

Примеры архивирования и восстановления

Сергей Бурладян
sburladyan@avito.ru

backup

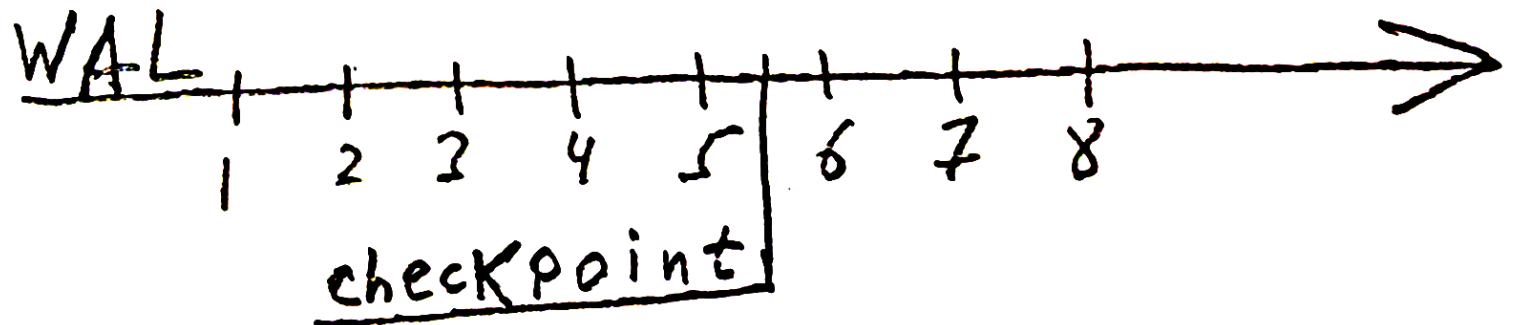
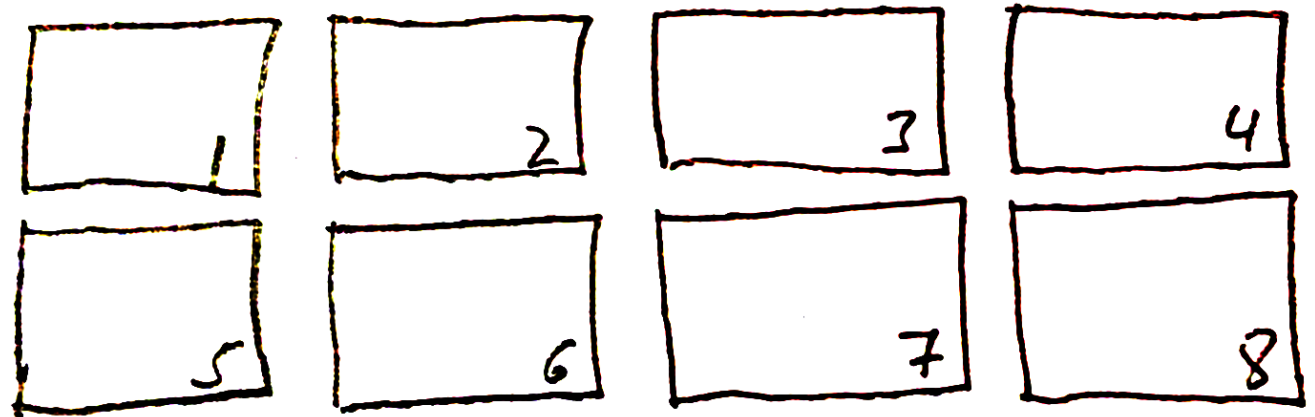
- pg_dump (repeatable read, serializable)
- data файлы
 - выключенный postgres (не интересно)
 - работающий postgres (PITR)

в процессе копирования куча меняется,
почему можно восстановить?

backup

- WAL (Write-Ahead Log)
- checkpoint
- pg_control

shared buffers



backup

- за время копирования файлов может пройти несколько checkpoint
- момент копирования pg_control случаен

если проигрывать WAL только с последнего checkpoint (redo) из pg_control то куча будет повреждена

backup

- backup_label

```
$ cat backup_label
```

```
START WAL LOCATION: 0/CB000020 (file 000000010000000000000000CB)
```

```
CHECKPOINT LOCATION: 0/CB000058
```

```
BACKUP METHOD: streamed
```

```
BACKUP FROM: master
```

```
START TIME: 2015-07-08 18:04:22 MSK
```

```
LABEL: pg_basebackup base backup
```

backup

- pg_basebackup
 - master: pg_start_backup() => backup_label
 - standby: ~~pg_start_backup()~~ => backup_label нет, как?
 - full_page_writes on на master
 - только через pg_basebackup
 - pg_control в последнюю очередь
 - minimum recovery ending location

backup

- резервная копия на NFS (через standby)
- NFS сервер аварийно перезагрузился, цел ли архив?
 - pg 9.2 нет checksum кучи
 - нет утилиты проверки (WAL через standby)
- два сервера для архива

backup

- pg_receivexlog

- pg 9.2 нет fsync

- не заметит:

- 0000000E0001F60C00000083

- 0000000E0001F60C00000084

- ~~0000000E0001F60C00000085~~

- 0000000E0001F60C00000086

- 0000000E0001F60C00000087

backup

- archive_command

```
test ! -f /mnt/server/archivedir/%f && cp %p  
/mnt/server/archivedir/%f
```

- fsync?
- NO
- dd conv=fsync

standby

- streaming (проблема «отставшего» архива)
- standby в бою, нужен ещё один
- archive_timeout не работает если нет транзакций

standby

- `archive_timeout` не работает если нет транзакций
- `select ping_wal()`
 `insert into ping_wal_table default values;`

pgq

- londiste

- два или больше consumer на один источник событий
londiste

```
$ pgqadm /etc/skytools/ticker.conf status
```

```
Consumer Lag LastSeen
```

```
-----
```

```
londiste.replika:
```

londiste_avito_new_1	1s	0s
londiste_avito_new_2	1s	0s

pgq

- londiste

- redo и undo

```
select ev_id, ev_time, ev_txid, ev_type from pgq.event_9_1
```

	ev_id bigint	ev_time timestamp with time zone	ev_txid bigint	ev_type text
1	3692218680	2015-02-04 17:59:35	308124210	I
2	3692218679	2015-02-04 17:59:35	308124219	I
3	3692218678	2015-02-04 17:59:35	308124082	I
4	3692218677	2015-02-04 17:59:35	308124206	I
5	3692218675	2015-02-04 17:59:35	308124207	I
6	3692218673	2015-02-04 17:59:35	308124208	I
7	3692218672	2015-02-04 17:59:35	308124188	I
8	3692218671	2015-02-04 17:59:35	308124200	I
9	3692218670	2015-02-04 17:59:35	308124202	I
10	3692218669	2015-02-04 17:59:35	308124205	I

pgq

- **londiste**

- **undo**

`subscriber undo [TICK_ID]` `rewind all tables with it UNDO log`

`subscriber add-undo-all` `enable UNDO log on all tables`

`subscriber remove-undo-all` `disable UNDO log on all tables`

`--enable-undolog replay: generate UNDO log`

pgq

- Iondiste

- триггер «нарезки»

```
execute 'insert into repca.items_c' || l_part || ' select ($1).* '  
using NEW;
```

```
declare
```

```
  _new repca.items_mv;
```

```
select NEW.item_id, NEW.cat_id into _new.item_id, _new.cat_id;  
return _new;
```

- долгое создание (сору с триггером)

pgq

- **londiste**

- быстрое создание копии реплики

```
1 pg_dump -Fc -h prod_host --serializable-deferrable dst_db \  
  | pg_restore -Fc -d dst_db
```

```
2 dst: select * from londiste.completed
```

```
3 src: select * from pgq.consumer where co_name = 'название londiste consumer'
```

```
4 src: select * from pgq.subscription where sub_consumer = 'id из (3)'
```

```
5 src: update pgq.subscription set  
      sub_last_tick = 'tick_id из (2)',  
      sub_batch = null, sub_next_tick = null  
where sub_consumer = 'id из (3)'
```


recovery

- `switch-master-db.sh`
 - stop standbys
 - promote
 - `londiste undo`
 - start standbys
 - `wait standbys timeline switch`
 - refresh DNS

recovery

- switch-master-db.sh
 - stop standbys
- ```
00000003.history
```
- |   |                            |
|---|----------------------------|
| 1 | 000000010000000000000000A0 |
| 2 | 000000020000000000000000F8 |
- londiste undo
    - асинхронная репликация

# recovery

- switch-master-db.sh

- wait standbys timeline switch

- master: 

```
select ('x' || left(
 pg_xlogfile_name(pg_current_xlog_location()),
 8)
)::bit(32)::int
```

- standby: 

```
psql -h "$host" -F ' ' -c 'IDENTIFY_SYSTEM' \
 'dbname=replication replication=true'
```

| systemid            | timeline | xlogpos    |
|---------------------|----------|------------|
| 6080348884119699418 | 1        | 1/4E0E42D0 |

Спасибо за внимание!