PostgreSQL Tutorials

← Back

How to tune PostgreSQL for memory

Tuning

Tushar Ahuja, Sr. QA Manager · Apr 6, 2020

This article looks at parameters that can be used to help manage memory in PostgreSQL. Recommended settings for each parameter are also provided.

1. Shared_buffers (integer)
2. Work_mem (integer)
3. Maintenance_work_mem (integer)
4. Effective_cache_size (integer)

In this post, we are going to look at some of the important `GUC parameters` recommended for memory management in PostgreSQL, which is helpful for improving the performance of your database server. All of these parameters reside under the `postgresql.conf` file (inside $PDATA directory), which manages the configurations of the database server.

shared_buffers (integer)

The `shared_buffers` parameter determines how much memory is dedicated to the server for caching data.

The default value for this parameter, which is set in `postgresql.conf`, is:

```
#shared_buffers = 128MB
```

The value should be set to 15% to 25% of the machine's total RAM. For example: if your machine's RAM size is 32 GB, then the recommended value for `shared_buffers` is 8 GB. Please note that the database server needs to be restarted after this change.

work_mem (integer)

The `work_mem` parameter basically provides the amount of memory to be used by internal sort operations and hash tables before writing to temporary disk files. Sort operations are used for order by, distinct, and merge join operations. Hash tables are used in hash joins and hash based aggregation.

The default value for this parameter, which is set in `postgresql.conf`, is:

```
#work_mem = 4MB
```

Setting the correct value of `work_mem` parameter can result in less disk-swapping, and therefore far quicker queries.

We can use the formula below to calculate the optimal `work_mem` value for the database server:

```
Total RAM * 0.25 / max_connections
```

The `max_connections` parameter is one of the `GUC parameters` to specify the maximum number of concurrent connections to the database server. By default it is set to 100 connections.

We can also directly assign `work_mem` to a role:

```
postgres=# alter user test set work_mem='4GB';

ALTER ROLE
```

maintenance_work_mem (integer)

The `maintenance_work_mem` parameter basically provides the maximum amount of memory to be used by maintenance operations like `vacuum`, `create index`, and `alter table add foreign key` operations.

The default value for this parameter, which is set in `postgresql.conf`, is:

```
#maintenance_work_mem = 64MB
```

It's recommended to set this value higher than `work_mem`; this can improve performance for vacuuming. In general it should be:

```
Total RAM * 0.05
```

effective_cache_size (integer)

The `effective_cache_size` parameter estimates how much memory is available for disk caching by the operating system and within the database itself. The PostgreSQL query planner decides whether it's fixed in RAM or not. Index scans are most likely to be used against higher values; otherwise, sequential scans will be used if the value is low. Recommendations are to set `Effective cache size` at 50% of the machine's total RAM.

For more details and other parameters, please refer to the PostgreSQL documentation: https://www.postgresql.org/docs/12/runtime-config-resource.html.

Hope it helps!

Tushar Ahuja, Sr. QA Manager

Why EDB?

Use Cases

Oracle Migration

Hybrid Cloud

High Availability

Solutions for

IT Professionals

Database Architects

Developers

Database Admins

Products

Databases

EDB Postgres Advanced Server

PostgreSQL

Tools

Postgres Enterprise Manager

Backup and Recovery Tool

Failover Manager

Open Source Projects

Migration Portal

Migration Toolkit

Replication Server

Services

Services Overview

Training

Getting Started

PostgreSQL Optimization

Enterprise Strategy

Custom Services

Support

Customer Support Portal

Support Overview

PostgreSQL Technical Support

Remote DBA Service

Cloud DBA Service

Technical Account Management

Resources

Docs

Blog

Webinars

PostgreSQL Tutorials

Training

Partners

White Papers

Customer Stories

Product Compatibility

Plans

Company

Events

Press Releases

Media Coverage

Customers

Follow Us

Twitter

LinkedIn

Facebook

YouTube