The GUCS:
A Three-Hour Tour

Josh Berkus
Red Hat OSAS
SCALE 16x

agenda

#what are GUCS?

#understanding GUCS

#simple basic configuration

#new GUCS

#grand tour

related stuff

~~#pg_hba.conf~~

~~#recovery.conf~~

~~#pg_ident.conf~~

~~#libpq settings~~

~~#psql \ commands~~

~~#system functions~~

What are GUCS?

#“Grand Unified Configuration Settings”

– A large set of global and session database settings which live in:

src/backend/utils/misc/guc.c

#Your main way of configuring and tuning the database.

– in addition to compile options, system configuration and schema tuning!

#Pronounced “gucks”

269 Switches and Levers

“269 Settings?!?”

“Don't worry, Skipper!”

The 19 (or so) Settings

most users need to adjust

- listen_addresses
- max_connections
- ssl (+3)
- shared_buffers
- work_mem
- maintenance_work_mem
- autovaccum_work_mem
- effective_cache_size
- random_page_cost
- data_checksums
- temp_file_limit
- statement_timeout
- idle_in_transaction_timeout
- archive_command
- archive_mode
- hot_standby_feedback
- wal_level
- wal_keep_segments
- logging settings (3-8)

Figuring things out.

Setting GUCS

6

7

8

Six GUCS Contexts

#session

#superuser

#reload

#restart

#backend

#internal

Six GUCS Contexts

#session, or “user”

- can be set on per-session basis at runtime
- can also be changed by altering the ROLE

#superuser

- can be set at runtime for the whole server, but only by the superuser

Six GUCS Contexts

#reload, or “sighup”

- requires a soft restart of the postmaster to change
- also called a “HUP”

#restart, or “postmaster”

- can only be changed with a hard restart

More on reloading

- Three methods for reloading

```
pg_ctl -D PGDATA reload
kill -HUP <postmaster.pid>
SELECT pg_reload_conf();
```

Six GUCS Contexts

#backend

- – developer settings which have to be set before session start

#internal

- – compile-time settings provided for reference. Unalterable.

7 Places to set GUCs

1. postgresql.conf file
2. config directory
3. SET
4. pg_settings
5. ALTER objects
6. command-line switches & PGOPTIONS
7. ALTER SYSTEM SET

postgresql.conf

#$PGDATA/postgresql.conf

– can be relocated or symlinked

#Primary way to change GUCS

– for performance

– for logging

– for defaults

postgresql.conf

#just a big list of settings

#defaults are “# commented” out

#if a setting appears multiple times, the last one takes effect

#can use “include” files

config directory

# include_if_exists='dir'

#directories containing multiple config files

#loaded in alphanum order

#better for config managers/automation

SET & SHOW

- #Access GUCS in a SQL session
  - SET changes
    - only applies to user and superuser contexts
    - best way to change things for only one session
    - can be embedded in stored procedures

```
SET enable_mergejoin = false;
```

  - SHOW displays
    - SHOW ALL lets you see all GUCS

```
SHOW shared_buffers;
```

SET & SHOW: set_config

#set_config(setting, value, is_local)

- alternative to SET
- better for scripts since it takes a string

```
SELECT set_config('work_mem','1GB', TRUE);
```

#current_setting (setting)

- alternative to SHOW

```
SELECT current_setting('lc_collate');
```

pg_settings

#Tabular representation of GUCS

- – a “system view”
- – can be queried
- – can be updated, which runs a SET on the relevant parameter

```
SELECT * FROM pg_settings
WHERE name = 'work_mem';

UPDATE pg_settings SET setting = '12MB'
WHERE name = 'work_mem';
```

ALTER objects

#You can ALTER some objects to change their configuration settings

- ROLES (users)
- DATABASEs
- FUNCTIONs
- Changes configuration setting when that object is used (unless overridden by a SET)

```
ALTER ROLE “data entry” SET
search_path = 'entry, public';
```

ALTER ROLE + DB

```
ALTER ROLE IN DATABASE accts SET
search_path = '$user, webview,
contrib, public';
```

#now can set ROLE-based settings just for a specific database

#essential for search_path, useful for others

ALTER objects

#Good way to discriminate between applications & users

- set search_path
  - sometimes for security
- primitive resource management
  - change work_mem, etc. for specific ROLES, applications or reports
- localization
  - if different users have different desired displays

```
ALTER DATABASE sao_paulo SET timezone = 
'America/SaoPaulo';
```

ALTER objects

```
CREATE FUNCTION check_password(uname TEXT, pass TEXT)
RETURNS BOOLEAN AS $$
DECLARE passed BOOLEAN;
BEGIN
        SELECT  (pwd = $2) INTO passed
        FROM    pwds
        WHERE   username = $1;

        RETURN passed;
END;
$$  LANGUAGE plpgsql
    SECURITY DEFINER
    -- Set a secure search_path: trusted
    -- schema(s), then 'pg_temp'.
    SET search_path = admin, pg_temp;
```

command-line and pgoptions

#Set options when you start PostgreSQL

- as command-line switches
- using the PGOPTIONS shell variable

#Good for testing

  - and for restarting with recovery options

#Also good with containers

```
postgres -c 'bgwriter_delay = 50ms'
export PGOPTIONS = '-c zero_damaged_pages = ON'
```

ALTER SYSTEM SET

# set postgresql.auto.conf settings via the command-line

# postgresql.auto.conf lives in PGDATA

# alternative to managing postgresql.conf via template

```
ALTER SYSTEM SET work_mem = '128MB';
```

“That's a lot of places for settings.”

New: pg_file_settings

```
gucs=# select sourcefile, sourceline, name from pg_file_settings;
```

sourcefile	sourceline	name
/var/lib/postgresql/data/postgresql.conf	59	listen_addresses
/var/lib/postgresql/data/postgresql.conf	64	max_connections
/var/lib/postgresql/data/postgresql.conf	113	shared_buffers
/var/lib/postgresql/data/postgresql.auto.conf	3	work_mem

Eight types of GUCS

#Boolean

#Integer

#Float

#Memory / Disk

#Time

#Strings

#ENUMs

#Lists

Eight types of GUCS

#Boolean

- take TRUE | FALSE or ON | OFF

```
enable_seqscan = on
```

#Integer

- take whole numbers only

```
max_connections = 700
```

Eight types of GUCS

#Float

- take decimal values

```
random_page_cost = 3.75
```

#Memory / Disk Size

- can take integers
- but can also take “computer units”: kB, MB, GB

```
work_mem = 512MB
shared_buffers = 4096
```

Eight Types of GUCS

#Time

- take time values (units vary)
- units are often omitted

```
authentication_timeout = 30s
```

#Strings

- take unchecked strings
- often refers to file locations or program names

```
log_directory = 'pg_log'
```

Eight Types of GUCS

#ENUMs

- take a string value from a prescribed list

```
wal_level = hot_standby
```

#Lists

- take a list of values, either freeform or ENUM

```
search_path = 'main, archive, “$user”'
```

What's New, GUCS?

23 settings eliminated

```
# trace_locks_oidmin
# wal_sender_delay
# max_fsm_pages
# regex_flavor
# krb_server_hostname
# add_missing_from
# krb_realm
# unix_socket_directory
# max_fsm_relations
# explain_pretty_print
# debug_deadlocks
# wal_debug
# sql_inheritance
# ssl_renegotiation_limit
# silent_mode
# trace_lwlocks
# trace_locks
# trace_locks_table
# custom_variable_classes
# log_btree_build_statsj
# checkpoint_segments
# trace_userlocks
# krb_srvname
```

69 settings added

```
# autovacuum_multixact_freeze_max_age
# autovacuum_work_mem
# backend_flush_after
# bgwriter_flush_after
# checkpoint_flush_after
# cluster_name
# data_checksums
# default_transaction_deferrable
# dynamic_shared_memory_type
# enable_gathermerge
# enable_indexonlyscan
# event_source
# exit_on_error
# force_parallel_mode
# gin_pending_list_limit
# hot_standby_feedback
# huge_pages
# idle_in_transaction_session_timeout
# ignore_checksum_failure
# lock_timeout
# log_file_mode
# log_replication_commands
# max_logical_replication_workers
# max_parallel_workers
# max_parallel_workers_per_gather
# max_pred_locks_per_page
# max_pred_locks_per_relation
# max_pred_locks_per_transaction
# max_replication_slots
# max_sync_workers_per_subscription
# max_wal_size
# max_worker_processes
# min_parallel_index_scan_size
# min_parallel_table_scan_size
# min_wal_size
# old_snapshot_threshold
# operator_precedence_warning
# parallel_setup_cost
# parallel_tuple_cost
# quote_all_identifiers
# replacement_sort_tuples
# restart_after_crash
# row_security
# session_preload_libraries
# ssl_ca_file
# ssl_cert_file
# ssl_crl_file
# ssl_dh_params_file
# ssl_ecdh_curve
# ssl_key_file
# ssl_prefer_server_ciphers
# synchronous_standby_names
# syslog_sequence_numbers
# syslog_split_messages
# temp_file_limit
# track_commit_timestamp
# track_io_timing
# transaction_deferrable
# unix_socket_directories
# vacuum_multixact_freeze_min_age
# vacuum_multixact_freeze_table_age
# wal_compression
# wal_consistency_checking
# wal_log_hints
# wal_receiver_status_interval
# wal_receiver_timeout
# wal_retrieve_retry_interval
# wal_sender_timeout
# wal_writer_flush_after
```

tuning disk flush

```
#backend_flush_after
#bgwriter_flush_after
#checkpoint_flush_after
#wal_writer_flush_after
```

identification

#cluster_name

#event_source

SSL config

```
#ssl_ca_file
#ssl_cert_file
#ssl_crl_file
#ssl_dh_params_file
#ssl_ecdh_curve
#ssl_key_file
#ssl_prefer_server_ciphers
```

Timeouts & Resource Limits

#old_snapshot_threshold

#lock_timeout

#idle_in_transaction_session_timeout

#autovacuum_work_mem

#max_pred_locks_per_page

#max_pred_locks_per_relation

#max_pred_locks_per_transaction

New replication stuff!

```
#max_logical_replication_workers
#max_replication_slots
#max_sync_workers_per_subscription
#wal_receiver_status_interval
#wal_receiver_timeout
#wal_retrieve_retry_interval
#wal_sender_timeout
#synchronous_standby_names
```

WAL compression

#wal_compression

#wal_log_hints

parallel query

```
#max_worker_processes
#max_parallel_workers
#max_parallel_workers_per_gather
#force_parallel_mode
#enable_gathermerge
#parallel_setup_cost
#parallel_tuple_cost
```

Getting Simpler: WAL size

#was:

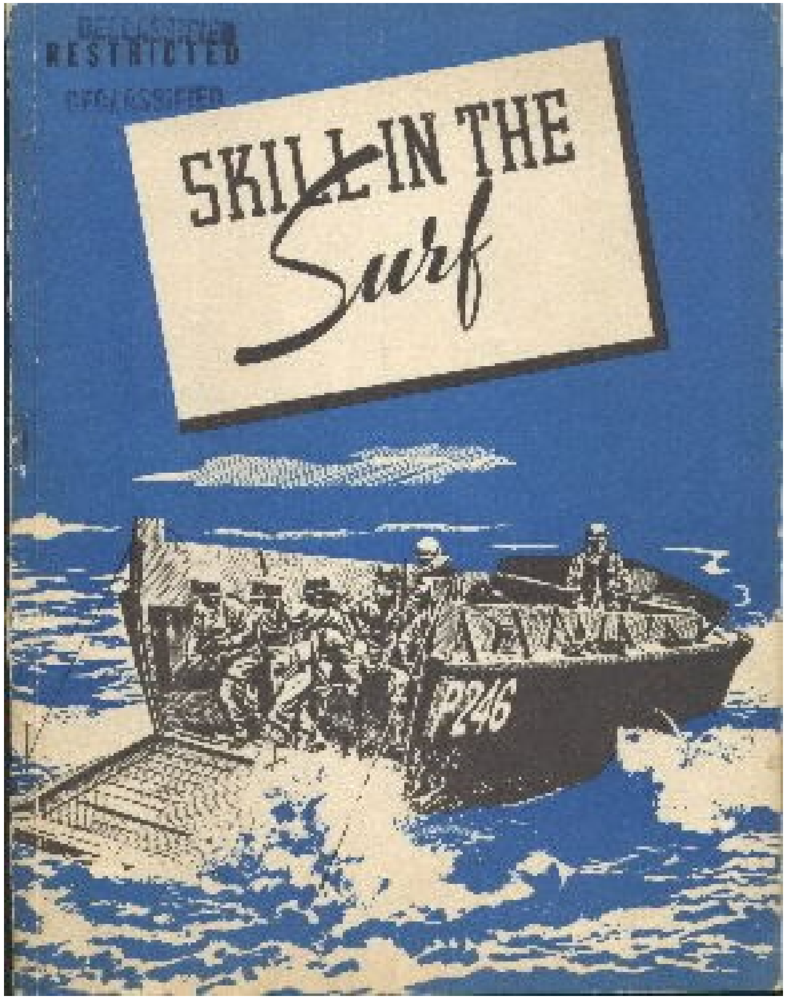
- checkpoint_segments
- wal_buffers
- checkpoint_timeout

#is:

- min_wal_size
- max_wal_size

… with good defaults!

GUCS Reference



The 19 (or so) Settings

most users need to adjust

- listen_addresses
- max_connections
- ssl (+3)
- shared_buffers
- work_mem
- maintenance_work_mem
- autovaccum_work_mem
- effective_cache_size
- random_page_cost
- data_checksums
- temp_file_limit
- statement_timeout
- idle_in_transaction_timeout
- archive_command
- archive_mode
- hot_standby_feedback
- wal_level
- wal_keep_segments
- logging settings (3-8)

GUCS: the Grand Tour

Categories (mine)

#file locations

#connection settings

#resource management

#maintenance

#WAL & checkpoints

#replication

#query tuning

#logging

#statistics

#locks

#locale & formatting

#other settings & defaults

#compatibility

#custom GUCS

#developer options

#presets

custom variable classes

#for supporting various PostgreSQL Extensions

```
custom_variable_classes =
'pg_stat_statements,auto_explain'

pg_stat_statements.max = 10000
pg_stat_statements.track = all

auto_explain.log_min_duration = '200ms'
auto_explain.log_nested_statements = on
auto_explain.log_format = json
```

Contact

#github.com/jberkus/annotated.conf/

#Josh Berkus: jberkus@redhat.com

- www.databasesoup.com
- @fuzzychef

#PostgreSQL: www.postgresql.org

#Red Hat: http://redhat.com/community