

Groonga meetup #2

林健太郎

クリアコード

Groonga meetup
2014/04/29





はじめに

- 開催の目的
- 今回の内容
- Any Questions ?



開催の目的(1)

「全文検索を
したい人の最
初の一歩を踏
みだすしよ
う」



今回の内容(1)

- 全文検索事始め
- GroongaやMroongaの特長の紹介
- 最新版の導入について
- GroongaやMroongaの環境構築



今回の内容(2)

- 題材の説明と実際の演習(随時)
- Any Questions?



全文検索事始め

- 既存のシステムに全文検索導入したい
- 手持ちのメタ情報を効率良く検索したい



Groongaの特長(1)

- 全文検索と即時更新
 - 参照ロックフリー
 - 更新しつつ検索OK
- カラムストアを採用
 - 集計クエリを高速に処理可能



Groongaの特長(2)

- 豊富なトークナイザー
 - 検索漏れないことを重視しますか？
 - 検索精度を重視しますか？
- 位置情報検索のサポート



Groongaの特長(3)

- アプリケーションへの組み込み
 - ライブラリとして活用可能
 - 実はバックエンドでGroongaが動いていることも



Groongaとパッケージ

- Debian
- Ubuntu
- CentOS
- Fedora
- Windows



Groonga on Debian

- Debian 7.4 (wheezy)
- Debian 8 (jessie)
- Debian unstable (sid)



Groonga on Ubuntu(1)

- Ubuntu 12.04 (Precise)
- Ubuntu 12.10 (Quantal)
 - 明日でサポート終了 4.0.1まで
- Ubuntu 13.04 (Raring)
 - すでにサポート終了 3.1.2まで



Groonga on Ubuntu(2)

- Ubuntu 13.10 (Saucy)
 - 7月にサポート終了
- Ubuntu 14.04 (Trusty)
 - 今月からサポート開始



Groonga on CentOS

- CentOS 5/6
 - 中国語のトークナイザーは未サポート
- CentOS 7?
 - まだrcベースらしいのでしばらく先



Groonga on Fedora(1)

- Fedora 20
 - 中国語のトークナイザーは未サポート
 - Groonga公式とFedora公式に違いはない
 - Fedoraでリリースするのはちょっと遅れる(2-3週)



Groonga on Fedora(2)

- リリース短縮にはKarmaが必要

fedora

Logout
kerhys's Home
My Updates (26)
New Update
Buildroot Overrides
Fedora 20
Metrics
Pending (24)
Testing (393)
Stable (6305)
Security (344)
Fedora 19
Metrics
Pending (32)
Testing (338)

[groonga-4.0.0-1.fc20 bugfix update](#)

Status: [stable](#) ⓘ
Release: [Fedora 20](#)
Update ID: FEDORA-2014-2312
Builds: [groonga-4.0.0-1.fc20](#) (logs)
Pushed: True
Date Submitted: 2014-02-10 06:59:59
Date Released: 2014-02-11 22:23:44
Submitter: [kerhys](#)
Karma: 😊 1
Stable karma: 3
Unstable karma: -3
Close bugs: True



FedoraでKarmaを稼ぐ

- Bohdiにログイン
- メニューのTestingを選択
- Groongaパッケージを選択
- works for meコメントを追加

<https://admin.fedoraproject.org/updates/>

FedoraでKarmaを稼ぐ(1)

- Bohdiにログイン
- メニューのTestingを選択
- Groongaパッケージを選択
- works for meコメントを追加

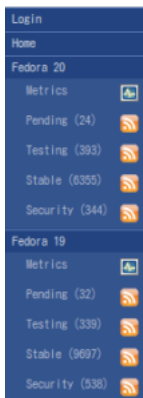
<https://admin.fedoraproject.org/updates/>

FedoraでKarmaを稼ぐ(2)

- Bohdiにログイン
- **メニューのTestingを選択**
- Groongaパッケージを選択
- works for meコメントを追加

FedoraでKarmaを稼ぐ(2)

■ Testingって？



The screenshot shows a vertical navigation menu for the Fedora Karma dashboard. It is divided into sections for Fedora 20 and Fedora 19. Each section includes a 'Metrics' link with a bar chart icon, and a list of build states with their respective counts and RSS feed icons.

Version	Category	Count
Fedora 20	Metrics	-
	Pending	24
	Testing	393
	Stable	6355
	Security	344
Fedora 19	Metrics	-
	Pending	32
	Testing	339
	Stable	9697
	Security	538

FedoraでKarmaを稼ぐ(3)

- Bohdiにログイン
- メニューのTestingを選択
- **Groongaパッケージを選択**
- works for meコメントを追加

FedoraでKarmaを稼ぐ(4)

- Bohdiにログイン
- メニューのTestingを選択
- Groongaパッケージを選択
- **works for me**コメントを追加

FedoraでKarmaを稼ぐ(4)

- works for me コメントって?
 - これを送信するとKarma +1

Feedback 

 [bodhi](#) - 2014-04-25 01:57:51
This update has been submitted for testing by clicku.

Add a comment

Untested Works for me Does not work



Mroongaの特長(1)

- 導入しやすいプラグイン形式
- 他のストレージエンジンとの連携
 - ラッパーモード



Mroongaの特長(2)

- データ更新が多くても検索性能が良い
- LIKEからの置き換えも比較的容易



MrOongaとパッケージ

- Debian
- Ubuntu
- CentOS
- Fedora
- Windows



MrOonga on Debian

- MySQL 5.5系
 - Debian 7.4 (wheezy)
 - Debian 8 (jessie)
 - Debian unstable (sid)



Mroonga on Ubuntu(1)

- MySQL 5.5系
 - Ubuntu 12.04 (Precise)
 - Ubuntu 12.10 (Quantal)
 - 明日でサポート終了 4.01まで
 - Ubuntu 13.04 (Raring)
 - すでにサポート終了 3.12まで



Mroonga on Ubuntu(2)

- MySQL 5.5系
 - Ubuntu 13.10 (Saucy)
 - 7月にサポート終了
 - Ubuntu 14.04 (Trusty)
 - 今月からサポート開始



Mroonga on CentOS 5

- CentOS 5
 - 以前はMySQLが古すぎたので、MySQL 5.6対応パッケージを提供
 - MySQL 5.5 (SCL) 対応パッケージに移行



Mroonga on CentOS 6

- CentOS 6
 - MySQL 5.1しか使えないというのは古い情報です！
 - MySQL 5.5対応(SCL)パッケージに移行



Mroonga on CentOS

- CentOS 5やCentOS 6でMySQL 5.6
 - wingリポジトリをお勧めしています

<http://wing-repo.net/>



Mroonga on Fedora

- Fedora 20
 - MySQL 5.5 & MariaDB 5.5対応
 - Fedora公式リポジトリでは未リリース
 - Mroonga含めてインストールしたいならGroongaリポジトリを登録する



環境構築してみよう

- Groongaのインストール
- Mroongaのインストール



どれをインストールしたらいい？(1)

- groonga
 - ちょっとお試して使うならコレ
 - コマンドラインで使うだけ
 - いろいろ指定すればサーバーとしても利用可



どれをインストールしたらいい？ (2)

- groonga-server-http
 - サーバー立てるならコレ
- groonga-server-gqtp
 - 特に理由がなければスルー



- groonga-httpd
 - 不満がでてきたらコレ
 - POSTしたいならコレ一択



- **groonga-httpd**
 - 不満がでてきたらコレ
 - POSTしたいならコレ一択



Install Groonga(1)

⌘ Groongaをインストールしてみましょう

- groonga-httpd

<http://groonga.org/ja/docs/install.html>



Install Groonga(2)

⌘ トークナイザーやノーマライザーをインストールしてみましょう

- groonga-tokenizer-mecab
- groonga-normalizer-mysql

<http://groonga.org/ja/docs/install.html>



Install Groonga(3)

⌘ Groongaが起動しているか確認しましょう

```
% curl http://localhost:10041/d/status
```



Install Groonga(4)

⌘ Groongaが起動していない場合は起動しましょう

```
% sudo service groonga-httpd start
```



Install Mroonga(1)

⌘ Mroongaをインストールしてみましょう

<http://mroonga.org/ja/docs/install.html>



Install Mroonga(2)

⌘ Mroongaが使えるか確認しましょう

```
mysql> show engines;
```



Install Mroonga(3)

⌘インストールされているとどうなる？

```
mysql> SELECT * FROM mysql.plugin;
+-----+-----+
| name   | dl      |
+-----+-----+
| mroonga | ha_mroonga.so |
+-----+-----+
1 row in set (0.00 sec)
```



Install Mroonga(4)

⌘インストールされているとどうなる？

```
mysql> SELECT * FROM mysql.func;
```

name	ret	dl	type
last_insert_grn_id	2	ha_mroonga.so	function
mroonga_snippet	0	ha_mroonga.so	function
mroonga_command	0	ha_mroonga.so	function
mroonga_escape	0	ha_mroonga.so	function

```
4 rows in set (0.00 sec)
```



今回の演習の題材

- 駅データ.jp
 - <http://www.ekidata.jp/>



題材のダウンロード

⌘以下から今回使うサンプルをダウンロードしてください

<http://packages.groonga.org/tmp/groonga-meetup-20140429.tar.gz>



Groonga編

- よく使うgroongaコマンド
 - table_create/table_remove
 - column_create/column_remove
 - select/load/delete/dump



テーブルのキーの話

- いろいろあるけど**TABLE_PAT_KEY**
- 長過ぎるキーは禁物 (ShortText)

```
table_create Company TABLE_PAT_KEY UInt32
```

データを追加/更新しよう

- 多才なloadコマンド
 - 追加も更新もこれ一つで

失敗時にわかりやすいロードの仕方(1)

- こうやってロードできるけど

```
load --table Company  
[  
  ["_key", "column1", "column2", ...  
  ["key1", "data1", "data2", ...
```

失敗時にわかりやすいロードの仕方(2)

- key:valueのペアがわかりやすい

```
load --table Company  
[  
{"_key": "key1", "column1":"data1", "column2":", ...
```

失敗時にわかりやすいロード の仕方(3)

⌘ 実際に試してみましよう

```
% 0001-load/load-fail.sh  
% 0001-load/load-succeed.sh
```

```
[[0, 1398668413.7959, 0.00316190719604492], 3]
```

データをより安全にロードする(3)

⌚ 実際に試してみましよう

⚠ **_key:3のstatusが抜けている**

```
% ./0001-load/load-fail.sh  
% ./0001-load/load-succeed.sh
```

```
[[0, 1398668413.7959, 0.00316190719604492], 3]
```

データを更新してみよう

⌘ 実際に試してみましよう

_key:3のstatusを更新します

```
% ./0002-update/update-status.sh
```


データを検索してみよう

⌘ 実際に試してみましよう

⚠ **1件もヒットしない**

```
% ./0003-search/search-name.sh
```

検索するにはインデックスが必要(1)

⌘ インデックスを追加してみましよう

```
% ./0003-search/add-index.sh
```

検索するにはインデックスが必要(2)

⌚ COLUMN_INDEXを指定するのがポイントです

```
table_remove Lexicon
table_create Lexicon TABLE_PAT_KEY ShortText --default_tokenizer ¥
  TokenBigram --normalizer NormalizerAuto
column_create Lexicon name_index COLUMN_INDEX|WITH_SECTION|WITH_POSITION Users name
column_create Lexicon sect_index COLUMN_INDEX|WITH_SECTION|WITH_POSITION Users sect
```

検索するにはインデックスが必要(3)

⌘再度検索してみましょう

```
% ./0003-search/search-name.sh
```

検索するにはインデックスが必要(4)

⌘ インデックスを覗いてみる

```
% ./0003-search/dump-lexicon.sh
```

データを削除してみよう(1)

⌘ キーをもとに削除する

```
delete Company 2
```

```
% 0004-delete/delete-by-key.sh
```

データを削除してみよう(2)

⌚ nameカラムの値をもとに削除する

```
delete Company --filter "name == 'JR北海道'"  
% 0004-delete/delete-by-name.sh
```

データを削除してみよう(3)

⌘ statusの値をもとに削除する

```
delete Company --filter 'status > 0'  
% 0004-delete/delete-by-status.sh
```


HTTP経由でのリクエストの基本

- コマンド名 + キー=値のペア

`http://localhost:10041/d/コマンド名?キー1=値1&キー2=値2....`

HTTP経由でテーブルを作成する(1)

⌘ table_createコマンドを使ってアクセス

```
% ./0005-create-table/create-table.sh  
curl 'http://localhost:10041/d/table_create?name=Company&type=TABLE_PAT_KEY&key_type=UInt32'
```

HTTP経由でカラムを作成する (1)

⌚ column_createコマンドを使って
アクセス

```
% ./0006-create-column/create-column.sh  
curl 'http://localhost:10041/d/column_create?table=Company&name=full_name&flags=COLUMN_SCALAR&type=ShortText'
```

POSTでデータを更新する(1)

⌚ POSTでデータをロードしてみま
しょう

```
% ./0007-post-data/post-data.sh
```

HTTPでデータを検索する(1)

⌘ HTTP経由で検索してみましよう

```
% ./0008-search-http/search-http.sh
```

管理画面を触ってみよう

🕒 ブラウザで次のURLにアクセスしてください

`http://localhost:10041/`

Mr oonga

編

モードの違いをおさえる(1)

- ラッパーモード
 - データは他のストレージエンジン
 - 全文検索をMroongaでがんばる
 - トランザクションでロールバックすると不整合発生
 - チューニングはMySQLの範囲で

モードの違いをおさえる(2)

- ストレージモード
 - 全部Mrroongaでがんばる
 - トランザクション何それ？



ストレージモード

- ENGINE=mroongaと覚えましょう



ラッパーモード

- コメントでエンジンを指定

```
ENGINE=mroonga COMMENT='engine "innodb"'
```



パーサーの指定

- コメントでパーサーを指定
- Groongaの用語だとトークナイザ

```
FULLTEXT INDEX (content) COMMENT 'parser "TokenMecab"'
```



ノーマライザーの指定

- コメントでノーマライザーを指定

```
FULLTEXT INDEX (content) COMMENT 'normalizer "NormalizerAuto"'
```

パーサー&ノーマライザーの指定

- 両方指定したいときは, で区切る

```
FULLTEXT INDEX (content) COMMENT 'parser "TokenMecab", normalizer "NormalizerAuto"'
```

MATCH AGAINSTとBOOLEAN MODE

- QUERY EXPANSION MODE
 - MySQLにおまかせの類似文書検索
- BOOLEAN MODE
 - 明示的に検索を制御



位置情報検索

☒ Qiitaの位置情報検索の記事を試してみよう

<http://qiita.com/groonga/items/7abca5004d2ae386d388>



感謝

- WEICさん
会場提供ありがとうございます！



おわり

- Any Questions?