MLModelScope: Evaluate and Profile ML Models at Scale and Across Stack

Cheng Li*, Abdul Dakkak*, Jinjun Xiong†, Wen-Mei Hwu*

{cli99, dakkak, w-hwu}@illinois.edu, jinjun@us.ibm.com

*University of Illinois Urbana-Champaign, †IBM Research Yorktown

Motivation

- The current landscape of ML is rife with diverse models, HW/SW stacks, and evaluation methodologies
- ML model performance is impacted by the interplay between frameworks, system libraries, compilers, and hardware platforms

Currently, evaluating ML (models, frameworks or systems) is both arduous and error-prone and there is lack of tools that

- makes it fair and simple to compare different ML innovations
- enables understanding ML model performance at each level of the HW/SW stack

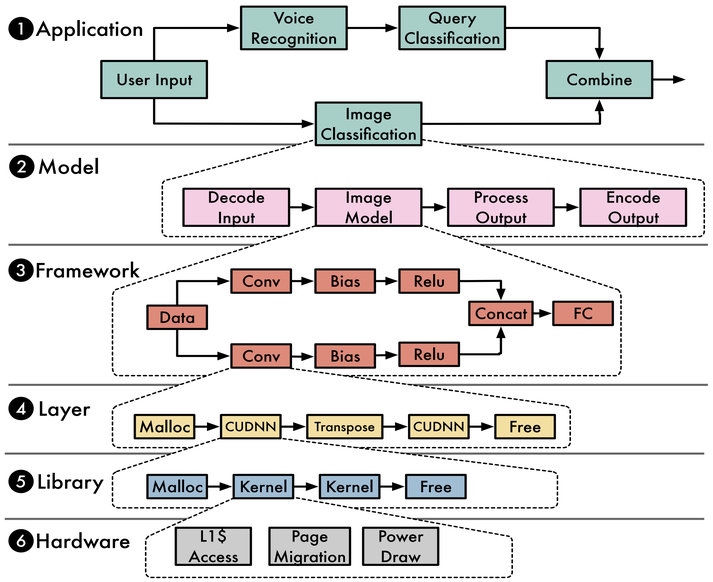

Figure 1: Execution of an AI application at different levels of HW/SW abstractions

MLModelScope

- An open-source, extendable and customizable framework to evaluate and profile ML models at scale and across stack
- Command line, API or web interface
- End-to-end profiling at different abstraction levels
- Built-in support for Caffe, Caffe2, CNTK, MXNet, PyTorch, TensorFlow, and TensorRT
- Runs on X86, PPC, ARM using CPU, GPU, and FPGA
- An online portal of continuously updated evaluation and profiling results

Modular Design

The system design is flexible enough to be extended to accommodate generic ML pipelines

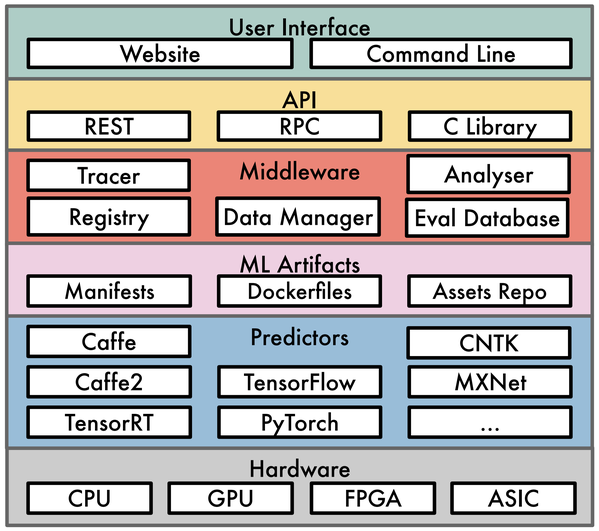

Figure 2: MLModelScope is built from a set of reusable components and is extendable and customizable

Evaluation at Scale

Key distributed design components:

- Common prediction interface which works for any Framework and Model Predictors
- Profilers and Tracers for across stack profiling

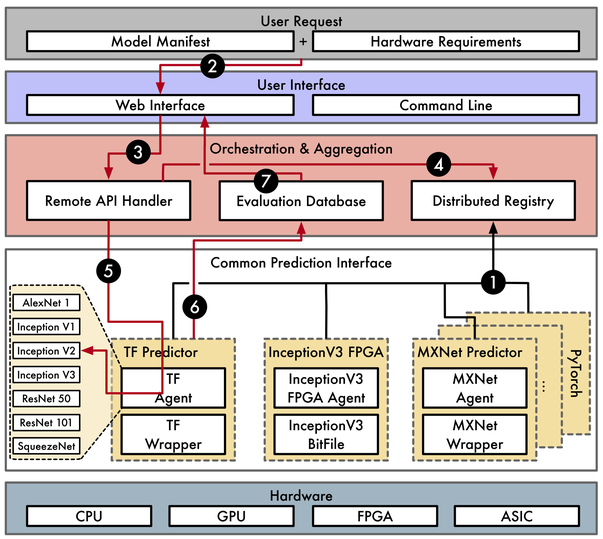

Figure 3: MLModelScope's runtime enables easy, scalable, and repeatable model evaluation across frameworks, models, and systems

Profiling Across Stack

- To introspect model performance across the HW/SW stack, currently researchers have to switch between tools and manually stitch the outputs (might not be possible)
- A scalable across-stack profiling scheme that correlates and aggregates profiles from different profiling providers into a single timeline
- Automatic model performance analysis, characterization, and reporting pipeline
- While we currently focus on ML model performance on GPUs, the across-profiling design is general and extensible

Case Study: MLPerf Inference ResNet50 v1.5

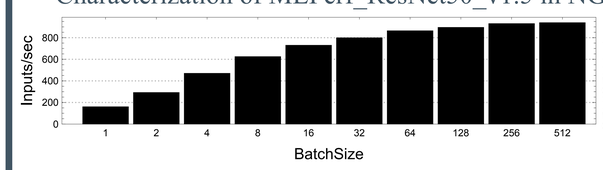

Figure 4: Throughput across batch sizes. Throughput saturates at batch size 256

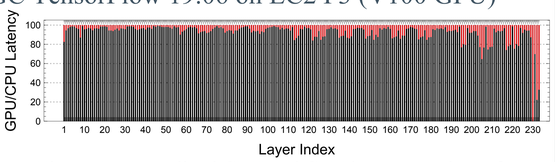

Figure 5: Normalized GPU and CPU latency per layer for batch size 256

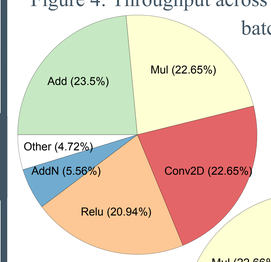

Figure 6: Layer Occurrence

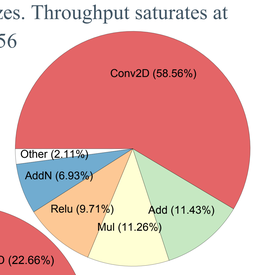

Figure 7: Layer Latency

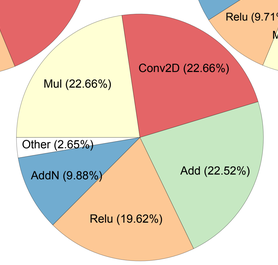

Figure 8: Layer Allocated Memory

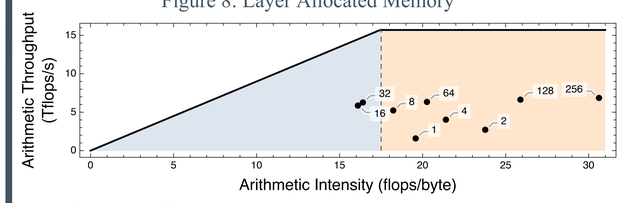

Figure 10: The roofline analysis of the model across batch sizes

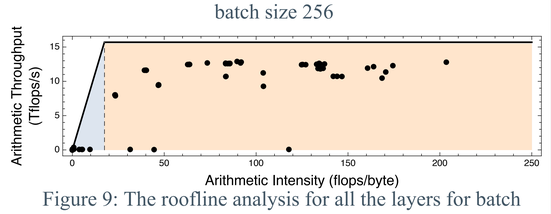

Figure 9: The roofline analysis for all the layers for batch size 256 on V100, which has an ideal arithmetical intensity of 17.44 flops/byte.

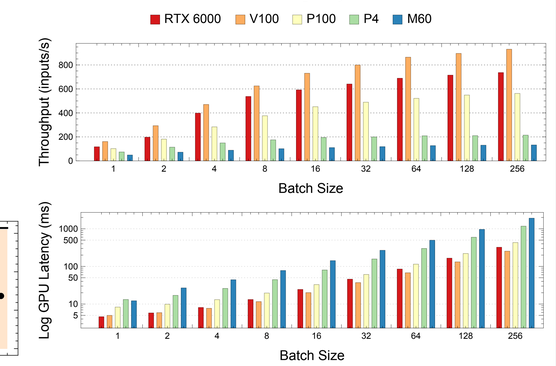

Figure 11: The throughput and GPU latency (log scale) across batch sizes and GPUs.

Resources

- Documentation at docs.mlmodelscope.org
- https://arxiv.org/abs/1811.09737
- https://arxiv.org/abs/1904.12437
- Learn more about the center's work at C3SR.com

C3SR center for cognitive computing systems research

IBM | ILLINOIS

docs.mlmodelscope.org