pgpool-II: Demonstration

Yoshiyuki Asaba
Tatsuo Ishii

pgpool Global Development Group
SRA OSS, Inc. Japan
Agenda

● History
● Overview of pgpool-II
● Demonstration
pgpool: the history

- V0.1: Started as a personal project (2003/6/27)
- V3.0: pgpool Global Development Group (2006/2)
- pgpool-II V1.0 (2006/9)
- pgpool-II V2.0 (2007/11)
- pgpool-II V2.1beta (2008/5)
Why pgpool?

• We need synchronous replication
• We need simple replication solution
• We need transparent replication solution
• We need high performance clustering solution
pgpool features

✔ Synchronous replication
  ✔ No need to change application semantics
✔ Easy to use
  ✔ Easy to configure
  ✔ Can replicate DDL
✔ Transparent replication
  ✔ Can be used with any programming languages
pgpool features (continued)

✔ Performance boost
  ✔ connection pooling
  ✔ load balancing
  ✔ parallel queries

✔ Enhance reliability
  ✔ automatic failover

✔ Easy to administrate
  ✔ GUI tool
  ✔ On line recovery
On line recovery

- “Fail back” a node without stopping pgpool
- Two-stage recovery steps
  - First stage
    - Recover data roughly
      - Can accept write queries
      - Some data might not be consistent between “master” and recovered node
  - Second stage
    - Recover data precisely
      - Wait until all clients disconnected
      - Precisely sync data
On line recovery using PITR

• First stage
  – Start base backup on master
  – Send base backup to slave
  – Take some time but queries can be accepted including write ones

• Second stage
  – Wait until all clients disconnected
  – Send archive logs generated on the first stage
  – Start PITR recovery on slave
pgpool-II+warm standby

- Warm standby system continuously makes a copy of the on line node
- Standby system takes over when the on line node goes down
- Need some components for warm standby system to be used in the real world
  - Detecting on line node failure
  - Redirecting DB connections to the new node
- pgpool-II can provide both
pgpool-II + warm standby

- pgpool-II operates in “raw” mode
- no replication
- no load balancing