

Zkušenosti z nasazení PostgreSQL pohledem DBA

PostgreSQL je enterprise

- Výběr OpenSource alternativy k MS SQL, Oracle, Sybase a DB2
- MySQL, FireBird, PostgreSQL...
 - Licenční situace
 - ACID
 - Point In Time recovery
 - Funkčnost a podpora

Začněte malým systémem

- Projděte dokumentaci
- Sežeňte si poradce, nezačínejte sami
- Výpadek nesmí mít dopad na klienty
- Jídelníček, evidence parkovacích míst...
- Hýčkejte si vývojáře / aplikační podporu, budete potřebovat jejich pochopení a trpělivost

Obnova, zálohování, DR

- Než začnete, promýšlejte jak chcete zálohovat
- Až to budete mít vymyšlené, zkuste si provést obnovu
 - Opravdu tu obnovu udělejte :-)
- Ochrana proti havárii datového centra
 - Log shipping
 - Streaming replikace
 - Logická replikace
 - OS cluster

Stanovte si pravidla

- Prozkoumejte koncept/architekturu PostgreSQL
- Nekomplikujte si život specialitami, pokud je nepotřebujete
 - Výchozí `search_path="$user", public`, lze nastavit pro
 - Cluster
 - Databáze
 - Role (uživatel)
- Co chcete zapisovat do logu?
 - Připojení, ukončení session...?
 - DDL příkazy?
 - Čekání na zámky

Nepoužívejte schema PUBLIC

- Můžete jej smazat, nebojte se toho
 - Nahradíte tak řadu „systémových“ oprávnění
- I „krabicový SW“, napsaný pro schema public lze nastavením search_path
 - Alter role my_user SET search_path TO my_schema, your_schema;
 - SEARCH_PATH default:
 - "\$user", public
 - Pokud public neexistuje, můžete využít \$user nebo nastavit vlastní schema

Oddělte aplikace

- Pokud aplikace sdílejí databázi, použijte různá schemata a přiděľujte explicitně oprávnění k schematům (usage)
- Pokud to jde, použijte samostatné databáze umístěné v samostatných tablespaces
- Můžete použít i více instancí (PG cluster adresář) na jednom OS
- Vlastní instalace SW pro každou instanci ?
 - **Flexibilita** ve verzích/záplatách, více práce s údržbou

Používejte více svazků

- [WIKI: File System Layouts](#)
- Tablespaces umístěte na samostatné svazky
- pg_xlog – nastavení velikosti podle svazku
 - checkpoint_segments
 - Verze 9.5: min_wal_size (80MB), max_wal_size (1GB, soft limit), viz též wal_keep_segments
- Pokud selže archivace transakčního logu (WAL, xlog), naroste i nad limit daný [vzorcem](#) v dokumentaci (nebo nad max_wal_size)
- Ve svazku pro WAL logy vždy mějte „dummy“ soubor, který můžete smazat, bude se to hodit, až se svazek zaplní

Svazky pro Postgres cluster I

- MountPoint-y jsou v adresáři `cluster_name/` (od 9.5 konfigurační parametr)
 - Zpravidla nese název podle aplikace pro kterou slouží
- `pgarchive/`
 - rezerva pro případ nedostupnosti zálohovacího serveru (WAL archivujeme pomocí rsync na zálohovací server)
- `pgbackup/`
 - Standardně zálohujeme cluster na jiný server, ale velice se hodí při potřebě udělat dump...

Svazky pro Postgres cluster II

- pgcluster/data/
 - Adresář kam ukazuje proměnná PGDATA (data/ protože initdb nemá rád lost+found) zde pak symbolické odkazy
 - pg_log -> ../../pglog
 - pg_xlog -> ../../pgxlog
- pglog/ - svazek pro log soubory postgres instance
- pgxlog/ - svazek pro transakční (WAL) logy
- pgdata/ - adresář pro přípojně body svazků tablespaces
 - Případně svazek pro data, pokud nemá smysl oddělovat tablespaces do vlastních svazků (testovací prostředí...)

Svazky pro Postgres cluster III

- Dokumentace [Database File Layout](#)
- Používaná sada svazků

```
... /WEATHER/pgarchive  
... /WEATHER/pgbackup  
... /WEATHER/pgcluster  
... /WEATHER/pgdata/system  
... /WEATHER/pgdata/hydro  
... /WEATHER/pgdata/temp  
... /WEATHER/pgdata/wind  
... /WEATHER/pglog  
... /WEATHER/pgxlog
```

WAL logy jsou cyklický buffer, ale...

- Při vyšší zátěži se vytvářejí nové 16MB soubory
- Po poklesu zápisů do WAL logů dojde k snížení počtu WAL logů na $3 * \text{checkpoint_segments}$
 - V dokumentaci to samozřejmě je, jen jsme začali s `checkpoint_segments` nastaveným „`floor(velikost_svazku_v_MB/16)`“ to nebylo dobré...
- Osvědčila se velikost svazku
 - alespoň $4 * \text{checkpoint_segments} * 16\text{MB}$
 - Pro 9.5 `max_wal_size` + rezerva

Table / index bloat

- Opět se hodí „výplň“ určená ke smazání, jinak je třeba zvětšit svazek

```
ERROR: could not extend file "pg_tblspc/16405/PG_9.3_201306121/16406/16443.24":  
No space left on device  
HINT: Check free disk space.  
ERROR: could not extend file "pg_tblspc/16405/PG_9.3_201306121/16406/16443.24":  
wrote only 4096 of 8192 bytes at block 3252805  
HINT: Check free disk space.
```

- Najít velkou tabulku/index kde je hodně „mrtvých“ řádek
 - n_live_tup `\dt+ \d pg_stat_all_tables`
 - n_dead_tup
 - Řešení je na wiki – SQL dotaz z skriptu: [check_postgres](#)
- Rebuild tabulky
 - Exkluzivní zámeček, „kopíruje“ soubor tabulky – místo na disku(!) `vacuum FULL VERBOSE ANALYZE myapp_schema.bloated_table;`

Aktivnější autovacuum

```
./check_postgres.pl -db dynadb --action=bloat
POSTGRES_BLOAT CRITICAL: DB "dynadb" (db dynadb) table
dt_schema.measurement_high rows:851942592 pages:14401445
shouldbe:7923108 (1.8X) wasted size:53070536704 (49 GB) ...
```

- U tabulek s velkým počtem update, delete operací nemusí vyhovovat výchozí nastavení pro autovacuum
 - Úprava konfigurace

```
#log_autovacuum_min_duration = -1      # -1 disables, 0 logs all actions and
log_autovacuum_min_duration = 250     # -1 disables, 0 logs all actions and
                                        # their durations, > 0 logs only
                                        # actions running at least this number
                                        # of milliseconds.
#autovacuum_vacuum_cost_delay = 20ms   # default vacuum cost delay for
autovacuum_vacuum_cost_delay = 4ms     # default vacuum cost delay for
                                        # autovacuum, in milliseconds;
#maintenance_work_mem = 16MB          # min 1MB
maintenance_work_mem = 1GB            # min 1MB
```

Ještě aktivnější autovacuum

- Když předchozí úpravy nestačí

PID	USERNAME	PRI	NICE	SIZE	RES	STATE	TIME	WCPU	CPU	COMMAND
10288	postgres	20	0	17G	1056M	sleep	0:01	2.32%	27.64%	...autovacuum

- Můžeme nabídnout pomocnou ruku...

```
#vacuum_cost_limit = 200           # 1-10000 credits  
vacuum_cost_limit = 5000          # 1-10000 credits
```

PID	USERNAME	PRI	NICE	SIZE	RES	STATE	TIME	WCPU	CPU	COMMAND
20023	postgres	20	0	18G	13G	run	8:57	0.78%	45.75%	...autovacuum

PID	USERNAME	PRI	NICE	SIZE	RES	STATE	TIME	WCPU	CPU	COMMAND
20023	postgres	20	0	18G	13G	disk	9:07	10.66%	9.15%	...autovacuum

- Když je vacuum potřeba pro více tabulek současně

```
#autovacuum_max_workers = 3        # max number of autovacuum subprocesses  
                                   # (change requires restart)
```

Výměna autovacuum procesu

- Připravili jsme parametry, ale hodiny běžící autovacuum proces není změnou ovlivněn – přeručíme autovacuum

```
vacuum verbose analyze my_schema.my_table;
```

```
56b07515.cb8f VACUUM waiting :LOG: sending cancel to blocking autovacuum PID 27106
56b07515.cb8f VACUUM waiting :DETAIL: Process 52111 waits for shareUpdateExclusiveLock...
56b07515.cb8f VACUUM waiting :STATEMENT: vacuum verbose analyze MY_TABLE;
56b07515.cb8f VACUUM waiting :LOG: process 52111 still waiting for shareUpdateExclusiveLock...
56b07515.cb8f VACUUM waiting :STATEMENT: vacuum verbose analyze MY_TABLE;
```

```
56ad600e.69e2 :ERROR: canceling autovacuum task
56ad600e.69e2 :CONTEXT: automatic vacuum of table "MY_TABLE"
```

```
56b07515.cb8f VACUUM waiting :LOG: process 52111 acquired ShareUpdateExclusiveLock...
56b07515.cb8f VACUUM waiting :STATEMENT: vacuum verbose analyze MY_TABLE;
56b07515.cb8f VACUUM :ERROR: canceling statement due to user request
56b07515.cb8f VACUUM :STATEMENT: vacuum verbose analyze MY_TABLE;
```

```
56b09b39.3e59 :LOG: skipping vacuum of "MY_TABLE" --- lock not available
```

```
56b09bb1.4e37 :LOG: automatic vacuum of table "MY_TABLE": index scans: 2
pages: 0 removed, 14401445 remain
tuples: 18639216 removed, 785360347 remain
buffer usage: 20751057 hits, 29081151 misses, 3893231 dirtied
avg read rate: 109.987 MiB/s, avg write rate: 14.725 MiB/s
system usage: CPU 154.05s/442.89u sec elapsed 2065.66 sec
```


Default privileges... jupí!

- Nastavit výchozí oprávnění při vhodném návrhu šetří práci do budoucna

```
ALTER DEFAULT PRIVILEGES
    [ FOR { ROLE | USER } target_role [, ...] ]
    [ IN SCHEMA schema_name [, ...] ]
    abbreviated_grant_or_revoke
```

- Jednorázově pomocí makra „ALL <OBJECT>“
 - Pro instalační skripty...
 - Uplatní se na objekty existující v okamžiku vydání příkazu

```
GRANT { { SELECT | INSERT ... }
        [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
        | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO role_specification [, ...] [ WITH GRANT OPTION ]
```

Archivní režim

- `archive_mode` vždy nastaven na `on`
- `wal_level` vždy nastaven alespoň na `archive`
- `archive_command` vždy nastaven
- Co dokáže zapomenutý parametr u `rsync WAL` logu v `archive_command`...

```
2016-01-26 02:24:03 CET postgres 56a6cab3.e7c:LOG: connection authorized: ...
```

```
WARNING: Unauthorized access to this system is forbidden!  
Violation of this restriction may constitute a legal responsibility  
according to the criminal and civil law of the Czech Republic.
```

```
VAROVANI: Zakaz neopravneneho pristupu do tohoto systemu!  
Porusení tohoto zakazu muze založit trestnepravni  
a občanskopravni odpovědnost.
```

Nastavení... která nám vyhovují I/III

- `shared_buffers = 2-16GB # podle cache hits`
- `work_mem = 64MB # podle RAM a počtu sessions`
- `maintenance_work_mem = 1GB # doposud nám stačí 3 workery`
- `shared_preload_libraries = 'pg_stat_statements'`
- `wal_level = archive`
- `checkpoint_segments = <vzorec> # započítat místo na výplň`
- `checkpoint_completion_target = 0.75 # rovnoměrné zápisy`
- `archive_mode = on # změna vyžaduje restart`
- `##testy`
 - `archive_command = '/bin/true' # lze měnit online`
- `##produkce`
 - `archive_command = 'rsync -a -q -e \"ssh -q\" %p ...%f'`

Nastavení... která nám vyhovují II/III

- `log_destination = 'stderr'`
- `logging_collector = on`
- `log_directory = 'pg_log'`
- `log_filename = 'postgresql-%d.log'`
- `log_truncate_on_rotation = on`
- `log_rotation_age = 1d`
- `log_rotation_size = 0`
- `log_checkpoints = on`
- `log_connections = on # audit?`
- `log_disconnections = on # audit?`
- `log_line_prefix = '%t %c %l %r %d %u %x %v %i:' # audit?`

Nastavení... která nám vyhovují III/III

- `log_lock_waits = on # log lock waits >= deadlock_timeout`
- `log_statement = 'ddl' # none, ddl, mod, all`
- `log_temp_files = 2048 # log temporary files equal or larger`
- `log_timezone = 'Europe/Prague'`
- `track_counts = on # default, ale raději uvádíme`
- `track_io_timing = on # VŽDY ověřit pg_test_timing`
- `autovacuum = on # default, ale raději uvádíme`
- `log_autovacuum_min_duration = 250`
- `default_tablespace = 'system' # Běžný uživatel nemá práva`
- `temp_tablespace = 'temp' # na vlastním svazku`

Záloha a DR

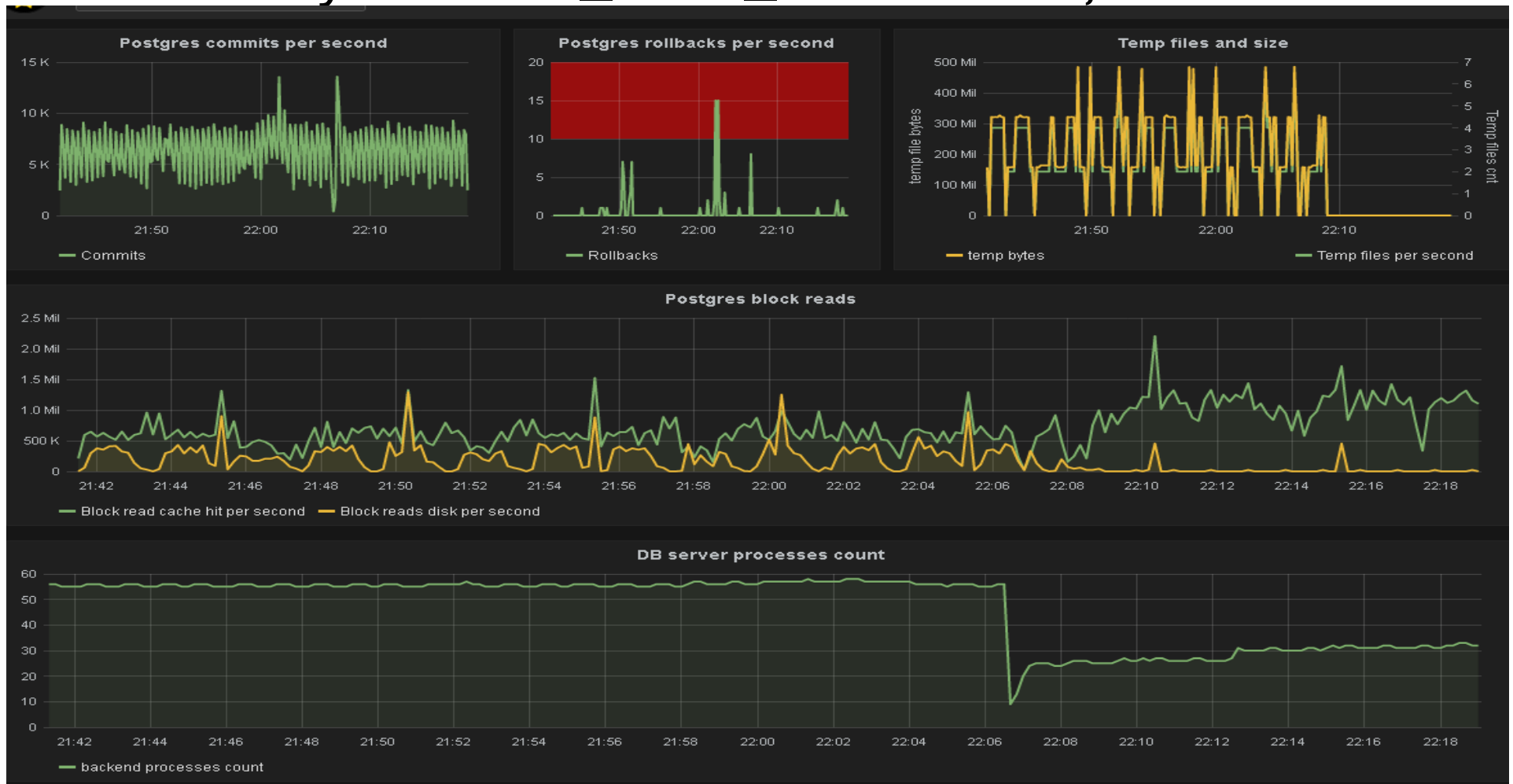
- Používáme BARMAN
- PgBackRest je určitě zajímavá alternativa
- DR řešíme RedHat PaceMaker clusterem
- Streaming replikace (se zpožděnou aplikací WAL záznamů) je jistě dobrá volba
- Testujte DR scénáře
 - Jeden úspěšný (!) test trval 6 hodin, než se povedlo obnovit provoz a to se vlastně nic nestalo...

Monitoring a connection pooling

- Monitoring agent Telegraf obsahuje plugin pro Postgres (telegraf → Influx → Grafana), ale opakovaně se přihlašuje...
 - Nastavené logování connect/disconnect je pro telegraf nepraktické
 - PgBouncer
 - Pro telegraf je potřeba nastavit `ignore_startup_parameters = extra_float_digits`
- Jboss (a další apl. servery) obsahují connection pooling
 - Obvykle rychlý restart instance (pro načtení statických parameterů) aplikacím nevadí – je to ale třeba **otestovat!**

Telegraf, influx, Grafana

- select * from system.pg_stat_statements
order by shared_blks_read desc;



Dotazy?

Nebo pár ukázek co jsme pokazili a co jsme se naučili?

Cluster v jednom svazku

- Taky jsme si to zkusili (pro vývoj možná?)

```
pgbench=# insert into x select * from y;  
PANIC: could not write to file "pg_xlog/xlogtemp.30107": No space left on device  
server closed the connection unexpectedly  
        This probably means the server terminated abnormally  
        before or while processing the request.  
The connection to the server was lost. Attempting reset: Failed.  
!>
```

- V logu je pak další „stížnost“

```
LOG: archive command failed with exit code 1  
DETAIL: The failed archive command was: /bin/false  
WARNING: archiving transaction log file "000000010000000000000003E" failed too many times, will try again later  
...  
LOG: terminating any other active server processes  
WARNING: terminating connection because of crash of another server process  
DETAIL: The postmaster has commanded this server process to roll back the current transaction and exit, because  
another server process exited abnormally and possibly corrupted shared memory.  
LOG: all server processes terminated; reinitializing  
LOG: database system was interrupted; last known up at 2016-01-27 16:50:57 CET  
LOG: database system was not properly shut down; automatic recovery in progress  
LOG: redo starts at 0/43000028  
LOG: redo done at 0/470096E0  
LOG: last completed transaction was at log time 2016-01-27 16:51:08.7932+01  
PANIC: could not write to file "pg_xlog/xlogtemp.4141": No space left on device  
LOG: startup process (PID 4141) was terminated by signal 6: Aborted  
LOG: aborting startup due to startup process failure
```

Nemažeme WAL logy

- Dokud nebyly archivovány

```
postgres@tdbatst1:pg_xlog$ ll
total 1048584
drwx----- 2 postgres dba          4096 Jan 28 08:34 archive_status
-rw-r--r-- 1 postgres dba 1073741824 Jan 27 16:22 space_filler
```

```
LOG:  database system shutdown was interrupted; last known up at 2016-01-28...
LOG:  invalid primary checkpoint record
LOG:  invalid secondary checkpoint record
PANIC: could not locate a valid checkpoint record
LOG:  startup process (PID 4099) was terminated by signal 6: Aborted
LOG:  aborting startup due to startup process failure
```

- Pokud místo logu smažeme výplň `space_filler`

```
LOG:  database system shutdown was interrupted; last known up at 2016-01-28 ...
LOG:  database system was not properly shut down; automatic recovery in progress
LOG:  redo starts at 0/48000090
LOG:  redo done at 0/48000090
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
```

Uvolní vacuum full místo ve svazku?

„Session1“

„session2“

„session3“

Umístění shora dolů odpovídá pořadí vykonávání příkazů.
AutoVacuum je vypnuté, jinak by se bez dostatečného provozu testcase nepovedl.

```
postgres=# show autovacuum;
 autovacuum
-----
 off
(1 row)

eomdb=# select pg_relation_filepath('eom_schema.y');
          pg_relation_filepath
-----
 pg_tblspc/16387/PG_9.4_201409291/16388/16427
(1 row)

eomdb=# insert into eom_schema.y select * from pg_class;
INSERT 0 298
```

```
eomdb=# begin;
BEGIN
eomdb=# select count(*) from eom_schema.y;
 count
-----
    298
(1 row)
```

```
eomdb=# insert into eom_schema.y select * from pg_class;
...
```

```
postgres@tedb02:16388$ du -sh 16427*
64K      16427
24K      16427_fsm
```

```
eomdb=# begin;
BEGIN
eomdb=# select count(*) from eom_schema.y where relname like 'pg%';
 count
-----
    924
(1 row)

eomdb=# select count(*) from eom_schema.y;
 count
-----
   1192
(1 row)
```

Procesy používající soubor tabulky

```
eomdb=# delete from eom_schema.y where relname like 'pg%';  
DELETE 924
```

```
eomdb=# select n_live_tup, n_dead_tup from pg_stat_all_tables where  
eomdb=#  schemaname='eom_schema' and relname = 'y';  
 n_live_tup | n_dead_tup  
-----+-----  
          1192 |      8486653  
(1 row)
```

```
postgres@tedb02:16388$ lsof 16427*
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
postmaste	38176	postgres	33u	REG	253,14	237568	131079	16427
postmaste	38176	postgres	34u	REG	253,14	24576	131322	16427_fsm
postmaste	56077	postgres	22u	REG	253,14	237568	131079	16427
postmaste	56438	postgres	60u	REG	253,14	237568	131079	16427

```
postgres@tedb02:16388$ du -sh 16427*
```

```
232K    16427  
24K     16427_fsm
```

```
eomdb=# end;  
COMMIT
```

```
eomdb=# commit;  
COMMIT
```

```
eomdb=# vacuum full verbose eom_schema.y;  
INFO:  vacuuming "eom_schema.y"  
INFO:  "y": found 0 removable, 1192 nonremovable row versions in 29 pages  
DETAIL:  924 dead row versions cannot be removed yet.  
CPU 0.00s/0.00u sec elapsed 0.01 sec.  
VACUUM
```

```
eomdb=# select pg_relation_filepath('eom_schema.y');  
           pg_relation_filepath  
-----  
pg_tblspc/16387/Pg_9.4_201409291/16388/16433  
(1 row)
```

Zjistíme procesy bránící uvolnění místa na svazku

```
postgres@tedb02:16388$ lsof 16427*
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
postmaster	38176	postgres	33u	REG	253,14	237568	131079	16427
postmaster	38176	postgres	34u	REG	253,14	24576	131322	16427_fsm
postmaster	56077	postgres	22u	REG	253,14	237568	131079	16427
postmaster	56438	postgres	60u	REG	253,14	237568	131079	16427

```
postgres@tedb02:16388$ lsof -p 56077 | grep -e 22u -2 -e DEVICE
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
postmaster	56077	postgres	22u	REG	253,14	0	131079	
/srv/app/dev/EOM/pgdata/eom/Pg_9.4_201409291/16388/16427 (deleted)								

Pokud známe předem PID procesů, které danou tabulku/index používaly, můžeme požádat uživatele, aby se odhlásili, nebo backend procesy zastavit a uvolnit tak místo. Na aktivní DB s connection poolom jsme museli po 15ti minutách restartovat instanci...

```
eomdb=# select pg_backend_pid();
pg_backend_pid
-----
          56077
(1 row)
```


Používané hračky

- Psql
- PgAdmin
- pg_top
- htop
- iotop
- Perf
 - Instalujeme debuginfo balíček, aby perf byl přehledný

perf + debuginfo

```
root@tdbatst1:~  
Samples: 246K of event 'cpu-clock', Event count (approx.): 21233172673  
9.19% [kernel] [k] _spin_unlock_irqrestore  
3.94% [kernel] [k] finish_task_switch  
2.64% postgres [.] base_yyparse  
2.44% [kernel] [k] audit_filter_syscall  
2.31% postgres [.] AllocSetAlloc  
2.01% postgres [.] SearchCatCache  
1.76% [kernel] [k] __do_softirq  
1.70% postgres [.] hash_search_with_hash_value  
1.40% libc-2.12.so [.] __strlen_sse42  
1.20% libc-2.12.so [.] __strcmp_sse42  
0.98% libc-2.12.so [.] _int_malloc  
0.93% [kernel] [k] scsi_request_fn  
0.75% postgres [.] XLogInsert  
0.75% postgres [.] expression_tree_walker  
0.74% libc-2.12.so [.] memcpy  
0.69% postgres [.] core_yylex  
0.61% postgres [.] LWLockAcquire  
0.61% postgres [.] palloc  
0.61% libc-2.12.so [.] malloc  
0.57% postgres [.] MemoryContextAllocZeroAligned  
0.52% postgres [.] ScanKeywordLookup  
0.50% pgbench [.] threadRun  
Press '?' for help on key bindings
```

pg_top, iotop, htop

```
postgres@pedb01:~/pg_top
last pid: 48503; load avg: 2.10, 2.38, 2.33; up 32+08:00:49 15:26:06
72 processes: 1 running, 71 sleeping
CPU states: 2.6% user, 0.0% nice, 0.9% system, 96.5% idle, 0.0% iowait
Memory: 276 used, 2256 free, 1846M buffers, 216 cached
DB activity: 83 tps, 0 rollbs/s, 0 buffer r/s, 100 hit%, 5195 row r/s, 77 row w/s
DB I/O: 0 reads/s, 0 KB/s, 23 writes/s, 74 KB/s
DB disk: 1.0 GB total, 0.8 GB free (17% used)
Swap: 4096M free

PID USERNAME PRI NICE SIZE RES STATE TIME WCPU CPU COMMAND
48001 postgres 20 0 8597M 427M run 0:03 6.02% 71.83% select fix_disabled_agenda()
28239 postgres 20 0 8592M 798M sleep 2:36 0.17% 0.40% COMMIT
46862 postgres 20 0 8591M 79M sleep 1:43 0.17% 0.40% COMMIT
9356 postgres 20 0 8591M 78M sleep 1:35 0.21% 0.40% select * from ACT_RU_VARIABLE? where EXECUTION_ID_ = $1? and TASK_I
49118 postgres 20 0 8591M 77M sleep 1:31 0.20% 0.40% COMMIT
46859 postgres 20 0 8596M 1761M sleep 9:16 0.52% 0.20% COMMIT
42659 postgres 20 0 8597M 1742M sleep 9:02 0.27% 0.20% SET SESSION CHARACTERISTICS AS TRANSACTION READ ONLY
9295 postgres 20 0 8638M 1781M sleep 8:38 0.48% 0.20% COMMIT
39985 postgres 20 0 8642M 1799M sleep 8:35 0.30% 0.20% COMMIT
53435 postgres 20 0 8592M 803M sleep 2:50 0.18% 0.20% COMMIT
51244 postgres 20 0 8592M 804M sleep 2:49 0.16% 0.20% COMMIT
48050 postgres 20 0 8593M 805M sleep 2:47 0.22% 0.20% select parameters0_.transfer_id as transfer3_34_0_, parameters0_.name as
45062 postgres 20 0 8592M 782M sleep 2:16 0.17% 0.20% SET SESSION CHARACTERISTICS AS TRANSACTION READ WRITE
39438 postgres 20 0 8591M 47M sleep 2:15 0.12% 0.20% COMMIT
39985 postgres 20 0 8642M 1799M sleep 8:35 0.95% 0.00% postgres: csops_owner csopsdb 10.177.16.112(40458) idle
46758 postgres 20 0 8635M 1783M sleep 8:17 1.69% 0.00% postgres: csops_owner csopsdb 10.177.16.112(40490) idle
49874 postgres 20 0 8592M 811M sleep 2:51 0.20% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53360) idle
51244 postgres 20 0 8592M 804M sleep 2:49 0.16% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53364) idle
55189 postgres 20 0 8591M 46M sleep 2:44 0.09% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53378) idle in transaction
55544 postgres 20 0 8591M 46M sleep 2:44 0.09% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53381) idle in transaction
67027 postgres 20 0 8592M 47M sleep 2:43 0.10% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53671) idle
69877 postgres 20 0 8591M 46M sleep 2:42 0.09% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53361) idle in transaction
7028 postgres 20 0 8591M 46M sleep 2:42 0.10% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53672) idle
88965 postgres 20 0 8592M 800M sleep 2:41 0.23% 0.00% postgres: csops_owner csopsdb 10.177.16.120(55225) idle
17631 postgres 20 0 8591M 46M sleep 2:41 0.09% 0.00% postgres: csops_owner csopsdb 10.177.16.120(53333) idle
28239 postgres 20 0 8592M 797M sleep 2:36 0.18% 0.00% postgres: csops_owner csopsdb 10.177.16.120(57387) idle
6540 postgres 20 0 8592M 798M sleep 2:33 0.18% 0.00% postgres: csops_owner csopsdb 10.177.16.120(57125) idle
```

pg_test_timing, pg_test_fsync

- pg_test_fsync pro nalezení optimální wal_sync_method
- track_io_timing – na tomto serveru lze použít bez obav

```
postgres@pgdb:pglog$ /usr/pgsql-9.4/bin/pg_test_timing
Testing timing overhead for 3 seconds.
Per loop time including overhead: 36.28 nsec
Histogram of timing durations:
< usec    % of total    count
   1      96.39480   79711777
   2       3.60221   2978778
   4       0.00173     1427
   8       0.00013      110
  16       0.00109     903
  32       0.00001       9
  64       0.00000       1
 128       0.00000       0
 256       0.00000       0
 512       0.00003      25
```

Oběd !