



TensorFlow-HRT

User Manual

2017-12-25

OPEN AI LAB

Reversion Record

Date	Rev	Change Description	Author
2017-12-25	0.1.0	Initial	Yuming Cheng Yu Wang
2018-02-08	0.1.1	Add Alexnet test	Yuming Cheng

catalog

1 PURPOSE	3
2 TERMINOLOGY	3
3 ENVIRONMENT	3
3.1 HARDWARE PLATFORM	3
3.2 SOFTWARE PLATFORM	4
4 INSTALL GUIDE	4
4.1 DOWNLOAD CODE	4
4.2 SETUP DEVELOPMENT ENVIRONMENT.....	4
4.2.1 Setup build server	4
4.2.2 Setup RK3399 device	5
4.3 COMPILE ACL ON BUILD SERVER	5
4.4 COMPILE TENSORFLOW ON BUILD SERVER	6
4.4.1 Update cross build setting	6
4.4.2 Run Tensorflow configure	6
4.4.3 Build pip package.....	7
4.5 COMPILE TENSORFLOW LABEL_IMAGE	9
4.6 RUN TESTS	9
5 CONFIGURATION GUIDE.....	12
5.1 ENABLE ACL IN COMPILE TIME.....	12
5.2 BYPASS ACL IN RUN TIME	12

1 Purpose

This guide help developers to use the code of TensorFlow-HRT (TensorFlow Heterogeneous Run Time) to improve the performance of their applications based on the TensorFlow framework.

2 Terminology

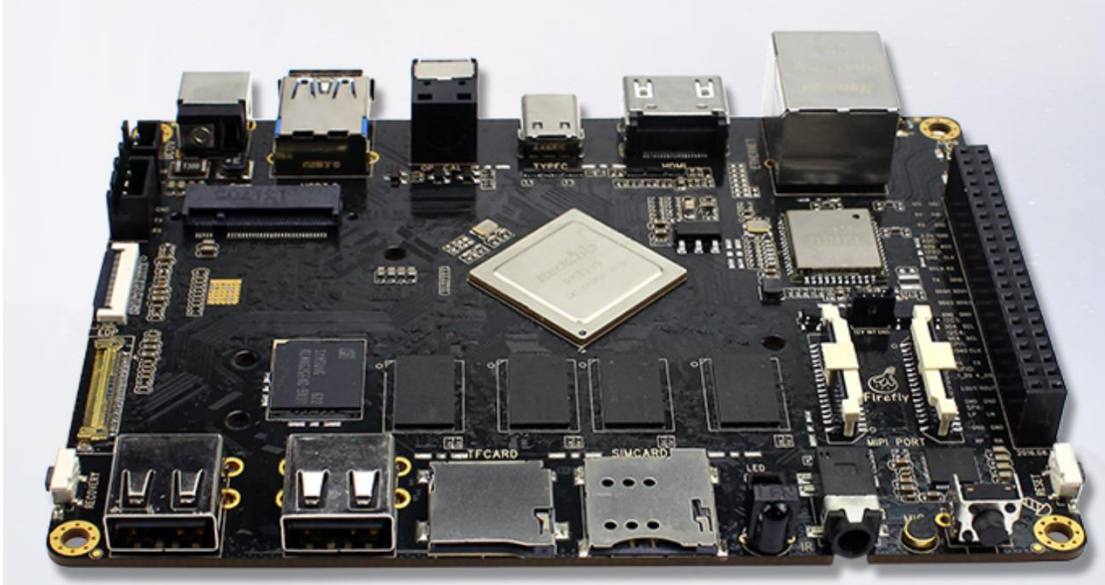
- ✧ **ACL:** Arm Compute Library
- ✧ **TensorFlow:** A deep learning library or framework.
- ✧ **TensorFlow-HRT:** Tensorflow Heterogeneous Run Time.
- ✧ **ACL/GPU:** In the below tables, it is specialized to mean using GPU by Arm Compute Library to test. (Mali: GPU from Arm)
- ✧ **ACL/Neon:** In the below tables, it is specialized to mean using Neon by Arm Compute Library to test. (Neon: ARM coprocessor supporting SIMD)
- ✧ **NCHW, NHWC:** Tensor format for input/output activations used in convolution operations.
The mnemonics specify the meaning of each tensor dimension sorted from largest to smallest memory stride. N = Batch, H = Image Height, W = Image Width, C = Number of Channels. NHWC is the default format in TensorFlow. NCHW often improves performance on GPUs.
- ✧ **OIHW, HWIO:** Tensor format for convolutional filters. The mnemonics specify the meaning of each tensor dimension sorted from largest to smallest memory stride. H = Kernel Height, W = Kernel Width, I = Input Channels, O = Output Channels. HWIO is the default filter format in TensorFlow. OIHW often improves performance on GPUs.

3 Environment

3.1 Hardware Platform

SoC: Rockchip RK3399

- ✧ GPU: Mali T864 (800MHz)
- ✧ CPU: Dual-core Cortex-A72 up to 2.0GHz (real frequency is 1.8GHz); Quad-core Cortex-A53 up to 1.5GHz (real frequency is 1.4GHz)



3.2 Software platform

Operating System: Ubuntu 16.04

4 Install Guide

4.1 Download code

```
git clone https://github.com/ARM-software/ComputeLibrary.git  
git clone https://github.com/OAID/TensorFlow-HRT.git
```

4.2 Setup Development Environment

4.2.1 Setup build server

1. Update build server to add arm64 architecture and backup apt sources.list.

```
sudo dpkg --add-architecture arm64  
sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak
```

2. create file [tsinghua.list](#) in /etc/apt/sources.list.d/ with following content.

```
deb [arch=amd64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial main restricted  
universe multiverse
```

```
deb [arch=amd64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-updates main
restricted universe multiverse
```

```
deb [arch=amd64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-security main
restricted universe multiverse
```

```
deb [arch=amd64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ xenial-backports main
restricted universe multiverse
```

```
deb [arch=arm64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ xenial main restricted
universe multiverse
```

```
deb [arch=arm64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ xenial-updates main
restricted universe multiverse
```

```
deb [arch=arm64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ xenial-security main
restricted universe multiverse
```

```
deb [arch=arm64] https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ xenial-backports main
restricted universe multiverse
```

3. Install crossbuild tools

```
sudo apt-get update
sudo apt-get install gcc-aarch64-linux-gnu g++-aarch64-linux-gnu scons rsync
sudo apt-get install python-numpy python-dev python-pip python-wheel
sudo apt-get install libpython2.7-dev:arm64
```

4. Install bazel (release 0.7.0 or above)

Tensorflow-HRT uses bazel to build. If bazel is not installed on your system, install it now by following guide at <https://docs.bazel.build/versions/master/install.html>.

4.2.2 Setup RK3399 device

```
sudo apt-get update
sudo apt-get install python-numpy python-scipy python-dev python-pip python-wheel rsync
```

Insert a USB thumb drive into Firefly USB port, create and active a swap partition (size >= 2GB) on ubuntu.

(<https://askubuntu.com/questions/103043/how-to-create-swap-partition-on-already-installed-ubuntu>)

4.3 Compile ACL on build server

```
cd ~/ComputeLibrary
git checkout 6bc7b9046ae6ed4e4b574675e0c597b5d39a7423
mkdir build
aarch64-linux-gnu-gcc openc1-1.2-stubs/openc1_stubs.c -linclude -shared -o
build/libOpenCL.so
```

```
scons Werror=1 -j8 debug=0 asserts=1 neon=1 openc1=1 embed_kernels=1 os=linux
arch=arm64-v8a
```

If no AID tools installed on Firefly, copy following 3 files in ACL build directory to Firefly /usr/lib/aarch64-linux-gnu/. Otherwise copy to Firefly /usr/local/AID/ComputeLibrary/lib/.

```
libarm_compute_core.so
libarm_compute_graph.so
libarm_compute.so
```

4.4 Compile Tensorflow-HRT on build server

4.4.1 Update cross build setting

Modify following 3 lines in “Tensorflow-HRT/tools/aarch64_compiler/CROSSTOOL” to ACL path in build server.

```
35: linker_flag: "-L/home/[user name]/ComputeLibrary/build"
36: linker_flag: "-L/home/[user name]/ComputeLibrary/build/openc1-1.2-stubs/"
44: cxx_builtin_include_directory: "/home/[user name]/ComputeLibrary"
```

4.4.2 Run Tensorflow-HRT configure

```
cd ~/Tensorflow-HRT
./configure
WARNING: Running Bazel server needs to be killed, because the startup options are
different.
You have bazel 0.7.0 installed.
Please specify the location of python. [Default is /usr/bin/python]:

Found possible Python library paths:
  /usr/local/lib/python2.7/dist-packages
  /usr/lib/python2.7/dist-packages
Please input the desired Python library path to use.  Default is [/usr/local/lib/python2.7/dist-
packages]

Do you wish to build TensorFlow with jemalloc as malloc support? [Y/n]:
jemalloc as malloc support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Google Cloud Platform support? [Y/n]: n
No Google Cloud Platform support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Hadoop File System support? [Y/n]: n
No Hadoop File System support will be enabled for TensorFlow.
```

Do you wish to build TensorFlow with Amazon S3 File System support? [Y/n]: n
No Amazon S3 File System support will be enabled for TensorFlow.

Do you wish to build TensorFlow with XLA JIT support? [y/N]: n
No XLA JIT support will be enabled for TensorFlow.

Do you wish to build TensorFlow with GDR support? [y/N]: n
No GDR support will be enabled for TensorFlow.

Do you wish to build TensorFlow with VERBS support? [y/N]: n
No VERBS support will be enabled for TensorFlow.

Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]: n
No OpenCL SYCL support will be enabled for TensorFlow.

Do you wish to build TensorFlow with CUDA support? [y/N]: n
No CUDA support will be enabled for TensorFlow.

Do you wish to build TensorFlow with MPI support? [y/N]: n
No MPI support will be enabled for TensorFlow.

Please specify optimization flags to use during compilation when bazel option "--config=opt" is specified [Default is -march=native]: -march=arm64-v8a

Add "--config=mkl" to your bazel command to build with MKL support.

Please note that MKL on MacOS or windows is still not supported.

If you would like to use a local MKL instead of downloading, please set the environment variable "TF_MKL_ROOT" every time before build.

Configuration finished

4.4.3 Build pip package

NOTE : needs to change ComputeLibrary path in following build cmd to its actual location

```
cd ~/Tensorflow-HRT
```

```
bazel build -c opt \
--cxxopt=-fexceptions \
--copt="-I/home/[user name]/ComputeLibrary/" \
--copt="-I/home/[user name]/ComputeLibrary/include" \
--cpu=aarch64 --crosstool_top=//tools/aarch64_compiler:toolchain \
--host_crosstool_top=@bazel_tools//tools/cpp:toolchain \
--host_copt="-DUSE_ACL=1" \
```



```

--copt="-DUSE_ACL=1" \
--copt="-DTEST_ACL=1" \
--copt="-DUSE_PROFILING=1" \
--incompatible_load_argument_is_label=false \
--verbose_failures \
//tensorflow/tools/pip_package:build_pip_package

bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/install/

mv /tmp/install/tensorflow-1.4.0-cp27-cp27mu-linux_x86_64.whl \
    tensorflow-1.4.0-cp27-cp27mu-linux_aarch64.whl

```

If there is a following build error:

```

/home/xxxx/.cache/bazel/_bazel_xxxx/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/external/nsync
c/BUILD:402:13: Configurable attribute "copts" doesn't match this configuration (would a
default condition help?).
Conditions checked:
  @nsync//:android_arm
...
  @nsync//:msvc_windows_x86_64.

```

Solution:

Open

“/home/xxx/.cache/bazel/_bazel_xxxx/xx/external/nsync/BUILD”, Add a default conditions there as marked as red line below:

```

NSYNC_OPTS_GENERIC = select({
  # Select the CPU architecture include directory.
  # This select() has no real effect in the C++11 build, but satisfies a
  # #include that would otherwise need a #if.
  ":gcc_linux_x86_64_1": ["-I" + pkg_path_name() + "/platform/x86_64"],
  ":gcc_linux_x86_64_2": ["-I" + pkg_path_name() + "/platform/x86_64"],
  ":gcc_linux_aarch64": ["-I" + pkg_path_name() + "/platform/aarch64"],
  ":gcc_linux_ppc64": ["-I" + pkg_path_name() + "/platform/ppc64"],
  ":clang_macos_x86_64": ["-I" + pkg_path_name() + "/platform/x86_64"],
  ":ios_x86_64": ["-I" + pkg_path_name() + "/platform/x86_64"],
  ":android_x86_32": ["-I" + pkg_path_name() + "/platform/x86_32"],
  ":android_x86_64": ["-I" + pkg_path_name() + "/platform/x86_64"],
  ":android_armeabi": ["-I" + pkg_path_name() + "/platform/arm"],
  ":android_arm": ["-I" + pkg_path_name() + "/platform/arm"],
  ":android_arm64": ["-I" + pkg_path_name() + "/platform/aarch64"],
  ":msvc_windows_x86_64": ["-I" + pkg_path_name() + "/platform/x86_64"],
  "//conditions:default": [],
}) + [

```

4.5 Compile Tensorflow-HRT label_image

Type the following commands:

```
cd ~/Tensorflow-HRT

bazel build -c opt \
--incompatible_load_argument_is_label=false \
--cxxopt=-fexceptions --copt="-I/home/[user name]/ComputeLibrary/" \
--copt="-I/home/[user name]/ComputeLibrary/include" \
--cpu=aarch64 \
--crosstool_top="//tools/aarch64_compiler:toolchain" \
--host_crosstool_top="@bazel_tools//tools/cpp:toolchain" \
--copt="-DUSE_ACL=1" --copt="-DTEST_ACL=1" \
--copt="-DUSE_PROFILING=1" \
--verbose_failures //tensorflow/examples/label_image/...
```

4.6 Run Tests

1. Copy and install build files

```
ssh firefly@192.168.3.211 "mkdir -p test/tensorflow/examples/label_image/data/"

cd ~/Tensorflow-HRT

scp acl_openailab/test/* firefly@192.168.3.211:/home/firefly/test/
scp tensorflow-1.4.0-cp27-cp27mu-linux_aarch64.whl \
firefly@192.168.3.211:/home/firefly/test/

scp tensorflow/python/kernel_tests/acl*.py firefly@192.168.3.211:/home/firefly/test/

scp bazel-bin/tensorflow/examples/label_image/label_image \
firefly@192.168.3.211:/home/firefly/test/

scp tensorflow/examples/label_image/data/grace_hopper.jpg \
firefly@192.168.3.211:/home/firefly/test/tensorflow/examples/label_image/data/

ssh firefly@192.168.3.211
sudo pip install tensorflow-1.4.0-cp27-cp27mu-linux_aarch64.whl
```

2. Run label_image with inception graph

```
ssh firefly@192.168.3.211

cd test
```

```
tar -C tensorflow/examples/label_image/data \
-xvzf inception_v3_2016_08_28_frozen.pb.tar.gz
```

```
LD_LIBRARY_PATH=/usr/local/lib/python2.7/dist-packages/tensorflow/ ./label_image
```

output message :

```
2018-02-09 02:11:40.789511: I tensorflow/core/kernels/acl_ops_common.cc:72] LOGACL:
ACL_CONV      :                               0.102086
ACL_RELU      :                               0.009732
...
ACL_RELU      :                               0.000179
ACL_CONV      :                               0.073248
ACL_RELU      :                               0.000197
ACL_POOLING   :                               0.000184
ACL_CONV      :                               0.351716
ACL_SOFTMAX   :                               0.001314
2018-02-09 02:11:46.382731: I tensorflow/examples/label_image/main.cc:250] military
uniform (653): 0.834305
2018-02-09 02:11:46.382821: I tensorflow/examples/label_image/main.cc:250] mortarboard
(668): 0.0218695
2018-02-09 02:11:46.382844: I tensorflow/examples/label_image/main.cc:250] academic
gown (401): 0.0103581
2018-02-09 02:11:46.382866: I tensorflow/examples/label_image/main.cc:250] pickelhaube
(716): 0.00800819
2018-02-09 02:11:46.382894: I tensorflow/examples/label_image/main.cc:250] bulletproof
vest (466): 0.00535092
```

3. Run label_image with Alexnet graph (Need to active swap partition)

```
ssh firefly@192.168.3.211
cd test
python label_image_foralexnet.py --graph=alexnet.pb --labels=synset.txt \
--input_width=227 --input_height=227 --input_layer=Placeholder \
--output_layer=prob --image=cat.jpg
```

output message :

```
2018-02-09 02:09:53.097309: I tensorflow/core/kernels/acl_ops_common.cc:62]
BYPASSACL:
2018-02-09 02:09:53.097347: I tensorflow/core/kernels/acl_ops_common.cc:72] LOGACL:
ACL_CONV      :                               0.187473
ACL_RELU      :                               0.008270
ACL_LRN       :                               0.037023
ACL_CONV      :                               0.120347
ACL_CONV      :                               0.111242
ACL_RELU      :                               0.001671
ACL_LRN       :                               0.013221
```

ACL_CONV	:		0.190410
ACL_RELU	:	0.000412	
ACL_CONV	:		0.067692
ACL_CONV	:		0.068334
ACL_RELU	:	0.000463	
ACL_CONV	:		0.045856
ACL_CONV	:		0.046772
ACL_RELU	:	0.000315	
ACL_RELU	:	0.000074	
ACL_RELU	:	0.000072	
ACL_SOFTMAX:		0.003691	
n02123394 Persian cat		0.404074	
n02127052 lynx, catamount		0.214068	
n02124075 Egyptian cat		0.0915406	
n02123045 tabby, tabby cat		0.0688691	
n02441942 weasel		0.0488651	

4. Run Unit test

```
ssh firefly@192.168.3.211
```

```
cd test
ls acl*.py | xargs -l {} python {}
```

output message :

```
testInceptionFwd_0 [4, 5, 5, 124] [1, 1, 124, 12] [4, 5, 5, 12] 1
testInceptionFwd_1 [4, 8, 8, 38] [1, 1, 38, 38] [4, 8, 8, 38] 1
testInceptionFwd_2 [4, 8, 8, 38] [1, 1, 38, 38] [4, 8, 8, 38] 1
...
2018-02-09 01:52:14.254189: I tensorflow/core/kernels/acl_ops_common.cc:72] LOGACL:
ACL_CONV      :
0.000292
2018-02-09 01:52:14.257272: I tensorflow/core/kernels/acl_ops_common.cc:62]
BYPASSACL:
2018-02-09 01:52:14.257364: I tensorflow/core/kernels/acl_ops_common.cc:72] LOGACL:
..
-----
Ran 62 tests in 4.272s
....
2018-02-09 01:52:35.640209: I tensorflow/core/kernels/acl_ops_common.cc:62]
BYPASSACL:
2018-02-09 01:52:35.640311: I tensorflow/core/kernels/acl_ops_common.cc:72] LOGACL:
2018-02-09 01:52:35.651992: I tensorflow/core/kernels/acl_ops_common.cc:62]
BYPASSACL:
2018-02-09 01:52:35.652097: I tensorflow/core/kernels/acl_ops_common.cc:72] LOGACL:
..
-----
```

```
Ran 2 tests in 0.071s
```

```
OK
```

5 Configuration Guide

5.1 Enable ACL In Compile Time

- ✧ Enable ACL functions by “-DUSE_ACL=1” in bazel build command
- ✧ Disable it without define “USE_ACL” in build command.

The Tensorflow-HRT disable ACL by default.

5.2 Bypass ACL In Run Time

- ✧ Set environment variable BYPASSACL

Type following command to skip ACL in runtime.

```
export BYPASSACL=364
```

Type following command to use ACL in runtime for ACL enabled build.

```
unset BYPASSACL
```