

PGConf.Russia 2023

PostgresPro

*** PostgreSQL 16 ***

Павел Лузанов

Обо мне

Павел Лузанов

Отдел образовательных программ Postgres Professional

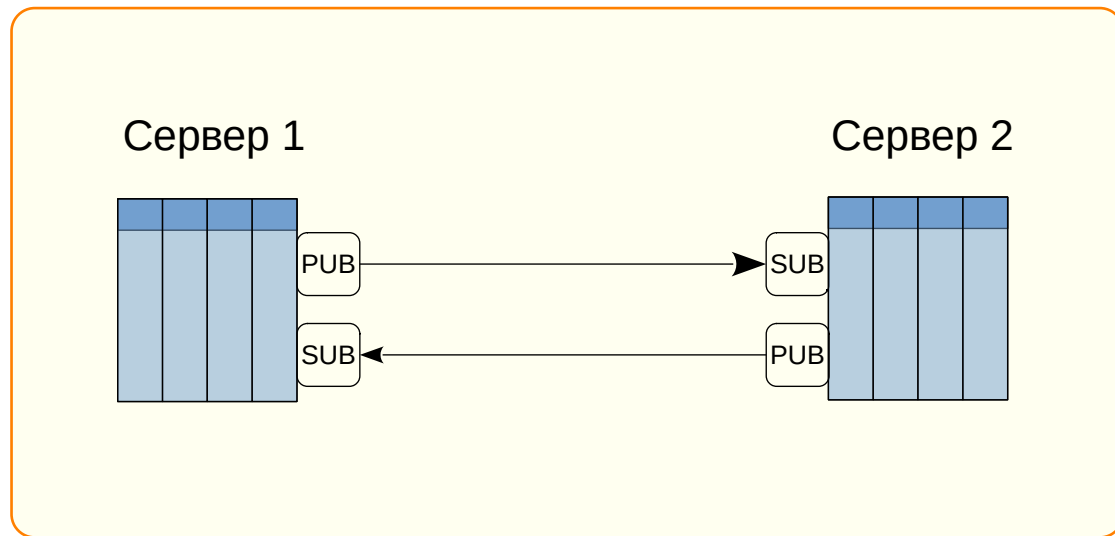
- edu@postgrespro.ru
- все в свободном доступе на <https://postgrespro.ru/education>
- Хабр <https://habr.com/ru/company/postgrespro/blog/>

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
```

Логическая репликация

Двунаправленная репликация



Завтра в 10:00

Андрей Рудометов
Сравнение двунаправленной репликации
в ваниле и Postgres Pro Multimaster

Логическая репликация

Разное

- Применение больших транзакций параллельными рабочими процессами
- Использование неуникальных индексов при `REPLICA IDENTITY FULL`
- `COPY` в двоичном формате может использоваться для начальной синхронизации

Завтра в 12:50

Vigneshwaran C
Logical replication internals

Методы сжатия

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

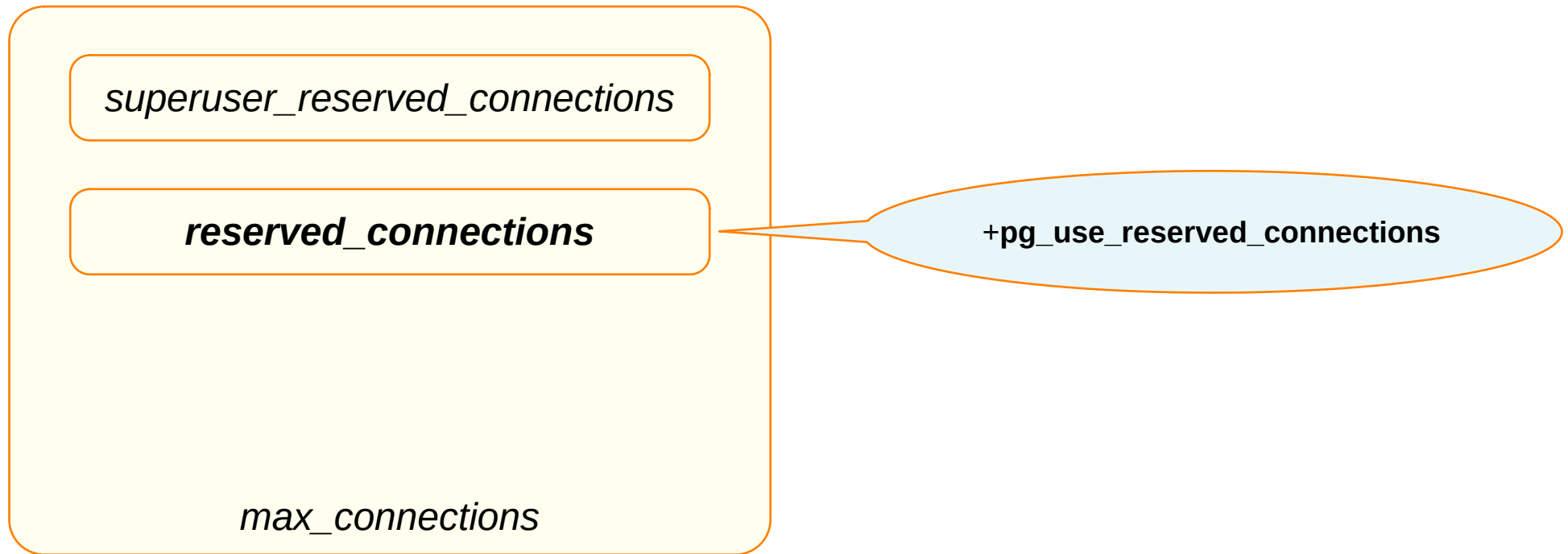
Настройка сервера

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>

Без суперпользователя

Резервирование подключений к серверу



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

Предопределенные роли

16	pg_maintain, 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;  
  
-- краткая форма  
postgres=# GRANT pg_maintain TO alice;  
  
alice=> VACUUM ANALYZE;          -- 1) наследование привилегий  
alice=> SET ROLE pg_maintain;    -- 2) переключение на роль
```

Без суперпользователя

Параметры членства в ролях

```
-- полная форма
=# GRANT pg_maintain 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

Роли с атрибутом CREATEROLE

```
alice=> \du alice
          List of roles
Role name | Attributes | Member of
-----+-----+-----
alice    | Create role | {bob,bob}
```

Мониторинг

HOT & fillfactor

```
pg_stat_all_tables
```

```
relid
```

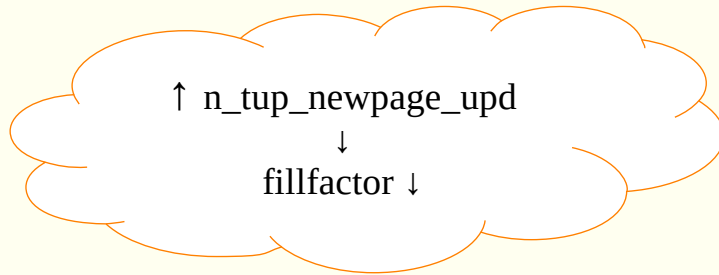
```
...
```

```
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, ...

io_object -- relation, temp_relation

io_context -- normal, vacuum, bulkread, bulkwrite

reads, writes, extends, op_bytes

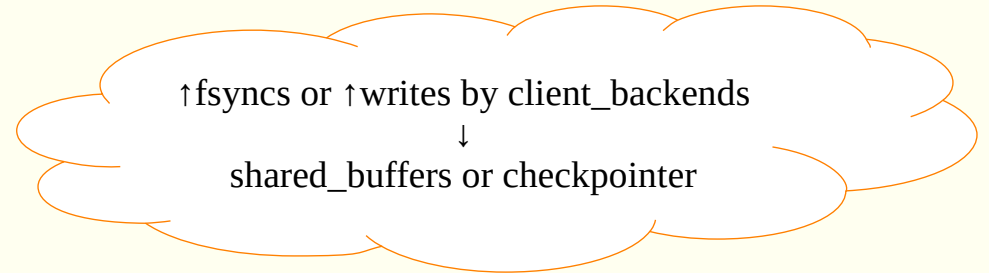
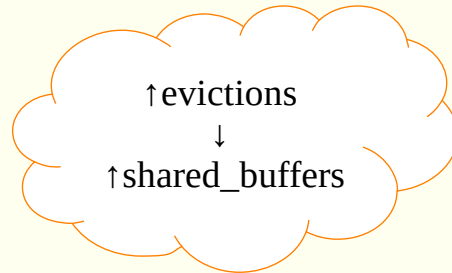
hits

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: статус завершения команды ОС

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

=# \echo :SHELL_EXIT_CODE
0
```

Клиентские приложения

libpq: балансировка подключений

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

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
```

SQL/JSON

Поддержка стандарта SQL

Work in progress...

PGConf.Russia 2023

PosgresPro

Спасибо!