

PGConf.Сибирь 2023

PosgresPro

16

Павел Лузанов

Обо мне

Павел Лузанов

Отдел образовательных программ Postgres Professional

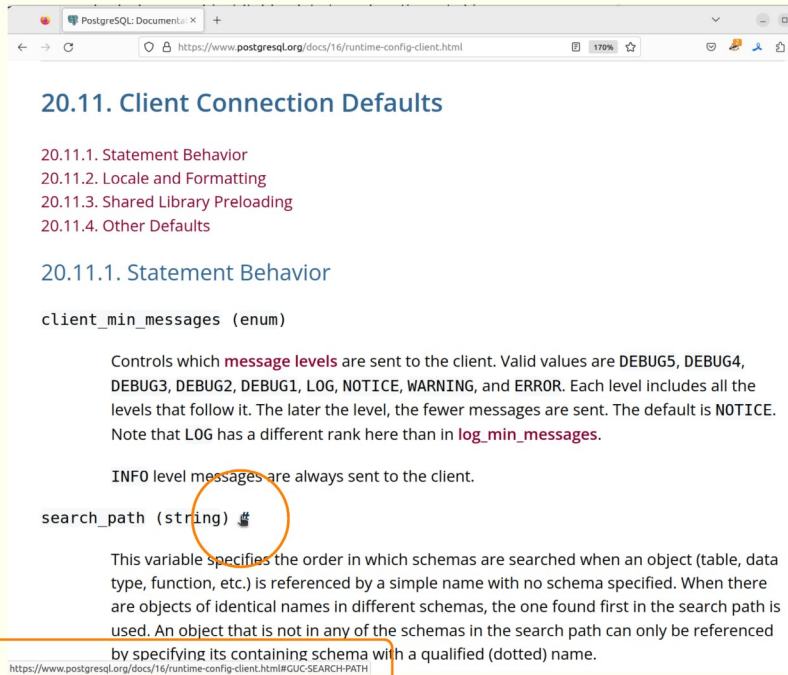
- edu@postgrespro.ru
- все в свободном доступе на <https://postgrespro.ru/education>
- Хабр
<https://habr.com/ru/company/postgrespro/blog/>

MESON: Новая система сборки сервера

```
$ git clone git://git.postgresql.org/git/postgresql.git
$ cd postgresql
$ meson setup build --prefix=$HOME/pg16
$ cd build
$ ninja
$ ninja install
```


Документация

Ссылки на разделы/секции внутри страниц



20.11. Client Connection Defaults


- 20.11.1. Statement Behavior
- 20.11.2. Locale and Formatting
- 20.11.3. Shared Library Preloading
- 20.11.4. Other Defaults

20.11.1. Statement Behavior

`client_min_messages` (enum)

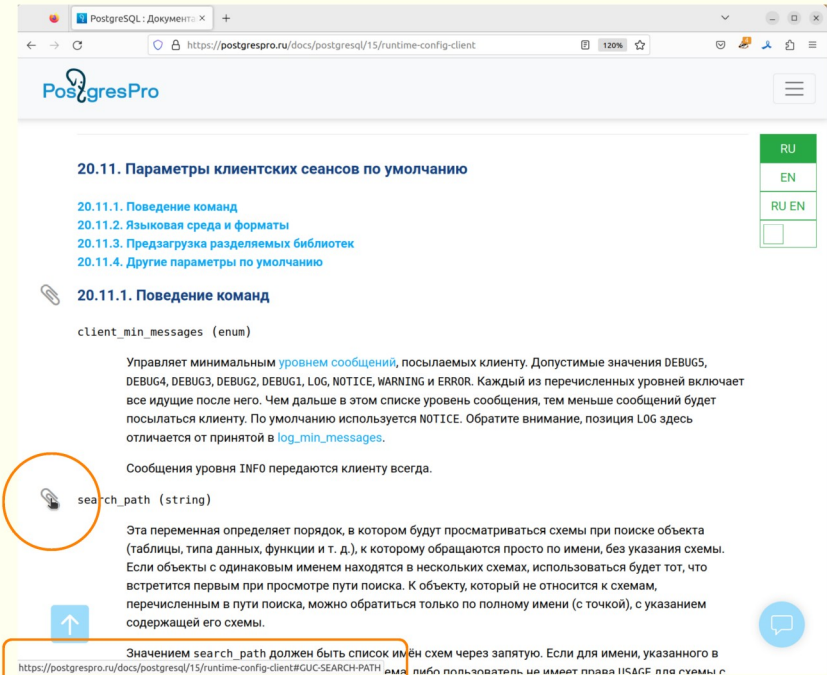
Controls which **message levels** are sent to the client. Valid values are DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, LOG, NOTICE, WARNING, and ERROR. Each level includes all the levels that follow it. The later the level, the fewer messages are sent. The default is NOTICE. Note that LOG has a different rank here than in `log_min_messages`.

INFO level messages are always sent to the client.

`search_path` (string) 

This variable **specifies** the order in which schemas are searched when an object (table, data type, function, etc.) is referenced by a simple name with no schema specified. When there are objects of identical names in different schemas, the one found first in the search path is used. An object that is not in any of the schemas in the search path can only be referenced by specifying its containing schema with a qualified (dotted) name.

<https://www.postgresql.org/docs/16/runtime-config-client.html#GUC-SEARCH-PATH>



20.11. Параметры клиентских сеансов по умолчанию


- 20.11.1. Поведение команд
- 20.11.2. Языковая среда и форматы
- 20.11.3. Предзагрузка разделяемых библиотек
- 20.11.4. Другие параметры по умолчанию

20.11.1. Поведение команд

`client_min_messages` (enum)

Управляет минимальным **уровнем сообщений**, посылаемых клиенту. Допустимые значения DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, LOG, NOTICE, WARNING и ERROR. Каждый из перечисленных уровней включает все идущие после него. Чем дальше в этом списке уровень сообщения, тем меньше сообщений будет посылаться клиенту. По умолчанию используется NOTICE. Обратите внимание, позиция LOG здесь отличается от принятой в `log_min_messages`.

Сообщения уровня INFO передаются клиенту всегда.

`search_path` (string) 

Эта переменная определяет порядок, в котором будут просматриваться схемы при поиске объекта (таблицы, типа данных, функции и т. д.), к которому обращаются просто по имени, без указания схемы. Если объекты с одинаковым именем находятся в нескольких схемах, использоваться будет тот, что встретится первым при просмотре пути поиска. К объекту, который не относится к схемам, перечисленным в пути поиска, можно обратиться только по полному имени (с точкой), с указанием содержащей его схемы.

Значением `search_path` должен быть список имен схем через запятую. Если для имени, указанного в `search_path`, не существует схемы, сервер выдаст ошибку. Если для имени, указанного в `search_path`, не существует права USAGE для схемы, сервер выдаст ошибку.

<https://postgrespro.ru/docs/postgresql/15/runtime-config-client.html#GUC-SEARCH-PATH>

Локализация

Удалены параметры `lc_ctype` и `lc_collate`

```
=# CREATE DATABASE db_icu  
    LOCALE_PROVIDER = icu ICU_LOCALE = 'ru-x-icu'  
    TEMPLATE = template0;
```

v15

```
=# SHOW lc_ctype;  
ERROR: unrecognized configuration parameter "lc_ctype"
```

Запрос к `pg_database`

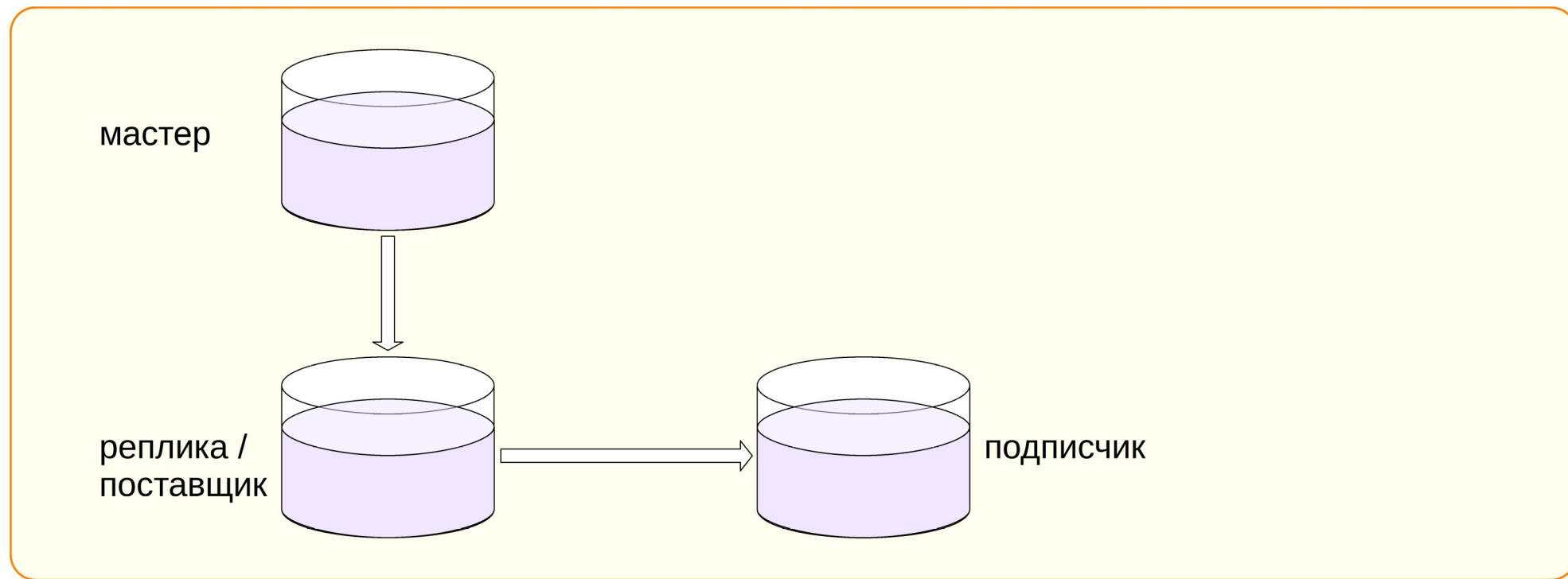
name	locprovider	ctype	collate	iculocale
db_icu	i	en_US.UTF-8	en_US.UTF-8	ru-x-icu
demo	c	en_US.UTF-8	en_US.UTF-8	

initdb: установка параметров конфигурации

```
$ initdb "-c wal_level=logical" "-c port=5402" -D ~/pg16/replica  
  
$ tail -2 ~/pg16/replica/postgresql.conf  
wal_level = logical  
port = 5402
```

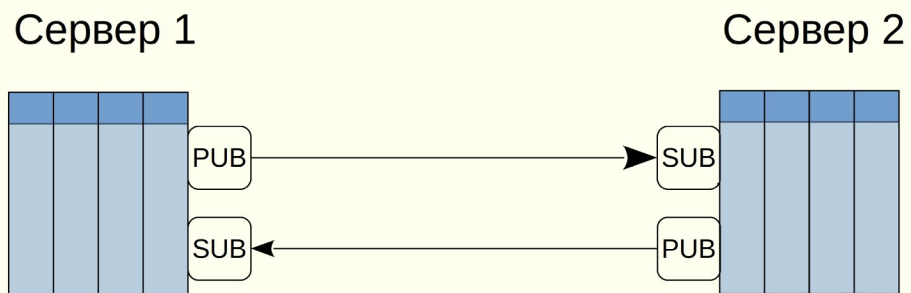
Логическая репликация

Публикация на реплике



Логическая репликация

Двухнаправленная репликация



Логическая репликация

Разное

+ Hash-индексы
в v17

- Использование неуникальных индексов В-дерево при REPLICATION FULL
- COPY в двоичном формате для начальной синхронизации
- *Применение больших транзакций параллельными рабочими процессами (только для разработчиков)*

Методы сжатия

pg_dump	zlib	lz4	zstd
pg_basebackup	gzip	lz4	zstd
pg_receivewal	gzip	lz4	
wal_compression	pglz	lz4	zstd
TOAST	pglz	lz4	
<i>Данные</i>		—	

Настройка сервера

Файлы конфигурации

	postgresql.conf	pg_hba.conf	pg_ident.conf
Включение файлов	<code>#include #include_if_exists #include_dir</code>	<code>#include #include_if_exists #include_dir</code>	<code>#include #include_if_exists #include_dir</code>
Поддержка regex		<code>dbname user</code>	<code>system-user pg-user</code>
Специальные значения для pg user		<code>+user all</code>	<code>+user all</code>
Доступ через SQL	<code>pg_file_settings</code>	<code>pg_hba_file_rules</code>	<code>pg_ident_file_mappings</code>

[Авто]очистка

Регулирование нагрузки на лету

```
SELECT
    pid
    , heap_blks_scanned, heap_blks_total -- прогресс по таблице
    , indexes_processed, indexes_total -- прогресс по индексам
FROM pg_stat_progress_vacuum
WHERE relid = 'bigtable'::regclass;
```

↑ autovacuum_vacuum_cost_limit
↓ autovacuum_vacuum_cost_delay

v17

Управление размером буферного кеша

```
VACUUM(ANALYZE, BUFFER_USAGE_LIMIT 0);
```

Без суперпользователя

Привилегия MAINTAIN

```
=# GRANT MAINTAIN  
ON pg_class  
TO alice;
```

VACUUM, ANALYZE,
VACUUM FULL, CLUSTER,
REINDEX,
REFRESH MATERIALIZED VIEW,
LOCK TABLE

```
=# GRANT pg_maintain  
TO alice;
```

MAINTAIN
на все отношения
в базе данных

Без суперпользователя

Привилегия MAINTAIN

```
=# GRANT MAINTAIN  
ON pg_class  
TO alice;
```

Откатили

```
=# GRANT pg_maintain  
TO alice;
```

Без суперпользователя

Резервирование подключений к серверу

superuser_reserved_connections

reserved_connections

max_connections

+pg_use_reserved_connections

Без суперпользователя

Предопределенные роли

16	pg_use_reserved_connections, pg_create_subscription
15	pg_checkpoint
14	pg_read_all_data, pg_write_all_data
11	pg_read_server_files, pg_write_server_files, pg_execute_server_program
10	pg_read_all_settings, pg_read_all_stats, pg_stat_scan_tables, pg_monitor
9.6	pg_signal_backend

Без суперпользователя

Членство в ролях

```
postgres=# CREATE ROLE alice LOGIN CREATEROLE INHERIT;  
postgres=# GRANT pg_read_all_data TO alice; -- краткая форма
```

Два способа воспользоваться привилегиями роли:

1. Наследование привилегий

```
alice=> SELECT * FROM any_table;
```

2. Переключение на роль

```
alice=> SET ROLE pg_read_all_data;
```

Без суперпользователя

Параметры членства в ролях

Полная форма

```
=# GRANT pg_read_all_data TO alice
   WITH ADMIN OPTION, -- OPTION or TRUE | FALSE
      INHERIT TRUE,    -- наследование привилегий
      SET FALSE      -- переключение на роль
   GRANTED BY postgres; -- теперь это важно!
```


Без суперпользователя

Роли с атрибутом CREATEROLE

```
alice=> SET createrole_self_grant = 'SET, INHERIT';
alice=> CREATE ROLE bob LOGIN;

GRANT bob TO alice WITH ADMIN TRUE GRANTED BY postgres;
GRANT bob TO alice WITH SET TRUE, INHERIT TRUE GRANTED BY alice;

-- Запрос к pg_auth_members
role | member | grantor | admin | inherit | set
-----+-----+-----+-----+-----+-----
bob  | alice  | postgres | t     | f       | f
bob  | alice  | alice    | f     | t       | t
(2 rows)
```

Без суперпользователя

psql: \drg – информация о членстве в ролях

```
alice=> \drg alice
                List of role grants
 Role name | Member of | Options      | Grantor
-----+-----+-----+-----
alice     | bob       | INHERIT, SET | alice
alice     | bob       | ADMIN        | postgres
(2 rows)
```

Без суперпользователя

psql: \du & \dg – удален столбец «Member of»

```
alice=> \du alice
      List of roles
Role name | Attributes
-----+-----
alice    | Create role
```

Мониторинг

HOT & fillfactor

```
pg_stat_all_tables
```

```
relid
```

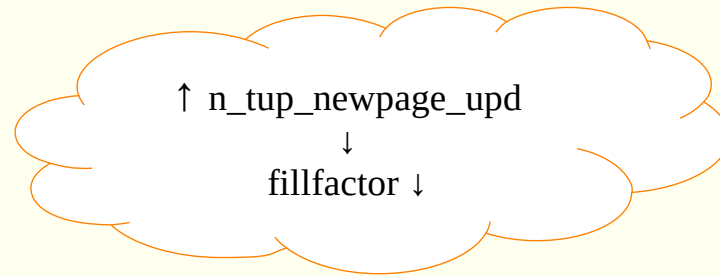
```
...
```

```
n_tup_upd
```

```
n_tup_hot_upd
```

```
n_tup_newpage_upd
```

```
...
```



Мониторинг

pg_stat_io: накопительная статистика ввода-вывода

pg_stat_io

backend_type -- client_backend, autovacuum worker, ...

object -- relation, temp_relation

context -- normal, vacuum, bulkread, bulkwrite

reads, writes, extends, op_bytes

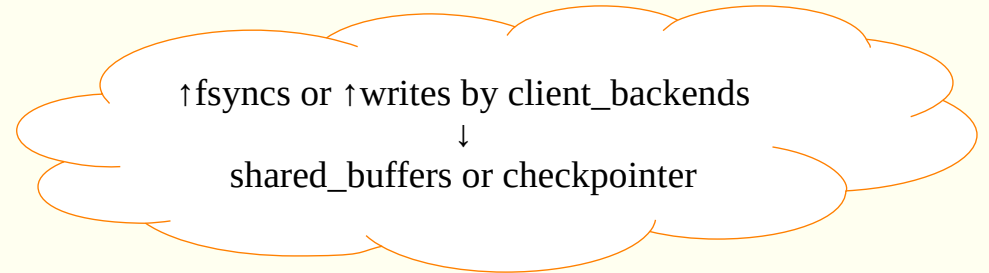
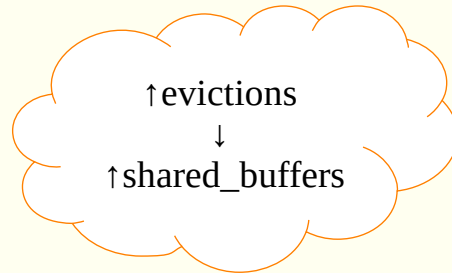
hits

evictions

reuses

fsyncs

stats_reset



Мониторинг

Общий план параметризованного запроса

```
EXPLAIN (COSTS OFF, GENERIC_PLAN)  
SELECT * FROM boarding_passes WHERE seat_no = $1;
```

Gather

Workers Planned: 2

-> Parallel Seq Scan on boarding_passes
Filter: ((seat_no)::text = \$1)

Клиентские приложения

psql: автоматическое завершение \watch

```
=# SELECT *  
   FROM pg_stat_activity  
   WHERE pid = $1  
\watch interval=1 count=5 min_rows=0
```

v16

v17

Клиентские приложения

psql: статус завершения команды ОС

```
=# \! pwd
/home/pal/pg16

=# \echo :SHELL_EXIT_CODE
0

=# \echo :SHELL_ERROR
false
```

Клиентские приложения

libpq: балансировка подключений

```
psql "host=replica1,replica2,replica3 load_balance_hosts=random"
```

libpq: требуемый метод аутентификации

```
psql 'host=localhost require_auth=md5,scram-sha-256'
```

Производительность

Параллельное выполнение полного и правого хеш-соединения

```
EXPLAIN (costs off) SELECT count(*)  
  FROM bookings b FULL OUTER JOIN tickets t USING (book_ref);
```

Finalize Aggregate

-> Gather

Workers Planned: 2

-> Partial Aggregate

-> **Parallel Hash Full Join**

Hash Cond: (t.book_ref = b.book_ref)

-> Parallel Seq Scan on tickets t

-> Parallel Hash

-> Parallel Seq Scan on bookings b

Производительность

Новый старый параметр

```
force_parallel_mode = debug_parallel_query
```

Новый параметр, пока не трогать!

```
debug_io_direct
```

SQL/JSON

Поддержка стандарта SQL

- Функции-конструкторы
 - JSON_ARRAY, JSON_ARRAYAGG, JSON_OBJECT, JSON_OBJECTAGG } v16
 - JSON, JSON_SCALAR, JSON_SERIALIZE } v17
- Предикаты
IS JSON [VALUE], IS JSON ARRAY, IS JSON OBJECT, IS JSON SCALAR
- ... work in progress ...

PGConf.Сибирь 2023

PostgresPro

17

16