Postgres PROFESSIONAL

Конфигурирование сервера

Темы

Файл postgresql.conf

Файл postgresql.auto.conf (ALTER SYSTEM)

Представление pg_settings

Работа с параметрами во время выполнения

Установка параметров для БД и/или ролей

Установка параметров для функций

С чего начать настройку

postgresql.conf

Основной файл конфигурации

Расположение

```
SHOW config_file;
```

по умолчанию в директории с данными, `$PGDATA`

Действия при изменении

файл читается один раз при старте сервера,
поэтому при изменении надо попросить сервер перечитать:

```
$ pg_ctl reload
$ kill -HUP
psql# select pg_reload_conf();
```

некоторые параметры требуют перезапуска сервера

postgresql.conf

Типы данных параметров

`fsync = on` (логическое значение)

`listen_addresses = 'localhost'` (строка)

`max_connections = 100` (целое число)

`autovacuum_vacuum_scale_factor = 0.2` (дробное число)

`shared_buffers = 128MB` (kB, MB, GB, TB)

`authentication_timeout = 1min` (ms, s, min, h, d)

`wal_level = minimal` (minimal, archive, hot_standby, logical)

postgresql.conf

Директивы включения дополнительных файлов

- `include` *`filename`*
- `include_if_exists` *`filename`*
- `include_dir` *`dirname`* (все *.conf файлы)

Разрешается вложенность include

Если параметр указан несколько раз, применяется последнее считанное значение

postgresql.auto.conf

Файл конфигурации, расположен вместе с `postgresql.conf`

Считывается после `postgresql.conf`

Специальная команда для редактирования

```
ALTER SYSTEM
SET conf_parameter TO value;
```

добавляет или изменяет строки

```
ALTER SYSTEM
RESET conf_parameter | ALL;
```

удаляет строку или все строки

проверяет значения

сама по себе ничего не изменяет в системе

применение изменений — аналогично `postgresql.conf`

При запуске сервера

Установка параметров при запуске сервера

```
$ pg_ctl start -o  (или postgres -c)
```

Имеют предпочтение перед файлами конфигурации

Для изменения требуется перезапуск сервера

Опции командной строки запуска сервера сохраняются в файле `postmaster.opts`

Представление pg_settings

Параметр и значение

`name, setting, unit`

Описание

`category, short_desc, extra_desc`

Допустимые значения

`vartype, min_val, max_val, enumvals`

Значения по умолчанию

`boot_val, reset_val`

Источник

`source, sourcefile, sourceline`

Контекст

`context`

Пример

```
postgres=# select * from pg_settings where name = 'shared_buffers';
-[ RECORD 1 ]---------------------------------------------------
name       | shared_buffers
setting    | 16384
unit       | 8kB
category   | Resource Usage / Memory
short_desc | Sets the number of shared memory buffers used by the server.
extra_desc |
context    | postmaster
vartype    | integer
source     | configuration file
min_val    | 16
max_val    | 1073741823
enumvals   |
boot_val   | 1024
reset_val  | 16384
sourcefile | /usr/local/pgsql/data/postgresql.conf
sourceline | 115
```

pg_settings.context

`internal`

изменить нельзя, задано при установке

`postmaster`

перезапуск сервера

`sighup`

повторное считывание файлов конфигурации

`backend`

при запуске нового сеанса

`superuser`

во время выполнения, только суперпользователь

`user`

во время выполнения, любой пользователь

Во время выполнения

Получение значения параметров

```
SHOW conf_parameter;
```

```
current_setting('conf_parameter')
```

```
SELECT setting, unit FROM pg_settings
WHERE  name = 'conf_parameter';
```

Установка/сброс значений параметров

```
SET [LOCAL] conf_parameter TO value;
```

```
set_config('conf_parameter', 'value', [true|false])
```

```
UPDATE pg_settings
SET    setting = 'value'
WHERE  name = 'conf_parameter';
```

```
RESET conf_parameter | ALL;
```
-- сброс значений

установка параметров — транзакционная

Параметры БД и ролей

Установка параметров для базы данных и/или роли

```
ALTER DATABASE dbname SET conf_parameter TO value;

ALTER ROLE rolename [IN DATABASE dbname]
SET conf_parameter TO value;
```

только для новых подключений к БД и/или роли

Информация сохраняется в `pg_db_role_setting`

Удаление параметров для базы данных и/или роли

```
ALTER DATABASE dbname RESET conf_parameter | ALL;

ALTER ROLE rolename [IN DATABASE dbname]
RESET conf_parameter | ALL;
```

Команды ничего не изменяют в БД и/или роли

Параметры функций

Установка параметров для функций

```
ALTER FUNCTION funcname SET conf_parameter TO value;
```

Удаление параметров для функций

```
ALTER FUNCTION funcname RESET conf_parameter | ALL;
```

Действует на время работы функции

Информация сохраняется в `pg_proc.proconfig`

Параметры пользователя

Можно определять собственные параметры

Объявление и установка значения

`postgresql.conf`	применить `pg_reload_conf()`
`SET, SET_CONFIG()`	возможно создание во время выполнения

Получить значение

`SHOW, CURRENT_SETTING()`

не отображается в `pg_settings`

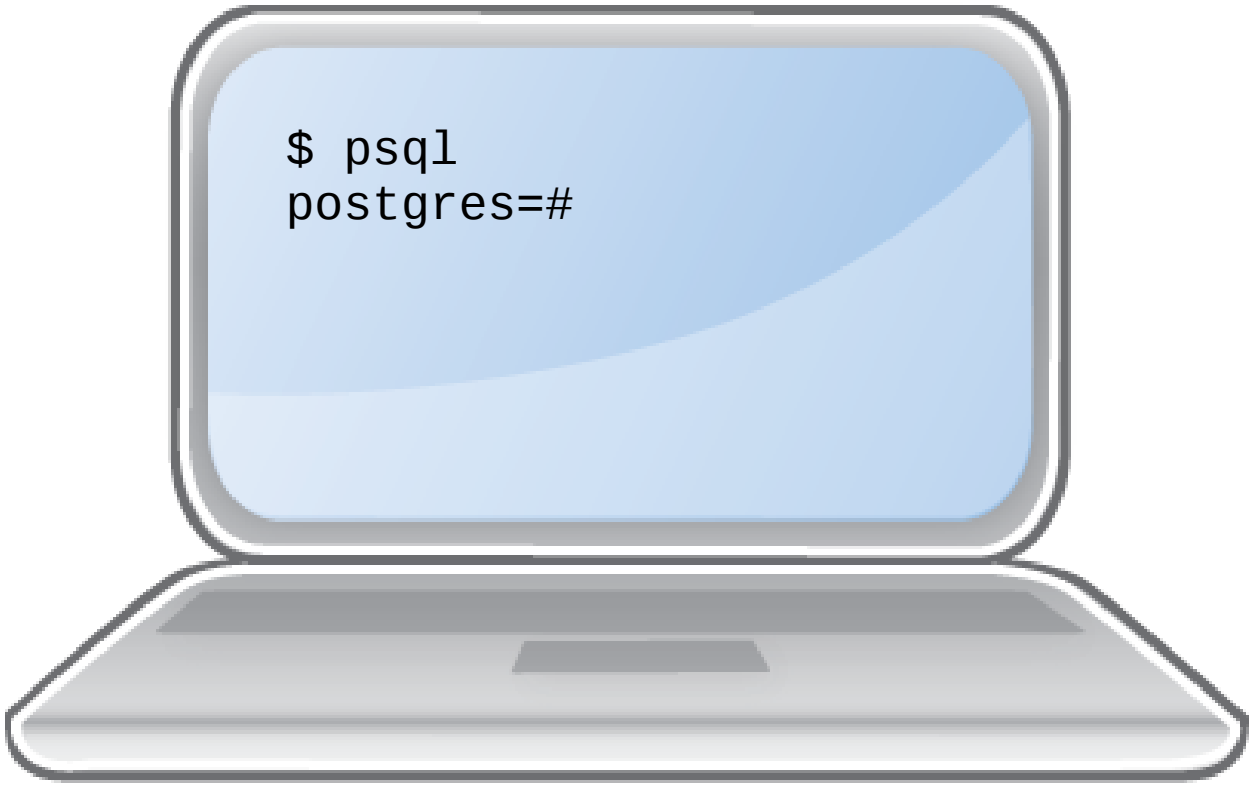
Формат имени: `domain.name`

Пример:

```
SET my_extension.param_name TO 'my_value';
```

Специальный параметр `application_name`

Демонстрация



С чего начать настройку

name	unit	boot_val	
listen_addresses		**localhost**	
shared_buffers	**8kB**	**1024**	**(8MB)**
work_mem	kB	4096	(4MB)
max_connections		100	
maintenance_work_mem	kB	65536	(64MB)
effective_cache_size	8kB	524288	(4GB)

С чего начать настройку

```
          name           | unit | boot_val
-------------------------+------+-----------
 listen_addresses        |      | localhost
 shared_buffers          | 8kB  | 1024          (8MB)
 work_mem                | kB   | 4096          (4MB)
 max_connections         |      | 100
 maintenance_work_mem    | kB   | 65536       (64MB)
 effective_cache_size    | 8kB  | 524288       (4GB)
```

Итоги

Рассмотрели файл конфигурации postgresql.conf

Узнали про настройку параметров через ALTER SYSTEM и файл postgresql.auto.conf

Рассмотрели системное представление pg_settings

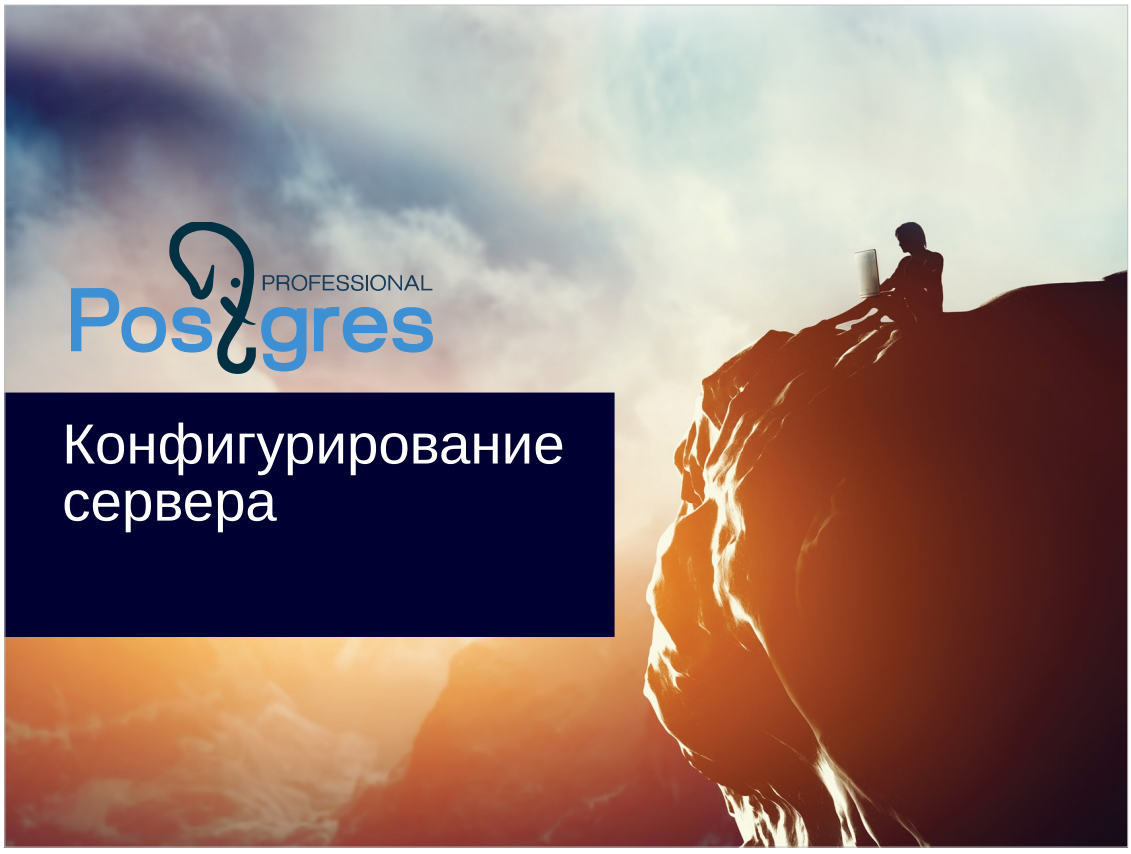
Познакомились с установкой/чтением параметров конфигурации во время исполнения

Узнали, как автоматически устанавливать параметры для новых подключений к БД и/или ролей, на время выполнения функций

Рассмотрели важные параметры для первоначальной настройки

Практика

1. Получить список параметров и их значений, для изменения которых требуется перезапуск сервера.
2. В файле postgresql.conf установите для параметра listen_addresses значение '*'.
3. Примените изменения в системе и убедитесь, что новые значения вступили в силу.
4. Используя команду ALTER SYSTEM установите значение параметра work_mem в 8 мегабайт.
5. Примените изменения в системе и убедитесь, что новые значения вступили в силу.
6. Для базы данных postgres установите значение параметра work_mem в 16 мегабайт.
7. Откройте новый сеанс с БД postgres.
   a) Проверьте значение work_mem.
   b) Установите в этом сеансе значение work_mem в 32 мегабайта.
   c) Как узнать, какое значение будет у work_mem после 'reset work_mem;'?
   d) Проверьте.



Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Темы

Файл postgresql.conf

Файл postgresql.auto.conf (ALTER SYSTEM)

Представление pg_settings

Работа с параметрами во время выполнения

Установка параметров для БД и/или ролей

Установка параметров для функций

С чего начать настройку

postgresql.conf

Основной файл конфигурации

Расположение

```
SHOW config_file;
```

по умолчанию в директории с данными, `$PGDATA`

Действия при изменении

файл читается один раз при старте сервера,
поэтому при изменении надо попросить сервер перечитать:

```
$ pg_ctl reload
$ kill -HUP
psql# select pg_reload_conf();
```

некоторые параметры требуют перезапуска сервера

Основной конфигурационный файл postgresql.conf по умолчанию расположен в директории с данными.

Это текстовый, хорошо документированный, файл, хранящий параметры в формате «ключ=значение».

Для вступления в силу внесенных в файл изменений, необходимо, чтобы сервер перечитал файл. Для некоторых параметров требуется перезагрузка сервера.

Подробную информацию о каждом конфигурационном параметре можно найти в документации:

http://www.postgresql.org/docs/current/static/runtime-config.html

postgresql.conf

Типы данных параметров

`fsync = on` (логическое значение)

`listen_addresses = 'localhost'` (строка)

`max_connections = 100` (целое число)

`autovacuum_vacuum_scale_factor = 0.2` (дробное число)

`shared_buffers = 128MB` (kB, MB, GB, TB)

`authentication_timeout = 1min` (ms, s, min, h, d)

`wal_level = minimal` (minimal, archive, hot_standby, logical)

Значения параметров могут быть одного из следующих типов:

- Логический
- Строковый
- Числовой (целое или дробное число)
- Размер памяти (в килобайтах, мегабайтах, и т.д)
- Интервал времени
- Список значений

postgresql.conf

Директивы включения дополнительных файлов

```
include filename
include_if_exists filename
include_dir dirname
```
(все *.conf файлы)

Разрешается вложенность include

Если параметр указан несколько раз, применяется последнее считанное значение

К файлу postgresql.conf при помощи специальных директив можно подключать дополнительные конфигурационные файлы.

include *filename* – подключает указанный файл

include_if_exists *filename* – подключает файл, при наличии файла и доступа к нему

include_dir *dirname* – подключает все файлы, заканчивающиеся на .conf, из указанной директории.

Если один и тот же параметр указан в конфигурационном файле (файлах) несколько раз, то использоваться будет значение считанное последним.

В версии 9.5 для удобного управления всеми конфигурационными файлами используется системное представление pg_file_settings:

http://www.postgresql.org/docs/devel/static/view-pg-file-settings.html
http://michael.otacoo.com/postgresql-2/postgres-9-5-feature-highlight-pg-file-settings/

postgresql.auto.conf

Файл конфигурации, расположен вместе с `postgresql.conf`

Считывается после `postgresql.conf`

Специальная команда для редактирования

```
ALTER SYSTEM
SET conf_parameter TO value;
```
добавляет или изменяет строки

```
ALTER SYSTEM
RESET conf_parameter | ALL;
```
удаляет строку или все строки

проверяет значения

сама по себе ничего не изменяет в системе

применение изменений — аналогично `postgresql.conf`

Самым последним считывается файл postgresql.auto.conf, расположенный там же где и postgresql.conf.

В отличии редактирования postgresql.conf, команда ALTER SYSTEM, перед записью в postgresql.auto.conf, выполняет проверку корректности введенного значения.

Команда ALTER SYSTEM выполняет только запись в файл postgresql.auto.conf и ничего не измененят в работающей системе. Для вступления изменений в силу нужно, чтобы сервер перечитал конфигурационные файлы.

Содержимое обоих файлов (posegresql.conf и postgresql.auto.conf) не может гарантировать того, что сервер PostgreSQL сейчас работает именно с такими значениями параметров.

Более подробная информация о команде ALTER SYSTEM:

http://www.postgresql.org/docs/current/static/sql-altersystem.html

При запуске сервера

Установка параметров при запуске сервера

```
$ pg_ctl start -o  (или postgres -c)
```

Имеют предпочтение перед файлами конфигурации

Для изменения требуется перезапуск сервера

Опции командной строки запуска сервера сохраняются в файле `postmaster.opts`

Параметры конфигурации можно задавать и в командной строке при запуске сервера. Пример:

```
$ pg_ctl start -l logfile -o "-c max_connections=50 -c shared_buffers=512MB"
```

Установленные таким образом параметры имеют предпочтение перед файлами конфигурации и могут быть изменены только при перезапуске сервера.

Чтобы узнать какие параметры были устрановлены в командной строке запуска сервера можно:

Выполнить запрос:

```
select * from pg_settings where source = 'command line';
```

Посмотреть содержимое файла postmaster.opts:

```
cat $PGDATA/postmaster.opts
/usr/local/pgsql/bin/postgres "-c" "max_connections=50" "-c" "shared_buffers=512MB"
```

Для удобства, в командной строке можно использовать сокращения для имен параметров:

http://www.postgresql.org/docs/current/static/runtime-config-short.html

Представление pg_settings

Параметр и значение

`name, setting, unit`

Описание

`category, short_desc, extra_desc`

Допустимые значения

`vartype, min_val, max_val, enumvals`

Значения по умолчанию

`boot_val, reset_val`

Источник

`source, sourcefile, sourceline`

Контекст

`context`

Представление pg_settings позволяет получить реальные действующие значения параметров конфигурации работающего сервера.

http://www.postgresql.org/docs/current/static/view-pg-settings.html

Пример

```
postgres=# select * from pg_settings where name = 'shared_buffers';
-[ RECORD 1 ]---------------------------------------------------------
name       | shared_buffers
setting    | 16384
unit       | 8kB
category   | Resource Usage / Memory
short_desc | Sets the number of shared memory buffers used by the server.
extra_desc |
context    | postmaster
vartype    | integer
source     | configuration file
min_val    | 16
max_val    | 1073741823
enumvals   |
boot_val   | 1024
reset_val  | 16384
sourcefile | /usr/local/pgsql/data/postgresql.conf
sourceline | 115
```

pg_settings.context

`internal`
: изменить нельзя, задано при установке

`postmaster`
: перезапуск сервера

`sighup`
: повторное считывание файлов конфигурации

`backend`
: при запуске нового сеанса

`superuser`
: во время выполнения, только суперпользователь

`user`
: во время выполнения, любой пользователь

Во время выполнения

Получение значения параметров

```
SHOW conf_parameter;
current_setting('conf_parameter')
SELECT setting, unit FROM pg_settings
WHERE  name = 'conf_parameter';
```

Установка/сброс значений параметров

```
SET [LOCAL] conf_parameter TO value;
set_config('conf_parameter', 'value', [true|false])
UPDATE pg_settings
SET    setting = 'value'
WHERE  name = 'conf_parameter';
RESET conf_parameter | ALL; -- сброс значений
```

установка параметров — транзакционная

Для получения значений параметров во время исполнения можно использовать три способа:

- Выполнение команды SHOW
- Вызов функции current_settings
- Запрос к представлению pg_settings

Для установки значений параметров также используются три способа:

- Выполнение команды SET
- Вызов функции set_config
- Выполнение UPDATE для представления pg_settings

Необязательное LOCAL в команде SET устанавливает значение только в рамках текущей транзакции. За это же отвечает третий параметр в функции set_config (true – для текущей транзакции)

Параметры БД и ролей

Установка параметров для базы данных и/или роли

```
ALTER DATABASE dbname SET conf_parameter TO value;
```

```
ALTER ROLE rolename [IN DATABASE dbname]
SET conf_parameter TO value;
```

только для новых подключений к БД и/или роли

Информация сохраняется в `pg_db_role_setting`

Удаление параметров для базы данных и/или роли

```
ALTER DATABASE dbname RESET conf_parameter | ALL;
```

```
ALTER ROLE rolename [IN DATABASE dbname]
RESET conf_parameter | ALL;
```

Команды ничего не изменяют в БД и/или роли

Возможна установка значений параметров на уровне базы данных, пользователя, комбинации пользователя и базы данных.

Для установки, изменения и удаления значений используются команды:

```
ALTER DATABASE … SET|RESET …
ALTER USER … [IN DATABASE ...] SET|RESET ...
```

Данные настройки будут действовать только для новых подключений к соответствующей базе данных и/или соответствующего пользователя.

http://www.postgresql.org/docs/current/static/sql-alterdatabase.html
http://www.postgresql.org/docs/current/static/sql-alterrole.html
http://www.postgresql.org/docs/current/static/catalog-pg-db-role-setting.html

Параметры функций

Установка параметров для функций

```
ALTER FUNCTION funcname SET conf_parameter TO value;
```

Удаление параметров для функций

```
ALTER FUNCTION funcname RESET conf_parameter | ALL;
```

Действует на время работы функции

Информация сохраняется в `pg_proc.proconfig`

Возможна установка параметров для отдельных функций. Такая установка будет действовать только на время работы функции.

Параметры пользователя

Можно определять собственные параметры

Объявление и установка значения

`postgresql.conf`	применить `pg_reload_conf()`
`SET, SET_CONFIG()`	возможно создание во время выполнения

Получить значение

`SHOW, CURRENT_SETTING()`

не отображается в `pg_settings`

Формат имени: `domain.name`

Пример:

```
SET my_extension.param_name TO 'my_value';
```

Специальный параметр `application_name`

Пользовательские параметры.

Приложения могут устанавливать дополнительные параметры для своих собственных целей. Например, различные расширения PostgreSQL при установке добавляют свои параметры в конфигурационный файл.

Для установки пользовательского параметра его можно добавить в файл postgresql.conf. Если сервер в это время работает, то для вступления в силу требуется перечитать конфигурацию.

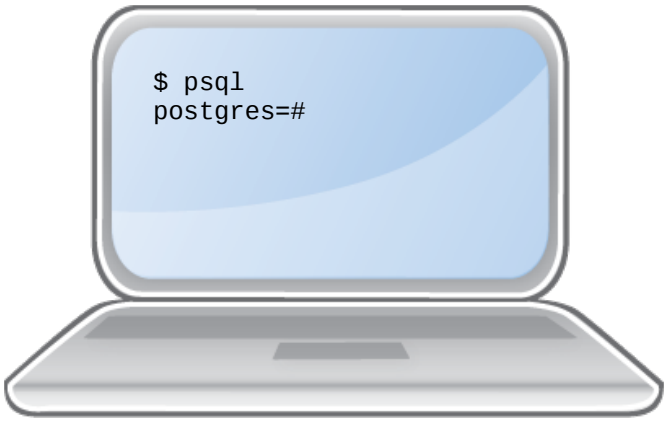
При дальнейшей работе, параметром можно управлять используя SET, SET_CONFIG() и SHOW, CURRENT_SETTING() для, соответственно, установки и получения значения.

Используя SET (без правки postgresql.conf), можно объявить новый параметр прямо во время выполнения. Это похоже на работу с глобальными переменными.

Имена таких параметров обязательно должны содержать точку (.). Слева от точки – имя приложения(расширения), справа – имя параметра.

Существует специальный параметр application_name, позволяющий приложениям устанавливать имя текущего приложения и контекст выполнения.

Демонстрация



С чего начать настройку

```
         name          | unit | boot_val
-----------------------+------+-----------
 listen_addresses      |      | localhost
 shared_buffers        | 8kB  | 1024        (8MB)
 work_mem              | kB   | 4096        (4MB)
 max_connections       |      | 100
 maintenance_work_mem  | kB   | 65536      (64MB)
 effective_cache_size  | 8kB  | 524288      (4GB)
```


Настройки параметров конфигурации по умолчанию (pg_settings.boot_val) рассчитаны на то, чтобы сервер PostgreSQL смог запуститься на практически любом оборудовании. Поэтому одной из первых задач администрирования является настройка на конкретном оборудовании под предполагаемую рабочую нагрузку.

Этой теме посвящено множество публикаций в сети (презентации, статьи, книги), хорошей начальной точкой может стать:

https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server

Далее рассмотрены несколько параметров, с которых следует начинать настройку:

- listen_addresses – список TCP/IP адресов, с которых сервер будет прослушивать подключения клиентских приложений. По умолчанию используется значение «localhost», разрешающее только локальные TCP/IP «loopback» соединения. Этот параметр можно установить в значение «*», что разрешает прослушивание клиентских подключений со всех сетевых интерфейсов, а затем управлять разрешениями на подключение в файле pg_hba.conf (рассматривается в теме «Подключение и аутентификация»).
- shared_buffers – количество памяти, выделяемое для разделяемой памяти. Значение с которого рекомендуется начинать настройку – 25% от общего размера оперативной памяти. Далее нужно смотреть на реальное использование shared_buffers в системе.

С чего начать настройку

```
         name          | unit | boot_val
-----------------------+------+-----------
 listen_addresses      |      | localhost
 shared_buffers        | 8kB  | 1024       (8MB)
 work_mem              | kB   | 4096       (4MB)
 max_connections       |      | 100
 maintenance_work_mem  | kB   | 65536     (64MB)
 effective_cache_size  | 8kB  | 524288     (4GB)
```

17

Продолжение:

- work_mem – количество оперативной памяти, которое выделяется серверному процессу для выполнения операций (например, сортировок). Чем больше значение, тем большие по объему операции будут выполняться в оперативной памяти. Однако значение этого параметра нужно рассматривать вместе с max_connections. Общее количество памяти, которое может быть выделено серверным процессам определяется как work_mem * max_connections.
- max_connections – максимальное количество одновременных подключений. Нужно смотреть вместе с work_mem. Важный фактор – будут ли использоваться программы для реализация пула соединений (например, pg_bouncer)
- maintenance_work_mem – количество памяти, которое отводится под обслуживающие процессы (vacuum).
- effective_cache_size – количество оперативной памяти, которое операционная система сможет выделить под дисковый кэш. Значение используется планировщиком для построения плана запроса. Не влияет на то, будет ли реально данное количество памяти выделяться операционной системой. Значение с которого рекомендуется начинать настройку – 50% от общего размера оперативной памяти.

Итоги

Рассмотрели файл конфигурации postgresql.conf

Узнали про настройку параметров через ALTER SYSTEM и файл postgresql.auto.conf

Рассмотрели системное представление pg_settings

Познакомились с установкой/чтением параметров конфигурации во время исполнения

Узнали, как автоматически устанавливать параметры для новых подключений к БД и/или ролей, на время выполнения функций

Рассмотрели важные параметры для первоначальной настройки

Практика

1. Получить список параметров и их значений, для изменения которых требуется перезапуск сервера.
2. В файле postgresql.conf установите для параметра listen_addresses значение '*'.
3. Примените изменения в системе и убедитесь, что новые значения вступили в силу.
4. Используя команду ALTER SYSTEM установите значение параметра work_mem в 8 мегабайт.
5. Примените изменения в системе и убедитесь, что новые значения вступили в силу.
6. Для базы данных postgres установите значение параметра work_mem в 16 мегабайт.
7. Откройте новый сеанс с БД postgres.
   a) Проверьте значение work_mem.
   b) Установите в этом сеансе значение work_mem в 32 мегабайта.
   c) Как узнать, какое значение будет у work_mem после 'reset work_mem;'?
   d) Проверьте.