Data Mining 2018

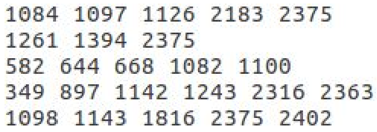
Homework 2 – Frequent Itemset Mining on GPGPU

Due: 23:59, Nov 3, 2018

In this assignment, you are required to implement a basic vertical frequent pattern mining program using GPGPU.

The input dataset and output format are in the following:

- Input:



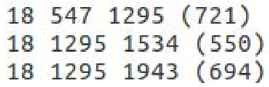
```
1084 1097 1126 2183 2375
1261 1394 2375
582 644 668 1082 1100
349 897 1142 1243 2316 2363
1098 1143 1816 2375 2402
```

Each line in the dataset represents one transaction. The numbers separated by space in each line are the items in the transaction.

- Output:

The output format is <itemset, support>. The items in each itemset is in ascending order.

For example,



```
18 547 1295 (721)
18 1295 1534 (550)
18 1295 1943 (694)
```

Part I. Setting up CUDA environment (25%)

- Download and install NVIDIA CUDA environment on your machine: http://developer.nvidia.com
- Read the CUDA document for setting your library and include path
- Build the CUDA Sample Code. Your machine should run the example code at
  <example code path>/NVIDIA_CUDA-<version>_Samples/1_Utilities/deviceQuery
- Screenshot your device query results, and then paste the results in your report. (25%)

Part II. Frequent Itemset Mining with GPGPU (75%)

- Implement a GPU version of Eclat algorithm using vertical bit vectors.
- You can implement in python (version >= 3.5) or C++.
- Correct implementation and result. You should add at least one kernel function to run on GPU. **(30%)**
- Your GPU version performance. It should be faster than CPU version. **(15%)**

- Report **(30%)**
  - Briefly describe how you implement the algorithms and your parallel design. The better the design, the higher the score. (15%)
  - Plot the execution time of your GPU mining (y-axis: execution time) (15%)
    - Different minimum support (0.1%, 0.08%, 0.06%, 0.04%, 0.02%)
    - Different block number (16, 32, 64, 128, 256)
    - Different thread number (16, 32, 64, 128, 256)

    Briefly explain the reason of the different execution time.
- A sample C++ code of CPU version Eclat is provided. You can complete this assignment by using **the sample code** or **your own code from HW1**. Specify which code you use in report.
  - Using the sample code
    - Fill the ***mineGPU*** function
    - You can add any other functions you need
    - Execute the executable with *executable_name data_file min_sup out_file*
      For example ./fim.out data.txt 0.001 output.txt
  - Your own code from HW1
    - If you use your own code, you can get **20%** extra bonus point of PART II.

    Ex: You get 25 points in PART I and 60 points in PART II. If you use the sample code, your final score is 25+60 = 85. If you use your own code, your final score is 25+60*1.2 = 97.
- We will run your code using command:

  ```
  ./eclat_cpu.sh $1 $2 $3
  ./eclat_gpu.sh $1 $2 $3
  ```

  $1: input file, $2: minimum support, $3: output file

  Ex: ./eclat_gpu.sh data.txt 0.001 output.txt   for minimum support 0.1%

[Hint] Use reduction technique to sum the support fast.

Submission

- Submit a zip file containing your code and report.pdf. Name the zip file to studentID.zip
The zip file must contain: eclat_cpu.sh, eclat_gpu.sh, report.pdf, your code (both CPU and GPU version).
- No Plagiarism. You have to implement the code by yourself (do not use other sample code on the Internet).
- Accept late submission for 2 days after the deadline.
- Wrong submitted format will get 10 points penalty.
- Late submission penalty is 15 points per day.
- It is your responsibility to make sure the submission is completed. Showing an unsubmitted set of homework after the due date will not work.