HW 4-2: Blocked All-Pairs Shortest Path (Multi-cards)

Deadline: 4 Jan 2020, 23:59

Contents

Goal

This assignment helps you get familiar with multi-cards CUDA programming by implementing a blocked all-pairs shortest path algorithm. Besides, to measure the performance and scalability of your program, experiments are required. Finally, we encourage you to optimize your program by exploring different optimizing strategies for optimization points.

Problem Description

Please refer to the homework 4-1.

Input / Output Format

Command line specification

```
srun -N1 -n1 -c2 --gres=gpu:2 ./hw4-2 INPUTFILE OUTPUTFILE
```

- `INPUTFILE`: The pathname of the input file. Your program should read the input graph from this file.
- `OUTPUTFILE`: The pathname of the output file. Your program should output the shortest path distances to this file.

Input/Output specification

- Except the ranges of V is different from homework 4-1, other specification please refer to homework 4-1.
  - $2 \leq V \leq 60000$
  - $0 \leq E \leq V \times (V - 1)$
  - $0 \leq \text{src}_i, \text{dst}_i < V$
  - $\text{src}_i \neq \text{dst}_i$
  - if $\text{src}_i = \text{src}_j$ then $\text{dst}_i \neq \text{dst}_j$ (there will not be repeated edges)
  - $0 \leq w_i \leq 1000$
  - No need to consider the condition of signed integer overflow

Report

Answer the questions below. You are recommended to use the same section numbering as they are listed.

1. Implementation
   a. How do you divide your data?
   b. How do you implement the communication?
   c. Briefly describe your implementation in diagrams, figures and sentences.
2. Experiment & Analysis

   We encourage you to show your results by figures, charts, and description.

   a. System Spec

      If you didn't use our `hades` server for the experiments, please show the CPU, GPU, RAM, disk of the system.

   b. Weak Scalability

      Observe weak scalability of the mulit-GPU implementations.

   c. Time Distribution

      Analyze the time spent in:

      - computing
      - communication
      - memory copy (H2D, D2H)
      - I/O of your program w.r.t. input size.

      > Note:
      >
      > - You should explain how you measure these time in your programs and compare the time distribution under different configurations.
      > - We encourage you to generate your own test cases!
      > - You should consider V^3 as problem size.

   d. Others

      Additional charts with explanation and studies. The more, the better.
3. Experience & conclusion

Grading

1. Correctness (35%)

   There are 7 testcases, each case worth 5 points. When judging, we will use hidden testcases similar to `/home/pp20/share/hw4-2/cases/[01-07].1`.
2. Performance (30%)
   - We have 6 performance testcases similar to `/home/pp20/share/hw4-2/cases/p[30-36].1`.
   - We will use 6 hidden test cases (denoted C) to run your code. Your performance score will be given by:

     $$\sum S(C) \times \frac{T_{best}(C)}{T_{yours}(C)}$$
   - C is a test case
   - $S(C)$ is the score allocated to the test case.
   - $T_{best}(C)$ is the shortest execution time of all students for the test case, excluding incorrect implementations.

- $T_{yours}(C)$ is your shortest execution time for the test case, excluding incorrect implementations.
- $\sum S(C) = 30$

3. Report (20%)

   Grading is based on your evaluation results, discussion and writing. If you want to get more points, design as more experiments as you can.

   Must be a PDF document.

4. Demo (15%)

   A demo session will be held in the Lab. You'll be asked questions about the homework.

> Note:
> Note: Please implement this problem using two GPUs. Single GPU version is not accepted and will get 0 points.

Submission

Upload these files to ILMS. Do not compress them.

- hw4-2.cu
- Makefile (optional, if not submitted, please check your code can compile with default Makefile we provide)
- hw4-2_{student_ID}.pdf

Resources

- Use the `hw4-cat` command to view the binary test cases in text format.
- Resources are provided under `hades:/home/pp20/share/hw4-2/`:
  - `Makefile` - example Makefile
  - Correctness Testcases: `hades:/home/pp20/share/hw4-2/cases/c*`
  - Performance Testcases: `hades:/home/pp20/share/hw4-2/cases/p*`
  - Testcases time limit: `hades:/home/pp20/share/hw4-2/cases/*.json`
- Name your code `hw4-2.cu`. And under same folder, type `hw4-2-judge` to run the test cases.
- Scoreboard: https://apollo.cs.nthu.edu.tw/pp20/scoreboard/hw4-2/

Final Notes

- Contact TA via pp@lsalab.cs.nthu.edu.tw or iLMS if you find any problems with the homework specification, judge scripts, example source code or the test cases.
- You are allowed to discuss and exchange ideas with others, but you are required to write the code on your own. You'll get **0 points** if we found you cheating.