



# 利用GPGPU進行雲端分析 之效能評估

0016016 余佳霖

# Status Report (1/3)

- Familiar with OpenCL parallel optimization strategy
  - Ex : matrix multiplication (200x ~ 600x speedup)
- Use these strategies to accelerate bigger program (HEVC)
  - Still working on it
  - Based on HM 12.0

	800	1024	1600	2048	3200
Serialize	4.263 s	16.351	45.813 s	258.812 s	377.182 s
Uncoalesced	1.704 s	3.989 s	14.823 s	32.017 s	118.497 s
Coalesced	0.099 s	0.174 s	0.742 s	1.269 s	5.884 s
Vector	0.074 s	0.151 s	0.582 s	1.143 s	4.674 s
TILE_N	0.055 s	0.114 s	0.427 s	0.889 s	3.423 s
TILE_M_N	0.042 s	0.086 s	0.331 s	0.665 s	2.628 s
Double	0.043 s	0.076 s	0.292 s	0.594 s	2.328 s
Tetraploid	0.031 s	0.057 s	0.211 s	0.441 s	1.686 s

# Status Report (2/3)

- Install and start using with gpgpu-sim for future analysis

```
root@tony-virtual-machine: /home/tony/cuda
-ptx_opcode_initiation_dp      8,8,8,130 # Opcode initiation intervals for double precision floating points <ADD,MAX,MUL,MAD,DIV>Default 8
,8,8,130
DRAM Timing Options:
nbk                8 # number of banks
CCD                2 # column to column delay
RRD                8 # minimal delay between activation of rows in different banks
RCD                12 # row to column delay
RAS                25 # time needed to activate row
RP                10 # time needed to precharge (deactivate) row
RC                35 # row cycle time
CDLR               6 # switching from write to read (changes tWTR)
WR                11 # last data-in to row precharge
CL                10 # CAS latency
WL                7 # Write latency
nbkgrp             1 # number of bank groups
CCDL               0 # column to column delay between accesses to different bank groups
RTPL               0 # read to precharge delay between accesses to different bank groups

Total number of memory sub partition = 6
addr_dec_mask[CHIP] = 0000000000000000          high:64 low:0
addr_dec_mask[BK]   = 0000000000003800          high:14 low:11
addr_dec_mask[ROW]  = 00000000fffc000          high:28 low:14
addr_dec_mask[COL]  = 00000000000007ff          high:11 low:0
addr_dec_mask[BURST] = 000000000000001f          high:5 low:0
sub_partition_id_mask = 0000000000000000

GPGPU-Sim uArch: clock freqs: 337500000.000000:600000000.000000:600000000.000000:800000000.000000
CPGPU-Sim uArch: clock periods: 0.00000000296296296:0.000000016666666666667:0.0000000166666666667:0.00000001250000000000
*** Initializing Memory Statistics ***
GPGPU-Sim uArch: interconnect node map (shaderID+MemID to icntID)
GPGPU-Sim uArch: Memory nodes ID start from index: 8
GPGPU-Sim uArch:  0  1  2
GPGPU-Sim uArch:  3  4  5
GPGPU-Sim uArch:  6  7  8
GPGPU-Sim uArch:  9 10 11
GPGPU-Sim uArch: 12 13
GPGPU-Sim uArch: interconnect node reverse map (icntID to shaderID+MemID)
GPGPU-Sim uArch: Memory nodes start from ID: 8
CPGPU-Sim uArch:  0  1  2
GPGPU-Sim uArch:  3  4  5
GPGPU-Sim uArch:  6  7  8
GPGPU-Sim uArch:  9 10 11
```

# Status Report (3/3)

- Simple machine learning program implementation
  - $K^{\text{th}}$  Nearest Neighbor learning
  - Neural Network
  - AdaBoost Algorithm
  - Support Vector Machine

# Future Schedule

- 2014/7 ~ 2014/9

Accelerate Image Processing Libraries based on OpenCL

- 2014/9 ~ 2014/10

Finish the HEVC parallel acceleration project

- 2014/11 ~ 2015/2

Using gpgpu-sim to analysis the OpenCL based program