CMake Trilinos?

Roscoe A. Bartlett
Department of Optimization & Uncertainty Estimation

Esteban J. Guillen
Department of Information Engineering

Sandia National Laboratories

Trilinos User Group Meeting, October 21, 2008
Outline

• What is CMake?

• User advantages in switching Trilinos to CMake

• Configuring, building, and installing Trilinos with CMake on Unix systems

• Native Microsoft Windows support
  – Self-extracting installer for Trilinos
  – Visual C++ project files and Windows CMake GUI
Overview of CMake

• CMake = “Cross-platform Make”

• CMake:
  – Build system primarily for C/C++ code
  – Front-ends to configure a software package
    • Command-line, Scripts, CURSES, GUIs
  – Back-ends that build code
    • Unix Makefiles, MS Visual C++ Projects, Eclipse Projects, ...
  – Packaging and installing
    • Tar/gzip, Windows self-extracting installers, PackageMaker, RPM, ...

• Platforms and usage:
  – Platforms:
    • Unix/Linux, MAC OSX, MS Windows, AIX, IRIX, ...
  – Internal Sandia use:
    • VTK/Titan, ParaView, ThreatView, ...
  – External use:
    • KDE, MySql, MiKTeX, (and many many more) ..

CMake is a full featured mature build system!
User advantages in switching Trilinos to CMake?

- Provide native support for MS Windows
  - Visual C++ projects
  - GUI binary installers

- Better user feedback for configuration errors

- Better support for shared libraries on many platforms

- More packaging and installation options

- Easier configuration for complex package dependencies
Current Status of Trilinos/CMake

• Our detailed evaluation of CMake for Trilinos is finished:

• We have a nearly complete CMake build system design in Trilinos Dev

• Current CMake enabled packages:
  – Teuchos, RTOp, Epetra, Triutils, EpetraExt, Thyra, RBGen

• Trilinos community close to making a decision to move to CMake?
Configuring Trilinos with CMake on Unix/Linux

- **CMake interactive mode:** [Not Recommended]

  ```
  $ cmake -i $TRILINOS_HOME
  ```

- **CCMake (CURSES):**

  ```
  $ ccmake $TRILINOS_HOME
  ```

- **CMake script files:**

  ```
  $ cmake -S script_file \ $TRILINOS_HOME
  ```

- **CMake command-line options:** [Recommended]

  ```
  $ cmake -D Trilinos_ENABLE_ALL_PACKAGES:BOOL=ON \ -D Trilinos_ENABLE_TESTS:BOOL=ON ... $TRILINOS_HOME
  ```
Creating a Configuration Script for CMake

```
#!/bin/sh
EXTRA_ARGS=$@
cmake \
  -D CMAKE_CXX_FLAGS:STRING="-g -O0 -ansi -pedantic -Wall" \
  -D DART_TESTING_TIMEOUT:STRING=600 \
  -D Trilinos_ENABLE_NOX:BOOL=ON \
  -D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON \
  -D Trilinos_ENABLE_EXAMPLES:BOOL=ON \
  -D Trilinos_ENABLE_TESTS:BOOL=ON \
  ... \
  $EXTRA_ARGS \
  ../../../Trilinos
$ ./do-configure -D VEROBSE_CONFIGURE:BOOL=ON
$ make -j4
$ ctest
$ make install

See example scripts:
  Trilinos/sampleScripts/*cmake
```
Special Configuration Modes for Trilinos

• Configuring Trilinos to build all packages with all tests and examples:

-D Trilinos_ENABLE_ALL_PACKAGES:BOOL=ON
-D Trilinos_ENABLE_TESTS:BOOL=ON
-D Trilinos_ENABLE_EXAMPLES:BOOL=ON

• Configuring a package(s) along with all of the packages it can use

-D Trilinos_ENABLE_Stratimikos:BOOL=ON
-D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON
-D Trilinos_ENABLE_TESTS:BOOL=ON
-D Trilinos_ENABLE_EXAMPLES:BOOL=ON

• Configuring Trilinos to disable a package(s) and all packages it depends on:

-D Trilinos_ENABLE_Stratimikos:BOOL=ON
-D Trilinos_ENABLE_Amesos:BOOL=OFF
Example: Enabling a Package and All Optional Packages

Diagram:

- Thyra
- RTOp
- Teuchos
- Triutils
- Epetra
- EpetraExt

Required Dependence: RTOp → Thyra, Teuchos → Thyra, Thyra → Triutils, Thyra → Epetra, Epetra → EpetraExt

Optional Dependence: RTOp → Teuchos, RTOp → Triutils, RTOp → Epetra, Triutils → Epetra, Teuchos → EpetraExt
Example: Enabling a Package and All Optional Packages

$ ./do-configure -DTrilinos_ENABLE_ALL_PACKAGES:BOOL=OFF \  
-DTrilinos_ENABLE_Thyra:BOOL=ON \  
-DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON

Configuring Trilinos build directory

... 

Enabling all optional packages for current set of enabled packages ...

-- Setting Trilinos_ENABLE_EpetraExt=ON because Trilinos_ENABLE_Thyra=ON
-- Setting Trilinos_ENABLE_Epetra=ON because Trilinos_ENABLE_Thyra=ON
-- Setting Trilinos_ENABLE_Triutils=ON because Trilinos_ENABLE_EpetraExt=ON

Enabling all remaining required packages for the current set of enabled packages ...

-- Setting Trilinos_ENABLE_RTOp=ON because Trilinos_ENABLE_Thyra=ON
-- Setting Trilinos_ENABLE_Teuchos=ON because Trilinos_ENABLE_Thyra=ON

Enabling all optional intra-package enables that can be if both sets of packages are enabled ...

-- Setting EpetraExt_ENABLE_Triutils=ON since Trilinos_ENABLE_EpetraExt=ON AND Trilinos_ENABLE_Triutils=ON
-- Setting Thyra_ENABLE_EpetraExt=ON since Trilinos_ENABLE_Thyra=ON AND Trilinos_ENABLE_EpetraExt=ON
-- Setting Thyra_ENABLE_Epetra=ON since Trilinos_ENABLE_Thyra=ON AND Trilinos_ENABLE_EpetraExt=ON

Final set of enabled packages:  Teuchos RTOp Epetra Triutils EpetraExt Thyra 6
DEMO
Checkout Trilinos From CVS
Open The CMake GUI
Define The Generator

Please select what build system you want CMake to generate files for. You should select the tool that you will use to build the project.

- Visual Studio 9 2008
- Use Defaults
- Compiler Setup
- Cross Compiler Setup

The default compilers will be used.
Configure

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Final set of enabled packages: 0

Probing the environment ...

Configuring individual Trilinos packages ...

Configuring done
Generated VC++ Project Files
Binary Installer
Welcome to the Trilinos 9.0d Setup Wizard

This wizard will guide you through the installation of Trilinos 9.0d.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.
Installed Files
```cpp
#include "Spectra_Version.h"

int main(int argc, _TCHAR* argv[])
{
    cout << Spectra_Version() << endl << endl;
    Spectra_SerialCom error;
    int NumElements = 1000;
    // Construct a map with NumElements and index base of 0
    Spectra_Map<1> Map(NumElements, 0, error);
    // Create x and b vectors
    Spectra_Vector x(Map);
    Spectra_Vector b(Map);
    b.Random();
    x.Update(2.0, 0.0); // x = 2*b
    double bnorm, xnorm;
    x.Norm2(bnorm);
    b.Norm2(xnorm);
    cout << "2 norm of x = " << xnorm << endl << "2 norm of b = " << bnorm << endl;
    int input = 0;
    std::cin >> input;
    return 0;
}
```
Executing The Epetra Example From VC++

```
Epetra Version 3.7d - 09/06/2007

2 norm of x = 36.1182
2 norm of b = 18.0591
```
Future for CMake Trilinos?

• Trilinos to start switching over to CMake immediately?

• Provide prototype versions of CMake build system in Trilinos 9.0.x minor releases?

• Options for next major Trilinos release (March 2009?)
  – A) Maintain full Autotools build system and only provide partial support for CMake build system? (Already done)
  – B) Full support for CMake build system for all released Trilinos packages and maintain basic Autotools build system for library install only? (Most likely)
  – C) Full support for CMake build system for all released Trilinos packages and drop Autotools support? (Least likely)

What does the Trilinos user community think about these options?