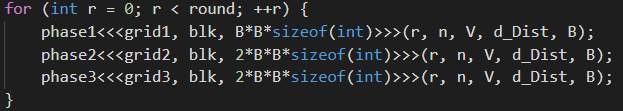
HW5 – Report

1. **Name:** 徐嘉駿, **Institute:** 資應所, **Student ID:** 107065528
2. **Implement**

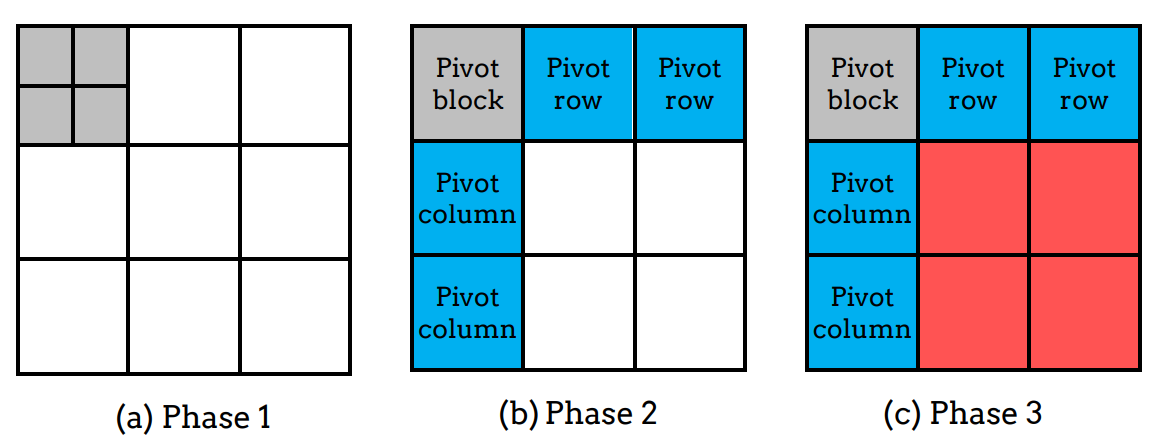
* **Divide Data:** 如同這次作業spec的切割方法，將data切成blocks，以下為configuration:
  1. **Blocking Factor:** 32
  2. **Blocks:** (data量/32)2
  3. **Threads:** 32\*32
* **Implementation:** 如同這次作業spec的實作方法，將過程分為好幾rounds，每round分為3個phases:



(1) **Phase 1:** 運算pivot block

(2) **Phase 2:** 運算pivot row blocks & pivot column blocks

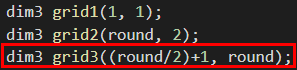
(3) **Phase 3:** 運算剩下的blocks



**Multi-GPUs:** 由於透過single GPU實驗時得知在phase3時要做最久，所以將phase3的n \* n矩陣切成上下兩塊，device 0做上半部，device 1 做下半部，如圖所示(假設6x6 matrix):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Device 0做黃色部分的data，device 1做綠色部分。



**Communication:** 在每一round開始之前，先將pivot row傳給需要更新的device，如上例所示(6x6 matrix)，在前3 round時，device 1只需要device 0的pivot row data，所以每round將pivot row由device 0傳給device 1，如下圖所示:

**Device 0 Device 1**

Round 1

Round 2

Round 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

後3 round時，device 0需要device 1的pivot row data，所以每round將pivot row由device 1傳給device 0，如下圖所示:

**Device 0 Device 1**

Round 3

Round 4

Round 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

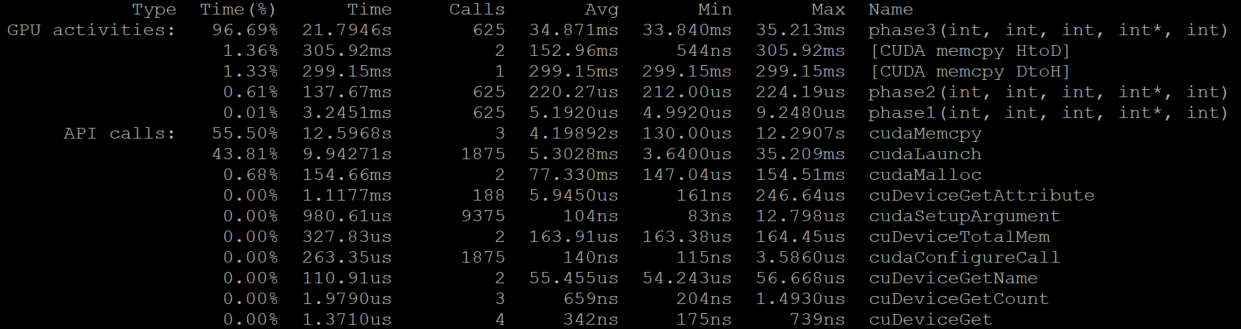
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Profiling Results**

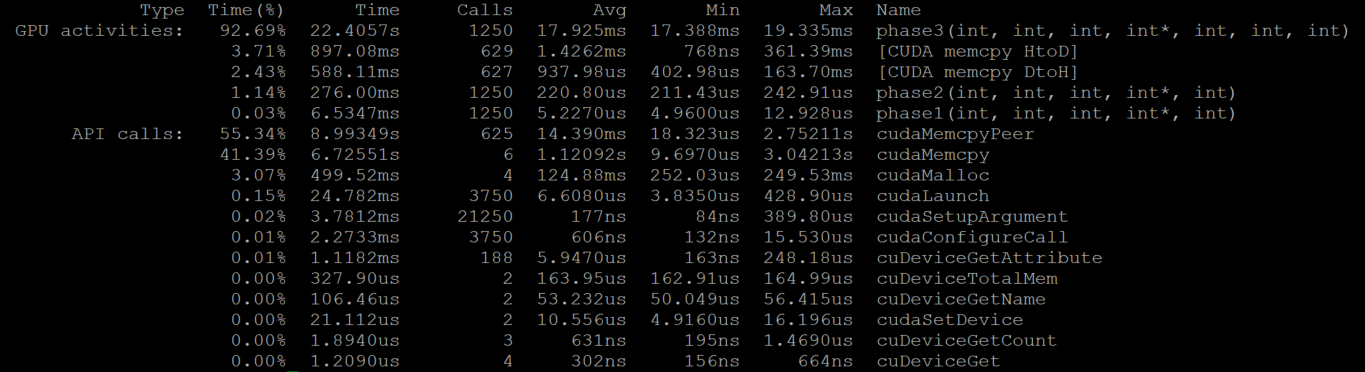
*使用p20k1做measurement*

以下為*nvprof*測量之各項結果:

* **Single GPU:**



* **Multiple GPUs:**



**結論:** 在single GPU & multiple GPUs裡，可以看到multiple GPUs在phase3明顯比single GPU少(multiple GPUs phase3 time必須除以2，因為nvprof會將兩個device time相加)，所以computing time明顯下降許多。此減少的時間變為增加communication (CudaMemcpyPeer)的時間。

1. **Experiment & Analysis**

* **Time Distribution:**

*使用以下test cases做measurement:*

*p11k1: vector ->11000, edge->505586*

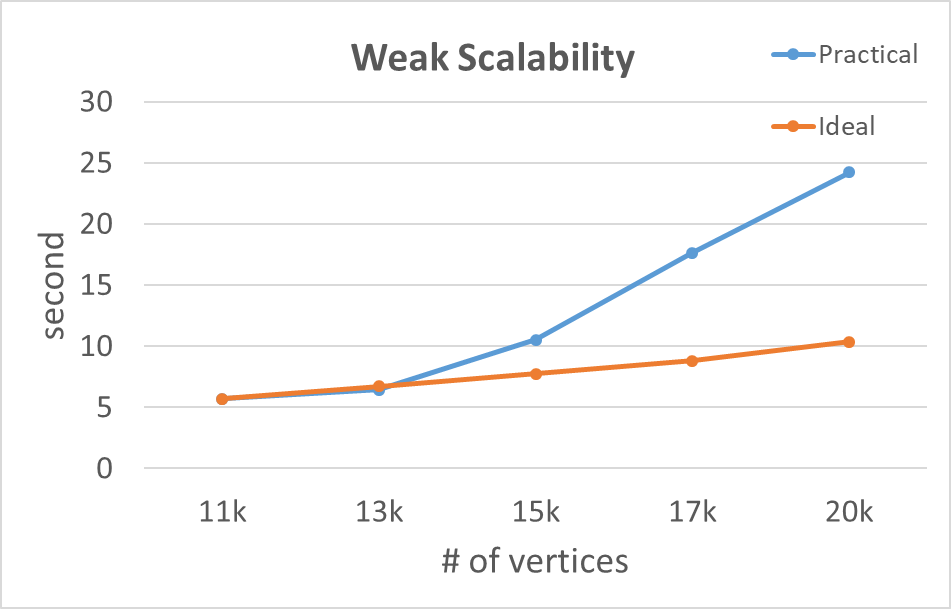
*p13k1: vector ->13000, edge->1829967*

*p15k1: vector ->15000, edge->5591272*

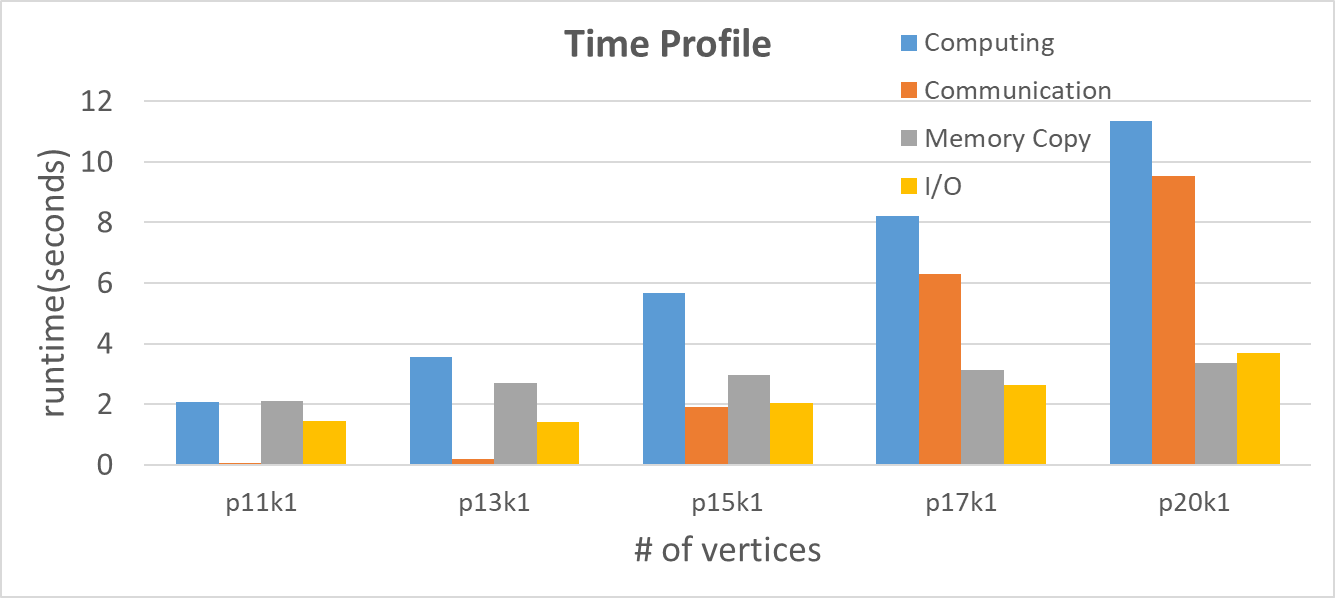
*p17k1: vector ->17000, edge->4326829*

*p20k1: vector ->20000, edge->264275*

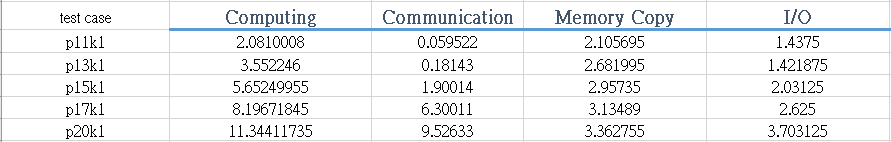
1. Weak Scalability



1. Time Distribution



數值:



1. **Conclusion**

這次的作業重點為GPU communication，一開始本來打算直接把整個vertex matrix互傳，但發現communication overhead會變得超級高，甚至比single GPU還差，所以只需要用的data變的十分重要，了解到這會直接嚴重影響performance。