CUDA Kernels for SpMV

Daniel Butnaru, Christoph Kowitiz

January 23, 2012
Table of Contents

1 Compressed Sparse Row Kernels
   • Kernel 1
   • Kernel 2

2 ELLPACK Kernel

3 Assignments
   • Page Rank
   • Heat Equation
Compressed Sparse Row (CSR) Kernel 1

First straightforward approach: each thread does a row times vector multiplication

```c
__global__ void k_csr_mat_vec_mm(ptr, J, Val, x, y) 
{
    int row = blockDim.x * blockIdx.x + threadIdx.x ;
    if (row < num_rows){
        float dot = 0;
        int row_start = ptr [ row ];
        int row_end = ptr [ row+1];

        for (int jj = row_start; jj < row_end; jj++) {
            dot += Val [ jj ] * x [ J [ jj ]];
        }

        y[row] += dot ;
    }
}
```
Compressed Sparse Row (CSR) Kernel 1 (cont.)

Observations:
- contiguos storage of column and values vectors, but
- kernel does **non-coalesced** memory access

**Access pattern:**

<table>
<thead>
<tr>
<th>ptr</th>
<th>[0 2 4 7 9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>indices</td>
<td>[0 1 1 2 0 2 3 1 3]</td>
</tr>
<tr>
<td>data</td>
<td>[1 7 2 8 5 3 9 6 4]</td>
</tr>
</tbody>
</table>

**Iteration 0**  [0 1 2 3 ]
**Iteration 1**  [ 0 1 2 3]
**Iteration 2**  [            2 ]
Compressed Sparse Row (CSR) Kernel 2

tbd. next session
ELLPACK (ELL) Kernel

Straightforward approach: each thread multiplies one row with the vector.

```c
__global__ void k_ell_mat_vec_mm (N, int num_cols_per_row,
                                   indices, data, x, y) {

    int row = blockDim.x * blockIdx.x + threadIdx.x;

    if ( row < N ){
        float dot = 0;
        for ( int n = 0; n < num_cols_per_row ; n ++){
            int col = indices [ N * n + row ];
            float val = data [ N * n + row ];
            if ( val != 0)
                dot += val * x [ col ];
        }
        y [ row ] += dot ;
    }
}
```
ELLPACK (ELL) Kernel 1 (cont.)

Observations:
- kernel **does** coalesced memory access
- x vector might not necessarily be accessed contiguously

Access pattern:

```
| data   | [1 2 5 6 7 8 3 4 * * 9 *] |
| indices| [0 1 0 1 1 2 2 3 * * 3 *] |

Iteration 0    [0 1 2 3 ]
Iteration 1    [          0 1 2 3 ]
Iteration 2    [                  0 1 2 3 ]
```
Assignment 1: Page Rank

- load M (matrix market format) matrix from disk (e.g. `flickr.mtx`)
  http://www.cise.ufl.edu/research/sparse/MM/Gleich/flickr.tar.gz
- transform it from COO to CSR
- apply pagerank on M
  - transpose M (during COO to CSR transformation)
  - make M stochastic
  - multiply using CSR kernel
  - regularize
  - stop on error criterium

Tips

- do on paper several COO to CSR transformations to make sure you understand the format
- write your own very small examples to test
- check for CUDA errors after each CUDA call
Assignment 2: Solve 1D heat equation

tbd. next session