer06.ini
5875 96% pgbouncer /etc/pgbouncer/pgbouncer14.ini
5606 86% /usr/bin/python /usr/bin/supervisord -c /etc/supervisor/supervisord.conf
452 57% kswapd1
451 44% kswapd0
3489 35% /sbin/rsyslogd -i /var/run/syslogd.pid -c 5

Итог

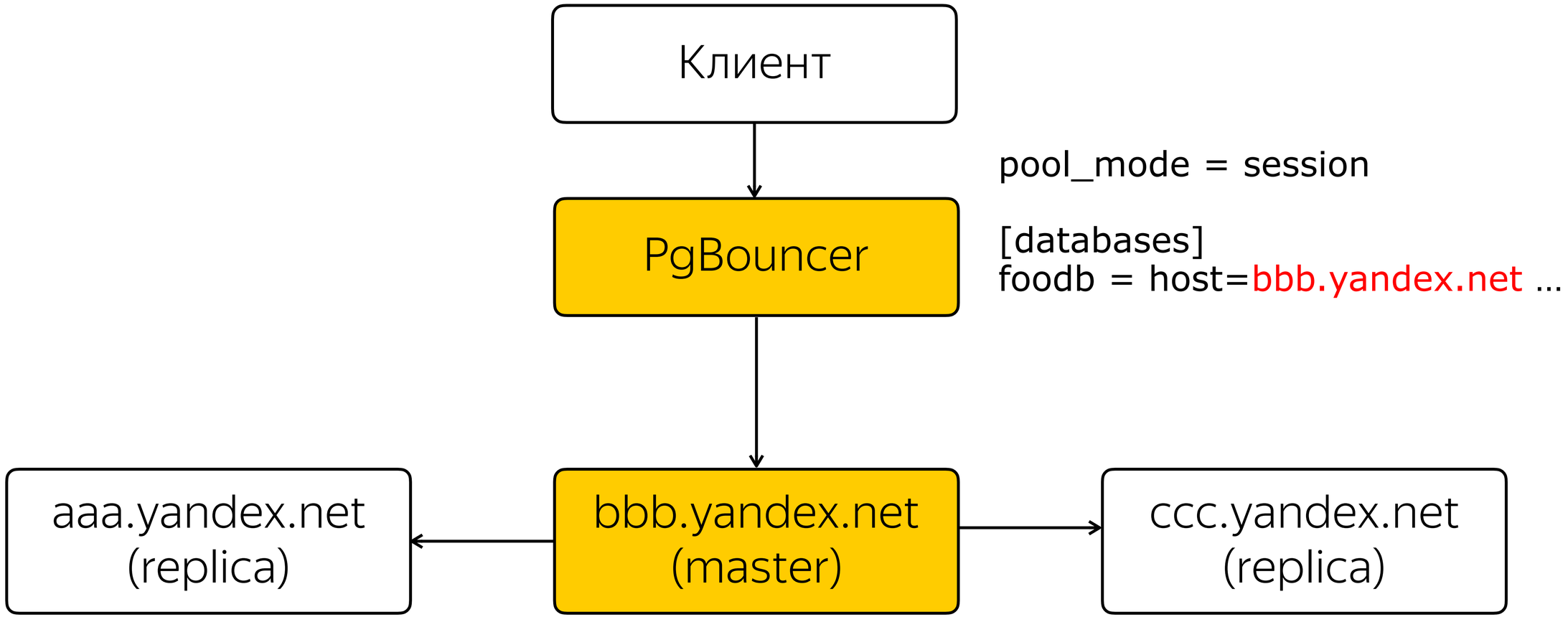


Итог

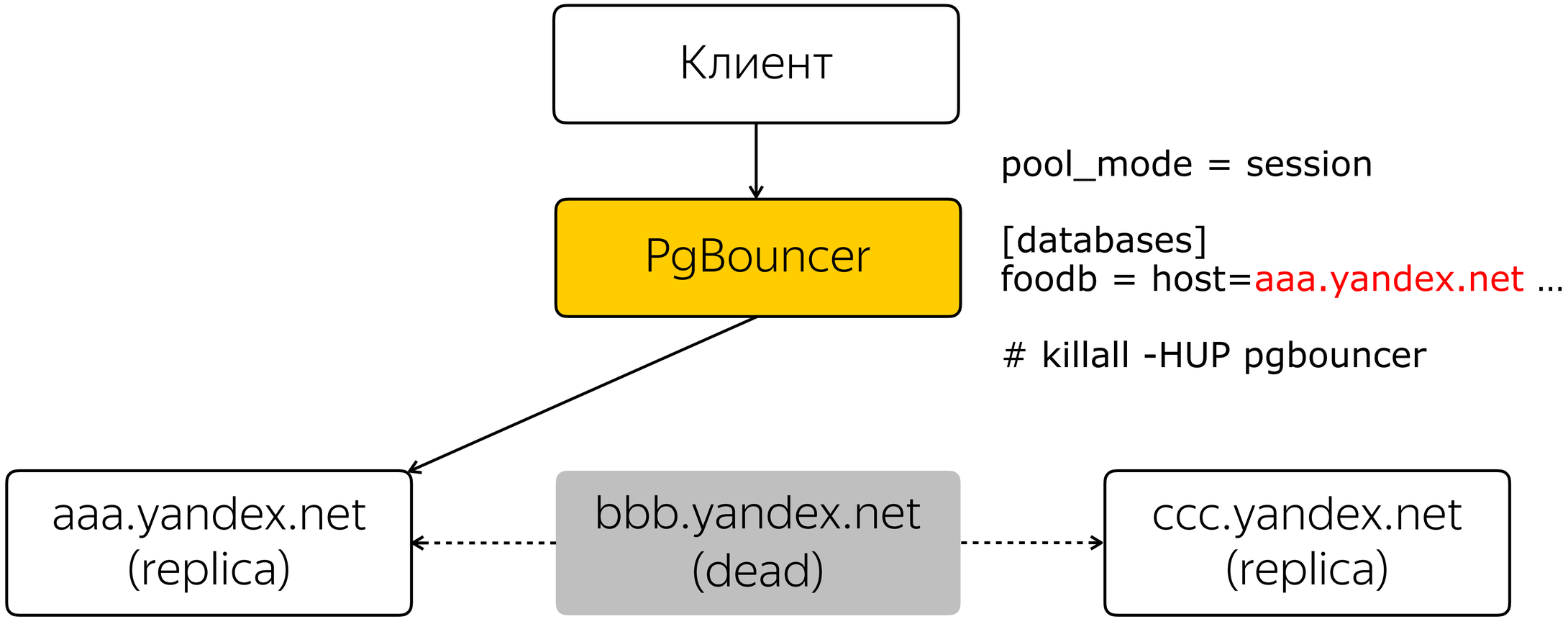
- Прозрачно для клиентов
- Способность держать любую нагрузку
- Контролируемое число idle-сессий

- Сложность обслуживания
- Нет контроля над распределением нагрузки по pgbouncer’ам

Обрыв соединений

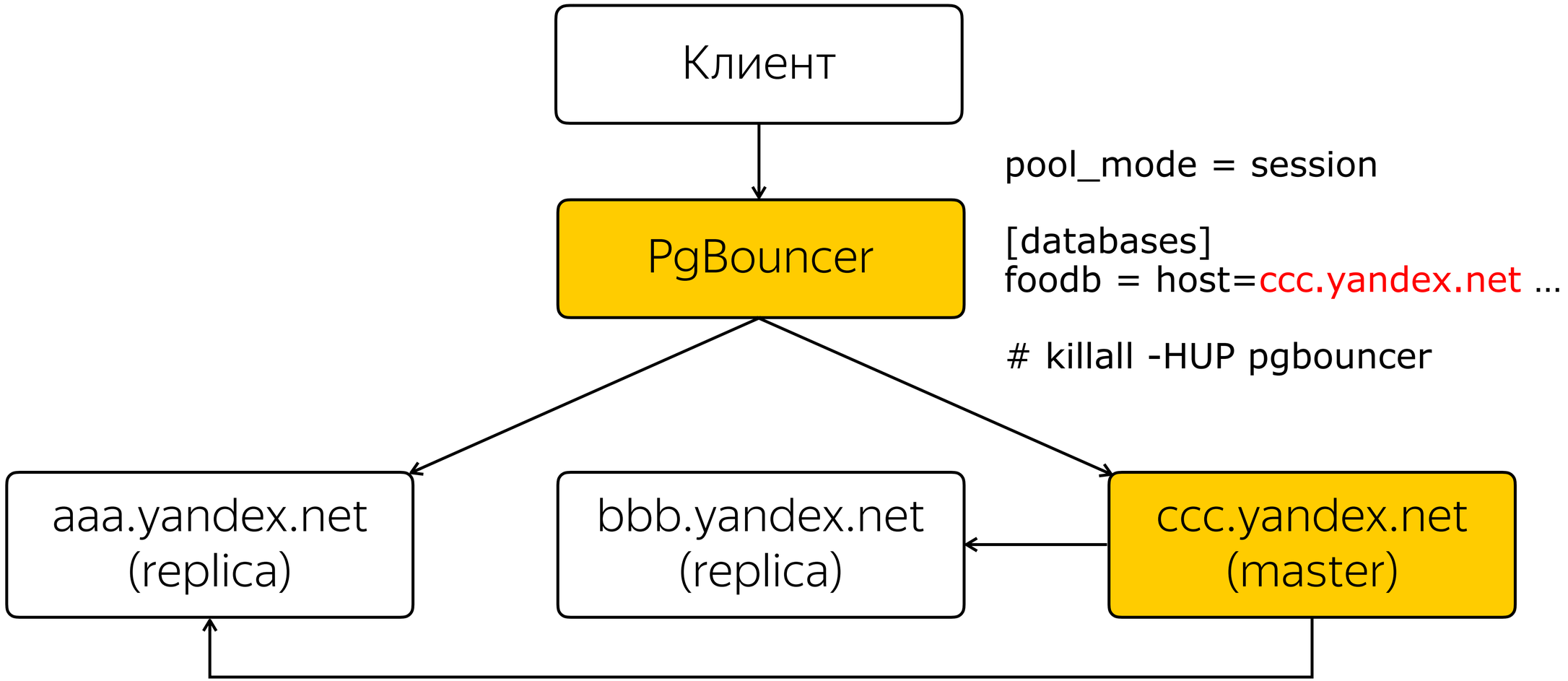


Обрыв соединений

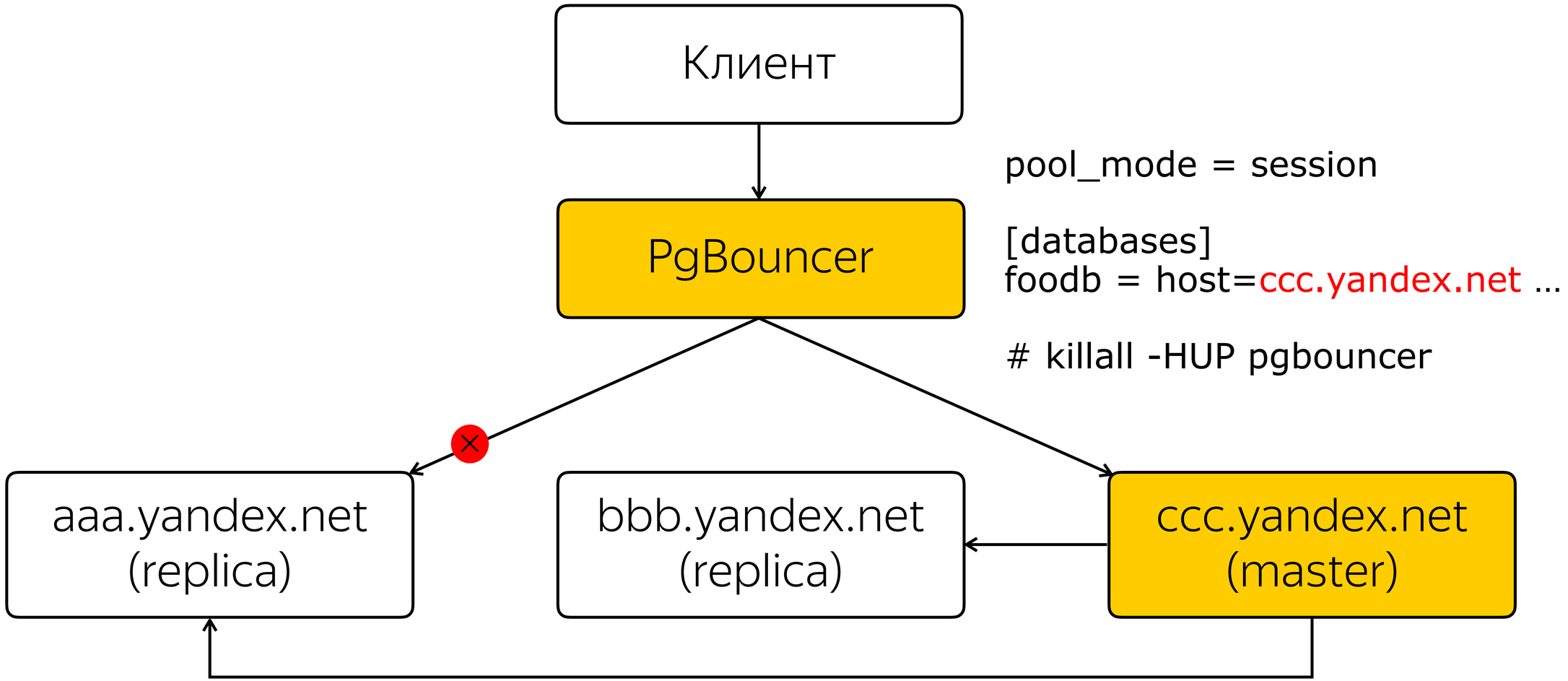



```
pool_mode = session

[databases]
foodb = host=aaa.yandex.net ...

# killall -HUP pgbouncer
```

Обрыв соединений



Обрыв соединений




```
pool_mode = session

[databases]
foodb = host=ccc.yandex.net ...

# killall -HUP pgbouncer
```

Наши изменения в PgBouncer

- SO_REUSEPORT
- max_client_pool_conn
- Обрыв соединений в session pooling
- Ещё пара специфичных для нас вещей

Почему не в open source?

Отмена выполняющегося запроса

Клиент здорового человека

- Откроет новое соединение без аутентификации
- Вызовет PQcancel, передав секрет backend’а из ProcArray
- postgresql.org/docs/current/static/libpq-cancel.html

Клиент курильщика

- Пошлёт TCP reset в соединение

github.com/pgbouncer/pgbouncer/pull/79

Чего хочется?

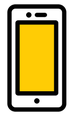
- Масштабирование по ядрам
- Гибкая настройка
- Трассировка конкретной клиентской сессии
- Смешанный тип пулинга
- Нормальные коды ошибок

В самом PostgreSQL?

Вопросы?

Владимир Бородин

Руководитель группы эксплуатации
систем хранения метаданных

 +7 (495) 739 70 00, доб. 7255

 @man_brain

 d0uble@yandex-team.ru

 https://simply.name