Introduction

- Tornado is not a trivial application to build
- Mixed source as it accesses low-level OpenCL drivers via JNI
- It uses the maven build system
- It is a multi module project.
- Compiling generates a complete (and distributable) SDK.

Changes

- Rationalised build system - end-to-end build automated by maven
  - In preparation for continuous integration (jenkins)
- Fixed issues in maven configuration —hard coded paths
- Now implemented a working versioning and release system
- tornado is now utilises the Java extension mechanism
- Tried to package tornado to eliminate passing dependencies onto user applications
  - Partly succeeded but may need some dependencies for code completion within IDEs
- Added extra flags into tornado wrapper scripts.

Issues:

- May need to execute clean before re-building

Pre-requisites:

- maven v3
- cmake 3.6 (or newer)
- OpenCL (preferably v1.2 or newer)
- gcc or clang/llvm
- python 2.7 (?)
- Tornado ready OpenJDK 1.8.0_131

Tested OS:

- OSX 10.13.2 (High Sierra)
- CentOS 6.8
- Fedora 21

Installing the Tornado Ready OpenJDK

- Distributed as a binary tar ball (e.g. jdk1.8.0_131_x86.tgz)
- Available in Dropbox (?)

- To see what files are inside the release:

```
[jclarkson@axleman opt]$ tar -tvf jdk1.8.0_131_x86.tgz
...
-rw-r--r-- jclarkson/jclarkson    109234 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbyLocale_zh_CN.jar
-rw-r--r-- jclarkson/jclarkson    123568 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbyLocale_ja_JP.jar
-rw-r--r-- jclarkson/jclarkson    112162 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbyLocale_de_DE.jar
-rw-r--r-- jclarkson/jclarkson    112091 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbyLocale_fr.jar
-rw-r--r-- jclarkson/jclarkson    229601 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbytools.jar
-rw-r--r-- jclarkson/jclarkson    266471 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbynet.jar
-rw-r--r-- jclarkson/jclarkson     49800 2017-03-15 08:32 jdk1.8.0_131/
db/lib/derbyoptionaltools.jar
-rw-r--r-- jclarkson/jclarkson       908 2017-03-15 08:32 jdk1.8.0_131/
db/README-JDK.html
```

```
-rw-r--r-- jclarkson/jclarkson     11560 2017-03-15 08:32 jdk1.8.0_131/
db/LICENSE
-r--r--r-- jclarkson/jclarkson        40 2017-03-15 08:32 jdk1.8.0_131/
LICENSE
[jclarkson@axleman opt]$
```

- As these files are unlikely to change I normally install this under ${HOME}/opt/jdk_1.8.0_131 as so
  -

```
[jclarkson@axleman opt]$ cd ~
[jclarkson@axleman opt]$ mkdir opt && cd opt
[jclarkson@axleman opt]$ tar -cvf jdk1.8.0_131_x86.tgz
...
jdk1.8.0_131/db/lib/derbyclient.jar
jdk1.8.0_131/db/lib/derbyLocale_zh_CN.jar
jdk1.8.0_131/db/lib/derbyLocale_ja_JP.jar
jdk1.8.0_131/db/lib/derbyLocale_de_DE.jar
jdk1.8.0_131/db/lib/derbyLocale_fr.jar
jdk1.8.0_131/db/lib/derbytools.jar
jdk1.8.0_131/db/lib/derbynet.jar
jdk1.8.0_131/db/lib/derbyoptionaltools.jar
jdk1.8.0_131/db/README-JDK.html
jdk1.8.0_131/db/LICENSE
jdk1.8.0_131/LICENSE
[jclarkson@axleman opt]$
[jclarkson@axleman opt]$ ls
jdk1.8.0_131
[jclarkson@axleman opt]$
```

- Once installed this JDK can be used by setting the JAVA_HOME environment variable

```
[jclarkson@axleman opt]$ export JAVA_HOME=${HOME}/opt/jdk1.8.0_131
```

- The problem with this is that this variable will be lost every time a new shell is started –– this happens during logon, opening a new terminal or even running a script.
- To avoid this I normally set the variable in my .bashrc file

```
[jclarkson@axleman ~]$ vi ~/.bashrc

export JAVA_HOME=${HOME}/opt/jdk1.8.0_131
```

2. Setting up a system specific build environment.

- As Tornado relies on using a custom Java JDK we need to tell maven about it before we are able to compile any source code.
- In Maven user (or system) specific build environments are configured in the ${HOME}/.m2/settings.xml

```
[jclarkson@axleman ~]$ vi ~/.m2/settings.xml
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
                              https://maven.apache.org/xsd/
settings-1.0.0.xsd">
```

```
    <interactiveMode/>
    <usePluginRegistry/>
    <offline/>
    <pluginGroups/>
    <servers/>
    <mirrors/>
    <proxies/>
    <profiles>
        <profile>
            <id>tornado-jvmci</id>
            <activation>
                <activeByDefault>true</activeByDefault>
            </activation>
            <properties>
                <jvmci.root>/home/jclarkson/opt/jdk1.8.0_131/</
jvmci.root>
                <jvmci.version>1.8.0_131</jvmci.version>
            </properties>
        </profile>
    </profiles>
    <activeProfiles/>
</settings>

- By default the tornado-ivmci profile will be visible to maven, just
  ensure that it points to the correct JVMCI enabled JDK.


3. Setup ssh access on github

- By default maven expects that github is accessed via ssh
- Make sure that you have uploaded your ssh public key

4. Setup cmake

- Try running

$ cmake -version

- If the version of cmake is > 3.6 then skip the rest of this step
- Otherwise try in install cmake

- For simplicity it might be easier to install cmake in your home
  directory.
-  Redhat Enterprise Linux / CentOS use cmake v2.8
-  We need a newer version so that OpenCL is configured properly

[jclarkson@axleman tornado]$ cd ~/Downloads
[jclarkson@axleman tornado]$ wget https://cmake.org/files/v3.10/
cmake-3.10.1-Linux-x86_64.tar.gz
[jclarkson@axleman tornado]$ cd ~/opt/
[jclarkson@axleman tornado]$ tar -tvf ~/Downloads/cmake-3.10.1-Linux-
x86_64.tar.gz
[jclarkson@axleman tornado]$ mv cmake-3.10.1-Linux-x86_64 cmake-3.10.1
[jclarkson@axleman tornado]$ export PATH=/home/jclarkson/opt/
cmake-3.10.1/bin/:$PATH
[jclarkson@axleman tornado]$ cmake -version
```

```
cmake version 3.10.1

CMake suite maintained and supported by Kitware (kitware.com/cmake).
[jclarkson@axleman tornado]$
```

4. Clone Tornado repository and build SDK

```
[jclarkson@axleman tornado]$ cd ${HOME}/git
[jclarkson@axleman tornado]$ git clone git@github.com:beehive-lab/
tornado.git
[jclarkson@axleman tornado]$ cd tornado

[jclarkson@axleman tornado]$ mvn package

Or if a custom installation of cmake is required then
[jclarkson@axleman tornado]$ mvn  -Dcmake.root.dir=/home/jclarkson/opt/
cmake-3.10.1/ package

...
[INFO]
------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] tornado ........................................... SUCCESS
[  9.991 s]
[INFO] tornado-runtime ................................... SUCCESS
[  0.738 s]
[INFO] tornado-collections ............................... SUCCESS
[  0.074 s]
[INFO] tornado-drivers ................................... SUCCESS
[  0.002 s]
[INFO] tornado-drivers-opencl ............................ SUCCESS
[  4.046 s]
[INFO] tornado-drivers-opencl-jni ........................ SUCCESS
[ 43.778 s]
[INFO] tornado-examples .................................. SUCCESS
[  0.643 s]
[INFO] tornado-benchmarks ................................ SUCCESS
[  2.013 s]
[INFO] tornado-unittests ................................. SUCCESS
[  0.461 s]
[INFO] tornado-assembly .................................. SUCCESS
[ 23.895 s]
[INFO]
------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO]
------------------------------------------------------------------------
[INFO] Total time: 01:25 min
[INFO] Finished at: 2018-01-01T18:44:53+00:00
[INFO] Final Memory: 45M/1075M
[INFO]
------------------------------------------------------------------------
```

- On success a complete Tornado SDK is build under ./dist

```
[jclarkson@axleman tornado]$ ls dist/
tornado-sdk  tornado-sdk.tar.gz
```

- The tarball is created to allows the SDK to be distributed easier to other machines or 3rd parties.

- Notice that the root directory of the SDK is versioned with a specific git commit

```
[jclarkson@axleman tornado]$ ls dist/tornado-sdk
tornado-sdk-0.0.2-SNAPSHOT-16a30a3
```

- The SDK is laid out according to the linux The Filesystem Hierarchy Standard, Version 2.2 <http://www.path-name.com/fhs/>.
  - see heir(7) man page

- This layout can then be used to generate OS specific packages -- e.g. rpm for RHEL/CENTOS/FEDORS, deb for Debian/Ubuntu.

- Files/libraries are designed to be relocatable and co-exist with other versions of the same package.
  - Think tornado version 1.0.0 and 2.0.0
- As a consequence there are a strict(-ish) set of guidelines surrounding how they are included in this filesystem.
  - The ones for Fedora can be found here and most distributions follow very similar guidelines.
    - https://fedoraproject.org/wiki/Packaging:Guidelines
    - https://fedoraproject.org/wiki/Packaging:Java
  - Please double check these guidelines before adding/changing the filesystem.

- By keeping to these guidelines now, we should be able to quickly integrate Tornado into key OS distributions very quickly once we open-source the project. :)

```
[jclarkson@axleman tornado]$ ls dist/tornado-sdk/tornado-sdk-0.0.2-
SNAPSHOT-16a30a3/
bin  etc  lib  share
```

5. Using the Tornado SDK

Two environment variables are required to use tornado: JAVA_HOME and TORNADO_SDK.

JAVA_HOME should be set to the custom JVMCI enabled JDK that was used to build tornado.

TORNADO_SDK should be set to the root directory of the tornado SDK. In our example this is ${HOME}/git/tornado/dist/tornado-sdk/tornado-sdk-0.0.2-SNAPSHOT-16a30a3/

Finally, I normally add the ${TORNADO_SDK}/bin to the PATH environment variable.

```
export PATH=${PATH}:${TORNADO_SDK}/bin
```

```
[jclarkson@axleman tornado-sdk-0.0.2-SNAPSHOT-16a30a3]$ tornado
tornado.examples.HelloWorld
tornado[  0,  0,  0]> hello
tornado[  3,  0,  0]> hello
tornado[  1,  0,  0]> hello
tornado[  2,  0,  0]> hello
tornado[  6,  0,  0]> hello
tornado[  7,  0,  0]> hello
tornado[  4,  0,  0]> hello
tornado[  5,  0,  0]> hello

[jclarkson@axleman tornado-sdk-0.0.2-SNAPSHOT-16a30a3]$ tornado-test.py
-V
```

Common Build Issues

Wrong version of Cmake

- Some distributions do not come with an up-to-date version of cmake
  - e.g. CentOS /usr/bin/cmake -> version 2.8
  - or /usr/bin/cmake3 -> version 3.6
  - In this situation we cannot use the OS provided version due to a limitation in the maven cmake plugin.
  - I will try and submit a patch to fix this - but it might take some time

Cannot find OpenCL

- Some OpenCL SDKs – like NVIDA CUDA, AMD, Intel and ARM – are not installed into a specific location.
- Instead they are located via an environment variable.
  - Try setting the appropriate variable for the SDK you are trying to build tornado against.
    - Note: this will not limit tornado into using a specific SDK.
    - Some environment variables that could help are:
      - AMDAPPSDKROOT
      - INTELOCLSDKROOT
      - CUDA_PATH
      - NVSDKCOMPUTE_ROOT
      - ATISTREAMSDKROOT