Test Compilation

Hao Deng

December 2023

1 Introduction

After compileing Libtorch with OpenBLAS/MKL, we want to test whether the compilation is successful. We will go through a simple way to test if the compilation is successful.

2 Validate Libtorch with MKL

2.1 Create CMakeLists.txt

Create a directory called ”test”: `mkdir test`.
Create a CMakeLists.txt file with the following content in the directory ”test”.
Replace ”Path_To_Libtorch_mkl” with your path to libtorch_mkl.

```
cmake_minimum_required(VERSION 3.0)
project("libtorch_mkl")

set(Torch_ROOT Path_To_Libtorch_mkl) # Need to replace libtorch path
list(APPEND CMAKE_PREFIX_PATH Path_To_Libtorch_mkl) # Need to replace libtroch path
find_package(Torch REQUIRED NO_MODULE ) #add torch

SET(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Ofast -march=native") #compile parameters

add_executable(tests test.cpp)


target_link_libraries(tests ${TORCH_LIBRARIES}) #link torch to tests
```

2.2 Create test.cpp

Create a test.cpp file with the following content in the directory ”test”:

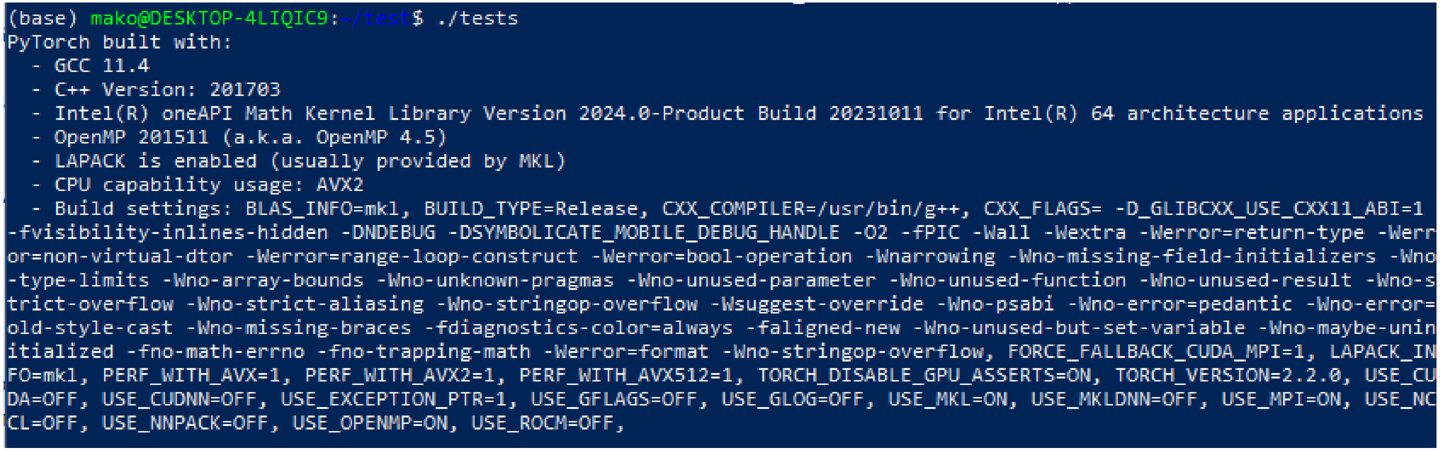
```
#include <torch/torch.h>
#include <iostream>

int main() {
    std::cout<<torch::show_config()<<std::endl;
    return 0;
}
```

2.3 Validation

Run the following command to validate:

```
rm -rf build
mkdir build
cd build
cmake ..
make
./tests
```

The result should look like this:



```
(base) mako@DESKTOP-4LIQIC9:~/test$ ./tests
PyTorch built with:
  - GCC 11.4
  - C++ Version: 201703
  - Intel(R) oneAPI Math Kernel Library Version 2024.0-Product Build 20231011 for Intel(R) 64 architecture applications
  - OpenMP 201511 (a.k.a. OpenMP 4.5)
  - LAPACK is enabled (usually provided by MKL)
  - CPU capability usage: AVX2
  - Build settings: BLAS_INFO=mkl, BUILD_TYPE=Release, CXX_COMPILER=/usr/bin/g++, CXX_FLAGS= -D_GLIBCXX_USE_CXX11_ABI=1
-fvisibility-inlines-hidden -DNDEBUG -DSYMBOLICATE_MOBILE_DEBUG_HANDLE -O2 -fPIC -Wall -Wextra -Werror=return-type -Werr
or=non-virtual-dtor -Werror=range-loop-construct -Werror=bool-operation -Wnarrowing -Wno-missing-field-initializers -Wno
-type-limits -Wno-array-bounds -Wno-unknown-pragmas -Wno-unused-parameter -Wno-unused-function -Wno-unused-result -Wno-s
trict-overflow -Wno-strict-aliasing -Wno-stringop-overflow -Wsuggest-override -Wno-psabi -Wno-error=pedantic -Wno-error=
old-style-cast -Wno-missing-braces -fdiagnostics-color=always -faligned-new -Wno-unused-but-set-variable -Wno-maybe-unin
itialized -fno-math-errno -fno-trapping-math -Werror=format -Wno-stringop-overflow, FORCE_FALLBACK_CUDA_MPI=1, LAPACK_IN
FO=mkl, PERF_WITH_AVX=1, PERF_WITH_AVX2=1, PERF_WITH_AVX512=1, TORCH_DISABLE_GPU_ASSERTS=ON, TORCH_VERSION=2.2.0, USE_CU
DA=OFF, USE_CUDNN=OFF, USE_EXCEPTION_PTR=1, USE_GFLAGS=OFF, USE_GLOG=OFF, USE_MKL=ON, USE_MKLDNN=OFF, USE_MPI=ON, USE_NC
CL=OFF, USE_NNPACK=OFF, USE_OPENMP=ON, USE_ROCM=OFF,
```

The desirable result should shows that BLAS_INFO=mkl.

3 Validate Libtorch with OpenBLAS

3.1 Create CMakeLists.txt

Create a directory called ”test”: `mkdir test`.
Create a CMakeLists.txt file with the following content in the directory ”test”.
Replace ”Path_To_Libtorch_OpenBLAS” with your path to libtorch_OpenBLAS.

```
cmake_minimum_required(VERSION 3.0)
project("libtorch_OpenBLAS")

set(Torch_ROOT Path_To_Libtorch_OpenBLAS) # Need to replace libtorch path
list(APPEND CMAKE_PREFIX_PATH Path_To_Libtorch_OpenBLAS) # Need to replace libtorch path
find_package(Torch REQUIRED NO_MODULE ) #add torch

SET(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Ofast -march=native") #compile parameters

add_executable(tests test.cpp)


target_link_libraries(tests ${TORCH_LIBRARIES}) #link torch to tests
```

3.2 Create test.cpp

Create a test.cpp file with the following content in the directory ”test”:

```
#include <torch/torch.h>
#include <iostream>

int main() {
    std::cout<<torch::show_config()<<std::endl;
    return 0;
}
```

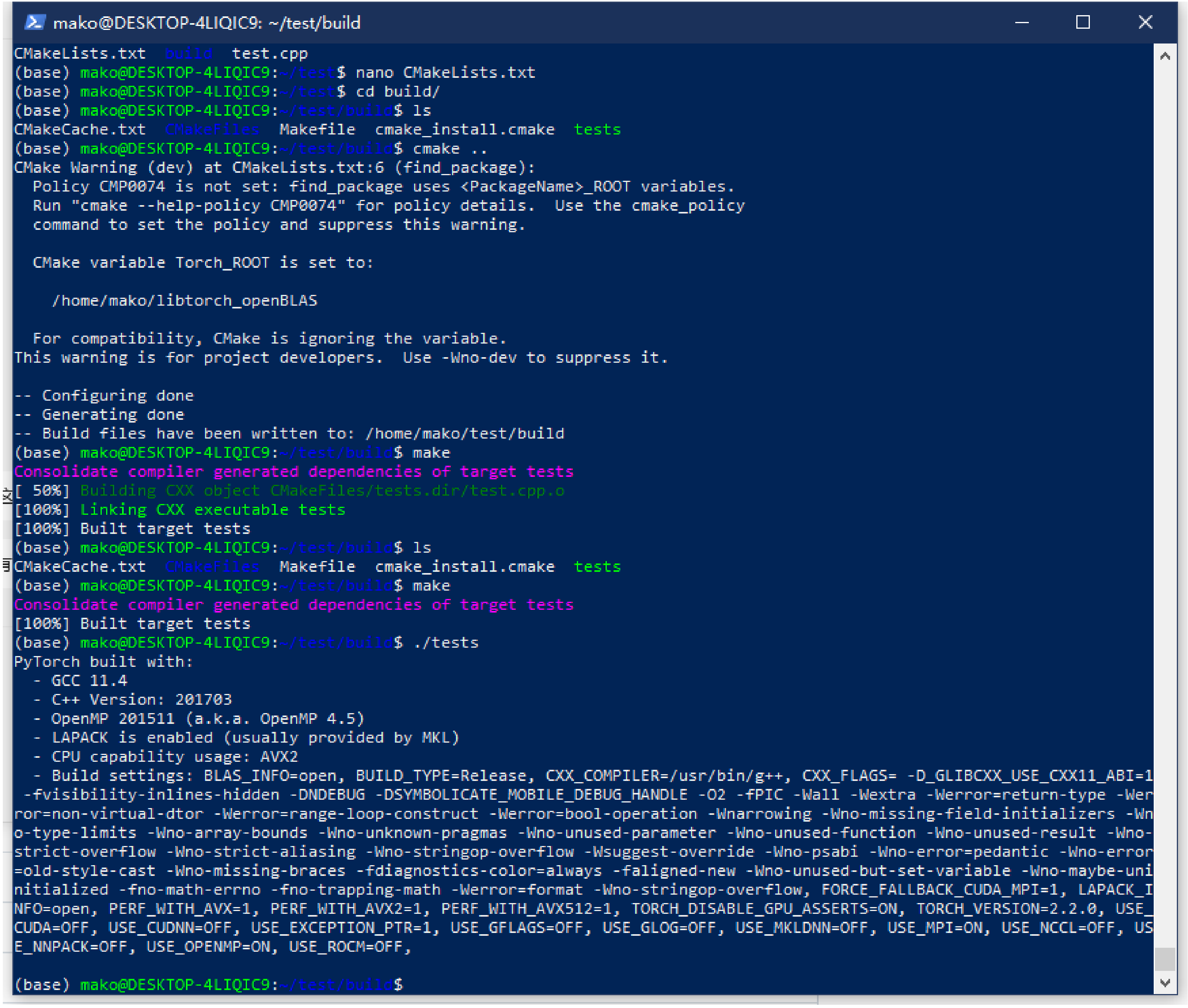
.

3.3 Validation

Run the following command to validate:

```
rm -rf build
mkdir build
cd build
cmake ..
make
./tests
```

The result should look like this:



```
CMakeLists.txt  build  test.cpp
(base) mako@DESKTOP-4LIQIC9:~/test$ nano CMakeLists.txt
(base) mako@DESKTOP-4LIQIC9:~/test$ cd build/
(base) mako@DESKTOP-4LIQIC9:~/test/build$ ls
CMakeCache.txt  CMakeFiles  Makefile  cmake_install.cmake  tests
(base) mako@DESKTOP-4LIQIC9:~/test/build$ cmake ..
CMake Warning (dev) at CMakeLists.txt:6 (find_package):
  Policy CMP0074 is not set: find_package uses <PackageName>_ROOT variables.
  Run "cmake --help-policy CMP0074" for policy details.  Use the cmake_policy
  command to set the policy and suppress this warning.

  CMake variable Torch_ROOT is set to:

    /home/mako/libtorch_openBLAS

  For compatibility, CMake is ignoring the variable.
This warning is for project developers.  Use -Wno-dev to suppress it.

-- Configuring done
-- Generating done
-- Build files have been written to: /home/mako/test/build
(base) mako@DESKTOP-4LIQIC9:~/test/build$ make
Consolidate compiler generated dependencies of target tests
[ 50%] Building CXX object CMakeFiles/tests.dir/test.cpp.o
[100%] Linking CXX executable tests
[100%] Built target tests
(base) mako@DESKTOP-4LIQIC9:~/test/build$ ls
CMakeCache.txt  CMakeFiles  Makefile  cmake_install.cmake  tests
(base) mako@DESKTOP-4LIQIC9:~/test/build$ make
Consolidate compiler generated dependencies of target tests
[100%] Built target tests
(base) mako@DESKTOP-4LIQIC9:~/test/build$ ./tests
PyTorch built with:
  - GCC 11.4
  - C++ Version: 201703
  - OpenMP 201511 (a.k.a. OpenMP 4.5)
  - LAPACK is enabled (usually provided by MKL)
  - CPU capability usage: AVX2
  - Build settings: BLAS_INFO=open, BUILD_TYPE=Release, CXX_COMPILER=/usr/bin/g++, CXX_FLAGS= -D_GLIBCXX_USE_CXX11_ABI=1
 -fvisibility-inlines-hidden -DNDEBUG -DSYMBOLICATE_MOBILE_DEBUG_HANDLE -O2 -fPIC -Wall -Wextra -Werror=return-type -Wer
ror=non-virtual-dtor -Werror=range-loop-construct -Werror=bool-operation -Wnarrowing -Wno-missing-field-initializers -Wn
o-type-limits -Wno-array-bounds -Wno-unknown-pragmas -Wno-unused-parameter -Wno-unused-function -Wno-unused-result -Wno-
strict-overflow -Wno-strict-aliasing -Wno-stringop-overflow -Wsuggest-override -Wno-psabi -Wno-error=pedantic -Wno-error
=old-style-cast -Wno-missing-braces -fdiagnostics-color=always -faligned-new -Wno-unused-but-set-variable -Wno-maybe-uni
nitialized -fno-math-errno -fno-trapping-math -Werror=format -Wno-stringop-overflow, FORCE_FALLBACK_CUDA_MPI=1, LAPACK_I
NFO=open, PERF_WITH_AVX=1, PERF_WITH_AVX2=1, PERF_WITH_AVX512=1, TORCH_DISABLE_GPU_ASSERTS=ON, TORCH_VERSION=2.2.0, USE_
CUDA=OFF, USE_CUDNN=OFF, USE_EXCEPTION_PTR=1, USE_GFLAGS=OFF, USE_GLOG=OFF, USE_MKLDNN=OFF, USE_MPI=ON, USE_NCCL=OFF, US
E_NNPACK=OFF, USE_OPENMP=ON, USE_ROCM=OFF,

(base) mako@DESKTOP-4LIQIC9:~/test/build$
```

The desirable result should shows that BLAS_INFO=open.