Habitat: A Runtime-Based Computational Performance Predictor for Deep Neural Network Training

Geoffrey X. Yu, University of Toronto/Vector Institute; Yubo Gao, University of Toronto; Pavel Golikov and Gennady Pekhimenko, University of Toronto/Vector Institute
https://www.usenix.org/system/files/atc21-yu.pdf

Presenter

Dipak Acharya
University of North Texas
Denton TX

Selecting GPU For Deep Learning Training

Desktop	Shared Cluster	Cloud
RTX 4090 RTX 2080 Ti GTX 1080	RTX 6000 RTX A5500	Nvidia H100 Nvidia A100 Nvidia V100

Build?

Buy?

Rent?

Performance

Solution: Predict the Performance of a GPU

Cost

Why Predict the Performance of DNN on a GPU

Measure performance directly

GPU Availability

Use publicly available Benchmarks

Only available for popular models

Use Heuristics

Proven to be not accurate

Always use the "Best" GPU

Performance changes based on model

Might be less cost effective

Observations

1. **Repetitive computation**
    - DNN training involves thousands repetitive forward and backward passes
2. **Building blocks of DNN**
    - DNNs are formed by combination of thousands of basic operators such as convolution, pooling, linear transform etc.
3. **Runtime information available**
    - DNN developers already have a lower tier GPU available to them which gives important runtime information

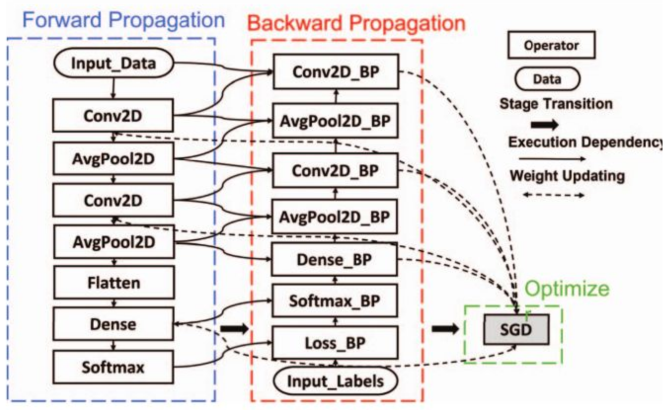
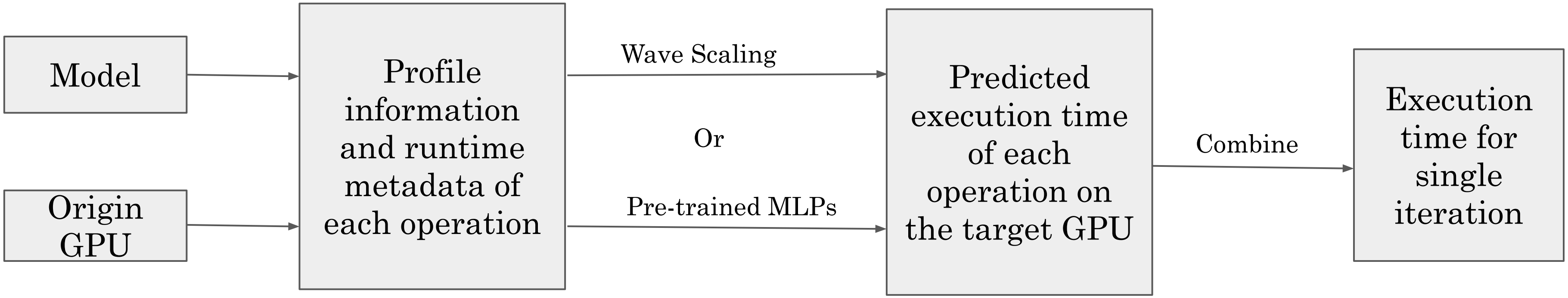
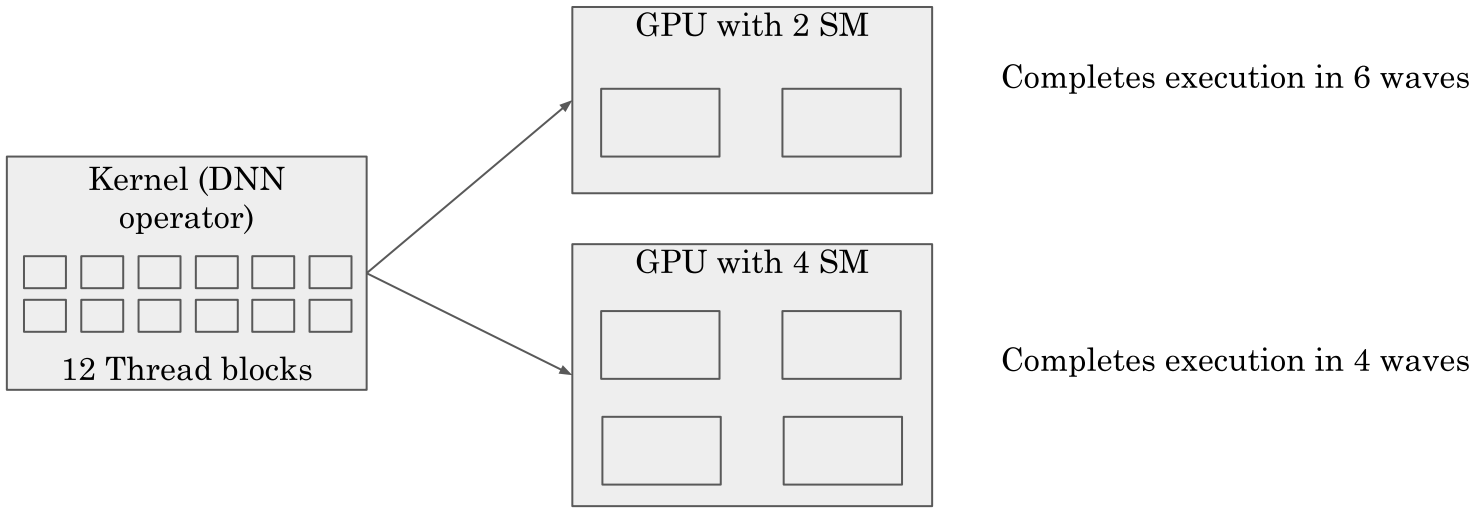


Fig. DNN model as a computational graph

Habitat


Model
Origin GPU
Profile information and runtime metadata of each operation
Wave Scaling
Or
Pre-trained MLPs
Predicted execution time of each operation on the target GPU
Combine
Execution time for single iteration

Habitat: Wave Scaling



Other factors that affect the execution: **Memory Bandwidth**, **Wave Size** and **Clock Frequency**

Wave Scaling

$$T_d = \left\lceil \frac{B}{W_d} \right\rceil \left(\frac{D_o}{D_d} \frac{W_d}{W_o} \right)^{\gamma} \left(\frac{C_o}{C_d} \right)^{1-\gamma} \left\lceil \frac{B}{W_o} \right\rceil^{-1} T_o$$

T_i = Execution Time

D_i = Memory Bandwidth

C_i = Clock Frequency

B = Number of thread Blocks in the Kernel

W_i = Number of thread Blocks in the Wave

$\gamma \in [0, 1]$ - Memory bandwidth boundedness

Selecting Gamma (γ)

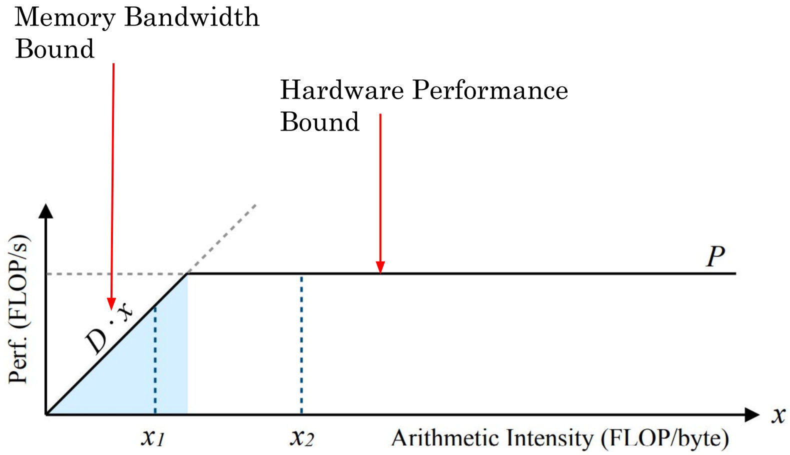
Roofline Model:

- Number of floating point operations per byte of data read/write (x)

- Kernel performance is minimum of
  - Hardware peak performance

    OR

  - Bandwidth times kernel’s arithmetic intensity



$$\gamma = \begin{cases} (-0.5/R)x + 1 & \text{if } x < R \\ 0.5R/x & \text{otherwise} \end{cases}$$

Habitat: MLP Predictor

- **Predict execution time of the kernel-varying operation**
  - Convolution, LSTMs, Batched matrix multiplication, linear layer
- **Input features**
  - Layer dimensions (eg. input/output channels sin convolution)
  - Memory capacity and Bandwidth of target GPU
  - Number of Streaming Multiprocessors (SMs) on target GPU
  - Peak FLOPS of the target GPU
- **Model architecture**
  - Input layer, 8 hidden layers and output layer
  - Each hidden layer with ReLU activation with 1024 units

MLP: Data Collection

- Data for the kernel-varying operations were collected from randomly sampled input configurations.
- Each operator uses a predefined range of parameters.

Operation	Features	Dataset Size
2D Convolution	$7+4$	$91,138 \times 6$
LSTM	$7+4$	$124,176 \times 6$
Batched Matrix Multiply	$4+4$	$131,022 \times 6$
Linear Layer	$4+4$	$155,596 \times 6$

- Data is collected for 6 different GPUs ranging 3 generations.

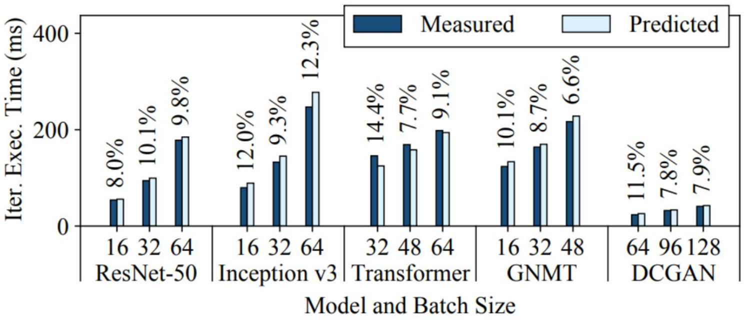
GPU	Generation	Mem.	Mem. Type	SMs	Rental Cost[6]
P4000 [65]	Pascal [63]	8 GB	GDDR5 [56]	14	–
P100 [62]		16 GB	HBM2 [4]	56	$1.46/hr
V100 [66]	Volta [67]	16 GB	HBM2	80	$2.48/hr
2070 [69]	Turing [72]	8 GB	GDDR6 [57]	36	–
2080Ti [70]		11 GB	GDDR6	68	–
T4 [71]		16 GB	GDDR6	40	$0.35/hr

Evaluation: Accuracy

Evaluation for models
Resnet-50, Inception v3, Transformer, GNMT, DCGAN

Experiments done with GPUs
V100, 2080 Ti, T4, 2070, P100, P4000

Average end-to-end accuracy across all experiments is **11.8%**

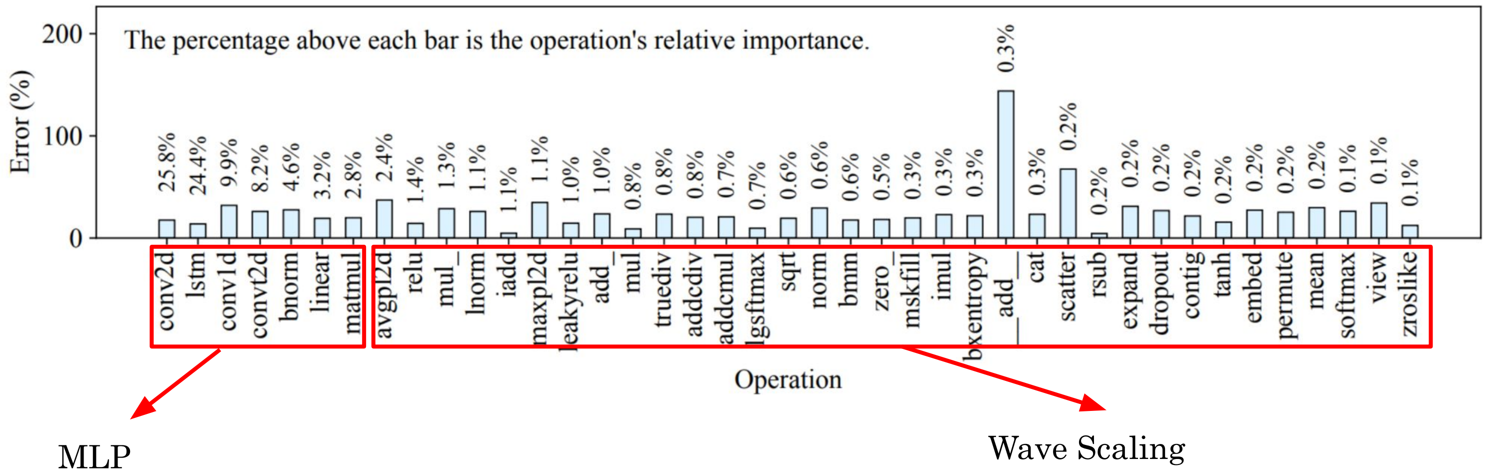


(a) Predictions onto the V100

Fig: Prediction errors for V100 GPU for different models

Refer Paper for errors breakdown for other GPUs

Evaluation: Error Contribution

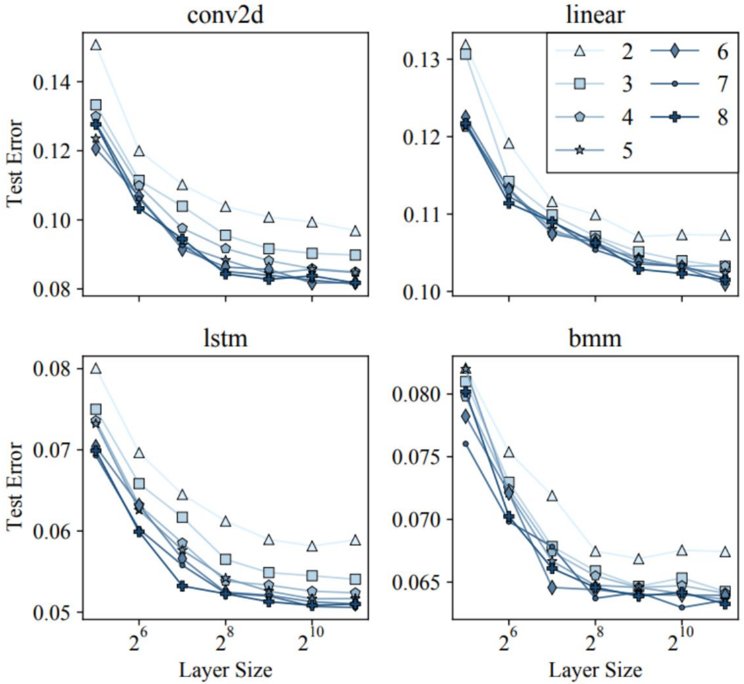


Both MLP and Wave Scaling give prediction within acceptable error range

Evaluation: MLP Configuration

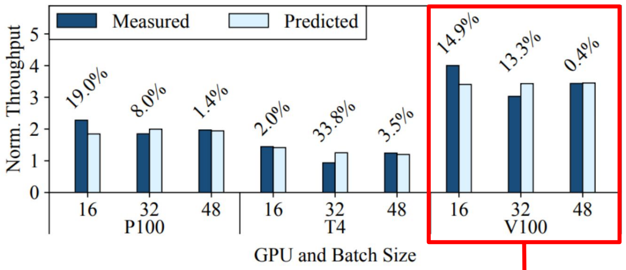
- More layer size give better accuracy
  - But increasing layer size beyond 2^{10} does not give any more improvement

- Increasing number of layers also increase the accuracy
  - Selecting 8 layers for the MLP gives acceptable accuracy

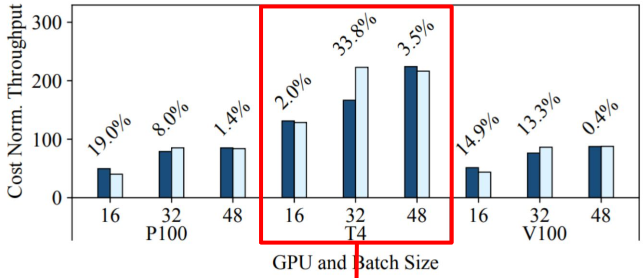


Evaluation: Making GPU Decisions

Selecting a GPU to rent, **P100** or **T4** or **V100**



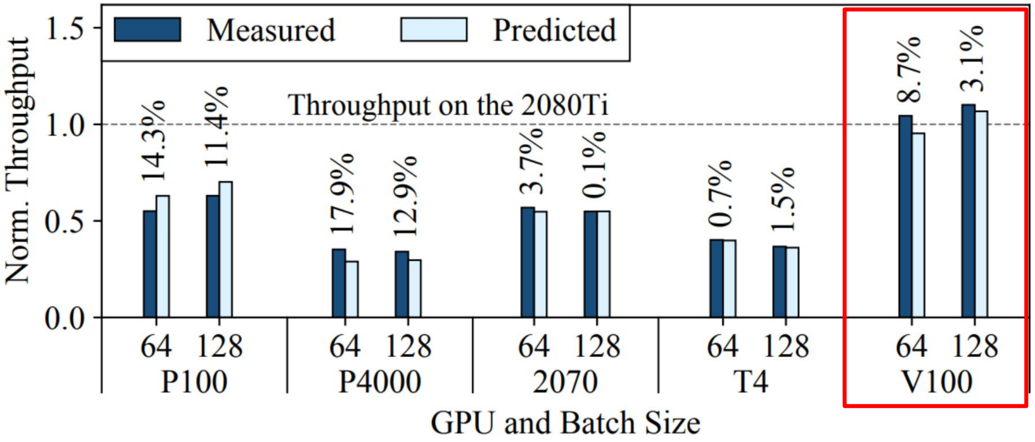
Best overall performance



Best price for performance

Habitat leads to correct decisions for selecting a GPU for DNN training

Evaluation: Is V100 Always Better?



V100 gives the best performance but is only marginally better than 2080 Ti, which is much more cost efficient.

Contributions

- **Wave Scaling**
  - Proposed a novel technique for scaling execution time of a Kernel on one GPU to another GPU

- **Habitat**
  - Implementation and evaluation of the tool that uses wave scaling combined with pre trained MLPs for predicting the end-to-end execution time of DNN training iteration from one GPU to another GPU.

Limitations

- Evaluation based on limited number of GPUs.
    - While this work evaluates only 6 GPUs from Pascal, Volta and Turing, while it does not evaluates Ampere architecture.
    - Running this experiment with 2 Ampere GPUs gave higher error compared to what is claimed by the authors.
- Potential scalability issues
    - With more complex GPU architectures in the future, more operators will become Kernel-varying.
    - The proposed solution may become unscalable as it will require training Large number of MLPs.
    - Furthermore compiler optimizations may result in more kernel varying operators
- Distributed training
    - As most demanding GPU tasks require cluster of GPUs rather than single GPU, Habitat will have little to no application in these situations.

Thank You

Questions