

PL/Proxy と PgBouncer のハンズオン・ストーリー

2009年6月20日

JPUG しくみ分科会

1. ppt で pl/proxy と pgbouncer の概要(概念)を説明

<https://developer.skype.com/SkypeGarage/DbProjects/PgBouncer>

<https://developer.skype.com/SkypeGarage/DbProjects/PlProxy>

2. OpenSolaris LiveCD の起動

概要は OSC2009Tokyo/Spring の冒頭。

3. libXm.so.4 と起動スクリプトによる接続設定

http://winpg.jp/~saito/JPUG_SGD/ からダウンロード

[] FIREFOX_RUN.sh	15-Jun-2009 10:43	1k
[] libXm.so.4	20-Jan-2009 14:02	2.0M
[] libXm.so.4.md5	15-Jun-2009 10:10	1k

FIREFOX_RUN.sh に実行許可を与えて起動。

4. ログイン

まず、ipa OSSLab へのログイン

(当日、ipa から提供されたアカウントを使用)

続いて、仮想マシン(CentOS5)へのログイン

```
user:jpug
```

5. ログイン環境の説明

postgresql-8.3.7

pgbouncer-1.3,

plproxy-2.0.8

をソースからインストールしてある。

~/bashrc に環境変数。

デモ用のスクリプトを ~jpug/pgsql/下に用意。

```
# cd ~jpug/pgsql
```

まず、postgres のインスタンスを 3つ DB を初期化し(initdb)、
ポートは、それぞれ postgresql.conf に設定。

port: 54320 -- フロントエンドのプロキシサーバとして使用

port: 54321 -- データパーティション 1

port: 54322 -- データパーティション 2

```
# sh 0.initdb-clusters.sh
```

jpug データベースを作成し、サーバを起動する。

```
# cluster-start.sh
```

6. それぞれの Postgres にアクセス(psql)してデータベースの状態確認をしてもらう

```
psql -p 54320 -l
```

```
jpug-# ¥d
```

```
psql -p 54321 -l
```

```
jpug-# ¥d
```

```
psql -p 54322 -l
```

```
jpug-# ¥d
```

7. pgbouncer の設定をってもらう

pgbouncer パラメータの説明と実際

```
# more pgbouncer_test.ini
```

```
port:6666
```

それぞれ、次のデータベース名にしてある

proxy -- フロントエンドのプロキシサーバ

part1 -- パーティション 1 のデータベース

part2 -- パーティション 2 のデータベース

```
## http://pgbouncer.projects.postgresql.org/doc/config.html
```

```
# man 5 pgbouncer
```

デーモンモードで起動し確認

```
# pgbouncer-start.sh
```

```
# psql -p 6666 proxy
```

8. pgbouncer 管理コンソールの説明

接続して操作してみる

```
# psql -p 6666 -U admin pgbouncer
```

```
pgbouncer=# show help;
```

```
pgbouncer=# show cofig;
```

```
pgbouncer=# show clients;
```

```
pgbouncer=# set pool_mode = 'statement';
```

```
pgbouncer=# reload;
```

```
## http://pgbouncer.projects.postgresql.org/doc/usage.html
```

```
# man 1 pgbouncer; /ADMIN CONSOLE
```

9. plpgsql,plproxy の設定

```
# https://developer.skype.com/SkypeGarage/DbProjects/PlProxy
```

PL/Proxy パラメータの説明と実際

```
# psql -s -p 54320 jpug -f $PGSHARE/contrib/plproxy.sql
```

10. pl/proxy の簡単接続確認

プロキシ関数によるリモート実行

```
# psql -s -p 54321 jpug -h 127.0.0.1 -f test1-1.remotetable.sql
```

```
# psql -s -p 54320 jpug -f test1-2.proxyfunction.sql
```

pgbouncer 経由で確認

```
# psql -s -p 6666 proxy -f test1-3.proxyfunc2pgbouncer.sql
```

11. HASH による分散データベースの確認

<http://plproxy.projects.postgresql.org/doc/tutorial.html>

plproxy スキーマを生成し、plproxy 構成関数の登録

```
# psql -s -p 54320 jpug -f test2-1.proxyfunction1.sql
```

ハッシュ値でパーティショニングされるリモート呼び出し

```
# psql -s -p 54320 jpug -f test2-2.proxyfunction1.sql
```

ハッシュ値でパーティショニングされる簡単なインサート

```
# psql -s -p 54321 -h 127.0.0.1 jpug -f test3-1.remotefunction.sql
```

```
# psql -s -p 54322 -h 127.0.0.1 jpug -f test3-1.remotefunction.sql
```

```
# psql -s -p 54320 jpug -f test3-2.proxyfunction.sql
```

プロキシでのデータ確認

```
# psql -p 54320 jpug -f test3-3a.getusermail.sql
```

パーティショニングされたデータベースのデータ確認

```
# psql -p 54321 jpug -f test3-3b.remoteuser.sql
```

```
# psql -p 54322 jpug -f test3-3b.remoteuser.sql
```

pgbouncer 経由で上記の一括確認

```
# psql -p 6666 proxy test3-3.proxyuser2pgbouncer.sql
```

12. レプリケーション方法

RUN On ALL の設定をする

```
$ diff test[34]-1.remotefunction.sql
```

```
5,6c5,6
```

```
< -- psql -s -p 54321 -h 127.0.0.1 juk -f test3-1.remotefunction.sql
```

```
< -- psql -s -p 54322 -h 127.0.0.1 juk -f test3-1.remotefunction.sql
```

```
---
```

```
> -- psql -s -p 54321 -h 127.0.0.1 juk -f test4-1.remotefunction.sql
```

```
> -- psql -s -p 54322 -h 127.0.0.1 juk -f test4-1.remotefunction.sql
```

```
14a15,16
```

```
> DROP FUNCTION ユーザ挿入(ユーザ I text, メール I text);
```

```

> --- 戻り値が変わるので消して作り直し
16c18
< RETURNS integer AS $$
---
> RETURNS TEXT AS $$ --- point!
18c20
< SELECT 1;
---
> SELECT current_user||'['||ユーザ名||','||メール||']' FROM ユーザ
WHERE ユーザ名 =$1; --- point!

$ diff test[34]-2.proxyfunction.sql
3c3,5
< -- psql -s -p 54320 juk -f test3-2.proxyfunction.sql
---
> -- psql -s -p 54320 juk -f test4-2.proxyfunction.sql
> DROP FUNCTION ユーザ挿入(ユーザ I text, メール I text);
> --- 戻り値が変わるので消して作り直し
5c7
< RETURNS integer AS $$
---
> RETURNS SETOF TEXT AS $$ --- point!
7c9
< RUN ON hashtext(ユーザ I);
---
> RUN ON ALL; --- point!
$

```

13. pl/proxy<->pgbouncer<->postgres の設定を試みる

スクリプトを編集して操作してみる

```

$ diff test[25]-1.proxyschema*.sql
2c2,4
< -- psql -s -p 54320 juk -f test2-1.proxyschema.sql
---

```

```
> -- ここでは、構成関数群を定義のうちパーティション取得のスキーマを
> -- PgBouncer 向けに変更する
> -- psql -s -p 6666 proxy -f test5-1.proxyfunction.sql
13,14c15,16
<         RETURN NEXT 'port=54321 host=127.0.0.1 dbname=juk user=juk
password=juk';
<         RETURN NEXT 'port=54322 host=127.0.0.1 dbname=juk user=juk
password=juk';
---
>     RETURN NEXT 'port=6666 host=127.0.0.1 dbname=part1';
>     RETURN NEXT 'port=6666 host=127.0.0.1 dbname=part2';
$

# psql -s -p 6666 proxy -f test4-2.proxyfunction.sql
```

14. その他、

参考までに、PgQ, SkypeTools に関する情報源

<https://developer.skype.com/SkypeGarage/DbProjects/SkyTools>

<http://kaiv.wordpress.com/category/plproxy/>