PostgreSQLで日本語全文検索

LIKEとpg_bigmとPGroonga

須藤功平　株式会社クリアコード

PostgreSQLアンカンファレンス@東京
2015-05-30

内容

PostgreSQLで使える
日本語全文検索方法を
順に紹介

1: 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速い
- デメリット
  - データ量に比例して遅くなる

「少ない」データ？

どのくらいなら
少ないのか
↓
計測結果と要件
で判断

計測

pg_bigmでいろんなデータを
日本語検索してみよう！
http://qiita.com/fujii_masao/items/87f1d94ff4d350a718aa

- 青空文庫の書籍一覧データ
- 住所データ
- 日本版Wikipediaの
  タイトル一覧データ

青空文庫：作品名

- データ：
  - 11,818件
  - 1レコード平均17バイト
- 速度：
  - 6.673ms

十分速い

住所データ：市区町村

- データ：
  - 147,769件
  - 1レコード平均14バイト
- 速度：
  - 70.684ms

十分速い

Wikipedia：タイトル

- データ：
  - 2,461,588件
  - 1レコード平均20バイト
- 速度：
  - 943.450ms

十分速い？

LIKEの計測結果

件数	平均サイズ	速度
11,818	17バイト	6.673ms
147,769	14バイト	70.684ms
2,461,588	20バイト	943.450ms

十分速いならLIKEでOK！

LIKE以外の選択肢

- pg_bigm
- PGroonga

2: pg_bigm

- メリット
  - データ量が多くても高速
  - ストリーミングレプリケーション可
- デメリット
  - 別途インストールしないといけない
  - インデックス作成が遅い
  - ヒット数に比例して遅くなる

遅い？

計測して確認

備考:

- LIKEでは計測していないがLIKEよりは確実に速い
- LIKEはヒット数に依らずすべて同程度の検索時間になる

計測

PGroongaとpg_bigmのベンチマーク結果

https://github.com/groonga/wikipedia-search/issues/2

- データ
  - 日本語版Wikipediaの本文
  - 1,846,514件
  - 1レコード平均3777バイト

インデックス作成

元データの ロード時間	インデックス 作成時間
16分31秒	5時間56分15秒

遅い？そうでもない？

ヒット数と検索時間

ヒット数	検索時間
361	0.107s
17,168	1.224s
22,885	2.472s
625,792	0.556s(*)

(*) 検索語が2文字以下ならヒット数が増えても遅くならない

(*) work_memを10MBに増やしている

遅い？そうでもない？

pg_bigmの計測結果

インデックス作成時間: 約6時間

検索時間

ヒット数	検索時間
361	0.107s
17,168	1.224s
22,885	2.472s
625,792	0.556s

遅くないならpg_bigmでOK！

残りの選択肢

- PGroonga

3: PGroonga

- メリット
  - インデックス作成が速い
  - データ量が多くても高速
  - ヒット数が多くても高速
- デメリット
  - 別途インストールしないといけない
  - ストリーミングレプリケーション×

インデックス作成

元データの ロード時間	インデックス 作成時間
16分31秒	25分37秒

インデックス作成：比較

PGroonga	pg_bigm
25分37秒	5時間56分15秒

非常に高速

ヒット数と検索時間

ヒット数	検索時間
368	0.030s
17,172	0.121s
22,885	0.179s
625,792	0.646s(*)

(*) work_memを10MBに増やしている

(*) 直接Groongaで検索すると0.085s

検索時間：比較

ヒット数	PGroonga	pg_bigm
368	0.030s	0.107s
17,172	0.121s	1.224s
22,885	0.179s	2.472s
625,792	0.646s	0.556s

非常に高速

おねがい

同じベンチマークを実行して
結果を貼ってください！

同じ傾向があるか確認したい

ベンチマークの実行方法↓

（ここでまとめたデータの生データも貼ってある）
https://github.com/groonga/wikipedia-search/issues/2

まとめ1

- データが少ないならLIKEで十分
  - 1レコード数十バイトなら
    百万件はいける
- データが多いならLIKEはツライ

まとめ2

- データ多→pg_bigmかPGroonga
  - 2文字以下での全文検索がほとんど→pg_bigm
  - ストリーミングレプリケーション要→pg_bigm
  - ヒット件数が多い→PGroonga
  - レコードサイズが大きい→PGroonga
  - 更新が多い→PGroonga
    （インデックス作成が速いから）

参考情報

PGroongaでも
レプリケーションできる！

pg_shardとPGroongaを使った
レプリケーション対応の
高速日本語全文検索可能な
PostgreSQLクラスターの作り方
http://www.clear-code.com/blog/2015/5/18.html