



# Benchmarking AI Inference: Where we are in 2020

Miro Hodak, David Ellison and Ajay Dholakia, Lenovo

---

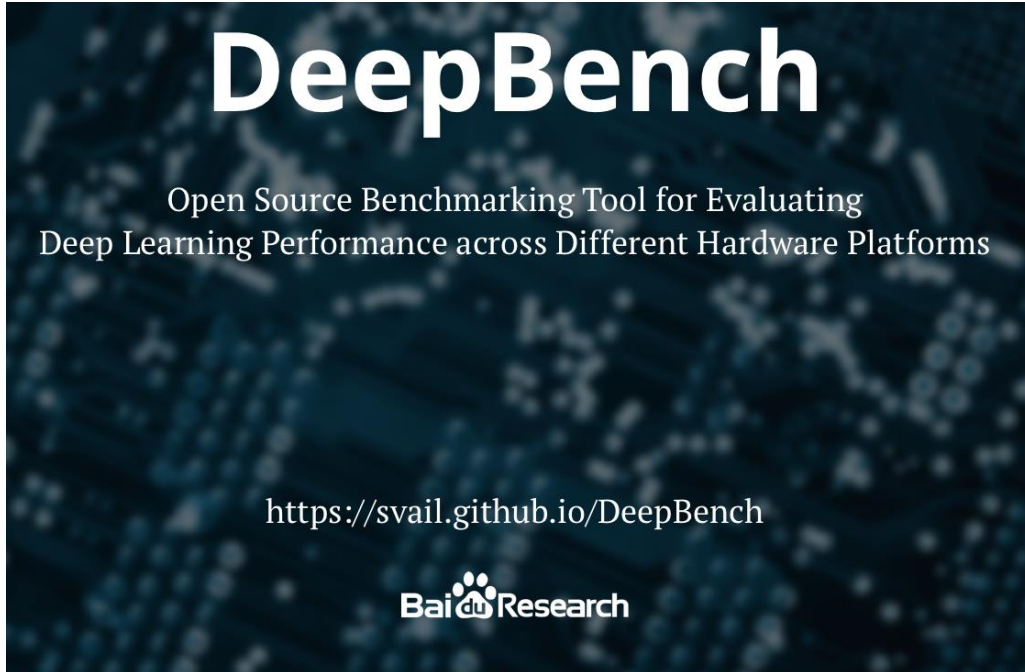
TPCTC 2020

August 31<sup>st</sup>, 2019

# + Outline

- AI Benchmarking
- AI Inference
- MLPerf Inference Benchmark: Overview
- Lenovo Experience
  - Latest Results
- Next Steps

# + AI Performance Measurement Efforts



**DeepBench**  
Open Source Benchmarking Tool for Evaluating  
Deep Learning Performance across Different Hardware Platforms

<https://svail.github.io/DeepBench>

Baidu Research

- Created by Baidu Research
- Released 2016
- Focused on evaluation of low-level operation on different chips



 **DAWN Bench**  
An End-to-End Deep Learning Benchmark and Competition

- Created at Stanford
- First results in 2018
- Time-to-accuracy as the main criterion
- Cost-based metric – for cloud services

# + AI Performance Measurement Efforts

- TPC defines two classes of benchmarks: Enterprise and Express.
  - Enterprise benchmarks are aimed at characterizing typically complex systems, wherein the benchmark specifications are provided but the implementation is left open for vendors to select based on using commercially available hardware and software products.
  - Express benchmarks, in contrast, are kit-based and require use of the kits to publish benchmark results.
  - Enterprise benchmarks have long development cycles, whereas, as the name suggests, Express benchmarks allow relatively shorter development cycles.
- TPC-AI benchmark development work is in progress and is not available to public at this time

- TPC Press Release, "Transaction Processing Performance Council (TPC) Establishes Artificial Intelligence Working Group (TPC-AI)," <https://www.businesswire.com/news/home/20171212005281/en/Transaction-Processing-Performance-Council-Establishes-Artificial>, 2017.
- R. Nambiar, S. Ghandeharizadeh, G. Little, C. Boden and A. Dholakia, "Industry Panel on Defining Industry Standards for Benchmarking Artificial Intelligence," in Nambiar, R., Poess, M. (eds.) TPCTC 2018, LNCS, vol. 11135, pp1-6, Springer (2018), Rio De Janeiro, Brazil.



# MLPerf

Fair and useful benchmarks for measuring training and inference performance of ML hardware, software, and services.

- Gaining acceptance to become the standard AI performance evaluation tool
- Official Supporters: 73 companies, 10 research institutions
- Training v0.7 released in 2020
- Inference v0.5 released in 2019

# + Lenovo and MLPerf

- Lenovo is participating
- Like most other workloads, enterprise adoption of AI / ML will be helped by benchmark demonstrations
- MLPerf suite covers broad range of use-cases
  - Training as well as Inference are important application areas
  - Different customer focus
- Lenovo will drive technical and strategic initiatives
  - Become active part of the AI / ML benchmarking community
  - Help create specific benchmarks
  - Influence future directions as the technology and the industry adoption evolves

# + MLPerf Benchmark Description - Training

- 7 Training categories
- Time-to-train the only metric
- No prescribed code to run, a set of rules to follow instead
  - Model, dataset, required accuracy, etc.
  - Each vendor can use their own implementation as long as code is available

Benchmark results (minutes)						
Image classification	Object detection, light-weight	Object detection, heavy-wt.	Translation, recurrent	Translation, non-recur.	Recommendation	Reinforcement Learning
ImageNet	COCO	COCO	WMT E-G	WMT E-G	MovieLens-20M	Go
ResNet-50 v1.5	SSD w/ ResNet-34	Mask-R-CNN	NMT	Transformer	NCF	Mini Go

- Adding metrics to incorporate energy efficiency will help with comparison

M. Hodak, D. Ellison, P. Seidel and A. Dholakia, *2018 IEEE International Conference on Big Data*, no. doi: 10.1109/BigData.2018.8621896, pp. 1945-1950, 2018.

M. Hodak and A. Dholakia, *Technology Conference on Performance Evaluation and Benchmarking*, pp. 82-93, 2018.

M. Hodak, M. Gorkovenko and A. Dholakia, *2019 IEEE International Conference on Big Data*, no. doi: 10.1109/BigData47090.2019.9005632, pp. 1814-1820, 2019.

M. Hodak and A. Dholakia, *11th TPC Technology Conference*, pp. 82-93, 2019.

# + AI Inference: Challenges in Performance Evaluation

- AI Inference is more complicated than Training
  - Requires a server-like infrastructure and multiple key parameters such as latency, throughput, and efficiency need to be balanced.
  - Must decide if running at the edge or data center. Edge infrastructure usage is growing, where latency is important, but that limits computational power.
  - Then must decide AI processing chips: CPUs, GPUs, ASICs like TPUs
  - Code designed for CPUs simply will not work on an ASIC, yet as an industry it is important to be able to fairly compare the two.
  - Currently, there is a lot of interest in this space, with over 100 companies targeting inferencing workloads
- Inferencing also needs to balance accuracy with various algorithmic considerations.
  - Network pruning, where a data scientist removes parameters from the network
  - Quantization to lower the numerical precision of model weights
- Multiple usage scenarios to consider
  - Which AI workload: Computer Vision (using CNNs), Natural Language Processing (using RNNs)
  - How queries are sent to the server: multiple patterns of inference requests.
- An effective AI benchmark must either control for or adapt to all the differences in hardware, power, and algorithmic considerations across the different usage scenarios.

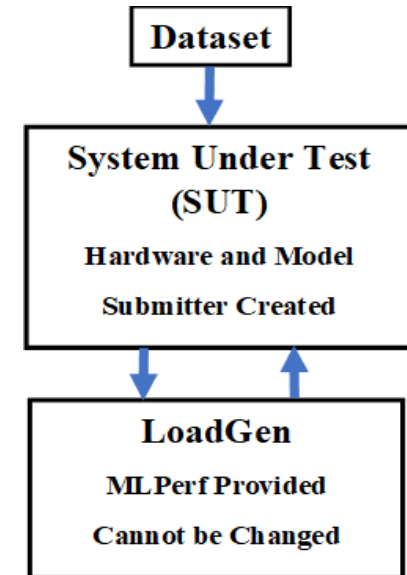


# + MLPerf Benchmark Closed vs Open

- MLPerf Inference has two divisions:
  - Closed
  - Open
- The closed section has strict rules to make sure that the results are comparable across different HW
- The open division gives submitters much more freedom, for example, they can change the model or use different metrics
- This paper focuses on the closed section.

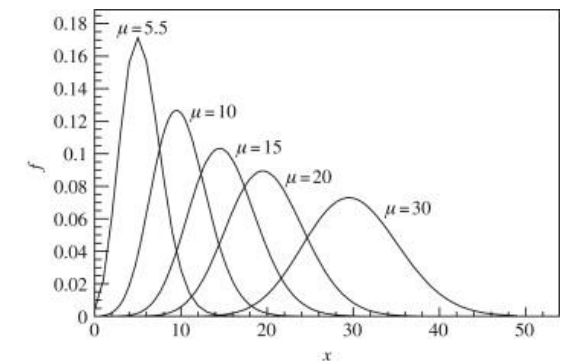
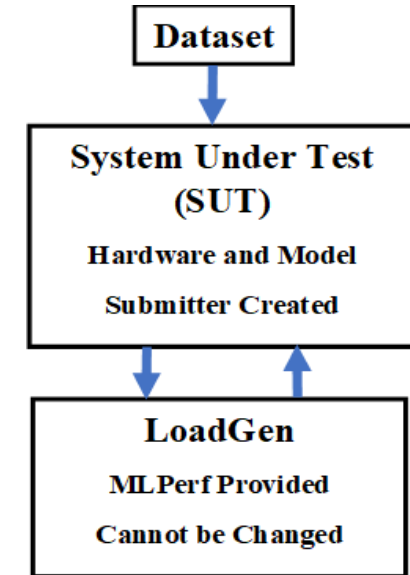
# + MLPerf Inference Benchmark - Overview

- The system works as follows: First, LoadGen sends a request to the SUT to load data set samples into memory. This “warm-up” action may include compilation and/or data preprocessing and is not counted as a part of the benchmark. Upon finishing the initialization, the SUT sends a signal back to LoadGen, which then starts to send queries according to a selected scenario.
- LoadGen supports four query-sending scenarios:
  - **Single-Stream:**
    - This scenario mimics systems where responsiveness is a critical factor such as offline AI queries performed on smartphones
    - LoadGen sends a single query to the SUT and waits for response. Upon response, completion time is recorded, and a new query is generated.
    - The metric is 90th percentile latency..
  - **Multistream:**
    - This scenario reflects systems that process input from multiple sensors.
    - Queries are sent at a fixed time interval with N samples.
    - A Quality of Service (QoS) constraint is imposed (1%) and the metric is the number of streams that the system can process while meeting the QoS constraint.
  - **Server:**
    - This scenario mimics a web service receiving queries from multiple clients.
    - LoadGen sends queries according to a Poisson distribution. A benchmark-specific latency bound is defined and only a small number of queries such as 1% for vision can exceed it.
    - The metric is the Poisson parameter representing queries per second that can be processed while meeting the latency bound requirement.
  - **Offline:**
    - This scenario covers batch processing applications like identifying people in a photo albums
    - LoadGen sends a single query containing all samples.
    - The metric is throughput in samples per second.



# + MLPerf Inference Benchmark - Overview

- The system works as follows: First, LoadGen sends a request to the SUT to load data set samples into memory. This “warm-up” action may include compilation and/or data preprocessing and is not counted as a part of the benchmark. Upon finishing the initialization, the SUT sends a signal back to LoadGen, which then starts to send queries according to a selected scenario.
- LoadGen supports four query-sending scenarios:
  - **Single-Stream:**
    - This scenario mimics systems where responsiveness is a critical factor such as offline AI queries performed on smartphones
    - LoadGen sends a single query to the SUT and waits for response. Upon response, completion time is recorded, and a new query is generated.
    - The metric is 90th percentile latency..
  - **Multistream:**
    - This scenario reflects systems that process input from multiple sensors.
    - Queries are sent at a fixed time interval with N samples.
    - A Quality of Service (QoS) constraint is imposed (1%) and the metric is the number of streams that the system can process while meeting the QoS constraint.
  - **Server:**
    - This scenario mimics a web service receiving queries from multiple clients.
    - LoadGen sends queries according to a Poisson distribution. A benchmark-specific latency bound is defined and only a small number of queries such as 1% for vision can exceed it.
    - The metric is the Poisson parameter representing queries per second that can be processed while meeting the latency bound requirement.
  - **Offline:**
    - This scenario covers batch processing applications like identifying people in a photo albums
    - LoadGen sends a single query containing all samples.
    - The metric is throughput in samples per second.



Poisson distribution

# + MLPerf Inference Benchmark – v0.5 Results

- Version 0.5 released on November 6<sup>th</sup> 2019. The following categories were included:
  - **Image Classification: ResNet-50 v1.5 with ImageNet data set**
  - **Image Classification: MobileNet-v1 with ImageNet data set**
  - **Object Detection: SSD w/ MobileNet-v1 with COCO data set**
  - **Object Detection: SSD w/ ResNet-34 with COCO 1200x1200 data set**
  - **Translation: NMT with WMT E-G data set**
- 20 possible benchmarks: 5 categories x 4 scenarios
- Datacenter Closed section: 37 entries/ 13 submitters
  - Chipmakers: Nvidia, Intel, Qualcomm, Google, Habana Labs, Alibaba, Centaur Technology, Hailo, FuriosaAI
  - Cloud providers: Alibaba, Tencent
  - OEM server vendor: Dell EMC

# + MLPerf Inference

- Best per-accelerator entries in MLPerf Inference v0.5 Offline benchmarks

	1 <sup>st</sup> place	2 <sup>nd</sup> place	3 <sup>rd</sup> place
	Score Submitter Accelerator	Score Submitter Accelerator	Score Submitter Accelerator
Image Recognition MobileNet/ImageNet	17,804 Dell EMC T4	17,474 Alibaba Cloud T4	14,602 Intel Xeon 9282
Image Recognition ResNet/ImageNet	69,307 Alibaba HanGuang 800	16,562 Nvidia Titan RTX	14,451 Habana Synapse
Object Detection SSD/MobileNet COCO	22,945 Nvidia Titan RTX	7609 Nvidia T4	7602 Dell EMC T4
Object Detection SSD/ResNet Coco 1200x1200	415 Nvidia Titan RTX	326 Habana Labs Synapse	164 Google TPU v3
Translation WMT E-G/NMT	1061 Nvidia Titan RTX	771 Google TPUv3	354 Dell EMC T4

# + MLPerf Inference Benchmark – v0.7

- Submission Deadline: September 04, 2020

Area	Task	Model	Dataset
Vision	Image classification	Resnet50-v1.5	ImageNet (224x224)
Vision	Object detection Large	SSD-ResNet34	COCO 1200x1200
Vision	Object detection Small	SSD-ResNet34	COCO 300x300
Vision	Medical image segmentation	3D UNET	BraTS 2019 (224x224x160)
Speech	Speech-to-text	RNNT	Librispeech dev-clean (samples < 15 seconds)
Language	Language processing	BERT	SQuAD v1.1 (max_seq_len=384)
Commerce	Recommendation	DLRM	1TB Click Logs

Area	Task	Datacenter Required Scenarios	Edge Required Scenarios
Vision	Image classification	Server, Offline	Single Stream, Offline
Vision	Object detection (large)	Server, Offline	N/A
Vision	Object detection (small)	N/A	Single Stream, Offline
Vision	Medical image segmentation	Offline	Single Stream, Offline
Speech	Speech-to-text	Server, Offline	Single Stream, Offline
Language	Language processing	Server, Offline	Single Stream, Offline
Commerce	Recommendation	Server, Offline	N/A

# + Experimental Setup

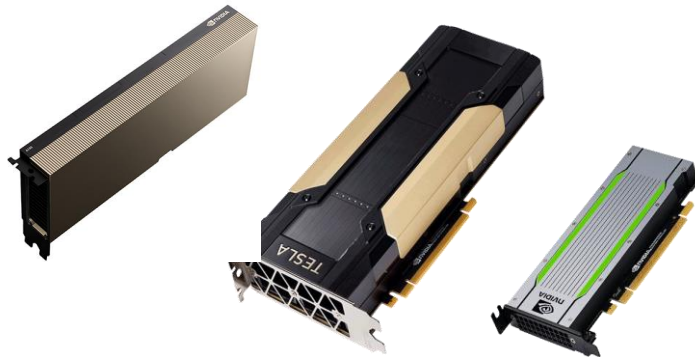
**SR670 - Datacenter**



**SE350 - Edge**



**A100, V100, T4**



## + MLPerf v0.7 experience (so far)

- Very different from other benchmarks
- Datasets and Models need to be downloaded first
  - Some are very large and/or servers are overloaded
  - Authentication may be required – downloads cannot be automated
  - This setup takes a long time
- Code options:
  - Write your own
  - Reuse from last submission (if category did not change)
  - Reference Implementation (not optimized)
- In general, need to use Inference engine provided by chip manufacturer:
  - TensorRT (Nvidia), OpenVino (Intel), etc,



# + Lenovo Unofficial MLPerf Results (Tensorflow)

Results from SR670:

- Inference throughput for different cards
- Also Inference per Watt - estimated, not part of MLPerf
- Efficiency not part of MLPerf right now, dividing by wattage can be used as an estimate

	V100S	V100 (32GB)	T4	RTX6000
<b>Performance Images per second</b>	<b>6145</b>	<b>5853</b>	<b>2035</b>	<b>5164</b>
<b>Estimated Performance/watt Images per second per watt</b>	<b>24.6</b>	<b>23.4</b>	<b>29</b>	<b>20.6</b>

# + Ongoing Work and Next Steps

- MLPerf Inference is clearly ahead of other efforts having already released a first version of the benchmarks and receiving entries from some of the most important players in the emerging AI industry.
  - This first version shows the value of accelerators across AI inference scenarios
- One of the most valuable aspects of MLPerf has been standardization of AI workloads enabling comparison across the systems.
- The reference implementations and submitter-created codes are free to use
- We have also identified areas for improvements:
  - No measure of power efficiency
  - No measure of per-device performance
  - Re-use requires significant effort and expertise

# + Summary and Conclusions

- MLPerf Benchmarks for AI are gaining popularity
- AI Inference usage is growing, driving the need for performance evaluation
- AI Inference systems vary widely in capabilities, power consumption, cost
- MLPerf Inference 0.5 is a key step forward
- Ongoing experimentation is delivering useful lessons for designing and selecting the appropriate systems for a variety of use cases