pgpool-II 3.5
How it will look like?

pgpool-II Global Development Group
Tatsuo Ishii
Upcoming pgpool-II 3.5

- Performance enhancement
- Improving watchdog
- Overhauling pcp commands
  - Improved handling of command argument
  - Do not pass password via command line any more (which causes security risks)
  - Concurrent execution of pcp commands
    - Allow to execute a pcp command while running pcp_recovery_node which takes sometime to run
- Obsoleting parallel query mode
  - Very few users and maintenance effort does not worth
- Expect to release in 2015 fall
Major improvements in pgpool-II 3.5

• Improving PCP commands
  • pcp commands are not easy to use (fixed argument positions, for example)
    – Standard command style introduced
  • Single threaded – while executing pcp command, other pcp commands cannot do anything
    – Now PCP commands can execute simultaneously
  • Enhancing security – current PCP commands need to pass password through an argument
    – psql like .pcprc file is introduced
• All of above are already in git repository
Watchdog issues

- Not easy to configure
  - Want to have same configuration files for each pgpool-II node
- Split brain problem
  - More robust way to avoid split brain problem by using voting
- Want to work with multiple networks
  - For example AWS needs replacement for VIP
- We are in the design discussion phase. All developers/users are encouraged to go into the discussion
  - More details can be found here:
    - http://pgpool.net/mediawiki/index.php/watchdog_feature_enhancement
Perforamance enhancement

• Using extended protocol (used in Java prepared statement) in pgpool-II is slow (as slow as half of simple protocol)

• Current implementation of pgpool-II extended protocol is not so great: it requires additional sync of protocol, and this is the source of the performance degradation
  • BTW, extended protocol is not as fast as simple protocol even if pgpool-II is not involved (80% of speed of simple protocol)

• First, need to understand: how extended protocol is handled?
Some details are omitted

Client

query

result

simple protocol

Server

Parse

Bind

Execute

Sync

result

extended protocol

Server

Parse

Sync

Server

Bind

Sync

result

Sync

Execute

Sync

result

Sync

extended protocol with pgpool-II

Sync is needed to handle multiple PostgreSQL more traffic

“Sync” requests send the result from server
In streaming replication, we could omit some of the Sync operations. This is especially useful when using an extended protocol with pgpool-II in 3.5. The diagram illustrates a simplified process:

- **Client**
  - Parse
  - Sync
  - Server
  - Bind
  - Sync
  - result
  - Execute
  - Sync
  - result

- **Server**
  - Parse
  - Bind
  - Execute
  - Sync
  - result

The diagram highlights the elimination of unnecessary Sync steps, improving efficiency and performance. The enhanced protocol further optimizes the process by reducing the number of Sync operations, making it more efficient for streaming replication.
Some details are omitted

Client → Server
query → Parse → Bind → Execute → Sync
result → result → result

Client → Server
Parse → Sync
Bind → Sync
Execute → Sync

Client → Server
Parse → Sync
Bind → Sync
Execute → Sync
result → result

extended protocol with pgpool-II

Sync is needed to handle multiple PostgreSQL more traffic

“Sync” requests sending the result from server

Copyright(c) 2015 SRA OSS, Inc. Japan
Performance comparison

Number of SELECTs per second issued by pgbench

pgpool-II 3.4

pgpool-II 3.5

1.7 times faster than 3.4!
Thank you!