

# **My First Patch – adding new option to pgbench**

Lightning Talk @PGConf.dev 2025  
Rintaro Ikeda

# About Me

I'm probably the least experienced with PostgreSQL in this room.

- I started using PostgreSQL in 2023.
- I began reading PostgreSQL source code in 2024.

Last week, I submitted my first patch modifying source code!

## Suggestion to add --continue-client-on-abort option to pgbench

From: ikedarintarof <ikedarintarof(at)oss(dot)nttdata(dot)com>  
To: pgsql-hackers(at)postgresql(dot)org  
Subject: Suggestion to add --continue-client-on-abort option to pgbench  
Date: 2025-05-10 13:45:31

Message-ID: [44334231a4d214fac382a69cceb7d9fc@oss.nttdata.com](#)

Views: [Raw Message](#) | [Whole Thread](#) | [Download mbox](#) | [Resend email](#)

Thread: 2025-05-10 13:45:31 from ikedarintarof <ikedarintarof(at)oss(dot)nttdata(dot)com>  

Lists: [pgsql-hackers](#)

# Problem of pgbench I aim to solve

Currently, a client stops sending requests when its transaction is aborted due to reasons other than serialization failures or deadlocks.

```
$ cat custom_script.sql
CREATE TABLE IF NOT EXISTS test (col1 serial, col2 int unique);
INSERT INTO test (col2) VALUES (random(0, 50000));

$ bin/pgbench -d postgres -f ../custom_script.sql -T 10
[...]
pgbench: error: client 0 script 0 aborted in command 1 query 0: ERROR: duplicate key
value violates unique constraint "test_col2_key"
DETAIL: Key (col2)=(28373) already exists.
transaction type: ../custom_script.sql
```

pgbench stops in a second

This issue may result in a lower server load than expected, especially when you specified the number of client, duration time.

# Patch I submitted

I proposed “—continue-client-on-error” option.

- when transaction fails, client rollback the transaction and start new one
- the number of clients remains unchanged

```
% bin/pgbench -d postgres -f ../insert_to_unique_column.sql -T 10 --failures-detailed --continue-client-on-error
```

```
transaction type: ../custom_script_insert.sql
```

```
[...]
```

```
number of transactions actually processed: 33552
```

```
number of failed transactions: 21901 (39.495%)
```

```
number of serialization failures: 0 (0.000%)
```

```
number of deadlock failures: 0 (0.000%)
```

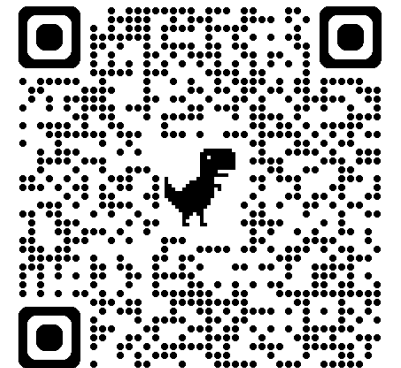
```
number of other failures: 21901 (39.495%)
```

```
latency average = 0.180 ms (including failures)
```

```
initial connection time = 2.857 ms
```

```
tps = 3356.092385 (without initial connection time)
```

pgbench keeps applying load  
for the specified time



If you are interested, give me feedback!