FB 08 Informatik
Lehrstuhl f. Parallele und Verteilte Architekturen

Prof. Dr. Bertil Schmidt
Dr. Tuan-Tu Tran

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

CUDA Practical – Winter Term 2013/14

CUDA Projects

Select one of the following projects to implement. You are required to give a 20 minutes presentation on the last day of the Praktikum (start from 1 pm of the Thursday, 17 April 2014). Your presentation must include following contents:

- Description of the problem and specific algorithm.
- CUDA parallelization: strategy, CUDA feature.
- Performance evaluation: compare your implementation to existing software or simple CPU code.
- Demo.

BASIC

Project 1: The Julia and Mandelbrot set (fractal)
Write a CUDA application to generate the Julia and/or the Mandelbrot set in the specialized region in the complex number space. Present the set in the 2D plane, with the real and the imaginary part of an complex number are the dimensions. Serial algorithms can be found in:
[1] **Lode Vandevenne**, "Julia and Mandelbrot Sets", http://lodev.org/cgtutor/juliamandelbrot.html

Project 2: CUDA Image Processing Library
Image processing is a natural for parallel processing in CUDA because pixels can be mapped directly to threads. Write a CUDA library to perform various image processing algorithm on given images. Some algorithm to consider: Sobel edge detection, Box Blur, Convolution …, which can be found in [1]:
[1] Lode Vandevenne, "Image Filtering", http://lodev.org/cgtutor/filtering.html

Project 3: CUDA Bruteforcer
In Cryptography, a brute-force attack, or exhaustive key search, is a strategy that can, in theory, be used against any encrypted data. Such an attack might be utilized when it is not possible to take advantage of other weaknesses in an encryption system that would make the task easier. It involves systematically checking all possible keys until the correct key is found. In the worst case, this would involve traversing the entire search space. Implement a CUDA Bruteforcer for your chosen cryptography algorithms. For example:

- Cryptographic hash function: MD5, SHA1
- Encryption algorithm: DES, AES

Project 4: N-body simulation
In [1], the authors presented ideas and pseudo-codes to efficiently map the all pair N-Body simulation onto the GPUs. Implement and evaluate their algorithm with CUDA.
[1] **Lars Nyland, Mark Harris and Jan Prins**, "Fast N-Body Simulation with CUDA", GPU Gems 3, http://http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html

Project 5. Efficient matrix-matrix Multiplication on GPU
Implement and evaluate the approaches to efficiently map the matrix-matrix multiplication algorithm onto the GPUs which are presented in [1]:
[1] **José M. Cecilia, José M. García, and Manuel Ujaldon.** "The GPU on the Matrix-Matrix Multiply: Performance Study and Contributions", PARCO, volume 19 of Advances in Parallel Computing, page 331-340. IOS Press, (2009), https://webs.um.es/jmgarcia/miwiki/lib/exe/fetch.php?id=pubs&cache=cache&media=parco09.pdf

ADVANCED

Project 6: Parallel Sorting
Sorting is one of the most important process in Information Technology and is a target for various studies and researches. Some sorting algorithms which can be efficiently parallelized are: quick sort, radix sort, bitonic sort, merge sort, etc. Take at least 1 algorithm, implement and evaluate it on GPU. Some ideas and pseudo-codes can be found in this document:
[1] Nadathur Satish, Mark Harris, Michael Garland, "Designing Efficient Sorting Algorithms for Manycore GPUs" , in Proc. IEEE International Symposium on Parallel & Distributed Processing, May 2009. http://mgarland.org/files/papers/gpusort-ipdps09.pdf

Project 7. Graph algorithm:
In [1], the authors present the ideas and the pseudo-codes of mapping 3 graph algorithms onto the GPU:

- Breadth First Search
- Single Source Shortest Path
- All pairs shortest path

Take at least one algorithm, implement and evaluate it.
[1] **P. Harris and P.J.Narayanan**, "Accelerating Large Graph Algorithm on the GPU using CUDA", HiPC'07 Proceedings of the 14th international conference on High performance computing
Pages 197-208, http://cvit.iiit.ac.in/papers/Pawan07accelerating.pdf

Project 8. Sparse Matrix algorithm.
Sparse Matrix is a matrix that the majority of its elements is 0. We can use some space-efficient data structure to store the matrix. Thus, it needs to use the specialized algorithms to work on these data structure. Take at least 1 algorithm and data structure presented in [1], implement and evaluate it on the GPU.
[1]. **Nathan Bell and Michael Garland, "**Sparse Matrix Vector Multiplication**", NVIDIA Technical Report NVR-2008-004, December 2008,** http://www.nvidia.com/docs/IO/66889/nvr-2008-004.pdf