

# PostgreSQLで 高速・高機能な日本語 全文検索

堀本泰弘

株式会社クリアコード

*July Tech Festa 2021 winter*  
2021-01-24





# 自己紹介



堀本 泰弘

全文検索エンジン

gr<sup>o</sup>nga pgr<sup>o</sup>nga

mr<sup>o</sup>nga rr<sup>o</sup>nga

の開発・サポート



# 今日のテーマ

# PostgreSQLで 高速・高機能な日 本語全文検索 を実現する



# 今日のテーマ

...の前に

**全文検索？**



# 全文検索？

全ての文書から特定の文字列を検索する



# 全文検索？



Google and the Google logo are trademarks of Google LLC.



# 全文検索？

```
% grep PGroonga ./*
./docker-compose.yml: PGroonga:
./docker-compose.yml:     POSTGRES_DB: PGroonga
./docker-compose.yml:     POSTGRES_PASSWORD: PGroonga
./docker-compose.yml:     POSTGRES_USER: PGroonga
```





# 全文検索の対象

## 大量のテキスト

- 例：Redmineのwiki・チケット
- 例：チャットログ
- 例：口コミ



# PostgreSQLの全文検索

- LIKE : 組み込み
- textsearch : 組み込み
- pg\_trgm : 標準添付
- pg\_bigm : プラグイン
- PGroonga : プラグイン



# PostgreSQLの全文検索

- **LIKE** : 組み込み
- **textsearch** : 組み込み
- pg\_trgm : 標準添付
- pg\_bigm : プラグイン
- **PGroonga** : プラグイン



# 参考

- pg\_trgmとpg\_bigmとPGroongaの比較は以下の記事を参照
  - <https://groonga.org/ja/blog/-2016/11/30/pgroonga-1.1.9.html>



# PostgreSQLの全文検索

名前	日本語	速度	機能
LIKE	○	△	×
texts- earch	×	○	△
PGroo- nga	○	○	○



# LIKE

- メリット
  - 標準で使える
  - インデックス作成不要(データ更新が遅くならない)
  - データが少なければ十分高速



# LIKE

- デメリット
  - データ量に比例して遅くなる
  - 類似文書検索、同義語検索等の便利な機能はない



# textsearch

- メリット
  - 標準で使える
  - インデックスを作成するので、データ量が多くても高速
  - 同義語検索、検索結果のランキング、結果のハイライトなど、便利な機能がある





# textsearch

- デメリット
  - 言語毎にモジュールが必要
    - 英語やフランス語のモジュールは組み込み
    - 日本語は別途インストールが必要
    - 日本語のモジュールは現在メンテナンスされていない

**PGroongaを使った全文検索**



# PGroonga ?

# 読み方



# PGroonga ?

PGroonga  
(ぴーじーるんが)



# PGroongaとは？

PostgreSQLで  
高速・高機能な全文検索を実現  
する拡張



# PGroongaの特徴

1. **簡単**に使える
2. **速い**
3. **全言語**対応



# 使い方



専用のクエリー  
を覚えなくてOK!





# 使い方

実際に全文検索  
してみましよう





# 実行例：テーブル定義

```
CREATE TABLE entries (  
  title text,  
  content text  
);
```



# 実行例： インデックス定義

```
-- 全文検索用インデックス  
CREATE INDEX entries_full_text_search  
ON entries  
-- 「USING pgroonga」= 「PGroongaを使う」  
USING pgroonga (title, content);
```



# 実行例：データ挿入

```
-- 普通に挿入するだけでよい  
INSERT INTO entries  
VALUES ('PGroongaで高速全文検索！',  
       '高速に全文検索したいですね！');
```



# 実行例：全文検索

```
SELECT title FROM entries
WHERE
-- &@~で全文検索
-- 「検索」と「高速」をAND検索
title &@~ '検索 高速' OR
content &@~ '検索 高速';
```



# 実行例 : LIKE

```
SELECT title FROM entries  
WHERE
```

```
-- LIKEでもインデックスが効く  
-- = アプリを書き換えずに高速化可能  
-- ただし~より性能が落ちる  
title LIKE '%検索%' OR  
content LIKE '%検索%';
```



# 使い方

簡単ですね！



# 速度

安定して速い

ベンチマーク





# 青空文庫の書籍一覧

件数	LIKE 速度[ms]	PGroonga 速度[ms]
11,818件	1.916	0.290



# 日本全国の住所

件数	LIKE 速度[ms]	PGroonga 速度[ms]
149,724件	17.277	0.850

# Wikipedia日本語版のタイトル

件数	LIKE 速度[ms]	PGroonga 速度[ms]
3,677,375 件	128.776	0.371



# ベンチマークのデータ

- 使用したSQLは以下を参照
  - <https://github.com/komainu8/-rabbit-slide-komainu8-july-tech-festa-2021-winter/tree/master/-benchmark>



# ベンチマークのデータ

- 追試用スクリプト
  - <https://github.com/komainu8/-rabbit-slide-komainu8-july-tech-festa-2021-winter/blob/master/-exec-benchmark.sh>
  - 使い方はREADME参照



# 機能

- 全文検索に必要なような機能は一通り揃っている
  - 同義語検索
  - 類似文書検索
  - 読みがな検索
  - 入力補完 etc..



# 機能

- 全文検索に必要なような機能は一通り揃っている
  - 同義語検索
  - 類似文書検索
  - **読みがな検索**
  - 入力補完 etc..



# 読みがな検索

「やきにく」  
ってどう書きます  
か？





# 読みがな検索

- やきにく
- 焼き肉
- 焼肉
- やき肉
- ヤキニク



# 読みがな検索

当然ですがどれも  
「やきにく」と読  
みます



# 読みがな検索

- 読みが同じなので、以下は全部同じものとして扱えます
  - やきにく
  - 焼き肉
  - 焼肉
  - やき肉
  - ヤキニク



# 読みがな検索

例えば  
「やきにく」で検  
索すると



# 読みがな検索

- 「やきにく」 **Hit!**
- 「焼き肉」 **Hit!**
- 「焼肉」 **Hit!**
- 「やき肉」 **Hit!**
- 「ヤキニク」 **Hit!**



## 異体字



# 読みがな検索

「広」と「廣」



# 読みがな検索

例えば人名の  
検索





# 読みがな検索

検索キーワード「広瀬」で

- 「広瀬」 **Hit**
- 「廣瀬」 **Hit**

となってほしい



# 読みがな検索

通常 of 検索  
検索キーワード「広瀬」で

- 「広瀬」のみ**Hit**



# 読みがな検索

読みがな検索なら  
検索キーワード「広瀬」で

- 「広瀬」 **Hit**
- 「廣瀬」 **Hit**



# 読みがな検索

両方ヒット！



# 読みがな検索

「広瀬」も  
「廣瀬」も  
読みが同じ



# 他にも

- それっぽい順でソート
- キーワードハイライト
- キーワードの周辺テキスト表示



# 他にも

- 電話番号検索
  - 090-1234-5678 と 090 1234 5678、(090)1234-5678 等
- fuzzy検索
  - typo対策
    - テクノロジーとテノクロジー



# まとめ

PGroongaで  
**高速高機能**な  
日本語全文検索が  
実現できます！



より詳しく知りたい人



# 参考

- PGroonga自体の解説
  - <https://www.slideshare.net/kou/-postgresql-conference-japan-2017>



# 参考

- PHPのマニュアル検索
  - <https://www.slideshare.net/kou/phpconference2017>
- Redmineのチケットを検索
  - <https://www.slideshare.net/kou/redminetokyo12>
  - [https://github.com/clear-code/redmine\\_full\\_text\\_search](https://github.com/clear-code/redmine_full_text_search)



# 参考

- AWSでPGroongaを使う
  - <https://slide.rabbit-shocker.org/-authors/komainu8/postgresql-conference-japan-2019/>