Efficient Matching of GPU Kernel Subgraphs

Robert Lim
University of Oregon

Background

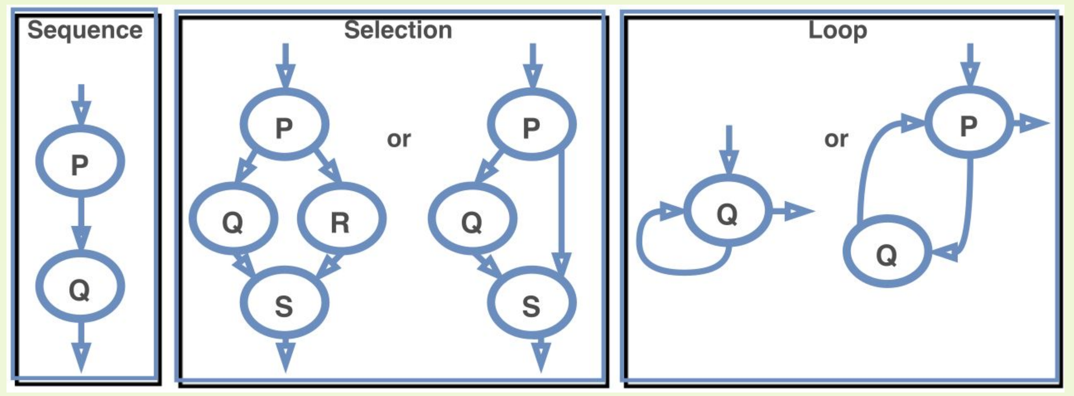


Figure 1. Base constructs of programs: sequence, selection and loop

Structured programming consists of base constructs that represent how programs are written. When optimizing programs, compilers typically operate on the intermediate representation (IR) of a control flow graph (CFG), which is derived from program source code analysis and represents basic blocks of instructions (nodes) and control flow paths (edges) in the graph. Thus, the overall program structure is captured in the CFG and the IR abstracts machine-specific intrinsics that the compiler ultimately translates to machine code. In particular, compilers can benefit from prior knowledge of optimizations that may be effective for specific CFG structures.

Objectives

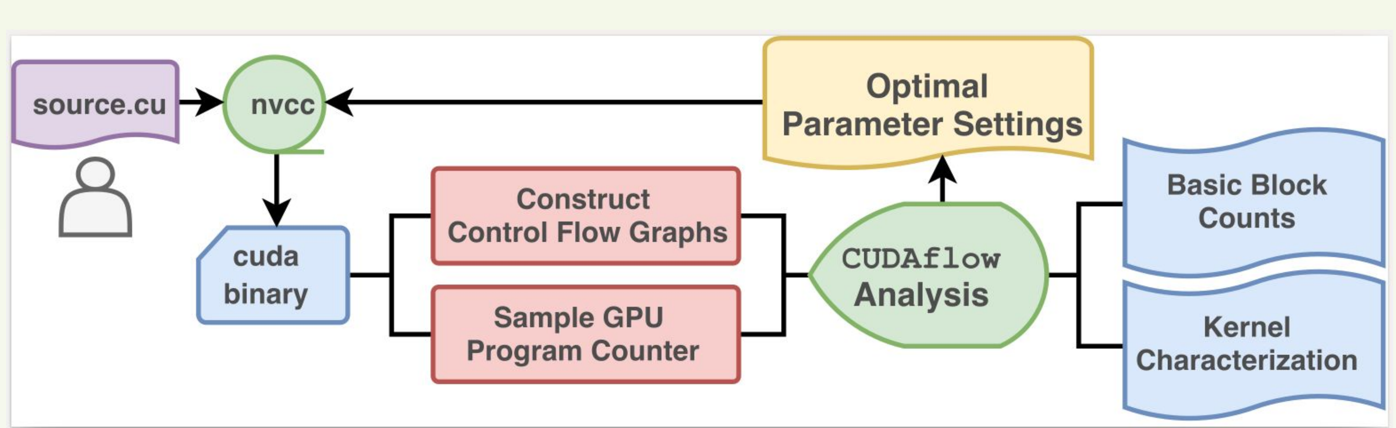
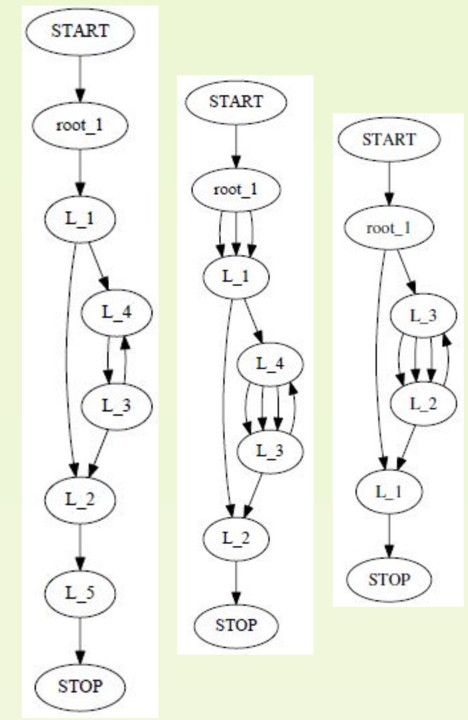


Figure 2. Overview of proposed methodology

- *Systematic process to construct control flow graphs for GPU kernels*
- *Techniques to perform subgraph matching on various kernel CFGs and GPUs*
- *Approaches to reveal thread divergence behavior based on CFG properties*

Methodology



$$\begin{array}{cccccc} R_1 & L_1 & L_4 & L_3 & L_2 & L_5 \end{array}$$
$$\begin{bmatrix} .21 & - & - & - & - & - \\ 0 & .04 & - & - & - & - \\ 0 & .04 & .38 & - & - & - \\ 0 & 0 & 0 & .08 & - & - \\ 0 & 0 & 0 & 0 & .21 & - \\ 0 & 0 & 0 & 0 & .02 & 0 \end{bmatrix}$$

$$\begin{array}{cccc} R_1 & L_3 & L_2 & L_1 \end{array}$$
$$\begin{bmatrix} .30 & - & - & - \\ 0 & .51 & - & - \\ 0 & 0 & 0 & - \\ 0 & 0 & 0 & .21 \end{bmatrix}$$

Figure 3. Control flow graphs (left) and transition probability matrices (right) for Pathfinder, comparing Kepler, Maxwell and Pascal GPUs.

- *Transition probability matrices calculated for each kernel subgraph*
- *Spline interpolation employed to scale transition matrix before performing pairwise comparisons*
- *Affinity scores for CFGs (S_1 and S_2 for G_1 and G_2) matched via similarity measures (IsoRank, Euclidean)*
- *Methodology evaluated on 3 GPUs (Kepler, Maxwell, Pascal)*

Results

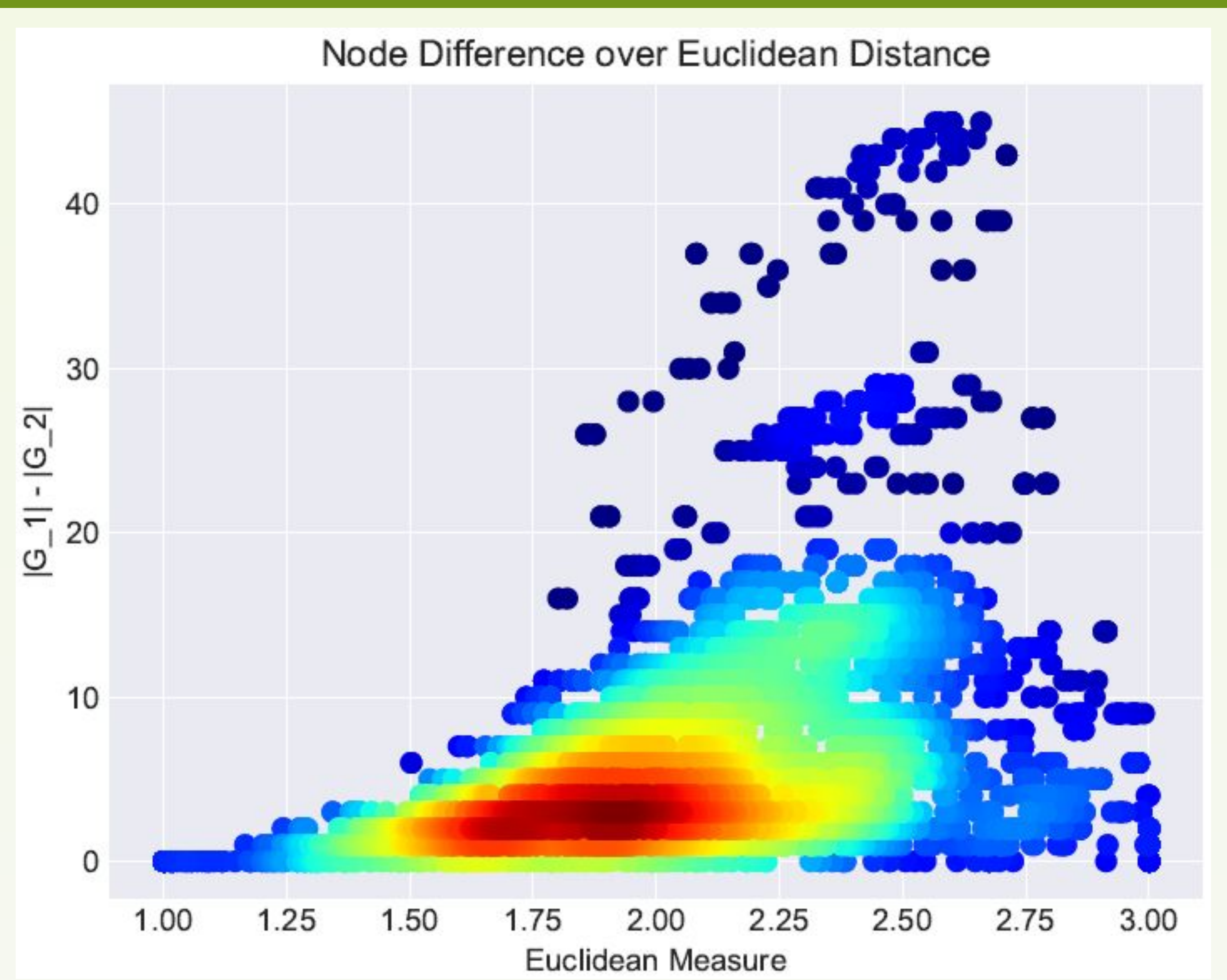


Figure 4. Differences in vertices between two graphs over Euclidean for all GPU kernel combinations, with color as frequency

Differences in vertices |V| for162 CFG kernel pairs over Euclidean measure (application, architecture, kernel)
- *Most matched CFGs had similarity score between 1.5 and 2.2, with size differences under 10 vertices*
- *As differences increased, similarity matching degraded due to interpolating missing information (expected)*

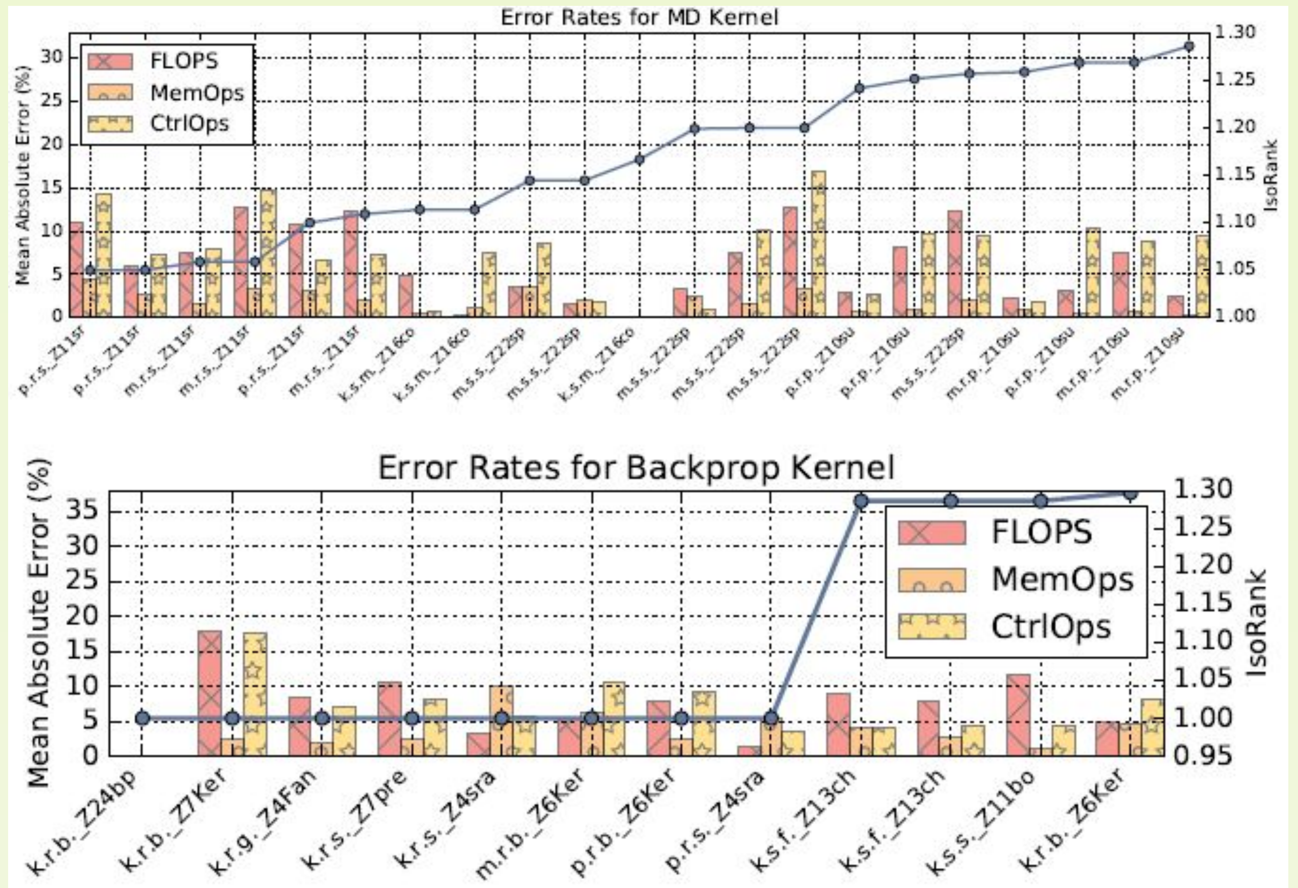


Figure 5. Error rates when estimating dynamic instruction mixes from static for select kernels.

Instruction mix estimation error rates for MD and Backprop kernels as a function of matched kernels, with IsoRank scores between 1.00 to 1.30
- *Subgraph matching for arbitrary kernels with IsoRank and instruction mixes within a 8% margin of error*

Discussion

- *Demonstrated control-flow-based methodology for analyzing performance of CUDA applications*
- *Combined static binary analysis with dynamic profiling to characterize kernel intensity (memory, compute)*
- *Identified similarities of new implementations through subgraph matching*

Future Work

- *Incorporate memory reuse distance statistics of a kernel to help optimize memory subsystem and expose compute/memory overlaps*
- *Characterize deep learning workloads to optimize placement of tasks on multi-node multi-GPU setups*

Acknowledgements

- *NVIDIA for providing access to PSG Clusters*
- *American Society for Engineering Education (ASEE)*

Corresponding Authors:
Boyana Norris
Allen Malony