Understanding and Fixing Autovacuum

Robert Haas - PGCONF.IN - February 23, 2023

How It Works

How is `VACUUM` different from autovacuum?

- `VACUUM` is an SQL command that performs certain maintenance operations.
- Autovacuum is a set of background processes that automatically run the `VACUUM` and `ANALYZE` commands.

What Does VACUUM Do?

- `VACUUM` recycles space consumed by deleted rows and the old version versions of update rows. If this is not done, your tables and indexes will “bloat,” becoming unreasonably large.

- `VACUUM` allows for the reuse of transaction IDs, or XIDs, and of multi-transaction IDs, or MXIDs. If this is not done, you will eventually run out of usable XIDs or MXIDs and the system will stop accepting write transactions.

- `VACUUM` updates the visibility map for tables, making index-only scans and future `VACUUM` operations more efficient. If this is not done, there are possible performance consequences, often minor.

- In short, if you don’t run `VACUUM` regularly, or if it fails to accomplish the tasks that it is intended to do, *bad things happen happen to your database*.

How Does Autovacuum Work?

- There is an *autovacuum launcher* process which normally runs all the time.

- It keeps track of what databases exist and regularly starts *autovacuum worker* processes in each database.

- An autovacuum worker looks at each table in the database to which it has been instructed to connect, and it decides whether to perform `VACUUM` or `ANALYZE` on that table.

How It Fails

How Can Autovacuum Go Wrong?

- As users, we hope that autovacuum will keep our database working properly: low bloat, plenty of available XIDs, etc.
- However, autovacuum is just a system for running `VACUUM` and `ANALYZE` commands automatically according to a set of configurable parameters.
- If the configuration isn't appropriate for the system, autovacuum may work correctly but fail to achieve the intended goals.
- Any problem with `VACUUM` or `ANALYZE` will also affect autovacuum.

What Problems Can `VACUUM` or `ANALYZE` Have?

- Typically, these problems happen with `VACUUM`, not `ANALYZE`.
- Could fail.
  - Then, autovacuum will keep retrying, maybe forever.
- Could run forever and never complete.
  - Then, the table won't get vacuumed, and maybe others won't, either.
- Could run for a really long time.
  - Same problems.
- Could finish successfully in a reasonable amount of time but without really getting any work done.
  - Then, autovacuum will keep retrying, maybe forever.

What Problems Can Occur In Autovacuum Itself?

- Autovacuum itself is very reliable provided that it is properly configured.

- People sometimes try to turn it off e.g. `autovacuum=off`or `autovacuum_naptime=’1 day’`

- Assuming that you don’t do that, the most common problem is that autovacuum either:
    - thinks tables need to be vacuumed when really they don’t, or
    - doesn’t think tables need to be vacuumed when really they do.

Problem Summary

- *Slow or Stuck*. `VACUUM` runs for a very long time on a table, maybe forever.

- *Spinning*. Same `VACUUM` is getting retried over and over again, either because either because it's failing with an error, or because it's succeeding but nothing useful is happening.

- *Skipped*. Autovacuum doesn't think that `VACUUM` is required, but actually it is.

- *Starvation*. Autovacuum is too busy and so some tables are not getting checked.

VACUUM Slow or Stuck

`VACUUM` is Running Forever on My Table - Why?

- Your table is just really big and nothing is actually wrong.
  - Just be patient!

- `VACUUM` has been configured to run too slowly.
  - The most common case.

- `VACUUM` is waiting for some other PostgreSQL process to complete some operation.
  - Kill off other processes that are accessing the same table so that `VACUUM` can proceed.

- Your disk is too slow, possibly because it’s failing.
  - Fix your disk or buy a new one.

- Your indexes are corrupted, causing an infinite loop.
  - `DROP` or `REINDEX` them.

How Can I Tell What's Happening?

- Check whether the process is consuming I/O or CPU time using `ps` or similar tools.

- Check `pg_stat_activity.wait_event_type`and `pg_stat_activity.wait_event`.

- Check `pg_stat_progress_vacuum`.

- For example, if you see that the wait event type is often `VacuumDelay`, then probably VACUUM is configured to run too slowly, but if it's often `IO`, then your disk isn't keeping up.

How Do I Make Vacuum Run Faster?

- Raise `vacuum_cost_limit`(or reduce `autovacuum_vacuum_cost_delay`).
- The required value of `vacuum_cost_limit` increases as your database gets larger and busier.
- In modern server versions, the default settings allow autovacuum to run at a rate of >100GB/hour, but the defaults were 10x lower before v12.
- If you have very large tables, you might need to raise the values further.
- The value is shared across all workers so if you have raised `autovacuum_max_workers`, each worker will run slower unless you also raise `autovacuum_cost_limit`.
- The setting doesn't take effect for autovacuum workers that are already running. In emergency situations, use `gdb` to set `VacuumCostActive = false`

VACUUM Spinning

Autovacuum is Processing a Table Over and Over - Why?

- Your table is just experiencing a lot of write activity and so it needs `VACUUM` frequently.
  - No problem!

- `VACUUM` is failing with an `ERROR` and autovacuum keeps retrying it.
  - Check the error message in the log file and proceed accordingly.

- Autovacuum keeps thinking the table needs vacuuming, but when `VACUUM` actually processes it, nothing gets any better.
  - Autovacuum is aware that there are old tuples within the table, but doesn’t realize that they can’t yet be cleaned up.

Repeated Autovacuum, But Not Helping

- Old rows cannot yet be removed because they are still required by either:
  - a long-running transaction (check `pg_stat_activity`) that needs to be killed, or
  - an unused replication slot (check `pg_replication_slots`) that needs to be dropped or reconnected, or
  - an old prepared transaction (check `pg_prepared_xacts`) that needs to be committed or aborted.

VACUUM Starved or Skipped

Autovacuum Is Not Running VACUUM on My Table - Why?

- Starved: All autovacuum worker processes have been busy doing other things, and therefore your table is not being checked.

- Skipped: Autovacuum decides the table doesn’t need vacuuming so it doesn’t do anything.

- To tell the difference, check `pg_stat_activity`to see how many autovacuum workers are running and compare to `autovacuum_max_workers`. Also check how long they have been running.

- If all workers are running and have been running for a long time, most likely the table is getting starved. Otherwise, it’s probably getting skipped.

If a Table is VACUUM-Starved

- There is no problem with this particular table. Rather, the problem is that overall autovacuum is not able to keep up.

- Solutions are the similar to when vacuum is slow - mainly, raise `vacuum_cost_limit`. In theory, you could need faster hardware.

- It’s also sometimes useful to increase `autovacuum_max_workers`but that rarely works by itself. If you do this, you may need to raise `vacuum_cost_limit`even higher since the budget is shared across all workers.
    - For instance, if you double the number of workers you might want to also double the cost limit (or more, if you want the cost limit for each worker to be higher).

If VACUUM Is Skipping a Table

- If the table is large, you may need to reduce `autovacuum_vacuum_scale_factor`. The default value of 0.2 is sometimes too big for large tables. You can reduce the system-wide value in postgresql.conf or via `ALTER SYSTEM`, or use `ALTER TABLE … SET (autovacuum_vacuum_scale_factor = ….)`

- Check whether any custom settings have been applied to the table by using psql's `\d` command. For example, `ALTER TABLE … SET (autovacuum_enabled = false)` can cause this problem. Likewise if you set `autovacuum=false` in the configuration file.

- You might have a problem with the statistics system. Try a manual `VACUUM` of the affected tables or of the entire database and see if autovacuum begins working normally again after that. If not, contact an expert.

- In general, no matter the reason, autovacuum will eventually run anyway, but by then your tables may be so badly bloated that the system is unusable.

EDB™

POWER TO POSTGRES

We are hiring. Be part of our team!

Scan Now to Apply!

careers@enterprisedb.com

@EDBPostgres /company/EDBPostgres www.enterprisedb.com

THANK YOU

Any questions?