

Perlと全文検索エンジン GroongaでMySQLの データを高速に全文検索する

堀本 泰弘 株式会社クリアコード

YAPC::Kyoto 2023
2023-03-19







自己紹介



堀本 泰弘

全文検索エンジン

grnga pgrnga

mrnga rrnga

の開発・サポート



今日のテーマ

MySQLのデータを
を**高速**に全文検索
する！



よくあるアプローチ

- InnoDBで全文検索
- Elasticsearchで全文検索
- Elasticsearch+Logstashで全文検索

InnoDBで全文検索



InnoDBで全文検索 メリット

- 手軽に全文検索できる 😊
- 専用の構文を覚えなくていい 😊
- メンテナンスコスト：小 😊



InnoDBで全文検索 デメリット

- 遅い 🙄
- 機能不足 🙄



使い方

```
CREATE TABLE products (  
  product_id int DEFAULT NULL,  
  product_title text,  
  product_description text,  
  product_brand text,  
  FULLTEXT KEY fts_description_idx  
    (product_title, product_description, product_brand) WITH PARSER ngram  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```




使い方

```
SELECT * FROM products
WHERE MATCH(product_title, product_description, product_brand)
AGAINST('+ハンドバッグ' IN BOOLEAN MODE');
```



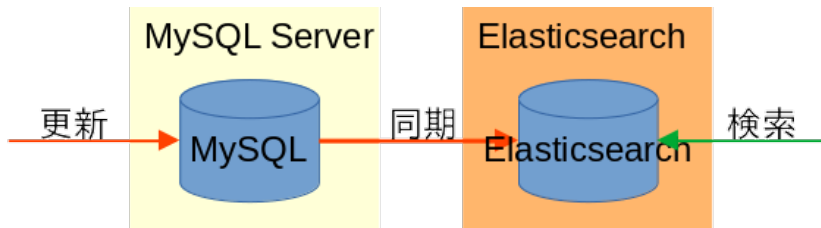
使い方

■ IN BOOLEAN MODE

- 検索文字列の先頭の特定の文字が特別な意味を持つ
 - +:AND検索
 - -:NOT検索
 - 演算子なし : OR検索

Elasticsearchで全文検索

Elasticsearchで全文検索 構成





Elasticsearchで全文検索 メリット

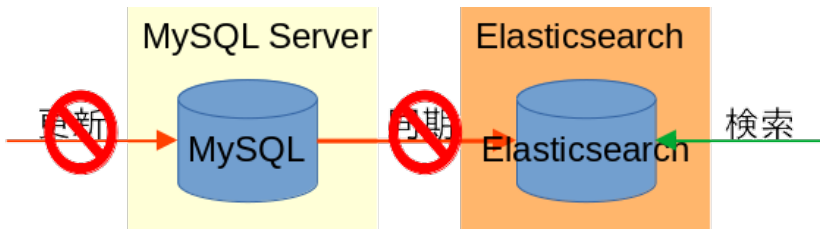
- InnoDBより速い 😊
- 多機能 😊



Elasticsearchで全文検索 デメリット

- アプリケーションの実装が複雑になる 🙄
 - 更新失敗時のケアが必要

Elasticsearchで全文検索 デメリット



Elasticsearch+Logstashで 全文検索

Logstashとは?



Logstashとは?

Elastic社開発の ログ収集ツール



Logstashとは?

データソースを
一元化して収集
できる


Elasticsearch+Logstash で全文検索 構成




Elasticsearch+Logstash で全文検索 メリット

- アプリケーションの実装を
シンプルにできる 😊

Elasticsearch+Logstash で全文検索 デメリット

- メンテナンスコスト：大 
 - Logstashの死活監視も必要

Elasticsearch+Logstash で全文検索 デメリット

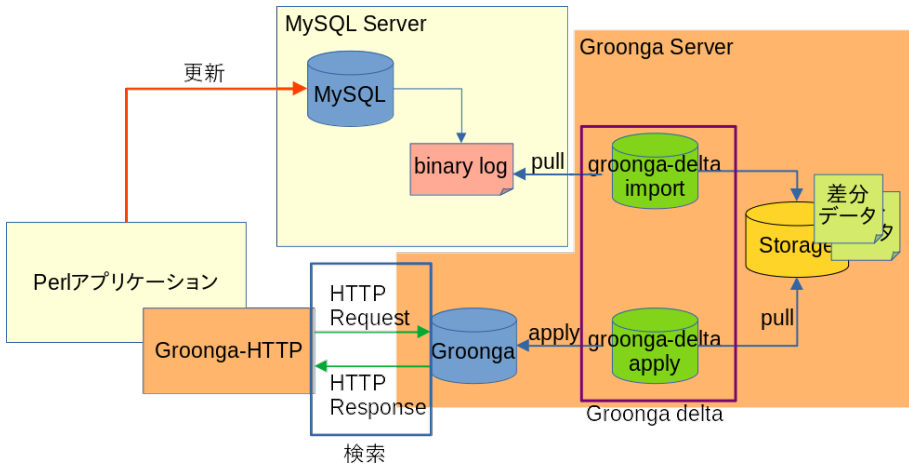
- データを削除するのに一手間必要

- Logstashはレコードの削除を検出できない
- deleteフラグを作って別途削除が必要

新しいアプローチ！

**MySQL + Groonga delta +
Groonga + Groonga-HTTP**



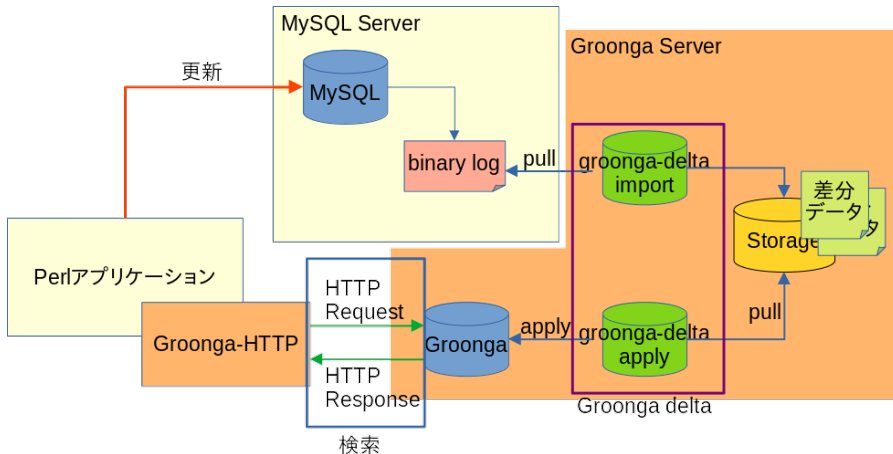
構成



Groongaとは？



Groongaとは？





Groongaとは？

オープンソースの
国産全文検索
エンジン



Groongaとは？

- 公式ドキュメント

<https://groonga.org/ja/>



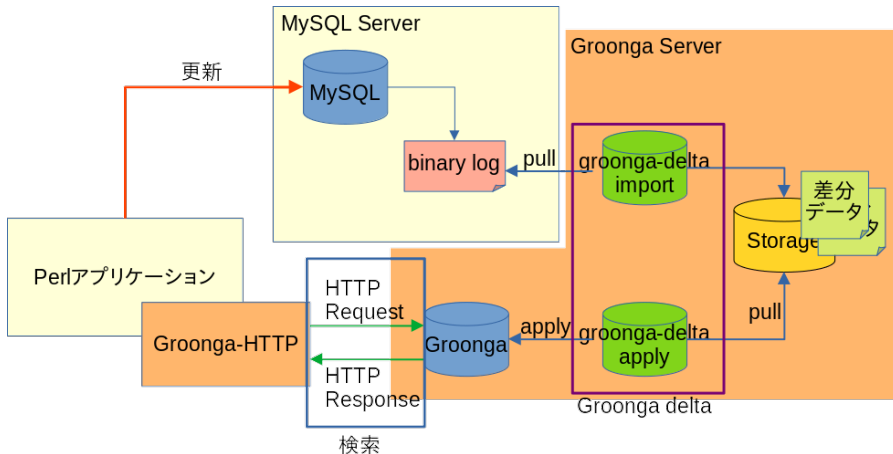
Groongaの特徴

- **高速**
- 登録した情報は即時検索可能
- 更新中でも検索性能が落ちない
- **多機能**（同義語、ハイライト、オートコンプリート、etc...）
- **HTTPでやり取りできる**

Groonga-HTTPとは？



Groonga-HTTPとは？





Groonga-HTTPとは？

GroongaのHTTP クライアント ライブラリー

Groonga-HTTPを使わない場合

Site

id	_key	title
1	http://- example- .org/	Example- .org
2	http://- example- .net/	Example- .net

Groonga-HTTPを使わない場合

```
use LWP::UserAgent;
use JSON;

$sua = LWP::UserAgent->new;
$sua->agent("MyApplication/1.0 ");

my $res = $sua->get('http://127.0.0.1:10041/d/select?table=Site');
if ($res->is_success) {
    my @decoded_contents = decode_json($res->content);
    my $n_hits = $decoded_contents[0][1][0][0][0];

    for (my $i = 2; $i < $n_hits+2; $i++) {
        print "\n";
        print "id=$decoded_contents[0][1][0][$i][0]\n";
        print "_key=$decoded_contents[0][1][0][$i][1]\n";
        print "title=$decoded_contents[0][1][0][$i][2]\n";
    }
}
```

Groonga-HTTPを使わない場合

```
id=1
```

```
_key=http://example.org/  
title=Example.org
```

```
id=2
```

```
_key=http://example.net/  
title=Example.net
```

インストール

```
% cpanm Groonga::HTTP
```



Groonga-HTTPを使う場合

```
use Groonga::HTTP;

my $groonga = Groonga::HTTP->new;
my $result = $groonga->select(table => 'Site');

for (my $i = 0; $i < $result->{'n_hits'}; $i++) {
    print "\n";
    print "$result->{'records'}[$i]->{'_id'}\n";
    print "$result->{'records'}[$i]->{'_key'}\n";
    print "$result->{'records'}[$i]->{'title'}\n";
}
```



Groonga-HTTPを使う場合

1

```
http://example.org/  
Example.org
```

2

```
http://example.net/  
Example.net
```




Groonga-HTTPの情報

公式ドキュメント

- <https://groonga.github.io/Groonga-HTTP/ja/>

Groonga deltaとは？

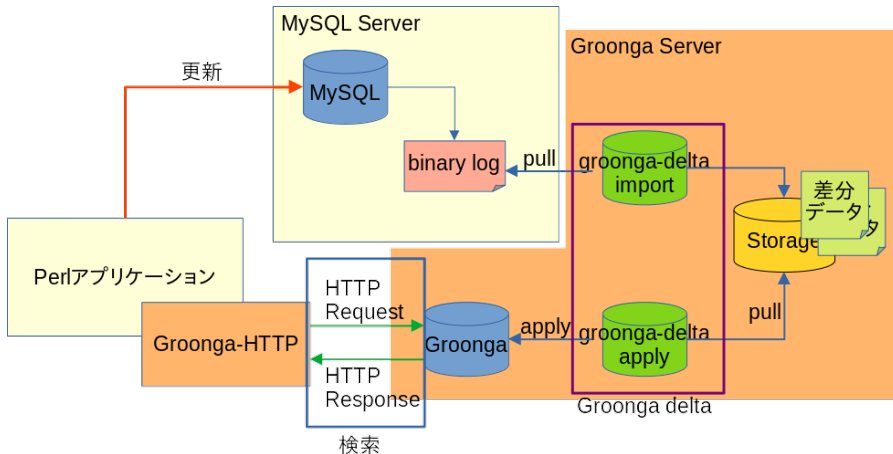


Groonga deltaとは？

MySQLのデータをリアルタイムでGroongaに同期するツール

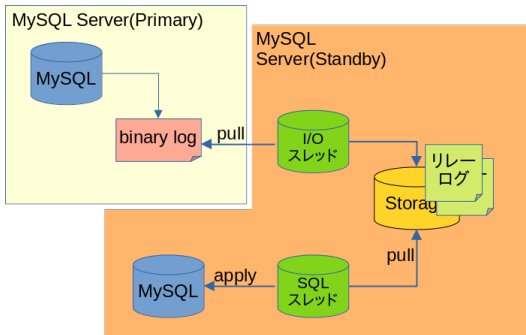


Groonga deltaとは？





MySQLの レプリケーションモデル



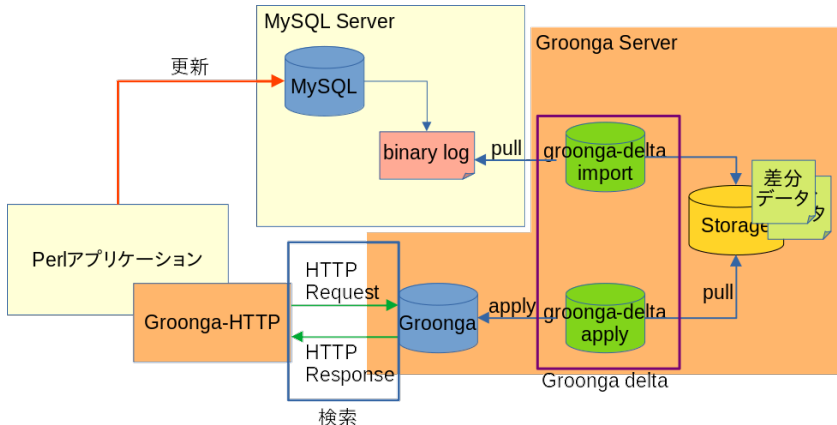


できること

- データの一括取り込み
- INSERT/UPDATE/DELETEの結果をリアルタイムで取り込む



MySQL+Groonga delta+Groongaの構成





MySQL+Groonga delta+Groongaの構成

データ更新は

MySQL

全文検索は

Groonga



MySQL+Groonga delta+Groongaなら！

- 速い 😊
- 多機能 😊
- アプリケーションをシンプルにできる！



MySQL+Groonga delta+Groongaなら！

- マスターデータはMySQL
GroongaのDBが破損したら、
切り捨てて新規に作り直すこと
ができる



MySQL+Groonga delta+Groongaなら！

MySQLのデータ
だけケアすれば
いい！



MySQL+Groonga delta+Groongaなら！

- Groongaが起動していなくてもデータの更新ができる

Groonga復旧後に再接続すると自動的にデータが同期される



MySQL+Groonga delta+Groongaなら！

Groongaに障害
が起きても更新を
ケアしなくて
いい！



MySQL+Groonga delta+Groongaなら！

- データの削除もできる 😊



MySQL+Groonga delta+Groongaなら！

- 専用の構文が必要 🙄
- メンテナンスコスト：大 🙄

他にもこんな方法が

Mroongaを使う



Mroongaとは？

MySQLの
ストレージ
エンジンとして
Groongaを
使える



Mroongaのメリット

- SQLが使える 😊
- 使うまでの設定がない
(パッケージをインストールして
終わり) 😊



Mroongaのデメリット

- Groongaに比べて遅い 🙄
- SQLを逸脱するような機能は使えない 🙄
- クラッシュセーフではない 🙄



ユースケース

どの方法が
どういうケースで
有効か？



ユースケース

- データ量が少ないケース
- あまり速度を求めているくないケース
- 気軽に全文検索したいケース



ユースケース

InnoDB



ユースケース

- 気軽に高速な全文検索を使いたいケース
- データリカバリーの方法が確立できている



ユースケース

Mroonga



ユースケース

- とにかく速度がほしいケース
- ファセット検索などGroongaにしかない機能を使いたいケース



ユースケース

MySQL +
Groonga delta
+ Groonga



まとめ

ユースケース

/	InnoDB	Mroonga	Groonga
データ 量大	×	○	◎
速度	×	○	◎
機能	×	○	◎
運用	◎	△	×



Groongaについての情報

- リリース自慢会

[https://www.youtube.com/playlist?-](https://www.youtube.com/playlist?list=PLLwHraQ4jf7PnA3GjI9v-90DZq8ikLk0iN)

[list=PLLwHraQ4jf7PnA3GjI9v-90DZq8ikLk0iN](https://www.youtube.com/playlist?list=PLLwHraQ4jf7PnA3GjI9v-90DZq8ikLk0iN)



Groongaについての情報

- グルカイ！

（Groonga開発者に聞け！）

[https://www.youtube.com/playlist?-](https://www.youtube.com/playlist?list=PLLwHraQ4jf7PBol9aq5B-oaM2Hnk6aCKs2)

[list=PLLwHraQ4jf7PBol9aq5B-oaM2Hnk6aCKs2](https://www.youtube.com/playlist?list=PLLwHraQ4jf7PBol9aq5B-oaM2Hnk6aCKs2)



サポート

Groongaや
Mroongaは
敷居が高いが
使いたい!



サポート

そんなときは



サポート

- チャット

<https://gitter.im/groonga/ja>



サポート

- GitHub Discussions

<https://github.com/groonga/-groonga/discussions>



サポート

- クリアコードのサポートサービス
<https://www.clear-code.com/-services/groonga.html>