FPGA-Accelerated Ripples

Reece Neff, Marco Minutoli, Antonino Tumeo, Michela Becchi {reece.neff, marco.minutoli, antonino.tumeo}@pnnl.gov, {mbecchi}@ncsu.edu

Introduction

- Influence Maximization is widely used in many fields to estimate the most influential set of nodes in a graph.
- The complexity and long run time opens the door for acceleration.
- Has been done in cuRipples[1] with GPU, but GPUs consume a lot of power. FPGAs are more power efficient, so they can potentially reduce power consumption and therefore operating costs.
  - This work builds off cuRipples to add FPGA acceleration

IMM Algorithm

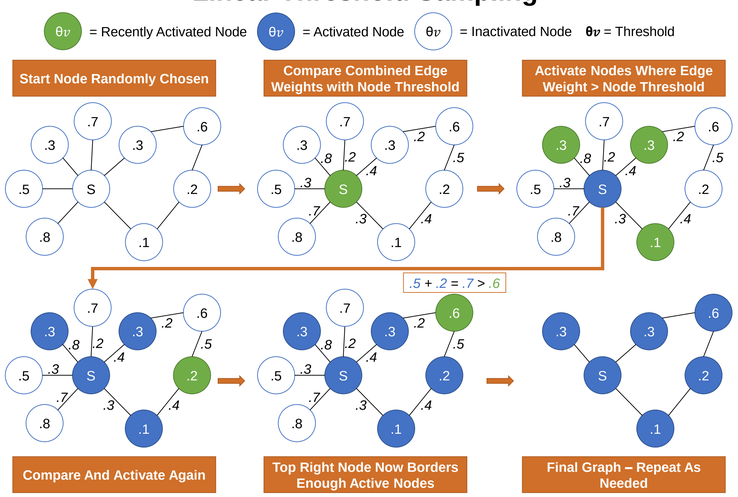
- The algorithm is separated into three main parts as shown in the pseudocode
- **Estimate Theta**: Estimate the number of graphs to be generated by the Sampling stage to achieve a good IMM estimation.

```
Input: G, k, ε
Output: S

begin:
  {R,θ}←EstimateTheta(G,k,ε)
  R←Sample(G,θ-|R|,R)
  S←SelectSeeds(G,k,R)
end
```

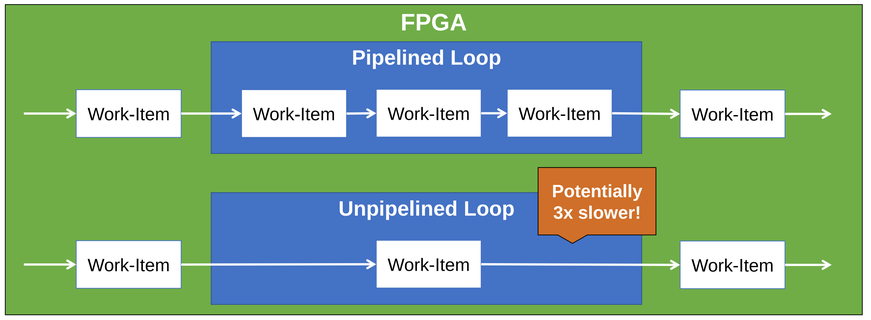
- **Sampling:** Generate several graphs determined by Estimate Theta using a given diffusion model. For this work, we use the Linear Threshold Model. *This stage is the target for FPGA acceleration.*
- **Seed Selection:** Find the node with the most occurrences in all graphs from the Sampling stage, and record that as the most influential seed. Remove all graphs containing this node and repeat the calculation until k nodes have been selected.

Linear Threshold Sampling



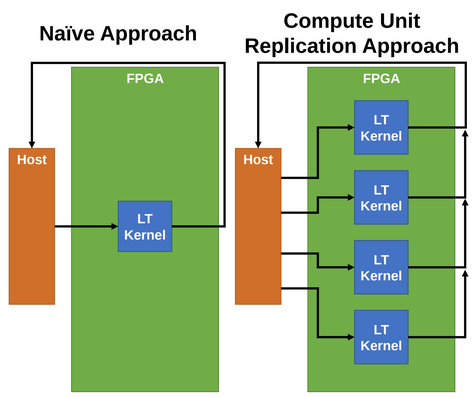
Challenges

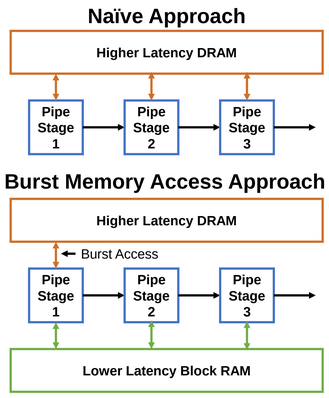
- Large, complex loops keep largest section of the kernel from being pipelined
- Irregular memory accesses results in multiple accesses to high latency DRAM



Optimizations

- Created four compute units to parallelize the kernel, increasing throughput
- Added burst memory accesses to reduce the amount of sporadic DRAM accesses
- Unrolled inner loops where possible to reduce loop latency and increase throughput

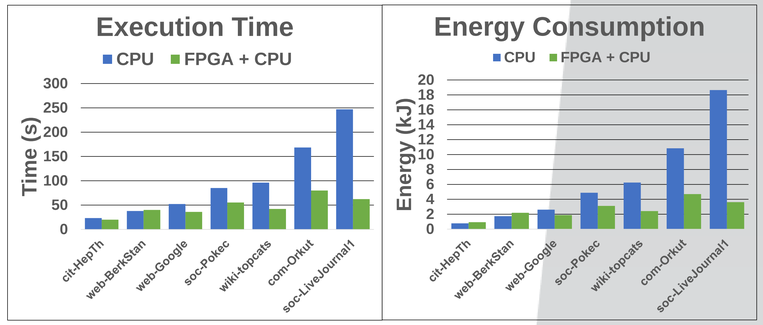




Experimental Setup

Component Type	Component
CPU	Intel Xeon E5-2367 w/ 8x32GB 2400MHz DDR4 RAM
FPGA	Xilinx Alveo U250 w/ 4x16GB DDR4 RAM (1x16GB used)
Input Graph	Selections from SNAP[2]
HLS Tool	Vitis Unified Software Platform 2020.2 (OpenCL to FPGA)

Results



Resource Utilization (% of total available)

Kernel	FFs	LUTs	BRAMs
Graph Initialization	5,763 (0.17%)	4,544 (0.28%)	4 (0.16%)
Linear Threshold	43,345 (1.32%)	32,323 (1.99%)	24 (0.95%)
All Kernels Combined	179,143 (5.44%)	133,836 (8.25%)	100 (3.96%)

Observations

- Up to **3.96x** speedup (soc-LiveJournal1 graph) with a positive correlation in speedup and the number of nodes in the graph.
  - In smaller graphs, the overhead for the FPGA diminishes the potential speedup from acceleration.
- Up to **5.15x** less energy consumption showing a similar positive correlation with the number of nodes.
- Each kernel uses few resources, so there is potential for even more parallelization as there are only four Linear Threshold kernels working in parallel.

Conclusion and Future Work

- We were able to achieve considerable speedup and lower power consumption on an FPGA-accelerated implementation of the IMM algorithm compared to a CPU implementation.
  - Optimizations from parallelizing, creating a more efficient dataflow, and loop unrolling helped increase performance.
- Parallelizing by eight compute units (as opposed to four) did not show considerable speedup due to DRAM bandwidth limitations, but there is future work in expanding to multiple Super Logic Regions since each have their own DRAM.
- Seed Selection is another stage that could benefit from FPGA acceleration due to its high execution time.

Contributions

- Implemented FPGA Acceleration for the IMM algorithm
- Explored the performance speedup of up to **3.96x**
- Explore the energy savings of up to **5.15x**

References

[1] Marco Minutoli, Maurizio Drocco, Mahantesh Halappanavar, Antonino Tumeo, and Ananth Kalyanaraman. 2020. CuRipples: Influence Maximization on Multi-GPU Systems. *InProceedings of the 34th ACM International Conference on Super-computing(Barcelona, Spain)(ICS '20)*. Association for Computing Machinery, New York, NY, USA, Article 12, 11 pages. https://doi.org/10.1145/3392717.3392750

[2] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data

U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

For additional information, contact:

Reece Neff | (509) 372-6591 | reece.neff@pnnl.gov

10/12/2021 | PNNL-SA-165848

Pacific Northwest NATIONAL LABORATORY

www.pnnl.gov