RISE:2017
RESEARCH, INNOVATION AND SCHOLARSHIP EXPO
Undergraduate
Category: Engineering and Technology
Degree Seeking: BS in Computer Science and Mathematics
Abstract ID# 1548
Northeastern University

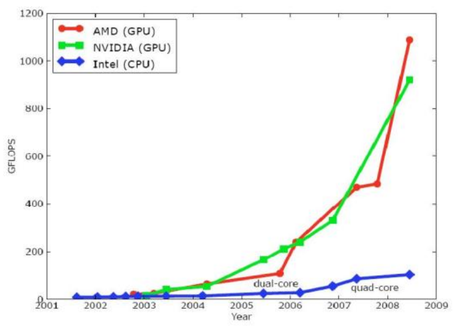
Dynamic Parallelism Performance Evaluation on Image Segmentation Algorithms

Daniel Goldstein, Julian Gutierrez, goldstein.d@husky.neu.edu
Northeastern University Computer Architecture Research Laboratory, *Northeastern University*, Boston, MA

Opportunity

Abstract

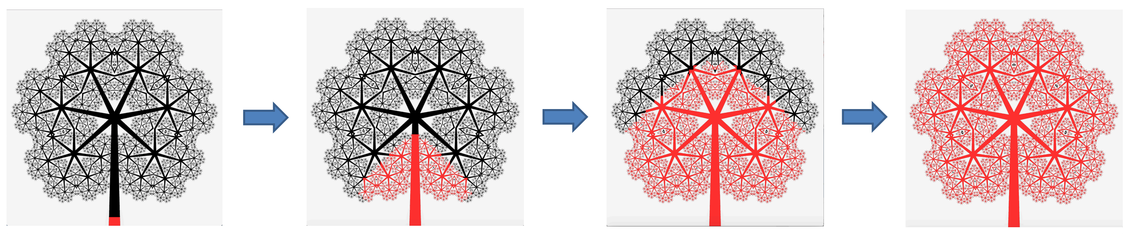
- CPU speeds are reaching a plateau
- Parallel computing through GPUs is a method for achieving faster computing speeds
- We compare CPU and GPU parallelization to analyze impact of dynamic parallelism in image-processing algorithms



GPU Processing Capabilities vs CPU

Goal

- Compare dynamic parallelism in CUDA to a split CPU-GPU parallelization to see which method is fastest for a Level-Set Segmentation algorithm

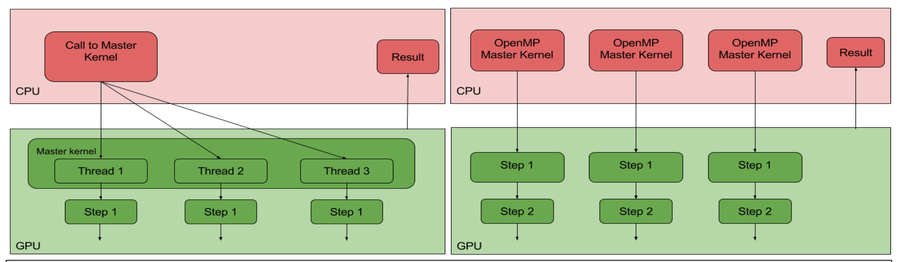


Evolution of a fractal test image for the Level-Set Segmentation Algorithm

Approach

Algorithm Modifications

- Largest factors in performance are: size of the image, number of objects, and the complexity of the objects
- Original algorithm employ's *dynamic parallelism*
  - A master kernel on the GPU calls subsequent kernels that perform the curve evolution (shown on left)
- Reconfigure the master kernel so that it runs on the CPU, entirely removing dynamic parallelism
  - Kernel is constructed into a for-loop on the CPU side which is parallelized using the OpenMP runtime (shown on right)
  - No longer any CUDA kernels spawning other kernels
- Adapt master kernel grid size to compare using one block with one thread per object vs one block per object with one thread each

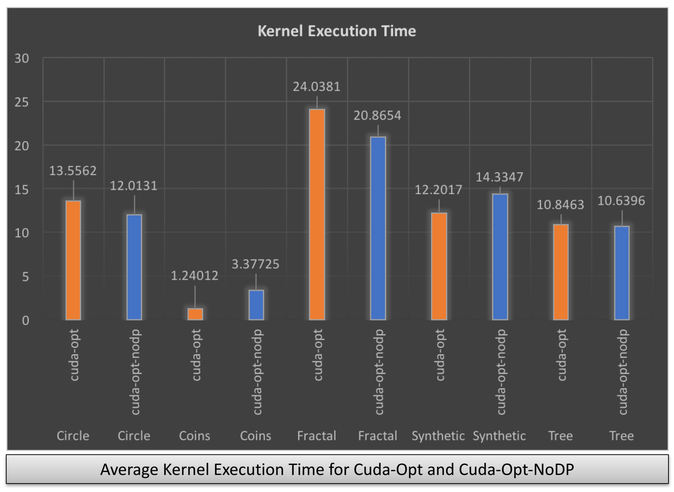


Master Kernel Configuration on the Dynamic Parallelism and CPU-GPU Algorithms

Data/Results

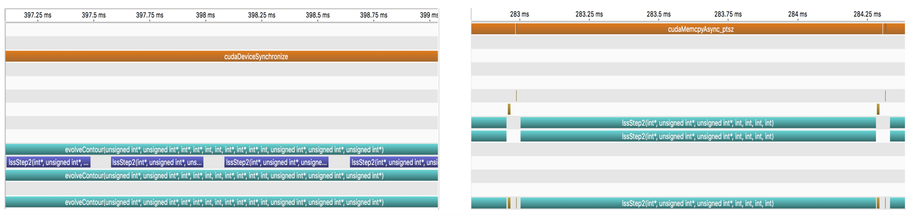
Kernel Execution Analysis

- Twenty trials were run for each sample image for the original and modified algorithm



Average Kernel Execution Time for Cuda-Opt and Cuda-Opt-NoDP

- Cuda-opt performed better on images with multiple objects, e.g. coins and synthetic, while CPU-GPU implementation was faster for single-object images
- For multiple-object images, CPU-GPU implementation experienced between 17%-172% slower times
- Large improvement in fractal image is largely due to the overhead required by the large number of *atomic operations* necessary in the original algorithm that were removed in the modified algorithm
- Similar behavior observed when run on an NVIDIA Pascal GTX 1080



NVPROF Timeline Between Step 2 Iterations in DP and CPU-GPU Algorithm for Fractal Test Image

Impact

Conclusion

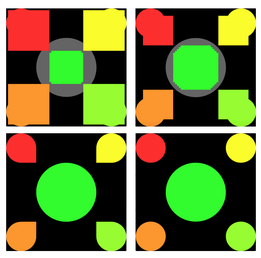
- Dynamic parallelism proved to be the most effective for images of multiple or an unknown number of objects
- CPU-GPU implementation would be best for more intricate images of one object
- Results can be extrapolated to other applications run on parallel hardware
- Further research would need to be done to conclude if these results remain consistent across different GPU architectures

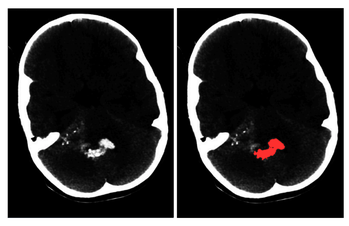
Value Proposition

- This project is unique because it
  - combines CUDA parallelism and OpenMP, two different environments that serve similar purposes
  - found optimized paths to parallelization, an underexplored field of comparison
- This experiment solves the problem of finding the best implementations to parallelize algorithms

Image-Segmentation Applications

- Cancer radiation theory
- Medical image processing
- Border security
- Self-driving cars





References: Gutierrez, J., Nina-Paravecino, F., & Kaeli, D. (2016, November). A fast level-set segmentation algorithm for image processing designed for parallel architectures. In *Proceedings of the Sixth Workshop on Irregular Applications: Architectures and Algorithms* (pp. 66-69). IEEE Press.
Lambert, F. "Tesla Is About To Increase Its Lead In Semi-Autonomous Driving W/ 'Tesla Vision': Computer Vision Based On NVIDIA'S Parallel Computing". *Electrek*. N.p., 2017. Web. 20 Mar. 2017.
Soulan, S. "In-Line Etching Process Control Using Dynamic Scatterometry". *Research Gate*. N.p., 2007. Web. 18 Mar. 2017.