CS542200 Parallel Programming
HW4: Blocked All-Pairs Shortest Paths

Josh Kao

National Tsing Hua University

2014/12/11

Outline I

All-Pairs Shorest Paths

Given an $N \times N$ matrix $D = [d(i,j)]$ where $d(i,j)$ represents the shortest-path distance from a vertex i to a vertex j.
Let $D^{(k)} = [d^{(k)}(i,j)]$ be the result which all the intermediate vertices are in the set $\{1, 2, \ldots, k\}$.

Floyed-Warshall Method

$$d^{(k)}(i,j) = \begin{cases} weight\ of\ (i,j) & if\ k = 0; \\ min(d^{(k-1)}(i,j), d^{(k-1)}(i,k) + d^{(k-1)}(k,j)) & if\ k \geq 1. \end{cases}$$

The matrix $D^{(N)} = [d^{(N)}(i,j)] = D$ gives an answer to APSP problem.

FW Algorithm

Pseudo Code

for $k \leftarrow 1$ **to** N **do**
  for $i \leftarrow 1$ **to** N **do**
    for $j \leftarrow 1$ **to** N **do**
      $d[i][j] \leftarrow min(d[i][j], d[i][k] + d[k][j]);$
    end
  end
end

Parallel Strategy

- For each **k**, calculate $d^{(k)}(i, j)$ in parallel

FW algorithm CUDA version

for $k \leftarrow 1$ **to** N **do**
  FW_kernel$<<<\#blocks, \#threads>>>(\ \cdots\ )$;
end

But ....

Q. How can we do APSP on multi-GPUs individually?

Blocked FW algorithm

- Partition matrix into N/B x N/B blocks of B x B submatrices.
- For intance, **N** = 6, **B** (Blocking Factor) = 2 :

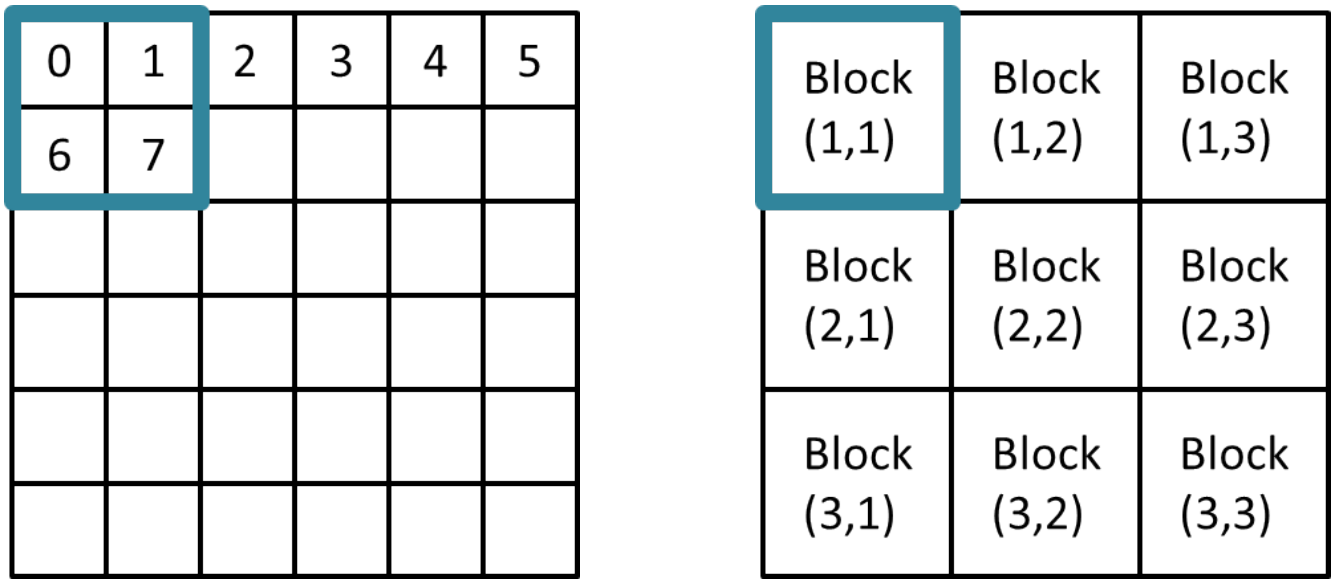
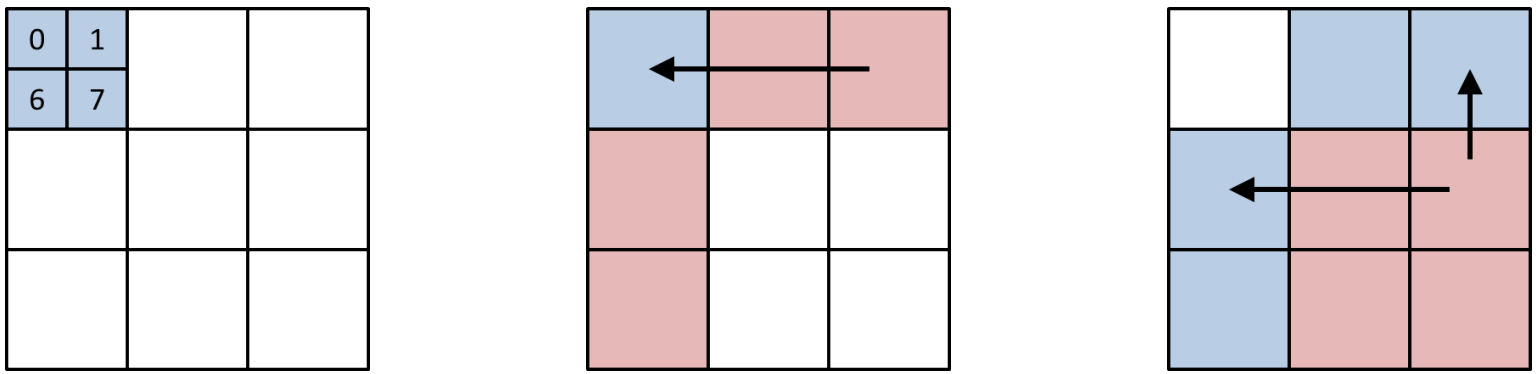


Figure: Divide a matrix by B = 2

Blocked FW algorithm

- 3 phases in each iteration
  - Phase 1: Self-dependent block
  - Phase 2: Pivot-row & Pivot-column blocks
  - Phase 3: Other blocks

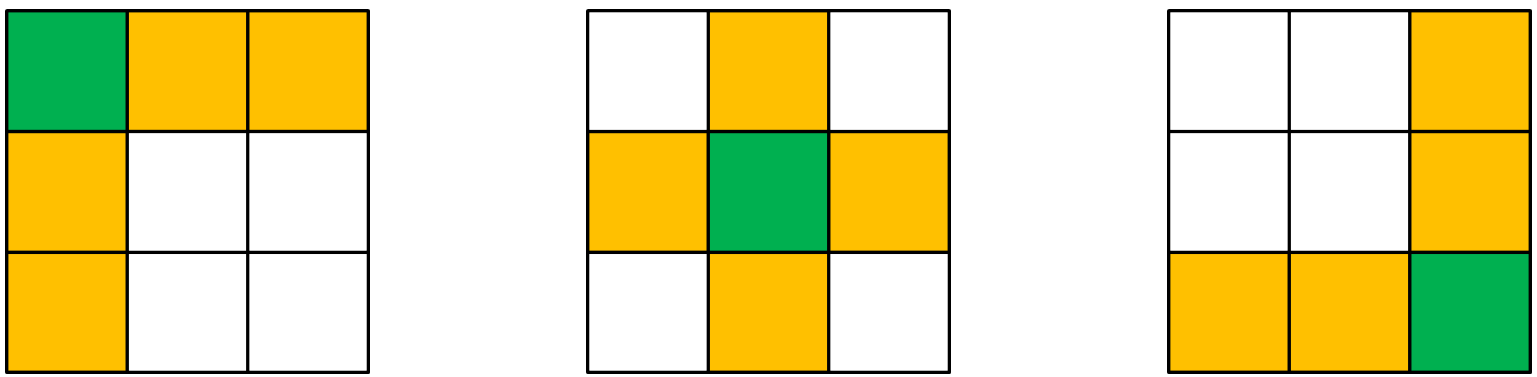


(a) Phase 1 (b) Phase 2 (c) Phase 3

Figure: Blocked algorithm in 1^{st} iteration

Blocked FW algorithm

- It will run **N/B** iterations
- In this example, it will run $6/2 = 3$ iterations.
  - Phase 1: **Green**; Phase 2: **Yellow**; Phase 3: **White**



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure: Blocked algorithm

Outline I

Execution Format

- Your program has to be able to **reading file**, and generate output in another file.
- Your program accepts two input parameters.

Format

$ ./{YOUR PROGRAM} {INPUT FILE} {OUTPUT FILE}

For instance,

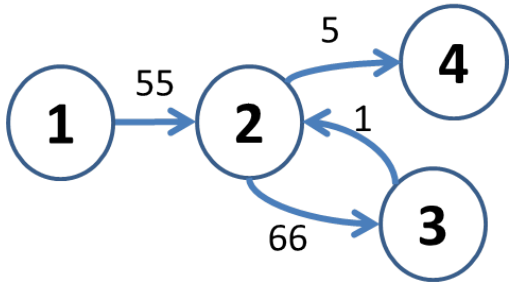
```
[kjs1095@lsalab ~]$ ./HW4_103065566_cuda.exe tiny_testcase output
[kjs1095@lsalab ~]$ diff -b output answer
```

Input Format

- The test case begins with a line contains 2 integers **N** and **M**
  - N : num of vertices $(1 \leq N \leq 6000)$
  - M : num of directed edges
- Each of the following **M** lines contains 3 integers **S**, **T** and **D**
  - S: source vertex id $(S \neq T)$
  - T: target vertex id
  - D: distance from **S** to **V** $(0 \leq D \leq 100)$

```
[kjs1095@lsalab ~]$ cat tiny_testcase
4 4
1 2 55
2 3 66
3 2 1
2 4 5
```

(a) Sample content of input



(b) Graph

Figure: Sample Input

Output Format

- Show the shortest distance from i^{th} vertex to j^{th} vertex
- If there is no path from i^{th} vertex to j^{th} vertex, output ***INF***

```
[kjs1095@lsalab ~]$ cat output
0 55 121 57
INF 0 66 5
INF 1 0 71
INF INF INF 0
```

Figure: Sample Output

Outline I

Implementation

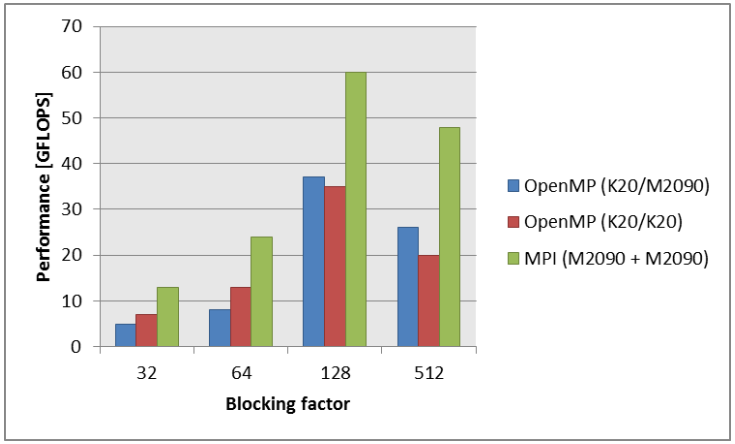
- Single GPU
- Multi GPU (MPI version)
- Multi GPU (OpenMP version)
- Makefile

Report

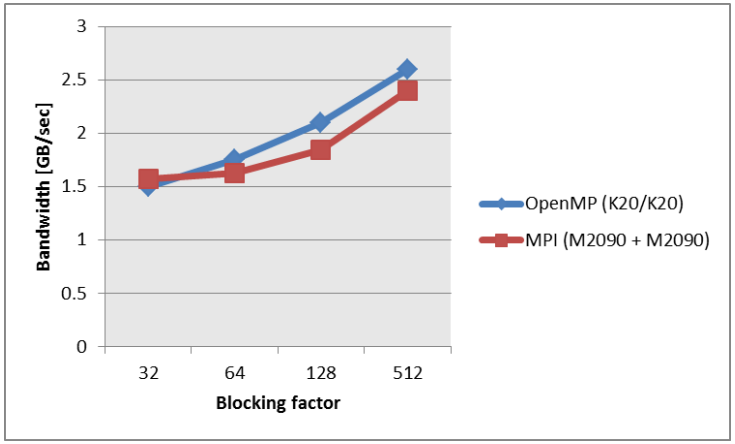
- **Implementation**
- **Profiling results**
- **Experiments**
  1. Weak Scalability & Time Distribution
  2. Blocking Factor
  3. Compare three implementations
  4. Others

  Think of the trend of the figure and explain.
- **Experience/Conclusion**

Example figures



(a) Performance trend



(b) Bandwidth trend

Figure: Example figures of blocking factor experiment

Outline I

Grading

1. **[4%]** Lab2
2. **[48%]** Correctness
3. **[30%]** Report
4. **[20%]** Demo
5. **[5%]** Bonus

Total score = *min*((1)+(2)+(3)+(4)+(5) , 100)
For each detail, check out the document on iLMS.

Outline I

Login to server

- **Host**: 140.114.91.176
- **Account**: Please refer to Account_Table.pdf
- **Password**: 1234userX [X=1 $\sim$ 67] (Default)

It's recommended to use ***passwd*** to change your password.

How to run MPI version?

- Execute **setting.sh** for connecting other machines
  - $. setting.sh
- Modify **hostfile**
  - Choose machines from gpucluster0 $\sim$ gpucluster3
- Modify **Makefile**
  - I used HW4_cuda_mpi.cu for example
  - You have to change it to your own file name.

Compile MPI version (Take HW4_cuda_mpi.cu for example)

```
$ make HW4_cuda_mpi.exe
```

Run MPI version

```
$ mpirun -np 2 -hostfile hostfile ./HW4_cuda_mpi.exe in1 out1
```

Reminder

- Upload **HW4_{Student-ID}.zip** to iLMS before 1/11(Sun) 23:59:59
  1. HW4_{Student-ID}_cuda.cu
  2. HW4_{Student-ID}_mpi.cu
  3. HW4_{Student-ID}_openmp.cu
  4. HW4_{Student-ID}_report.pdf
  5. Makefile
- Please **start your work ASAP** and do not leave it until the last day!
- Late submission penalty policy please refer to syllabus.
- Asking questions on iLMS or through e-mail is also welcome!