

# Высокая доступность и балансировка нагрузки

Муравьёв С.К.

Национальный исследовательский ядерный университет «МИФИ»  
Кафедра №36 «Информационные системы и технологии»

1 ноября 2013

# Резервное копирование

Резервное копирование (англ. backup) — процесс создания копии данных, для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

Резервное копирование в PostgreSQL:

- SQL дамп
- Архивирование файлов кластера
- Постоянное архивирование

# SQL дамп

# Архивирование файлов кластера

# Журнал опережающей записи

- Перед выполнением любого изменения в базе данных СУБД делает соответствующую запись в Журнал опережающей записи WAL (write ahead log).
- Если в процессе изменения самой базы данных произойдёт сбой, то база данных может быть возвращена в целостное состояние путём “проигрывания” записей из WAL, сделанных после последней контрольной точки.
- Все изменения в базе данных, сделанные до контрольной точки, гарантировано записаны на диск.

# Постоянное архивирование

# Репликация

Репликация (англ. replication) — механизм синхронизации содержимого нескольких копий базы данных.

Различают синхронную репликацию, когда все реплики обновляются в одной транзакции, и асинхронную, когда обновления одной реплики распространяются на другие с некоторой задержкой.

Репликация в PostgreSQL:

- Поточковая репликация
- Каскадная репликация
- Синхронная репликация

# Потоковая репликация

Резервный сервер соединяется с основным, и получает от него записи журнала WAL по мере их появления, не дожидаясь заполнения сегментов WAL.



# Настройка основного сервера

Остановка основного сервера:

```
#service postgresql-9.3 stop
```

/var/lib/pgsql/9.3/data/postgresql.conf

```
listen_address = '*'  
wal_level = hot_standby  
max_wal_senders = 2
```

/var/lib/pgsql/9.3/data/pg\_hba.conf

```
host replication all 192.168.0.2/32 trust
```

# Настройка резервного сервера

Остановка резервного сервера:

```
#service postgresql-9.3 stop
```

/var/lib/pgsql/9.3/data/postgresql.conf

```
listen_address = '*'  
hot_standby = on
```

/var/lib/pgsql/9.3/data/recovery.conf

```
standby_mode = 'on'  
primary_conninfo = 'host=192.168.0.1'
```

# Запуск потоковой репликации

Копирование данных с основного сервера на резервный:

```
#rsync -av --exclude pg_xlog --exclude postgresql.conf  
data/* 192.168.0.2:/var/lib/pgsql/9.3/data/
```

Запуск резервного сервера:

```
#service postgresql-9.3 start
```

Запуск основного сервера:

```
#service postgresql-9.3 start
```

# Мониторинг репликации на основном сервере

Текущая позиция записи в WAL

```
SELECT pg_current_xlog_location();
```

Сводная таблица со статистикой:

```
SELECT * FROM pg_stat_replication;
```

# Мониторинг репликации на резервном сервере

Признак нахождения в режиме восстановления

```
SELECT pg_is_in_recovery();
```

Позиция последней полученной записи WAL

```
SELECT pg_last_xlog_receive_location();
```

Позиция последней восстановленной записи WAL

```
SELECT pg_last_xlog_replay_location();
```

Время восстановления последней транзакции

```
SELECT pg_last_xact_replay_timestamp();
```

# Каскадная репликация

Каскадная репликация позволяет резервным серверам пересылать поступающие записи WAL на другие серверы, что позволяет уменьшить число прямых подключений к ведущему серверу и снизить нагрузку на внешние каналы связи.

Настройка осуществляется также как и в случае с репликацией между основным и резервным серверами.

# Синхронная репликация

Помимо настроек, необходимых для асинхронного режима, для синхронной репликации необходимо сделать следующее:

- Присвоить имена резервным серверам в файле `/var/lib/pgsql/9.3/data/recovery.conf`:

```
primary_conninfo='application_name=srv host=192.168.0.1'
```

- В файле `/var/lib/pgsql/9.3/data/recovery.conf`, на основном сервере, перечислить имена резервных серверов на которые может выполняться синхронная репликация<sup>1</sup>:

```
synchronous_standby_names='srv1;srv2;...'
```

---

<sup>1</sup>В каждый момент времени синхронная репликация может выполняться только на один сервер

- Настроить потоковую репликацию между двумя серверами
- С помощью функций мониторинга убедиться в работе репликации
- Проверить возможность выполнения запросов на чтение и запись на основном и резервном серверах
- Настроить аутентификацию резервного сервера по паролю
- Настроить каскадную репликацию с резервного сервера на дополнительный резервный сервер
- Настроить синхронную репликацию
- Настроить и активировать межсетевые экраны



# Менеджеры репликации

- Slony
- pgpool-II
- Bucardo

Slony — система асинхронной репликации в режиме ведущий-ведомый для СУБД PostgreSQL, обладающая следующими основными возможностями:

- Поддержка нескольких резервных узлов
- Репликация между разными версиями PostgreSQL
- Репликация между разными аппаратными платформами и операционными системами
- Поддержка частичной репликации, то есть репликации только определённого набора таблиц
- Поддержка репликации разных таблиц на разные резервные узлы
- Поддержка репликации разных таблиц с разных мастеров на один резервный узел

pgpool-II — прослойка, работающая между серверами и клиентами СУБД PostgreSQL. Она предоставляет следующие функции:

- Управление пулами соединений
- Репликация
- Балансировка нагрузки
- Обеспечение высокой доступности
- Ограничение лишних соединений
- Параллельные запросы

# Установка pgpool

```
#yum install -y pgpool-II-93
```

Bucardo — система асинхронной репоикации в режиме мультимастер и ведущий-ведомый для СУБД PostgreSQL, обладающая следующими основными возможностями:

- Балансировка нагрузки
- Обеспечение высокой доступности

# Отказоустойчивость

Отказоустойчивость — это свойство технической системы сохранять свою работоспособность после отказа одного или нескольких составных компонентов.

При построении отказоустойчивого кластера СУБД необходимо решить ряд задач:

- В случае отказа основного сервера резервный должен автоматически и достаточно быстро принять на себя всю нагрузку
- Необходим механизм включения нового резервного сервера в кластер и восстановления работоспособности отказавших серверов
- После восстановления работоспособности бывшего основного сервера, он должен быть предупреждён о наличии нового основного сервера
- Переключение серверов должно происходить прозрачным для клиентов способом

# Механизмы обеспечения отказоустойчивости в PostgreSQL

- Ручное переключение ведущего сервера

```
#/usr/pgsql-9.3/bin/pg_ctl promote
```

- Автоматическое переключение ведущего сервера

/var/lib/pgsql/9.3/data/recovery.conf:

```
trigger_file = '/trigger/file/name'
```

# Балансировка нагрузки

Балансировка нагрузки (англ. load balancing) — распределение задач между несколькими серверами с целью оптимизации использования ресурсов и сокращения времени выполнения запросов.



# Партиционирование данных