

# pgpool-II demonstration

pgpool Global Development Group

Yoshiyuki Asaba

y-asaba@sraoss.co.jp

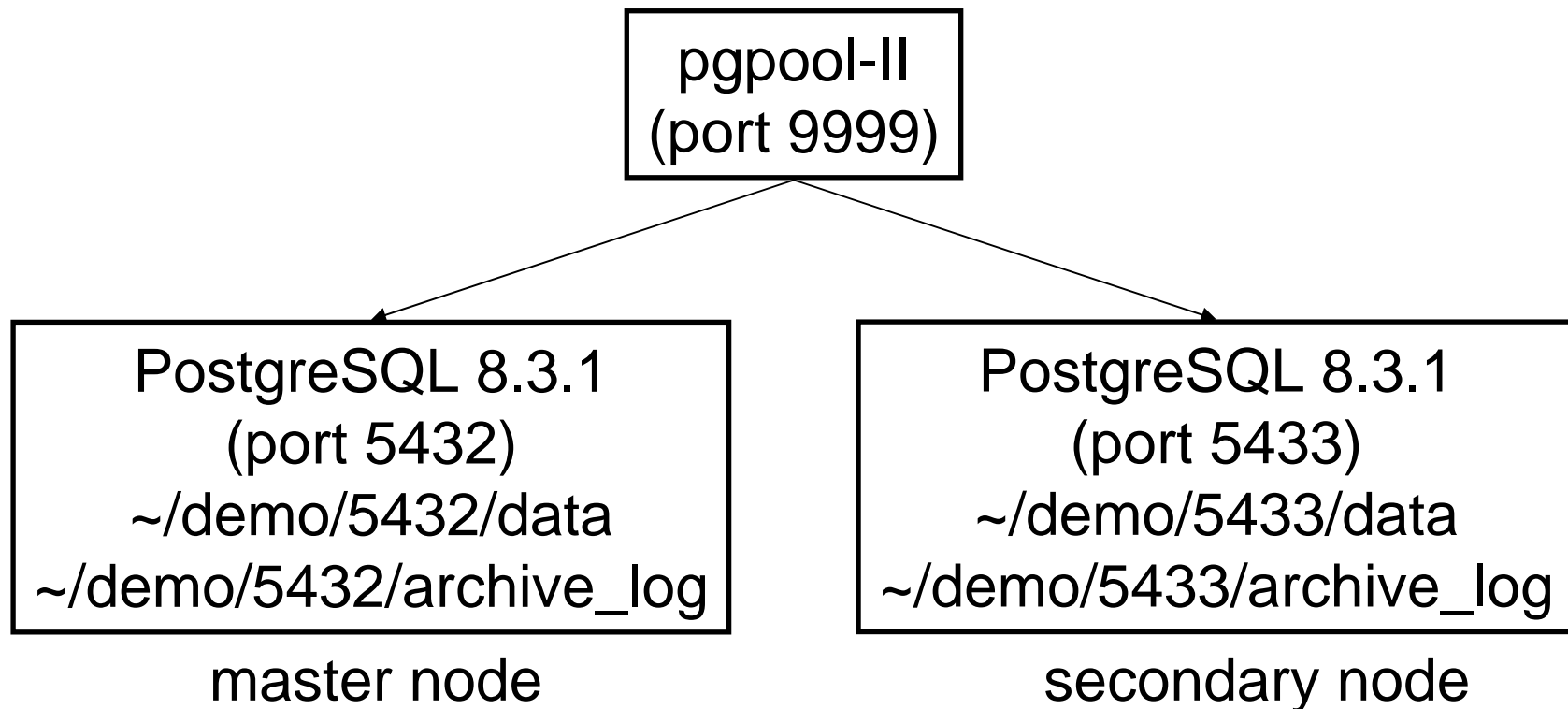
# agenda

1. Replication & online recovery demo
2. Warm standby with pgpool

# Online recovery

# Environment

- pgpool and PostgreSQL servers are on localhost.



# Pgpool setting(1)

- pgpool.conf

```
# Master node info
backend_hostname0 = 'localhost'
backend_port0 = 5432
backend_data_directory0 = '/home/y-asaba/demo/5432/data'

# Secondary node info
backend_hostname1 = 'localhost'
backend_port1 = 5433
backend_data_directory1 = '/home/y-asaba/demo/5433/data'
```

# Pgpool setting(2)

- pgpool.conf

```
# Enable replication  
replication_mode = true
```

```
# Enable load balance  
load_balance_mode = true
```

```
# Online recovery user  
recovery_user = 'y-asaba'
```

```
# The first stage script  
recovery_1st_stage_command = 'base-backup.sh'
```

```
# The second stage script  
recovery_2nd_stage_command = 'pgpool-recovery'
```

# 1st stage script

- Create base backup

```
#!/bin/sh

PORT=5432
DATA=$HOME/demo/$PORT

psql -c "select pg_start_backup('pgpool-recovery')" postgres

# Generate recovery.conf.
echo "restore_command = 'cp $DATA/archive_log/%f %p'" >$DATA/data/recovery.conf

# Copy base backup.
rsync -az --delete -e ssh --exclude postmaster.pid --exclude postmaster.opts ¥
      --exclude pg_log --exclude postgresql.conf $1/ $3/

psql -c 'select pg_stop_backup()' postgres
```

# 2nd stage script

- Archive a current XLOG.

```
#!/bin/sh  
  
# Archive a current xlog.  
psql -c 'select pg_switch_xlog()' postgres
```



# pgpool\_remote\_start

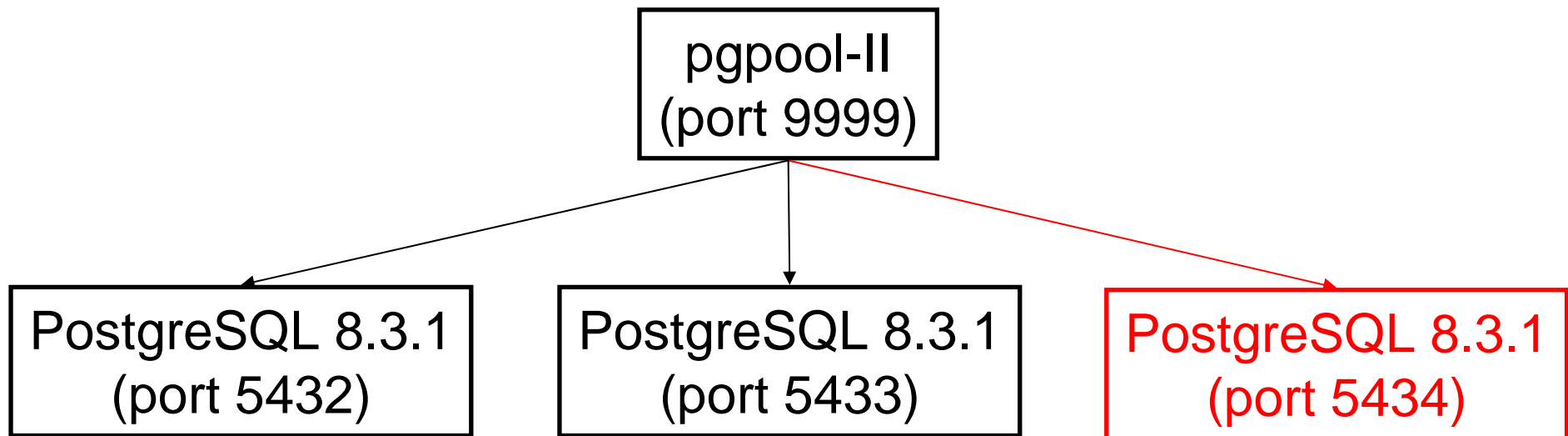
- Script for starting up PostgreSQL
  - The first argument is a recovery target name.
  - The second argument is a recovery target's PGDATA

```
#!/bin/sh
DEST=$1
DESTDIR=$2
PGCTL=/home/y-asaba/bin/pg_ctl

$PGCTL -w -D $DESTDIR start
```

# Scale up

- Add a new backend node without stopping pgpool.

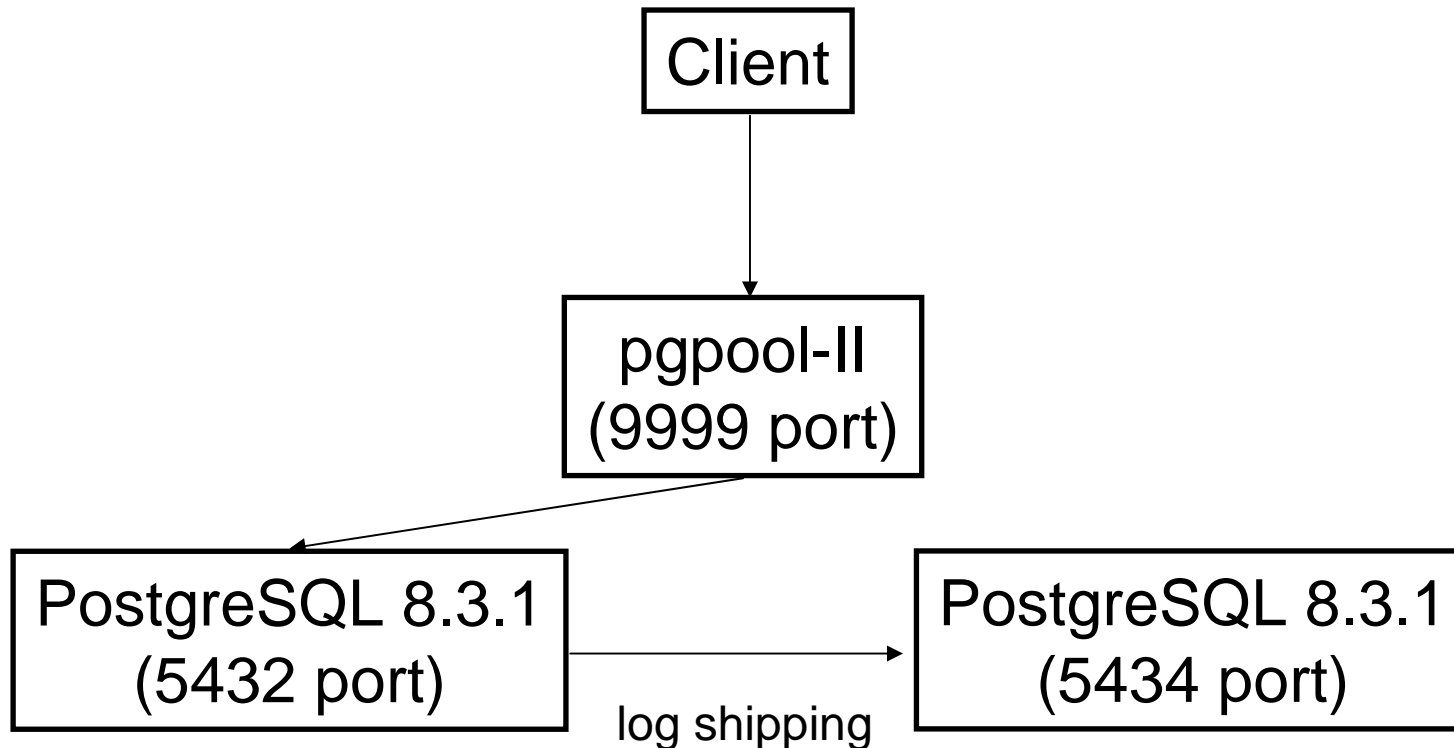


pgpool-II & warm standby

# pgpool-II & warm standby

- Clients does not change connection host.
- Pgpool-II does health checking and failover process.

– I've fixed a failover bug today... Please download CVS HEAD.



# pgpool-II&warm standby demo

1. Set up warm standby
2. Set up pgpool
3. Start up pgpool
4. `createdb -p 9999 warm`
5. `pgbench -i -p 9999 warm`
  - `SELECT COUNT(*) FROM accounts;`
6. Shutdown PostgreSQL
7. Check failover

# Set up warm standby

- recovery.conf

```
restore_command = 'pg_standby -s -t /tmp/pgsql.trigger  
~/demo/5432/archive_log %f %p %r'
```

- postgresql.conf

```
port = 5434  
archive_mode = on  
archive_command = 'cp -f %p ~y-asaba/demo/5432/archive_log/%f'  
archive_timeout = 60
```

# Set up pgpool

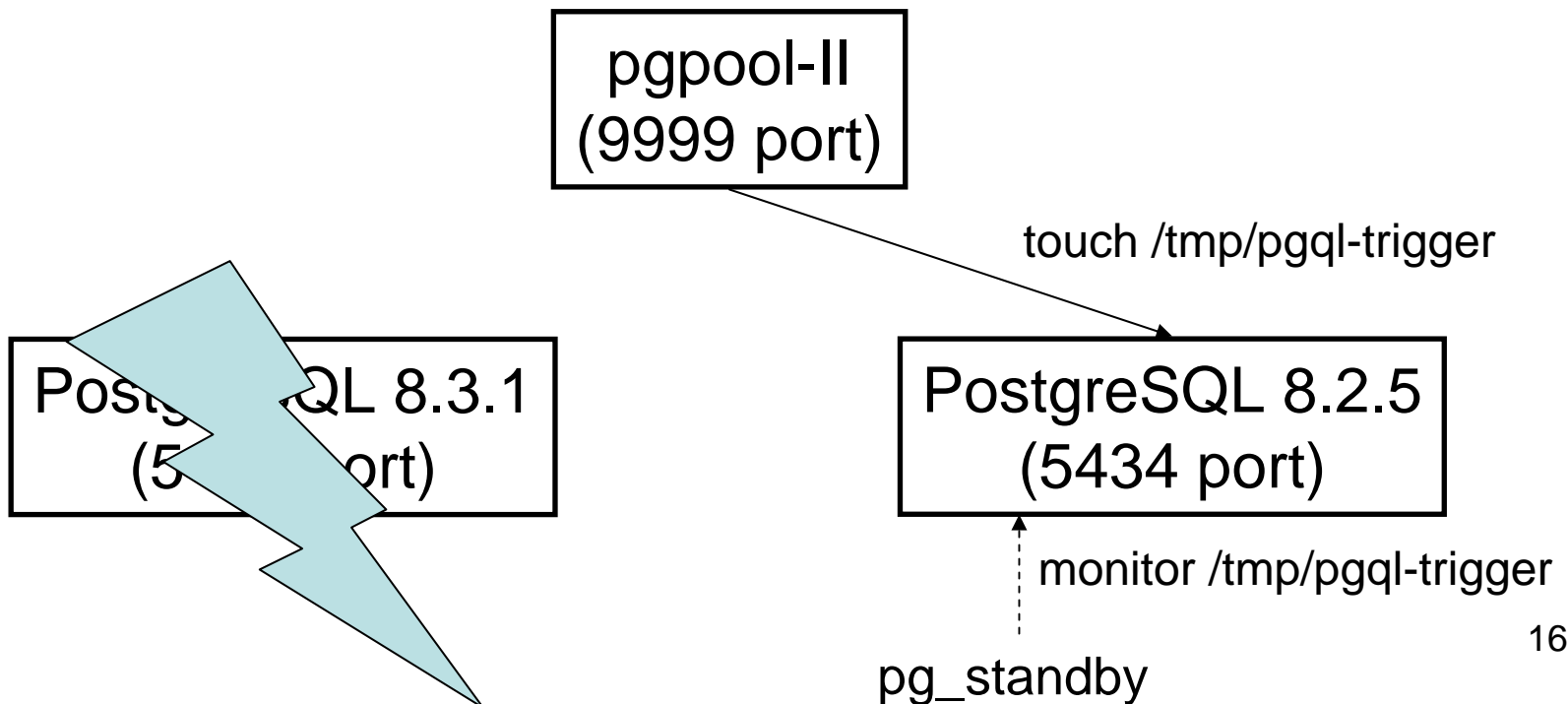
- pgpool.conf
  - Need to generate ssh key without pass phrase

```
backend_hostname0 = 'localhost'
backend_port0 = 5432
backend_hostname1 = 'localhost'
backend_port1 = 5434
failover_command = 'ssh localhost touch /tmp/pgsql.trigger'

replication_mode = false
master_slave_mode = false
parallel_mode = false
```

# Failover

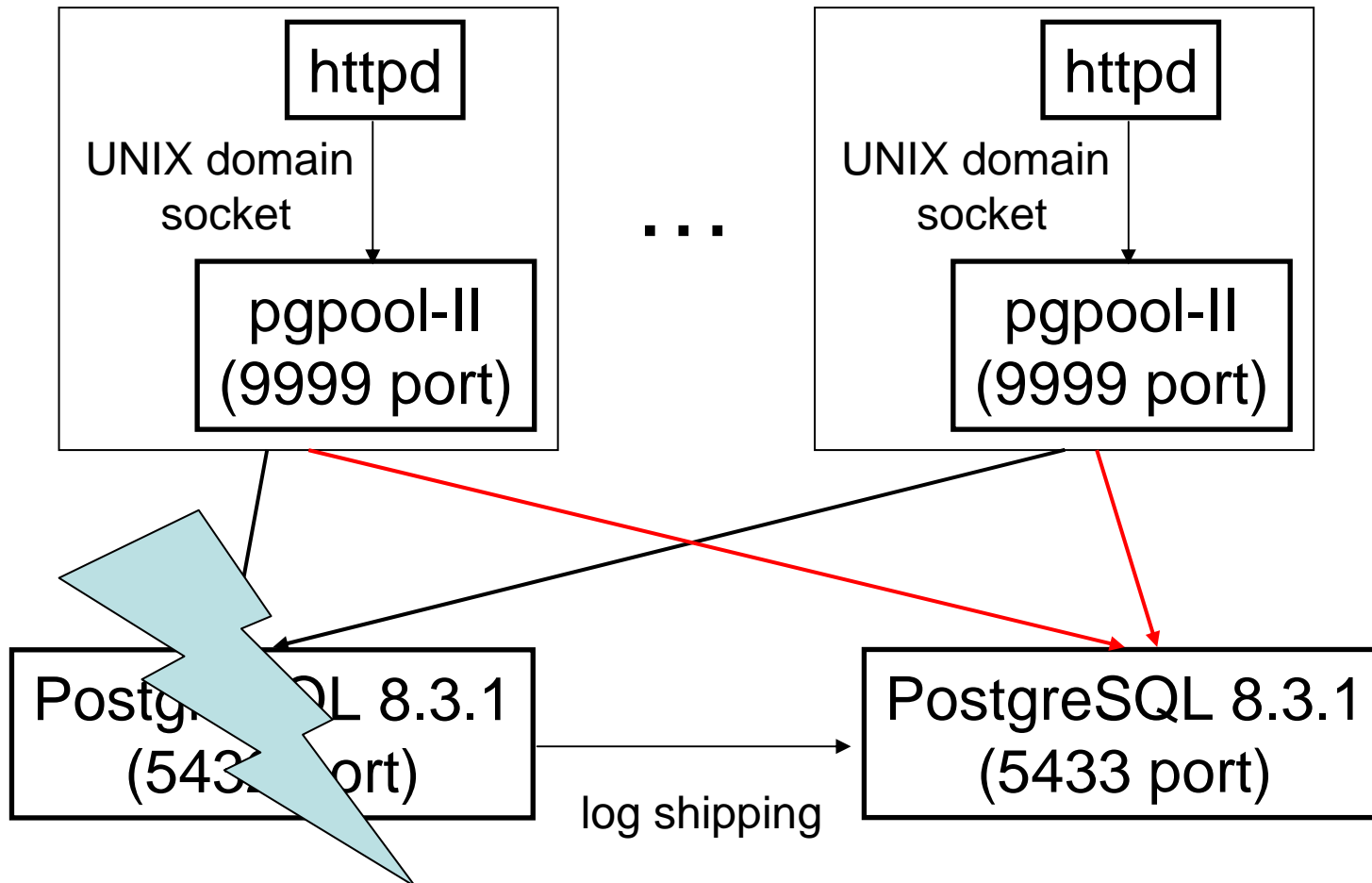
- Pgpool can execute a script at failover.
  - pgpool.conf
  - failover\_command = 'ssh localhost touch /tmp/trigger'





# Single Point of Failure?

- No.



Thanks.