Vitis AI 3.0 Webinar Q&A

1) Does Vitis™ AI 3.0 enable users to deploy custom operators on AIE-ML cores? If some PyTorch operation is not supported, can it be coded by the developer and deployed on the AIE-ML?

The workflow mentioned is supported today and will be enhanced and documented for ease of use.

During early access, the DPU IP is encapsulated in a pre-built board image and is not available separately. At production release, the AIE-ML DPU will be provided as encrypted source code, enabling users to recompile the DPU kernel using the Vitis AIE compiler. Once this is available, users can integrate custom AIE functions and AMD-provided AIE libraries alongside the DPU. We refer to this workflow as "X+ML".

If you have requirements for this workflow, please contact your local AMD sales rep or email us at amd_ai_mkt@amd.com. We welcome further discussion on this topic and may potentially provide early access support.

2) It seems that the C20 DPU configuration uses 20 AIE-ML tiles per batch engine to support AI inference. If a user wants to scale up AI inference performance on a single batch beyond 20 AIE-ML tiles, what needs to be done?

The current AIE-ML DPU architecture leverages a total of 20 AIE-ML tiles per batch engine and is designed with the intent that the user can parameterize the number of batch engines as any integer between 1 and 14. The implication is that a maximum of 280 AIE-ML tiles can be supported by the AIE-ML DPU (C20 DPU, BATCH_N = 14).

Currently, a BATCH_N = 1 DPU IP that uses more than 20 tiles is not supported; however, we are open to discussions on this topic for strategic applications. Please reach out to your local sales contact if you have an application that would benefit from this capability.

3) Is the Vitis AI ONNX Runtime (VOE) a replacement for the Vitis AI quantizer and compiler tools?

No. The intent of the VOE is to enable users to take advantage of the many benefits provided by the ONNX Runtime. In the 3.0 release, the Vitis AI quantizer has been augmented to optionally export a quantized ONNX model that can be ingested by the ONNX Runtime. The Vitis AI compiler has been integrated into the ONNX Runtime to enable on-the-fly compilation of the DPU partitions within the ONNX Runtime environment.

In the 3.0 release, the Vitis AI quantizer ingests TensorFlow and PyTorch models for deployment in the ONNX Runtime. In a future release, we plan to update the Vitis AI quantizer to ingest ONNX models, enabling a complete end-to-end workflow to deploy floating-point ONNX models directly on AMD hardware targets.

The traditional workflows remain intact and will continue to be supported. For additional information, please email us at amd_ai_mkt@amd.com.

4) Does the Vitis AI ONNX Runtime Execution Provider (VOE) support a variety of Vitis AI targets, including the Versal™ AIE and AIE-ML architecture and Zynq™ UltraScale+™ devices?

With VOE we have integrated the Vitis AI compiler into the ONNX Runtime. In this context, the compiler performs on-the-fly compilation of subgraphs that are partitioned to the DPU by the ONNX Runtime. The intent of this early release of VOE is to enable users to experiment with and test this new workflow on all Vitis AI-supported Edge platforms, including the Versal AI Core Series VCK190 evaluation board, Kria KV260 Vision AI Starter Kit and the newly released Versal AI Edge Series VEK280 evaluation kit.

Today, we have tested and verified several examples found here: Vitis-AI/examples/vai_library/samples_onnx at master · Xilinx/Vitis-AI (github.com)

For additional information on this workflow, start here: Programming with VOE • Vitis AI User Guide (UG1414) • Reader • Documentation Portal (xilinx.com)

5) Are there any examples available that enable developers to quantize and compile YOLOv5 and YOLOv6? What about YOLOv7 and YOLOv8?

With the release of Vitis AI 3.0 we now have support for YOLOv5, YOLOv6, and YOLOX. YOLOv7 support development is underway with our engineering team, and YOLOv8 will follow.

It is important that users understand that the original versions of these models were released under the GNU Public License (GPL). The GPL is incompatible with Vitis AI, which is released under the more permissive Apache 2.0 license. For the above reasons, we do not release our source models with the Vitis AI Model Zoo. The use of GPL sources in commercial applications should be evaluated by the end user prior to productization.

We do provide our floating-point models and training scripts as source code to users upon request. Please reach out to amd_ai_mkt@amd.com for more details.

6) Will Vitis AI TVM (https://tvm.apache.org/) integration be updated with the new compiler improvements?

We will continue to support TVM integration as an experimental workflow into the foreseeable future.