Analyzing Deep Learning Model Inferences for Image Classification using OpenVINO

Zheming Jin (zjin@anl.gov)

Acknowledgement: This work used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Motivation

- Deep learning model inference on an integrated GPU may be desirable
- Deep learning model inference on a CPU is still of interests to many people
- Gain a better understanding of how a model is executed using the vendor-specific high-performance library on a GPU, and the effectiveness of using the half-precision floating-point format and 8-bit model quantization

The OpenVINO Toolkit

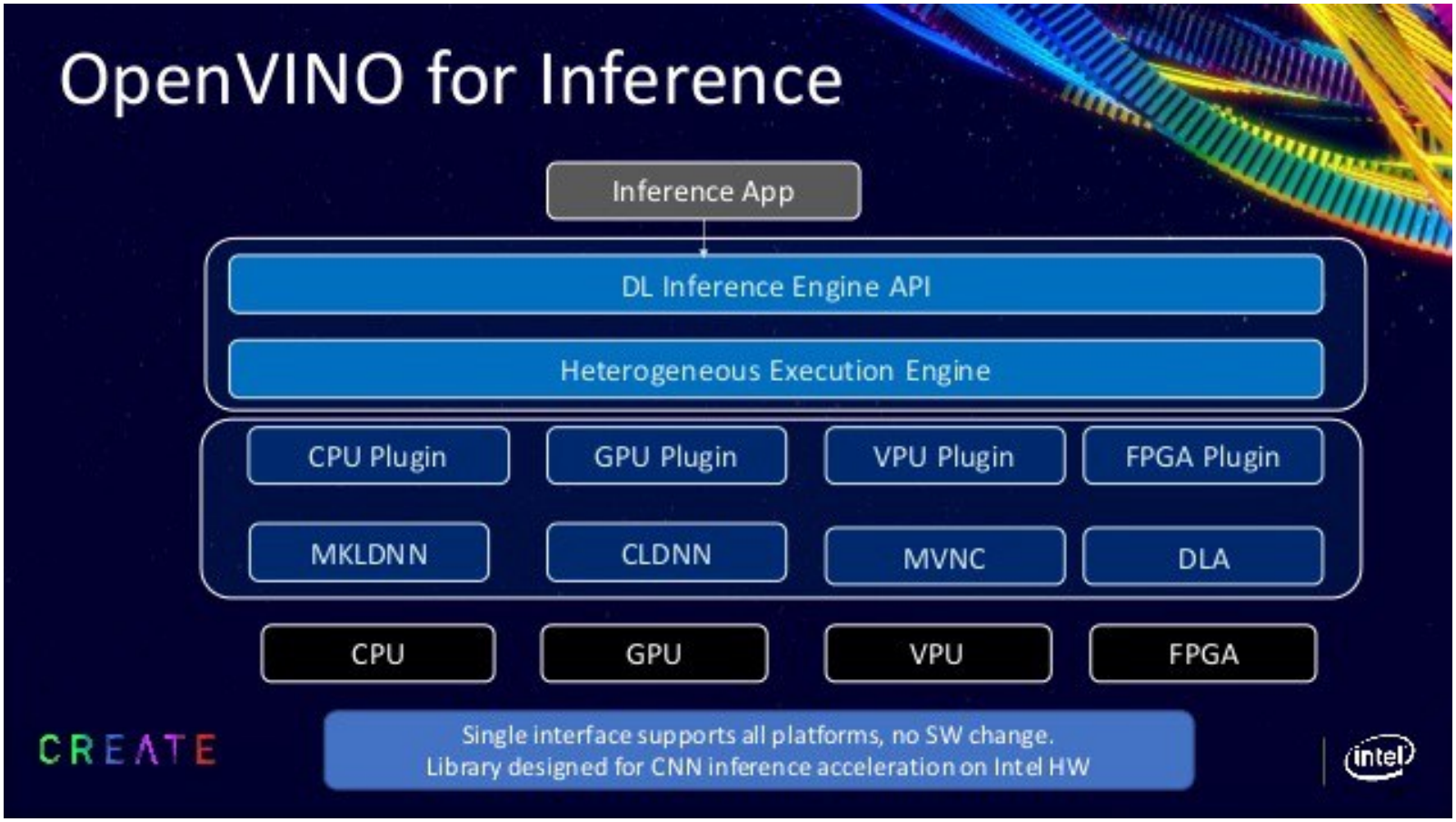


Image credit: Intel

Summary of optimizing and deploying a pretrained Caffe model

- Convert a Caffe model to intermediate representation (IR) using the Model Optimizer for Caffe
  - IR consists of .xml (network topology) and .bin (weights and biases binary) files
  - Optimized IR: node merging, drop unused layers, etc.
- Test the model using the Inference Engine via the sample applications
  - C++ APIs to read IR, set input/output formats, and execute the model on a device
  - Heterogeneous plugin for each device (CPU, GPU, FPGA, etc.)

Experimental setup (continued)

- Intel Xeon E3-1585v5 microprocessor
  - CPU: four cores and each core supports two threads
  - Integrated GPU: 72 execution units
- OpenCL 2.1 NEO Driver: version 19.48.14977
- API version of the inference engine is 2.1
- CPU/GPU plugins build version is 32974
- Operating system is Red Hat Linux Enterprise 7.6 (kernel version 3.10.0-957.10.1)

Experimental setup

- Choose Pretrained Caffe models, which will be shown in the next slide, for image classification from Open Model Zoo
- Calibration dataset is 2000, a subset of ImageNet 2012 validation set
- Measure the latency of model inference
  - Batch size and the number of infer requests are one
  - Latency is averaged over 32 iterations
- Note INT8 inference on the integrated GPU and FP16 inference on the CPU are currently not supported

Performance of 14 pretrained Caffe models for image classification

TABLE I. PERFORMANCE OF MODEL INFERENCES. INT8 INFERENCE ON THE GPU AND FP16 INFERENCE ON THE CPU ARE NOT SUPPORTED.

Unit: FPS	CPU		iGPU	
	FP32	*INT8*	*FP32*	*FP16*
AlexNet	79	89	275	331
VGG16	10	15	27	58
DenseNet-121	59	75	53	65
DenseNet-169	20	31	23	30
GoogleNet-v1	107	154	154	174
GoogleNet-v4	15	23	23	35
ResNet-50	53	78	91	139
ResNet-101	25	39	45	71
SqueezeNet1.0	212	302	295	370
SqueezeNet1.1	462	573	454	502
Inception-ResNet-v2	14	19	20	30
MobileNet-v1-1-224	275	335	337	472
MobileNet-v2	307	315	267	332
SE-ResNext-50	21	32	28	34

Results obtained using an Intel Xeon E3-1585 v5 microprocessor
CPU: four cores, two thread per core, running at 3.5 GHz
Integrated GPU (iGPU): 72 executing units running at 1.1.5 GHz

Performance comparison between the CPU and GPU

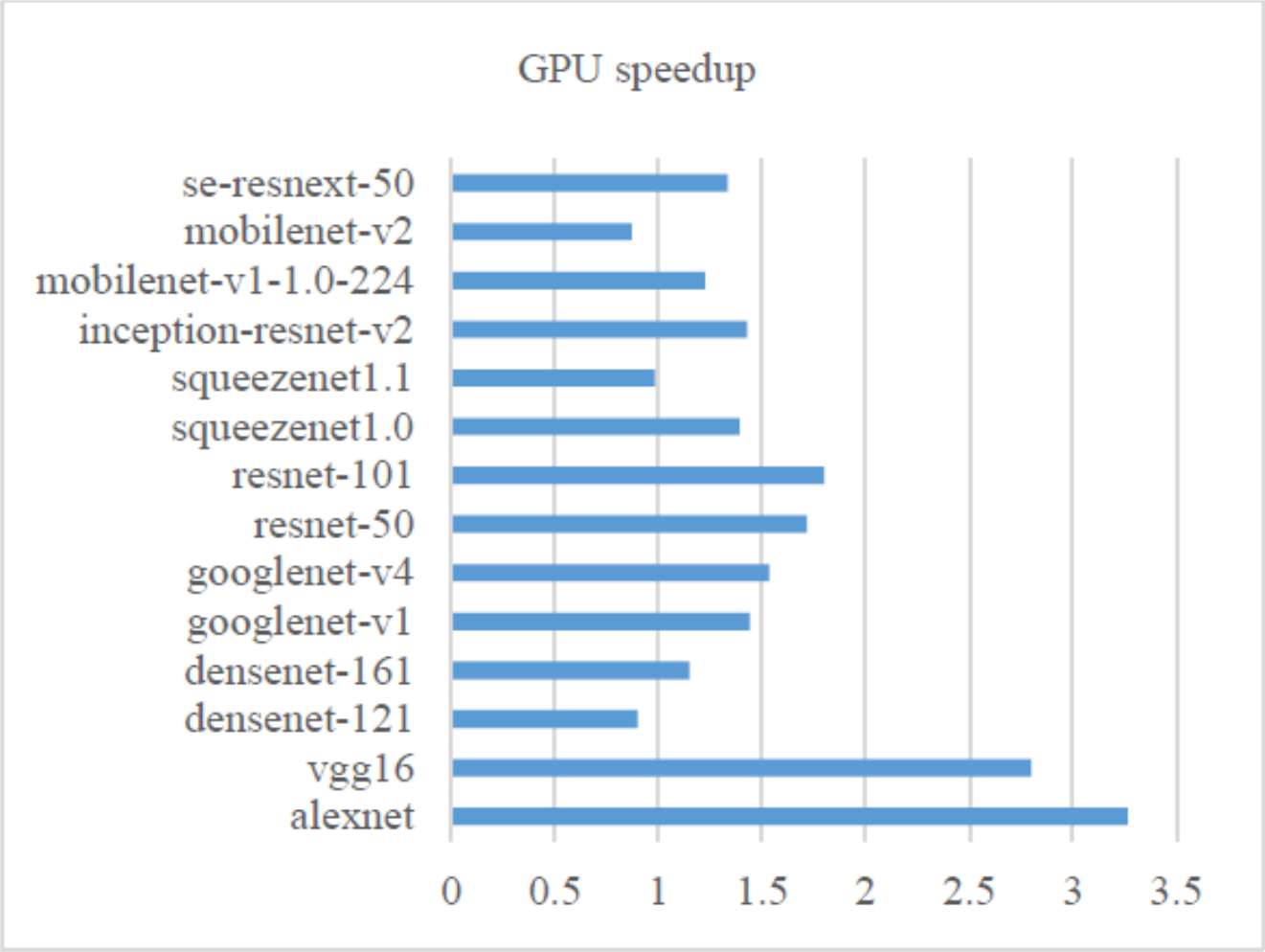
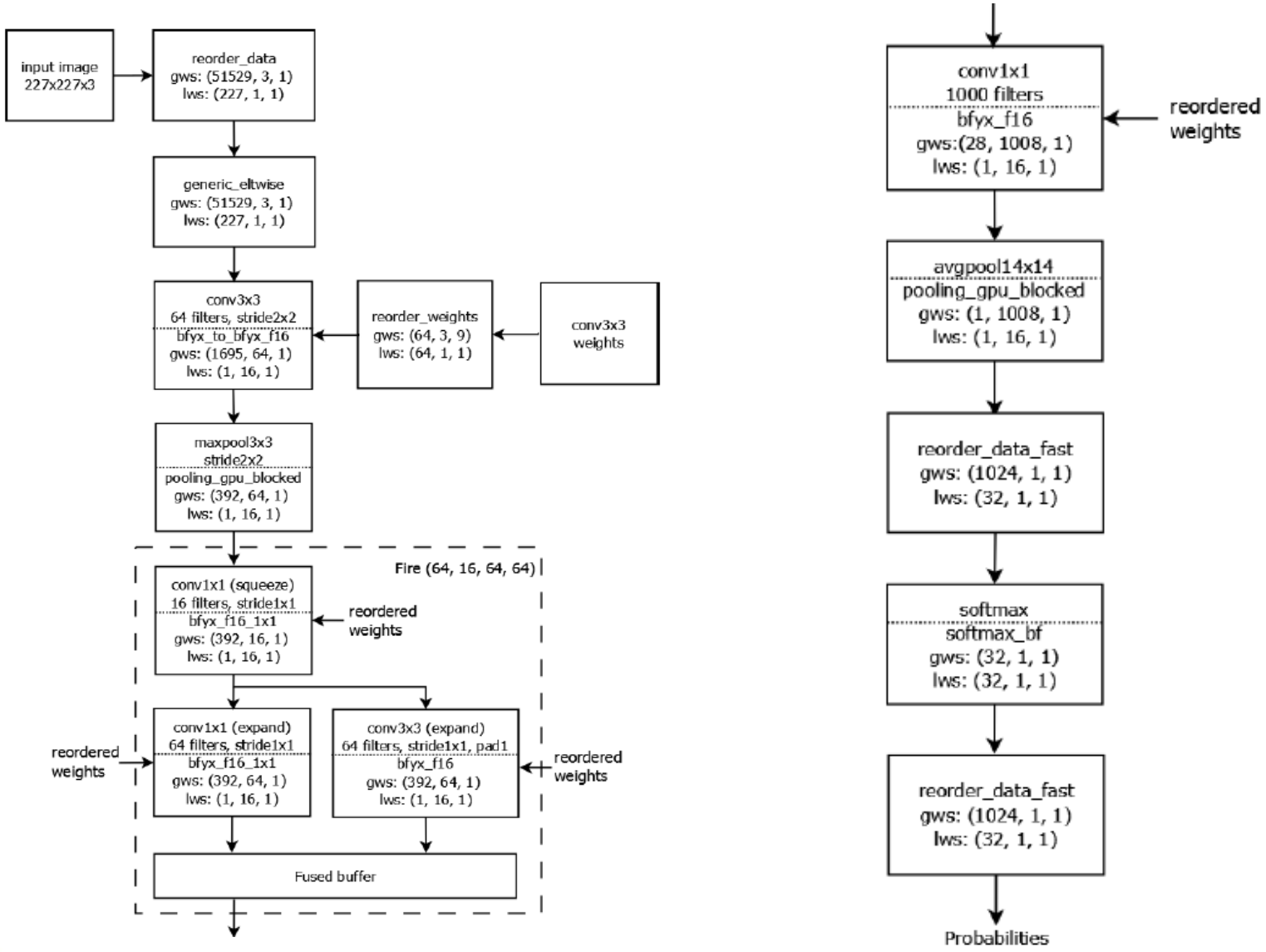


Fig. 1. Performance speedup of FP32 model inferences using the iGPU

Results obtained using an Intel Xeon E3-1585 v5 microprocessor
CPU: four cores, two thread per core, running at 3.5 GHz
iGPU: 72 executing units running at 1.1.5 GHz

Implementation of Squeezenet1.1 using clDNN



Squeezenet 1.1 on the CPU (MKLDNN) and GPU

TABLE II. IMPLEMENTATIONS OF EACH LAYER ON THE DEVICES

Layer name	GPU impl. (FP32)	CPU impl. (INT8)
input reorder	reorder_data	jit_uni_I8
scaleShift/Add	generic_eltwise_ref	jit_avx2_FP32
integer ScaleShift/Add	N/A	reorder_jit_uni_FP32
conv1	bfyx_to_bfyx_f16	pooling_jit_avx2_I8
conv1_relu	optimized away	optimized away
fire*N*/squeeze1×1	bfyx_f16_1x1	pooling_jit_avx2_I8
fire*N*/squeeze1×1_relu	optimized away	optimized away
fire*N*/expand1×1	bfyx_f16_1x1	pooling_jit_avx2_I8
fire*N*/expand1×1_relu	optimized away	optimized away
fire*N*/expand3×3	bfyx_f16	pooling_jit_avx2_I8
fire*N*/expand3×3_relu	optimized away	optimized away
fire*N*/concat	optimized away	concat_ref_I8
maxpool1	pooling_gpu_blocked	pooling_jit_avx2_I8
maxpool3	pooling_gpu_blocked	pooling_jit_avx2_I8
maxpool5	pooling_gpu_blocked	pooling_jit_avx2_I8
conv10	bfyx_f16	pooling_jit_avx2_I8
conv10 reorder	N/A	reorder_jit_uni_FP32
avgpool10	pooling_blocked	pooling_jit_avx_FP32
avgpool10 reorder	reorder_data_fast	N/A
softmax	softmax_bf	softmax_ref_any_FP32
postprocess	reorder_data_fast	reorder_jit_uni_FP32

Comparison to other studies [1,2]

- FP32 image classification and object detection on an Intel Skylake 18-core CPU with the AVX512 instruction set
  - Current work is focused on the performance improvement using an AVX-2 CPU which is common for edge devices
- Performance of three image classification models using OpenVINO on the AWS DeepLens platform that features an Intel Graphics HD 505 iGPU
  - Current work obtains 10X more speedup on our iGPU using the current toolkit

[1] Liu, Y., Wang, Y., Yu, R., Li, M., Sharma, V. and Wang, Y., 2019. Optimizing CNN Model Inference on CPUs. In 2019 USENIX Annual Technical Conference (pp. 1025-1040).

[2] Wang, L., Chen, Z., Liu, Y., Wang, Y., Zheng, L., Li, M. and Wang, Y., 2019, August. A Unified Optimization Approach for CNN Model Inference on Integrated GPUs. In Proceedings of the 48th International Conference on Parallel Processing

Summary

- The quantized models are 1.02X to 1.56X faster than the FP32 models on the target CPU
- The FP16 models are 1.1X to 2X faster than the FP32 models on the target iGPU
- The iGPU is on average 1.5X faster than the CPU for the FP32 models

Thanks