



User Guide ZenDNN v3.0

April 2021

© 2020 - 2021 Advanced Micro Devices, Inc. All rights reserved.

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. Any unauthorized copying, alteration, distribution, transmission, performance, display, or other use of this material is prohibited.

Trademarks

AMD, the AMD Arrow logo, AMD-V, AMD Virtualization, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

Windows is a registered trademark of Microsoft Corporation.

MMX is a trademark of Intel Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Document Information

Software Version: 3.0
Document Version: 3.0
Last Updated: 21st April 2021

Contents

1	Introduction	1
2	Scope	1
3	Release Highlights	1
4	Supported OS and Compilers.....	1
5	Prerequisites.....	2
6	AOCC and AOCL (BLIS) Library Installation	2
	6.1 General Convention	2
	6.2 BLIS - AOCC Compiled.....	3
	6.3 AOCC Installation	3
7	Runtime Dependencies	3
8	Logs.....	4
9	License	5
10	Technical Support	5

1 Introduction

ZenDNN (Zen Deep Neural Network) Library accelerates deep learning inference applications on AMD CPUs. This library, which includes APIs for basic neural network building blocks optimized for AMD CPUs, targets deep learning application and framework developers with the goal of improving inference performance on AMD CPUs across a variety of workloads, including computer vision, natural language processing (NLP), and recommender systems. ZenDNN is based on oneDNN/DNNL v1.3 and leverages its basic infrastructure and APIs. However, ZenDNN optimizes several APIs and adds new APIs which are currently integrated into TensorFlow and ONNXRT, with PyTorch support available in a future ZenDNN release. ZenDNN uses AMD BLIS library for its BLAS API needs.

2 Scope

The scope of ZenDNN is to support AMD EPYC CPUs on the Linux® platform. ZenDNN v3.0 offers optimized primitives, such as Convolution, MatMul, and Elementwise, that improve performance of many convolutional neural networks, recurrent neural networks, transformer-based models, and recommender system models. Other neural network models may still be run with ZenDNN via TensorFlow/ONNXRT default paths.

3 Release Highlights

The following are the highlights of this release:

- Python version 3.7.9 has been used to generate the TensorFlow wheel file (*.whl).
- Python version 3.8.5 has been used to generate the ONNXRT wheel file (*.whl).
- **NHWC** and Blocked Format (NCHWc8) are supported
 - NHWC is the default format.
- ZenDNN library is integrated with TensorFlow v1.15 and ONNXRT v1.5.1
 - For optimal leverage of ZenDNN primitives on AMD CPUs, the above framework versions should be used. Users may encounter errors when attempting to use other versions of the above frameworks.
 - ZenDNN library is intended to be used in conjunction with the frameworks mentioned above. Users will encounter errors when attempting to use ZenDNN independently.

The latest information on the ZenDNN release and installers is available on [AMD Developer Central](#).

4 Supported OS and Compilers

This release of ZenDNN supports the following Operating Systems (OS) and compilers:

OS:

- Ubuntu® 18.04 LTS
- Ubuntu® 19.10
- Ubuntu® 20.04 LTS
- Red Hat® Enterprise Linux® (RHEL) 8.3

Compilers:

- GCC 7.5 and later
- [AOCC \(AMD Optimizing C/C++ Compiler\) 3.0](#)

- CentOS 8.3

5 Prerequisites

This release of ZenDNN has the following prerequisites:

- Conda must be installed and initialized with its path set properly so that the Conda env can be activated from the terminal.
Note: If conda is not available, `zendnn_release_setup.sh` script will fail.
- Python 3.7.9 for TensorFlow binary and Python 3.8.5 for ONNXRT binary as `whl` file must be generated with the same Python version.
Note: Python 3.8 is not supported with the TensorFlow binary.
- AOCC 3.0 and AOCL (BLIS) 3.0.6 must be installed.
Note: While GCC 7.5 and later are also supported compilers, AOCC is recommended for optimal performance of the ZenDNN library.

6 AOCC and AOCL (BLIS) Library Installation

AOCC: It is a high performance, production quality code generation tool. AOCC can be downloaded from [AMD Developer Central](#).

BLIS: It is a portable open-source software framework for instantiating high-performance Basic Linear Algebra Subprograms (BLAS), such as, dense linear algebra libraries. BLIS is part of AOCL and can be downloaded from [AMD Developer Central](#).

6.1 General Convention

The following points must be kept in mind while installing AOCC and AOCL:

- Change to the preferred directory where ZenDNN will be downloaded
- This parent folder is referred to as folder `<compdir>` in the steps below.
- It is good practice to keep AOCC 3.0 and AOCL 3.0.6 downloads in the same parent folder.
- Assume that the parent folder for user setup follows this convention: `/home/<user-id>/my_work`.

6.2 BLIS - AOCC Compiled

Complete the following steps:

1. Execute the command `cd <compdir>`
2. Download `aocl-linux-aocc-3.0-6.tar.gz`
3. Execute the following commands:
 - a. `tar -xvf aocl-linux-aocc-3.0-6.tar.gz`
 - b. `cd aocl-linux-aocc-3.0-6`
 - c. `tar -xvf aocl-blis-linux-aocc-3.0-6.tar.gz`
 - d. `cd amd-blis`

This will set up the environment for BLIS AOCC path:

```
export ZENDNN_BLIS_PATH=$(pwd)
```

For example, `export ZENDNN_BLIS_PATH=/home/<user-id>/my_work/aocl-linux-aocc-3.0-6/amd-blis`

6.3 AOCC Installation

Complete the following steps to install AOCC:

1. Execute the command `cd <compdir>`
2. Download `aocc-compiler-3.0.0.tar` from the [AMD developer portal](#).
3. Execute the command `tar -xvf aocc-compiler-3.0.0.tar`
4. Execute the command `cd aocc-compiler-3.0.0`
5. Execute the command `bash install.sh`

This will install the compiler and display the AOCC set up instruction.

This will set up the environment for the AOCC path

```
export ZENDNN_AOCC_COMP_PATH=$(pwd)
```

For example, `export ZENDNN_AOCC_COMP_PATH=/home/<user-id>/my_work/aocc-compiler-3.0.0`

The `bashrc` file can be edited to setup `ZENDNN_AOCC_COMP_PATH` env path.

For example:

```
export ZENDNN_AOCC_COMP_PATH=/home/<user-id>/my_work/aocc-compiler-3.0.0
```

```
export ZENDNN_BLIS_PATH=/home/<user-id>/my_work/aocl-linux-aocc-3.0-6/amd-blis
```

7 Runtime Dependencies

To use ZenDNN, the following runtime libraries must be installed:

- GNU C library (glibc.so)
- GNU Standard C++ library (libstdc++.so)
- Dynamic linking library (libdl.so)
- POSIX Thread library (libpthread.so)
- C Math Library (libm.so)
- OpenMP (libomp.so)
- Python 3.7.9 for TensorFlow and Python 3.8.5 for ONNXRT

Since ZenDNN is configured to use OpenMP, a C++ compiler with OpenMP 2.0 or later is required for runtime execution.

8 Logs

Logging is disabled in the ZenDNN library by default. It can be enabled using the environment variable **ZENDNN_LOG_OPTS** before running any tests. Logging behavior can be specified by setting the environment variable **ZENDNN_LOG_OPTS** to a comma-delimited list of **ACTOR:DBGLVL** pairs.

For example:

- To turn on info logging, use **ZENDNN_LOG_OPTS=ALL:2**
- To turn off all logging, use **ZENDNN_LOG_OPTS=ALL:-1**
- To only log on errors, use **ZENDNN_LOG_OPTS=ALL:0**
- To only log info for ALGO, use **ZENDNN_LOG_OPTS=ALL:-1,ALGO:2**
- To only log info for CORE, use **ZENDNN_LOG_OPTS=ALL:-1,CORE:2**
- To only log info for API, use **ZENDNN_LOG_OPTS=ALL:-1,API:2**
- To only log info for PROF (profile), use **ZENDNN_LOG_OPTS=ALL:-1,PROF:2**
- To only log info for GRAPH, use **ZENDNN_LOG_OPTS=ALL:-1,GRAPH:2**

The different ACTORS are as follows:

- ALGO
- CORE
- API
- TEST
- PROF
- GRAPH

The Different Debug Levels (DBGLVL) are as follows:

```
enum LogLevel {  
    LOG_LEVEL_DISABLED = -1,  
    LOG_LEVEL_ERROR = 0,  
    LOG_LEVEL_WARNING = 1,  
    LOG_LEVEL_INFO = 2,  
};
```

9 License

Please refer to the AMD ZenDNN EULA for more details on the third-party programs used in this release and their corresponding licenses. Upon selecting a ZenDNN package to download, Developer Central will redirect users to said AMD ZenDNN EULA document. The terms and conditions set forth in the EULA must be accepted before the download will commence.

This distribution includes the following third-party softwares, which are governed by separate license terms described in the AMD ZenDNN EULA:

- [oneDNN](#)
- [Xbyak](#)
- [Boost C++ Libraries](#)
- [Tensorflow](#)
- [Caffe](#)
- [gtest](#)

10 Technical Support

Please email zendnnsupport@amd.com for questions, issues, and feedback regarding ZenDNN.