pgpool-II, past, now and future

pgpool-II Global Development Group
Tatsuo Ishii

SRA OSS, INC.

27th June 2003: the birth of pgpool

- Only connection pooling and fail over
- Up to 2 PostgreSQL servers
- Only version 2 protocol is supported (version 3 protocol = PostgreSQL 7.4 was not released yet)
  - 4,719 lines in C

Friday 27th June 22:54:46 JST
[pgsql-jp: 30256] A connection pool server for PostgreSQL: pgpool

Hi,

I have created a connection pool server for PostgreSQL for any programing languages including PHP. Although it is still under development, please try if you like.

ftp://ftp.sra.co.jp/pub/cmd/postgres/pgpool/pgpool-0.1.tar.gz
(note: this URL is still valid!)

# Of course pgpool is under open source license, similar to the BSD license of PostgreSQL.
The reason why I developed pgpool is, we cannot use connection pooling with PHP.
I know PHP has "persistent connection" which caches the connection to DB, but it creates as many as the number of Apache process and it stresses DB. Pgpool will limit the number of connections to DB and improve the performance in this regard.
(Original text was in Japanese)

Palaeoloxodon naumanni
(acient elephant)

April 2004:pgpool 1.0

- Implemented “native replication mode”(at that time PostgreSQL does not have replication)
- Deal with canceling query and large objects
- 5,890 lines in C
- Random minor release policy (e.g. add new functions to minor releases)

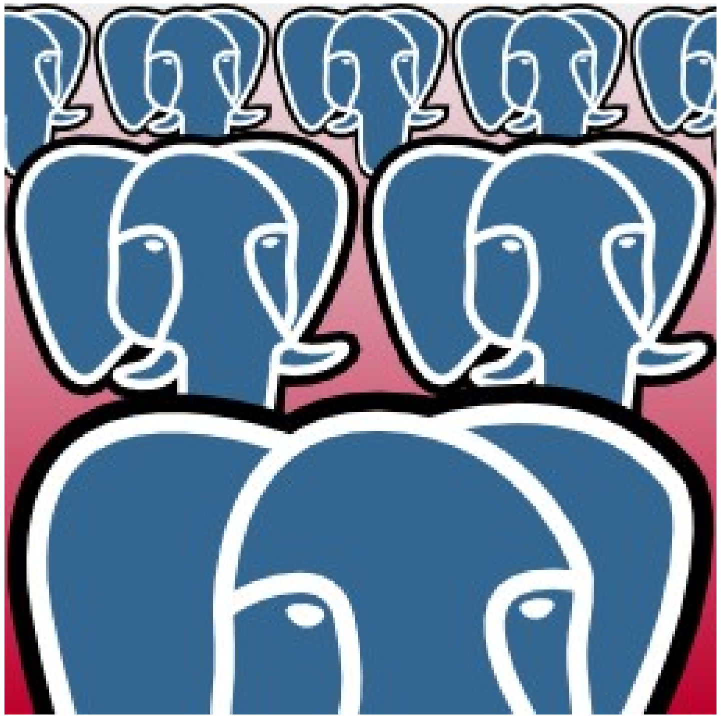
Became modern elephant but toddling

Evolved to pgpool 2.0

- June 2004 merely two months after 1.0
- I seemed to work hard :-)
- Deal with V3 protocol
- 7,750 lines in C
- Feburary 2005 2.5 was released. Health checking and master slave mode added2.5
  - the last release as “pgpool”

September 2006:pgpool-II

- Team project, rather than a personal project of pgpool
  - Grant from IPA
- Lots of new functions
  - Up to 128 PostgreSQL servers possible
  - SQL parser imported from PostgreSQL
  - pgpool-II control command (PCP)
  - A GUI tool (pgpoolAdmin) added
  - Parallel query
  - 73,511 lines in C, 10 times bigger than pgpool



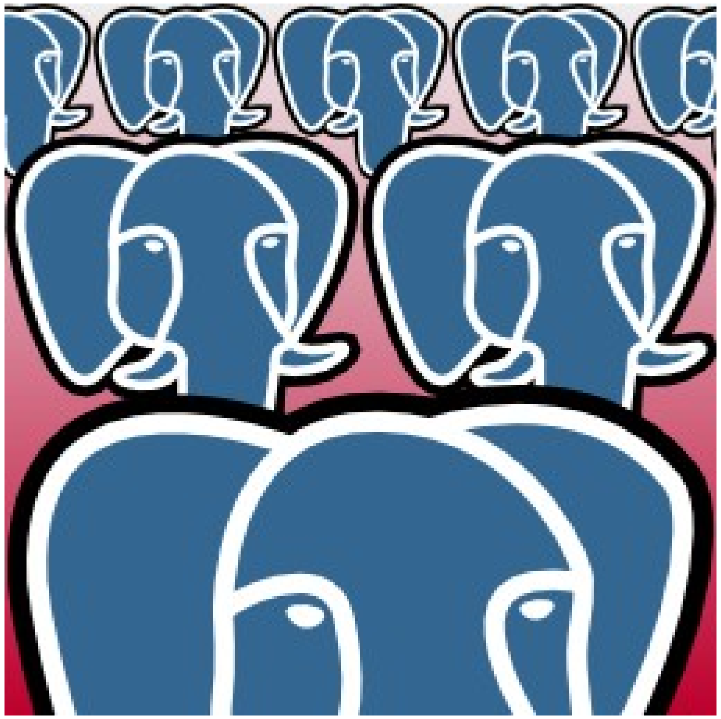
November 2011:Moving to pgpool.net

- Until that time hosted in pgfoundry. But instable...
- Decided to have new site pgpool.net
- Moved from CVS to git
  - Great help from French PostgreSQL community. Thanks!

Moving is not easy...

- In pgpool.net we provide info in English and Japanese (if no time to write in Japanese, English info is provided at least.

Current development team



- Tatsuo Ishii
  - Organizing the team. Writing codes
- Ahsan Hadi
  - User relationship, benchmarking
- Muhammad Usama
  - New committer since pgpool-II 3.4
- Yugo Nagata
  - In charge of watchdog. Release engineering and RPM packaging
- Nozomi Anzai
  - In charge of pgpoolAdmin and the installer. RPM packaging

Current status of pgpool-II

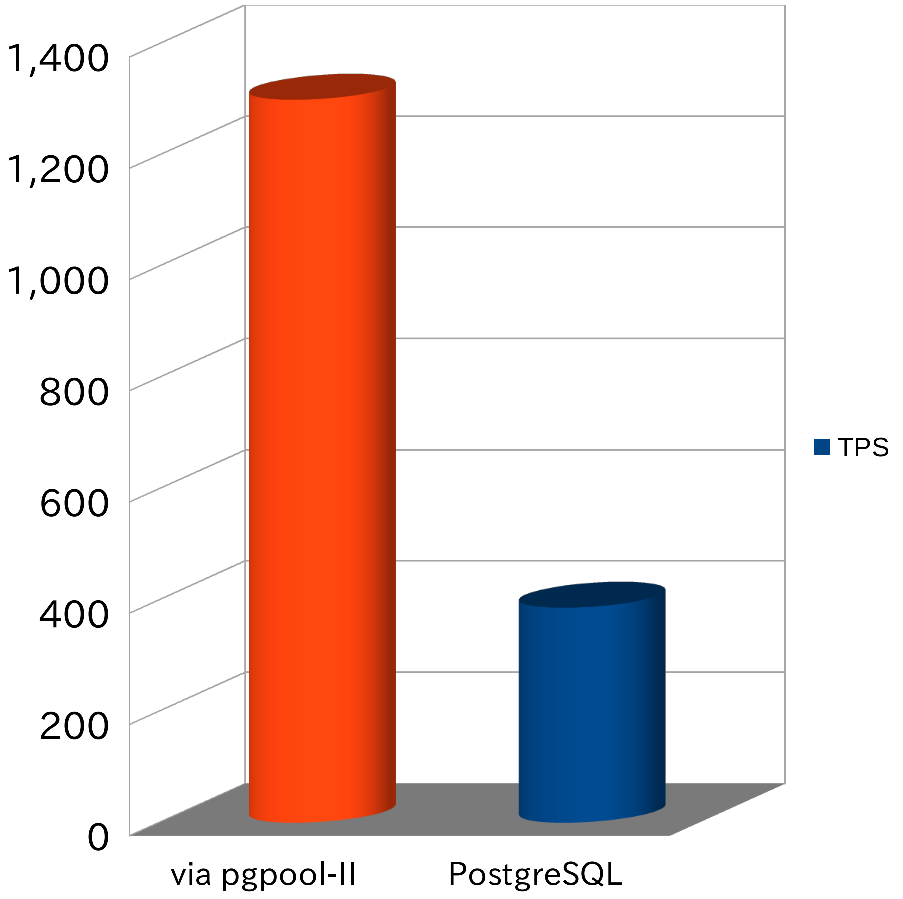
- Total clustering management tool for PostgreSQL
  - Managing streaming replication
  - Query dispatching to primary and standby servers
  - Load balancing
  - Managing fail over
  - In memory query cache
- High availability of pgpool-II itself
  - watchdog

Major functions of pgpool-II

Performance	Connection pooling Load balancing In memory query cache
High availability	automatic fail over fail over script follow master script watchdog
Cluster management	On line recovery
Application and the cluster relationship	Query dispatching

Performance (1)

- Connection pooling
  - Connection overhead to PostgreSQL
  - Overcome the problem by keeping the connections
  - Allow to set the lifetime of connection pooling
  - Not so great for Java because it has its own connection pooling



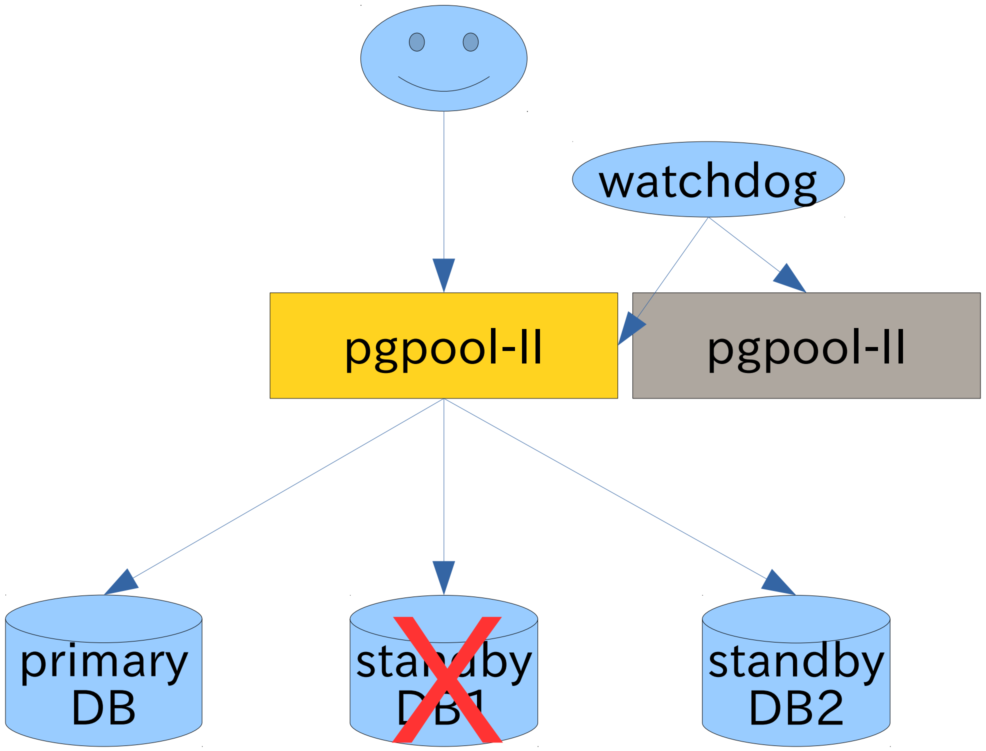
PostgreSQL 9.4+pgpool-II 3.4.1
pgbench -S -C -c 1 -T 30

Performance (2)

- Load balancing
  - Distribute SELECTs to multiple PostgreSQL servers to enhance over all performance
  - More effective with heavier queries
  - Performance boost proportional to the number of PostgreSQL at the best
- In memory query caching
  - The SELECT results are kept so that next SELECT can reuse them. Extremely fast because completely avoid access to the database
  - Do not use on a frequently modified database (the cache hit rate should be 70% or more)

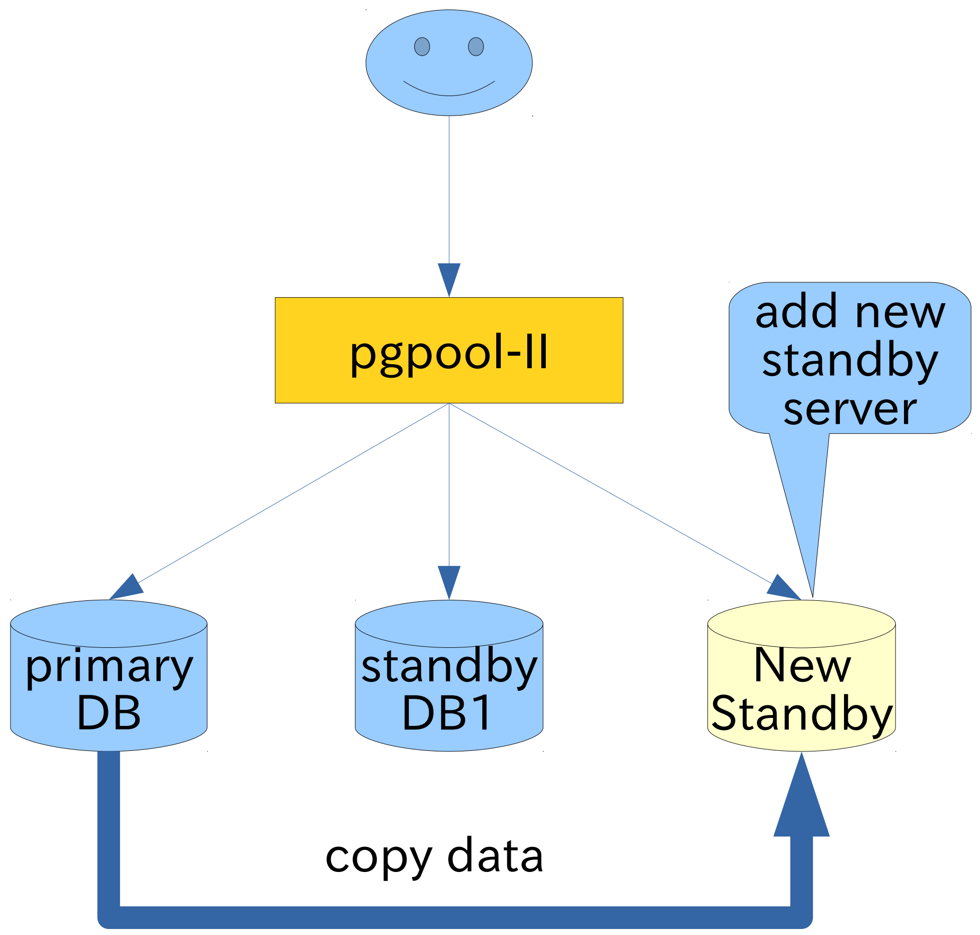
High availability

- Automatic fail over
  - Continue the operation even if one of PostgreSQL servers goes down. Remaining servers inherits the duty
  - Customizable fail over script
- watchdog
  - High availability for pgpool-II itself
  - More fine control than pgpool-HA



Cluster management

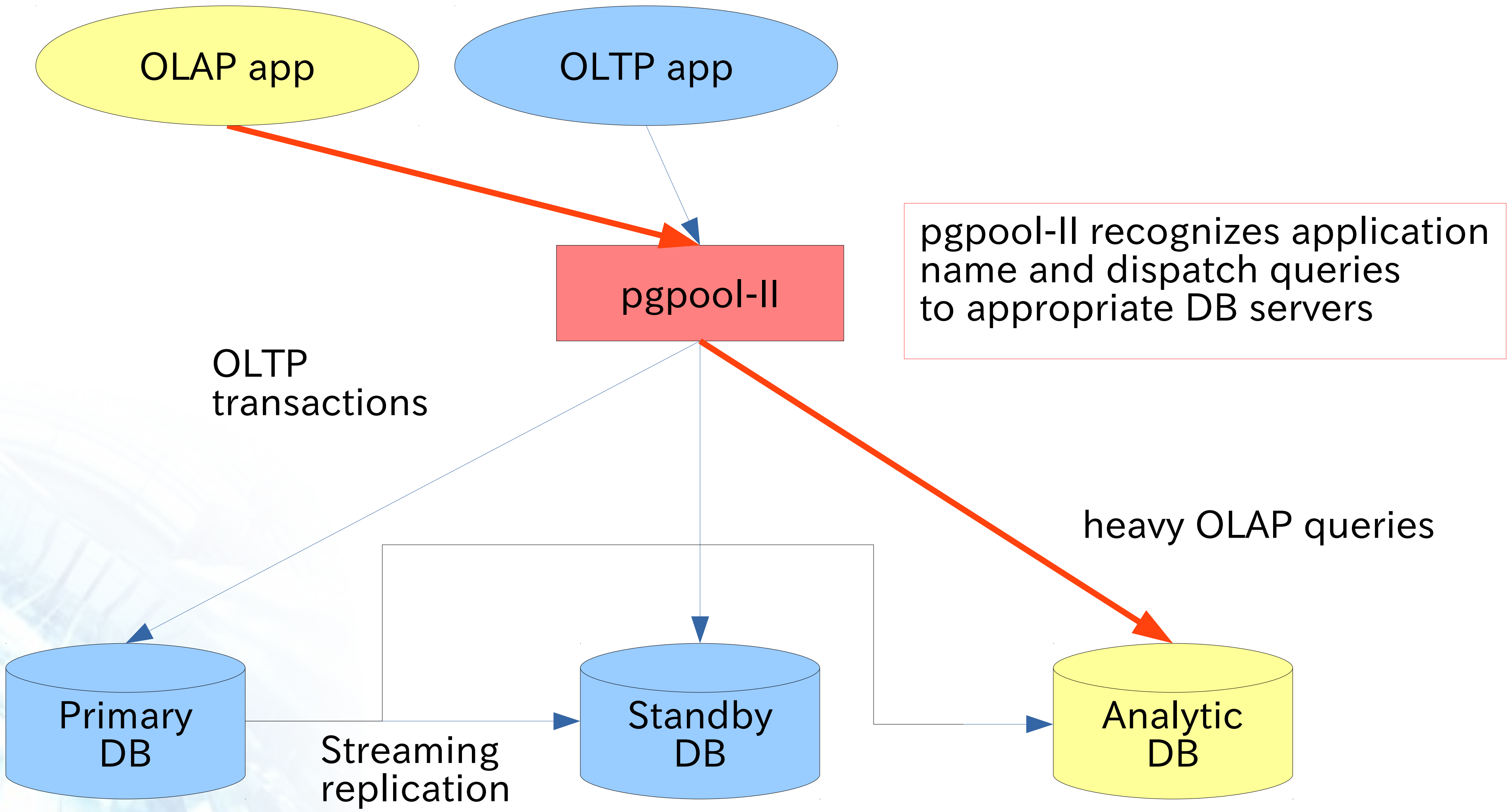
- Allow to recover a standby server without stopping the database clustering system (on line recovery)
  - Predefined script can be used for necessary operations including data coying from primary server and starting the standby server
  - Same technique can be used to add new standby server
  - New standby server is available for new sessions



Relationship database clustering system and applications

- Certain queries cannot be sent to standby servers. Some of them are not obvious
  - DML
  - Add or delete databases
  - Add or delete users
  - VACUU/REINDEX
  - One of LOCK statements
  - Temporary tables
  - Serializable isolation level
  - And more....
- Annoying for applications
- pgpool-II transparently handles query dispatching to primary and standby servers

Executing heavy OLAP queries without disturbing OLTP transactions



pgpool-II recognizes application name and dispatch queries to appropriate DB servers

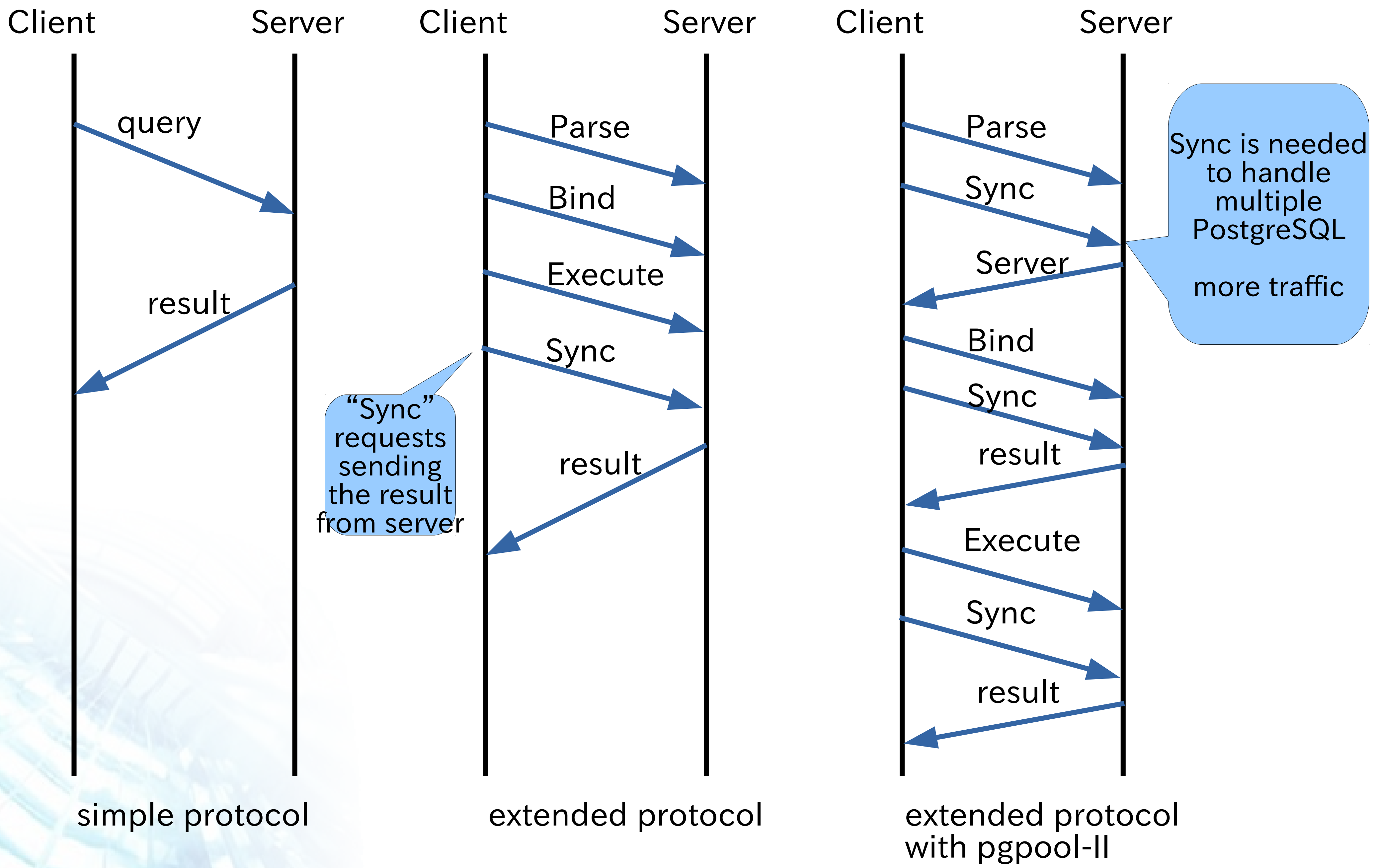
So what is next?

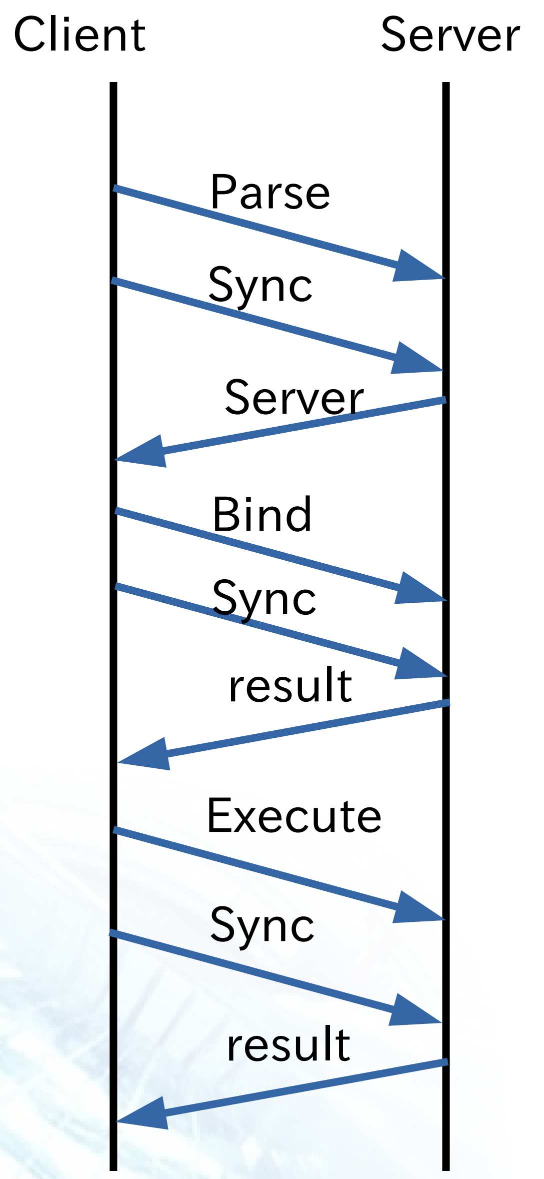
- Upcoming pgpool-II 3.5
  - Performance enhancement
  - Improving watchdog
  - Overhauling pcp commands
    - Improved handling of command argument
    - Do not pass password via command line any more (which causes security risks)
    - Concurrent execution of pcp commands
      - Allow to execute a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

Perforamance enhancement

- Using extended protocol (used in Java prepared statement) in pgpool-II is slow (as slow as half of simple protocol)
- Implementation of pgpool-II extended protocol is not so great
  - BTW, extended protocol is not as fast as simple protocol even if pgpool-II is not involved (80% of speed of simple protocol)
- So how extended protocol is handled?

Some details are omitted



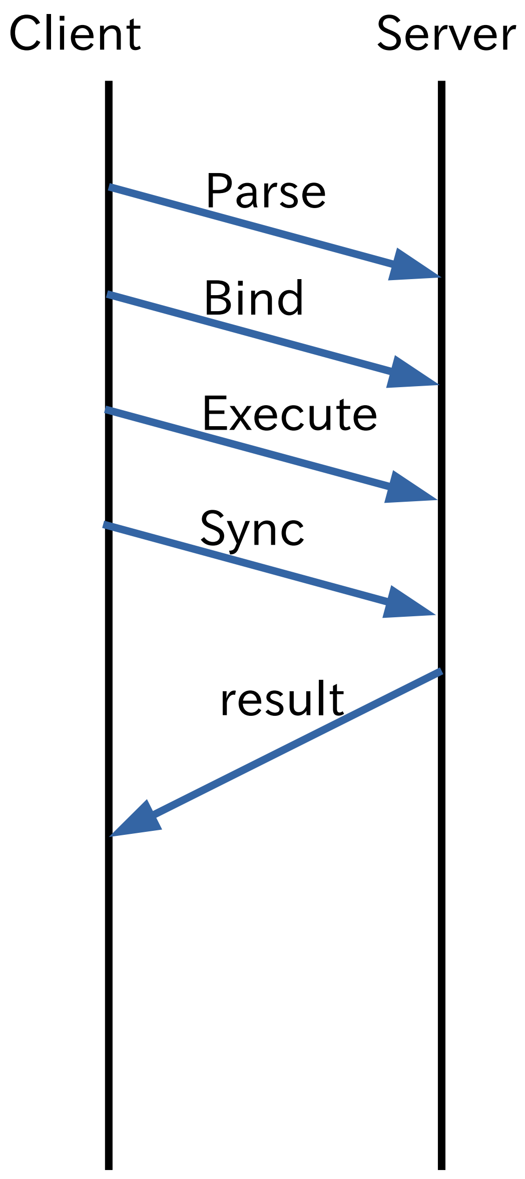


extended protocol
with pgpool-II

Too many Sync

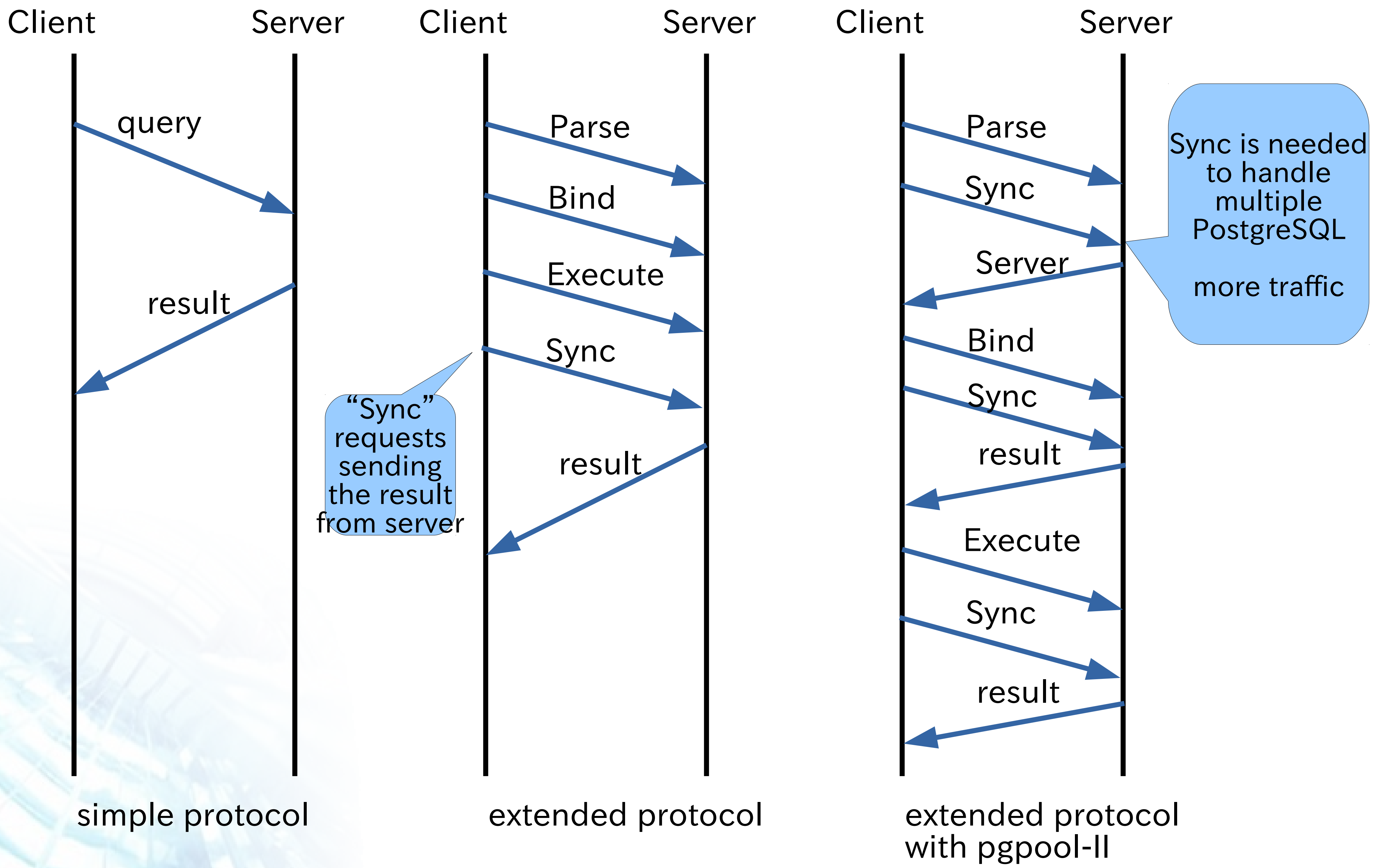
In streaming replication we could omit some of Sync

enhanced!

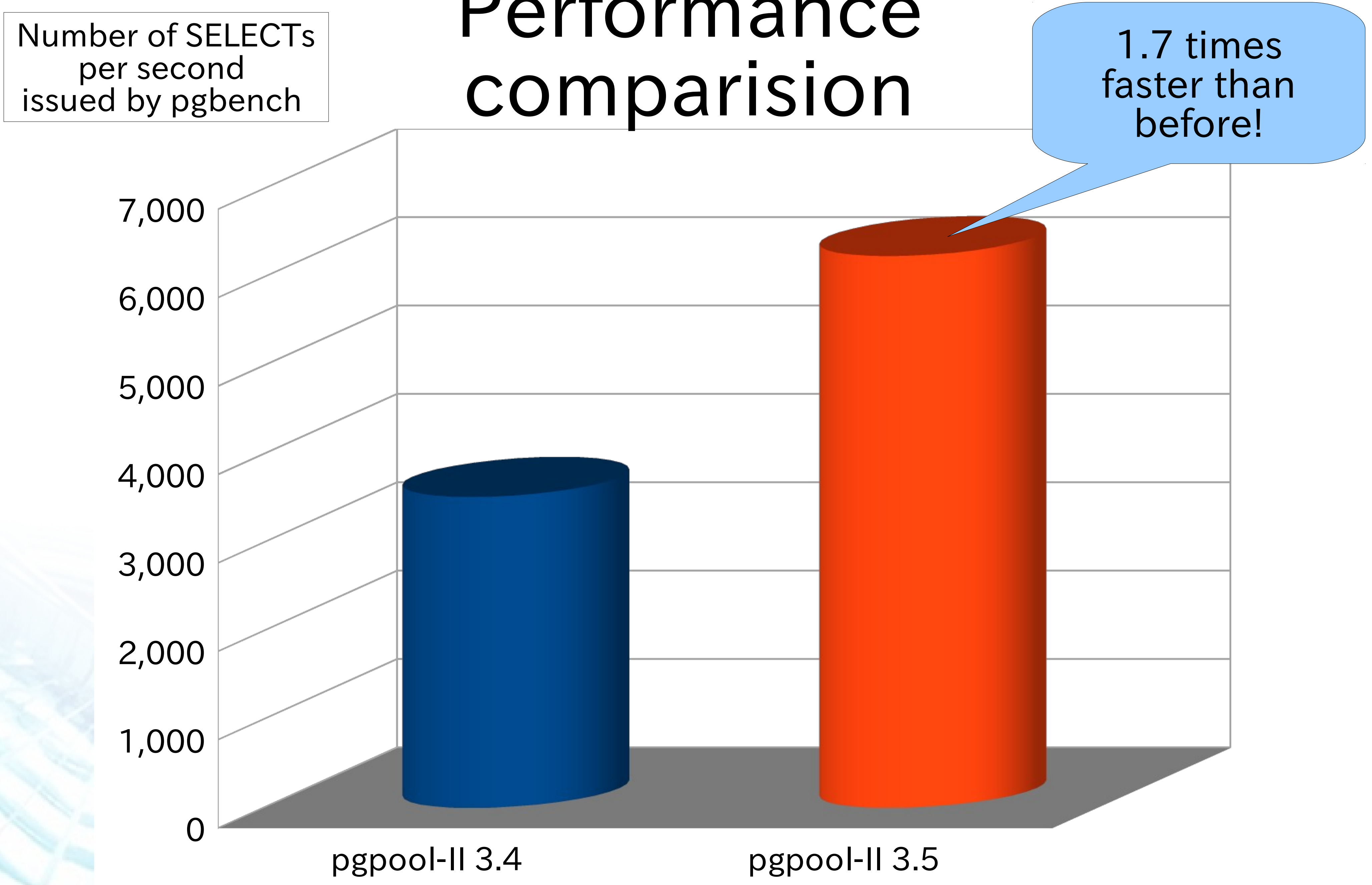


extended protocol
with pgpool-II in 3.5

Some details are omitted



Performance comparision



Future plans

- Follow PostgreSQL evolution
  - Logical replications
  - BDR (Bi Directional Replication)
  - parallel queries
  - Need to think what PostgreSQL aims for
- Overhauling documents
  - Structure is not terribly good
    - More like PostgreSQL?
  - Hard to maintain because we write in plain HTML
    - Looking for good authoring tools
    - Sphinx?

SRA OSS, INC.

Thanks!

- Call for developers!
- Benefits joining gpool-II development
  - Easier than PostgreSQL
    - 1/10 in size
    - You can learn SQL parser, exception manager which are imported from PostgreSQL
  - Experiencing network programming and multi process programming