Configuring and Tuning PostgreSQL and EDB Postgres Advanced Server

GUIDE FOR LINUX USERS

Last updated: July 30, 2020

EnterpriseDB | www.enterprisedb.com

EDB

These recommendations for tuning for EDB Postgres Advanced Server (EPAS) and PostgreSQL represent a starting point. Benchmarks and other measurements are required for proper tuning. This document is not intended to be an exhaustive list of configuration settings or recommendations, only the most important parameter settings that are not already the default values.

For optimal tuning we recommend engaging with EDB's Professional Services team or a qualified EDB partner.

Operating system-level recommendations:

Mount Points

We recommend having separate mount points with dedicated IOPs for the EPAS/PG Cluster. For better performance, we recommend using SSDs

1. `/pgdata:` Mount point for data directory of EPAS/PG
PGDATA directory: `/pgdata/data`

2. `/pgwaldata/:` mount point for WAL (Write Ahead Log).

WAL directory for EPAS/PG: `/pgwaldata/wal`

Depending on the usage of indexes and the amount of indexes needed for user-defined tables, as per workload, we recommend having a separate mount point for indexes.

File System

We recommend using the XFS file system for PG/EPAS data directory, WAL, and any other mount points hosting PG/EPAS data.

Read-ahead

Increasing disk read ahead improves I/O throughput by reducing the number of requests to disk.
This is typically set at 128 kB, but it's generally recommended that this be set to 4096 kB on the disk hosting the database or tablespaces.

This information can be obtained with the following command:

```
cat /sys/block/<device name>/ queue/ read_ahead_kb
```

In order to benchmark the effect of increasing read ahead, it can be changed instantly by running:

```
echo 4096 > /sys/block/sda/queue/ read_ahead_kb
```

For this to be set permanently, add it to /etc/ rc.local so that it takes effect upon reboot.

Note: For systems that do not support /etc/rc.local, use the equivalent startup script that is run after the destination runlevel has been reached.

I/O scheduler

We recommend using the `deadline` I/O scheduler for the CentOS/RHEL 7 and `mq-deadline` for CentOS/RHEL 8.directly to building PostgreSQL.

Other Operating System Parameters

Below are EDB's suggestions for tuning OS for EPAS/PostgreSQL:

```
mkdir /etc/tuned/edb
echo "
[main]
summary=Tuned profiles for
EnterpriseDB Postgres Advanced
Server
[cpu]
governor=performance energy_perf_
bias=performance min_perf_pct=100
[disk]
readahead=>4096
[sysctl]
vm.overcommit_memory=2
vm.swappiness=1
vm.dirty_ratio=30
vm.dirty_background_ratio=10
[vm]
transparent_hugepages=never
" > /etc/tuned/edb/tuned.conf
systemctl enable --now tuned
tuned-adm profile edb
```

Initializing the EPAS/ PG Cluster

initdb Command

```
/usr/edb/as12/bin initdb
      --auth=peer \
    --auth-host=scram-sha-256 \
      --pgdata=/pgdata/data\
      --waldir=/pgwaldata/wal \
     --encoding=UTF-8 \
      --data-checksums \
       --no-locale
```

Update the service file as shown below:

```
sed -i 's#/var/lib/edb/as12/data#/
pgdata/data#g' /usr/lib/systemd/
system/edb-as-12.service

systemctl enable edb-as-12
systemctl start edb-as-12
```

Configuration & Authentication

max_connections

The optimal number for max_connections is roughly 4 times the number of CPU cores. This formula often gives a very small number which doesn't leave much room for error. The recommended number is the `GREATEST(4 x CPU cores, 100)`. Beyond this number, a connection pooler such as pgbouncer should be used.

password_encryption

It is recommended that people should use `scram-sha-256` for this parameter.

Resource Usage

shared_buffers

This parameter has the most variance of all. Some workloads work best with very small values (such as 1GB or 2GB) even with very large database volumes. Other workloads require large values. A reasonable starting point is the `LEAST(RAM/2, 10GB)`.

work_mem

The recommended starting point for work_mem is `((Total RAM - shared_buffers)/ (16 x CPU cores))`.

maintenance_work_mem

This determines the maximum amount of memory used for maintenance operations like VACUUM, CREATE INDEX, ALTER TABLE ADD FOREIGN KEY, and data-loading operations. These may increase the I/O on the database servers while performing such activities, so allocating more memory to them may lead to these operations finishing more quickly. The calculated value of `15% x (Total RAM - shared_buffers) / autovacuum_max_workers` upto 1GB is a good start.

effective_io_concurrency

This parameter is used for read-ahead during certain operations and should be set to the number of disks used to store the data. However, improvements have been observed by using a multiple of that number. For SSD based disk, it is recommended to set this value to 200.

Write-Ahead Log

wal_compression

When this parameter is on, the PostgreSQL server compresses a full-page image written to WAL when full_page_writes is on or during a base backup. Set this parameter to '`on`'

wal_log_hints

This parameter is required in order to use pg_ rewind Set it to '`on`'.

wal_buffers

This controls the amount of memory space available for back ends to place WAL data prior to sync. WAL segments are 16MB each by default, so buffering a segment is very inexpensive memory-wise. And larger buffer sizes have been observed to have a potentially very positive effect on performance in testing. Set this parameter to `64MB`.

checkpoint_timeout

Longer timeouts reduce overall WAL volume but make crash recovery take longer. The recommended value is a minimum of `15 minutes`.

checkpoint_completion_target

This determines the amount of time in which PostgreSQL aims to complete a checkpoint. This means a checkpoint need not result in an I/O spike and instead aims to spread the writes over a certain period of time. The recommended value is `0.9`.

max_wal_size

Checkpoints should always be triggered by a timeout for better predictability. This parameter should be used to protect against running out of disk space. The recommended value is half to two-thirds of the available disk space where the WAL is located.

archive_mode

Because changing this requires a restart, it should be set to '`on`'.

archive_command

A valid archive_command is required if archive_mode is on. Until archiving is ready to be configured, a default of '`: to be configured`' on POSIX systems or '`cd .`' on Windows is recommended.

Query Tuning

random_page_cost

If using SSD disks, the recommended value is `1.1`.

effective_cache_size

This should be the sum of `shared_buffers` and the Linux buffer cache (as seen in the buff/ cache column of the `free` command).

cpu_tuple_cost

Specifies the relative cost of processing each row during a query. It is currently set to 0.01, but this is likely to be lower than optimal and should be increased to `0.03` for a more realistic costing.

Reporting and Logging

logging_collector

This parameter should be `on` if log_destination includes `stderr` or `csvlog`.

log_directory

If the logging_collector is on, this should be set to someplace outside of the data directory.
This way, the logs are not part of base backups.

log_checkpoints

This should be set to `on`.

log_line_prefix

The prefix should at least contain the time, the proc id, the line number, the user and database, and the application name.

Suggested value: '`%m [%p-%l] %u@%d app=%a`'
Note: Don't forget the space at the end

log_lock_waits

Set to `on`. This parameter is essential in diagnosing slow queries.

log_statement

Set to '`ddl`'. In addition to leaving a basic audit trail, this will help determine at what time a catastrophic human error occurred, such as dropping the wrong table.

log_temp_files

Set to `0`. This will log all temporary files created, suggesting that work_mem is incorrectly tuned.

timed_statistics (EPAS)

Controls the collection of timing data for the Dynamic Runtime Instrumentation Tools Architecture (DRITA) feature. When set to on, timing data is collected. Set this parameter to on.

Autovacuum

log_autovacuum_min_duration Monitoring autovacuum activity will help in tuning it.
Suggested value: `0`.

autovacuum_max_workers

This is the number of workers that autovacuum has. Default value 3 and requires a DB restart to be updated. Please note that each table can have only one worker working on it. So increasing workers only helps in parallel and more frequent vacuuming across tables. The default value is low, therefore it is recommended to increase this value to `5`.

Client Connection Defaults

idle_in_transaction_session_timeout

Sessions that remain idle in a transaction can hold locks and prevent vacuum.
Suggested value: `10 minutes`.

lc_messages

Log analyzers only understand untranslated messages. Set this to '`C`'.

shared_preload_libraries

Adding pg_stat_statements is low overhead and high value. This is recommended but optional.
