Layer-Parallel Training with GPU Concurrency of Deep Residual Networks via Nonlinear Multigrid

September 24, 2020

Andrew C. Kirby

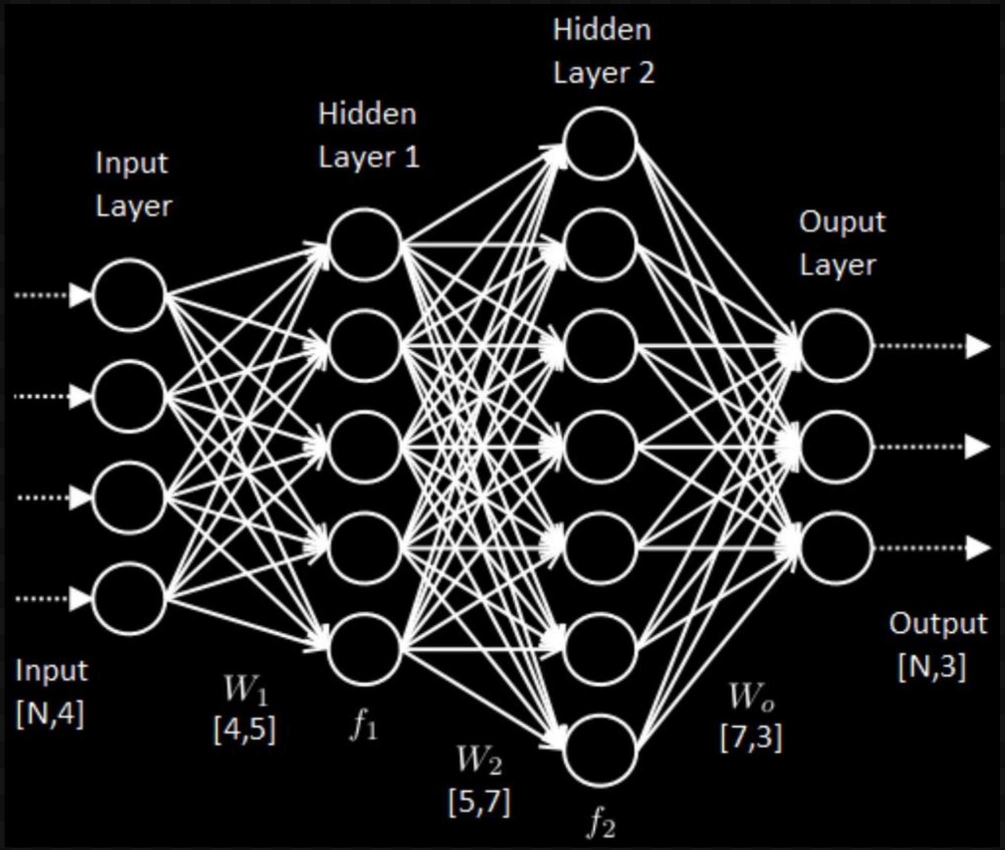
Siddharth Samsi, Michael Jones, Albert Reuther
Jeremy Kepner, Vijay Gadepally

MIT Lincoln Laboratory Supercomputing Center

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Motivation

- Information travels sequentially through network
  - Forward propagation sets the neural states
  - Back propagation updates the weights and bias
  - Amdahl's Law
    - Theoretical speedup limited by serial execution
- Goal: minimize loss by updating the weight coefficients and bias vectors iteratively
  - Updates to unknowns are *approximate*
    - Optimization opportunities for accelerating training



Source: https://miro.medium.com/max/2500/1*ZB6H4HuF58VcMOWbdpcRxQ.png

Deep Residual Networks

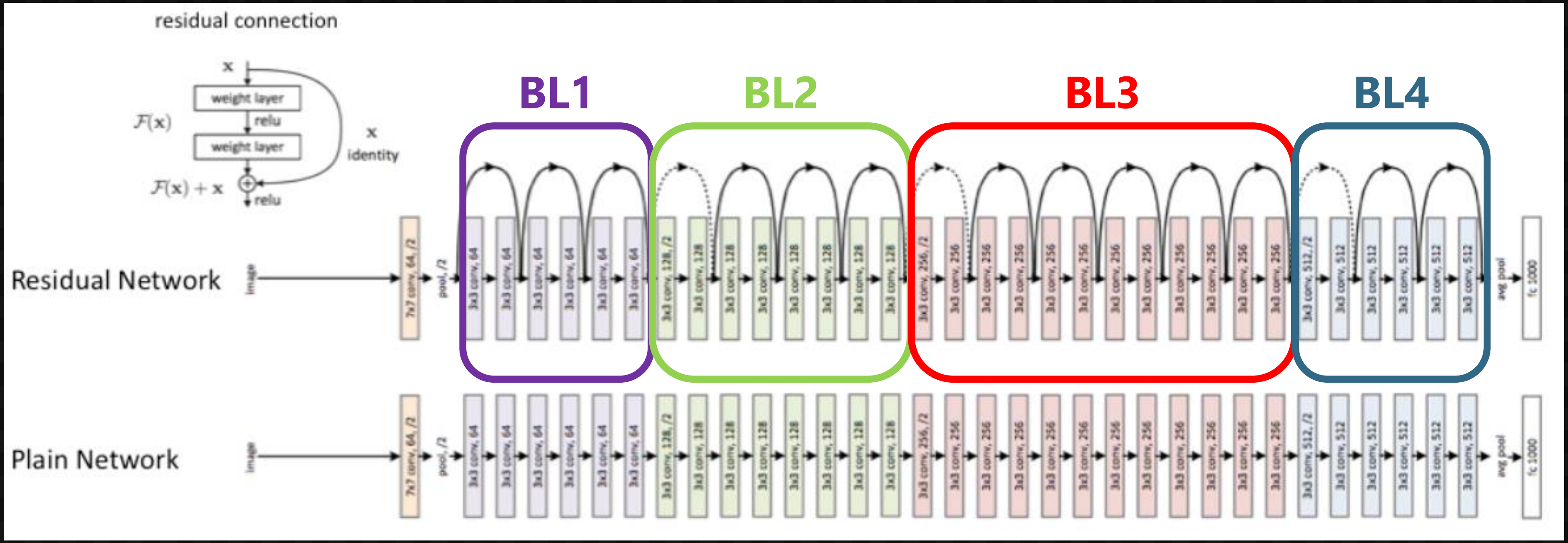
$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n} + hF\left(\boldsymbol{u}^{n}; \boldsymbol{\theta}^{n}\right), \quad \text{for } n = 0, \ldots, N-1$$

Parallelization Approaches

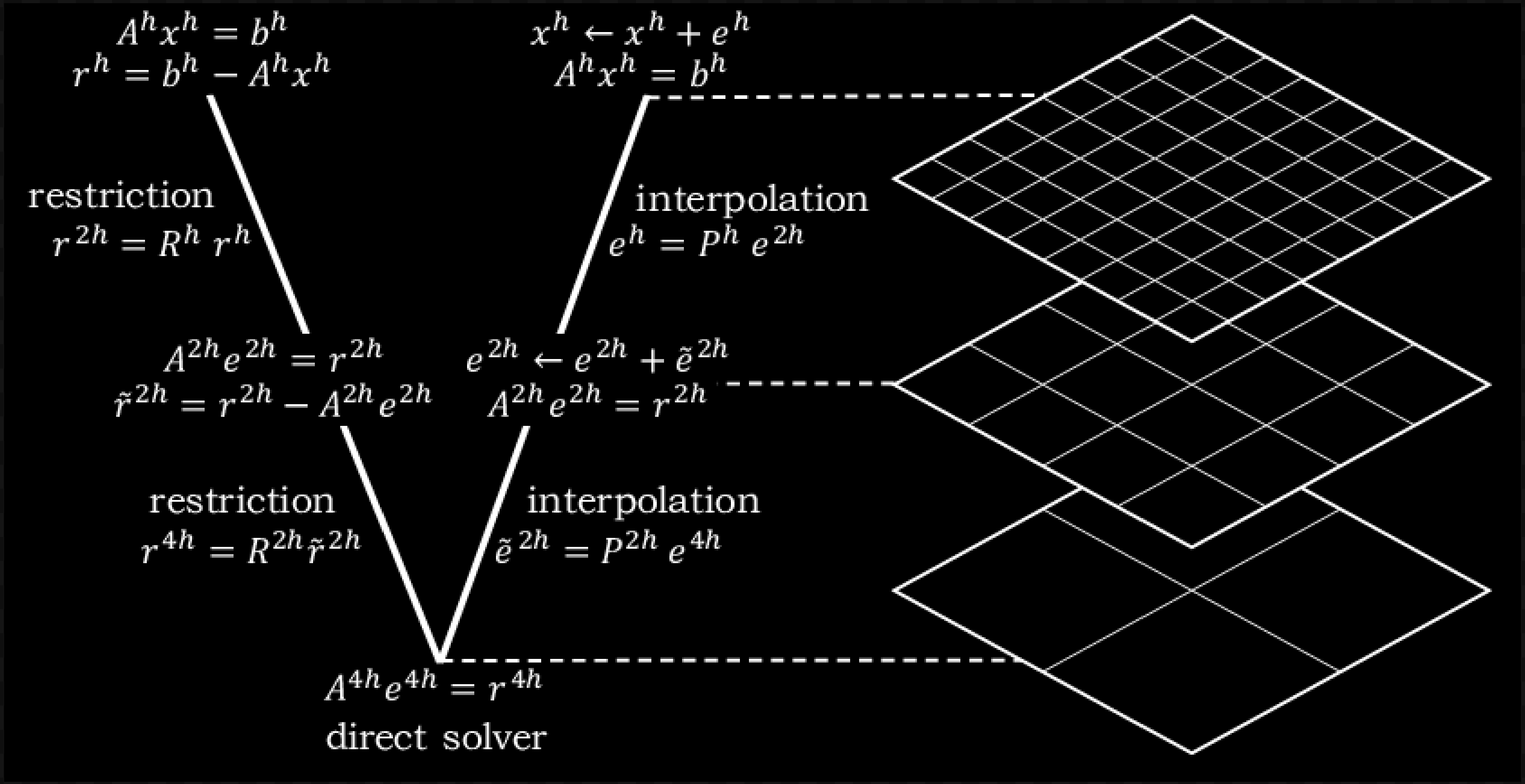
- Data-Parallel (easy – e.g. Horovod)
- Model-Parallel (hard – not *model-partitioned* serial propagation of data)

True Model Parallelism via Iterative Approach: Solve Each Block in Parallel

- Combine with Data-Parallel Techniques for multiplicative parallelization
  - e.g. DP = 4-way parallelism, MP = 2-way parallelism, Total = 8-way parallelism



Multigrid

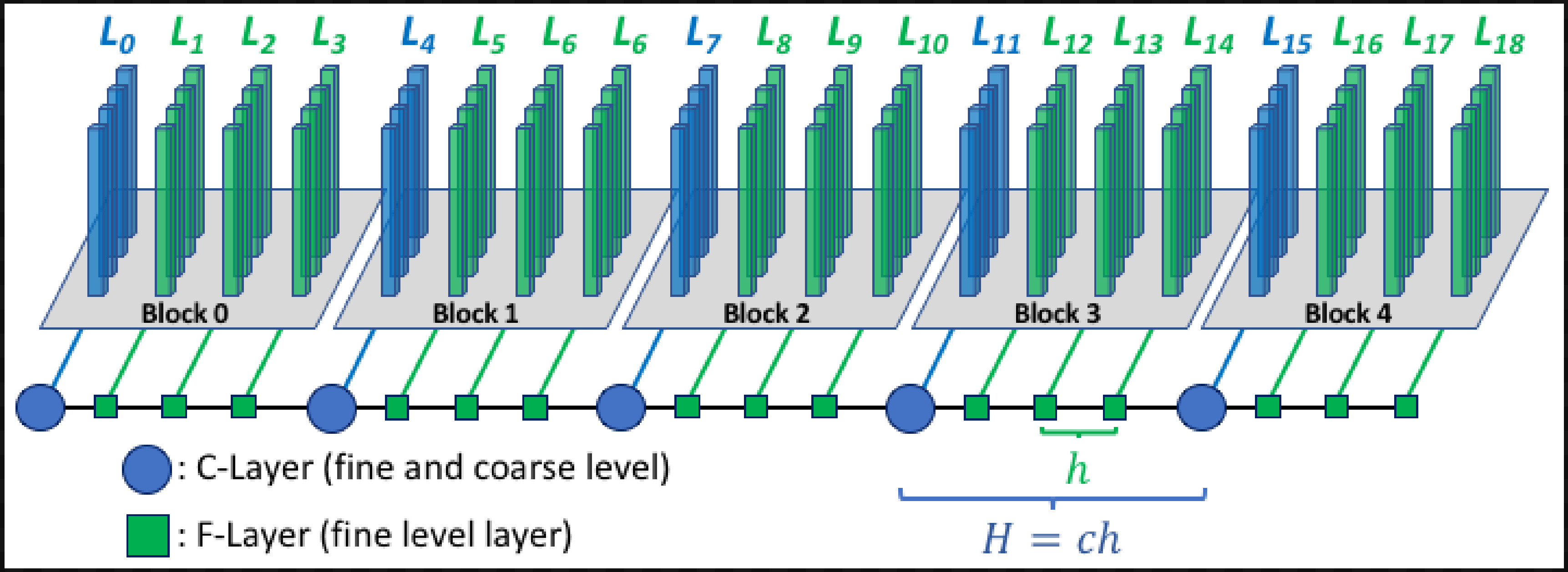


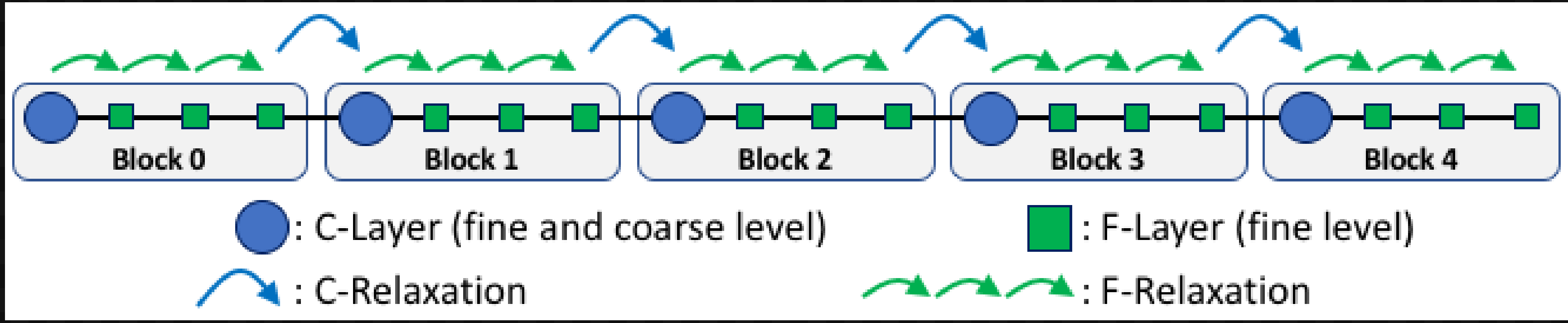
[Reference] S. Gunther, et. al, "Layer-Parallel Training of Deep Residual Neural Networks", SIAM.

Source: https://www.researchgate.net/figure/Illustration-of-the-multigrid-V-cycle_fig2_328599327

Network Partitioning

- Iterative Solution Procedure
  - Propagate guess in block and 1st neighbor layer, restrict, solve coarse solution, update





Implementation & Algorithm Verification

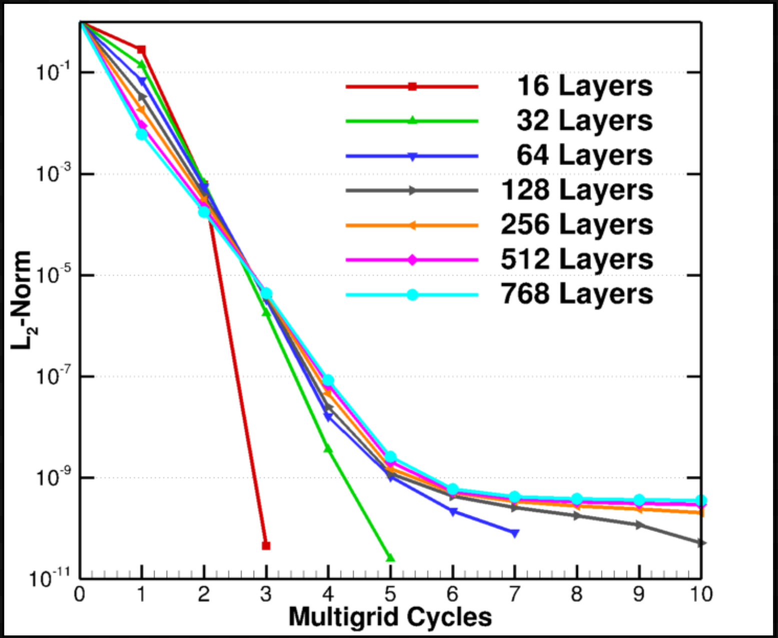
Implementation

- Forward Propagation Only (Back Propagation possible)
- C++: Wrap CuDNN kernels
  - Enabled CUDA Streams for asynchronous execution
- MPI:
  - Point-to-Point Communications (i.e. layer to layer comm.)
  - No Collective Communication needed (e.g. no use of NCCL)
- TX-GAIA Supercomputer: 32 GB NVIDIA V100s

Algorithm 1: ResNet Multigrid FAS Scheme

1. FCF-Relaxation to $\boldsymbol{L}_h\left(\boldsymbol{U}_h, \boldsymbol{\theta}_h\right) = f_h$
2. Restrict residual $\boldsymbol{R}_h$ and approximate solution $\overline{\boldsymbol{U}}_h$:
   $\boldsymbol{R}_H^n \leftarrow \boldsymbol{R}_h^{nc}$, $\overline{\boldsymbol{U}}_H^n \leftarrow \overline{\boldsymbol{U}}_h^{nc}$ for $n = 0, \ldots, N_H - 1$
3. Solve $\boldsymbol{L}_H\left(\boldsymbol{V}_H, \boldsymbol{\theta}_H\right) = \boldsymbol{L}_H\left(\overline{\boldsymbol{U}}_H, \boldsymbol{\theta}_H\right) + \boldsymbol{R}_H$
4. Solve coarse-level error: $\delta \boldsymbol{U}_H = \boldsymbol{V}_H - \overline{\boldsymbol{U}}_H$
5. Update fine-level approximate solution:
   $\overline{u}_h^{nc} \leftarrow \overline{u}_h^{nc} + \delta \boldsymbol{u}_H^n$ for $n = 0, \ldots, N_H - 1$
6. Check convergence:

if $||\boldsymbol{R}_h|| \leq tol$ **then**
  Return;
end
Proceed to step 1;

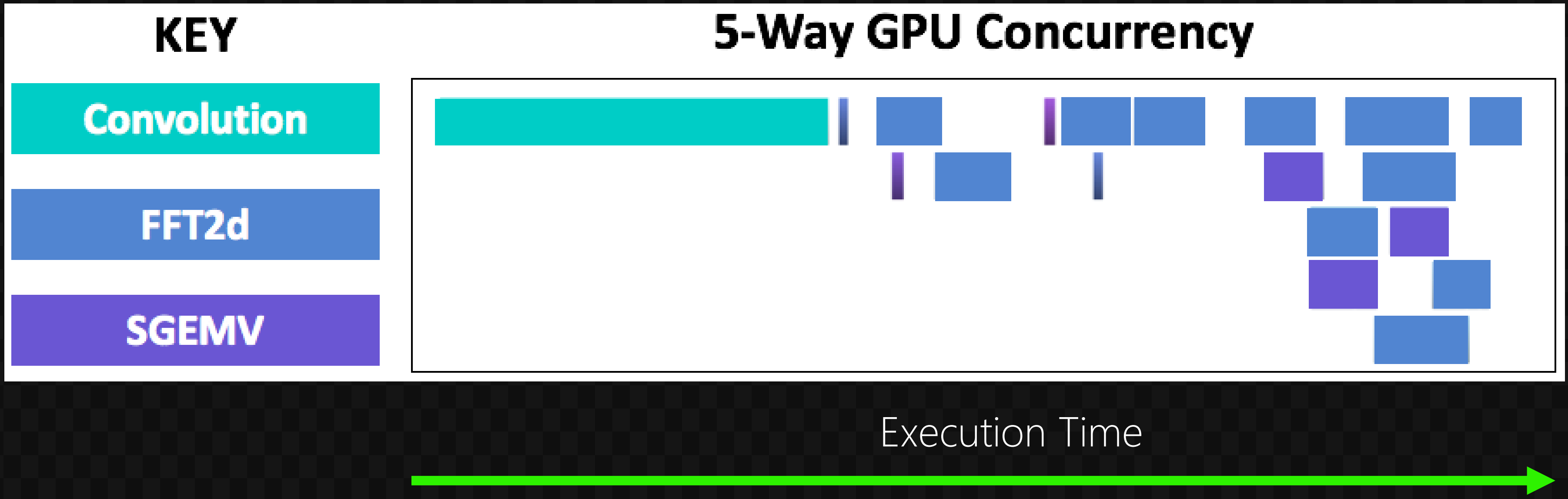


Results: GPU Concurrency

<u>Implementation</u>

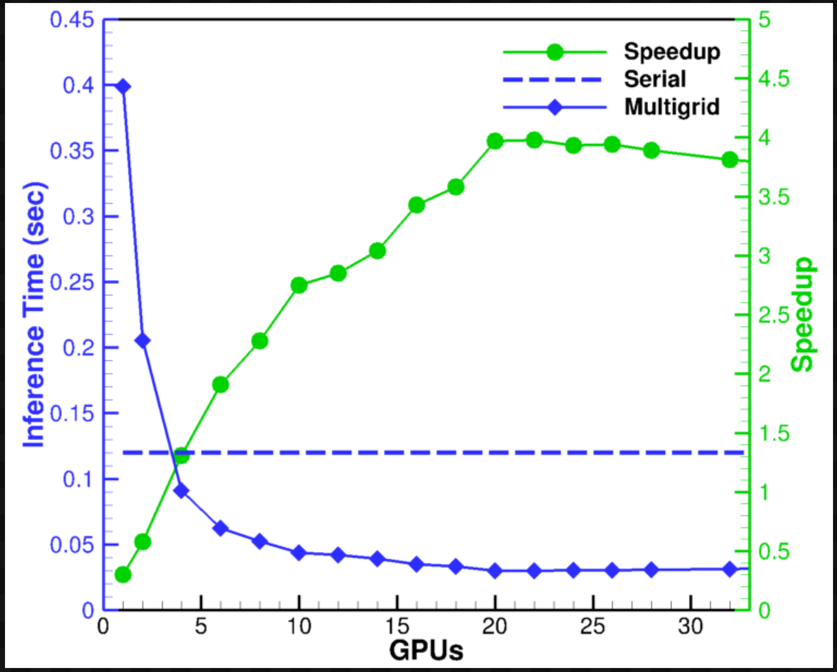
- Wrapped CuDNN kernels
  - Enabled CUDA Streams for asynchronous execution

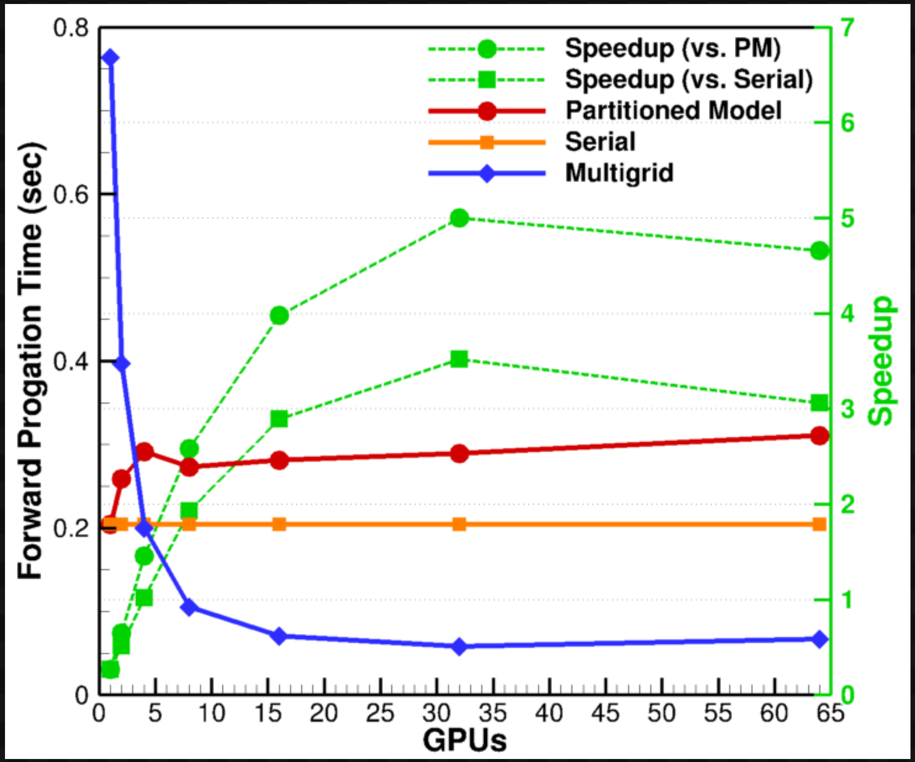
- Placed multiple layer-blocks on same GPU – each with own CUDA Stream
  - Used OpenMP to parallel launch blocks from different CPU-threads

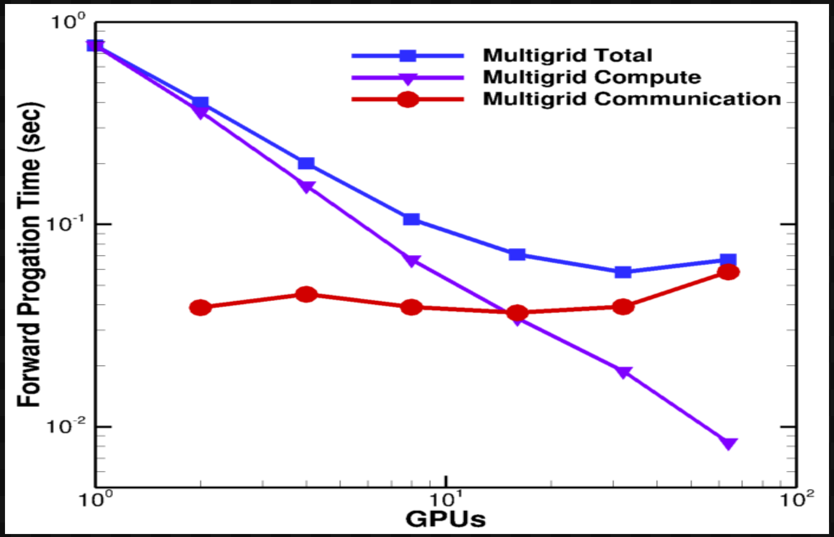


Results

- MNIST Data Set
- 4,096 Layers
  - 7 x 7 convolution layers
  - 50 output channels
  - Padding size: 1
- 3,248,524 parameters





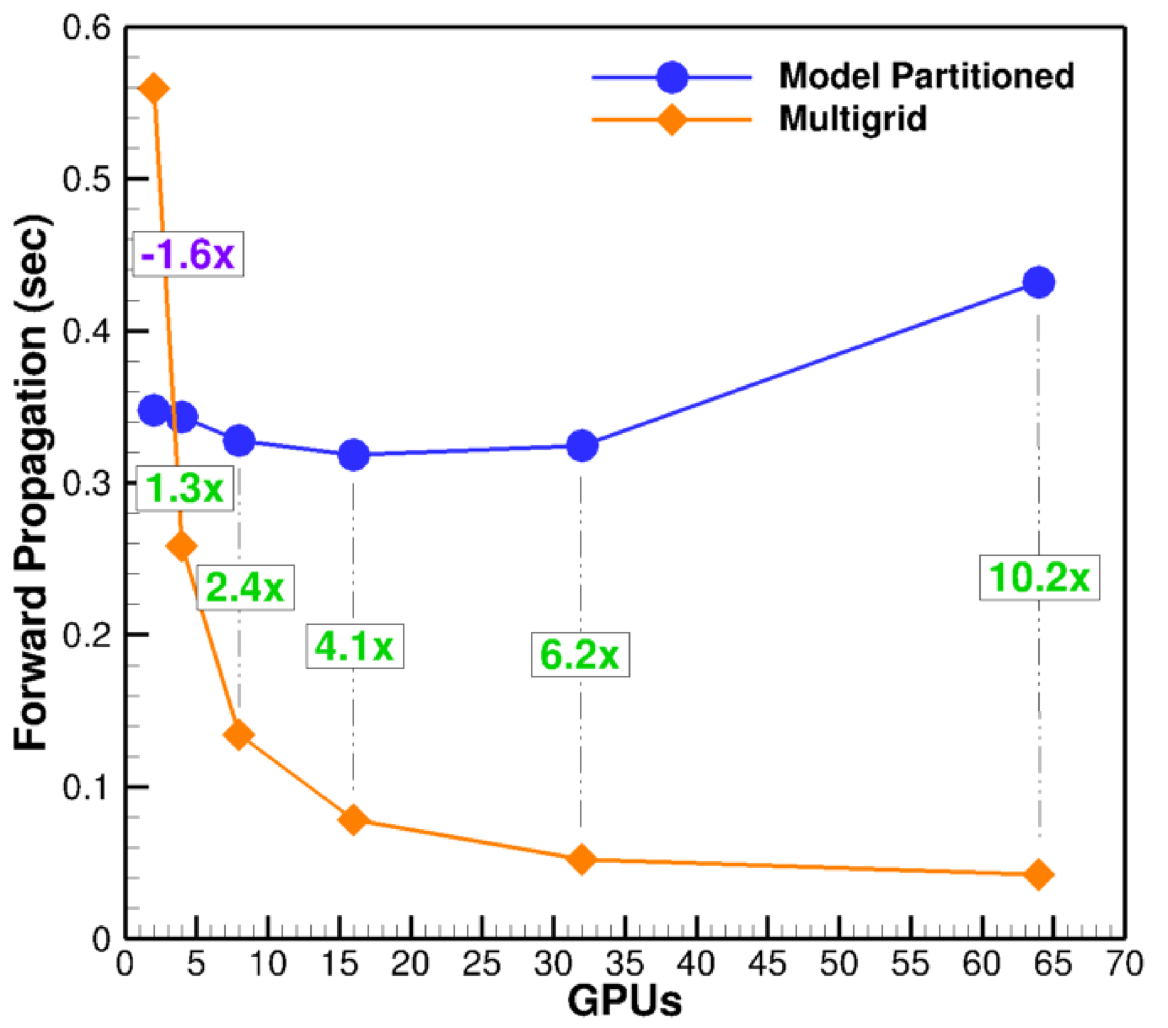


Results

- MNIST Data Set
- 4,115 Layers
  - 7 x 7 convolution layers
  - 20 output channels
  - 16 fully connected layers
  - Padding size: 1
- 2,071,328,150 parameters

4 GPUs: Computation to communication ratio is 92.8%

64 GPUs: Computation to communication ratio is 34.5%



Acknowledgements

Lincoln Laboratory Leadership

Dave Martinez, Steve Rejto, Marc Zissman

Lincoln Laboratory Supercomputing Center

William Arcand, David Bestor, William Bergeron, Chansup Byun, Matthew Hubbell, Michael Houle, Anna Klein, Peter Michaleas, Lauren Miliechin, Julie Mullen, Andrew Prout, Antonio Rosa, Charles Yee

Compute Time

TX-Green AI Accelerator (TX-GAIA)

Funding

MIT-Air Force AI Innovation Accelerator

Thank You

Questions?