

Resource Usage

Database Administration Lab Guide 1

2025/2026

Consider again the simplified invoice processing system with the following database schema (Figure 1):

Client: Id, Name, Address, Data.

Product: Id, Description, Price, Ref, Data.

Invoice: Id, ProductId, ClientId, Price, Data.

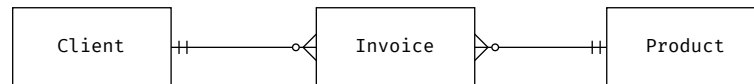


Figure 1: Benchmark's schema

The application provides the following operations:

Sell: Add invoice record.

Account: List names of products sold to some client.

Top10: List currently 10 most sold products

Using the benchmark, explore the relation between performance and configuration of memory management parameters:

- `shared_buffers`
- `work_mem`

Steps

1. Check the documentation of `shared_buffers`¹ and `work_mem`².
2. Adjust concurrency level (number of clients, e.g., 1, 2, 4, 8, 16) and database size (e.g., `-s` set to 10, 15, 20) for the default configuration.
3. Adjust configuration parameters, **down** and **up**, and repeat the benchmark.

Questions

1. What is the impact of work memory and shared buffer allocation in the maximum achievable throughput?
2. Can the baseline performance be improved?

Learning Outcomes Relate resource usage with performance. Use memory configuration parameters to influence system performance.

¹<https://www.postgresql.org/docs/18/runtime-config-resource.html#GUC-SHARED-BUFFERS>

²<https://www.postgresql.org/docs/18/runtime-config-resource.html#GUC-WORK-MEM>

PostgreSQL HowTo

With Docker

1. Create the container:

```
$ docker run --name postgres -e POSTGRES_PASSWORD=postgres \
  -p 5432:5432 -d postgres:18
```

2. Access the psql client:

```
$ docker exec -it postgres psql -U postgres
```

3. Create a new database testdb:

```
# in psql
psql> create database testdb;
```

```
# or using createdb
docker exec -it postgres createdb -U postgres testdb
```

4. Connect to the new database:

```
psql> \c testdb
```

5. Get the list of relations:

```
psql> \d
```

6. To restart the server:

```
$ docker restart postgres
```

7. To stop the container:

```
$ docker stop postgres
```

8. Sending commands to psql without logging in:

```
$ docker exec postgres psql -U postgres -c "select 1"
```

Update system parameters

```
-- Check the parameter's current value (psql)
SHOW <param>;
```

```
-- Change the value for the current session (reverts back to the previous
-- value when we open a new session; not valid for shared_buffers)
SET <param> = <x>;
```

```
-- Change the value permanently
```

```
-- option 1: using ALTER SYSTEM
-- (for shared_buffers, we need to restart the server;
-- for most other parameters the pg_reload_conf() is enough)
ALTER SYSTEM SET <param> = <x>;
SELECT pg_reload_conf();
```

```
-- option 2: editing the config file
SHOW config_file;
-> /path/to/postgresql.conf
-- edit the file, e.g., with vim
-- restart the server
```