



GPU Exercises

STRONGnet Summer School, June 14-25, 2011

All files for this exercise are located in the `/home/gpu/glattice` directory on host `bam2`. A documentation (html/pdf) for the used classes can be found in the `doc` directory.

Exercise 4: Coalesced memory access

In the staggered formulation of Lattice QCD one frequently encounters products of SU3 matrices with color vectors at each lattice site. GPU and CPU codes might prefer different memory layouts. The file `su3timesvect.cu` demonstrates how to do this multiplication using the **GPU**`su3`- and **GPU**`cvect3`-classes on the CPU and measure the elapsed time.

- a) Implement a CUDA version of the above code and measure its performance. (Remember to allocate the device memory and copy the data from the host to the device. However, do not include the copying in the timing measurements)
- b) Use the classes **GPU**`su3array` and **GPU**`cvect3array` to store the data in a layout that allows for coalesced access. The classes allow for memory allocation on the device or host by the first argument of the constructor and also provide copy functions for your data. Implement a second CUDA kernel that uses these array classes via the corresponding accessors (returned by the `getAccessor()` function) and measure its performance.
- b*) Optional: Without looking at the details of the array classes used in b) implement a coalesced memory layout for the SU3 matrices and color vectors.
- c) Implement a CPU version using the array classes used in b) and measure its performance.

Exercise 5: Plaquette calculation

The plaquette is defined as

$$\frac{1}{18V_4} \sum_{x \in V_4} \sum_{\mu=0}^3 \sum_{\nu=0}^{\mu-1} \text{Re Tr } U_\mu(x) U_\nu(x + \mu) U_\mu^\dagger(x + \nu) U_\nu^\dagger(x) .$$

The file `plaquette.cpp` contains a driver that initializes a **GPU**`su3lattice` object and calls the plaquette calculation on the CPU and GPU.

- a) In the class **GPU**`su3lattice` implement the `plaquette()` function that calculates the plaquette on the CPU. The links are stored in the **GPU**`su3array` `h_smeared` field. The necessary functions for index calculations are located in the file `cu-su3lattice.hcu`. The link $U_\mu(x)$ is stored at position $x + V_4\mu$ in the `h_smeared` field.
- b) Implement the function `dplaquette()` that calculates the plaquette using a CUDA kernel. The links are stored in **GPU**`su3array` `d_smeared` field on the device. Use the `reduce` function from the previous exercises for the sum over lattice sites x .