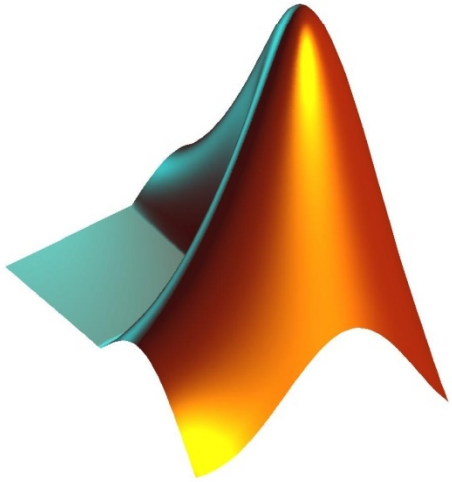
MATLAB with CUDA

By
Vishwanath Sarkar
NIT Bhopal

MATLAB



MATLAB

Advantages :

1. Large collection of built in functions.
2. Custom C or Fortran codes can be called directly from MATLAB as they were MATLAB built in functions using MEX files.
3. Very efficient and easy to use for image and signal processing tasks.

Major Restriction :

1. Slow speed of processing in few image processing and computer vision tasks.

   For ex. Object tracking

CUDA

CUDA

1. Gives computationally intensive applications access to the tremendous processing powers of the latest GPUs.
2. Uses a C-like programming language and does not require remapping algorithms to graphics concepts.
3. Exposes several hardware features that are not available via the graphics API.

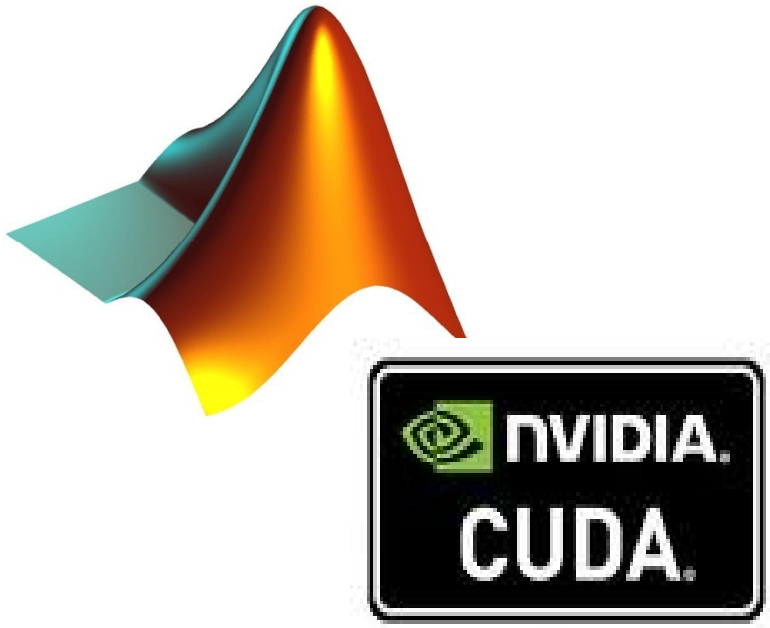
For ex.

Shared Memory

a. It is a small (currently 16KB per multiprocessor) area of on-chip memory which can be accessed in parallel by blocks of threads.
b. Allows caching of frequently used data and can provide large speedups over using textures to access data.

c. Combined with a thread synchronization primitive, this allows cooperative parallel processing of on-chip data, greatly reducing the expensive off-chip bandwidth requirements of many parallel algorithms.
d. Benefits a number of common applications such as linear algebra, Fast Fourier Transforms, and image processing filters.

MATLAB+CUDA



1. MATLAB could be easily extended via MEX files to take advantage of the computational power offered by the latest NVIDIA graphics processor unit (GPU).
2. It offers remarkable speedup for computer vision tasks.

Solutions

Solution 1
MATLAB plug-in for CUDA

1. This MATLAB plug-in for CUDA provides: acceleration of standard MATLAB 2D FFTs and CUDA/MEX example plug-in and build environment using Chris Bretherton's Fourier spectral simulation of 2D fluid flow MATLAB scripts from his course material at the University of Washington.
2. When MATLAB makes 2D FFT calls of any size, the NVIDIA plug-in intercepts them and handles them with a MEX file that in-turn utilizes an optimized CUDA FFT implementation on the GPU.
3. This is transparent to MATLAB users.
4. The plug-in is available for download for both Windows and Linux here :

   http://developer.nvidia.com/object/matlab_cuda.html

5. In order to interface CUDA and MATLAB, user have to slightly modify the MEX infrastructure. CUDA files have a .cu suffix and needs to be compiled with a specific compiler (nvcc).
6. The scripts created to compile CUDA MEX files are very easy to use. From the command prompt in MATLAB, the user needs to invoke nvmex instead of the regular mex
7. script:

   nvmex -f nvmexopts.bat filename.cu -IC:\cuda\include -LC:\cuda\lib –lcudart
8. CUDA drivers, toolkits and sdk examples are available for download at

   http://www.nvidia.co.in/object/cuda_get_in.html

9. A simple example that creates a MEX file which calls CUDA to add two length-5 vectors can be obtained here *http://www.mathworks.com/matlabcentral/fileexchange/22436-example-of-cuda-and-matlab-and-nothing-else-for-windows*
10. A very good guide for using CUDA and MATLAB mex can be obtained from here

    http://www.cs.ucf.edu/~janaka/gpu/index.htm

    The nvmex tool for such files is also available for dowload at link above.

Solution 2
The Jacket Engine for MATLAB®

1. Jacket is the GPU Engine for MATLAB. Jacket enables standard MATLAB code to run on any NVIDIA CUDA capable GPU, from the GeForce 8400 to the Tesla C1060.
2. Jacket introduces new data types to MATLAB which let you move your data and computations to the GPU.
3. **Example MATLAB code:**

```
>> G = gdouble( C );          % Create a GPU matrix
>> G = fft( G );              % Perform a GPU FFT
>> G = G * G;                 % GPU Matrix Multiply
>> C = double( G );           % Bring back to CPU
```

4. It just change the data type and you can start tapping into the GPU's tremendous power .
5. Its **not free,** have a 15 days free trial. It is **available for purchase** through the link *http://www.accelereyes.com/*

Solution 3
GPUmat

1. A new freeware tool called GPUmat allows user to run the Matlab code directly on a CUDA-enabled GPU.
2. The execution is transparent to the user.
3. It enables up to a 40x speedup.
4. It's completely free to download and use.
5. Detailed information available at
   http://www.gp-you.org/index.php?option=com_content&view=article&id=46&Itemid=54
6. User modules available for download at
   http://sourceforge.net/projects/gpumatmodules/

CUDA enabled Products

1. A list of CUDA enabled products can be obtained from

- *http://www.nvidia.co.in/object/cuda_learn_products_in.html*

2. Broad Categorization :

a. GeForce : Gaming, Video processing tasks, digital photography, standard applications.

b. Quadro : Provides a full range of robust professional solutions, spanning from graphics boards, visual computing systems, to software development tools that have become the standard for professional visualization environments.

c. Tesla : High performance computing, Cluster level computing.

References

- Links provided in presentation
- *Accelerating MATLAB with CUDA*, Massimiliano Fatica (mfatica@nvidia.com), NVIDIA,Won-Ki Jeong (wkjeong@cs.utah.edu), University of Utah

Thank you!