STEPS TO INSTALL TENSORFLOW IN RASBERRY PI WITHOUT BAZEL WITH WHEEL FILE:

```
$ sudo apt-get install -y libhdf5-dev libc-ares-dev libeigen3-dev gcc gfortran libgfortran5 \libatlas3-base libatlas-base-dev libopenblas-dev libopenblas-base libblas-dev \liblapack-dev cython3 libatlas-base-dev openmpi-bin libopenmpi-dev python3-dev

$ sudo pip3 install pip --upgrade

$ sudo pip3 install keras_applications==1.0.8 --no-deps

$ sudo pip3 install keras_preprocessing==1.1.0 --no-deps

$ sudo pip3 install numpy==1.20.3

$ sudo pip3 install h5py==3.1.0

$ sudo pip3 install pybind11

$ pip3 install -U --user six wheel mock
$wget"https://raw.githubusercontent.com/PINTO0309/Tensorflow-bin/master/tensorflow-2.5.0-cp37-none-linux_armv7l_download.sh"
$ ./tensorflow-2.5.0-cp37-none-linux_armv7l_download.sh
$ sudo pip3 uninstall tensorflow
$ sudo -H pip3 install tensorflow-2.5.0-cp37-none-linux_armv7l.whl

【Required】 Restart the terminal.
```

The code below I haven't tested yet the code above led to the installation of TF.

Native build procedure of Tensorflow v2.0.0 C API for RaspberryPi / arm64 devices (armhf / aarch64)

This section describes how to build Tensorflow v2.0.0 C API (RaspberryPi / arm64 devices). Since we get an error when we build according to the **official tutorial**, I devised my own procedure. In particular, when using armhf (armv7l), an error `undefined symbol: __atomic_fetch_add_8` will occur, so an option must be added to the build script. My procedure can avoid `undefined symbol: __atomic_fetch_add_8` errors and complete the build successfully. When using RaspberryPi4, it will be completed in about 6 hours. Prebuilt binaries and installers can be downloaded from my **Github repository**.

Environment:

- RaspberryPi3 / 4 (Raspbian Buster / Debian Buster, armhf, glibc2.28)
- **Scaleway arm64(aarch64) Debian Buster, 32 core / RAM 32GB**
- microSD card 32GB
- Tensorflow v2.0.0
- **Bazel 0.26.1**

Procedure:

Install required packages:

```
$ sudo apt-get install -y \
make cmake wget curl libhdf5-dev libc-ares-dev libeigen3-dev
libatomic1 \
openmpi-bin libopenmpi-dev libatlas-base-dev zip unzip
```

Clone Tensorflow

```
$ cd ~

$ git clone -b v2.0.0 https://github.com/tensorflow/tensorflow.git

or

$ git clone -b v1.15.0 https://github.com/tensorflow/tensorflow.git

$ cd tensorflow/tensorflow/tools/lib_package


### Install openjdk-8-jdk armhf
$ curl -sc /tmp/cookie
"https://drive.google.com/uc?export=download&id=1LQUSal55R6fmawZS9zZuk
6-5ZFOdUqRK" > /dev/null
$ CODE="$(awk '/_warning_/ {print $NF}' /tmp/cookie)"
$ curl -Lb /tmp/cookie
"https://drive.google.com/uc?export=download&confirm=${CODE}&id=1LQUSa
l55R6fmawZS9zZuk6-5ZFOdUqRK" -o
adoptopenjdk-8-hotspot_8u222-b10-2_armhf.deb
$ sudo apt-get install -y
./adoptopenjdk-8-hotspot_8u222-b10-2_armhf.deb

### Install Bazel 0.26.1 armhf
$ wget
https://github.com/PINTO0309/Bazel_bin/raw/master/0.26.1/Raspbian_Debi
an_Buster_armhf/openjdk-8-jdk/install.sh
$ ./install.sh

### Build Tensorflow
$ sudo bazel --host_jvm_args=-Xmx512m build \
--config=noaws \
--config=nohdfs \
--config=noignite \
--config=nokafka \
--config=nonccl \
--config=v2 \
--local_resources=4096.0,3.0,1.0 \  #<===[If Pi3]
--local_resources=1024.0,0.5,0.5 \
--copt=-mfpu=neon-vfpv4 \
--copt=-ftree-vectorize \
--copt=-funsafe-math-optimizations \
```

```
--copt=-ftree-loop-vectorize \
--copt=-fomit-frame-pointer \
--copt=-DRASPBERRY_PI \
--host_copt=-DRASPBERRY_PI \
--linkopt=-Wl,-latomic \
--host_linkopt=-Wl,-latomic \
//tensorflow/tools/lib_package:libtensorflow


### Install openjdk-8-jdk armhf
$ curl -sc /tmp/cookie
"https://drive.google.com/uc?export=download&id=1LQUSal55R6fmawZS9zZuk
6-5ZFOdUqRK" > /dev/null
$ CODE="$(awk '/_warning_/ {print $NF}' /tmp/cookie)"
$ curl -Lb /tmp/cookie
"https://drive.google.com/uc?export=download&confirm=${CODE}&id=1LQUSa
l55R6fmawZS9zZuk6-5ZFOdUqRK" -o
adoptopenjdk-8-hotspot_8u222-b10-2_armhf.deb
$ sudo apt-get install -y
./adoptopenjdk-8-hotspot_8u222-b10-2_armhf.deb

### Install Bazel 0.26.1 armhf
$ wget
https://github.com/PINTO0309/Bazel_bin/raw/master/0.26.1/Raspbian_Debi
an_Buster_armhf/openjdk-8-jdk/install.sh
$ ./install.sh

### Build Tensorflow
$ sudo bazel --host_jvm_args=-Xmx512m build \
--config=noaws \
--config=nogcp \
--config=nohdfs \
--config=noignite \
--config=nokafka \
--config=nonccl \
--local_resources=4096.0,3.0,1.0 \ #<===[If Pi3]
--local_resources=1024.0,0.5,0.5 \
--copt=-mfpu=neon-vfpv4 \
--copt=-ftree-vectorize \
--copt=-funsafe-math-optimizations \
--copt=-ftree-loop-vectorize \
--copt=-fomit-frame-pointer \
--copt=-DRASPBERRY_PI \
--host_copt=-DRASPBERRY_PI \
--linkopt=-Wl,-latomic \
```

```
--host_linkopt=-Wl,-latomic \
//tensorflow/tools/lib_package:libtensorflow
```

```
FOR AARCH64:
```

```
### Install openjdk-8-jdk aarch64
$ curl -sc /tmp/cookie
"https://drive.google.com/uc?export=download&id=1VwLxzT3EOTbhSzwvRF2H4
ChTQyTQBt3x" > /dev/null
$ CODE="$(awk '/_warning_/ {print $NF}' /tmp/cookie)"
$ curl -Lb /tmp/cookie
"https://drive.google.com/uc?export=download&confirm=${CODE}&id=1VwLxz
T3EOTbhSzwvRF2H4ChTQyTQBt3x" -o
adoptopenjdk-8-hotspot_8u222-b10-2_arm64.deb
$ sudo apt-get install -y
./adoptopenjdk-8-hotspot_8u222-b10-2_arm64.deb

### Install Bazel 0.26.1 aarch64
$ wget
https://github.com/PINTO0309/Bazel_bin/raw/master/0.26.1/Raspbian_Debi
an_Buster_aarch64/openjdk-8-jdk/install.sh
$ ./install.sh

### Build Tensorflow
$ sudo bazel --host_jvm_args=-Xmx512m build \
--config=noaws \
--config=nohdfs \
--config=noignite \
--config=nokafka \
--config=nonccl \
--config=v2 \
//tensorflow/tools/lib_package:libtensorflow
```

【Pattern_4】Build_Tensorflow_v1.15.0_Debian_Buster_aarch64

```
### Install openjdk-8-jdk aarch64
$ curl -sc /tmp/cookie
"https://drive.google.com/uc?export=download&id=1VwLxzT3EOTbhSzwvRF2H4
ChTQyTQBt3x" > /dev/null
$ CODE="$(awk '/_warning_/ {print $NF}' /tmp/cookie)"
$ curl -Lb /tmp/cookie
"https://drive.google.com/uc?export=download&confirm=${CODE}&id=1VwLxz
T3EOTbhSzwvRF2H4ChTQyTQBt3x" -o
adoptopenjdk-8-hotspot_8u222-b10-2_arm64.deb
```

```
$ sudo apt-get install -y
./adoptopenjdk-8-hotspot_8u222-b10-2_arm64.deb

### Install Bazel 0.26.1 aarch64
$ wget
https://github.com/PINTO0309/Bazel_bin/raw/master/0.26.1/Raspbian_Debi
an_Buster_aarch64/openjdk-8-jdk/install.sh
$ ./install.sh

### Build Tensorflow
$ sudo bazel --host_jvm_args=-Xmx512m build \
--config=noaws \
--config=nogcp \
--config=nohdfs \
--config=noignite \
--config=nokafka \
--config=nonccl \
//tensorflow/tools/lib_package:libtensorflow
```

Prebuilt binaries are generated in

`bazel-bin/tensorflow/tools/lib_package/libtensorflow.tar.gz.`

INSTALL TENSORFLOW C API

```
$ tar -C /usr/local -xzf libtensorflow.tar.gz
```

This github repo seems amazing:

https://github.com/PINTO0309/Tensorflow-bin/#usage: