Using streaming replication of PostgreSQL with pgpool-II

Tatsuo Ishii
President/PostgreSQL committer
SRA OSS, Inc. Japan
About SRA OSS

• Part of SRA group
  • SRA group is an independent software house in Japan
    □ Employees: 1,700
    □ Sales: 3.4 billion yen

• SRA OSS
  • Established in 2005
  • Specialized in OSS support and consultations
  • Has over 1,000 support customers

• Closely working with PostgreSQL community
About me

- PostgreSQL & pgpool-II developer
- Working as a PostgreSQL evangelist (16 talks in 2010)
- Working for SRA OSS, Inc. Japan

Copyright(c)2011 Tatsuo Ishii
Why need replication?

Because it:

- Saves your life when one of your data is lost due to hard disk crash
- Prevents the database system from accidental stopping due to hardware or network troubles. Thus allow you to continue business
- Increases database performance by adding database servers
How to make database replicas?
Transaction log

buffers

database

transaction log

Copyright(c)2011 Tatsuo Ishii
Idea of streaming replication

Read/Write queries

Primary database

buffers

transaction log

transaction log

Standby database

buffers

Read only queries
Architecture of streaming replication

- Read/write query
  - Primary
    - WAL log
    - WAL sender
  - Standby
    - WAL log
- Read query
  - WAL log
  - Archive log
Promoting standby

- Read/write query
- Read query
- Primary
  - WAL log
  - WAL sender
  - WAL receiver
- New Standby
  - Tatsuo Ishii

(Right side of diagram shows the flow of data from the primary to the new standby, including the Wallog and Wal sender/receiver processes.)
Benefits of streaming replication

- PostgreSQL built-in replication available since 9.0
- Easy to setup
- Low overhead
- Everything can be replicated
  - Slony-I does not replicate DDLs, DCLs and large objects
- No application changes necessary as long as accessing primary

Copyright(c)2011 Tatsuo Ishii
Things you might care about

- Replication lag (so called “asynchronous replication”)
- No automatic failover
- No built-in load balancing
- Can't send write queries to standby
  - DDLs, DCLs, DMLs
  - SELECTs including functions which write to databases
  - Some locking commands etc.
Why pgpool-II?

- Some of problems can be solved or at least moderated by using pgpool-II
What is pgpool-II?

- A cluster management tool dedicated for PostgreSQL
- OSS project(pgfoundry, BSD License)
- Rich features
  - Synchronous replication
  - Load balancing, automatic failover, connection pooling etc.
  - Collaborating with other replication tools
    - Streaming replication, Slony-I
- Working as proxy between database clients and PostgreSQL

Copyright(c)2011 Tatsuo Ishii
Basic idea of pgpool-II

Transparent against both PostgreSQL server and client

Copyright(c)2011 Tatsuo Ishii
The architecture of pgpool-II

- pcp process
- pgpool-II parent
- pgpool-II child
- worker process
- Client
- Admin
- PostgreSQL

Health Checking
Replication Delay checking
Query

Copyright(c)2011 Tatsuo Ishii
Pgpool-II helps Streaming replication

Pgpool-II does query dispatching and failover

Read/write queries

Read query

Streaming replication

Write query

Primary

Standby

Copyright(c)2011 Tatsuo Ishii
Pgpool-II helps users who use streaming replication

- Automatic query dispatching
  - Redirect DDLs, DCLs, DMLs and some commands (e.g. LOCK) to primary
  - Allow to specify functions which are writing to DB (they need to be executed on the primary)
  - Allow to load balance SELECTs
- Boost performance by enabling connection cache
- Automatic failover if one of database servers goes down
- Recover down node without stopping pgpool-II (online recovery)
  - Adding standby servers without stopping pgpool-II

Copyright(c)2011 Tatsuo Ishii
Load balance SELECTs in an explicit transaction

```
BEGIN;
SELECT
SELECT
UPDATE
SELECT
```

Transaction starts
Load balance ok
Load balance ok
Write to database
Load balance is not ok anymore because the table might be changed

Copyright(c)2011 Tatsuo Ishii
Monitoring replication delay

- log_standby_delay
  - 'none': no logging delay
  - 'if_over_threshold': log only when delay exceeds delay_threshold
  - 'always': always log delay

- If replication delay is too much, do not send SELECTs to standby

```
2010-06-28  15:51:32 LOG: pid 13223: Replication of node:1 is behind 1228800 bytes from the primary server (node:0)
2010-06-28  15:51:42 LOG: pid 13223: Replication of node:1 is behind 3325952 bytes from the primary server (node:0)
2010-06-28  15:52:02 LOG: pid 13223: Replication of node:1 is behind 974848 bytes from the primary server (node:0)
2010-06-28  15:52:12 LOG: pid 13223: Replication of node:1 is behind 2990080 bytes from the primary server (node:0)
2010-06-28  15:52:22 LOG: pid 13223: Replication of node:1 is behind 901120 bytes from the primary server (node:0)
2010-06-28  15:52:22 LOG: pid 13223: Replication of node:1 is behind 2433024 bytes from the primary server (node:0)
```
What's new in pgpool-II next version (3.1)?

- Allow to use regular expression in black/white_function_list
- Allow to use syslog
- Better handling for more than 1 standby case
- After finishing online recovery, pgpool-II does not restart child process any more (thus client sessions are not disturbed)
- Import PostgreSQL 9.0 SQL parser
Demonstration

DB client

Pgpool-II 3.1

Primary PostgreSQL

Streaming replication

Standby PostgreSQL

Copyright(c)2011 Tatsuo Ishii
URL etc.

- Pgpool-II can be downloaded here:
  - http://pgfoundry.org/projects/pgpool/
- Twitter
  - @pgpool2
- SRA OSS's website