PostgreSQL Configuration

An introduction

Credits

Mostly taken from Josh Berkus' presentation at PGCon 2008 on GUCs.

Note: Many of these ideas are being added to documentation in 8.4

Two main files

- Both are located in $PGDATA
  - Unix typically symlinked to /etc/postgres/...
  - Windows: Program Files/PostgreSQL/config
  - Location can be changed
- postgresql.conf
  - Most Grand Unified Configuration Settings, (GUCs)
- pg_hba.conf
  - Security connection settings

GUCs

- May be specified in command-line switches
- Some can be set on individual objects
  - ALTER ROLE basic_user SET search_path = 'safe';
- Can be checked from via SQL
  - SELECT * FROM pg_settings;
  - SHOW pg_settings;
- Some can also be set via SQL for that session
  - UPDATE pg_settings SET setting = '12MB' where name = 'work_mem';
  - SET work_mem = '12MB';

GUCs

Usually, just use postgresql.conf

GUC Contexts

- Can be read from pg_settings
- Contexts
  - User – Runtime, per session
  - Superuser – Runtime, Per instance, superuser only
  - Sighup – Require a soft reset
  - Postmaster – Require a hard restart
  - Backend – Developer settings
  - Internal – Compile time settings

Important Settings

- listen_addresses
- pg_hba.conf
- max_connections
- shared_buffers
- work_mem
- maintenance_work_mem
- wal_buffers

listen_addresses

- tells Postgres what IPs to listen on
- default is localhost, which will prevent external connections
- most development/test environments should be set to * for all
- production environments should be set to the server's IP address
- This corresponds closely with pg_hba.conf

pg_hba.conf

- A whitelist of allowed connections
- Supoprts
  - Unix sockets
  - TCP Connections (IPv4 and Ipv6)
  - Local connections
- What this contains depends on desired security
  - May need a extremely open network
  - May be able to restrict a limited whitelist of IP addresses

max_connections

- The number of sessions open agains the database at one time
- Don't forget superuser_reserved_connections (defaults to 3)
- Defaults to 100

max_connections

- If application has lots of small transactions, higher numbers will be useful
  - It can be raised, but probably shouldn't go beyond 1000
  - More connections == More memory usage
- In processing-intensive applications such as data warehousing, a smaller number of allowed connections will provide each connection with better resources

shared_buffers

- Postgres working memory
- Used for managing connections, active operations
- But some things are not included here
  - in-memory sorts
  - vacuuming and analyze operations

shared_buffers

- Defaults to around 8 MB
- Recommended to be around 1/4th of available memory in dedicated environments
- should also look at SHMMAX and SHMALL
- these should be in sync with the shared buffers

work_mem

- dedicated memory given to each operation
- each statement can make multiple operations
- idea is to give operations a maximum amount of memory, without going into swap
- this is a user-context setting
  - can be tweaked before a crazy query

work_mem

- recommended for DBs with simple operations:
  - available physical RAM / max_connections
- for DBs with complex operations (data warehousing):
  - available physical RAM / (2 * max_connections)

maintenance_work_mem

- same as work_mem, but for vacuum, etc
- recommended to be available RAM/8

wal_buffers

- The size of the write-ahead-log (wal) files
- default is 8 kB
- SMP machines are better with 8 MB

Questions?

Ian Bailey
ianb@nulogy.com