2020_OS_Fall_HW2: ETL process

繳交期限：10/05-10/26 13:00

I. 作業目標

- 請撰寫一支以多執行緒（Multi-Thread）開發的轉換程式，其功能是將CSV檔案轉換並輸出成JSON檔案。
- 請觀察及分析程式執行期間，包括但不限於CPU、Memory、Disk I/O的使用情況，探討作業系統是如何服務我們的程式。

II. 測試資料

> 請撰寫一支程式，自行產生出符合以下條件的測試資料，做為此次作業的輸入資料。

- 請在**單一檔案**產生N筆資料，每筆資料包含**20個**亂數值，並滿足以下條件：
  - 亂數範圍: -2147483648 ~ 2147483647 即 -2^{31} ~ 2^{31}-1 (4 bytes int)
  - 每個亂數值以 | 字符分隔，每筆資料以 換行字符（LF） 分隔。
- 測試資料的檔案大小至少為1GB，意即你所撰寫的程式至少要能夠處理到此大小的測資。
- 請以 UTF-8 的字元編碼儲存，並將此測資檔案命名為 input.csv 。

你所產生的測試資料（input.csv），應該會長得像這樣：

```
1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20
2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21
3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22
4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22|23
...(略)
```

III. CSV to JSON轉換程式

請撰寫一支能將CSV格式轉換成JSON格式的轉換程式，且必須滿足以下基本要求：

1. 必須能正確地將CSV檔案內的資料轉換為JSON格式並輸出。

    - 輸入資料的檔名需命名為 `input.csv` 。
    - 轉換後的輸入資料需命名為 `output.json` 。
2. **必須以多執行緒（Multi-Thread）開發，並在執行指令中包含 `threads` 參數，可供動態調整程式所用的執行緒（thread）數量。**
3. 必須能夠處理符合前項所定義之測試資料條件的輸入資料。
    - 輸入資料檔案（input.csv）中包含數筆亂數資料。
    - 每筆資料需含有**20個亂數值**，並以 | 字符分隔。
4. 必須能夠處理檔案大小為1GB的輸入資料（input.csv）。
5. 輸入資料中的每筆數值資料，轉換成JSON時，請以 `col_{INDEX}` 做為KeyName。
6. 當CSV檔案轉換成JSON格式後，必須與原本CSV資料的排序相同，亦即CSV的第一行為JSON的第一個Object以此類推。
7. 請勿在轉換程式中，**強制包含**產生測試資料的流程，因為助教會使用自己的輸入資料（input.csv）。

（選擇性）你也可以使用GPU開發CUDA程式來完成本項作業。

範例輸入（input.csv）：

```
1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20
2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21
3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22
4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22|23
```

範例輸出（output.json）：

```
[
   {
      "col_1":1,
      "col_2":2,
      "col_3":3,
      "col_4":4,
      "col_5":5,
      "col_6":6,
      "col_7":7,
      "col_8":8,
      "col_9":9,
      "col_10":10,
      "col_11":11,
      "col_12":12,
      "col_13":13,
      "col_14":14,
      "col_15":15,
      "col_16":16,
      "col_17":17,
      "col_18":18,
      "col_19":19,
      "col_20":20
   },
   {
      "col_1":2,
      "col_2":3,
      "col_3":4,
      "col_4":5,
      "col_5":6,
      "col_6":7,
      "col_7":8,
      "col_8":9,
      "col_9":10,
      "col_10":11,
      "col_11":12,
      "col_12":13,
      "col_13":14,
      "col_14":15,
      "col_15":16,
      "col_16":17,
      "col_17":18,
      "col_18":19,
      "col_19":20,
      "col_20":21
   },
   {
      "col_1":3,
      "col_2":4,
      "col_3":5,
      "col_4":6,
      "col_5":7,
```

```
        "col_6":8,
        "col_7":9,
        "col_8":10,
        "col_9":11,
        "col_10":12,
        "col_11":13,
        "col_12":14,
        "col_13":15,
        "col_14":16,
        "col_15":17,
        "col_16":18,
        "col_17":19,
        "col_18":20,
        "col_19":21,
        "col_20":22
    },
    {
        "col_1":4,
        "col_2":5,
        "col_3":6,
        "col_4":7,
        "col_5":8,
        "col_6":9,
        "col_7":10,
        "col_8":11,
        "col_9":12,
        "col_10":13,
        "col_11":14,
        "col_12":15,
        "col_13":16,
        "col_14":17,
        "col_15":18,
        "col_16":19,
        "col_17":20,
        "col_18":21,
        "col_19":22,
        "col_20":23
    }
]
```

IV. 如何開始

1. 請依照 **II.** 的說明，產生出撰寫此作業需要使用的輸入資料 `input.csv` 。
2. 請撰寫一支「CSV to JSON」的轉換程式，須符合 **III.** 所述之基本要求，並將執行後的結果儲存為 `output.json` 。

3. 請使用任意工具或方法分析、觀察你所撰寫的程式，並優化你的程式（例如：降低執行時間）。
    - 試著用盡電腦的計算能力，讓它集中在處理你的程式上，以電腦閒置資源最小化為目標。
4. 將你所觀察到的現象，試著思考作業系統背後的行為，撰寫出一份完整的效能分析報告。
    - 此部分的內容將會是作業評分的重點，盡你所能說明的越詳盡越好。
    - 請嘗試往觀察CPU、Memory、Disk I/O的使用情形著手，對上述系統資源的觀察，並試著去結論作業系統應該要如何服務該些程式，使每隻程式能得到最好的服務。
5. 將你撰寫的程式碼及說明文件，依照作業繳交的規定，於期限內上傳到Moodle平台。

V. 說明文件

> 說明文件的格式不拘，你也可以使用Markdown來完成。
> 你在撰寫此份說明文件時，必須要包含下列基本內容。

學號：
姓名：
系級：

開發環境：

- OS: Ubuntu 20.04.1
- CPU: Intel® Core™ i7-10700 CPU @ 2.90GHz × 16
- Memory: 32GB
- Programming Language(version): Java 1.8.0_261
    - 必須包含版本資訊

程式執行時間：

- 請在你的程式中加入量測執行時間的程式碼，以精準的獲取此數值。
- 請測量不同執行緒（thread）數量下程式的執行時間。

程式開發與使用說明：

- 你是如何開發這支程式，程式在處理資料的流程及邏輯為何？
- 你的程式該如何使用，請詳細說明執行的步驟。
- 輸入資料（input.csv）會與程式執行檔在相同的目錄下，因此不必有設置輸入資料路徑的參數。

```
# （重要）請確保助教能夠按照此步驟執行你的程式。
# 在程式執行的指令中，請提供可設置執行緒（thread）數量的參數。

# Java Example
# Compile
$ javac ./YourSourceCode.java
# Run
$ java ./YourSourceCode [threads]
```

效能分析報告：

> 分析報告的內容建議包含以下內容，但你也可以自由發揮。
> 此部分的內容將會是作業評分的重點，盡你所能說明的越詳盡越好。

- **在運行你所開發的「CSV to JSON」轉換程式下：**
  - 請觀察程式執行期間各個Stage對電腦資源使用情形。
    - Stage最少會包含讀檔、資料處理、輸出三個階段。
  - 請觀察並比較不同執行緒（thread）數量下，程式的執行狀況、系統資源的使用。
  - 請觀察系統效能以及OS是如何服務我們的程式，並結論OS的設計要提供哪些優化服務。
- 你可以搭配圖片、圖表或外部資料來說明。

VI. 作業繳交

- 繳交期限：10/05-10/26 13:00
  - 逾期繳交將按下規則**採連續扣分**。
    - 逾期一日：得分扣10分。
    - 逾期二日：再扣20分。
    - 逾期三日：再扣30分。
    - 逾期四日以上：得分以0分計算。
- 請將你的「程式原始碼」、「說明文件」打包成ZIP壓縮檔（請命名為 `HW2_你的學號.zip` ）。
  - 說明文件 -> 請繳交Markdown（.md），或是PDF檔案。
    - 若你使用HackMD寫文件，可以接受以匯出的HTML檔案繳交。
    - 若你用其他形式撰寫文件，請轉換成PDF格式繳交。
  - 不需要繳交測試所使用的輸入檔（input.csv）及輸出檔（output.json）。
- **請再次確認你的程式能夠正常執行，且說明文件中有包含指定的內容。**
- 請將打包好的壓縮檔，上傳到Moodle的作業中，即可完成此次作業的繳交。

VII. 評分項目

> 作業會以下列原則評分，滿分為100分。

- **CSV to JSON轉換程式**
  - **是否滿足 III. 所規定之基本要求，若無法完全達成則本次作業以0分計算。**
  - 轉換結果和資料順序的正確性。
  - 此程式的執行效率（速度）。
- **說明文件：效能分析報告的內容是本項評分的重點。**
  - 文件是否有包含指定的內容。
  - 報告內容的完整度及正確性。
  - 對系統資源觀察的程度以及呈現的說明內容。
  - 使用不同執行緒數執行程式，對此程式觀察並比較。

VIII. 注意事項

- 你可以使用任意程式語言撰寫作業，但助教只會用Ubuntu環境執行你的程式，因此建議使用Linux OS來撰寫此份作業。
- 你可以用任何OS（Windows、Windows Subsystem for Linux、Virtual Machine等）開發，但如同前項所述，助教只會在Ubuntu中執行你的程式。
- 你可以在虛擬機器（Virtual Machine）上撰寫程式及分析效能，但請在報告中註記你是使用虛擬機器，並提供關於該虛擬機器的基本資訊（像是：OS、vCore、Memory size等）。
- **助教在評分時只會使用1GB大小的測試資料作為輸入資料（input.csv）。**
- **嚴格禁止互相抄襲程式碼，助教會進行程式碼比對，違者此次作業以零分計算。**

IX. 參考資料

- Pthread (https://computing.llnl.gov/tutorials/pthreads/)