Intelligent Devices and Cloud Computing
Final Project Report

鄭理文 王以彥

1. 題目 :

Real-time Parking Lots Assistant

2. 成員 :

r05942103 王以彥 : 負責資料蒐集、處理、架設 spark cluster
r05942080 鄭理文 : 負責後端管理、web UI、app 設計及呈現

3. 摘要 :

我們的 project 構想，是想幫助在路上行車時，常常會遇到想停車但不知道附近哪裡有停車場，或是開到停車場卻發現大排長龍沒有停車位，因此設計了 web 和 app 介面，結合停車場資訊以及 Google Map，在地圖上呈現停車場位置和資訊，讓使用者能即時知道附近停車場狀況。

4. 作法 :

4-1 資料來源 :

資料來源採用政府資料公開平台-新北市路外公共停車場資訊、新北市公有路外停車場即時賸餘車位數兩份公開資料，網址見附錄，原本預計採用新北市及台北市停車場資訊，但因網路上公開資訊中，台北市停車場並沒有隨時更新的開放資料，故最後只採用新北市停車場資訊(每 3~5 分鐘平台上的停車場資訊會做一次更新)。

4-2 系統架構 :

一、使用 python 每分鐘為頻率至政府公開平台上抓取最新資料。
二、於 google cloud platform 上，使用 Google Dataproc 建立一個 spark cluster，包含 4 CPU master 以及 2 個各 2 CPU worker。
三、資料經過 spark 處理後，存入 MongoDB，並用 node.js、Express 管理後端部分，本次 project MongoDB 架設在實驗室 server 上。
四、結合 Google Map API，使用 HTML、CSS、JavaScript 建立 web ui，使用 Swift 建立 IOS App。

4-3 實作細節 - 存取靜態資訊 :

首先先處理的是停車場資訊靜態資料，也就是只須在一開始更新一次即可，其中包含停車場 ID、名稱、地區、地址、費率、經緯度、車位

總數、營業時間等固定資訊，實作方法為附檔中的 save_static_info.py，利用 python 的 request，向資料網站請求資料，因原始資料中的座標系統為 TWD97 資料，故需轉換成常用的經緯度，接著使用 pymongo package，將所有資料直接存入 MongoDBserver。

4-4 實作細節 - spark 即時資料處理 ：

實作方法為練習使用 pyspark 對資料進行處理，再藉由 mongo-spark connector 將資料存入 MongoDB，如附檔 spark_final.py。

一、首先建立 spark session，並至公開平台抓取即時剩餘車位資料

```
spark = SparkSession.builder.appName("FinalProject").getOrCreate()

link = "http://data.ntpc.gov.tw/api/v1/rest/datastore/382000000A-000292-002"
req = requests.get(link)
data = json.loads(req.text)
```

二、從 MongoDB 中讀取靜態停車場資訊，儲存至 dataframe

```
print("read from mongodb")
df_parking = spark.read.format("com.mongodb.spark.sql.DefaultSource").load()
```

三、處理即時車位資料，資料中包含停車場 ID 及剩餘車位，有部分停車場並沒有提供即時剩餘車位，在資料中是顯示-9，轉換為”No Data”，並重新命名 column name。

```
print("build new dataframe")
df_new = spark.createDataFrame(data["result"]["records"])
df_new = df_new.withColumn('AVAILABLECAR', regexp_replace('AVAILABLECAR', '-9', 'No Data'))
df_new = df_new.select(col("ID").alias("park_id"),col("AVAILABLECAR").alias("remain_car"))
```

四、將靜態資料和動態資料結合，最後經由 mongo-spark connector 存入 MongoDB 中。

```
df_final = df_parking.join(df_new, df_parking["park_id"] == df_new["park_id"])

print("write to db")
df_final.write.format("com.mongodb.spark.sql").mode("overwrite").save()
```

4-5 實作細節 – 即時更新 ：

利用 linux 系統中的 crontab 工作排程，設定每分鐘執行一次 script(附檔中的 spark_script.sh)，在 script 中設定 MongoDB 的相關資訊，並設定 mongo-spark package 版本。

```
/usr/lib/spark/bin/spark-submit --master "local[4]"  \
                        --conf "spark.mongodb.input.uri=mongodb://140.112.41.157:27018/cloud-final.parkingInfo?readPreference=primaryPreferred" \
                        --conf "spark.mongodb.input.partitioner=MongoShardedPartitioner" \
                        --conf "spark.mongodb.output.uri=mongodb://140.112.41.157:27018/cloud-final.parkinglots" \
                        --packages org.mongodb.spark:mongo-spark-connector_2.11:2.0.0 \
                        spark_final.py
```

4-6 實作細節 – 後端 :

使用 mongoose 讀取 MongoDB 內的資料

```
mongoose.connect('mongodb://140.112.41.157:27018/final_data');
mongoose.connection.once('open', function(cb) {
  return console.log('Database inited');
});
```

在 server 內自行寫 API 使 Web 和 APP 取得資料

```
router.route('/googlemap/:parkinglot_id')
  .get(function(req, res) {
    parkinglot.findOne({
      park_name: req.params.parkinglot_id
    }).exec(function(err, result) {
      res.send(result);
    })
  })
```

透過 Express 呈現 Frontend 的網頁

```
app.use(express.static('../front_end'));
var googlemap = require('./routes/api_googlemap');
app.use('/googlemap', googlemap);
```

4-7 實作細節 – 前端 web :

利用 JQuery 取得 Backend 寫好的 API

```
$.ajax("http://140.112.41.157:3001/googlemap/googlemap/", {
  type: 'GET',
  success: function(results) {
    alert("總資料數： "+results.length + " 筆"+ " \n" +"資料來源： 新北市政府");
```

從 Google API 申請而得到的 token

```
<script type="text/javascript" src="jquery.js"></script>
<script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCO4ZYLVfg2r34jXwNR-u5ce7eO2P3Gjmg&call
```

初始化設定 GoogleMap 執行環境

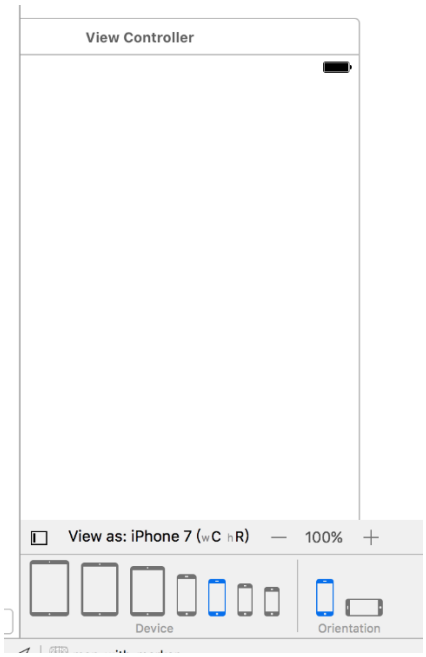
```
function initMap() {

  map = new google.maps.Map(document.getElementById('map'), {
    zoom: 14,
    center: new google.maps.LatLng(25.019303, 121.542425),
    mapTypeId: 'Google'
  });
}
```

程式成功執行

```
osboxes@osboxes:~/Desktop/googlemap_project/backend$ node server.js
=========================Server is starting=========================
listening to 3001 port
Database inited
Mongoose: mpromise (mongoose's default promise library) is deprecated, plug in y
our own promise library instead: http://mongoosejs.com/docs/promises.html
```

4-8 實作細節 – 前端 IOS App :

制定此 APP 運用的介面



從 Google API 申請而得到的 token

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions
    [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
  GMSServices.provideAPIKey("AIzaSyCO4ZYLVfg2r34jXwNR-u5ce7eO2P3Gjmg")
  return true
}
```

初始化設定 GoogleMap 執行環境

```
let camera = GMSCameraPosition.camera(withLatitude: 25.019303, longitude: 121.542425, zoom: 14.0)
let mapView = GMSMapView.map(withFrame: CGRect.zero, camera: camera)
self.view = mapView
```

從 Backend 的 API 取得資料並確認為 JSON 格式

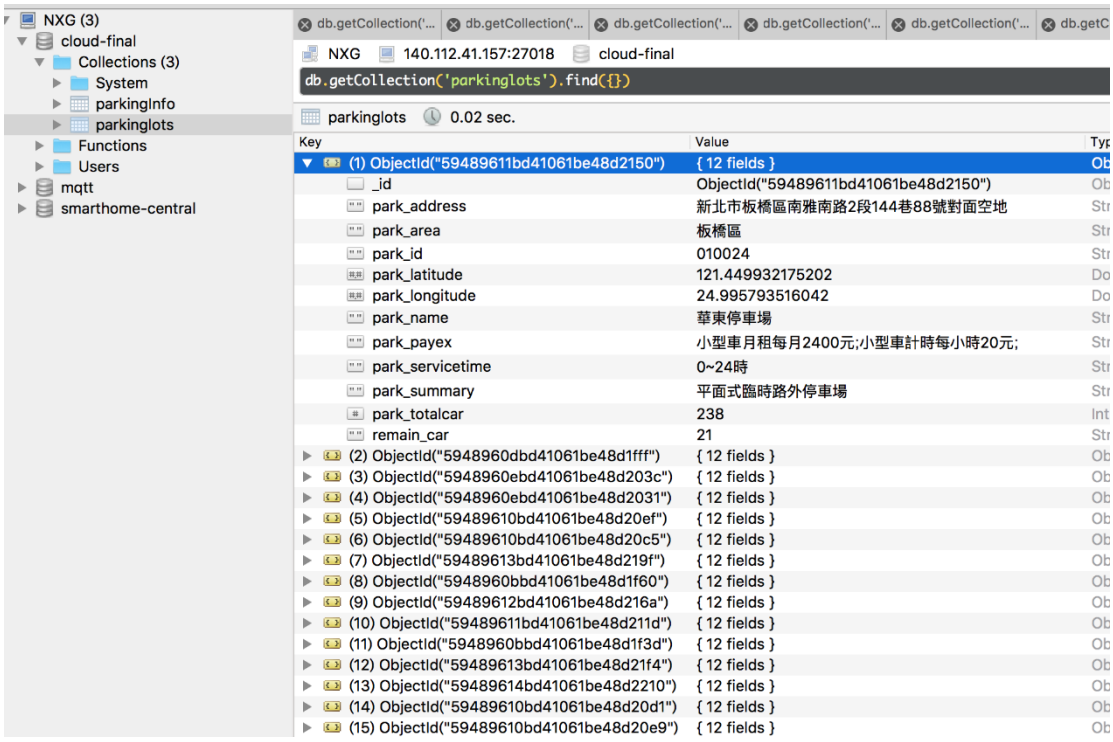
```
    URLSession.shared.dataTask(with: (url as URL?)!, completionHandler: {(data, response, error) ->
        Void in
        if let jsonObj = try? JSONSerialization.jsonObject(with: data!, options: .allowFragments) as?
            NSArray {
```

標記停車場位於的點

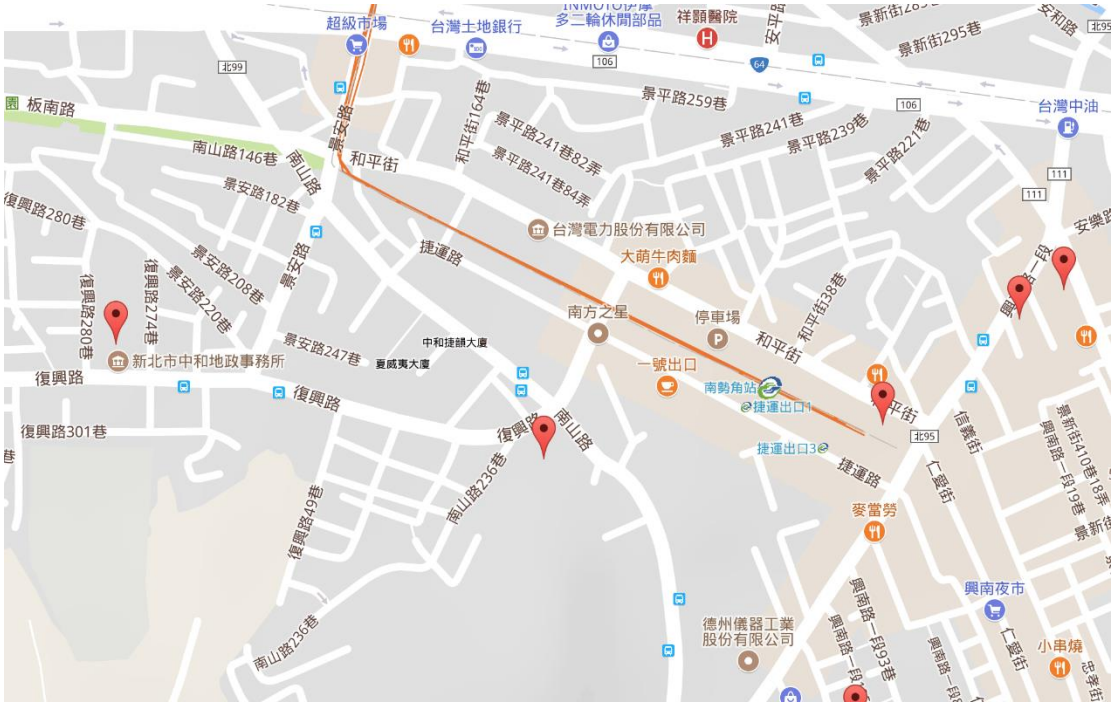
```
let park_marker = GMSMarker()
park_marker.position = CLLocationCoordinate2D(latitude: park.long, longitude:
    park.lati)
  print(park.lati)
park_marker.title = park .name
park_marker.snippet = " Remaining:  \(park.rem) \n Pay: \(park.pay) \n Business
    Hours: \(park.time)"
park_marker.map = mapView
}
```

5. 成果

於 MongoDB database 中確認所有資料正確無誤



web ui 成果如下圖，於 google map 上可看到我們標出的各個停車場標點



點擊標記可查看即時剩餘車位等即時資訊

app 部份各個停車場地圖資訊如下圖

點擊標記可查看剩餘車位等即時資訊

6. 討論

在這次的 final project 中，主要結合了之前作業中的項目，運用到了 spark 做資料處理，用 MongoDB 做資料庫，並用 HTML、CSS、JavaScript 呈現網頁，以及 Swift 建立 IOS App，主要遇到比較多問題是在如何運用 Google Map API，尤其是 app 部分，因之前並沒有接觸過 swift，在使用上遇到蠻多語法上的問題。

總結起來我們的 project 還有些許地方可以做改進 ：

- 目前我們是設定 google map 起始座標畫面在台大博理館，可以改為使用使用者目前位置來做起始座標。
- 目前畫面上顯示的是所有停車場，可以嘗試只顯示使用者周遭幾公里內的停車場，或許可以讓使用者在顯示全部和顯示周遭之間切換。
- 可以考慮讓使用者對停車場做一些搜尋篩選的動作，例如搜尋目前還有車位或車位數大於一定值的停車場。
- 雖然台北市停車場並沒有動態即時資料，但有靜態停車場資訊，應該可以也將這些資訊放上 google map，使用者雖然無法確認這些停車場的即時剩餘車位，但還是能知道台北市周遭哪邊會有停車場，以及其相關資訊。

7. 附錄及相關參考資料

- 新北市公有路外停車場即時賸餘車位數 ：http://data.gov.tw/node/26701
- 新北市路外公共停車場資訊 ：http://data.gov.tw/node/26653
- mongo-spark connector : https://docs.mongodb.com/spark-connector/master/python-api/#pyspark-shell
- run pyspark script with crontab : https://stackoverflow.com/questions/39604706/running-pyspark-using-cronjob-crontab
- Setting up Google Cloud Dataproc with Python 3 : https://blog.sourced.tech/post/dataproc_jupyter/
- https://console.developers.google.com/apis/dashboard?project=parkingproject-1496993400667&hl=zh-tw&duration=PT1H
- Swift JSON Tutorial https://www.simplifiedios.net/swift-json-tutorial/

- http://www.learnswiftonline.com/mini-tutorials/how-to-download-and-read-json/
- https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/ControlFlow.html
- https://developers.google.com/maps/documentation/ios-sdk