A Complex Project build by CMake

J.Y.LI[1]*

Abstract
In this article we have been use CMake build a complex project. Through this method we can packing many libraries and invoke in main module. In this time we have been use CMake as the project build tool, using **ROOT** as a graphic library, use **FFTW** as Fast Fourier Transform algorithm demo. Even we have been built a library implementation by ourselves. Through many build method we have successfully build the complex project.

在本文中我们通过使用CMake的方法实现了复杂工程的构建，通过这种方法我们可以通过使用不同的技术封装成库通过主模块进行调度，其中我们通过使用CMake来做工程构建工具。期间我们的主模块通过调度以root作为图形库，以FFTW作为快速傅里叶变换库，以自建的库作为工程的数学算法库，通过不同的手段构建项目中的库。

Keywords
Cmake — Root — FFT

[1] *school of physics , Huazhong University of Science & Technology, WuHan ,China*
[1] *Software engineering , Dalian Neusoft University of Information, DaLian ,China*
***Corresponding author**:lijianying12@gmail.com
***Document last build date : May 1, 2013. Document version : 0.1**

Contents

Introduction

In recent work we have got a special requirement, It's a computing project, we have to use FFT algorithm and many special algorithm to write programs. Even need plot diagram in real-time output, it is too complex to implementation in the situation without a framework. So we need to use CMake as the project build tool cause of its convenience and have a good libraries support. The target algorithm needs many technology supports we may include it in project subdirectory and give library compiler parameters. Use these packaged libraries need use CMake give a full solution to link them.

The program ROOT is a data analysis framework, with the core of the ROOT: CINT. Using it can be invoking in a C++ library and it also have the same function of ROOT itself. But why do we use root as the real-time diagram output library? Because it has a faster speed than any other diagram what I know. The syntax of root script is same as the C++ syntax, so it is simple to use as a C/C++ programmer. Nevertheless the best feature is that it has a rich function for data analysis.

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions. It support MPI for parallel computing. The reason why we using it cause of it's the most developed library for FFT algorithm. To read the sample project open the file attach.

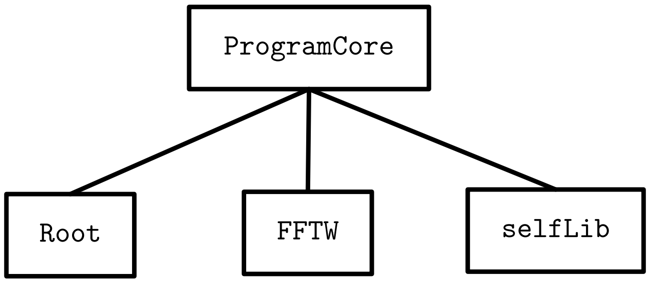

Figure 1. Program outline

1. Simple use of CMake

The basic using of CMake is very simple. The most things you need concentrate on is how to build you project. Use CMake you may know Out of source build. In this way you can make all your .o files, various temporary depend on files and even the binary executable without cluttering up your source tree. If you want to do like this. You may make a new directory out of the source code tree, and then go into the build dir run the command **cmake [Source directory]**. The source directory is you source top dir it may contant the CMakeLists.txt.Also you can generate the project for Eclipse CDT4. Use CMake parameter **-G "Eclipse CDT4 - Unix Makefiles"**. run the command then you can import the build directory as an Eclipse project.

2. Root Library deploy

In this section we have a target that build a root integrate library for Core module. The Origin library address: here In this web page introduction the using XCode write ROOT code. Then we can take an analysis on this code understand these code and transform it to our library.

Now here are our analysis and the code we have been write for the core module. We explained all of the statement of CMake code one by one on the attachment file. If you have been red the file **code/src/event/CMakeLists.txt** then you will be know the Root library is how to deploy. If you want for a test just compile the library and write a simple C++ program invoke the library.

But there are even something maybe you must notice:
1. To use this library you may write plot script in root engine. If you have the script you can use the code as a C++ statement in the library.
2. To add a plot function what you aim at. You need create a cpp file and correspondence h file in the library in directly. e.g. in the subdir event have file graphics.cpp and graphics.h in this file have function testGraphics it's the target function to help you plot the diagram file.
3. Using Draw function to plot the graph. Use the SaveAs function save the graph as a file. Its support EPS JPG PNG even GIF.
4. If you need a script support you may find many sample plot file in root install directory the path is : **[root install dir]/share/doc/root/tutorials** to run this samples you may run the command in the directory **root demos.C**

Other information you may need:

1. ROOT official website: ClickHere
2. For install root we recommend use source file.
3. For more script support download ROOT manual
4. Or search in GOOGLE recomend key words Cern root [target object] [Your problem]

3. FFTW library deploy

In this section build the FFTW library for Core library. the introduction of FFTW see here

Compile requirement : When configure the source code you may use the –enable-shared –with-pic. Using this statement for compile shared library support.

It is very simple to implementation the requirement. First you need find the fftw include directory manual. if you configured the path (–prefix) you may find in the prefix path in that parameter. In the file **code/src/FFT/CMakeLists.txt** have an example. FFTW configure sample. the /opt/fftw is this testing prefix dir.

But why not use FindFFTW.cmake? cause of this testing computer have many fftw installed position. I have to point out the position. The using of the library may the same as the ROOT do.

4. sefl library build

It is very simple to build a library that all of the code write by your self. Only create the souce file as usually. Create the project build file **CMakeLists.txt**. The file in the code folder position: **code/src/lib1/CMakeLists.txt**

Then in the requirement we need using this library for scientific computing. so we need change the compiler to icc or icpc. then you need the order in the CMakeLists.txt
SET(CMAKE_C_COMPILER icc)
SET(CMAKE_CXX_COMPILER icpc)

5. Main module project build

In previous sections include ROOT FFTW and self library build. the subproject could use and run out of the main module. It's mean that you can use the project for other project even out change any code!

to include the subproject have many step :

1. add the subdirectory for the main project using the order : add_subdirectory
2. add the link director for the main project using the order : link_directories
3. include directorys find the order in the file and add it to that list.
4. in the tail of the file add use the order target_link_libraries to link the main program to you library.

The project sample in the path : **code/src/CMakeLists.txt** you may find that it is very simple to use the CMake to build you project.

6. Debug module code blocks

Some times we need write some debug code blocks for output the memory information. bug when we run the program don't need such codes. Then we can add some code in CMakeLists.txt file
IF(CMAKE_BUILD_TYPE MATCHES Debug)
ADD_DEFINITIONS(-DDEBUG)
MESSAGE(DebugModeOpen)
ELSE(CMAKE_BUILD_TYPE MATCHES Debug)
MESSAGE(ReleaseModeOpen)
ENDIF(CMAKE_BUILD_TYPE MATCHES Debug)
And then you can write the source code file just like **#ifdef DEBUG** and write **#endif** at the end of the DEBUG code block.

conclusion

If you have a big requirement to use so many library and technology. It is a good way to build the project using CMake. Have Fun!

system information:CentOS 6.3 — CMake version : 2.8.10.2 — FFTW version : 3.3.3 — ROOT version : 5.34/04