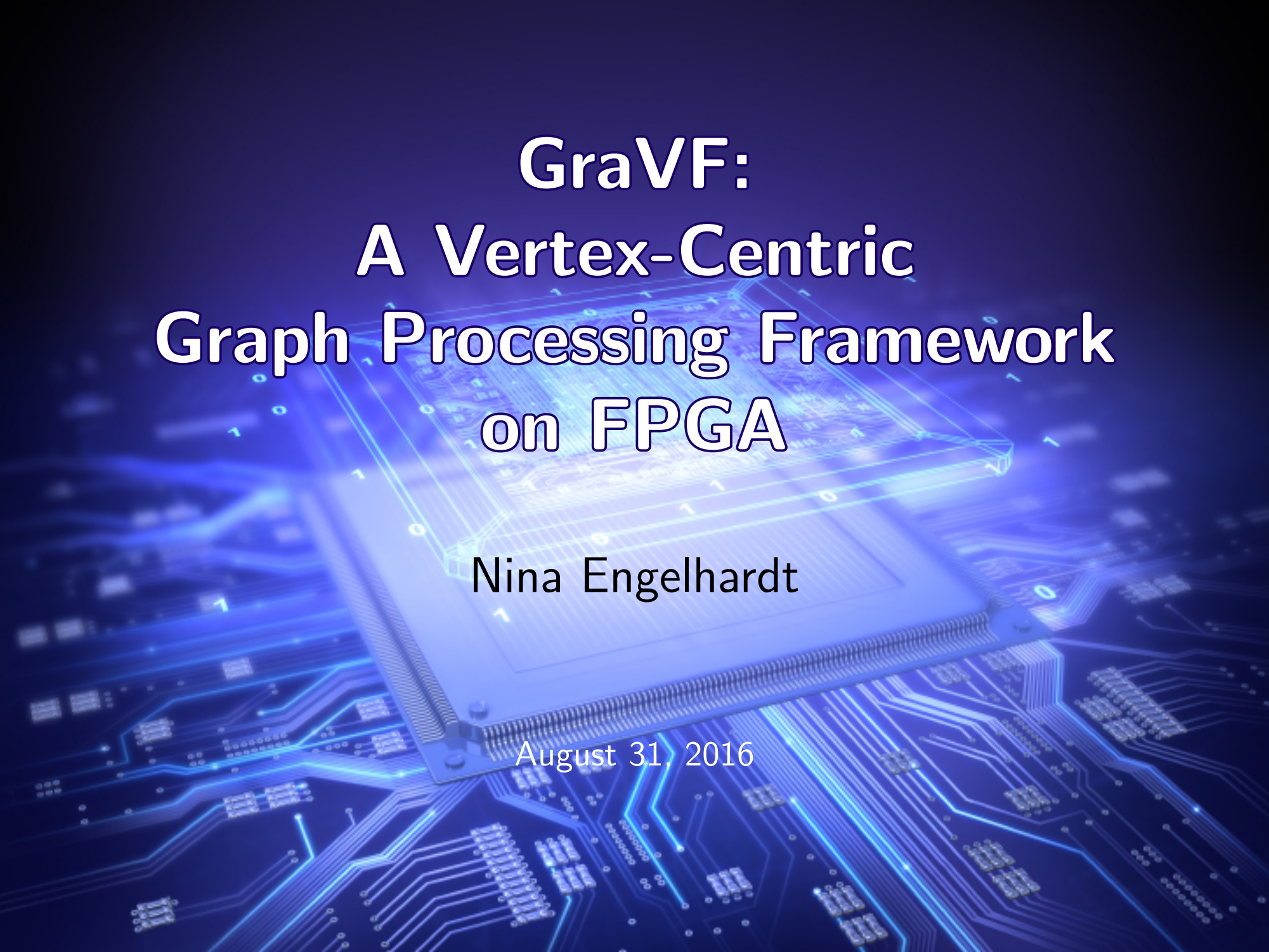
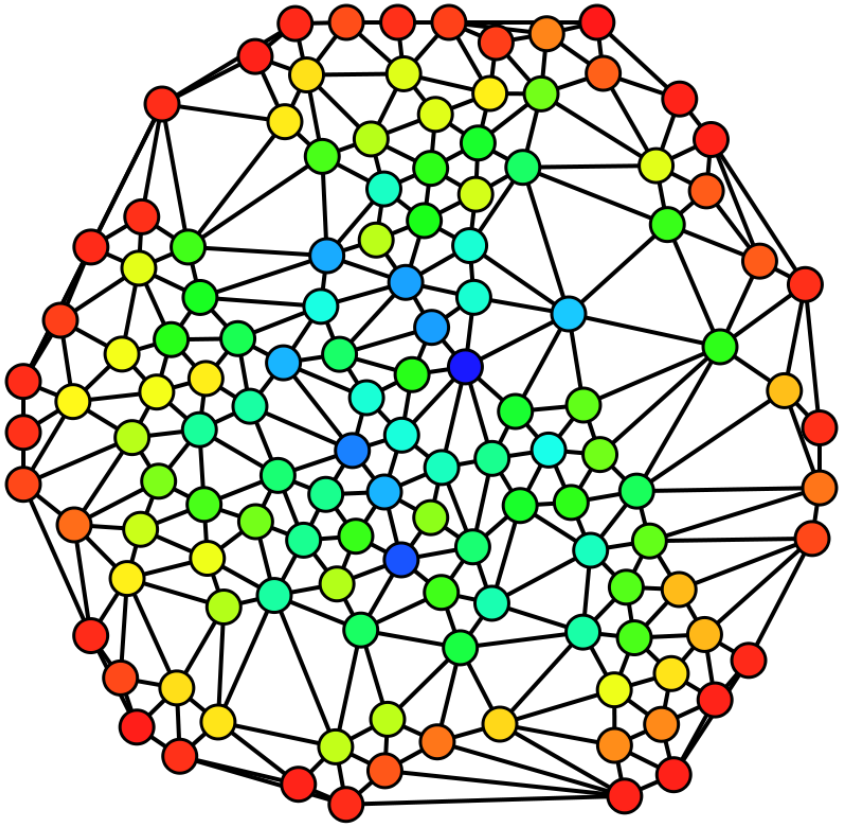


GraVF: A Vertex-Centric Graph Processing Framework on FPGA

Nina Engelhardt

August 31, 2016

Graphs and Graph Traversal Algorithms



1

Vertex-centric Programming Model:

- From POV of an individual vertex
- Receive messages from neighbors, calculate, send messages to neighbors
- Global barrier after each iteration

Synchronous Vertex-centric Programming Model

Advantages:

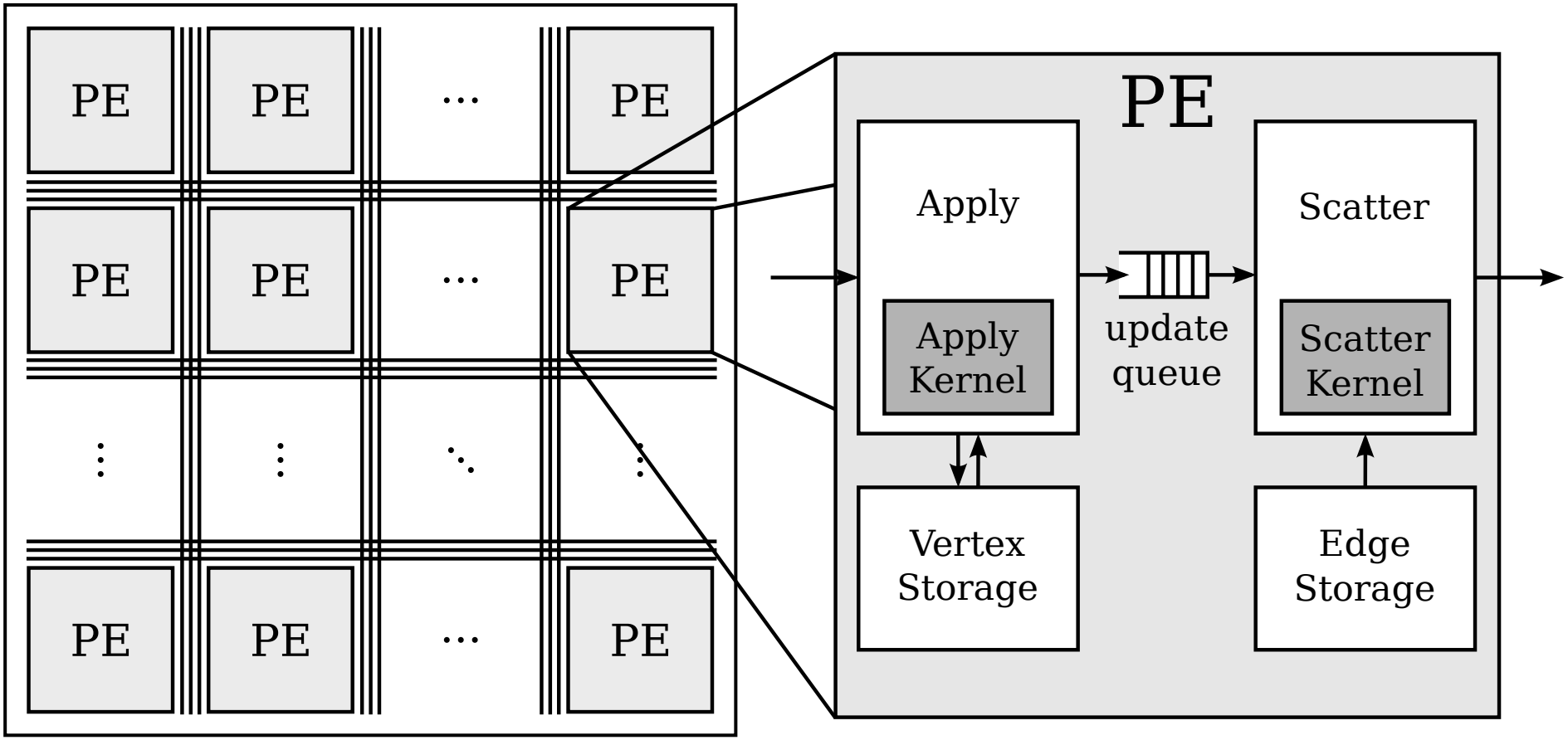
- naturally distributed
- only very short kernel to write

Challenges:

- barrier means stragglers can unduly delay whole computation
- have to store all messages from one superstep to the next

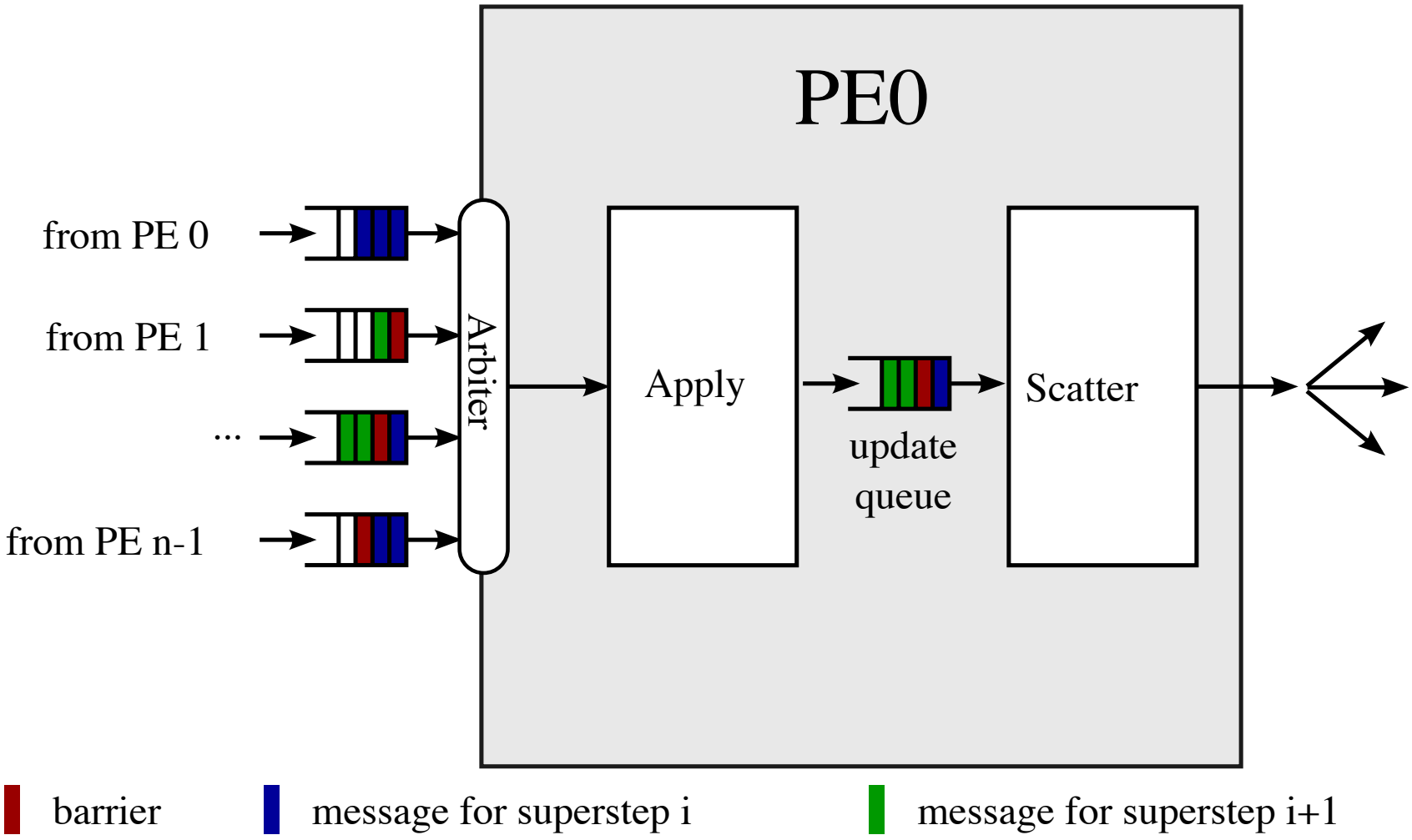
$\rightarrow$ *Floating barrier* [1]: PE broadcasts barrier message when it finishes superstep, allowing overlapping of supersteps

Architecture modifications



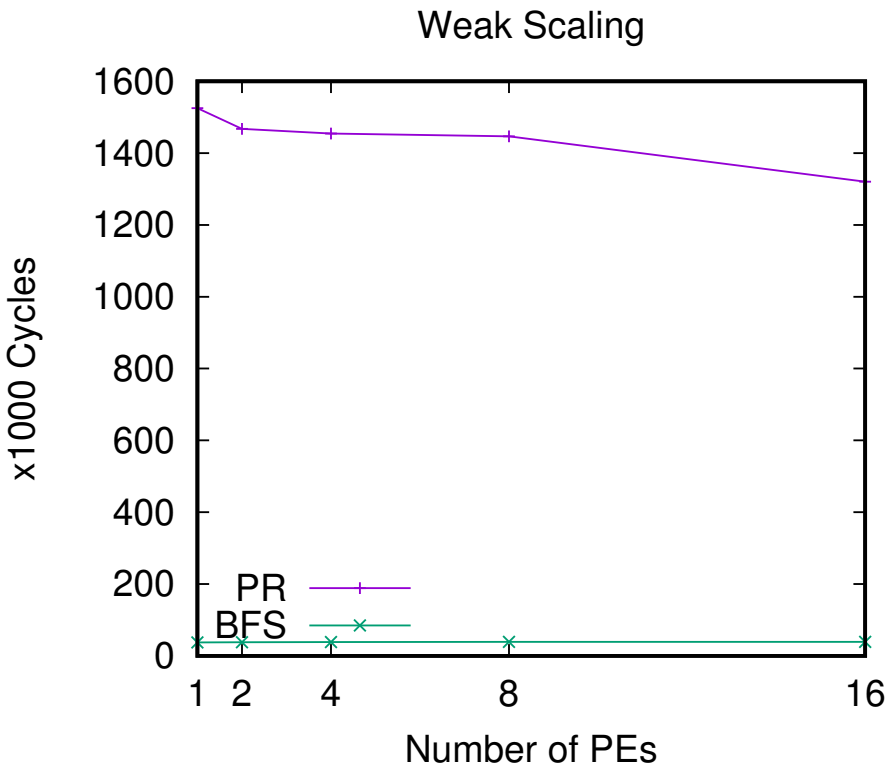
- Split Vertex Kernel in 2 phases: Apply and Scatter
- Everything is pipelined: free to move registers

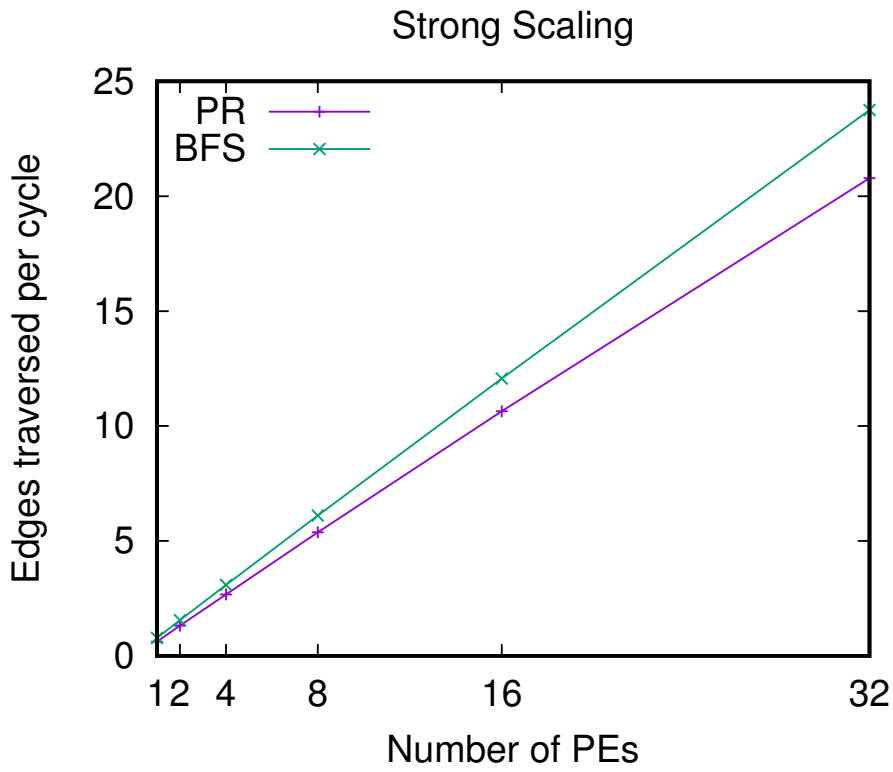
Barrier Synchronization



- Apply ahead of Scatter: messages processed immediately
- update queue only $2|V|$ entries instead of $|V|^2$

Results





3-3.5 GTEPS: comparable to hand-implemented solutions

Future Work

- Extend communication to multiple FPGA boards
- Include data transfers: direct SSD access from FPGA

Thank you for listening

Works Cited I

[1] Betweenness Centrality; By Claudio Rocchini, CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=1988980


Qingbo Wang, Weirong Jiang, Yinglong Xia, and V. Prasanna.
A message-passing multi-softcore architecture on FPGA for breadth-first search.
In *Field-Programmable Technology (FPT), 2010 International Conference on*, pages 70–77, Dec 2010.