

Модель исполнения программ на видеопроцессорах AMD

Остапенко А.Д.

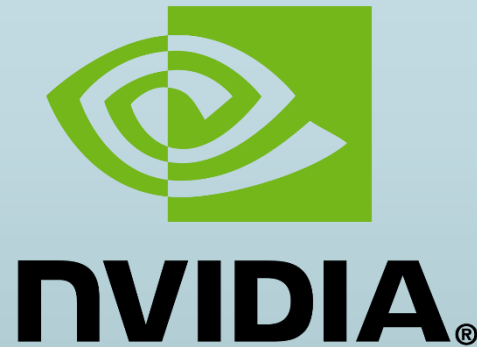
Институт математики, механики и компьютерных наук им. И.И. Воровича

Кафедра алгебры и дискретной математики

Научный руководитель: Гуда С. А.

Модель Hong, Kim

Простая аналитическая модель оценки времени выполнения параллельных программ для Nvidia.



Модель Hong, Kim

$$Mem_L_Uncoal = Mem_LD + (\#Uncoal_per_mw - 1) \times Departure_del_uncoal$$

$$Mem_L_Coal = Mem_LD$$

$$Mem_L = Mem_L_Uncoal \times Weight_uncoal + Mem_L_Coal \times Weight_coal$$

$$Weight_uncoal = \frac{\#Uncoal_Mem_insts}{(\#Uncoal_Mem_insts + \#Coal_Mem_insts)}$$

$$Weight_coal = \frac{\#Coal_Mem_insts}{(\#Coal_Mem_insts + \#Uncoal_Mem_insts)}$$

$$Departure_delay = (Departure_del_uncoal \times \#Uncoal_per_mw) \times Weight_uncoal + Departure_del_coal \times Weight_coal$$

$$MWP_Without_BW_full = Mem_L / Departure_delay$$

$$MWP_Without_BW = MIN(MWP_Without_BW_full, \#Active_warps_per_SM)$$

$$MWP = MIN(MWP_Without_BW, MWP_peak_BW, N_{pw})$$

$$Mem_cycles = Mem_L_Uncoal \times \#Uncoal_Mem_insts + Mem_L_Coal \times \#Coal_Mem_insts$$

$$Comp_cycles = \#Issue_cycles \times (\#total_insts)$$

$$CWP_full = \frac{Mem_cycles + Comp_cycles}{Comp_cycles}$$

$$CWP = MIN(CWP_full, N)$$

$$N = \#Active_warps_per_SM$$

$$\#Rep = \frac{\#Blocks}{\#Active_blocks_per_SM \times \#Active_SMs}$$

If (MWP is N warps per SM) and (CWP is N warps per SM)

$$Exec_cycles_app = (Mem_cycles + Comp_cycles + \frac{Comp_cycles}{\#Mem_insts} \times (MWP - 1)) \times \#Rep$$

Else if (CWP >= MWP) or (Comp_cycles > Mem_cycles)

$$Exec_cycles_app = (Mem_cycles \times \frac{N}{MWP} + \frac{Comp_cycles}{\#Mem_insts} \times (MWP - 1)) \times \#Rep$$

Else

$$Exec_cycles_app = (Mem_L + Comp_cycles \times N) \times \#Rep$$

Перенос модели на AMD

- 1) Определение параметров видеокарты используемых в модели(Mem_LD, Departure_del_uncoal, Departure_del_coal).
- 2) Определение скорости выполнения операций.
- 3) Применение модели и корректировка.



Определение параметров видеокарты используемых в модели

```
int i = get_global_id(0), k = -1;
float s = 0;
s += x[++k*h + i];
...
s += x[++k*h + i];
res[0] = s;
```

Departure_del_uncoal

```
#define IND (++k * h + i) * 33
int i = get_global_id(0), k = -1;
float s = 0;
s += x[IND];
...
s += x[IND];
res[0] = s;
```

Departure_del_coal

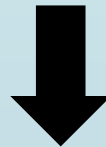
```
long s = 0;
int mh = m*h;
for (int i = 0; i < mh; i += h)
s += x[i];
res[0] = s;
```

Mem_LD

Определение скорости выполнения операций

В чем проблема?

```
x = 736.0f / x;
```



```
w: MAX_DX10      ____, KC0[0].x, -KC0[0].x
1 z: SETGT_DX10   ____, PV0.w, (0x6F800000, 7.922816251e28f).x
2 y: CNDE_INT     R0.y, PV1.z, 1065353216, 796917760
3 w: MUL_e        ____, KC0[0].x, PV2.y
4 x: RCP_e        ____, PV3.w
  y: RCP_e        ____, PV3.w
  z: RCP_e        ____, PV3.w
5 z: MUL_e        ____, PV4.x, (0x44380000, 736.0f).x
6 y: MUL_e        R0.y, R0.y, PV5.z
```

Определение скорости выполнения операций

Решение:

Нужно написать программу, которая бы запускала микробенчмарки (огромное количество однотипных операций) и считала *throughput* для каждой операции.

$$\text{Throughput} = \frac{\text{global_work_size} * N}{T * \text{GpuMHz} * \text{computeUnits} * 1e6}$$

Определение скорости выполнения операций

Результат работы программы на видеокарте HD 6970:

Throughput			Нормированные	
Операция	Обычный тип	Векторный тип	Обычный тип	Векторный тип
Сложение с float	21	40	2	1
Вычитание с float	22	40	2	1
Умножение с float	22	40	2	1
Деление с float	2	2	20	20
sin	1		40	
native sin	2		20	

Применение модели

```
uint i,j, id = get_global_id(0), sz = get_global_size(0);
for (int i = id; i<N; i += sz)
{
    float loc_x = x[i];
    float sum=.0f;
    for (j=0; j<M; j++)
        sum += 1.0f / (loc_x * loc_x + j);
    result[i] = sum;
}
```

Тестовая программа

Применение модели

$$1. \text{ Exec_cycles_app} = (\text{Mem_cycles} + \text{Comp_cycles} + \frac{\text{Comp_cycles}}{\text{Mem_insts}} * (\text{MWP} - 1)) * \text{Rep}$$

$$2. \text{ Exec_cycles_app} = (\text{Mem_cycles} * \frac{N}{\text{MWP}} + \frac{\text{Comp_cycles}}{\text{Mem_insts}} * (\text{MWP} - 1)) * \text{Rep}$$

$$3. \text{ Exec_cycles_app} = (\text{Mem_L} + \text{Comp_cycles} * N) * \text{Rep}$$

Результаты:

Exec_app_cycles	2347008
Время работы (расчетное)	0,002607787
Время работы (измеренное)	0,0028