Data Mining HW5

FIM algorithm on GPGPU

Due: 03:00, Dec 27, 2017

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

Input:

The retail dataset (attached in the zip file)

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 33
34 35 36
37 38 39 40 41 42 43 44 45 46 47
39 40 48 49
39 40 49 50 51 52 53 54 55 56 57 58 59
33 42 60 61 62 63
4 40 49
```

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

Part I. CUDA Installation (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 now 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 CUDA(75%)

- Implement a GPU version of Eclat algorithm using vertical bit vectors.
- The output format should be the same with output.txt. For example:

  1 40 (125)

  Each line represents one frequent itemset. The numbers in each line separated by a space is the items in the itemset. The number in the parentheses is the support of that itemset.
- You should be able to compile the provided sample code with the makefile.
- Execute the executable with ***executable_name data_file min_sup out_file*** for example ./fim.out retail.txt 0.001 outfile

- The sample code provides the CPU version of Eclat algorithm with no output. You can turn on the output by uncommenting the last for loop of ***mineCPU*** function.
- Fill the ***mineGPU*** function to perform vertical mining on GPU. Do not modify existing code except for the ***mineGPU*** function.
- You can add as many functions as you like. (You have to add at least one kernel function to run on GPU).
- Your code should be able to build with the provided makefile. **(10%)**
- Your code should produce correct output (correct itemsets and correct support value). **(15%)**
- Your GPU version should be faster than the CPU version. **(15%)**
- Plot the execution time of the GPU mining with
  - Different support value (0.06%, 0.07%, 0.08%, 0.09%, 0.1%)
  - Different block number (16, 32, 64, 128, 256)
  - Different thread number (16, 32, 64, 128, 256)

  Post the result in the report and add a short note about the reason of the different execution time. (5% each, 15% total)
- Briefly explain your parallel design in the report. The better the design, the higher the score. **(20%)**

[Hint 1] Use shared memory to store the intermediate result (ex: support value).
[Hint 2] Use reduction technique to sum the support fast.

Submission

- Submit a zip file containing main.cu and report.pdf. Do not add other files.
- Plagiarism is strongly prohibited.
- Accept late submission for 2 days after the deadline.
- Late submission penalty is 15 points per day.