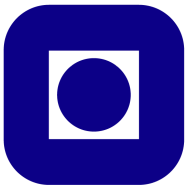

NTNU

TDT4200 - Parallel Computing

Problem set 2

Mathias Ose

October 7, 2015

Part 1

1a

Maxwell is a GPU architecture, with homogenous cores with threaded execution.
big.LITTLE is a CPU architecture, with heterogenous cores with threaded execution.
Vilje has a clustered supercomputer architecture, with homogenous cored with threaded execution and NUMA.
A typical CPU is homogenous with threaded execution.

1b

SIMT is a subcategory of SIMD. A scheduling algorithm groups threads with the same instructions together is warps, and the threads in a warp are executed together concurrently.

Maxwell implements SIMT. Nvidia provides compilers for various languages that link in their libraries automatically to make the programs SIMT-runnable.

1c

i) SIMD/SIMT. The kernel of the program is executed as multiple threads on multiple data.
ii) MIMD. Heterogenous computing does not lend itself to neither SI nor SD easily.
iii) MIMD. Lots of different programs are simultaneously running, each with their own datasets.
iv) MIMD. In for example a modern PC, lots of different programs with their own data are running at the same time. A single core context switches between different programs frequently.

2a

A thread is the execution of a set of instructions on some data. Multiple threads make up a block. In each block, threads are executed in paralell. Blocks are organized into a grid. Each block in the grid is executed by a core, not synced. If there are more blocks than cores they are queued.

2b

$$t_{CPU} = 3500 * n * \log_2 n$$
$$t_{GPU} = 35 * n * \log_2 n + \frac{7n}{r}$$

let $t_{CPU} = t_{GPU}$, solve for n

$$n = 2^{\frac{1}{495r}}$$

2c

Kernel 2 is faster. Kernel 1 is "throwing" away a lot of paralellism by not using all the threads spawned for anything productive.

2d

i) A grouping of 32 threads within the same block that execute at the same time. The better warps are utilized (by using all 32 slots and by running as many concurrent warps as possible), the better the performance will be.
ii) The proportion of active warps out of the maximum possible active warps. Resource limits or conflicts cause occupancy to be reduced.
iii) Placing data that needs to be accessed together near each other in physical memory to reduce the number of accesses needed.
iv) Sections of global memory where data needed by threads that can not fit in registers can be stored. Performance will be better if local memory isn ot used and data can be fitted into registers instead.
v) Memory that can be used by all the threads. Conflict-prone if the threads need to write to it.

Part 1

1c

```
$ time ./gpu_version
transfer time: 0.355072 ms

real    0m0.216s
user    0m0.090s
sys     0m0.097s
```

The transfer time is 0.001643852% of the running time.