VACUUM's page truncation considerations

Primary Contact muiwamo (user) *How do I change this value?*

Last modified 1 month ago by muiwamo.

vacuum_truncate parameter and the motivations

`vacuum_truncate` parameter will be introduced in PostgreSQL 18.

This parameter enables and disables the VACUUM's (including automatic vacuum) page truncation phase throughout the database cluster.

What we can learn from vacuum_truncate parameter

Truncating a page acquires an `ACCESS EXCLUSIVE` lock on the respective tables.

Normally, the impact is likely to be limited, because when focusing only on the read/write database clusters (including the Aurora writer instance), the page truncation phase will be given up when the VACUUM can't acquire an ACCESS EXCLUSIVE lock, or if there is a lock that conflicts with an ACCESS EXCLUSIVE lock in other transactions.

However, such evasive behavior do not work in hot standby (including Aurora reader instances).

Page truncation can cause problems in cases where the applications does not have much access to the table in the primary cluster (Aurora writer instance) but there is frequent access to the tables in hot standby (Aurora reader instances).

Since an ACCESS EXCLUSIVE lock on the primary will be replicated to the hot standby, that lock makes queries on hot standbys wait, and application failures may occur.

There are also use cases where the workload is distributed to the hot standby, so for example, when recovering from a situation where VACUUM is blocked, the possibility that the page will be truncated also increases, so it would be necessary to explain the risk of such an impact on the hot standby side.

[1] https://www.postgresql.org/docs/18/runtime-config-vacuum.html#GUC-VACUUM-TRUNCATE

The motivation was discussed in commit messages and mailing list[2,3].

It's important point when talking about Aurora PostgreSQL multi-AZ setups.

[2] https://github.com/postgres/postgres/commit/0164a0f9ee12e0eff9e4c661358a272ecd65c2d4

```
Add vacuum_truncate configuration parameter.
This new parameter works just like the storage parameter of the
same name: if set to true (which is the default), autovacuum and
VACUUM attempt to truncate any empty pages at the end of the table.
It is primarily intended to help users avoid locking issues on hot
standbys.  The setting can be overridden with the storage parameter
or VACUUM's TRUNCATE option.
```

[3] https://www.postgresql.org/message-id/flat/Z2DE4lDX4tHqNGZt%40dev.null

vacuum_truncate パラメータとそのモチベーション

vacuum_truncate パラメータが PostgreSQL 18 で です。

このパラメータはデータベースクラスタ で VACUUM バキューム む）の Truncation phase、ページの り めを とするパラメータです。

ページの り めは、テーブルに する ACCESS EXCLUSIVE ロックを します。

プライマリデータベース（Aurora のライターインスタンス む）のみに すると、ACCESS EXCLUSIVE ロックが できない や、 にユーザのワークロードで ACCESS EXCLUSIVE ロックと するロックがあると をやめ は です。

しかし、ホットスタンバイ（Aurora のリーダーインスタンスを む）のワークロードは に されません。

これにより、VACUUM がページの り めを うテーブルに し、ライターインスタンスではあまりアクセスが い、ホットスタンバイでは なアクセスがあるといったケースで になることがあります。

プライマリで した ACCESS EXCLUSIVE ロックはホットスタンバイ でも ACCESS EXCLUSIVE ロックが されることから、ホットスタンバイ のクエリがロック により をすることになるので、アプリケーションの につながることがあります。

ホットスタンバイへワークロードを させているユースケースもあるので、 えば VACUUM がブロックされている からの においては、ページの り めが われる も まるため にホットスタンバイ へのこういった のリスクを しておくことは と えます。

Tags: PostgreSQL Aurora PostgreSQL