

Что потребуется начинающему разработчику PostgreSQL?

Илья Евдокимов, «Тантор Лабс»

* Обо мне

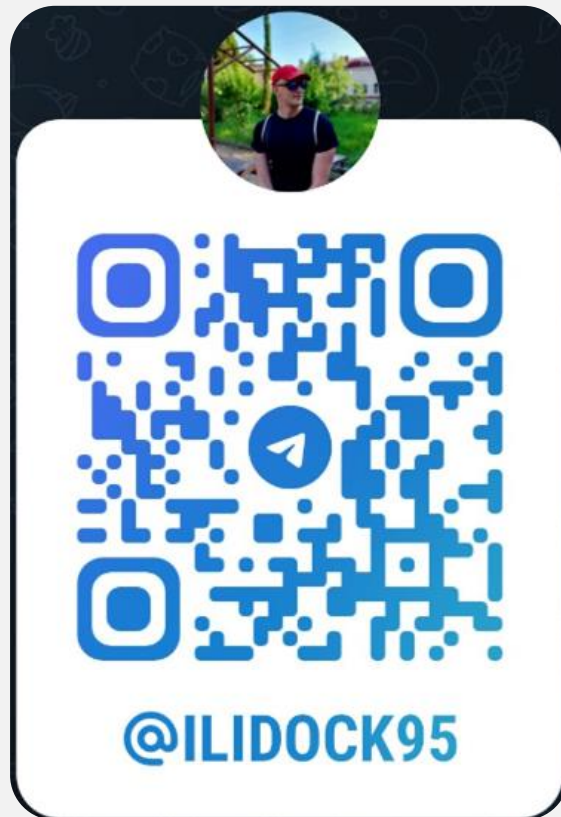


Образование:

- Магистратура ВМК МГУ им. Ломоносова

Опыт работы:

- Auriga (USB-hub 3.0)
- InfoTeCS (ПАК)
- Tantor(PostgreSQL)



* Что такое разработка PostgreSQL?



* O PostgreSQL



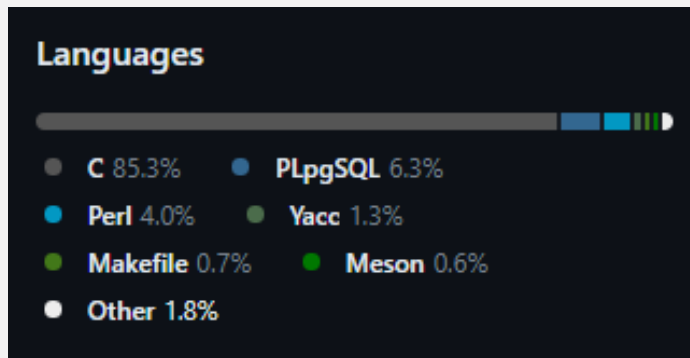
Язык C

1,3+ миллиона строк кода

30+ лет ведется разработка

Open-source проект

Документация SGML



Perl - это как та старая ржавая лопата в сарае: никто не помнит, когда она там появилась, все говорят, что ее пора заменить на что-то новое и блестящее, но когда приходит время перекопать огород, все опять берут ее в руки.

* Почему PostgreSQL?



Расширения

Объектно-реляционная

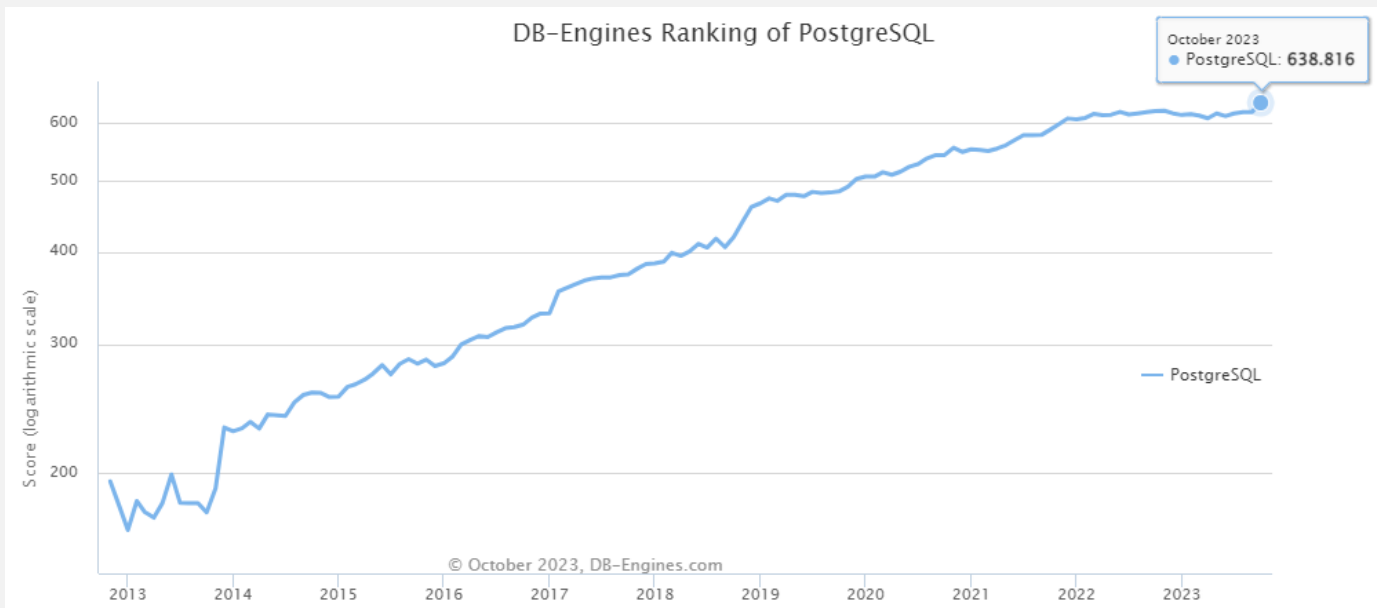
Всегда совместим с ACID

Становится популярней...

<https://survey.stackoverflow.co/2023/>



https://db-engines.com/en/ranking_trend/system/PostgreSQL



Team core



Peter Eisentraut	EDB	Dresden, Germany
Andres Freund	Microsoft	San Francisco, USA
Magnus Hagander	Redpill Linpro	Stockholm, Sweden
Jonathan Katz	AWS	New York, NY
Tom Lane	Pittsburgh, Pittsburgh	Pennsylvania, USA
Bruce Momjian	EDB	Philadelphia, Pennsylvania, USA
Dave Page	EDB	Oxfordshire, United Kingdom

* Contributor, Major contributor, Committer



- Team core, 7
- Major contributors, 41
- Contributors, 100+
- Committers, pgsql-committers@lists.postgresql.org

The screenshot shows the PostgreSQL Commitfest 2023-11 page. The status summary indicates 218 patches need review, 33 are waiting on the author, 25 are ready for a committer, 11 are committed, and 1 has been withdrawn. The total number of patches is 288, which is highlighted with a red box. Below the summary, there is a section for active patches, with the first patch being a bug fix by Kyotaro Horiguchi.

commitfest.postgresql.org/45/

Home / Commitfest 2023-11

Commitfest 2023-11

Search/filter Shortcuts New patch

Status summary: Needs review: 218. Waiting on Author: 33. Ready for Committer: 25. Committed: 11. Withdrawn: 1. **Total: 288.**

Active patches

Patch	Status	Ver	Author
Bug Fixes			
Error "initial slot snapshot too large" in create replication slot	Needs review		Kyotaro Horiguchi (horiguti),

* Клонирование репозитория PostgreSQL 16



```
git clone -b REL_16_STABLE https://github.com/postgres/postgres
cd postgres/
```

```
./configure --prefix=/usr/local/pgsql \  
    --enable-tap-tests \  
    --with-python \  
    --with-icu \  
    --with-lz4 \  
    --with-zstd \  
    --enable-debug \  
    --enable-cassert
```

```
make -j 4  
make install
```

* Новая группа и пользователь postgres



```
groupadd -r postgres
useradd -r -g postgres -d /var/lib/postgresql \
        -s /bin/bash postgres

mkdir -p /var/lib/postgresql
chown postgres:postgres /var/lib/postgresql
chmod 700 /var/lib/postgresql

# Для unix_socket_directories
# В случае #define DEFAULT_PGSOCKET_DIR  "/var/run/postgresql"

mkdir -p /var/run/postgresql
chown postgres:postgres /var/run/postgresql
```

* Инициализация кластера



```
chown postgres:postgres /var/log/postgresql  
su - postgres  
mkdir /var/lib/postgresql/my_data  
  
/usr/local/pgsql/bin/initdb \  
    -D /var/lib/postgresql/my_data
```

* Настройка postgresql.conf



```
cat >> /var/lib/postgresql/my_data/postgresql.conf << EOL
logging_collector = on
log_directory = '/var/log/postgresql/'
log_filename = 'postgresql.log'
log_statement = 'all'
log_min_messages = debug1
log_duration = on
log_destination = 'csvlog'
log_error_verbosity = verbose
log_lock_waits = on
EOL
```

* Запуск БД и проверка лога



```
/usr/local/pgsql/bin/pg_ctl \  
-D /var/lib/postgresql/my_data \  
start
```

```
ps f --forest -u postgres | grep -v "f --forest -u postgres"
```

```
>>  
      PID TTY          STAT       TIME COMMAND  
    10015 ?            Ss          0:00 /usr/local/pgsql/bin/postgres -D /var/lib/postgresql/my_data  
    10016 ?            Ss          0:00 \_ postgres: logger  
    10017 ?            Ss          0:00 \_ postgres: checkpointer  
    10018 ?            Ss          0:00 \_ postgres: background writer  
    10020 ?            Ss          0:00 \_ postgres: walwriter  
    10021 ?            Ss          0:00 \_ postgres: autovacuum launcher  
    10022 ?            Ss          0:00 \_ postgres: logical replication launcher
```

```
tail -f /var/log/postgresql/postgresql.csv
```

```
>>  
      ..."starting background worker process ""logical replication  
launcher""",,,,,,,,"do_start_bgworker, postmaster.c:5723",,,,,,"postmaster",,0
```

* Длительный запрос

```
su - postgres -c "/usr/local/pgsql/bin/psql -c 'select pg_sleep(30)'"
```

```
ps f --forest -u postgres | grep -v "f --forest -u postgres"
```

```
>>
  PID TTY          STAT       TIME COMMAND
 10207 ?            Ss          0:00 /usr/local/pgsql/bin/psql -c select pg_sleep(30)
 10015 ?            Ss          0:00 /usr/local/pgsql/bin/postgres -D /var/lib/postgresql/my_data
 10016 ?            Ss          0:00 \_ postgres: logger
 10017 ?            Ss          0:00 \_ postgres: checkpointer
 10018 ?            Ss          0:00 \_ postgres: background writer
 10020 ?            Ss          0:00 \_ postgres: walwriter
 10021 ?            Ss          0:00 \_ postgres: autovacuum launcher
 10022 ?            Ss          0:00 \_ postgres: logical replication launcher
 10209 ?            Ss          0:00 \_ postgres: postgres postgres [local] SELECT
```

* Как взаимодействуют процессы



```
lsof | head -n 1 && lsof | grep '/SYSV' | grep postgres
```

```
>>
```

COMMAND	PID	USER	FD	
postgres	10015	postgres	DEL	# postgres -D /var/lib/postgresql/my_data
postgres	10017	postgres	DEL	# postgres: checkpointer
postgres	10018	postgres	DEL	# postgres: background writer
postgres	10020	postgres	DEL	# postgres: walwriter
postgres	10021	postgres	DEL	# autovacuum launcher
postgres	10022	postgres	DEL	# logical replication launcher
postgres	10029	postgres	DEL	# postgres [local] SELECT

А где logger?

* Отладка



Для отладки необходимо знать идентификатор процесса, к которому следует подключиться.

Пример получения идентификатора процесса, который не является бэкендом:

```
ps uax | grep backgr
>>
postgres 10018 0.0 0.0 197104 ... 0:00 postgres: background writer
```

Пример получения идентификатора процесса бэкенда:

```
postgres=# select pg_backend_pid();
 pg_backend_pid
-----
          10594
(1 row)
```

```
postgres:~$ sudo gdb -p 10594
```

* Логирование в исходном коде



Уровни логирования в порядке критичности:

1. DEBUG5, DEBUG4, DEBUG3, DEBUG2, **DEBUG1**: Различные уровни отладочных сообщений. DEBUG5 самый подробный, DEBUG1 наименее подробный.
2. LOG или INFO: Информационные сообщения. Обычно это сообщения, которые могут быть полезными для администраторов, но которые не указывают на проблемы.

... NOTICE, WARNING, ERROR, FATAL, PANIC

```
char path[MAXPGPATH];
strncpy(path, "test", sizeof(path) - 1);
path[sizeof(path) - 1] = '\0';
elog(DEBUG1, "Value of path is %s", path);
```

* Assert

Синтаксис:

```
Assert(pointer != NULL);
```

Пример:

```
Datum
my_assert_function(PG_FUNCTION_ARGS)
{
    int32 arg = PG_GETARG_INT32(0);

    elog(LOG, "Received argument: %d", arg);
    Assert(arg != 1);

    if (arg == 1)
        PG_RETURN_TEXT_P(cstring_to_text("success"));
    else
        PG_RETURN_NULL();
}
```



Тестирование



Файлы и утилита `pg_regress`:

```
src/test/regress/sql/my_test.sql  
src/test/regress/expected/my_test.out  
src/test/regress/parallel_schedule
```

Запуск тестов:

```
# Тесты ядра БД  
make check  
  
# Тесты расширений  
make check-world PG_TEST_EXTRA='contrib isolation tap recovery modules kerberos ssl  
wal_consistency_checking ...'
```

При разработке нового функционала важно проверить:

- Ничего не сломалось: запустить все `make check`
- Ничего не замедлилось: воспользоваться `pgbench`
- Обратная совместимость: если менялся существующий функционал
- Codestyle: `pgindent`

* Проверка производительности

```
cat > /tmp/perf_test.sql << EOL
select my_assert_function();
EOL

/usr/local/pgsql/bin/pgbench \
    -c 2 \           # количество клиентов
    -j 2 \           # количество рабочих потоков внутри pgbench
    -T 5 \           # длительность в секундах
    -f /tmp/perf_test.sql \
    -p 5432 \
    -U postgres \
    postgres         # имя БД

>>
number of transactions actually processed: 60825
number of failed transactions: 0 (0.000%)
latency average = 0.164 ms
initial connection time = 2.987 ms
tps = 12171.657897
```

- Расположена в директории `doc/src/sgml/`
- XML-подобный формат
- Описание нового функционала должно было добавлено в существующие `sgml` файлы

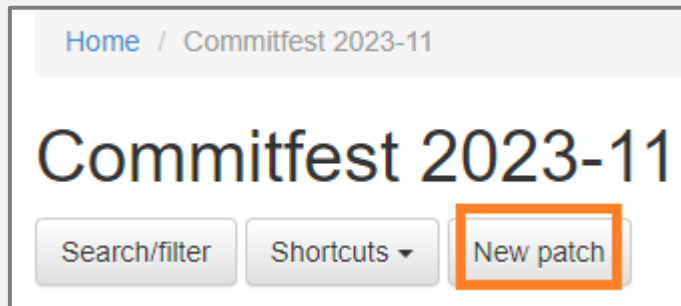
* Создание патча



```
git add <changed_files>  
git diff --cached > new_feature.patch
```

<https://commitfest.postgresql.org/>

Выбрать запись "Open" или "Future".



* Работа с образом



Инструкции (в репозитории)

https://github.com/EvdokimovIlia/my_first_extension/blob/main/laboratory_1.txt



Докер-образ

https://hub.docker.com/repository/docker/ilidock95/pg_bootcamp_5_oct/general

Спасибо за внимание!

