いろいろ考えると 日本語の全文検索も MySQLがいいね！

須藤功平　　日本MySQLユーザ会

OSC2014 Tokyo/Fall
2014/10/18

目標

日本語対応の
全文検索機能を
（そこそこ仕組みをわかった上で）
実装できる

前提

- MySQLを使っている
- まあまあかそこそこのデータ量
  - ビッグデータ云々じゃない
    （そういう人はHadoopの枠に行っているよね？）
- 日本語テキストを検索したい
- でも、全文検索をよく知らない

全文検索について

全文検索とは…

とりあえず動かそう

- データベース作成
- テーブル作成
- データ投入
- 全文検索！

データベース作成

```
CREATE DATABASE full_text_search;
USE full_text_search;
```

テーブル作成

```
CREATE TABLE memos (
  content TEXT
) DEFAULT CHARSET=utf8mb4;
```

データ投入

```
INSERT INTO memos
  VALUES ("Hello world!"),
         ("Good-bye world!"),
         ("Hello MySQL!");
```

全文検索！

```
SELECT *
  FROM memos
  WHERE content LIKE "%Hello%";
-- +--------------+
-- | content      |
-- +--------------+
-- | Hello world! |
-- | Hello MySQL! |
-- +--------------+
```

全文検索！ - もっと

- AND
- OR
- 大文字小文字無視
- 日本語

全文検索！ - AND

```
SELECT *
  FROM memos
  WHERE content LIKE "%Hello%" AND
        content LIKE "%world%";
-- +--------------+
-- | content      |
-- +--------------+
-- | Hello world! |
-- +--------------+
```

全文検索！ - OR

```
SELECT *
  FROM memos
  WHERE content LIKE "%Good%" OR
        content LIKE "%MySQL%";
-- +-----------------+
-- | content         |
-- +-----------------+
-- | Good-bye world! |
-- | Hello MySQL!    |
-- +-----------------+
```

大文字小文字無視

```
SELECT *
  FROM memos
  WHERE content LIKE "%mysql%";
-- +--------------+
-- | content      |
-- +--------------+
-- | Hello MySQL! |
-- +--------------+
```

大文字小文字無視 - 理由

```
SHOW TABLE STATUS LIKE "memos"\G
-- ...
-- Collation: utf8mb4_general_ci
-- ...
```

Collation

- 照合順序（って言われてわかる？）
- 文字の比較ルール
  - 「a」と「b」はどっちが大きい？
  - 「a」と「A」は等しい？
- utf8mb4_general_ci
  - 同じようなアルファベットは同一視
    （直感的だけど雑な説明）

日本語 - データ投入

```
INSERT INTO memos
  VALUES ("こんにちは"),
         ("こんばんは");
```

日本語 - 全文検索！

```
SELECT *
  FROM memos
  WHERE content LIKE "%こんにち%";
-- +-----------------+
-- | content         |
-- +-----------------+
-- | こんにちは      |
-- +-----------------+
```

全文検索の実装方法まとめ

- DEFAULT CHARSET=utf8mb4
- INSERT
- LIKE "%キーワード%"
  - ANDもORも日本語も可
  - 大文字小文字無視も可

日本語をもっと！

- いわゆる全角アルファベット対応

全角アルファベット

```
SELECT *
  FROM memos
  WHERE content LIKE "%Ｈｅｌｌｏ%";
-- +-----------------+
-- | content         |
-- +-----------------+
-- +-----------------+
```

Collation変更

```
ALTER TABLE memos
  MODIFY COLUMN
    content
      TEXT
      COLLATE utf8mb4_unicode_ci;
```

utf8mb4_unicode_ci

- Unicode的に同じ文字を同一視
  （直感的だけど雑な説明）
  - 例：全角文字半角文字を同一視
    参考：MySQL 5.5 の unicode collation で同一視される文字
    http://tmtms.hatenablog.com/entry/20110416/mysql_unicode_collation
- utf8mb4_general_ciより遅い

全角アルファベット

```
SELECT *
  FROM memos
  WHERE content LIKE "%Ｈｅｌｌｏ%";
-- +--------------+
-- | content      |
-- +--------------+
-- | Hello world! |
-- | Hello MySQL! |
-- +--------------+
```

採用を検討

- 機能は十分？
  - やりたいことと相談
- 性能は十分？
  - データ量・リソースと相談
  - 実データが望ましい！！！
  - Webの情報は参考程度で実際に計測

性能検討例

- データ：livedoorグルメ
  https://github.com/livedoor/datasets
  - 件数：約20万口コミ
- CPU：Core i7 2.80GHz

文字数の傾向

```
SELECT
    AVG(CHAR_LENGTH(comment)) AS average,
    MIN(CHAR_LENGTH(comment)) as min,
    MAX(CHAR_LENGTH(comment)) as max
  FROM ratings_all;
-- average:  380.2013
-- min:         2
-- max:      6243
```

%ラーメン%

```
SELECT COUNT(*) AS count
  FROM ratings_all
  WHERE comment LIKE "%ラーメン%";
-- count: 31428
-- 0.898sec
```

%ラーメン%の傾向

実行時間は総件数に比例

総件数	時間（秒）
1000	0.01
5000	0.03
10000	0.05
100000	0.50
205832	0.89

AND

```
SELECT COUNT(*) AS count
  FROM ratings_all
  WHERE
    comment LIKE "%ラーメン%" AND
    comment LIKE "%焼き肉%";
-- count: 69
-- 1.01sec
```

ANDの傾向

条件数1の場合とあまり変わらない

総件数	時間（秒）	条件数1
1000	0.01	0.01
5000	0.03	0.03
10000	0.06	0.05
100000	0.57	0.50
205832	1.01	0.89

OR

```
SELECT COUNT(*) AS count
  FROM ratings_all
  WHERE
    comment LIKE "%ラーメン%" OR
    comment LIKE "%焼き肉%";
-- count: 31994
-- 1.37sec
```

ORの傾向

2倍いかないくらいには増える

総件数	時間（秒）	条件数1
1000	0.02	0.01
5000	0.05	0.03
10000	0.09	0.05
100000	0.77	0.50
205832	1.37	0.89

考察

- 自分たちのデータ量は？
  - 各レコードのテキストサイズ
  - 総件数
- どのくらい性能が必要？
  - 1リクエストのレスポンスタイム
  - 単位時間あたりの処理数

考察例

- 自分たちのデータ量は？
  - 各テキストサイズ：400文字くらい
    （データセットと同じくらい）
  - 総件数：2,3万くらい
- どのくらい性能が必要？
  - s/req: 0.5秒以内
  - queries/s: 10

実行時間

0.2秒以内には終わりそう

総件数	時間（秒）
10000	0.05
100000	0.50

スループット

- 1クエリー0.2秒
- 1秒で5クエリー
- CPUコア2つで10qpsいけそう

（実際は他の条件も加わる→もっと時間がかかるはず）

考察結果は？

- LIKEで十分？
  - 機能面と性能面を検討
  - MySQLでLIKEで日本語全文検索！
    （全文検索エンジンが必要ないなら使わなくてよい）
- LIKEだと不十分？
  - 機能面？性能面？
  - 別の選択肢を検討

別の選択肢

- MySQLベースの全文検索機能
- 全文検索サーバーと連携

全文検索機能のスキーマ

```
CREATE TABLE ratings_all_index (
  comment TEXT,
  FULLTEXT INDEX (comment)
  -- ↑を追加するだけ
) DEFAULT CHARSET=utf8mb4;
```

全文検索機能の検索方法

```
SELECT COUNT(*) AS count
  FROM ratings_all_index
  WHERE
    MATCH (comment)
    AGAINST ("+ラーメン +焼き肉"
                  IN BOOLEAN MODE);
```

全文検索機能のよいところ

- 簡単！
- 高速な検索！
  - インデックスを使った検索
    （LIKEは逐次検索）
- MySQLとよく統合されている
  - データ登録→インデックス自動更新
  - トランザクションにも対応

全文検索機能の悪いところ

- 日本語未対応
- 更新が遅い

日本語未対応

- 対策
  - アプリケーション側で前処理
    参考：MySQL Casual Talks Vol.4
    「MySQL-5.6で始める全文検索 ～InnoDB FTS編～」
    http://www.slideshare.net/y-ken/my-sql-56innodb-fts
- トレードオフ
  - インフラの管理コストと
    アプリのメンテコスト
    （簡単に使えるというメリットは減る）

更新が遅い

- ベンチマーク結果
  - 前述のスライドを参照
- 対策
  - 速いディスクを使う
  - あまり更新しない
- 検討ポイント
  - どのくらい更新があるか

全文検索機能のまとめ

- データは安全
  - トランザクションを使える
  - レプリケーションもできる
- 検索は速い・更新は遅い
- 日本語対応にはひと手間必要

全文検索機能を使う？

- そこそこのデータ量がある
- アプリ側のひと手間を許せるならアリ
- 更新が少ないならアリ

突然の質問

MySQLの特徴と
言えば？

（期待する答えがでるまで聞きます）

プラグイン機能

- 一部の機能を追加できる
  - ストレージエンジン・UDF・…
- 全文検索機能も追加できる

全文検索プラグイン

Mroonga

Mroongaのスキーマ

```
CREATE TABLE ratings_all_index (
  comment TEXT,
  FULLTEXT INDEX (comment)
  -- ↑と↓を追加するだけ
) ENGINE=Mroonga
  DEFAULT CHARSET=utf8mb4;
```

Mroongaの検索方法

```
-- MySQL標準の方法と同じ
SELECT COUNT(*) AS count
  FROM ratings_all_index
  WHERE
    MATCH (comment)
    AGAINST ("+ラーメン +焼き肉"
                 IN BOOLEAN MODE);
```

Mroongaのよいところ

- 簡単！
- 高速な検索と更新！
- 日本語対応！（開発者が日本人）
- MySQLとそれなりに統合
  - データ登録→インデックス自動更新

Mroongaの悪いところ

- トランザクション非対応
- NULL非対応
- 別途インストールが必要

トランザクション非対応

- 対策
  - レプリケーションをして、
    マスターをInnoDB、
    スレーブをMroongaにする
- 参考
  - 多くの全文検索システムは
    トランザクション非対応

NULL非対応

- 対策
  - NULLを使わない

別途インストールが必要

- 対策：パッケージを使う
- パッケージ
  - CentOS 6, 7
  - Fedora（公式）
  - Debian/Ubuntu 安定版リリース
  - Windows
  - OS X（Homebrew/MacPorts）

Mroongaのまとめ

- 日本語対応
- 検索も更新も速い
- インストール作業が必要
- 使うときは簡単

Mroongaを使う？

- そこそこのデータ量がある
- 更新が多い
- トランザクションとNULLが
  なくてもよいならアリ
- 開発者を信頼できるならアリ

ここまでの話のポイント

全文検索方法の
詳細を
知らなくても
使える

別の選択肢

- ~~MySQLベースの全文検索機能~~
- 全文検索サーバーと連携

全文検索サーバー

- Solr
- Elasticsearch
- Groonga
- Sphinx(http://sphinxsearch.com/)
- Amazon CloudSearch

違い

- 機能面
  - 全文検索初心者が使う分には
    どれも不足なし
- 性能面
  - まあまあかそこそこのデータ量
    （1台のサーバーでさばける量）
    ならどれも十分な性能

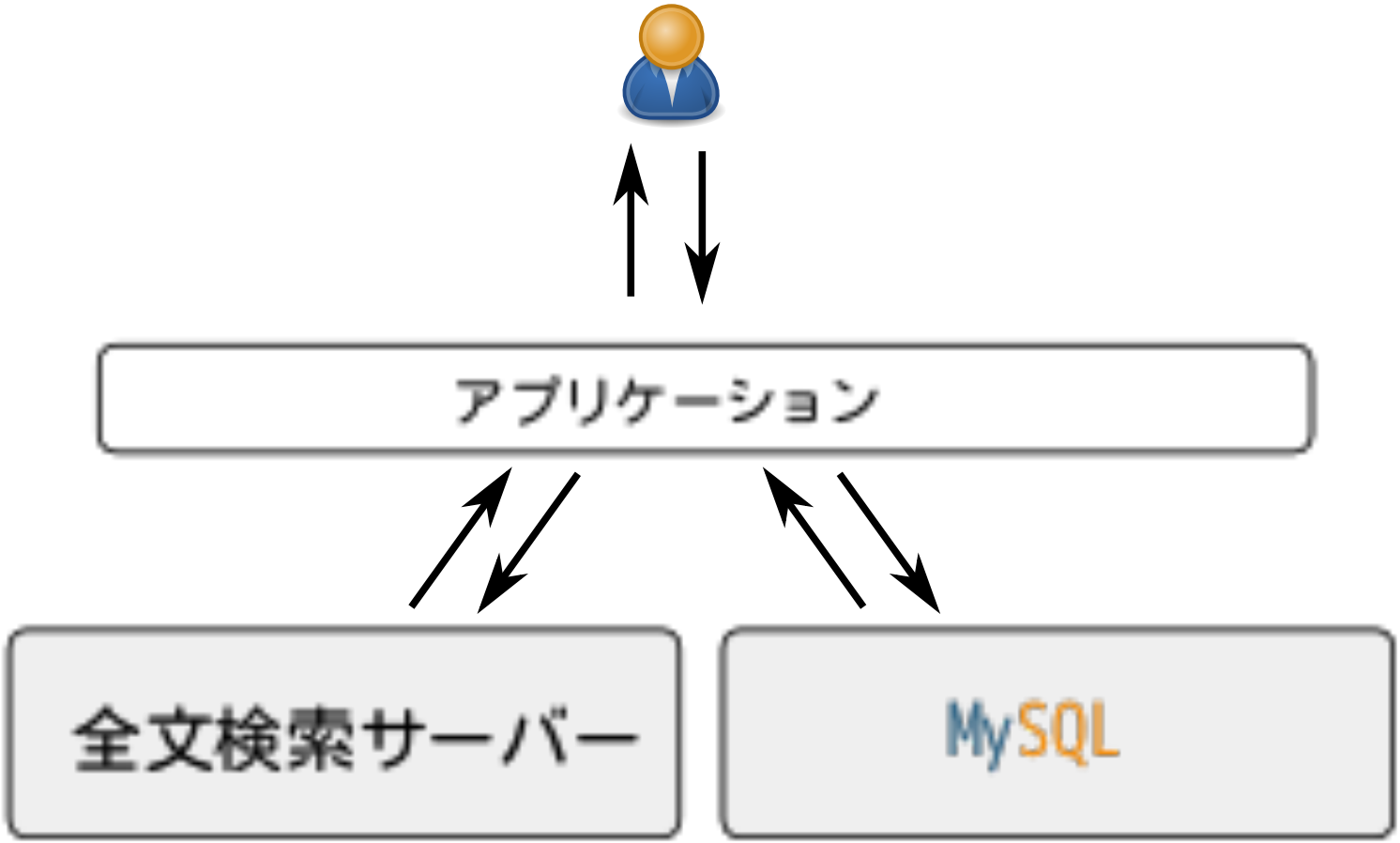
機能面：補足1

- 全文検索対応の進め方1
  - いきなりカンペキを目指さない
  - まず動かして実際に試す
  - 改良したいという箇所に気づく
  - 1つずつ改良しながら
    仕組みを学んでいく

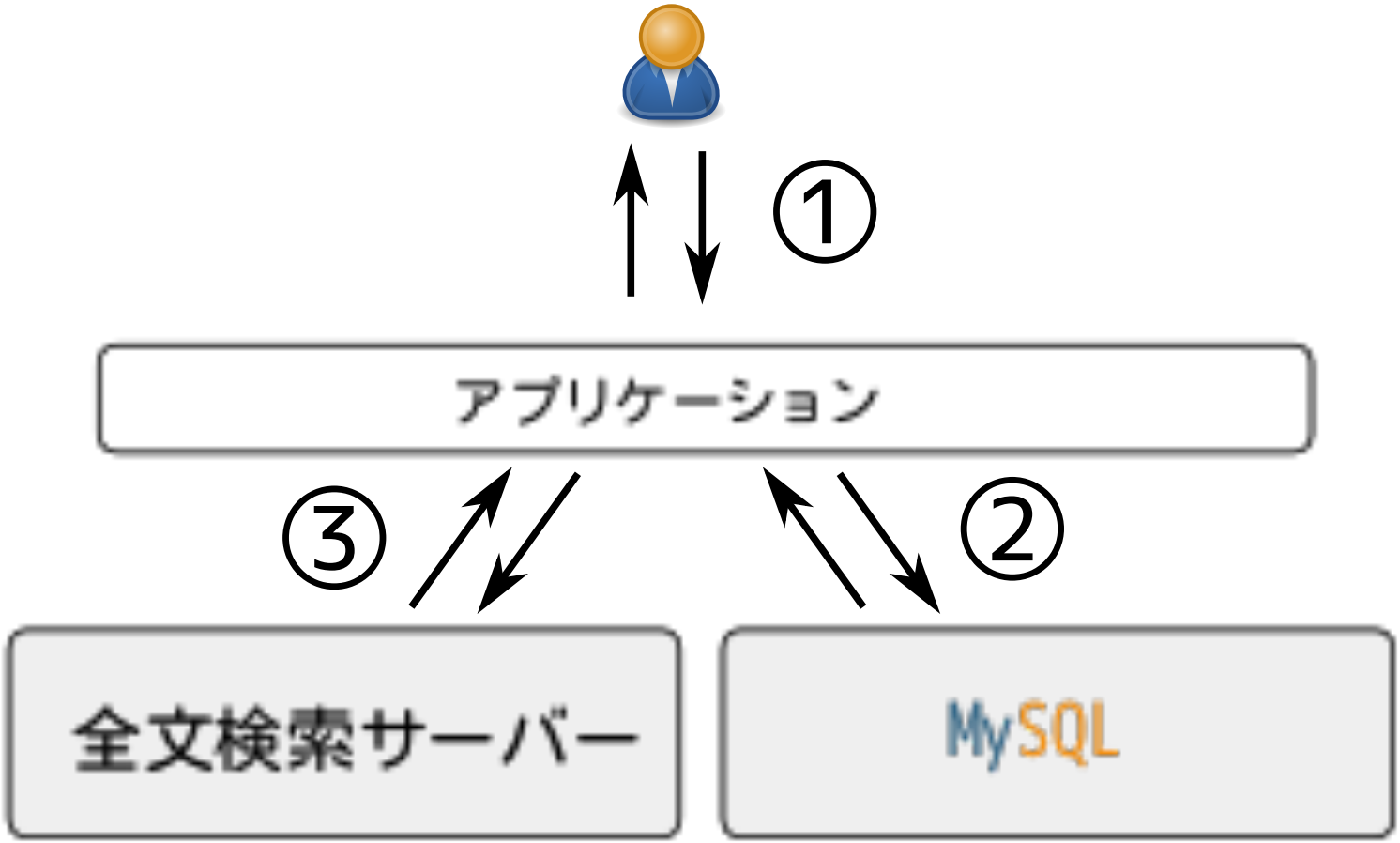
機能面：補足2

- 全文検索対応の進め方2
  - 詳しい人に相談
  - MySQLユーザ会のブースへ！

連携



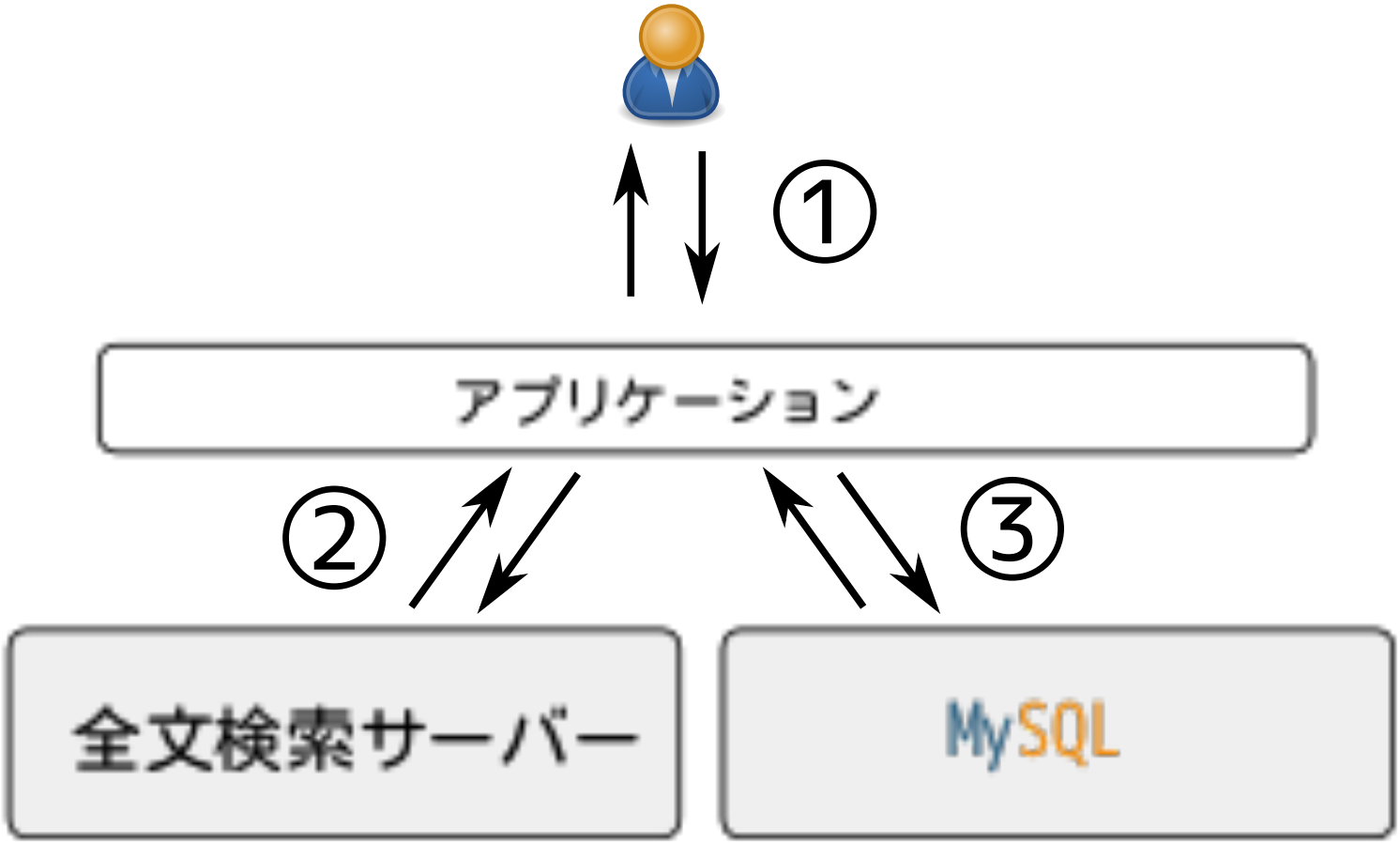
アプリの動作：更新



アプリの動作：更新

- 更新時
  - トランザクション開始
  - MySQLにデータ投入
  - 全文検索サーバーにもデータ投入
  - トランザクション終了
- ロールバックはアプリの仕事

アプリの動作：検索



アプリの動作：検索

- 検索時
  - 全文検索サーバーで検索
  - 見つかったレコードIDを条件にMySQLで検索
  - スコアとか結果をマージして表示
- JOINはアプリの仕事
  - 全文検索と他の検索を混ぜるときはどうするか考えてみよう

アプリの開発

- 実装
  - ライブラリーを使う
  - 更新・検索をサポートしてくれる
- テスト
  - 各自全文検索サーバーを用意
  - 開発環境の用意が面倒になる
    （VagrantやDockerを使うといいかも）

インフラ

- MySQLとは別に
  全文検索サーバーを管理
  - パラメーターの設定
  - 落ちた時どうする？
- システムが必要なリソース増加
  - 例：専用マシンを追加

連携したときのよいところ

- SQLが苦手なクエリーを効率よく実現できる
  - ファセット・タグ検索
- チューニングできる
  - トークナイザーを変える
  - フレーズ検索→近傍検索
  - トークンの正規化方法を変える等…

トークナイザー

- 空白区切り（英語やタグの検索に便利）
- N-gram
  - 適合率↓検索漏れ↓
  - アルファベットの文章で遅い
- 形態素解析
  - 適合率↑検索漏れ↑
  - 新語に弱い

N-gram

- 文字種が多い言語に向いている
  - 日本語
- 文字種が少ないと効率が悪い
  - 英語
- 文字種により挙動を変えて改善
  - 日本語：N-gram、英語：単語単位

形態素解析

- 区切り方が複数パターンある
  - 全パターンインデックスに登録
  - ↑は検索漏れは↓が適合率も↓かも
- 検索時に使い分ける
  - ヒットしなかったら N-gramにフォールバック

フレーズ検索→近傍検索

- 「日野でラーメン」で検索
  - ○「日野でラーメン」
  - ×「日野でみそラーメン」
- 「日野 ... ラーメン」で検索
  - ○「日野でみそラーメン」
  - ○「日野で塩ラーメン」

正規化

- 英単語のステミング
- 濁点を無視する？しない？
  - すし=ずし
  - ハハ=パパ=ババ
- 同義語はいつ展開？

連携したときのよいところ

- SQLが苦手なクエリーを効率よく実現できる
  - ファセット・タグ検索
- チューニングできる
  - トークナイザーを変える
  - フレーズ検索→近傍検索
  - トークンの正規化方法を変える等…

連携したときの悪いところ

- メンテナンスコストが増える
  - 開発面でもインフラ面でも
- 必要なリソースが増える
  - ランニングコストが増える
- ある程度全文検索の知識が必要

サーバー連携のまとめ

- 日本語対応
- 検索も更新も速い
- チューニングできる
- 導入・運用は手間が増える
- 使うときも手間が増える

全文検索サーバーを使う？

- そこそこのデータ量がある
- チューニングしたい
- 導入・運用・開発の手間増加が割に合うならアリ

参考：PostgreSQL

- 日本語未対応
  - プラグインで対応
- 完全転置インデックスではない
  - インデックスだけ使うと誤検出あり
  - ↑の後にLIKEで誤検出を除去

まとめ

- LIKEで十分ならLIKEでいい
- LIKEで不足ならMroonga
- それでも不足なら
  - マスターデータはMySQL
  - 検索対象は全文検索サーバー

つまり！

いろいろ考えると
日本語の全文検索も
MySQLがいいね！

MySQLが役に立つね

お知らせ1

- MySQLユーザ会ブースあり
  - 隣はOracleのMySQLの人たち
- 17:15-別のMySQL枠あり
  - OracleのMySQLの人の話

お知らせ2

- MariaDBにMroongaバンドル！
  - 10.0.15から組み込み！
  - 別途インストールしなくてよい！

お知らせ3

- 検索エンジンについて知りたい
  - 「検索エンジン自作入門」
- いい肉の日に渋谷で Groongaイベント！
  - 「いい肉 Groonga」で検索
  - 自作本のサイン会をやるよ！

お知らせ4

- Groongaをもっと知りたい
  - Groongaドキュメント読書会
  - 詳細は↑で検索
  - Mroongaが使っている
    全文検索エンジンの理解を深める会
  - 1,2ヶ月に1回開催