JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Day 5

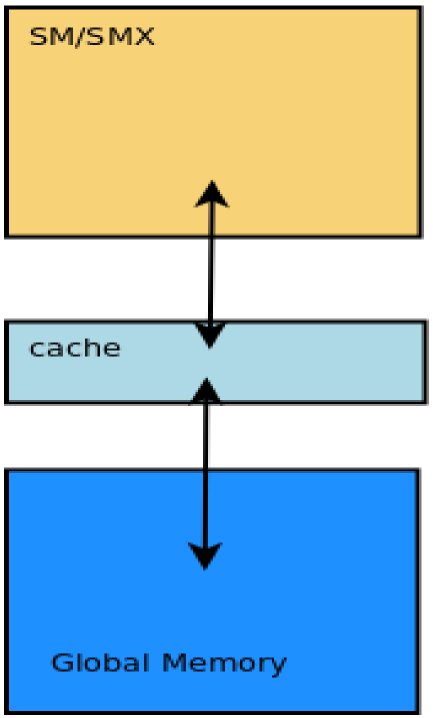
- Revise the idea of coalescing global memory access
- CUDA Visual Profiler

Dr. Tuan-Tu Tran
trant@uni-mainz.de

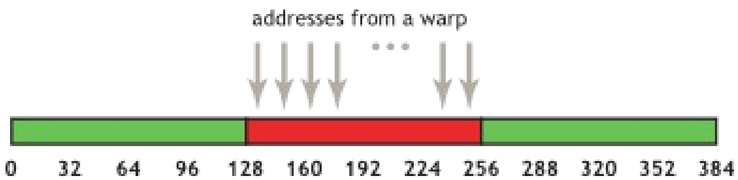
Part 1

Revise the idea of coalescing global memory access

When a SM/SMX accesses to a global memory



Cache line: 128 Bytes



1 cache access transaction

2 cache access transaction

More details can be found in (*in the relation with the cache and the capability of the device*):

NVIDIA, CUDA C Best Practice Guide,
http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html#coalesced-access-global-memory

Efficient implementation

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

- To read N integer from global memory to shared memory:

which one is better ?

Method 1

```
__global__ void kernel(int* gData, int N)
{
	extern int sData[];
	int nThread = blockDim.x;
	int tid = threadIdx.x;
	int threadLoad = N/nThread;
	for(int i=0;i<threadLoad;i++)
	{
		int index = tid + i*nThread;
		if(index < N) sData[index] = gData[index];
	}

}
```

Method 2

```
__global__ void kernel(int* gData, int N)
{
	extern int sData[];
	int nThread = blockDim.x;
	int tid = threadIdx.x;
	for(int i=tid;i<N;i+=nThread)
	{
		Sdata[i] = gData[i];
	}

}
```

Efficient implementation

- To read N integer from global memory to shared memory:

Ex: N = 64, nThread = 32

Method 1: 4 cache transactions

```
__global__ void kernel(int* gData, int N)
{
    extern int sData[];
    int nThread = blockDim.x;
    int tid = threadIdx.x;
    int threadLoad = N/nThread;
    for(int i=0;i<threadLoad;i++)
    {
        int index = tid + i*nThread;
        if(index < N) sData[index] = gData[index];
    }

}
```

Method 2: 2 cache transactions

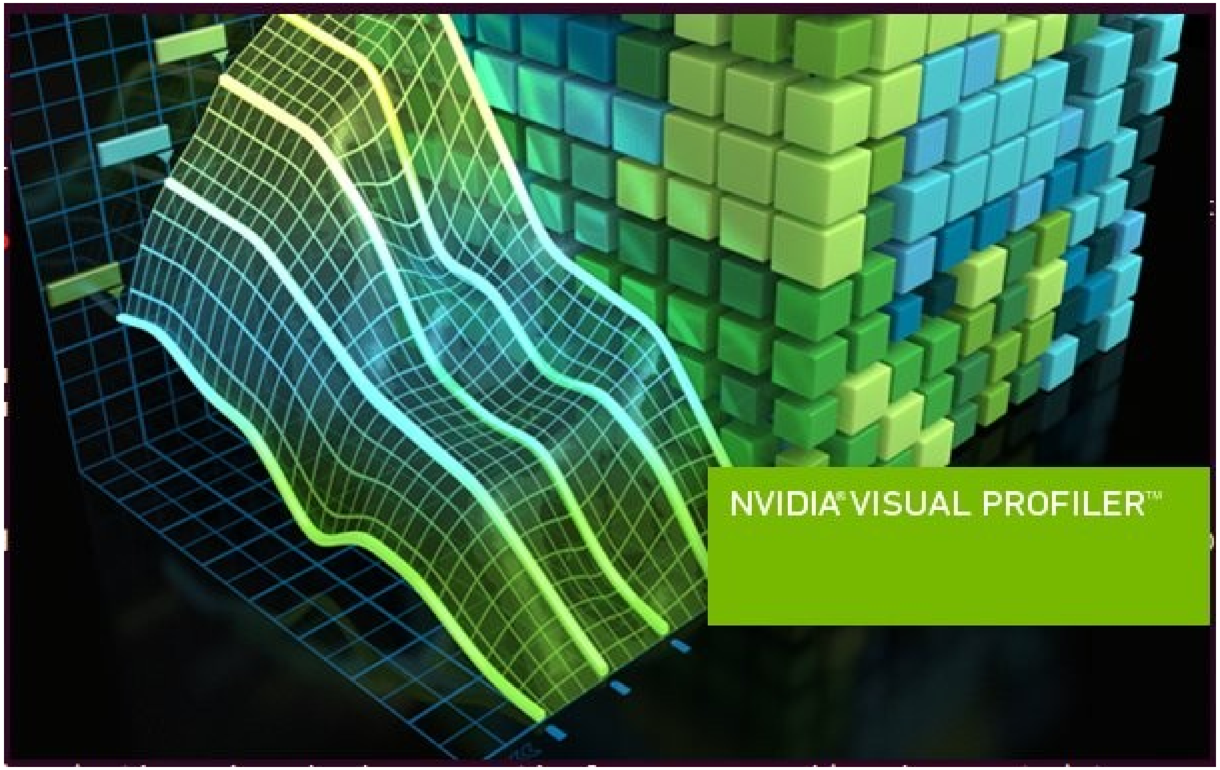
```
__global__ void kernel(int* gData, int N)
{
    extern int sData[];
    int nThread = blockDim.x;
    int tid = threadIdx.x;
    for(int i=tid;i<N;i+=nThread)
    {
        Sdata[i] = gData[i];
    }

}
```

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Part 2

CUDA Visual Profiler



NVIDA Visual Profiler

A power tool to examine and analyze CUDA applications.

The profiler is included in the NVIDIA Software Development Kit (SDK) and is installed in the computed node of Mogon.

- By remote access, you can launch the NVIDIA Visual Profiler:
  - With graphical interface: remote access with the X server forwarding (ex: ssh -X …), type: nvvp
  - With command line: nvprof.

    ```
    Ex: nvprof --print-gpu-trace ./cudaReduction -i 100MInt.dat -n 10000000
    ```

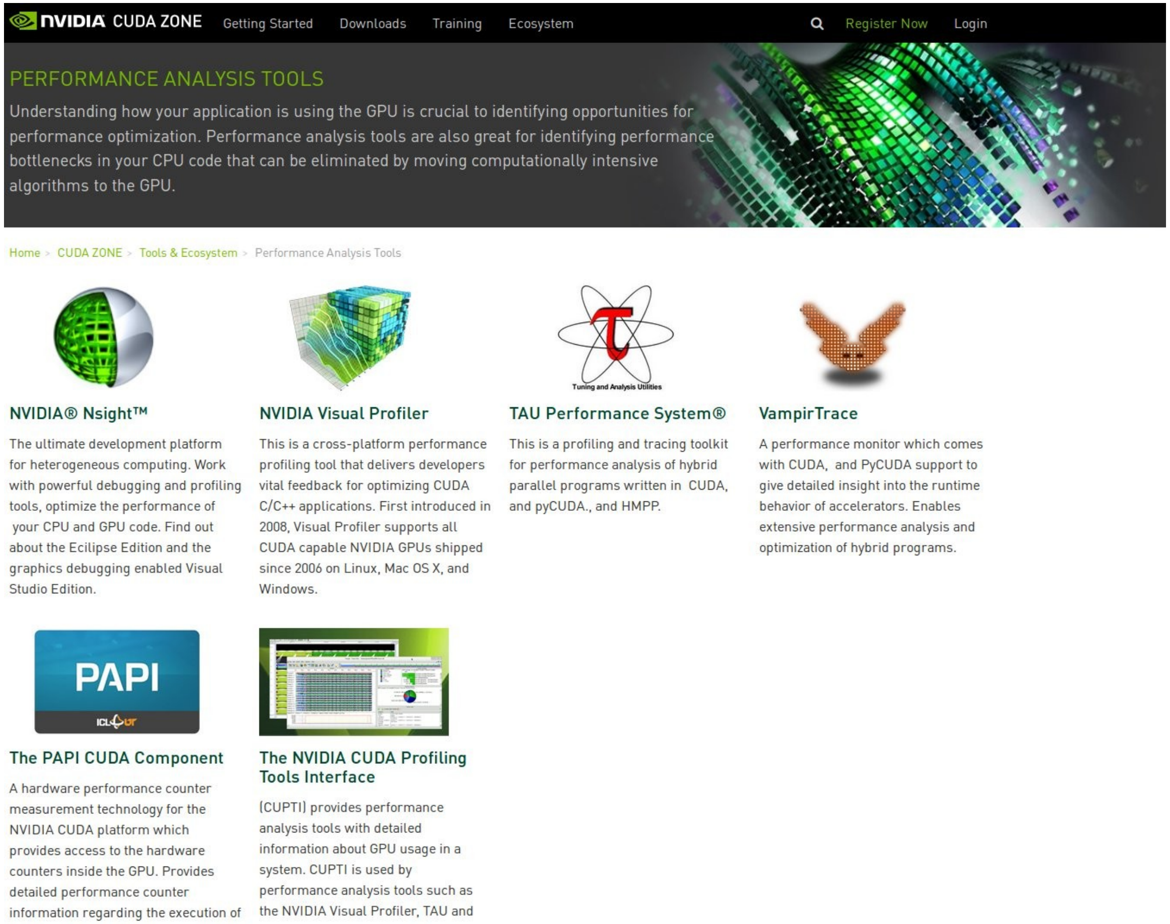
- More details can be found in:

  NVIDA Profiler User's Guide,
  http://docs.nvidia.com/cuda/profiler-users-guide/

NVIDIA Tools for CUDA

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

https://developer.nvidia.com/performance-analysis-tools



Question ?