



更に memcached を追加することでより速度を向上させた [2]. memcached とは分散メモリキャッシュシステムを構築することが出来るソフトウェアで、データベースから読み出した結果などをメモリに保存し、次に同じデータが参照されたときにメモリから即座に返すことができる。仕組みを図 2 に示す。

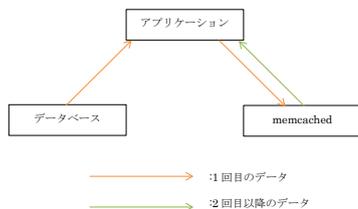


図 2 memcached を用いた検索速度改善の仕組み

検索ワードをキーとし、検索結果を配列に格納してキーと配列を memcached のメモリへ保存する。以降その検索ワードで検索した場合、検索結果をメモリから持ってくることで大幅に検索時間が短縮され、0.01 秒単位となった。

### 3.3 マイ本棚機能

検索した書籍はお気に入り登録することができ、登録された書籍は本棚に表示される。本棚は図 3 のようになっている。



図 3 My 本棚機能の画面

本棚内では登録した書籍の表紙画像が表示され、本棚内での検索やフォルダへの移動を行うことができる。表紙画像をクリックすることで本の詳細を見ることができる。また、詳細画面からコメントの入力や登録した書籍をより細分化して分けるフォルダへの保存ができる。

## 4 openBD 検索 API の実装

以上のように、MySQL に格納した openBD の書誌データを利用することで書籍検索・管理アプリケー

ションを作成することができた。しかし、openBD API の利用を拡大するためには、MySQL ユーザではない第三者にも検索機能をもっと利用しやすく提供する必要がある。そのため、検索機能を API として提供し、検索結果は JSON 形式で返すように実装を試みた。図 4 にその形式の参考案を示す。

```

var openbd_search_obj = {
  search_word: "検索語", // [文字列] 検索単語
  search_index: "検索対象インデックス", // [文字列] 検索対象インデックス
  search_date: "西暦-月-日 時刻", // [ISO形式] 検索日時
  num_detected_books: ヒット数, // [整数] ヒット数
  detected_books: [
    {
      isbn: ISBN番号, // [文字列] ISBN番号
      detected_value: "ヒットした値" // [文字列] 該当値
    }
  ]
}; // openbd_search_obj.detected_books[0], ...
  
```

図 4 openBD Search API: JSON 形式案

現状、API キーを referer のドメインと組み合わせる仕組みを導入し、API の利用を安全に行えることを示すための実装を行っているところである。

## 5 今後の課題

本研究で作成した書籍検索・登録アプリケーションについては、全体的にソースコードが荒くなってしまったり、memcached に関しても 2 回目以降でなければ機能しない等、全体的に実力不足を痛感する欠点が残ってしまった。メモリへ事前に結果を保存することで始めから memcached を機能させることが出来るという話を聞いたので、それを実践して検索をより便利にしていきたい。

また、openBD 検索 API については、テストを繰り返してそれなりに使えるレベルに達した時点で広く外部公開していきたいと考えている。

## 参考文献

[1] openBD, <https://openbd.jp/>  
 [2] MySQL 5.6 リファレンスマニュアル, 16.6.2 memcached の使用, <https://dev.mysql.com/doc/refman/5.6/ja/ha-memcached-using.html>