

PGroonga運用技法

~PGroongaのWALを
放置していませんか？~

堀本泰弘

クリアコード

PostgreSQL Conference Japan 2020
2020-11-13





自己紹介



堀本 泰弘

全文検索エンジン

gr^{cloud}nga pgr^{cloud}nga
mr^{cloud}nga rr^{cloud}nga

の開発・サポート



PGroongaの紹介

PGroongaとは？



PGroongaの紹介

PostgreSQL経由 で 高速な全文検索



PGroongaの特徴

1. **簡単**に使える
2. **速い**



PGroongaの使い方

```
CREATE TABLE memos (  
  id integer,  
  content text  
);
```

```
CREATE INDEX pgroonga_content_index  
ON memos  
USING pgroonga (content);
```



PGroongaの使い方

準備完了



PGroongaの使い方

データを投入して
検索してみよう！



PGroongaの使い方

```
INSERT INTO memos
VALUES (1,
        'PostgreSQLはリレーショナル・データベース管理システムです。');

INSERT INTO memos
VALUES (2,
        'Groongaは日本語対応の高速な全文検索エンジンです。');
```



PGroongaの使い方

```
SELECT *  
  FROM memos  
 WHERE content &@ '全文検索';
```

```
-- id | content  
-- ---+-----  
--  2 | Groongaは日本語対応の高速な全文検索エンジンです。  
-- (1 row)
```



PGroongaの使い方

簡単ですね！



PGroongaの速度

追加されたデータ
は**即時**検索可能



PGroongaの速度

つまり



PGroongaの速度

検索可能になるの
が**速い**



PGroongaの速度

更新中も検索性能
が落ちない



PGroongaの速度

ヒット数が多くて
も検索性能が
落ちない



PGroongaの速度

つまり



PGroongaの速度

安定して速い



PGroongaの速度

もちろん
検索も**速い**



PGroongaの速度

- ベンチマーク
 - <https://pgroonga.github.io/ja/reference/pgroonga-versus-pg-bigm.html>



PGroongaの特徴

継続してメンテナ
ンスされている



PGroongaの特徴

PostgreSQL13
にも対応済み！



参考情報

- PGroongaについてのより詳しい解説
 - <https://www.slideshare.net/kou/postgresql-conference-japan-2017>
- 公式ドキュメント
 - <https://pgroonga.github.io/ja/>



今日のテーマ

PGroongaの
WALを
うまく使おう



PGroongaの WALは 復旧時に使用する



PGroongaのWAL

なぜWALを使うのか？



PGroongaのWAL

PGroongaは
インデックス
サイズが**大きい**



WALがない場合
REINDEXで復旧



PGroongaのWAL

データ量が多いと
REINDEXに時間
かかる



PGroongaのWAL

復旧が長引く



PGroongaのWAL

WALを使って
復旧時間を短縮



PGroongaのWAL

使うだけなら以下を
postgresql.conf に書いただけ

```
pgroonga.enable_wal = on
```


ただ使っている
だけだと、
困ることがある

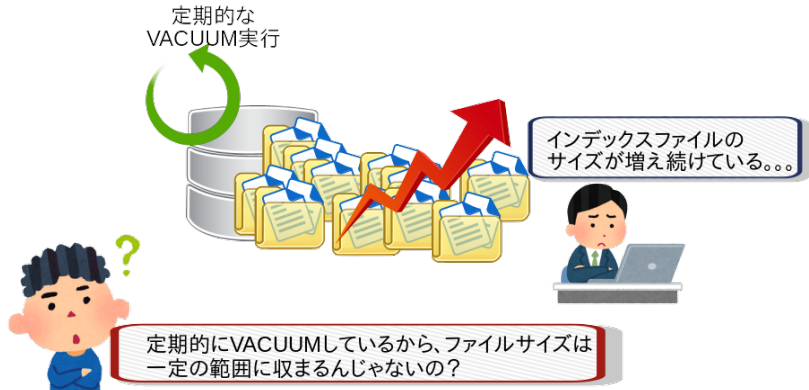


PGroongaのWALの注意点

例えば

- PGroongaのインデックスのファイルサイズが増加し続ける
 - 定期的にVACUUMしているのに...

PGroongaのレプリケーション





ファイルサイズ増加の原因

原因は
**PGroongaの
WAL**



ファイルサイズ増加の原因

PGroongaの
WALは削除しな
いと
増え続ける



ファイルサイズ増加の原因

1. PGroongaの**WAL**が有効
2. **何もせずにずっと運用**



ファイルサイズ増加の原因

ストレージが食いつぶされていく...



ファイルサイズの増加がとまらない





ファイルサイズの
増加を防ぐには？



定期的な メンテナンス



PGroongaのWALの運用方法

1. REINDEXによるインデックス再構築
2. WALを定期的に削除



REINDEX

- REINDEXすると、インデックスを0から再生成
- 古いインデックスは、VACUUM時に削除
 - このときにWALも削除される



REINDEX

REINDEXは
時間がかかる



WALの定期的な削除

`pgroonga_wal_truncate`



WALの定期的な削除

以下のように使う

```
SELECT pgroonga_wal_truncate('インデックス名');
```



WALの定期的な削除

未適用のWALを
削除しないように
注意



削除前に確実な
WALの適用が
必要



PGroongaのWALの運用方法

`pgroonga_wal_apply`



PGroongaのWALの運用方法

以下のように使う

```
SELECT pgroonga_wal_apply('インデックス名');
```



メンテナンス方法の選択

1. REINDEXによるインデックス再構築
 2. WALを定期的に削除
- のどちらが良いのか？



メンテナンス方法の選択

- REINDEX
 - 時間がかかる
 - REINDEX中は検索、更新がブロック
 - 実行は手間ではない



メンテナンス方法の選択

- WALの定期的な削除
 - REINDEXと比較して時間はかからない
 - テーブルのロックが必要
 - (レプリケーションしている場合は)すべての待機系でWALの適用が必要



メンテナンス方法の選択

復旧にかかる時間
をどれくらい許容
できるか？



メンテナンス方法の選択

例えば、以下のようなケースなら
REINDEが手間がかからなくて良い

1. 夜間はサービス提供しない
2. REINDEX実行が夜間で完了



メンテナンス方法の選択

例えば、以下のようなケースなら
WALの削除を選択

1. 長時間のサービス停止が難しい



最後に

- PGroongaについての疑問等は、GitHub、Gitterでも受付
- ドキュメントも充実
 - <https://pgroonga.github.io/ja/>



最後に

より突っ込んだお話がしたい場合は

↓ ↓

問い合わせ先：

<https://www.clear-code.com/contact/>