

RaspberryPi2 に全文検索のための Groonga ( MariaDB + Mroonga ) を入れようかと思った次第。

みたいな感じになるのだろうか？ とりあえずやってみる。

## MariaDB のインストール

apt-get を使ってバイナリを入れるのが一番簡単なんですが、groonga のリポジトリには i386 用と amd 用しか用意されていない模様 (汗)

また mariadb-10.0.15 から mroonga がバンドルされたようなんですが、raspbian の jessie の apt-get でインストールできる mariadb-10.0.22 にはバンドルされていない。

MariaDB についてバンドルしてリリースされました

<http://mroonga.org/ja/blog/2014/11/29/release.html>

兎にも角にもソースから入れるしかないんですが、groonga や mecab との関係が良くわからないので、とりあえずコンパイルしてみた (汗)

で。

MariaDB-10.1.11 のソースに付いてる Mroonga の Groonga は 5.0.5、Mroonga は 5.04、normalizer も付いてて、MeCab も別途インストールしてあれば mecab-config から使えるようになってるっぽい。

```
MariaDB [(none)]> show variables like 'mroonga_libgroonga_version';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| mroonga_libgroonga_version | 5.0.5 |
+-----+-----+

MariaDB [(none)]> show variables like 'mroonga_version';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| mroonga_version | 5.04 |
+-----+-----+
```

しかし、折角ソースからコンパイルするなら最新版が良いかなと。この時点 (2016/2/5) での最新版は groonga-5.1.2 と mroonga-5.12。

とりあえず MariaDB のソースを取得。

Downloads - MariaDB

<https://downloads.mariadb.org/>

```
# cd /usr/local/src
# wget "https://downloads.mariadb.org/f/mariadb-10.1.11/source/mariadb-10.1.11.tar.gz/from/http%3A//ftp.yz.yamagata-u.ac.jp/pub/dbms/mariadb/?serve" -O mariadb-10.1.11.tar.gz
# tar zxvf mariadb-10.1.11.tar.gz
# cd mariadb-10.1.11
```

コンパイル用ドキュメント。

Generic Build Tnstructions

<https://mariadb.com/kb/en/mariadb/generic-build-instructions/>

manual で BUILD/autorun.sh から ./configure でもよさげな気がしたのですが、cmake が流行モノみたいなのでインスコ。

```
# apt-get install build-essential cmake
```

で、早速 cmake してみると ...

```
# cmake . (略)
:
-- Could NOT find PkgConfig (missing: PKG_CONFIG_EXECUTABLE)
:
-- Could NOT find Curses (missing: CURSES_LIBRARY CURSES_INCLUDE_PATH)
CMake Error at cmake/readline.cmake:85 (MESSAGE):
  Curses library not found. Please install appropriate package,
  remove CMakeCache.txt and rerun cmake.On Debian/Ubuntu, package name is libncurses5-dev, on
  Redhat and derivates it is ncurses-devel.
```

error やら warning やらで必要なものを入れる。

```
# apt-get install pkg-config libncurses5-dev bison libxml2-dev
```

改めて cmake する前に、バンドル版の古い groonga、mroonga がインスコされては困るので無効にする方法を探した。

Mroonga-bundled MariaDB cannot disable Mroonga

<https://github.com/mroonga/mroonga/issues/24>

と思ったら mariadb-10.0 系と 10.1 系でプラグインの外し方が変わった模様。

groonga is compiled even with -DPLUGIN\_MROONGA=NO

<https://mariadb.atlassian.net/browse/MDEV-8969?qjql=project%20%3D%20MDEV%20AND%20component%20%3D%20%22Storage%20Engine%20-%20Mroonga%22>

エラーやワーニングで中途半端に出来てしまった CMakeCache.txt を削除し、再チャレンジ。

```
# rm CMakeCache.txt
# cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql ¥
-DMYSQL_DATADIR=/usr/local/mysql/data ¥
-DMYSQL_UNIX_ADDR=/var/run/mysqld/mysqld.sock ¥
-DDEFAULT_CHARSET=utf8 ¥
-DDEFAULT_COLLATION=utf8_general_ci ¥
-DWITH_EXTRA_CHARSETS=complex ¥
-DPLUGIN_MROONGA=NO
# make -j4
# make install
```

ちなみにオプションで

```
-DBUILD_CONFIG=mysql_release
```

を付けると MySQL 系のオプションの初期設定(下記参照先)にしてくれるみたいだが、ここではやらない。

#### 2.9.4 MySQL Source-Configuration Options

<https://dev.mysql.com/doc/refman/5.5/en/source-configuration-options.html>

グループを作成。

```
# groupadd mysql
# useradd mysql -g mysql -s /sbin/nologin
```

データベースの初期設定。

```
# cd /usr/local/mysql
# scripts/mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data
```

カレントディレクトリが /usr/local/mysql のとき、basedir が . で datadir が .data となり、上記の basedir と datadir は省略しても大丈夫そう。user を省略すると datadir に作成されるファイルの所有者が、上記の例では root になる。

インストール先の権限設定。これをやっておかないと mysqld が data ディレクトリとかに書き込めないため起動できない。

```
# chown -R mysql .
```

cnf および起動スクリプトのコピー。

```
# cp support-files/my-medium.cnf /etc/my.cnf
# cp support-files/mysql.server /etc/init.d/mysqld
```

自動起動の設定と手動起動。

```
# update-rc.d mysqld defaults
# /etc/init.d/mysqld start
```

起動できないときは大体パーミッションか、data ディレクトリとかのパスがおかしいとき。もし basedir や datadir がデフォルトから変更しており、うまく認識されていないときは

```
# vim /etc/init.d/mysqld

basedir=/path/to/mysql
datadir=/path/to/mysql/data
```

みたいに手動設定もできる。

さて、mysqld に接続するための mysql クライアントだが、インストール先の /usr/local/mysql/bin だとパスが通っていないため

```
# ln -s /usr/local/mysql/bin/mysql /usr/local/bin
```

でシンボリックを作成しておく。

とりあえず root に任意パスワードを設定して接続確認。

```
# /usr/local/mysql/bin/mysqladmin -u root password 'new-password'
# mysql -u root -p

MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.1.11-MariaDB, for Linux (armv7l) using readline 5.1

Connection id:          4
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MariaDB
Server version:         10.1.11-MariaDB-log Source distribution
Protocol version:      10
Connection:             Localhost via UNIX socket
Server characterset:    utf8
Db characterset:        utf8
Client characterset:    utf8
Conn. characterset:     utf8
UNIX socket:           /var/run/mysql/mysql.sock
Uptime:                 1 hour 31 min 3 sec

MariaDB [(none)]> show engines;
```

この時点では Mroonga は無し。

いつのまにか MariaDB が起動できなくなった

MariaDB をソースからインストールした後、nginx と php5-fpm を apt-get でインストールしたら MariaDB が起動できなくなった (汗)。

php5-fpm と合わせて php5-mysql をインストールした際、mysql-common (mysql-5.5 用の /etc/mysql/my.cnf が入っている) もインストールされて

```
/etc/my.cnf      ソースから配置した mariadb-10.1 の my.cnf
/etc/mysql/my.cnf apt-get install で入ってしまった mysql-5.5 の my.cnf
```

となり起動時に上書きされ、パスは違うが設定は古いわで落ちていた模様。

ちなみにエラーログはこんな感じ。

```
[Note] Using unique option prefix 'key_buffer' is error-prone and can break in the future. Please
use the full name 'key_buffer_size' instead.
:
[ERROR] Can't find messagefile '/usr/share/mysql/errmsg.sys'
[ERROR] Aborting
```

/etc/my.cnf を /etc/mysql/my.cnf に移動して OK となる。

MySQL - 設定ファイル my.cnf 読み込み順序

<http://www.mk-mode.com/octopress/2013/06/03/mysql-config-file-read-order/>

## MeCab のインストール

日本語の文章を検索するに当たって、分かち書き？っていうんでしょうか、上手に日本語を分解、解析できたほうが検索インデックスも上手に作成できて、結果的に検索もうまくいく。その部分をしてくれるのが MeCab です。形態素解析エンジンっていうみたいですけど、よくわかりません（汗）

MeCab : Yet Another Part-of-Speech and Morphological Analyzer

<http://taku910.github.io/mecab/>

本家より本体と推奨辞書 (ipadic) を取得しておきます。

いまどきは Linux もほとんど UTF8 なので charset を指定して make。

```
# wget "https://drive.google.com/uc?export=download&id=0B4y35FiV1wh7cEnt0XlicTFaRUE" -O mecab-0.996.tar.gz
# tar zxvfp mecab-0.996.tar.gz
# cd mecab-0.996
# ./configure --with-charset=utf8 --enable-utf8-only
# make
# make install
```

続けて辞書を make すると、libmecab.so がまだシステムに認識されておらず

```
libmecab.so.2: cannot open shared object file: No such file or directory
```

となるので ldconfig をかましておくこと。

```
# wget "https://drive.google.com/uc?export=download&id=0B4y35FiV1wh7MWV1SDBCSXZMTXM" -O mecab-ipadic-2.7.0-20070801.tar.gz
# tar zxvfp mecab-ipadic-2.7.0-20070801.tar.gz
# cd mecab-ipadic-2.7.0-20070801
# ./configure --with-charset=utf8
# ldconfig
# make
# make install
```

実行ファイルは /usr/local/bin/mecab と mecab-config、辞書は

```
# mecab -D
filename: /usr/local/lib/mecab/dic/ipadic/sys.dic
version: 102
charset: utf8
type: 0
size: 392126
left size: 1316
right size: 1316
```

な感じ。

```
# echo " すもももももものうち " | mecab
```

```

すもも 名詞, 一般, *, *, *, *, すもも, スモモ, スモモ
も      助詞, 係助詞, *, *, *, *, *, も, も, も
もも    名詞, 一般, *, *, *, *, *, もも, もも, もも
もも    助詞, 係助詞, *, *, *, *, *, も, も, も
もも    名詞, 一般, *, *, *, *, *, もも, もも, もも
の      助詞, 連体化, *, *, *, *, *, の, の, の
うち    名詞, 非自立, 副詞可能, *, *, *, *, *, うち, ウチ, ウチ
EOS

```

ほかにも辞書がいくつもあるようで、下は IPADIC + な模様。  
ここでは はインストールしないことにする。

NAIST Japanese Dictionary

<https://osdn.jp/projects/naist-jdic/>

```

# tar zxvfp mecab-naist-jdic-0.6.3b-20111013.tar.gz
# cd mecab-naist-jdic-0.6.3b-20111013
# ./configure --with-charset=utf8
# make
# make install

```

## Groonga のインストール

コンパイルに必要なものを揃えておく。

```
# apt-get install zlib1g-dev liblz2-dev libmsgpack-dev libzmq-dev libevent-dev
```

mecab がパッケージからインストールされている場合、libmecab-dev も必要。

```

# wget http://packages.groonga.org/source/groonga/groonga-5.1.2.tar.gz
# tar zxvf goonga-5.1.2.tar.gz
# cd goonga-5.1.2.tar.gz
# ./configure CFLAGS="-O0 -g3" CXXFLAGS="-O0 -g3" PKG_CONFIG_PATH=/usr/lib/pkgconfig

```

configure をかましたときに最後の方で

```

Tokenizers:
MeCab:      yes
CPPFLAGS:   -I/usr/local/include
LDFLAGS:    -L/usr/local/lib
LIBS:       -lmecab
KyTea:      no

```

と出れば MeCab はアリ、KyTea はナシみたいな感じ。

```
# make -j4
```

cast で warning でまくりだが無視する (汗)

```
# make install
```

Groonga 自体が機能しているのかの確認は

Groonga をインストールして HTTP サーバを立ち上げる手順

<http://qiita.com/tamano/items/a02423587fc83df7f05a>

## groonga-normalizer-mysql のインストール

autogen.sh で autoreconf: not found とか言われるので先に入れておく。

```
# apt-get install autoconf libtool
```

configure を作る。

```
# ./autogen.sh
# ./configure CFLAGS="-O0 -g3" CXXFLAGS="-O0 -g3" PKG_CONFIG_PATH=/usr/lib/pkgconfig
```

configure の最後の方で

```
Groonga
CFLAGS:          -I/usr/local/include/groonga
Libraries:       -L/usr/local/lib -lgroonga
install directory:  ${GROONGA_PLUGINS_DIR}/normalizers
```

な感じに Groonga を認識していれば問題ないと思う。  
よければコンパイルとインストール。

```
# make
# make install
```

/usr/local/lib/groonga/plugins/normalizers に mysql.so とかが出来れば OK。

## mroonga のインストール

やり方はいままでと一緒。

```
# wget http://packages.groonga.org/source/mroonga/mroonga-5.12.tar.gz
# tar xvzf mroonga-5.12.tar.gz
# cd mroonga-5.12
# ./configure ¥
  --with-mysql-source=/usr/local/src/mariadb-10.1.11 ¥
  --with-mysql-build=/usr/local/src/mariadb-10.1.11 ¥
  --with-mysql-config=/usr/local/mysql/bin/mysql_config
# make
# make install
```

mroonga の有効化。

```
# /usr/local/mysql/bin/mysql -u root -p < /usr/local/share/mroonga/install.sql
```

有効化で何をやっているのかは install.sql の中身を読むよろし。

## mariadb の mroonga を確認する

install.sql が無事決まっていればエンジンに Mroonga が表示されるはず。

