Большая база объявлений

Тюрин Михаил
mtyurin@avito.ru

pgconf.ru
2015

Всем привет!

Михаил Тюрин, Avito.ru

- главный системный архитектор, базы данных
- с PostgreSQL с 2004-ого, с версий 7.4/8.0, региональный web-стартап!
- в **Авито** с 2009-ого:
  - один из **четырех** первых разработчиков; DBA / sql / plpgsql / php / skytools / linux
  - DBA и DB-team lead, развитие смежных компонент
  - архитектура, хранение и обработка данных в целом
  - работа с базами данных PostgreSQL – основное хранилище:
    - директор по разработке и девопс-директор, отдел «аналитиков»
    - DB-team: DBA и sql-девелоперы; – 7 человек *// есть 2 (две) вакансии*
    - DevOps/SysAdm – 0.5 человека «на базе» *// есть вакансии*
    - отдельные серверсайд разработчики
    - отдельные DWH разработчики («аналитики»)
  - отвечаю в том числе за то, чтобы данные «лежали» как надо и где надо

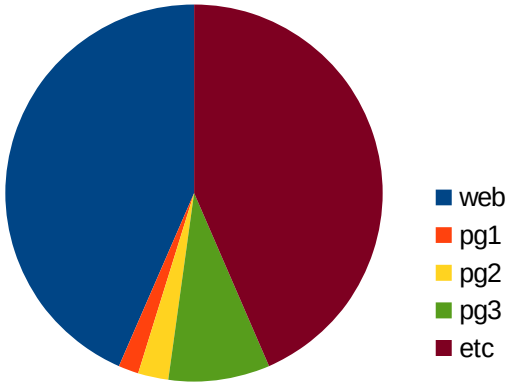
Avito в цифрах

- Авито – доска N1 в России (Европе)
  - «большой» и мобильный сайты; api – приложения и смежные проекты
  - топ 10 сайт в России
    - 250M просмотров и
    - 7M посетителей в сутки
  - 25M активных объявлений,
  - 700K новых объявлений ежедневно

- ~ 500 человек штат (кто все эти люди?!)
  - 80 человек «отдел» разработки
  - DB-team: 7 человек, + несколько выделенных серверсайд девелоперов

Еще цифры

- 200+ серверов на основной платформе
  - ~ 100 web app серверов – основные клиенты баз
  - ~ 30 серверов с pg
    - ~ 10 одновременно под нагрузкой
    - **4 – под высокой**
- postgres под нагрузкой
  - «мастер» база – db, «m1»
    - 1.5TB; 3KTPS и 1.5K IUD/sec в среднем
      - 8K iops (5–12K) (w/r 2/6K), w/r 25/80 MB/s – 60% utilization SSD raid
      - 20-50 active pg connections (intel 20(40ht) CPU, 376GB DDR)
  - «стендбай» – db-sb: 600TPS
  - «мастер2» – db2, «m2»: HDD raid, 2.5TPS (amd 16CPU, 80GB DDR) // ! uptime: 500d
  - «repca» – db-re: 7KTPS, «in-memory» pg (16 CPU)



Как мы к этому шли

- начало: php + memcached и postgres – **PostgreSQL**
- первые шаги: + pgpool – **fail**
- postgresql + tsearch – ok
- skype: **pgbouncer** – **OK OK OK !!!** *

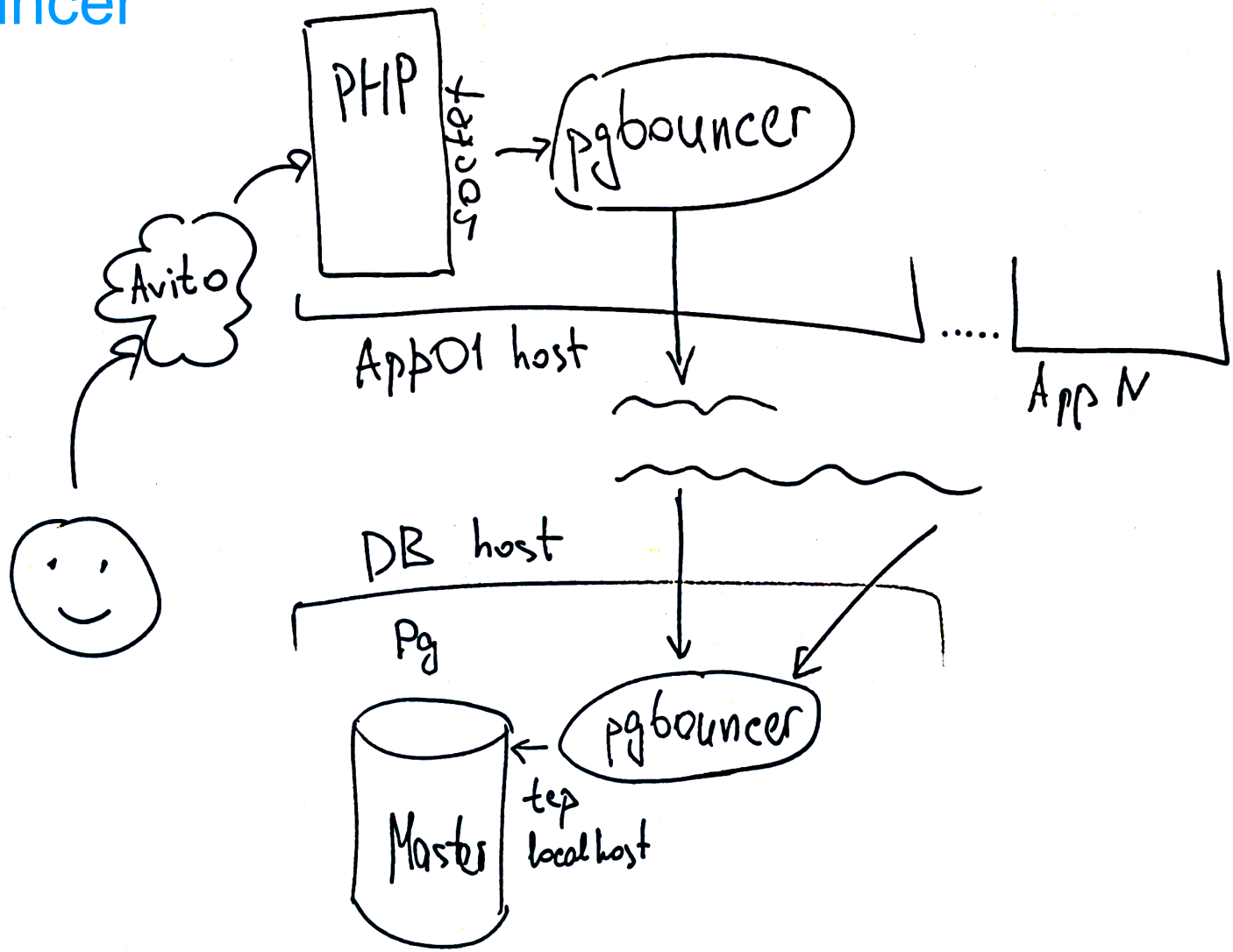
Продолжаем идти

- skytools: walmgr – **fail**
- частичный индекс WHERE ACTIVE – ok
- sphinx – **OK**
- skytools: php-deamon – **OK**
- upgrade на 9.0 – bug
  https://wiki.postgresql.org/wiki/20110408pg_upgrade_fix

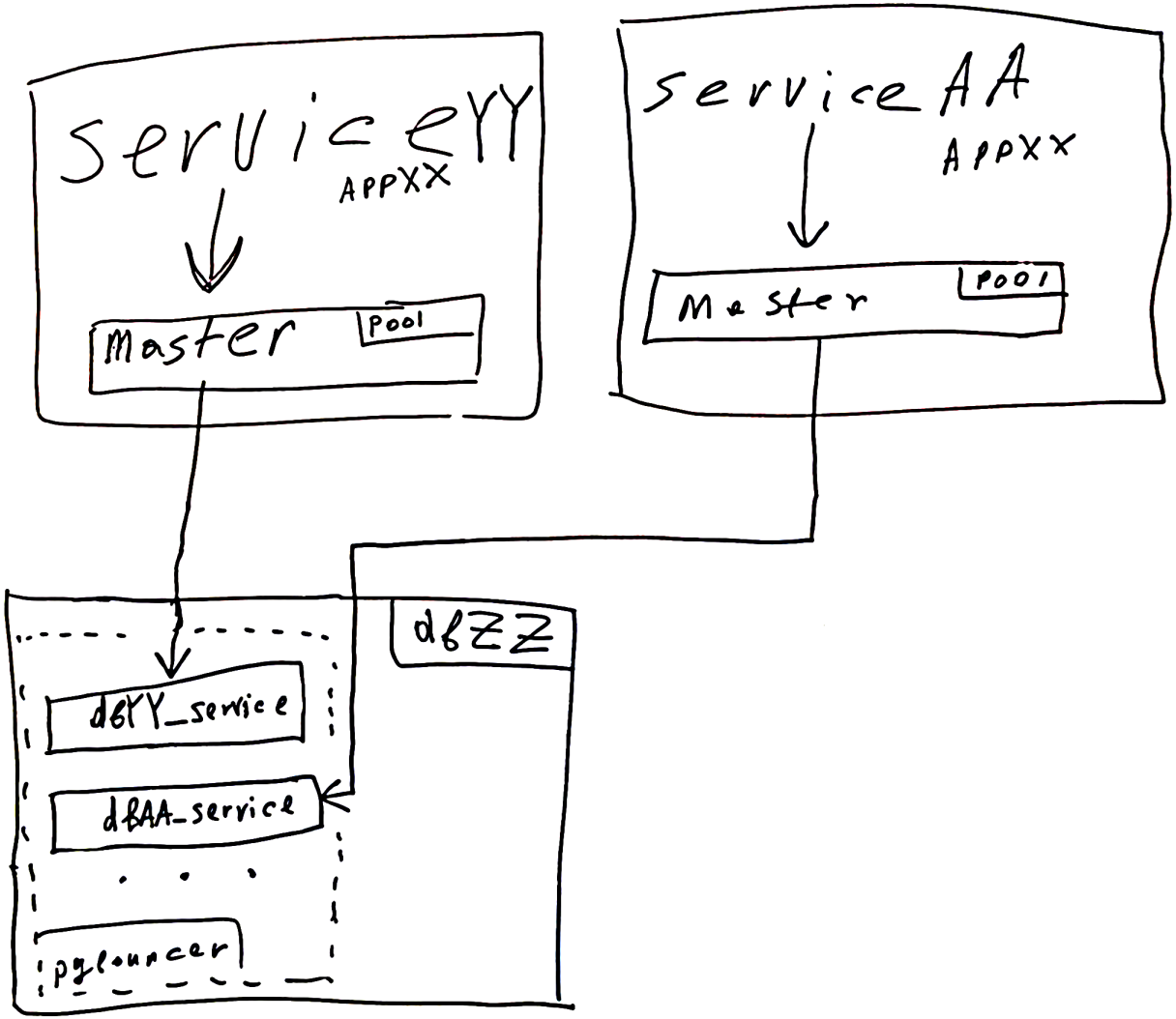
  ERROR: could not access status of transaction ######

  DETAIL: could not open file "pg_clog/####": No such file or directory=20
- split: m1/m2 – **OK** *
- **mat-view** на deffered triggers – **OK** *
- skytools: londiste – «repca» – **PGQ – OK OK !!** *
- pitr: pg_basebackup, pg_archivecleanup – **OK** *
- skytools: plproxy – хранилище «xdb» – **OK** *
- pgq consumer – «xrpc» – **OK** *

PgBouncer


PHP
socket
pgbouncer
Avito
App01 host
App N
DB host
Pg
pgbouncer
Master
tcp
localhost

PgBouncer


serviceYY
APPXX
Master
Pool
serviceAA
APPXX
Master
Pool
dbZZ
dbYY_service
dbAA_service
pgbouncer

PgBouncer

- pool коннектов
- мультиплексор: transaction pooling
- очередь запросов на одном процессе
  - два пула: на клиенте и на сервере
    - защита от «съедания» центрального ресурса одним клиентом
    - распределение обработки очереди между двумя процессами
    - дополнительный удобный прокси уровень
  - nice -10 при нагрузке и высоксом la при cpu-bound
  - поднятие более одного pgbouncer на сервере базы под высоким трафиком/нагрузкой
- connect query: *select x_init()*
- proxy
  - маршрутизация трафика
  - смена юзера: [app --> pgbouncer] == > [pgbouncer – client_usr --> db]
  - смена параметров коннектов: x_init() / x_init('log = on')
- http://pgbouncer.projects.pgfoundry.org/doc/usage.html#_admin_console

db split

← split start

1) init standby

2.1) promote standby

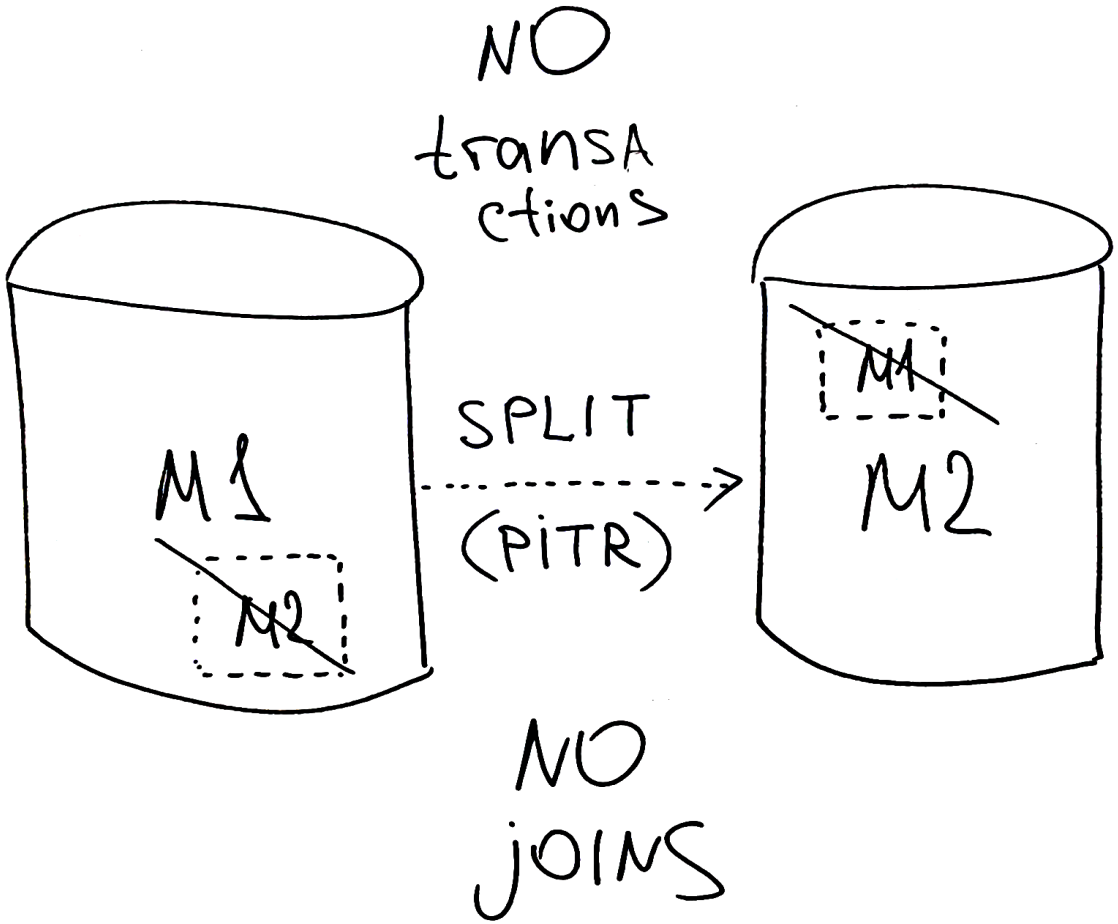
2.2) switch connections m2 to new host-m2

← split finish

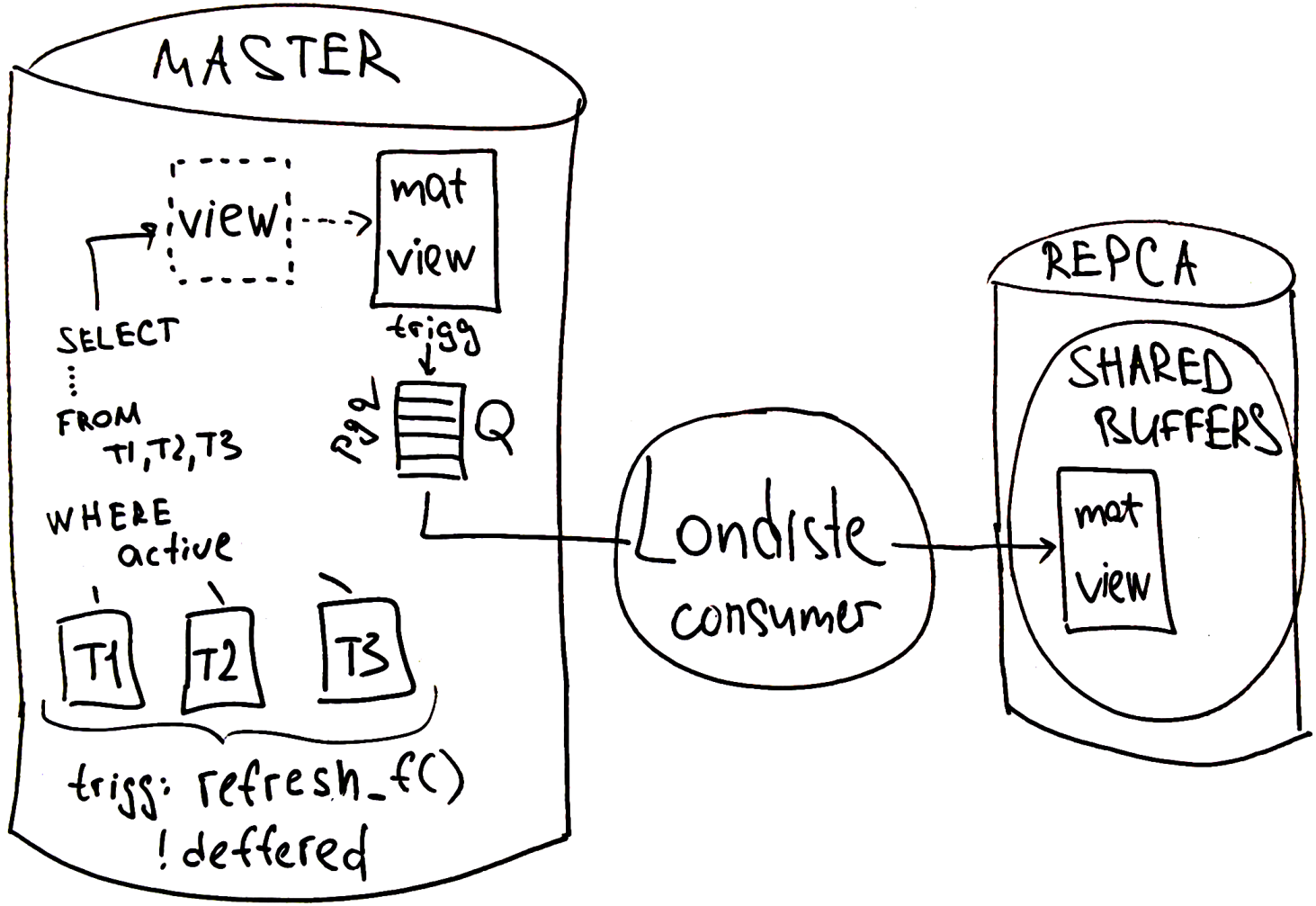
3) drop database:

- m2 on host-m1
- m1 on host-m2

NO tx && NO joins



«repca»! – londiste: trigger based logical replication



repca! – londiste: trigger based logical replication

- «матвью» активных айтимов

  select item_id as id, get_hstore_pararms(category_id, item_id) as params from items where active

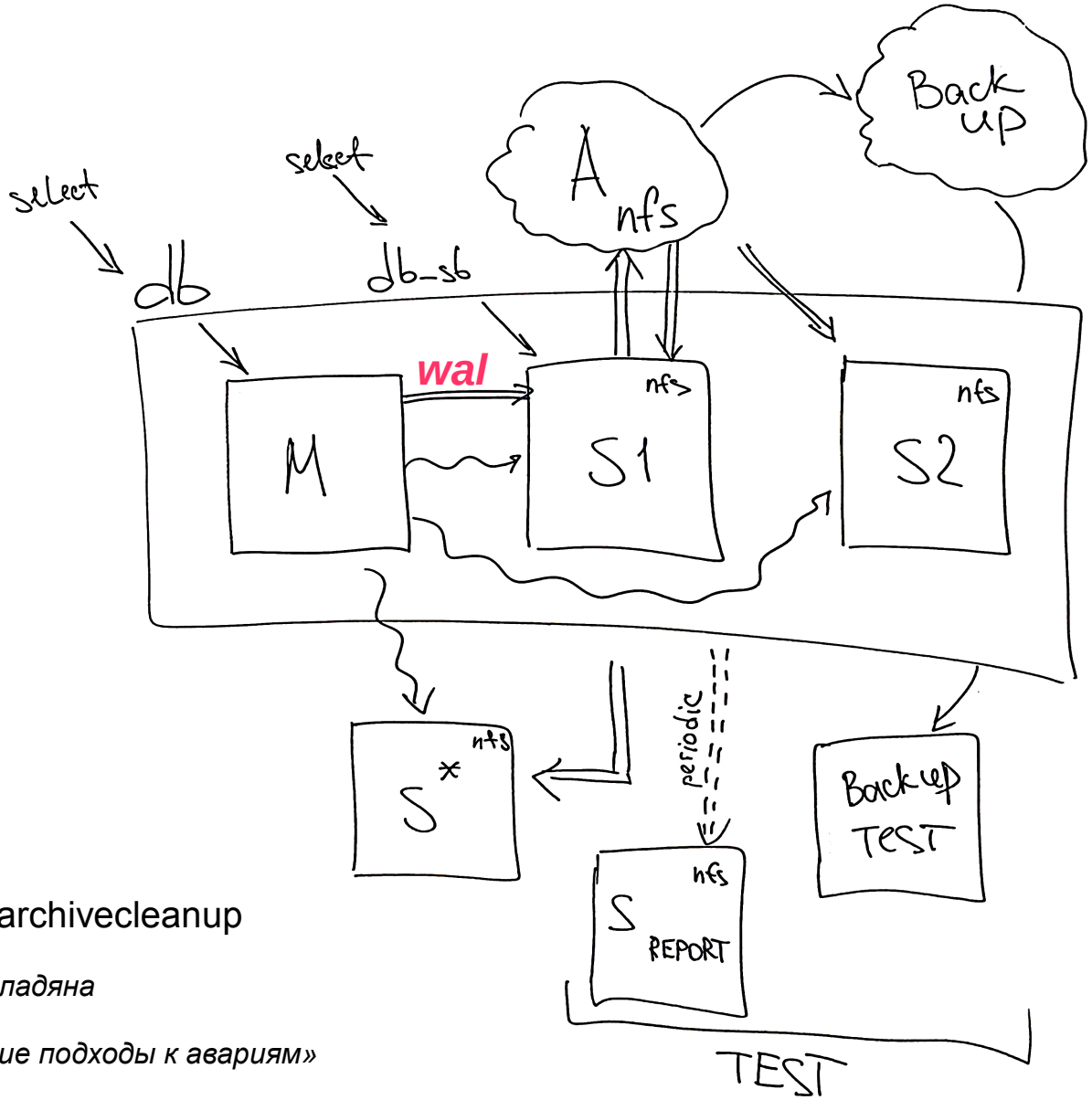
  http://www.pgcon.org/2008/schedule/attachments/63_BSDCan2008-MaterializedViews.pdf

  http://www.pgcon.org/2008/schedule/attachments/64_BSDCan2008-MaterializedViews-paper.pdf

  + deffered trigger: refresh_function()

- **hstore**!!! // привет Олегу и Федору
  - не правим «матвью» при изменении полей таблиц объявления
- pgq: londiste «матвью» => repca
- repca в шаред буфферах – 7 К TPS
- восстановление:
  - резервирование — ! два коснумера на одну очередь (knowhow)
  - UNDO log (супер knowhow) и REDO фича londiste
  - *!!! см. доклад Сергея Бурладяна «Восстановление и другие подходы к авариям»*

Архив



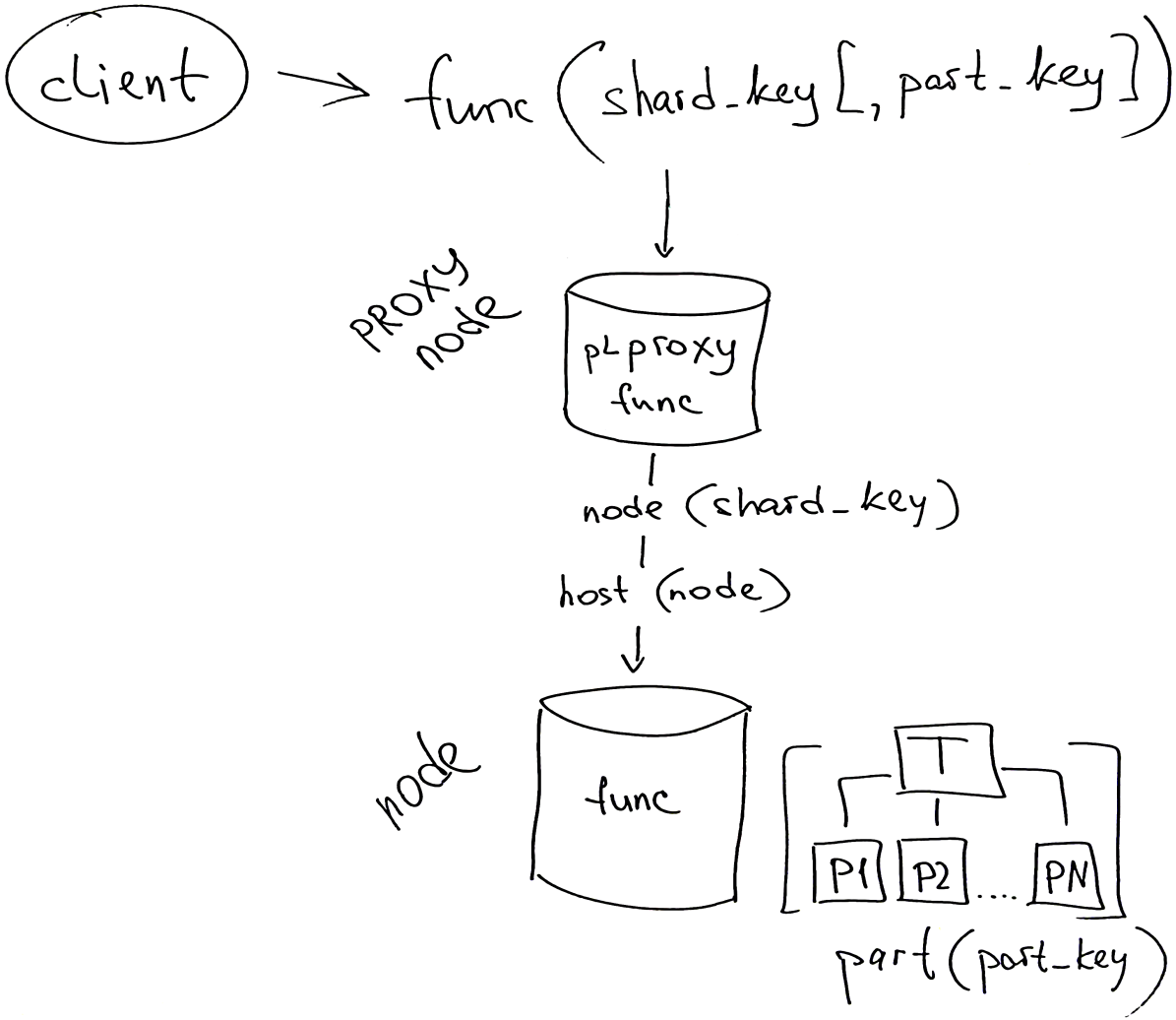
PITR:

pg_basebackup, pg_archivecleanup

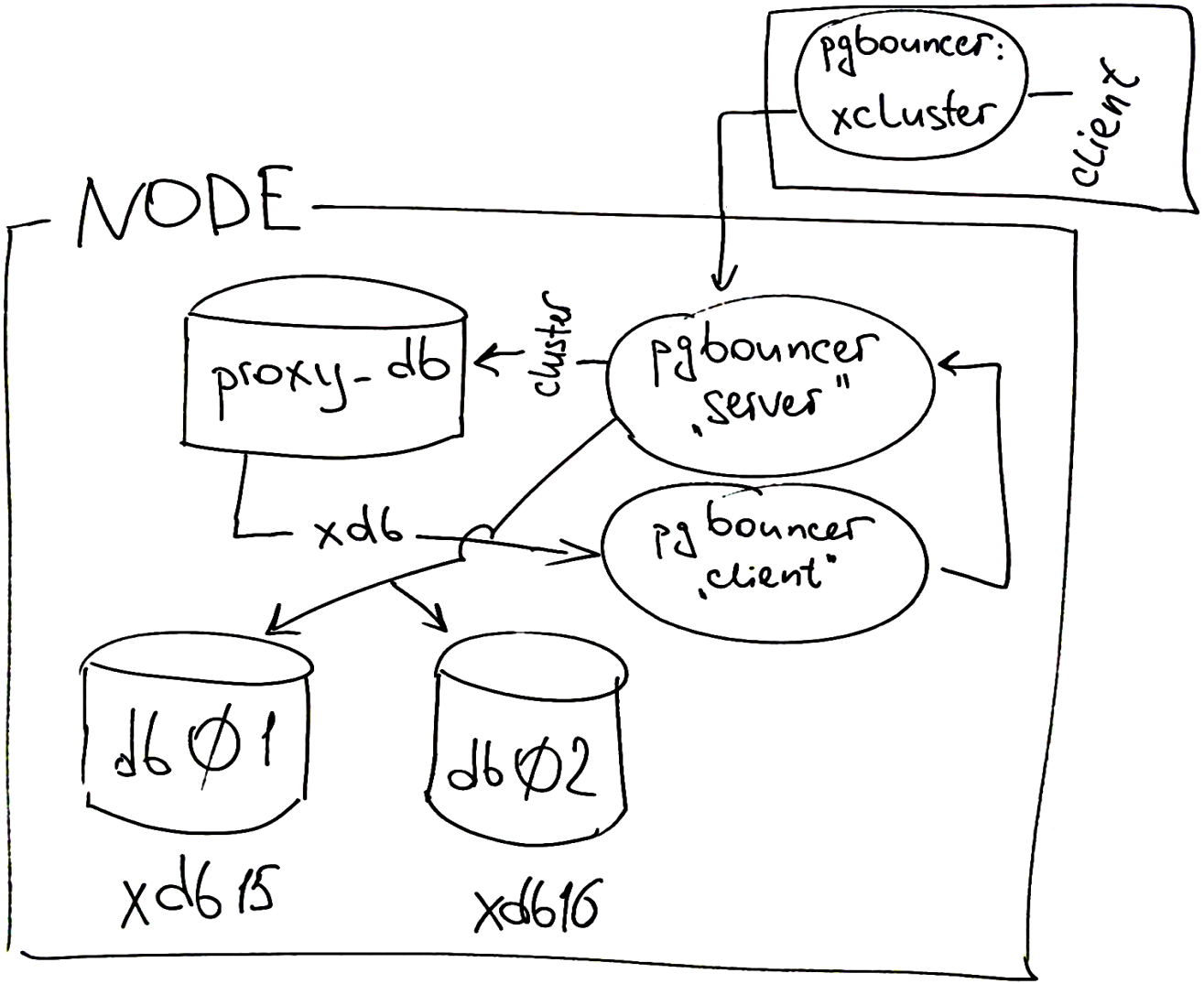
!!! см. доклад Сергея Бурладяна

«Восстановление и другие подходы к авариям»

«XDB» – распределенное хранилище: логическая схема



«XDB» – физическая схема



«XDB»: пулы и размер

- ***pgbouncer.ini_xdb-node_server***

  ;; для внешних клиентов

  xcluster = host=localhost user=xuser pool_size=30 datestyle='ISO,DMY'

  ;; для внутреннего обратного подключения из plproxy

  db_01 *= host=localhost user=xuser pool_size=10 datestyle='ISO,DMY'*

  db_02 *= host=localhost user=xuser pool_size=10 datestyle='ISO,DMY'*

- **pgbouncer.ini_xdb-node_client // ROUTER**

  xdb_**01** = host=xdb-**node01** port=6432 dbname=**db_01** pool_size=10 datestyle='ISO,DMY'

  xdb_**02** = host=xdb-**node01** port=6432 dbname=**db_02** pool_size=10 datestyle='ISO,DMY'

  xdb_**03** = host=xdb-**node02** port=6432 dbname=**db_01** pool_size=10 datestyle='ISO,DMY'

  ...

  xdb_**15** = host=xdb-**node08** port=6432 dbname=**db_01** pool_size=10 datestyle='ISO,DMY'

  xdb_**16** = host=xdb-**node08** port=6432 dbname=**db_02** pool_size=10 datestyle='ISO,DMY'

- CREATE FUNCTION **public.xdb_name()** RETURNS text LANGUAGE sql AS

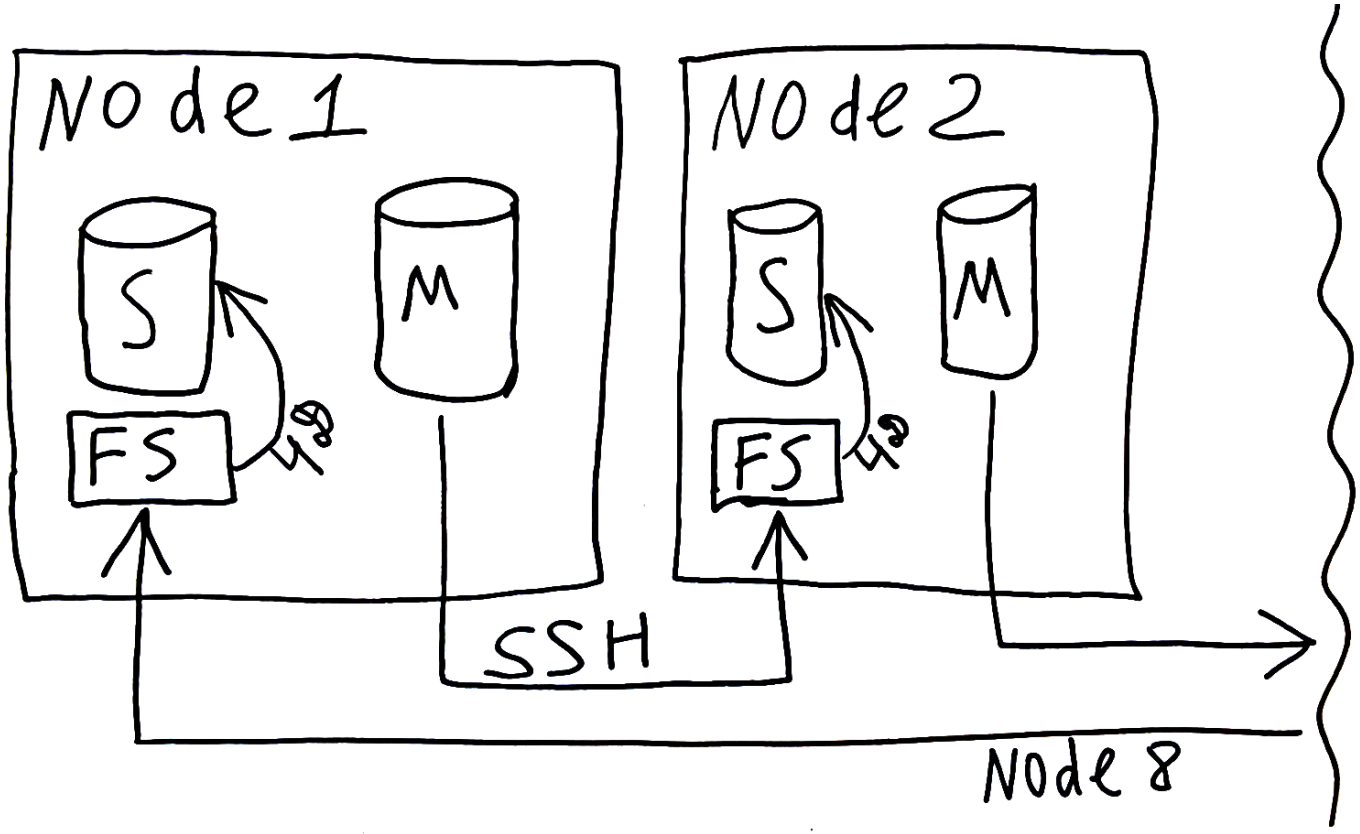
  $_$ select format('%s %s', current_setting('xdb.server'), current_database()); $_$;

- 7 TB: отправленные письма (7 месяцев по закону), логи, дампы

«XDB»: архив

ноды резервируют сами себя – отложенный стендбай как архив:

!! см. доклад Сергея Бурладяна ***«Восстановление и другие подходы к авариям»***



«xrpc»

- «базы» вызывают друг друга асинхронно и получают асинхронный коллбек
- FUNCTION xrpc._call(queue text, _dbconn text, _func text, args hstore) RETURNS bigint

```
return pgq.insert_event(    queue,                      -- queue_name
                            DEFAULT_CALL_TYPE,          -- ev_type
                            '1.1',                      -- ev_data      -- rpc version
                            _dbconn,                    -- ev_extra1
                            _func,                      -- ev_extra2
                            args::text,                 -- ev_extra3
                            now()::timestamp::text      -- ev_extra4
) ; -- call_id: event_id;
```

- xrpcd – consumer, для каждой базы собирает свои вызовы xrpc.do_call(func, args)

```
execute 'select '''' || replace(func, '.', '''.''') || '''( $1 )' using args;
```

- xrpcd обрабатывает аварии через трекание batch_id // xrpc.set_batch_done
- примеры:
  - «геокодер» – где-то по дороге ходит в Яндекс
  - асинхронная «транзакционная» обработка больших групп объектов пачками
  - очередь модерации

PGQ

«+»

- **! транзакционность**
- минималный оверхед на выборку событий
- отсутствие блокировок
- удобный maintenance очередей

«-»

- нет отложенных событий – by design // отправь письмо ~~через 5 минут~~
- текстовый формат
- консумер переспрашивает источник по таймауту (1s), нет подписки на событие на сокете

Спасибо за внимание!

Михаил Тюрин
mtyurin@avito.ru