Mroonga
と
PGroonga
初心者向け情報

須藤功平　クリアコード

MySQLとPostgreSQLと日本語全文検索
2016-06-09

Mroonga・PGroonga

- Mroonga（むるんが）
  - MySQLに
    高速日本語全文検索機能を追加する
    プロダクト
- PGroonga（ぴーじーるんが）
  - PostgreSQLに
    高速日本語全文検索機能を追加する
    プロダクト

使いどころ

- Mroonga
  - 速さが欲しい
  - トランザクションはいらない
- PGroonga
  - 機能が欲しい
  - トランザクションも欲しい

速い？

速さ：検索1

キーワード：テレビアニメ
（ヒット数：約2万3千件）

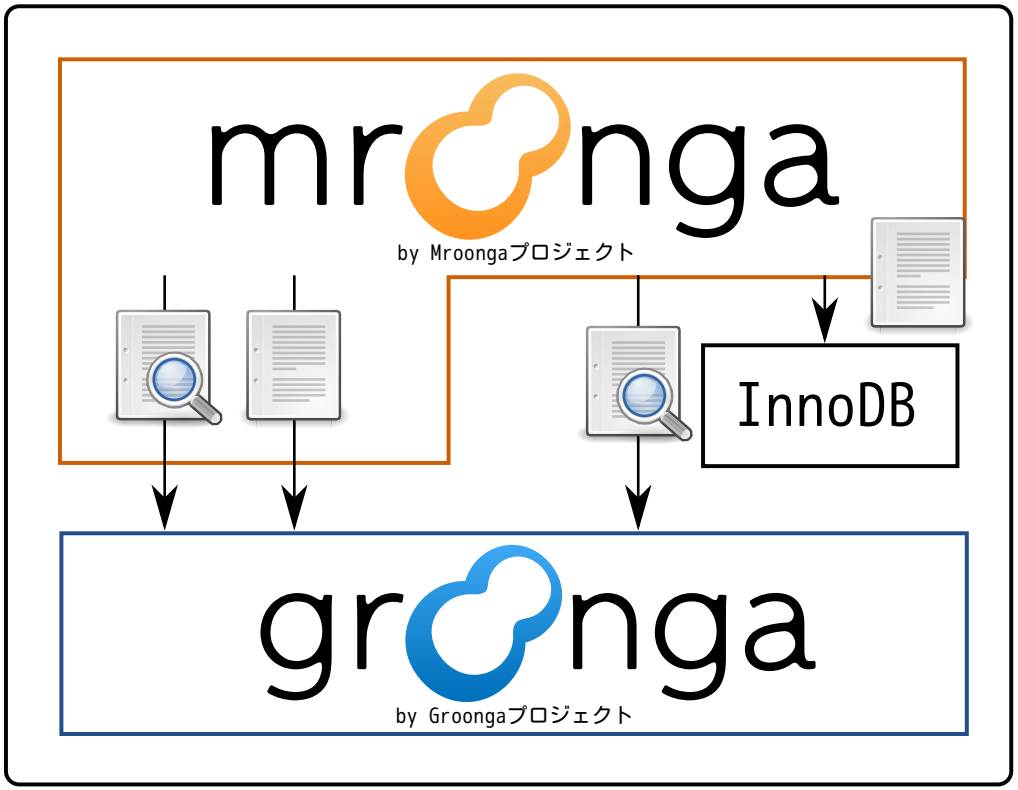
InnoDB ngram	3m2s
InnoDB MeCab	6m20s
Mroonga:1	0.11s
pg_bigm	4s
PGroonga:2	0.29s

詳細は前回の資料を参照
http://slide.rabbit-shocker.org/authors/kou/mysql-and-postgresql-and-japanese-full-text-search/

Mroongaのモード

- ストレージモード
  - ストレージとインデックスに Groongaを使う
- ラッパーモード
  - ストレージは 既存ストレージエンジンをラップ
  - 全文検索・位置情報インデックスだけ Groongaを使う

Mroongaのモード



MySQL

データ

インデックス

ストレージ
モード

ラッパー
モード

ストレージモード

- 全文検索が速い
- それ以外の条件も速い
  数値の比較とか
- 挿入も更新も削除も速い
- ソートも速い
- トランザクションはない

ラッパーモード

- 全文検索が速い
- 全文検索以外はラップ対象のストレージエンジンに依存
- トランザクション
  - ストレージのみ
  - インデックスには効かない
    ロールバックしたらインデックスの再構築が必要

使い分け

- できればストレージモード
- ムリならラッパーモード

ストレージモードの条件

- NULLがないこと
  - あるならテーブルをわけるか
    ラッパーモード
- トランザクションを使わない
  - 例：追記のみ（ログとか）
  - 例：同時に更新されないようにする
  - 例：スレーブにする（後述）

オススメの使い方

- テーブル定義
  - デフォルトで使う（いい感じになる）
  - カスタマイズは慣れてからで十分
- 検索
  - IN BOOLEAN MODEを使う（いつもの検索）
  - *D+プラグマを使う（デフォルトAND）

テーブル定義

```
CREATE TABLE items (
  title TEXT,
  FULLTEXT INDEX (title)
  -- ↑COMMENTでカスタマイズしない
) ENGINE=Mroonga
  DEFAULT CHARSET=utf8mb4;
```

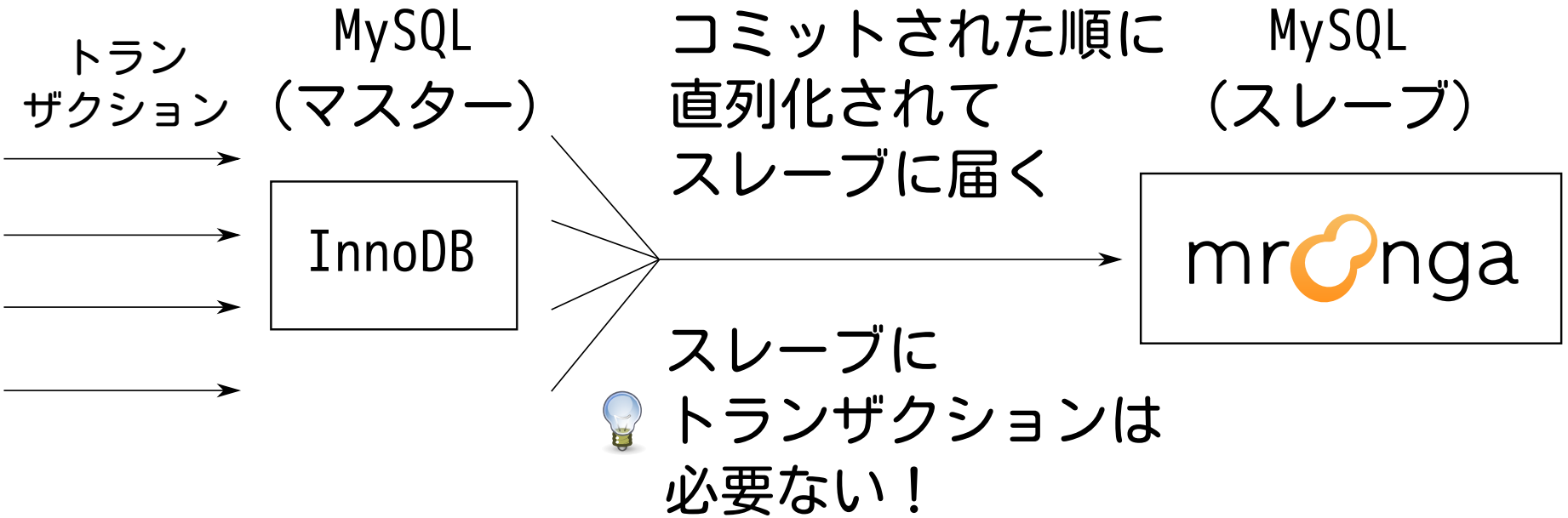
検索

```
SELECT * FROM items
  WHERE
    MATCH(title)
          -- ↓*D+プラグマ
    AGAINST('*D+ 激安 人気'
            IN BOOLEAN MODE);
        -- ↑IN BOOLEAN MODE
```

トランザクション欲しい！

- レプリケーションで対応可能
- 構成
  - マスター：InnoDB
  - スレーブ：Mroonga
  - MySQLはマスターとスレーブで
    違うストレージエンジンを使える！

トランザクションとレプリケーション



レプリケーション手順1

マスターmy.cnf:

```
[mysqld]
log-bin=mysql-bin
server-id=1
```

バイナリーログを有効にしてサーバーIDを指定

レプリケーション手順2

スレーブmy.cnf:

```
[mysqld]
server-id=2
```

サーバーIDを指定

レプリケーション手順3

マスター:

```
CREATE USER 'repl'@'%' IDENTIFIED BY '1qazXSW@';
GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%';
```

レプリケーション用ユーザーを作成

レプリケーション手順4

マスター:

```
SHOW MASTER STATUS\G
--                File: mysql-bin.000001
--            Position: 855
--        Binlog_Do_DB:
--    Binlog_Ignore_DB:
-- Executed_Gtid_Set:
```

バイナリーログの情報を確認

レプリケーション手順5

スレーブ:

```
CHANGE MASTER TO
  MASTER_HOST='192.168.0.9',
  MASTER_LOG_FILE='mysql-bin.000001',
  MASTER_LOG_POS=855;
START SLAVE USER='repl'
            PASSWORD='1qazXSW@';
```

レプリケーション開始

レプリケーション手順6

マスター:

```
CREATE TABLE items (
  title text
) DEFAULT CHARSET=utf8mb4;
```

マスターでテーブル作成

レプリケーション手順7

スレーブ:

```
ALTER TABLE items
  ENGINE=Mroonga,
  ADD FULLTEXT INDEX (title);
```

スレーブだけMroongaに変更

レプリケーション手順8

マスター:

```
INSERT INTO items
  VALUES ('データベース管理システム');
```

マスターでデータ登録

レプリケーション手順9

スレーブ:

```
SELECT * FROM items
  WHERE MATCH(title)
        AGAINST('*D+ データ 管理'
                IN BOOLEAN MODE);
-- +--------------------------------------+
-- | title                                |
-- +--------------------------------------+
-- | データベース管理システム             |
-- +--------------------------------------+
```

スレーブで全文検索

まとめ

- できればストレージモード
- テーブル定義：デフォルトでOK
- 検索：IN BOOLEAN MODEと*D+
- トランザクション：スレーブ

PGroongaのいいところ

- 検索条件を柔軟に指定可能
  - MySQL：拡張はAGAINST()内で頑張る
  - PostgreSQL：演算子を追加可能
- トランザクション対応
  - ロールバックも効く

オススメの使い方

- テーブル定義
  - 主キーを指定する
- インデックス定義
  - デフォルトで使う（いい感じになる）
- 検索
  - search_path設定→@@演算子を使う

テーブル定義

```
CREATE TABLE items (
  id integer PRIMARY KEY,
  title text
);
```

スコアー取得時に必要なので主キーを指定

インデックス定義

```
CREATE INDEX pgroonga_items_index
          ON items
       USING pgroonga (id, title);
```

主キーをインデックス対象にする

search_pathを設定

```
-- 現在のセッションのみ有効
SET search_path TO "$user",public,pgroonga,pg_catalog;
-- user1ユーザーのみ有効
ALTER ROLE user1
  SET search_path TO "$user",public,pgroonga,pg_catalog;
-- db1データベースでは永続的に有効
ALTER DATABASE db1
  SET search_path TO "$user",public,pgroonga,pg_catalog;
```

pg_catalogの前にpgroongaを入れる

なぜ設定する必要があるか

PostgreSQLに@@があるから

- 組み込みの@@より
  PGroongaの@@を優先したい
- 優先しないと
  インデックススキャンと
  シーケンシャルスキャンで
  結果が異なる

@@演算子で検索

```
SELECT *,
       pgroonga.score(items) AS score
  FROM items
  WHERE title @@ '激安 人気'
  ORDER BY score DESC;
```

独自演算子がうれしい例

入力補完

https://github.com/pgroonga/pgroonga/tree/master/examples/completion

テーブル作成

```
CREATE TABLE dictionary (
  term text,
  readings text[], -- 配列
  english text
);
```

データ

```
INSERT INTO dictionary VALUES
('.',
 ARRAY['ナカグロ', 'ナカポチ'],
 '(n) middle dot ...'),
-- (...),
('踵',
 ARRAY['カカト', 'キビス', 'クビス'],
 '(n) (uk) heel ...');
```

インデックス定義

```
CREATE INDEX pgroonga_index
  ON dictionary
  USING pgroonga (
    -- ↓前方一致・前方一致RK検索用
    term     pgroonga.text_term_search_ops_v2,
    -- ↓配列に対する前方一致・前方一致RK検索用
    readings pgroonga.text_array_term_search_ops_v2);
```

検索：独自演算子+OR

```
SELECT term, readings, english
  FROM dictionary
 WHERE term &^ 'nak' OR
       -- ↑前方一致検索
       readings &^~> 'nak'
       -- ↑前方一致RK検索
 ORDER BY term LIMIT 10;
```

結果例：nak

前方一致RK検索でヒット

```
     term      |     readings          |      english
---------------+-----------------------+--------------------
 ・            | {ナカグロ,ナカポチ}   | (n) middle dot ...
 ＮａＫ        | {ナック}              | (n) NaK (sodiu...
 なくても良い  | {ナクテモヨイ         | (exp) (1) (uk) ...
 ...           |     ...               | ...
(10 rows)
```

結果例：なか

前方一致RK検索でヒット

```
     term      |     readings          |      english
---------------+-----------------------+--------------------
 ・            | {ナカグロ,ナカポチ}   | (n) middle dot ...
 泣かされる    | {ナカサレル}          | (v1,vi) (1) to ...
 泣かせる      | {ナカセル}            | (v1,vt) (1) to ...
 ...           |     ...               | ...
(10 rows)
```

結果例：中

前方一致RK検索でヒット

```
    term     |     readings          |      english
-------------+-----------------------+-------------------
 中          | {ウチ,ジュウ,チュウ}   | (n,adj-no) (1) ...
 中々        | {ナカナカ}            | (adv,adj-na) ..
 中２        | {チュウニ}            | (n) second-year
 ...         |     ...               | ...
(10 rows)
```

レプリケーション

- PostgreSQL標準機能：×
  - プラグインはWALを使えない
- pglogical：○
  詳細：http://www.clear-code.com/blog/2016/3/22.html
- PostgreSQL 9.6：たぶん○
  - プラグインでもWALを使える
  - 正式リリースまでに対応したい

まとめ

- トランザクション対応
- テーブル定義：主キーを指定
- インデックス定義：デフォルト
- 検索：search_pathと@@
- レプリケーション：pglogical

相談場所

- メーリングリスト
  - groonga-dev
- チャット
  - https://gitter.im/groonga/ja
- Twitter：@groonga
- 勉強会

勉強会

「Groongaで学ぶ全文検索」

- 全文検索について学ぶ会
- 内容は参加者が知りたいこと
- ほぼ隔週金曜夜開催
  - 次回は6月17日
    https://groonga.doorkeeper.jp/events/45556