

PASI 2011 - Sparse - Lab 1

Summary

In this lab we'll implement SpMV for the CSR and ELL formats using CUDA.

Step 1: Extract skeleton code

```
$ unzip lab1.zip
```

Step 2: Compile initial skeleton with make

```
$ cd lab1
$ nvcc csr_simple.cu -o csr_simple
$ nvcc csr_bench.cu -o csr_bench
$ nvcc ell_simple.cu -o ell_simple
$ nvcc ell_bench.cu -o ell_bench
```

Step 3: Run initial binaries

```
$ ./csr_simple
Comparing CSR SpMV results...
  CPU result  GPU result
    7.0000    0.0000
   26.0000    0.0000
   12.0000    0.0000
   38.0000    0.0000
```

Step 4: Implement SpMV for CSR format using CUDA in kernels.h

Edit kernels.h with your favorite editor, e.g.

```
$ gvim kernels.h
```

Add new `__global__` functions named `csr_kernel` and `ell_kernel` to perform the SpMV operation. Launch the `csr_kernel` function from inside `csr_spmv_device` and launch the `ell_kernel` function from inside `ell_spmv_device`. For example

```
    csr_kernel<<<grid_size, block_size>>>( ... );
```

and

```
    ell_kernel<<<grid_size, block_size>>>( ... );
```

When you're ready to compile,

```
$ make
```

and then run

```
$ ./csr_simple
```

to perform a simple test of your CSR SpMV kernel. If that succeeds then run

```
$ ./csr_bench
```

to test the performance of your CSR kernel.

Step 5: Implement SpMV for ELL format using CUDA in kernels.h

Same as the previous step, using `ell_simple` and `ell_bench` respectively.