

1. サーバ設定

1.1 IP アドレス設定

サーバは以下の 6 台を用いる。

- pgpool-II サーバ 2 台
- DB サーバ 3 台
- 上位サーバ 1 台

OS は全サーバで CentOS 6.4 x86_64 とする。

pgpool-II のサーバは NIC を 3 つ持っているとする(eth0, eth1, eth2)。

このうち、eth0 をサービス提供と DB 接続。残りの eth1, eth2 をハートビート通信回線に用いる。

各サーバの /etc/sysconfig/ifcfg-ethX を編集して、以下の固定 IP アドレスを割り当てる。ネットマスクは全て 255.255.255.0 とする。

pgpool-II サーバ1:

eth0: 192.168.100.101
eth1: 192.168.98.101
eth2: 192.168.99.101

pgpool-II サーバ 2:

eth0: 192.168.100.102
eth1: 192.168.98.102
eth2: 192.168.99.102

上位サーバ: 192.168.100.10

DB サーバ 1: 192.168.100.11

DB サーバ 2: 192.168.100.12

DB サーバ 3: 192.168.100.13

また、仮想 IP アドレスとして 192.168.100.200 を用いる。

1.2 事前に必要なソフトウェアのインストール

pgpool-II サーバ、DB サーバには PostgreSQL 9.3 を rpm でインストールしておく。

```
# wget http://yum.postgresql.org/9.3/redhat/rhel-6-x86_64/pgdg-centos93-9.3-1.noarch.rpm
# rpm -U pgdg-centos93-9.3.1.noarch.rpm
# yum install postgresql93 postgresql93-server
```

postgres ユーザのホームディレクトリは全サーバで同じ(/var/lib/pgsql)であることを前提とする。

postgres ユーザのパスワードを postgres にしておく。

pgpool-II サーバには以下がインストールされていることを確認し、もしなければインストールしておく。
(これらは全て、pgpoolAdmin のインストールのために必要)

- httpd
- php-pgsql
- php-mbstring
- php

1.3 その他設定

全サーバで iptables, SELinux は無効にしておく。

ntpd を設定し、時刻を合わせておく。(上位サーバ 192.168.100.10 を基準とする。)

各サーバのホスト名 (hostname コマンドの実行結果) で、他のサーバからアクセスできるようにしておく。

- 各サーバの /etc/hosts に以下のホスト名を追加する。

```
192.168.100.101 pgpool_server1
192.168.100.102 pgpool_server2
192.168.100.10 trusted_server
192.168.100.11 db_server1
192.168.100.12 db_server2
192.168.100.13 db_server3
```

- 各サーバの /etc/sysconfig/network を編集して、HOSTNAME に各々のホスト名を設定する。

2. pgpool-II サーバの設定

2.1 インストーラの実行

pgpool-II はコミュニティ wiki¹で配布されているインストーラ (installer-pg93-3.3.2.tar.gz) を使ってインストールする。インストーラに含まれる install.sh を root ユーザで実行する。

まず、pgpool-II サーバ 1 で install.sh を実行する。

- node 0 には 192.168.100.101 を指定
- node 1 には 192.168.100.102 を指定
- Is this node 0? には yes と入力
- 途中何度か postgres のパスワード入力を求められるので、postgres と入力
- pgpoolAdmin のユーザ名/パスワードは、両方共 admin と入力
- replication mode は stream と入力
- load balancing, query cache, watchdog には全て yes と入力
- delegate IP address には 192.168.100.200 と入力
- method of watchdog lifecheck は heartbeat と入力
- pgpoolAdmin の言語は ja と入力
- その他は、全てデフォルト値 or yes と入力

次に、pgpool-II サーバ 2 で install.sh を実行する。

- node 0 には 192.168.100.101 を指定
- node 1 には 192.168.100.102 を指定
- Is this node 0? には no と入力
- 途中何度か postgres のパスワード入力を求められるので、postgres と入力
- あとの質問には全て yes と入力

1 <http://www.pgpool.net>

2.1 設定済みの内容

インストーラにより、apache ユーザから pgpool-II を利用するための環境が設定されている。
(apache ユーザなのは、pgpoolAdmin の利用を前提としているため)

2.1.1 自動フェールオーバー(設定済み)

インストーラにより以下のスクリプトが作られており、設定済みなので、そのまま利用する。

- /etc/pgpool-II/failover.sh
 - pgpool.conf の failover_command で指定されている。
 - DB ノードがダウンしたときに実行されるスクリプト。
 - プライマリ DB がダウンした場合には、pg_ctl promote を実行してスタンバイ DB を新プライマリに昇格させる。
 - フェイルオーバーのログは /var/log/pgpool/failover.log に出力される。

2.1.2 ifconfig/arping 実行権限の設定(設定済み)

watchdog では仮想 IP の制御に ifconfig/arping を用いるため、部分的に root 権限が必要になる。
インストーラにより自動的に setuid が設定された ifconfig/arping が用意され、apache ユーザのみがアクセスできるディレクトリ(/var/private/nobody)に格納されているのでこれをそのまま利用する。

2.2 設定の変更

バックエンド、上位サーバ、watchdog のハートビート信号の設定を変更する。

/etc/pgpool-II/pgpool.conf を以下のように編集する。

2.2.1 バックエンド情報を修正

```
backend_hostname0 = '192.168.100.11'
backend_port0 = 5432
backend_weight0 = 1
backend_data_directory0 = '/var/lib/pgsql/data'
#backend_flag0 = 'ALLOW_TO_FAILOVER'

backend_hostname1 = '192.168.100.12'
backend_port1 = 5432
backend_weight1 = 1
backend_data_directory1 = '/var/lib/pgsql/data'
#backend_flag1 = 'ALLOW_TO_FAILOVER'

#backend_hostname2 = '192.168.100.13'
#backend_port2 = 5432
#backend_weight2 = 1
#backend_data_directory2 = '/var/lib/pgsql/data'
#backend_flag1 = 'ALLOW_TO_FAILOVER'
```

※ 最後のバックエンド情報がコメントアウトされているのは、バックエンドを追加するテストケースのための準備。

2.2.2 上位サーバを追加

```
trusted_servers = '192.168.100.10'
```

2.2.3 ハートビート信号の送り先を修正

```
(pgpool-II サーバ 1)
heartbeat_destination0 = '192.168.98.102'
heartbeat_destination_port0 = 9694
heartbeat_device0 = ""
heartbeat_destination1 = '192.168.99.102'
heartbeat_destination_port1 = 9694
heartbeat_device1 = ""
```

```
(pgpool-II サーバ 2)
heartbeat_destination0 = '192.168.98.101'
heartbeat_destination_port0 = 9694
heartbeat_device0 = ""
heartbeat_destination1 = '192.168.99.101'
heartbeat_destination_port1 = 9694
heartbeat_device1 = ""
```

2.3 SSH の設定

pgpool-II サーバの apache ユーザから、各 DB サーバの postgres ユーザにパスワード無しで SSH ができるようにする。

各 pgpool-II サーバで以下を実行する。

```
# su - apache
$ ssh-copy-id postgres@192.168.100.11
$ ssh-copy-id postgres@192.168.100.12
$ ssh-copy-id postgres@192.168.100.13
```

各サーバの `~apache/.ssh/config` (または `/etc/ssh/ssh_config`) に以下を記載する。

```
StrictHostKeyChecking no
```

3 DB サーバの設定

3.1 pgpool-II が内部で使用する C 言語関数のインストール

全ての DB サーバで、インストーラに含まれる `pgpool-II-pg93-3.3.2-1.pgdg.x68_64.rpm` をインストールしておく。

3.2 ディレクトリの作成

全ての DB サーバで以下のディレクトリを作成する。

- WAL アーカイブディレクトリ

```
# mkdir -p /var/lib/pgsql/archivedir
# chown postgres.postgres /var/lib/pgsql/archivedir
```

- ログディレクトリ(オンラインリカバリ時のログが格納される)

```
# mkdir -p /var/log/pgpool
# chmod 777 /var/log/pgpool
```

3.3 設定ファイルのインストール

DB サーバ1(のみ)で以下の設定を行う。

1. 事前に、pgpool-II サーバの installer ディレクトリ内 edited に、インストーラにより作成された 設定ファイル一式があるので DB サーバ1の適当なディレクトリ(仮に files とする)にコピーしておく。
2. クラスタを作成する。

```
# su postgres
$ initdb -D ~postgres/data --no-locale -E UTF8
```

3. 作成された data フォルダに、files の中の postgresql.conf, pg_hba.conf, basebackup-stream.sh, pgpool_remote_start, config_for_script をコピーする。
4. basebackup-stream.sh, pgpool_remote_start のパーミッションは 755 に、config_for_script のパーミッションは 444 に変更する。
5. basebackup-stream.sh を以下のように編集

```
@@ -5,21 +5,12 @@
dest_host=$2
dest_dir=$3

-source /etc/pgpool-II/config_for_script
+source ./config_for_script

log=$PGPOOL_LOG_DIR/recovery.log

-if [ $dest_host = $NODE0_HOST ]; then
- master_host=$NODE1_HOST
- master_port=$NODE1_PORT
-
-elif [ $dest_host = $NODE1_HOST ]; then
- master_host=$NODE0_HOST
- master_port=$NODE0_PORT
-
-else
- exit 1
-fi
+master_host=$HOSTNAME
+master_port=5432
```

```
echo "-----" >> $log
date >> $log
```

6. pgpool_remote_start を以下のように編集

```
@@ -7,7 +7,7 @@
    exit 1
fi

-source /etc/pgpool-II/config_for_script
+source ./config_for_script

dest_host=$1
dest_dir=$2
```

なお、各スクリプトの役割は以下の通り。

- /var/lib/pgsql/data/basebackup-stream.sh
 - pgpool.conf の recovery_1st_stage_command オプションで指定されている。
 - DB ノードをオンラインリカバリするときに実行されるスクリプト
 - プライマリ DB のベースバックアップを取得する
 - ログは /var/log/pgpool/recovery.log に出力される
- /var/lib/pgsql/data/pgpool_remote_start
 - DB ノードをオンラインリカバリした直後に実行されるスクリプト
 - リカバリ後の DB を再起動させる。

3.4 DB の設定

DB サーバ1(のみ)で以下の SQL を実行する。

- admin user (super user) を作成

```
# CREATE USER admin PASSWORD 'admin' SUPERUSER;
```

- EXTENSION をインストール

```
psql -p 5432 -U postgres template1 -c "CREATE EXTENSION pgpool_regclass;"
psql -p 5432 -U postgres template1 -c "CREATE EXTENSION pgpool_recovery;"
```

3.5 SSH の設定

DB サーバの postgres ユーザから、他の DB サーバの postgres ユーザにパスワード無しで SSH できるようにする。

- 各 DB サーバで 以下を実行

```
# su - postgres
$ ssh-copy-id postgres@192.168.100.11
$ ssh-copy-id postgres@192.168.100.12
```

```
$ ssh-copy-id postgres@192.168.100.13
```

- `~postgres/.ssh/config` (または `/etc/ssh/ssh_config`) に以下を記載

```
StrictHostKeyChecking no
```

3.6 DB サーバ 2 をオンラインリカバリで作成

3.6.1 DB サーバ 1 で PostgreSQL を起動

```
$ /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/pgsql/data start
```

3.6.2 各 pgpool-II サーバで pgpool-II を起動

ブラウザで以下の URL、ユーザ、パスワードで pgpoolAdmin にログインし、GUI から pgpool-II を起動する。

- URL: `http://192.168.100.101/pgpoolAdmin/` (`http://192.168.100.102/pgpoolAdmin/`)
- ユーザ: `admin`
- パスワード: `admin`

3.6.3 DB サーバ 2 をオンラインリカバリ

pgpoolAdmin のリカバリボタンを押下する。または以下の様に、pcp コマンドでリカバリする。

```
$ pcp_recovery_node 0 localhost 9898 admin admin 1
```

これにより DB サーバ 2 がスレーブとして起動する。