Accelerating MATLAB with CUDA

Massimiliano Fatica
NVIDIA
mfatica@nvidia.com

Won-Ki Jeong
University of Utah
wkjeong@cs.utah.edu

NVIDIA
MATLAB can be easily extended via MEX files to take advantage of the computational power offered by the latest NVIDIA GPUs (G8x):

- G80 is a massively parallel processor (up to 128 processors running at 1.5Ghz)
- Current generation has support of IEEE-754 single precision, double precision will be available towards the end of the year.

Programming the GPU for computational purpose was a very cumbersome task before CUDA:

- CUDA is a C-like programming language and includes FFT and BLAS libraries. CUDA is freely available (Windows and Linux)

This work shows the feasibility and benefits of this approach
To interface CUDA and MATLAB we had to slightly modify the MEX infrastructures:

CUDA files have a .cu suffix and need to be compiled with a specific compiler (nvcc)


**Initial focus on 2D FFTs:**
FFT-based methods could be often used in single precision.

**Application selected:**
Pseudo-spectral solution of Euler equation
## Results

### 2D isotropic turbulence: 1024x1024 mesh, 400 RK4 steps on Windows, Opteron 250, Quadro FX5600.

<table>
<thead>
<tr>
<th></th>
<th>Runtime</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard MATLAB</td>
<td>8098 s</td>
<td></td>
</tr>
<tr>
<td>Overload FFT2 and IFFT2</td>
<td>4425 s</td>
<td>1.8x</td>
</tr>
<tr>
<td>Overload Non-linear term</td>
<td>735 s</td>
<td>11x</td>
</tr>
<tr>
<td>Overload Non-linear term, FFT2 and IFFT2</td>
<td>577 s</td>
<td>14x</td>
</tr>
</tbody>
</table>

### Advection of elliptic vortex: 256x256, Linux, Opteron 250, Quadro FX5600.

<table>
<thead>
<tr>
<th></th>
<th>Runtime</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard MATLAB</td>
<td>168 s</td>
<td></td>
</tr>
<tr>
<td>Overload Non-linear term, FFT2 and IFFT2</td>
<td>14.9 s</td>
<td>11x</td>
</tr>
</tbody>
</table>