Elastic Search vs Solr

403410033 資工四 曾俊宏

系統環境

- VMWare Workstation
  - Ubuntu 16.04 LTS
  - Intel® Core™ i5-7500 CPU @ 3.40GHz × 4 (4 cores for VM)
  - 8GB Ram for VM
  - 100 GB SSD for VM

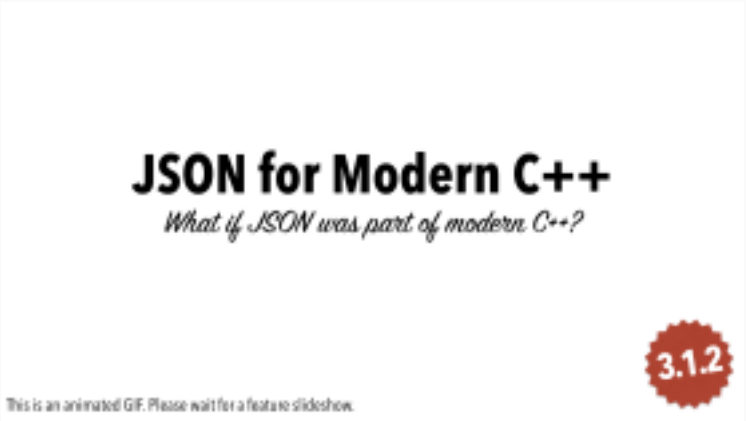
測試資料 ettoday 11GB file

- 有一些utf8解析會失敗的 record
  - 在 轉json 或是 做斷辭出現exception ->ignore
- 實際有成功使用到的record數: 4812350
- 前處理
  - 只取用 url, title, keyword, image link, body
  - 先對 body 斷好詞， 每 十萬個 record 切成一個 json 檔案

```
const char *recordHeading[HEADINGCOUNT] = {
    "Gais_REC",  "url",       "MainTextMD5", "UntagMD5", "SiteCode",
    "UrlCode",   "title",     "Size",        "keyword",  "image_links",
    "Fetchtime", "post_time", "Ref",         "BodyMD5",  "Lang",
    "IP",        "body",      "botVer",      "Time"
};
```

工具

- CppJieba
- C++ 5.4
- JsonCpp
  - 只要宣告 json type 的變數之後，就當作 c++ map 來放資料，最後再用 dump() 即可得到 Json format std::string!
- OpenCC
  - 把 cppJieba 的字典轉成繁體
- Elastic Search 6.2.2
- Solr 7.3
- Python 3.6
- Java 8



cppJieba 斷詞

- 珍惜生命，有python可以用的話真的很美好
- 使用 cppJieba 提供的辭典
- 真的很快 `./jieba  1518.52s user 105.91s system 99% cpu 27:15.16 total`
  - 針對body做斷詞
  - 單線程 27分15秒 vs python 的 5+ 小時
  - 每 10000 record 約 2.5 秒
- Cut with HMM
  - 比較精確 (w/ vs w/o)
    - 字典沒有的政府部門詞彙: 衛福部 疾管署 vs 衛 福 部 疾 管 署
    - 數字: 共達 2978 人 vs 共 達 2 9 7 8 人

cppJieba使用心得

- 坑:
  - fatal error: 'limonp/Logging.hpp' file not found
  - 解: 複製 deps/limonp 資料夾 到 include/cppjieba底下即可
- 內建有Dict, Hmm, IDF, StopWord 這幾個資料(簡體)
  - 用OpenCC翻譯一下就可以用囉!
- 在cppJieba資料夾下建立src資料夾放code，include即可使用

```
#ifndef SEGMENTATION_H
#define SEGMENTATION_H

#include "../include/cppjieba/Jieba.hpp"

#include <iostream>
#include <string>
#include <vector>

using namespace std;

const char *const DICT_PATH = "../dict/jieba.dict.fan.utf8";
const char *const HMM_PATH = "../dict/hmm_model.fan.utf8";
const char *const USER_DICT_PATH = "../dict/user.dict.fan.utf8";
const char *const IDF_PATH = "../dict/idf.fan.utf8";
const char *const STOP_WORD_PATH = "../dict/stop_words.fan.utf8";

class Segmentation
{
private:
    cppjieba::Jieba jieba;

public:
    Segmentation()
        : jieba(DICT_PATH, HMM_PATH, USER_DICT_PATH, IDF_PATH, STOP_WORD_PATH)
    {
        cerr << "Init Segmentation" << endl;
    }

    void printSegmentationResult(vector<string> &res);
    void performSegmentation(string &s, vector<string> &res);
    string getSegmentationString(vector<string> &res);
};

#endif
```

- ***demo.cpp*** 裡有他的各式用法
  - 基本上從這裡研究起就對啦!
- 基本上就是用 string 的資料型態送入 query, 送 vector<string> 進去接答案即可
- 最後那個 true 是要使用 hmm 的意思
- 結論:
  - pro: 效能好很多
  - con: 使用說明資料不如python Jieba多

```
void Segmentation::performSegmentation(string &s, vector<string> &res)
{
    res.clear();

    // cerr << "Cut With HMM" << endl;
    // cerr << s << endl;
    // cerr << "----------------------------------------" << endl;
    jieba.Cut(s, res, true);
}
```

Solr / ES Insertion/Search

- Insertion
  - ES: python 套件 elastic-search -> 30分
  - Solr: POST -> 1小時10分
- Search on 1000 queries (隨機從 keyword 欄位抓取)
  - ES: 15 秒
  - Solr: 23 秒