Build TVM and TVM Android RPC

Created by	xiaolong
Created time	@May 28, 2025 11:01 AM
Tags	

```
rm -rf build && mkdir build && cd build
cp ../cmake/config.cmake .
```

Config build in config.cmake

```
# controls default compilation flags
echo "set(CMAKE_BUILD_TYPE RelWithDebInfo)" >> config.cmake
# LLVM is a must dependency
echo "set(USE_LLVM \\"llvm-config --ignore-libllvm --link-static\\")" >> config.cmake
echo "set(HIDE_PRIVATE_SYMBOLS ON)" >> config.cmake
# GPU SDKs, turn on if needed
echo "set(USE_CUDA   OFF)" >> config.cmake
echo "set(USE_METAL  ON)" >> config.cmake
echo "set(USE_VULKAN OFF)" >> config.cmake
echo "set(USE_OPENCL ON)" >> config.cmake
# FlashInfer related, requires CUDA w/ compute capability 80;86;89;90
echo "set(USE_FLASHINFER OFF)" >> config.cmake
echo "set(FLASHINFER_CUDA_ARCHITECTURES YOUR_CUDA_COMPUTE_CAPABILITY_HERE)" >> config.cmake
echo "set(CMAKE_CUDA_ARCHITECTURES YOUR_CUDA_COMPUTE_CAPABILITY_HERE)" >> config.cmake
```

Build libtvm using cmake (A success build should produce libtvm and libtvm_runtime under /path-tvm-unity/build/ directory.)

cmake .. && cmake --build . --parallel 4

Install the build into the environment

cd /path-to-tvm-unity/python
pip install -e .

Validate TVM installation

python -c "import tvm; print(tvm.**file**)" #locate TVM python package
python -c "import tvm; print(tvm._ffi.base._LIB)" # Confirm which TVM lib is used
python -c "import tvm; print('\n'.join(f'{k}: {v}' for k, v in tvm.support.libinfo().items()))" # reflect tvm build option

TVM Android RPC – Full Build & Installation Guide

1. Prerequisites

- Host PC: Ubuntu
- Packages: `openjdk-11-jdk`, `maven`, `cmake`, `gcc/g++`
- Android SDK: `$ANDROID_HOME`
- Build-Tools: 30.0.3 (zipalign/apksigner)
- Android NDK: 29.0.13113456

- Device: Android 11+ (arm64-v8a)

2. Build TVM4J (Java API & JNI)

```bash

# TVM root

cd
~/Downloads/tvm-0.19.0

# Compile Java core + JNI libs

make
jvmpkg -j$(nproc)

# (Optional) install tvm4j-core into
local Maven

make jvminstall

output：

*

jvm/core/target/tvm4j-core-0.0.1-SNAPSHOT.jar

*

jvm/native/linux-x86_64/.../libtvm4j_runtime_packed.so(x86_64)
```

3. Prepare Android RPC Project

```bash

cd
apps/android_rpc

**Gradle wrapper **

gradle/wrapper/gradle-wrapper.properties`

distributionUrl=https://services.gradle.org/distributions/gradle-8.9-all.zip

**build.gradle
 modification**
```

gradle

sourceSets {

main {

jni.srcDirs = []

jniLibs.srcDirs =
['src/main/jniLibs']   //

}

}

packagingOptions
{

jniLibs.useLegacyPackaging = true

exclude
'META-INF/*'

}

aaptOptions {

noCompress 'arsc'
}
copy .so
mkdir
-p app/src/main/jniLibs/arm64-v8a
cp
<ARM64>/libtvm4j_runtime_packed.so
app/src/main/jniLibs/arm64-v8a/
cp
$ANDROID_NDK_HOME/toolchains/llvm/prebuilt/linux-*/sysroot/usr/lib/aarch64-linux-android/libc++_shared.so
app/src/main/jniLibs/arm64-v8a/

4. Build Unsigned APK

```bash
```

```
./gradlew
clean

./gradlew assembleRelease   #generate
app/build/outputs/apk/release/app-release-unsigned.apk
```

5. Align & Sign APK

```bash

# 4-byte
alignment

$ANDROID_HOME/build-tools/30.0.3/zipalign -f -p 4
app/build/outputs/apk/release/app-release-unsigned.apk
app/build/outputs/apk/release/app-release-unsigned-aligned.apk

#
Generate keystore

dev_tools/gen_keystore.sh   # Password &
alias = 123456

#
Sign

$ANDROID_HOME/build-tools/30.0.3/apksigner sign   --ks
dev_tools/tvmrpc.keystore   --ks-key-alias tvmrpc   --ks-pass
pass:123456   --key-pass pass:123456   --out
app/build/outputs/apk/release/tvmrpc-release.apk
app/build/outputs/apk/release/app-release-unsigned-aligned.apk
```

6. Install & Verify

```bash

adb uninstall
org.apache.tvm.tvmrpc

adb install -r
app/build/outputs/apk/release/tvmrpc-release.apk
```