



Сегодня хочу рассказать об установке и небольшой настройке сервера бд postgresql для работы с базами 1С. Задача не сложная, но есть небольшие нюансы как по настройке, так и по выбору дистрибутива. Существуют несколько модификаций postgresql для 1С. Я расскажу про установку двух из них — одна от самой компании 1С, вторая от postgrespro.

Теоретический курс по основам **сетевых технологий**. Позволит системным администраторам упорядочить и восполнить пробелы в знаниях. Цена очень доступная, есть бесплатный доступ. Все подробности по . Можно пройти тест на знание сетей, бесплатно и без регистрации.

Содержание:

- 1 Введение
- 2 Подготовка сервера к установке postgresql
- 3 Установка сборки PostgreSQL для платформы 1С от postgrespro
- 4 Установка сборки PostgreSQL от 1С
- 5 Настройка postgresql для увеличения производительности 1С
- 6 Заключение
- 7 Дополнительные материалы по Debian

Данная статья является частью единого цикла статей про сервер Debian.

Введение

Расскажу немного о возможности работы 1С, при которой можно сэкономить деньги на лицензиях, используя дистрибутив linux и бесплатную бд postgresql. Сервер 1С обычно используют в одной из следующих связок:



- Сервер 1C на Windows + MSSQL Server. Тут все понятно, самая распространенная и самая дорогая связка. Требуется лицензия для MSSQL сервера.
- Сервер 1C на Linux + PostgreSQL. Вариант с максимальным использованием бесплатного ПО. Я давно не использовал эту связку. Последний раз тестировал года 4 назад и мне не понравилась работа 1C сервера на linux. Я использовал дистрибутив Debian. Настройка не сильно сложная, но и не сказать, что простая. Надо ставить драйвера для hasp, чтобы сервер увидел ключ. Сервер работал не стабильно, иногда не получалось сделать выгрузку базы, приходилось перезапускать службу сервера. В целом работало, но не очень надежно. Мне тогда довелось пообщаться плотно с компанией, которая на постоянной основе использовала такую связку и админ тоже жаловался, что приходится сервер 1C перезапускать каждую ночь, тогда более ли менее стабильно все работает. С тех пор я не тестировал работу 1C на линуксе, так что про текущее положение дел ничего сказать не могу.
- Сервер 1C на Windows + PostgreSQL. Более привычный для настройки и управления вариант. Сервер ставится как обычно на windows машину, настраивается традиционным способом, а в качестве бд выступает бесплатная postgresql. О такой связке я и хочу сегодня рассказать.

Стандартная postgresql не будет нормально работать с базами 1C. В чем там проблема, я не знаю, не разобрался. Для работы с 1C используют отдельные сборки, специально под это заточенные. Мне известны 3 сборки postgresql для работы с 1C:

1. От Etersoft — <http://etersoft.ru/products/postgre>
2. От Postgrespro — https://postgrespro.ru/products/1c_build
3. И от самой 1C. Скачать можно через портал поддержки пользователей.

Я установлю и настрою для сравнения сборку от 1C и от Postgrespro, чтобы попытаться понять, где будет выше производительность. Настраивать все будем на сервере **Debian 8**. Если у вас еще нет настроенного сервера, то рекомендую мои статьи по установке и настройке debian. Я создал 2 одинаковые виртуальные машины для теста и установил на каждую из них разную версию базы данных.

Подготовка сервера к установке postgresql

Если вы читали мою статью по настройке debian, то наверно уже обновили систему. Если нет, то сделаем это:

```
# apt-get update && apt-get upgrade
```

Скофигурируем локали. Нам необходимо, чтобы в системе были 2 локали: **en_US.UTF-8** и **ru_RU.UTF-8**. При этом русская должна быть установлена по умолчанию. Выполняем конфигурацию с помощью команды:

```
# dpkg-reconfigure locales
```



Отмечаем указанные выше локали и выбираем ru_RU.UTF-8 дефолтной.



Проверить список установленных локалей можно командой:

```
# locale -a
```

Устанавливаем необходимые пакеты:

```
# apt-get install ssl-cert libossp-uuid16 libxslt1.1
```

Устанавливаем вручную еще один пакет:

```
# wget http://ftp.ru.debian.org/debian/pool/main/i/icu/libicu48_4.8.1.1-12+deb7u3_amd64.deb  
# dpkg -i libicu48*.deb
```

Редактируем системный параметр **kernel.shmmax**. Какое точно значение для наилучшей производительности выставлять, я не знаю. Не смог найти однозначного ответа. Видел рекомендацию, что в половину оперативной памяти на сервере. Я сделал у себя именно так. Вычислить размер параметра можно по формуле: Mb*1024*1024. У меня виртуальный сервер с 8Gb памяти, значение параметра получается $4096*1024*1024 = 4294967296$. Добавляем это значение в */etc/sysctl.conf*.

```
kernel.shmmax = 4294967296
```

Применяем внесенное изменение:

```
# sysctl -p
```

Проверить текущее значение параметра можно командой:



```
# cat /proc/sys/kernel/shmmax
```

У нас все готово для установки непосредственно postgresql.

Установка сборки PostgreSQL для платформы 1C от postgrespro

Будем устанавливать самую свежую сборку на момент написания статьи — PostgreSQL 9.6.1. Подключаем репозиторий:

```
# sh -c 'echo "deb http://1c.postgrespro.ru/deb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/postgrespro-1c.list'
```

Устанавливаем gpg ключ

```
# wget --quiet -O - http://1c.postgrespro.ru/keys/GPG-KEY-POSTGRESPRO-1C | apt-key add -
```

Обновляем список пакетов:

```
# apt-get update
```

Устанавливаем PostgreSQL

```
# apt-get install postgresql-pro-1c-9.6
```

Запускаем базу данных и добавляем в автозагрузку:

```
# systemctl start postgresql  
# systemctl enable postgresql
```

Установка сборки postgresql для работы с 1C от postgrespro закончена. Можно начинать работать с базой данных. Но перед этим необходимо задать пароль суперпользователя базы данных и выполнить небольшой тюнинг для увеличения быстродействия базы данных. С дефолтными настройками 1C будет



работать очень медленно. Настройкой мы займемся позже, после того, как поставим вторую сборку от 1C.

Установка сборки PostgreSQL от 1C

Свежую версию postgresql от 1C можно скачать в пользовательском разделе на сайте 1C по адресу <https://releases.1c.ru/total>



Мы устанавливаем postgres на сервер Debian, поэтому скачиваем deb пакет для нашей системы.



Если у вас нет доступа к порталу 1C, можете забрать актуальный файл на момент написания статьи у меня — <https://yadi.sk/d/hWi1PMBd1B7QIA>.

Передаем файл на сервер, например через sftp или ssh. Распаковываем содержимое архива в отдельную папку:

```
# mkdir postgres
# tar -xvf postgresql-9.4.2-1.1C_amd64_deb.tar.bz2 -C postgres
```

В архиве были следующие файлы:

```
# cd postgres
# ls -l
```

```
итого 5420
-rw-r--r-- 1 120250 июн 2 2015 libpq5_9.4.2-1.1C_amd64.deb
-rw-r--r-- 1 3657036 июн 2 2015 postgresql-9.4_9.4.2-1.1C_amd64.deb
-rw-r--r-- 1 1069692 июн 2 2015 postgresql-client-9.4_9.4.2-1.1C_amd64.deb
-rw-r--r-- 1 72172 авг 11 2015 postgresql-client-common_154.1.1C_all.deb
-rw-r--r-- 1 163684 авг 11 2015 postgresql-common_154.1.1C_all.deb
-rw-r--r-- 1 457782 июн 2 2015 postgresql-contrib-9.4_9.4.2-1.1C_amd64.deb
```



Устанавливаем все пакеты:

```
# dpkg -i *.deb
```

Все, установка postgresql от 1C закончена. Запускаем и добавляем в автозагрузку:

```
# systemctl start postgresql  
# systemctl enable postgresql
```

Теперь приступаем к настройке для увеличения производительности базы 1C.

Настройка postgresql для увеличения производительности 1C

Настройка это, конечно, громко сказано. Я не занимался серьезно тюнингом postgresql и тестированием. Бегло поискал в интернете информацию и использовал наиболее популярные советы. Так что мои рекомендации не претендуют на реально проверенные и надежные решения. Используйте на свое усмотрение. Я просто делюсь тем, что сам нашел.

Перед изменением конфига, зададим пароль административной учетной записи нашего сервера бд. Для этого заходим в систему от пользователя postgres:

```
# su postgres
```

Меняем пароль внутреннего пользователя бд postgres:

```
# /usr/bin/psql -U postgres -c "alter user postgres with password 'postgrespwd';"
```

postgrespwd — пароль пользователя для административного доступа к базе данных. Эту учетную запись необходимо использовать для добавления новой базы в Сервер 1C.

Возвращаемся к тюнингу postgres. В первую очередь рекомендую сервис pg tune — <http://pgtune.leopard.in.ua>. В нем можно указать характеристики сервера и получить рекомендации по настройке. Так как у нас 2 сервера разных версий — первый 9.6, второй 9.4, рекомендации будут немного отличаться. Имейте



это ввиду. Вот пример рекомендаций для первого сервера от postgrespro:



Укажите эти параметры в файле конфигураций. В моем случае это файл `/etc/postgresql/9.6/main/postgresql.conf`. Мне знакомый скинул готовый файл настроек с рекомендуемыми параметрами для 1С. Я не знаю происхождение этого файла, но настройками воспользовался. В итоге с учетом этого конфига и сайта **pgtune** я получил вот такой файл конфигурации postgres:

```
data_directory = '/var/lib/postgresql/9.6/main'
hba_file = '/etc/postgresql/9.6/main/pg_hba.conf'
ident_file = '/etc/postgresql/9.6/main/pg_ident.conf'
external_pid_file = '/var/run/postgresql/9.6-main.pid'
listen_addresses = '*'
port = 5432
max_connections = 10
unix_socket_directories = '/var/run/postgresql'
ssl = true
ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'
ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key'
shared_buffers = 2GB
temp_buffers = 32MB
work_mem = 104857kB
maintenance_work_mem = 512MB
autovacuum_work_mem = 64MB
dynamic_shared_memory_type = posix
shared_preload_libraries = 'online_analyze, plantuner'
bgwriter_delay = 10ms
bgwriter_lru_maxpages = 1000
bgwriter_lru_multiplier = 10
fsync = on
synchronous_commit = off
wal_buffers = 16MB
```



```
checkpoint_timeout = 55min
max_wal_size = 2GB
min_wal_size = 1GB
checkpoint_completion_target = 0.9
effective_cache_size = 6GB
geqo = off
default_statistics_target = 100 # range 1-10000
default_statistics_target = 10
cursor_tuple_fraction = 0.0001
from_collapse_limit = 2
join_collapse_limit = 2
log_line_prefix = '%t [%p-%l] %q%u@%d '
log_timezone = 'localtime'
cluster_name = '9.6/main'
stats_temp_directory = '/var/run/postgresql/9.6-main.pg_stat_tmp'
log_autovacuum_min_duration = 0
autovacuum_max_workers = 3
autovacuum_naptime = 15s
autovacuum_vacuum_threshold = 90
autovacuum_analyze_threshold = 9
autovacuum_vacuum_scale_factor = 0.0002
autovacuum_analyze_scale_factor = 0.0001
autovacuum_vacuum_cost_delay = 10ms
autovacuum_vacuum_cost_limit = 10000
datestyle = 'iso, dmy'
timezone = 'localtime'
lc_messages = 'ru_RU.UTF-8'
lc_monetary = 'ru_RU.UTF-8'
lc_numeric = 'ru_RU.UTF-8'
lc_time = 'ru_RU.UTF-8'
default_text_search_config = 'pg_catalog.russian'
max_locks_per_transaction = 150
max_pred_locks_per_transaction = 150
```




```
backslash_quote = on  
escape_string_warning = off  
standard_conforming_strings = off  
online_analyze.threshold = 50  
online_analyze.scale_factor = 0.1  
online_analyze.enable = on  
online_analyze.verbose = off  
online_analyze.min_interval = 10000  
online_analyze.table_type = 'temporary'  
plantuner.fix_empty_table = false
```

Не забудьте перезапустить postgresql после изменения настроек:

```
# systemctl restart postgresql
```

Для версии 9.4 настройки будут немного отличаться. Там как минимум не будет параметров `min_wal_size` и `max_wal_size`. Но есть и другие отличия. Призываю не копировать мой конфиг и вставлять к себе, а сравнивать мои параметры и ваш дефолтный конфиг и изменять непосредственно значения параметров.

Есть хороший материал на тему оптимизации postgresql для работы с 1C — <http://nixway.org/2015/12/06/optimizacija-postgresql-pod-server-1c-predpriyatiya>. Я не буду повторяться и копировать информацию. Можете сами ознакомиться.

Заключение

В моем случае база 1C работала примерно одинаково в обоих случаях. В итоге я решил остановиться на сборке от postgrespro, так как там более свежая версия 9.6. У меня была база 10 гб. Работала она приемлемо, кроме некоторых операций — когда идет чтение по всей базе для построения каких-нибудь списков. Тюнинг различных параметров не привел к уменьшению времени выполнения подобных запросов. Так что этот вопрос остается открытым. Простого решения найти не удалось. Что точно нужно изменить для ускорения работы подобных запросов — не знаю. Разбор этой ситуации описал в отдельном материале — Ускорение работы 1C с postgresql и диагностика проблем производительности.

Заметил, что во время выполнения запроса на 100% нагружено одно ядро процессора. Погуглил тему, стало ясно, что это особенность архитектуры. Один запрос обрабатывает одно ядро. Простых способов решения данной проблемы не существует. Более подробно я не стал разбираться. На ум сразу пришло



установить какой-то кэш, чтобы сохранять тяжелые запросы. Нашел информацию про **pg_bouncer**. Установить не получилось, он сходу из пакетов не встал с моей версией от postgrespro, ругнулся на зависимости. Надо разбираться и собирать из исходников. Нет уверенности, что это поможет, не стал заморачиваться, так как нет много времени на решение этой задачи. Буду рад любым подсказкам и комментариям по теме.

Если вы решите использовать описанное решение в продакшене, вам пригодится материал на тему бэкапа и восстановления баз postgresql.

Напоминаю, что данная статья является частью единого цикла статьей про сервер Debian.

Онлайн курс Основы сетевых технологий

Теоретический курс с самыми **базовыми знаниями по сетям**. Курс подходит и начинающим, и людям с опытом. Практикующим системным администраторам курс поможет упорядочить знания и восполнить пробелы. А те, кто только входит в профессию, получают на курсе базовые знания и навыки, без воды и избыточной теории. После обучения вы сможете ответить на вопросы:

- На каком уровне модели OSI могут работать коммутаторы;
- Как лучше организовать работу сети организации с множеством отделов;
- Для чего и как использовать технологию VLAN;
- Для чего сервера стоит выносить в DMZ;
- Как организовать объединение филиалов и удаленный доступ сотрудников по vpn;
- и многое другое.

Уже знаете ответы на вопросы выше? Или сомневаетесь? Попробуйте пройти тест по основам сетевых технологий. Всего 53 вопроса, в один цикл теста входит 10 вопросов в случайном порядке. Поэтому тест можно проходить несколько раз без потери интереса. Бесплатно и без регистрации. Все подробности на странице .

Дополнительные материалы по Debian



Рекомендую полезные материалы по Debian:

Настройки системы

- Установка
- **Базовая настройка**
- Настройка сети
- Обновление 8 до 9
- Обновление 7 до 8
- Включение логов cron

Подробная установка Debian 9 Stretch с помощью графического инсталлятора со скриншотами и пояснениями к каждому пункту установщика.

Базовая настройка сервера Debian. Приведены практические советы по улучшению безопасности и удобства администрирования.

Подробное описание настройки сети в Debian - задать ip адрес, dhcp, отключить ipv6, dns, hostname, статические маршруты и др.

Обновление предыдущей версии Debian 8 Jessie до последней Debian 9 Stretch. Подробная инструкция с описанием по каждому этапу обновления.

Обновление версии Debian 7 wheezy до Debian 8 Jessie. Подробная инструкция с описанием по каждому этапу обновления.

Включение записи логов cron в Debian в отдельный файл и настройка ротации этого файла. Отключение логов в syslog.

Настройка программных комплексов

- Proxmox
- Шлюз в интернет
- Установка Asterisk
- Asterisk+Freepbx
- PostgreSQL для 1С
- Настройка pptp

Подробное описание установки гипервизора proxmox на raid1 mdadm на базе операционной системы Debian 8. Приведены практические советы по настройке.

Настройка интернет шлюза на Debian. Включает в себя настройку iptables, nat, dhcp, dns, iftop.

Чистая установка Asterisk 13 на сервер под управлением Debian 8. Никаких дополнений и GUI, только vanilla asterisk.

Установка Freepbx 12 и Asterisk 13 на сервер под управлением Debian/Ubuntu. Подробное описание и разбор ошибок установки.

Рассказ об установке и небольшой настройке сервера бд postgresql для работы с базами 1С. Задача не сложная, но есть небольшие нюансы как по настройке, так и по выбору дистрибутива.

Описание установки и настройки pptp сервера в Debian с передачей статических маршрутов клиенту для организации доступа к ресурсам сети.

Разное



- Бэкап с помощью rsync
- Тюнинг postgresl для 1C

Подробное описание настройки бэкапа с помощью rsync на примере скрипта инкрементного архива на системе Centos, Debian, Ubuntu, Windows.
Ускорение работы 1C с postgresql и диагностика проблем производительности

Помогла статья? Есть возможность отблагодарить автора