

PI/proxy

The PostgreSQL Company
Command Prompt, Inc.

Joshua D. Drake

jd@commandprompt.com

Purpose

- Cross database queries
- Horizontal Partitioning
 - Sort of... Federated databases
- Gotchas

Requirements

- PostgreSQL 8.2.5
 - Yes it can run on 8.2.0 but nobody runs less than stable release right?
- PostgreSQL 8.3
 - But it doesn't exist yet.
- pgxs
 - `apt-get install postgresql-server-dev-8.2`
- Pgfoundry
 - <http://pgfoundry.org/projects/plproxy>

Cross Database Queries

Sometimes you just have to:

```
SELECT * FROM
    dblink('dbname=users_2005
           host=192.168.3.254',
           'SELECT userid FROM ...
```

But wouldn't it be great if you could:

```
SELECT userid_return('linuxpoet');
```

You always use functions to control your data flow right?

So what do we do?

(install PL/proxy then...)

- **Create a function on server 192.168.3.5**

```
CREATE OR REPLACE FUNCTION userid_return(text) RETURNS integer AS
$$
    SELECT CASE WHEN id IS NULL THEN 0 ELSE id END
    FROM users
    WHERE username = $1;
$$ LANGUAGE 'SQL';
```

- **Create a function on localhost**

```
CREATE OR REPLACE userid_return(text) RETURNS integer AS
$$
    CONNECT 'dbname=users_2005 host=192.168.3.254 port=6000';
$$
LANGUAGE 'plproxy';
```

Which does what exactly?

The function `userid_return(text)` on the localhost is a thin wrapper to allow execution of a remote function. The two functions must be named identically. The `CONNECT` argument is used to determine which remote server to connect to.

If I was a Dolphin

```
mysql> CREATE DATABASE menagerie;
```

```
mysql> USE menagerie;
```

```
Database changed
```

Sorry Wrong Database

Consider remote validity

```
CREATE TABLE sessions
  (id bigserial PRIMARY KEY,
   userid integer CHECK(is_valid_user(userid) IS TRUE),
   sdate timestamp DEFAULT current_timestamp);
```

Function is_valid_user(integer):

```
CREATE OR REPLACE FUNCTION is_valid_user(integer) RETURNS boolean AS
$$
    CONNECT 'dbname=users_2005 host=192.168.3.254 port=6000';
$$
LANGUAGE 'plproxy';
```


Wait, what just happen?

- **Created function on 192.168.3.254**

```
CREATE OR REPLACE FUNCTION is_valid_user(integer) RETURNS boolean AS
$$
    SELECT CASE WHEN id = $1
                THEN TRUE
                ELSE FALSE
            END
    FROM users
    WHERE id = $1;
$$ LANGUAGE 'SQL';
```

Then.. on the localhost

```
CREATE OR REPLACE FUNCTION is_valid_user(integer) RETURNS boolean AS
$$
    CONNECT 'dbname=users_2005 host=192.168.3.254 port=6000';
$$
LANGUAGE 'plproxy';

CREATE TABLE sessions
(id bigserial PRIMARY KEY,
userid integer CHECK(is_valid_user(userid) IS TRUE),
sdate timestamp DEFAULT current_timestamp);
```

Lastly to prove the perversion

```
192.168.3.254> select * from users;
```

```
id | username |          created
```

```
-----+-----+-----
```

```
1 | linuxpoet | 2005-10-19 17:44:28.819438
```

```
localhost> INSERT INTO sessions (userid,sdate)
```

```
VALUES (2,current_timestamp);
```

```
ERROR: new row for relation "sessions" violates check constraint  
"sessions_userid_check"
```

```
localhost> INSERT INTO sessions (userid,sdate)
```

```
VALUES (1,current_timestamp);
```

```
INSERT 0 1
```

Wait, that means...

Exactly, you can have a check constraint that checks the validity of data on a local relation against the validity of data on a remote relation.

Horizontal Partitioning

- PL/proxy has the ability to not only perform basic data checks on remote partitions but can also use multiple partitions in various ways to achieve greater scalability.
 - ANY – Using the RUN ON 'ANY' method within a PL/proxy function will cause PL/proxy to choose an arbitrary partition to perform the function execution on. (Consider usernames may be on every node).

ALL or nothing

- ALL – Using the RUN ON 'ALL" method will cause PL/proxy, to execute the desired function on "ALL" nodes simultaneously (in parallel).
- The key is that it executes simultaneously. You are not waiting for a single partition to return data before the function can be executed on the next partition.
- Once all results have been returned via the nodes, PL/proxy will then perform a UNION ALL on the data and return it to the client.

EXACT

- The RUN ON 'EXACT' mode causes PL/proxy to run on exactly "1" node. The node is specified within the function body.

Gotchas

- PL/proxy should be considered Alpha software. Although it is being used in production by some companies, it is fragile and documentation is non-existent. When used for specific purposes it is very stable.
- No software should be able to crash the backend.

Thanks

- I think the title says it all.