PostgreSQL v prostředí rozsáhlých IPTV platforem

Kdo jsme?

- Vyvíjíme a provozujeme IPTV a OTT řešení
- Klíčový zákazníci
  - O2 (O2TV, iVysílání České Televize)
  - Orange
  - T-Mobile
  - Swisscom Broadcast
- End-to-end multiscreen řešení vvvinuté in-house
  - STB, Mobile, PC, SmartTV
  - Streamers, Video storage
  - Media aquisition
  - Middleware

PostgreSQL v nangu.TV

- Hlavní databázová technologie
- Používáme od prvních verzí platformy, začátky na PostgreSQL 8.0
- 60 provozních instancí PostgreSQL
- Většina PostgreSQL 9.4, částečně ještě PostgreSQL 9.2
- Připravujeme se na přechod na PostgreSQL 9.6
- Middleware – billing, provisioning, reporting
- Backend – informace o uložení dílčích video chunků, metadata, video objekty
- Keystore – uložení klíčů pro DRM
- Oauth – centrální SSO, repozitář aplikací, tokeny

Dopady růstu zákazníků na DB

- Původní zákazníci cca 20 tisíc uživatelů, jedno datové centrum, databáze běží na serverech společně s aplikacemi, stovky TPS
- Nyní zákaznické báze o velikosti stovek tisíc uživatelů
- Dedikované databázové servery pro každou z aplikací
- 20,000 transakcí za sekundu ve špičkách
- Write heavy - aplikace 1,2TB zápisů denně
- Vysoká citlivost na latence
- Clustery s geografickou redundancí
- Plány na 1,000,000 uživatelů v blízké budoucnosti
- Potřeba škálovat do šířky

Evoluce DB stacku

- Přechod na dedikované DB servery, každá aplikace má samostatný cluster
- RAID1 → RAID10 → enterprise SSD
- Upgrade a optimalizace DRBD
- Přechod od DRBD ke asynchronní streaming replikaci
- Přechod na PostgreSQL 9.4
- Samostatné repliky pro analytické dotazy
- Zavedení centrálního connection poolingu

Optimalizace aplikací

- Optimalizace klíčových dotazů (low-hanging fruit)
- Zavedení lepšího cachování na všech přístupových úrovních
- Odsun netransakčních dat do vhodnějších úložišť
  - Cassandra – úložiště aplikačních logů, obrázků
  - Elasticsearch – search API
  - RabbitMQ – Messaging
  - Syslog – logování STB boxů
- Partitioning dat
- Zavedení connection poolů pro aplikace
- Lepší využití prepared statementů
- Využití partial indexů

Problémy spojené s růstem báze

- Problémy s lock contention (spin locks)
  - Patch „Use SnapshotDirty rather than an active snapshot to probe index endpoints.“ pro 9.2
  - „FOR KEY SHARE and FOR NO KEY UPDATE“ v 9.3
  - Další optimalizace v 9.4, 9.5
- Latence způsobené Extension locks
- Problémy s IO spiky check pointu
- Problémy se statistikami
- Komplikovaná údržba DB

Na cestě k 1,000,000 uživatelů

- Rozšíření používání partitioningu
- Implementace shardingu
- Rozdělení monolitických aplikací na menší samostatné servisy
- Model Eventual consistency
- Analytické dotazy směřovány na samostatnou repliku
- Ochrana proti Thundering herd

Hardware

- X10DRH-CLN4
- 2x Intel E5-2650v4
- 256 GB RAM
- 2x 300GB C10K900 SAS – OS a logy
- Intel SSD DC P3700 Series (PCI-E, NVMe)
- (optional) 2x 300GB C10K900 SAS – WAL

Konfigurace PostgreSQL

max_connections = 120
shared_buffers = 5GB
temp_buffers = 128MB
work_mem = 12MB (22MB)
maintenance_work_mem = 6400MB
effective_io_concurency = 32
checkpoint_segments = 256
checkpoint_timeout = 15min
checkpoint_completion_target = 0.9
hot_standby_feedback = off
stats_temp_directory = '/mnt/pg_stat_tmp'
synchronous_commit = on
autovacuum_max_workers = 6
autovacuum_naptime = 6

random_page_cost = 1.5
effective_cache_size = 200GB
geqo_threshold = 20
log_min_duration = 500
log_checkpoints = on
log_line_prefix = '%t:%r:%u@%d:[%p]: '
log_lock_waits = on
log_temp_files = 0
track_activity_query_size = 10240
deadlock_timeout = 2s
restart_after_crash = off

Konfigurace OS

- Debian GNU/Linux Jessie, Kernel 3.16
- Ext4 s noatime
- Sysctl
  - vm.dirty_background_ratio = 5
  - vm.dirty_ratio = 10
  - vm.swapiness = 1
  - net.ipv4.tcp_keepalive_time=300
  - net.ipv4.tcp_keepalive_intvl=60
  - net.ipv4.tcp_keepalive_probes=10

Vysoká dostupnost

- Původní řešení nad Pacemaker/Corosync
- DRBD resource – síťový RAID1
- Filesystem resource
- Plovoucí IP adresa
- LSB PostgreSQL
- Nevýhody:
  - Dlouhý čas failoveru (zejména při nekorektním ukončení primary node)
  - DRBD nad SSD disky výrazně degraduje výkon (značné zlepšení DRBD 8.4.3)
  - Pro více jak 2 nody příliš komplikovaná konfigurace
  - Cluster musí být v jednom L2 segmentu

Pacemaker se Streaming replikací

- Odpadá overhead DRBD
- Rychlejší failover
- Bezproblémové zapojení více jak 2 nodů
- Pacemaker pgsql resource (nebo PAF)
- Slave nodes lze využít v hot standby režimu pro offload části čtecích dotazů
- Stále nutnost provozu na společném L2 segmentu
- Problémy při vyšších latencích mezi nody
- Situace s balíčkováním v Debianu není ideální

Patroni a HAProxy

- Patroni – bootstrap, monitoring, failover
- Consul – quorum, distributed configuration store
- HAProxy – monitoruje stav jednotlivých nodes přes patroni, směřuje provoz na aktuální master node
- Tolerantní vůči vyšším latencím mezi nody
- Není závislé na jedom L2 segmentu a společné VIP adrese

PgPool-II slepá cesta

- Vysoké HW nároky (ve srovnání s pg bouncerem)
- Nepodporuje transaction/statement pooling
- Značná degradace výkonu při použití JDBC
- Nepodporuje multistatement queries
- Loadbalancing aplikace využívající JDBC/Hibernate přinášela řadu problémů
- Těžkopádná konfigurace

Connection pooling - pgbouncer

- Pro backend connection pool na DB serverech či dedikovaných poolerech
- Původních 400 connections zredukováno na 50
- Použití v transaction režimu
- Snížení zátěže DB o 1/5
- Minimální paměťové a CPU nároky
- Možnost aktivace rezervního poolu při překročení nastaveného času odezvy
- Velmi dobrá dokumentace a tooling

Údržba - Autovacuum

- Pro malé a statické tabulky defaultní nastavení
- Pro často modifikované tabulky nastavení per tabulka

```
alter table T set (autovacuum_vacuum_cost_delay = 10);
alter table T set (autovacuum_vacuum_cost_limit=1000);
alter table T set (autovacuum_vacuum_scale_factor=0.02);
alter table T set (autovacuum_analyze_scale_factor=0.01);
```

- Stále je však nutné příležitostné provedení VACUUM FULL
- **Pro často modifikované tabulky je finálním řešením partitioning**

Údržba - reindexace

- Od 9.2 podpora CREATE INDEX Concurrently, nad 9.2 ale problémy se zámky na extrémně často modifikovanými tabulkami
- Od 9.4 již bez problémů možné provádět za plného provozu
- Je třeba mít dostatek prostoru pro další kopii indexu

```
create index CONCURRENTLY tmp_tabulka_klic on chunk using btree (chunk_day);
```

```
begin; drop index tabulka_klic; alter index tmp_tabulka_klic rename to tmp_tabulka; commit;
```

Monitoring - Grafování

- Na DB serverech collectd pro sběr dat
- Data přeposílání na centrální servery kde jsou uloženy v RRD
- Jednoduchý frontend Collection3/Cacti
- Frontend je pomalý, z povahy RRD data postupně méně přesná
- Postupný přechod na InfluxDB/Grafana
- Zvažujeme Prometheus

Monitoring – Grafované metriky

- Počet commitů a rollbacků za sekundu
- Pro klíčové tabulky počet zvakouvaných řádek za sekundu
- Statistika bg writeru
- Utilizace autovacuum workers
- Počet connections
- Zpoždění streaming replikace
- Velikost klíčových tabulek
- Celková disk usage
- Disk IOPS, Latency, Traffic, percent utilization
- Síťový bandwitdh
- Load, CPU, Swap, vmstat

Vyhodnocení logů

- Analyzátor logů pgBadger
- Vyhodnocení na týdenní bázi, při řešení incidentů
- Častější kontroly po upgradech DB, aplikačního SW
- Nejčastější Slow queries
- Nejdelší individuální slow queries
- Zámky
- Checkpointy
- Temp files
- Centralizované logování přes ELK

Alerting

- Přecházíme na Icingu s distribuovanou architekturou
- Aktuálně většina instalací nad Nagios3
- Několik metod alertingu – SMS, E-mail, Jabber konference, Kibana dashboard
- Historie uložená v Elasticsearch, frontend Kibana
- Většinu sledovaných metrik přes check_posgres

Alerting - metriky

- Idle in transactions
- Last vacuum klíčových tabulek
- Last analyze klíčových tabulek
- Velikost tabulek
- Příliš dlouhé queries
- Počet zámků
- Počet konexí / backendů
- Table bloat
- Index bloat
- Stav replikace
- Replikační zpoždění
- Maximální čekání v pgbounceru
- Počet čekajících klientů v pgbounceru

Zálohování

- Základní metoda pg_dump
  - Zálohy v custom formátu na denní bázi
  - Throttling - nice, ionice, bwlimit
  - Přes rsync na dedikovaný server
- Pro velké platformy používáme PG Barman
  - Kontinuální záloha přes WAL archiving
  - PITR až na úrovni konkrétního XID
  - Zálohování po síťi či na dedikovanou SAN
  - Monitoring stavu zálohování přes Nagios/Icinga
  - Připravujeme přechod na 2.0 a zálohování pře Streaming replikaci
- Nepodceňovat pravidelné testování záloh

Upgrade - Scénář

- Obvyklá doba integrace nové verze PostgreSQL – 6 měsíců
- Zajištění kompatibility s novou verzí, výměna JDBC driverů, příprava balíčků s pluginy
- Testovací provoz v interním QA – funkční a integrační testy
- Zátěžové testy
- Nasazení na staging platformy zákazníků
- Nasazení na menší produkční platformy
- Nasazení na klíčové produkční platformy

Upgrade - Zátěžové testy

- Jako zdroj dat použity provozní data velkých platforem
- Identické HW prostředí jako v produkci
- Stav databáze převzat přímo z produkce (data z barmana), nikoliv čistý dump
- Custom aplikace na přehrávání aplikačních logů
- Přehrávání provozu zachyceného pgsharkem
- Zrychlené přehrání provozu pro simulaci zvýšené záťěže
- Pro některé aplikace syntetické scénáře pro Gattling
- Vyhodnocujeme pg_stat_statements, pg_buffercache, slow queries, locks, temp files, errors
- Vyhodnocují se aplikační latence a chyby

Upgrade – Produkční platformy

- Simulace upgrade v lokálním labu
- Získání dat produkce z barmana (nikoliv čistý dump/restore)
- Stejná verze, konfigurace a prostředí jako na produkci
- Sleduje se celkový čas, obsazené místo v průběhu i po upgrade, chyby
- Záloha pro případný rollback a rollback scénář
- Na produkci následně upgrade přes pg_upgrade s využitím hardlinků
- Nasazení optimalice konfigurace podle výsledků záťěžových testů
- Spuštění analyze nad všemi tabulkami
- Aktuálně testujeme bezvýpadkové upgrady přes pg_logical

Otázky

- Otázky ?

We are hiring!

Roman Fišer
2^{nd} level Support Team Leader

roman.fiser@nangu.tv
nangu.TV, a.s. | U Plynarny 1002/97 | 101 00 Praha 10 | Czech Republic | www.nangu.tv