AIXPRT Community Preview user guide

Table of Contents

Introduction

AIXPRT has the capability of running on the following platforms:

- AMD CPU
- AMD GPU (ROCm supported devices)
- Intel CPU
- Intel GPU (OpenVINO supported devices)
- Intel Neural Compute Stick
- Intel HDDL-R Accelerator
- NVIDIA GPU (CUDA supported devices)

The workloads are implemented using the publicly available libraries and SDKs for each platform.

Getting started (all network models)

The AIXPRT installation package is available via a compressed file at AIXPRT.com (testers may request access here), as well as on GitHub. The first step after receiving access to AIXPRT on GitHub is to clone the AIXPRT repository.

1. Download the AIXPRT repository. Since the repository contains large files, over 50MB in size, the git lfs package must be installed and the repository must be cloned (A zip file of the repository will not include the large files).

    a. Install git lfs. The instructions are found at https://packagecloud.io/github/git-lfs/install and are listed in the following 2 steps.

    ```
    curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
    sudo apt-get install git-lfs
    ```

    b. Clone the AIXPRT repository (You may need to enter your credentials for each large file).

    ```
    git clone https://github.com/BenchmarkXPRT/AIXPRT.git
    ```

2. Navigate to AIXPRT/Modules/Deep-Learning/README.md of the cloned branch and follow instructions to run the benchmark.

3. AIXPRT runs on a default configuration provided by the workloads. However, once a default config file is generated, user can edit the config to change workload behavior.

    NOTE: A config file will be generated at AIXPRT/Config/{config_name}.json after the first run of index.py. One can edit this config file to run the specific workload in a specific way or use one of the provided config files.

Intel OpenVINO (CPU and GPU) on Windows

This module contains workloads to evaluate the system performance of use cases related to image classification and object detection using the Intel Distribution of OpenVINO toolkit for Windows. It contains the resnet50_v1 and ssd-mobilenet workloads and runs single and multi-batch size scenarios.

System requirements

Operating System

- Windows 10

CPU

- 6th to 10th generation Intel Core and Intel Xeon processors
- Intel Pentium processor N4200/5, N3350/5, N3450/5 with Intel HD Graphics

Installation and system configuration

1. Install all dependencies

   a. Register on the Intel OpenVINO toolkit page (link below) and download the OpenVINO package for Windows.

      https://software.intel.com/en-us/openvino-toolkit/choose-download/free-downloadwindows

   b. Install the full package version of Intel OpenVINO by following the instructions at https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_windows.html.

   c. Install pillow, numpy, pywin32, wmi, and opencv using the following script:

```
pip install pillow numpy pywin32 wmi opencv-python
```

Building the workloads

1. Make sure you are connected to an active internet connection.
2. Run the AIXPRT install and compile scripts:

```
cd AIXPRT/install compile_AIXPRT_sources.bat <path/to/AIXPRT>
<path/to/OpenVINO>
```

NOTE: In Windows, the “python3” command is not recognized by default. Please make a copy of python.exe (located at the python installation path) and rename it as python3.exe.

Running the benchmark

1. Navigate to the AIXPRT harness directory:

```
cd AIXPRT/Harness
```

2. Run the benchmark using the following script:

```
python3 index.py
```

3. If targeting the GPU, edit AIXPRT/Config/{filename.json} to set “hardware” to GPU.

Precompiled Intel OpenVINO (CPU and GPU) on Windows

To simplify the installation process for some testers, we offer an AIXPRT download package with a precompiled version of the Intel Distribution of OpenVINO toolkit for Windows. This package only runs OpenVINO and does not contain the TensorFlow and TensorFlow-TensorRT frameworks. It contains the resnet50_v1 and ssd-mobilenet workloads and runs single and multi-batch size scenarios.

System requirements

Operating System

- Windows 10

CPU

- 6th to 10th generation Intel Core and Intel Xeon processors
- Intel Pentium processor N4200/5, N3350/5, N3450/5 with Intel HD Graphics

Installation and system configuration

1. Download and unzip the installation package.
2. Open a Command Prompt in Windows.
3. Navigate to AIXPRT_CP2_OpenVINO\install, and run the following command, which will install all dependencies and prepare the benchmark to run:

```
setup_AIXPRT.bat
```

Running the benchmark

1. Navigate to the AIXPRT harness directory:

```
cd AIXPRT/Harness
```

2. Run the benchmark using the following script:

```
python3 index.py
```

3. If targeting the GPU, edit AIXPRT/Config/{filename.json} to set “hardware” to GPU.

Intel OpenVINO (CPU and GPU) on Linux

This module contains workloads to evaluate the system performance of use cases related to image classification and object detection using the Intel Distribution of OpenVINO toolkit for Linux. It contains the resnet50_v1 and ssd-mobilenet workloads and runs single and multi-batch size scenarios.

System requirements

Operating System

- Ubuntu 16.04 LTS (64-bit)

CPU

- 6th to 10th generation Intel Core and Intel Xeon processors

– Intel Pentium processor N4200/5, N3350/5, N3450/5 with Intel HD Graphics

Installation and system configuration

1. Install all dependencies

   a. Register on the Intel OpenVINO toolkit page (link below) and download the OpenVINO for Linux toolkit. https://software.intel.com/en-us/openvino-toolkit/choose-download/free-download-linux

   b. Install the full package version of Intel OpenVINO by following the instructions at https://software.intel.com/en-us/articles/OpenVINO-Install-Linux.

   c. Install pillow using the following script:

   ```
   sudo apt-get install python-pip python-numpy pip
   install Pillow
   ```

Building the workloads

1. Make sure you are connected to an active internet connection.

2. Grant the necessary executable permissions to the compile_AIXPRT_sources.sh file which is located at ~AIXPRT/install:

   ```
   chmod 777 compile_AIXPRT_sources.sh
   ```

3. Run the AIXPRT install script:

   ```
   cd AIXPRT/install

   sudo ./compile_AIXPRT_sources.sh <path/to/AIXPRT> <path/to/OpenVINO>
   ```

Running the benchmark

1. Navigate to the AIXPRT harness directory:

   ```
   cd AIXPRT/Harness
   ```

2. Run the benchmark using the following script:

   ```
   python3 index.py
   ```

TensorFlow (CPU) on Windows

This module contains workloads to evaluate the system performance of use cases related to image classification and object detection using TensorFlow. It contains the resnet50_v1 and ssd-mobilenet workloads and runs single and multi-batch size scenarios.

Workloads are built and tested using the TensorFlow (version 1.12) framework. For more information about TensorFlow please go to https://www.tensorflow.org. Workloads run with fp32 precision by default.

System requirements

Operating System

- Windows 10 and above

CPU

- Please consult the TensorFlow website for minimum hardware and software requirement information: https://www.tensorflow.org/install.

Installation and system configuration

1. Install Python3 for Windows.

   https://www.python.org/downloads/windows/

2. Navigate to the Python3 installation directory and duplicate the python.exe file. Rename the duplicated file to python3.exe.

3. Install all dependencies.

   ```
   pip3 install pyreadline pip3
   install Pillow pip3 install
   opencv-python pip3 install –
   upgrade wmi pip3 install –
   upgrade pypiwin32
   ```

4. Install TensorFlow.

   - For Intel and AMD CPUs, please read the instructions here:

     https://www.tensorflow.org/install/pip.

   - To install TensorFlow with AMD ROCm support, follow the instructions here:

     https://rocm.github.io/dl.html.

Running the benchmark

1. Navigate to the AIXPRT harness directory:

   ```
   cd AIXPRT/Harness
   ```

2. Run the benchmark using the following script:

   ```
   python3 index.py
   ```

3. If running on a GPU target, please edit the AIXPRT/Config/{filename.json} to set "hardware" to GPU.

TensorFlow (CPU and GPU) on Linux

This module contains workloads to evaluate the system performance of use cases related to image classification and object detection using TensorFlow. It contains the resnet50_v1 and ssd-mobilenet workloads and runs single and multi-batch size scenarios.

Workloads are built and tested using the TensorFlow (version 1.12) framework. For more information about TensorFlow please go to https://www.tensorflow.org. Workloads run with fp32 precision by default.

System requirements

Operating System

- Ubuntu 16.04 LTS (64-bit) and above

CPU and GPU

- Please consult the TensorFlow website for minimum hardware and software requirement information: https://www.tensorflow.org/install.

Installation and system configuration

1. Install all dependencies.

   ```
   sudo apt-get update sudo apt-get install python3
   python3-numpy python3-pil sudo apt-get install python
   python-numpy python-pil
   ```

2. Install TensorFlow.

   - For Intel and AMD CPUs, please read the instructions here: https://www.tensorflow.org/install.

     ```
     sudo apt-get install python-pip pip
     install tensorflow
     ```

   - For NVIDIA GPUs, follow the instructions here: https://www.tensorflow.org/install/gpu.

     a. Remove any installed drivers.

     ```
     sudo apt-get purge nvidia*
     ```

     b. Retrieve the repository.

     ```
     sudo add-apt-repository ppa:graphics-drivers/ppa
     ```

     c. Update the apt request.

     ```
     sudo apt update
     ```

     d. Confirm that the desired driver is present.

     ```
     ubuntu-drivers devices
     ```

     e. Install the desired driver.

     ```
     sudo apt install nvidia-410 reboot
     ```

     f. Verify that the desired driver is installed.

     ```
     nvidia-smi

     sudo apt-get install python-pip pip install tensorflow-gpu==1.12.0
     ```

- To install TensorFlow with ROCm support, follow the instructions here:

  https://rocm.github.io/dl.html.

  NOTE: For AMD-GPUs, Ubuntu 18.04 supports the latest drivers, so we recommend using Ubuntu 18.04 in those instances.

Running the benchmark

1. Navigate to the AIXPRT harness directory:

```
cd AIXPRT/Harness
```

2. Run the benchmark using the following script:

```
python3 index.py
```

3. If running on a GPU target, please edit the AIXPRT/Config/{filename.json} to set "hardware" to GPU.

TensorFlow with NVIDIA TensorRT (GPU)

This module contains workloads to evaluate the system performance of use cases related to Image Classification and Object Detection using TensorFlow with TensorRT optimizations. It contains the resnet50_v1 and ssd-mobilenet workloads and runs single and multi-batch size scenarios.

System requirements

Operating System

- Ubuntu 16.04 LTS (64-bit) and above

GPU

- CUDA-enabled NVIDIA GPUs. For more info, visit https://developer.nvidia.com/cuda-gpus.

For NVIDIA Tegra Xavier

- JetPack 4.1.1

Installation and system configuration

Install all dependencies

1. For NVIDIA discrete graphics

   a. Install CUDA 10 from https://developer.nvidia.com/cuda-downloads.

   b. Restart the system.

   c. Install docker from https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-from-apackage. Run lsb_release -a to show you the Ubuntu version and name. You will also need to install the cli and container packages.

   d. Test the installation of the docker image.

   ```
   sudo docker run hello-world
   ```

e. Run the following command to be able to run as a regular user. If it does not work on the first try, reboot the system.

```
sudo usermod -a -G docker $USER
```

f. Install NVIDIA-docker from https://github.com/NVIDIA/nvidia-docker.

g. Pull and run TensorRT docker container v19.04

```
docker pull nvcr.io/nvidia/tensorflow:19.04-py3
```

2. If using NVIDIA Tegra Xavier, flash JetPack v4.2 from https://developer.nvidia.com/embedded/JetPack.

Running the benchmark

1. Choose the target machine and run the following commands:

a. If using NVIDIA discrete graphics, run the docker image.

```
nvidia-docker run -v <Path_to_AIXPRT_directory>:/workspace/AIXPRT --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 -it --rm nvcr.io/nvidia/tensorflow:19.04-py3
```

b. If using NVIDIA Tegra Xavier, install the necessary prerequisites:

```
sudo apt-get install libhdf5-serial-dev hdf5-tools

sudo apt-get install python3-pip

sudo apt-get install zlib1g-dev zip libjpeg8-dev libhdf5-dev

sudo pip3 install -U numpy grpcio absl-py py-cpuinfo psutil portpicker grpcio six mock requests gast h5py astor termcolor

sudo pip3 install -U numpy grpcio absl-py py-cpuinfo psutil portpicker grpcio six mock requests gast h5py astor termcolor

pip3 install --extra-index-url https://developer.download.nvidia.com/compute/redist/jp/v42 tensorflow-gpu

sudo apt-get install python3-matplotlib
```

2. Add the models directory to PYTHONPATH to install tensorflow/models and Run the TF Slim setup.

3. Run script setup.sh inside /workspace/AIXPRT.

a. If the script fails, run the following commands manually:

```
git clone https://github.com/tensorflow/models.git

cd models

export PYTHONPATH="$PYTHONPATH:$PWD"

cd research

export PYTHONPATH="$PYTHONPATH:$PWD"

For x86:
```

```
wget -O protobuf.zip

https://github.com/google/protobuf/releases/download/v3.0.0/protoc-3.0.0-linux-x86_64.zip
```

For Tegra:

```
wget -O protobuf.zip

https://github.com/protocolbuffers/protobuf/releases/download/v3.7.1/protoc-3.7.1-linux-aarch_64.zip

unzip protobuf.zip

./bin/protoc object_detection/protos/*.proto --python_out=.

cd slim

python setup.py install

pip install requests pillow
```

4. Navigate to the AIXPRT harness directory:

```
cd AIXPRT/Harness
```

5. Run the benchmark using the following script:

```
python3 index.py
```

Results

When the test is complete, the benchmark saves results to AIXPRT/Results. To submit results, please follow the instructions in ResultSubmission.md or at https://www.principledtechnologies.com/benchmarkxprt/aixprt/2019/submit-results.php.

Support

If you need technical support or have any questions, please send a message to BenchmarkXPRTsupport@principledtechnologies.com.