POSTGRESQL @SKYPE

The Untold

POSTGRESQL @ SKYPE

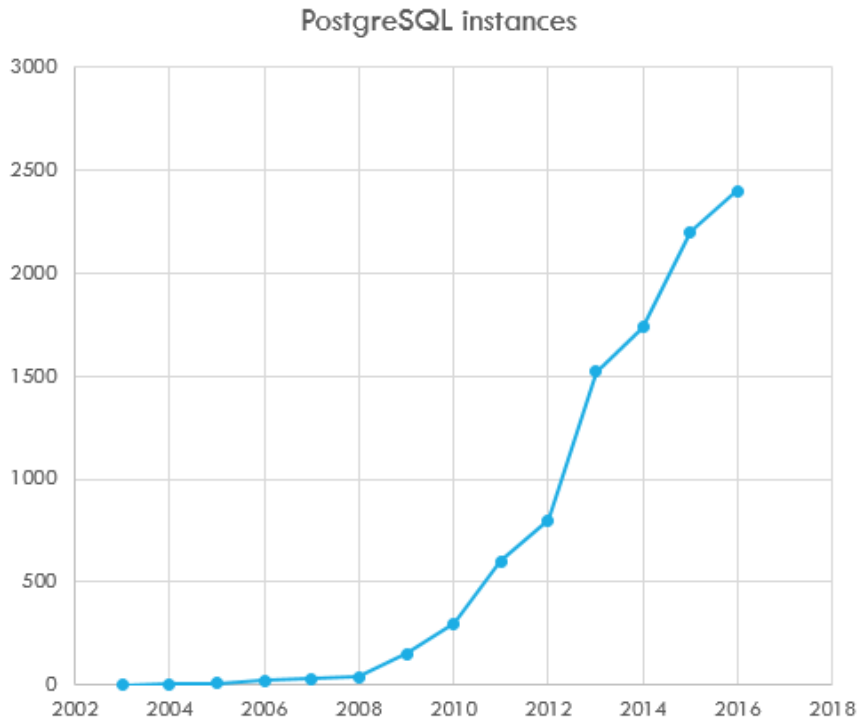
Distributed

Large scale

24x7

Automated

Platform as a Service



SENIOR DBA'S

Every hour there's a huge IO spike on databases

Senior DBA's are puzzling over this

Nothing comes to mind ...

Newbie comes in and makes a suggestion

Everyone laughs :D

Turns out ext3 is not that good for large files

LOCK QUEUE

Run a long transaction. Can be just a SELECT. For example pg_dump on a large database.

Now try to obtain an AccessExclusive on the table.

Run another query against the same table?

Should be fine, right?

lock_timeout helps!

INPUT VALIDATION

This is an regular expression to validate RFC-822 compliant email addresses.

And yes, you can implement this in a PostgreSQL stored procedure ☺

CONNECTING TO DATABASES

iptables based load balancer

DNS addressing

Short TTL

Reconnect on failures

IPv6 and DNS round-robin

Packet size limitations

SPLIT BRAIN

Accidentally enable both primary and failover as active. DNS happily passes queries to both.

Spend several weeks reconciliating the differences. In the end, compensate with Skype credit.

londiste3 enables writes only on the primary (y)

SNOWBALLING YOURSELF

Maintenance on one of the central storage arrays.

Increased latency for simple operations. Response time from 10ms to 20ms. What's the big deal?

Ok. How about going from 1000 rps to 500?

Double the number of concurrent queries to get back to 1000 rps.

Ended up taking down most of core databases.

SCALING READS

Scale out read operations to replica databases.

Naturally, all writes go to master.

Do not bother to check for replication lag.

Profit ☺

CLEANING UP

Need to remove 200 billion rows from the database to reclaim space.

Takes 2 months if minding the replication.

Fortunately there's **session_replication_role.**

Delete first on the standby, then primary. Done in 2 weeks. Profit!

Discover bunch of discrepancies ☹

Updates on the master were not replicated, because nothing to update on the replica.

DEALING WITH BLOAT

Need to reclaim the dead space left by large DML operation

Options?

CTAS + deltas, then rename

Will this work?

```
BEGIN;
ALTER TABLE t RENAME TO t_old;
ALTER TABLE rebuilt_t RENAME TO t;
END;
```

Replication helps

UNBALANCED CLUSTERS

Shard by *hashtext(SkypeId) % n*

Assume that data will be uniformly distributed

The reality often disagrees ☺

Can easily end up with hot spots within the cluster

CATCH ME IF I CAN

Two different versions of Skype client

Two different understandings of contact list

Each overwrites the contactlist to reflect reality

Endless loop follows

http://www.pidjin.net

ALL YOUR BASE ARE BELONG TO US

PUBLIC grants to procedures

SECURITY DEFINER functions as superuser

EXECUTE within stored procedure

SELECT f('t; CREATE LANGUAGE …')

MD5 authentication

POSTGRESQL BUGS?

A few were encountered:

- Cached plans invalidation
- SSL renegotiation dropping connections
- Unsafe CLUSTER
- VACUUM not keeping up
- etc. etc.

However, most times we just shoot ourselves in the foot!

CLOSING THOUGHTS

PostgreSQL rocks!

Questions? mpihlak || martin.pihlak@skype.net