CMake: Improving Software Quality and Process

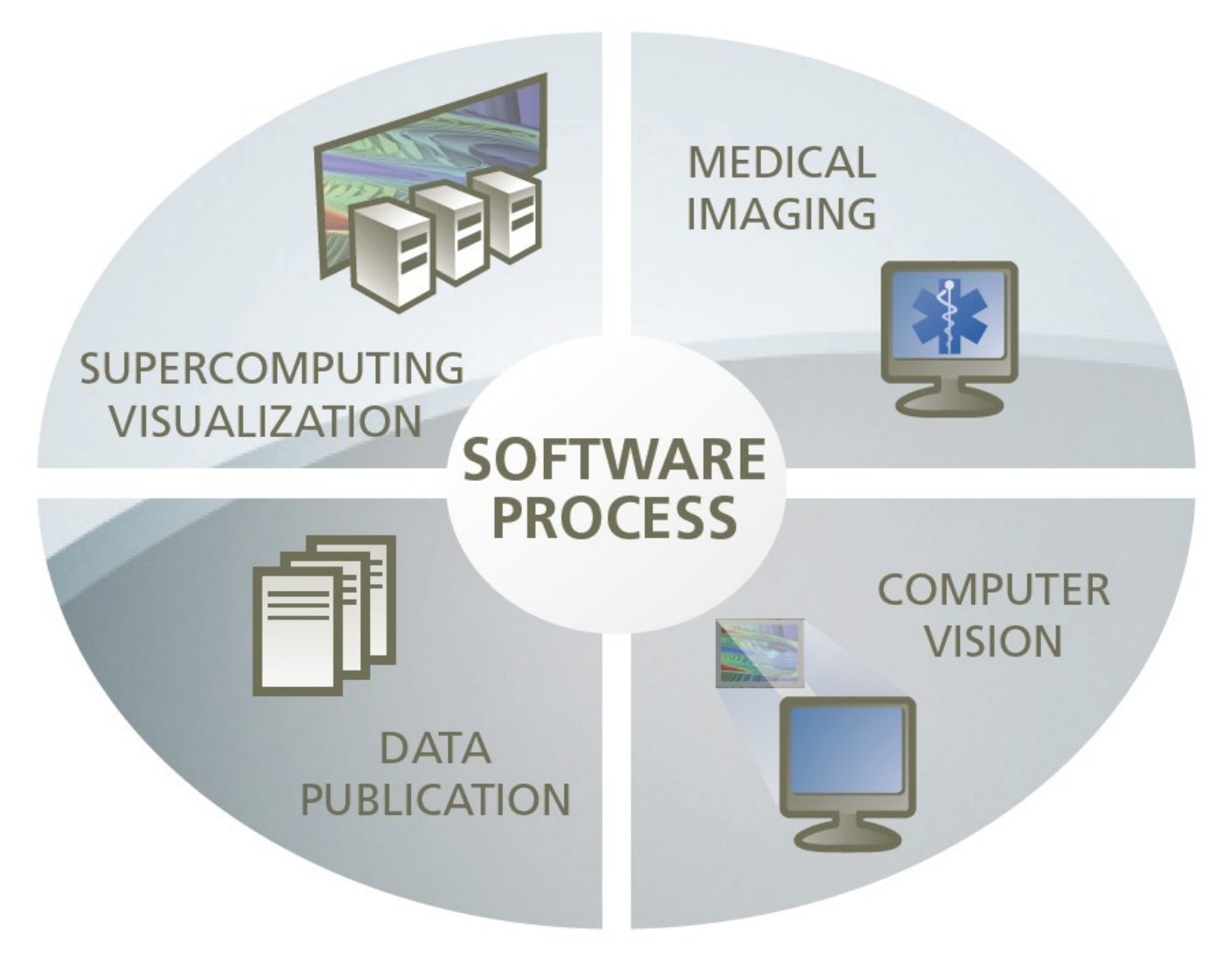
January 17, 2010
Camp KDE

Dr. Marcus D. Hanwell
R&D Engineer
Kitware, Inc.
marcus.hanwell@kitware.com

Kitware: The Company

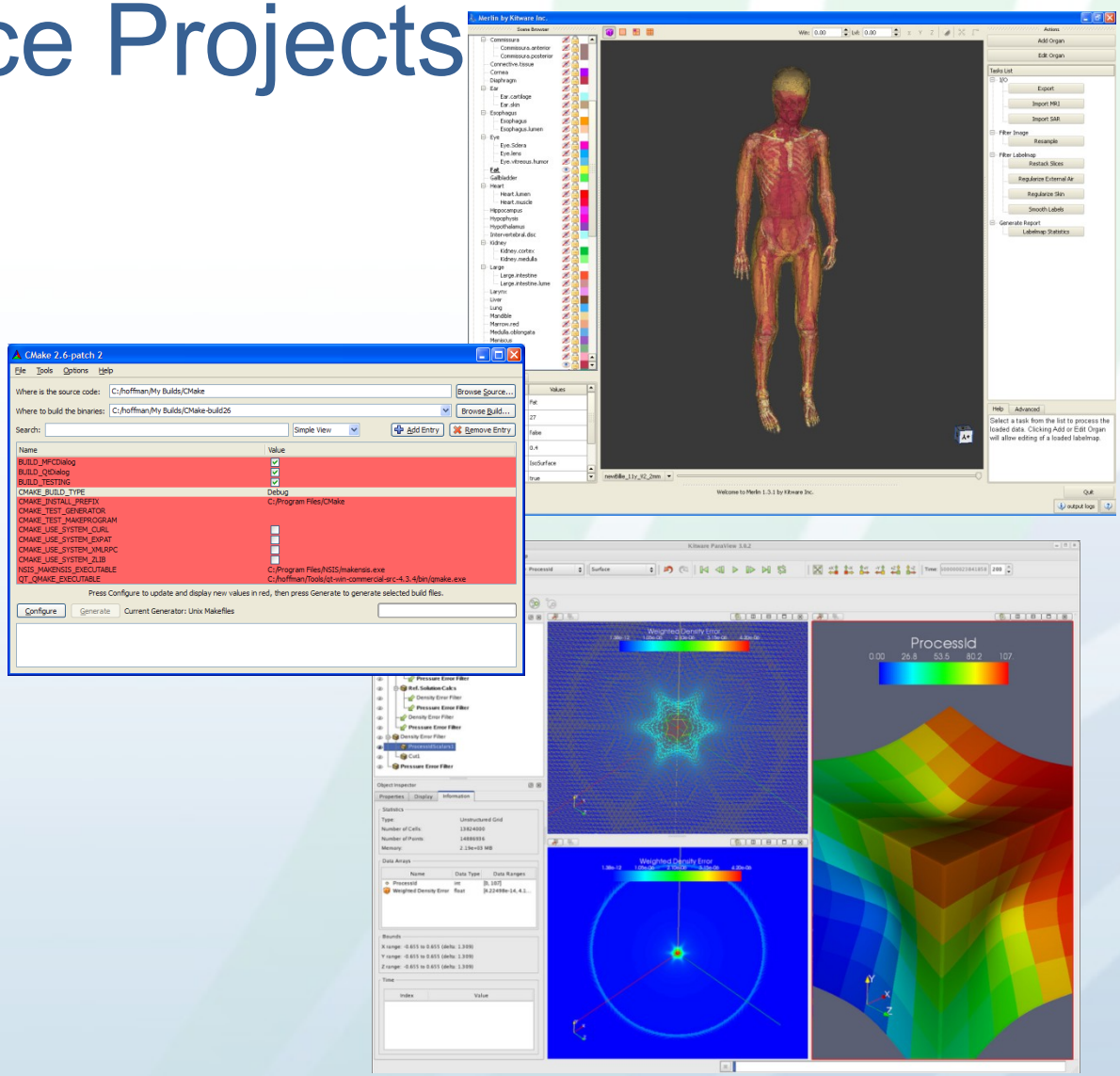
- Founded in 1998
- Founded by 5 previous employees of GE Corporate Research
- Privately held, profitable from creation
- Revenues projected at $9 million in 2009
  - ~$14 million if subcontractors are included
- Approximately 65 employees
  - ~30% growth in 2009
  - More than 30 PhDs

Kitware: Core Technologies



Open Source Projects

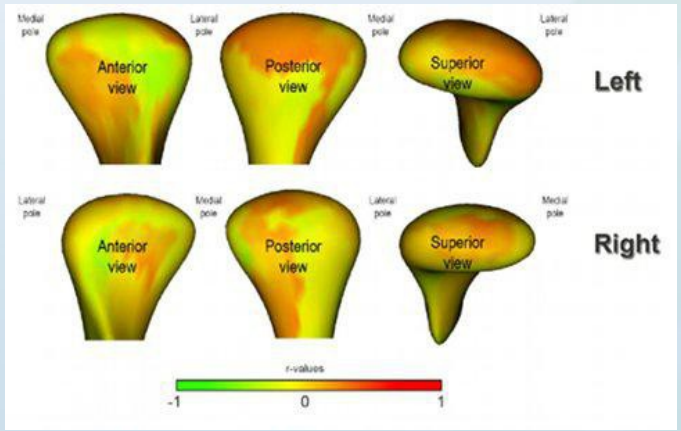
- CMake
- CDash
- VTK
- ITK
- ParaView
- IGSTK
- BatchMake

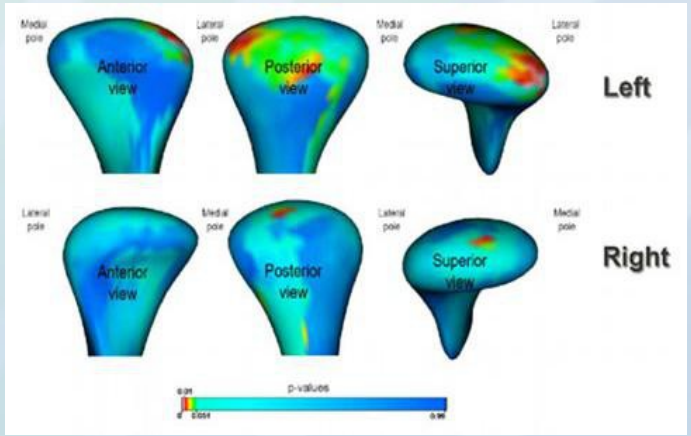


Kitware

NAMIC: National Alliance for Medical Computing

- An NIH national center for biomedical computing
- Kitware leads the NAMIC engineering core
- Committed to open science (source and data)
- New funding to port Slicer (open source) to Qt
- Aimed at leading edge clinical researchers
- More funding bodies are seeing value in open source

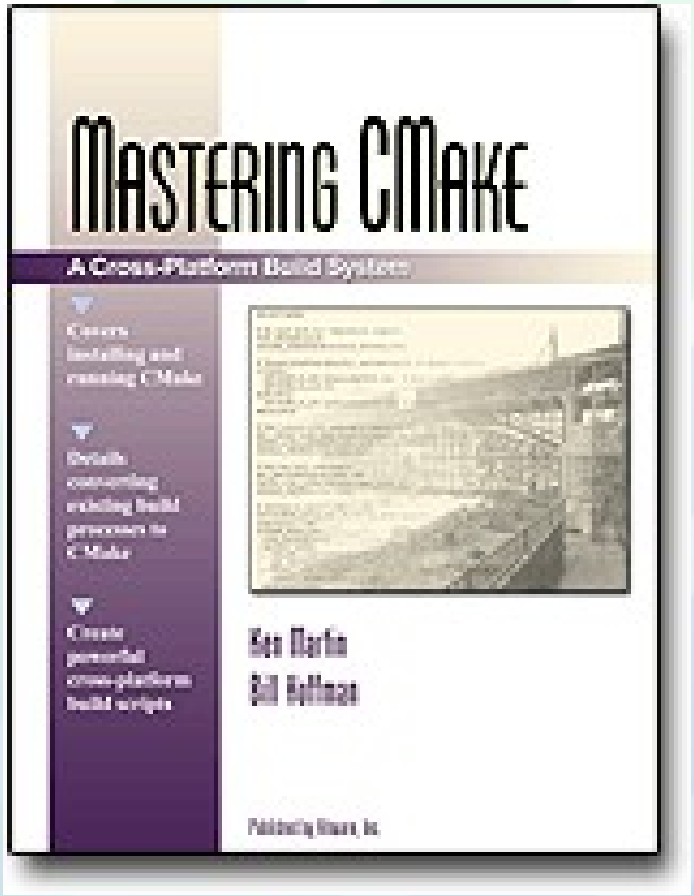




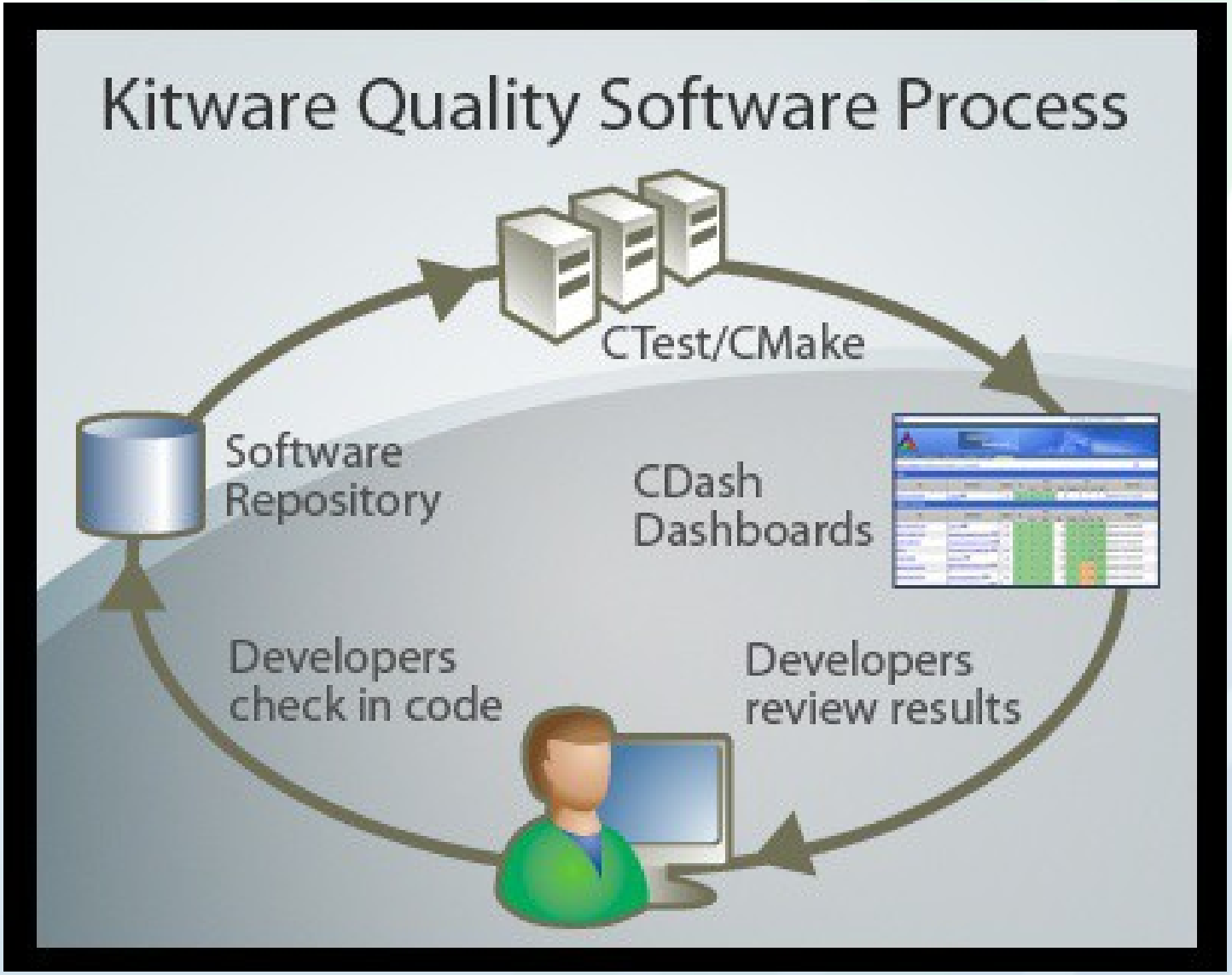
Kitware

CMake: Cross Platform Build System

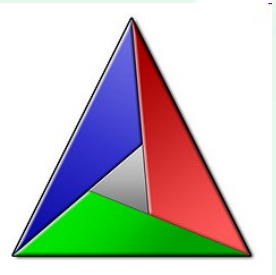
- Why CMake?
- CMake features
- Using CMake
- Creating CMake input files
- Where to get help
  - Mastering CMake book
  - Web page: www.cmake.org
  - http://www.cmake.org/Wiki/CMake
  - mailing list: cmake@cmake.org



Kitware


Kitware Quality Software Process
CTest/CMake
Software Repository
CDash Dashboards
Developers check in code
Developers review results

Key Components

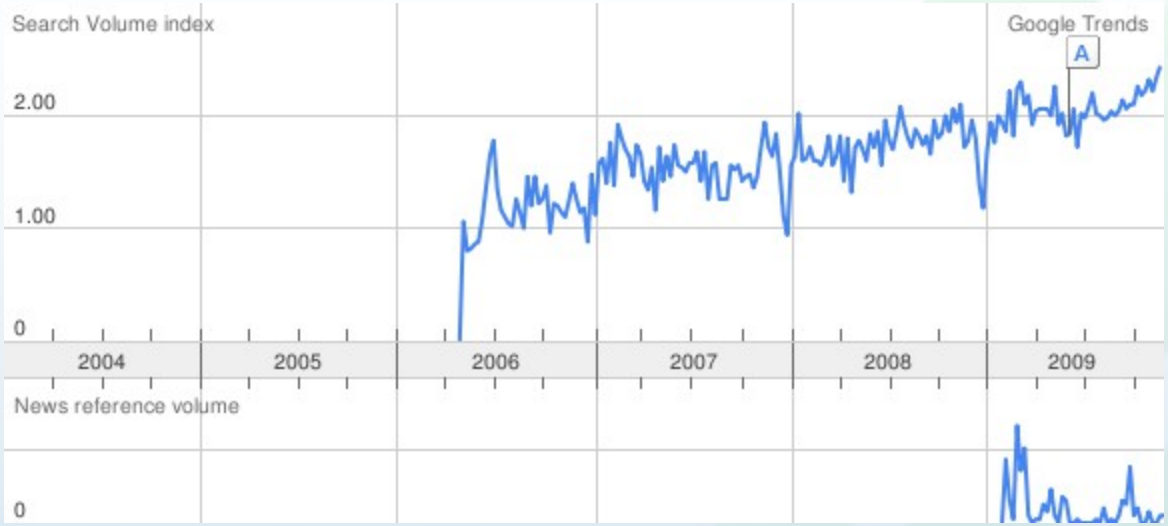


- CMake
  - Cross platform build
- CTest
  - Software testing and submission

- CDash
  - Aggregation of test results on a web server
- CPack
  - Cross platform software packaging

Why CMake? Everyone is using it

KDE 2006 – Tipping Point!



- 1200+ downloads per day from www.cmake.org
- Major Linux distributions and Cygwin provide CMake packages
- KDE, Second Life, Boost (experimentally), many others

Kitware

How CMake Changes The Way We Build C++

- Boost aims to give C++ a set of useful libraries like Java, Python, and C#
- Give C++ compile portability, like the compile once and run everywhere of Java, and C#
  - Same build tool and files for all platforms
  - Easy to mix both large and small libraries

Who Is Involved?

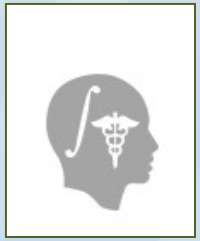
Users

- KDE
- Second Life
- ITK
- VTK
- ParaView
- Trilinos
- Scribus
- Boost (experimentally)
- MySQL
- LLVM
- Many more

Supporters

- Kitware
- ARL
- National Library of Medicine
- Sandia National Labs
- Los Alamos National Labs
- NAMIC

Kitware



Autonomous Systems Use CMake!

Argo is fully automated, with the onboard computers using code built using CMake. The helicopter is also semi-automated with similar code built using CMake!

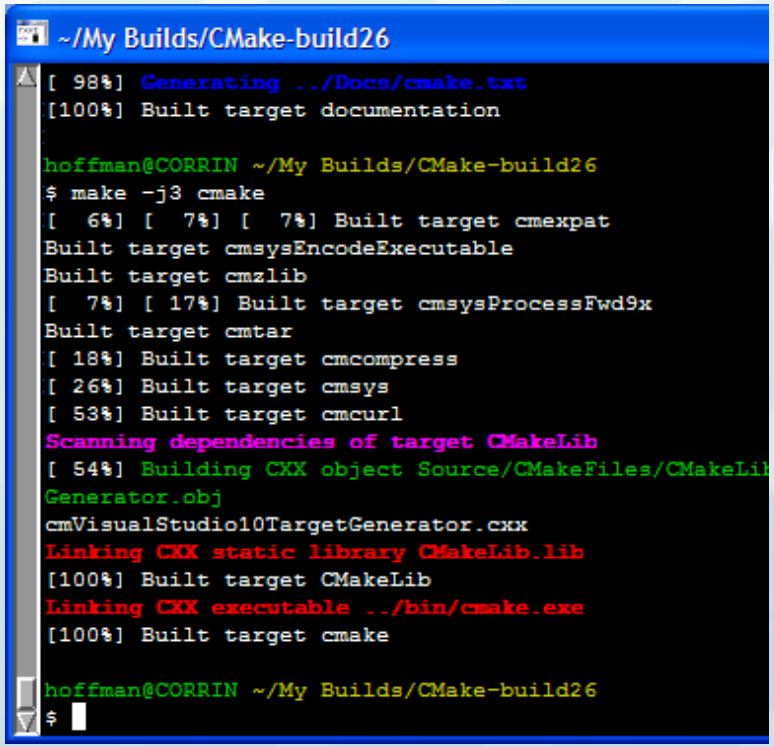
Kitware

CMake Features

- One simple language for all platforms
  - Windows, Mac OS X, Linux, UNIX variants
  - HPC/embedded platforms via cross-compilation (ParaView/Python)
- Generates native build systems
  - Makefiles (GNU, NMake, Borland, etc.)
  - KDevelop, Eclipse
  - Visual Studio 6, 7, 8, 9 IDE
  - Xcode
- Out-of-source build trees leave source clean
- Interactive configuration via GUI
- Multiple configurations (debug, release, etc.)

CMake Features - continued

- Automatic analysis
  - Implicit dependencies (C, C++, Fortran)
  - Transitive link dependencies
  - Ordering of linker search path and RPATH
- Advanced Makefile generation
  - Modular, Fast, Parallel
  - Color and progress display
  - Help targets – make help
  - Preprocessor targets – make foo.i
  - Assembly targets – make foo.s



Kitware

Input to CMake

- Simple scripting language in CMakeLists.txt file(s)
- Built-in commands for common rules
  - `add_library(MyLib MyLib.cxx)`
  - `add_executable(MyExe MyMain.cxx)`
- Example project using Qt 4 and new syntax:

```
cmake_minimum_required(VERSION 2.8)
project(QtProject)
find_package(Qt4 4.5.2
  COMPONENTS QtCore QtGui QtOpenGL REQUIRED)
include(${QT_USE_FILE})
add_executable(mycoolapp main.cpp)
target_link_libraries(mycoolapp ${QT_LIBRARIES})
```

Input to CMake: KDE Application

- Very little needs to change,

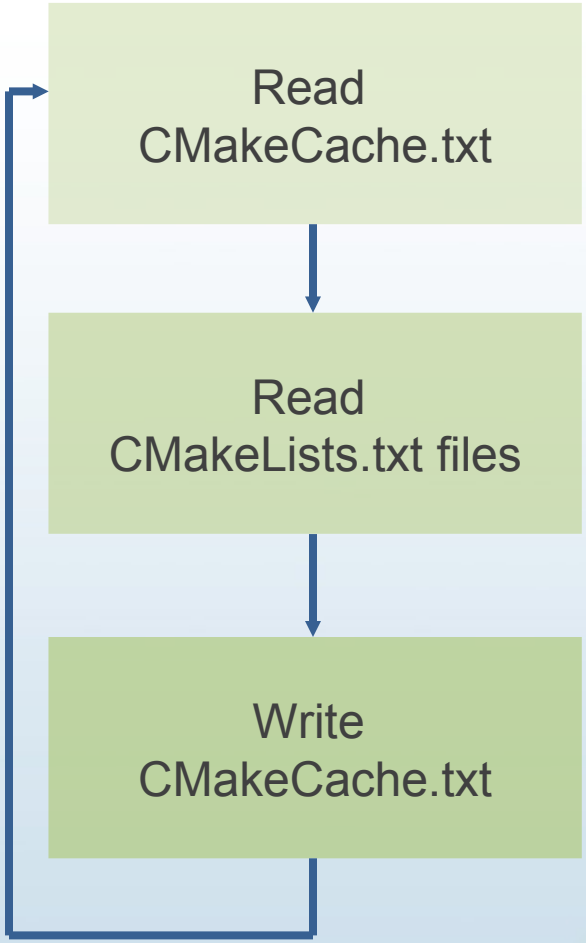
```
cmake_minimum_required(VERSION 2.8)
project(KDEProject)
find_package(KDE4 4.4 REQUIRED)
include(KDE4Defaults)
include_directories(${KDE4_INCLUDES})
kde4_add_executable(mykdeapp main.cpp)
target_link_libraries(mykdeapp
  ${KDE4_KDEUI_LIBS})
install(TARGETS mykdeapp
  ${INSTALL_TARGETS_DEFAULT_ARGS})
```

CMake Scripts

- cmake -E command
  - Cross platform command line utility
  - E.g. copy or remove files, conditionally copy...
- cmake -P script.cmake
  - Cross platform scripting utility
  - Does not generate a CmakeCache
  - Ignores all commands specific to build environment generation

CMake Process

Configure Step



Generate Step

Write
Makefiles or projects

Kitware

Automatic Testing Benefits

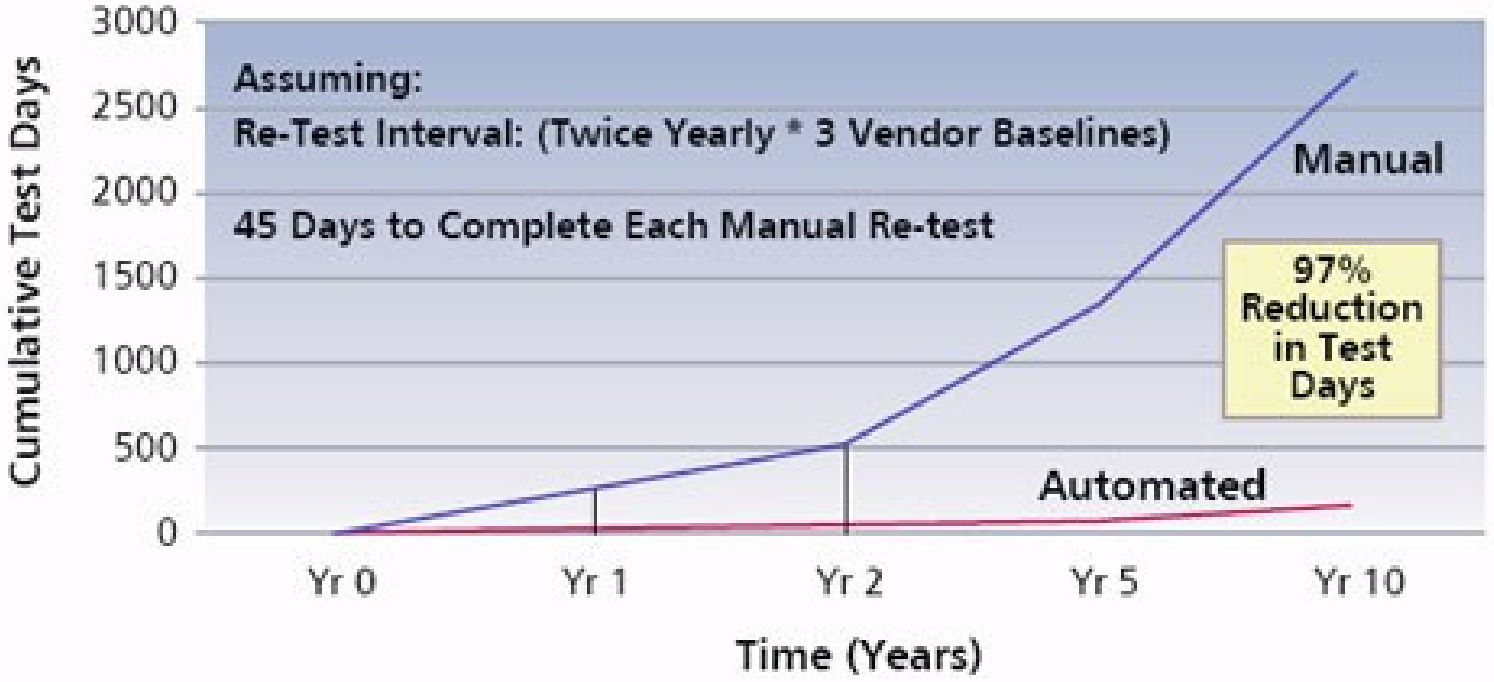
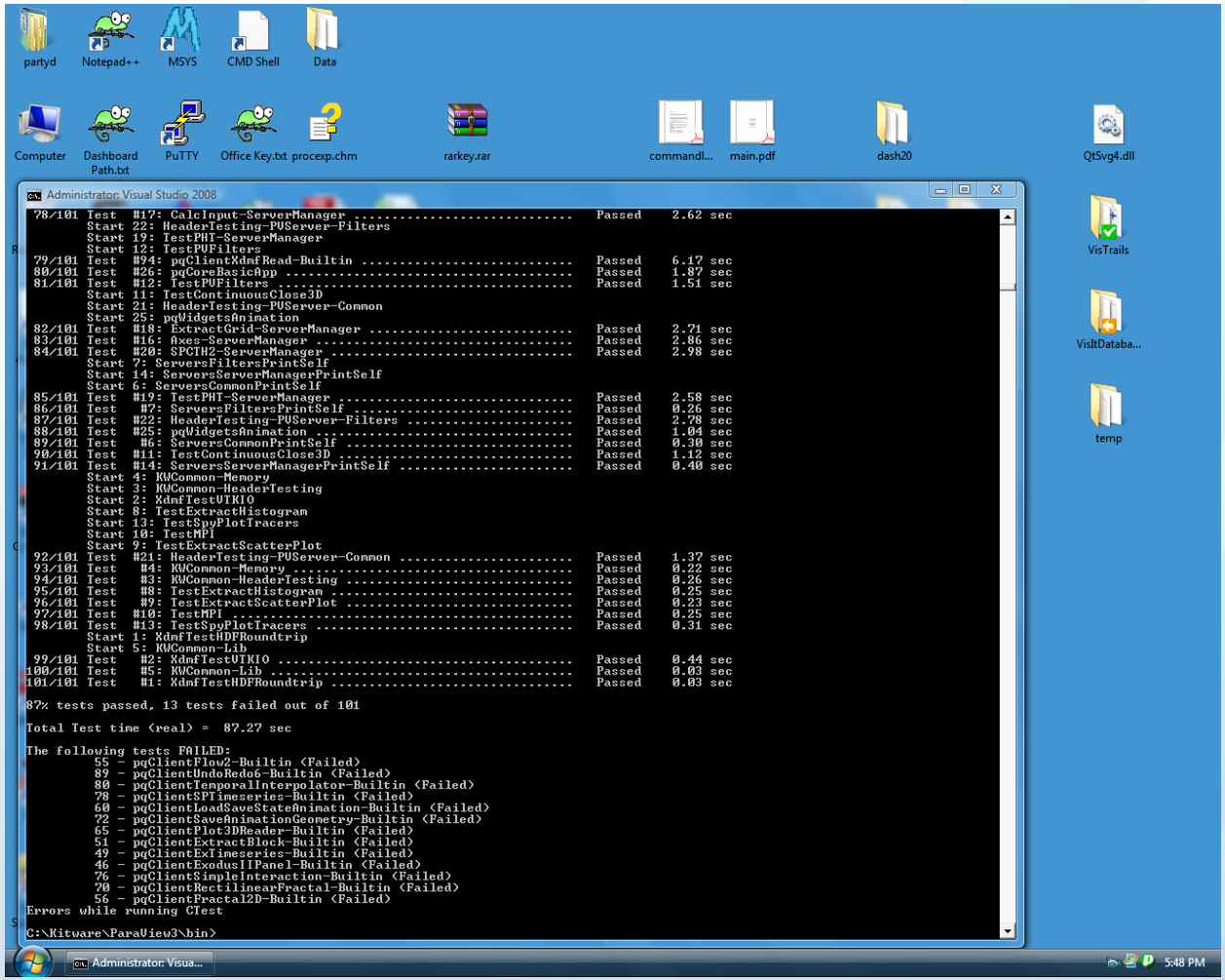


Figure 2. "Example Automated Software Testing Savings over Time"

"Automated Software Testing,"
1999, Dustin, et al, Addison Wesley

Kitware

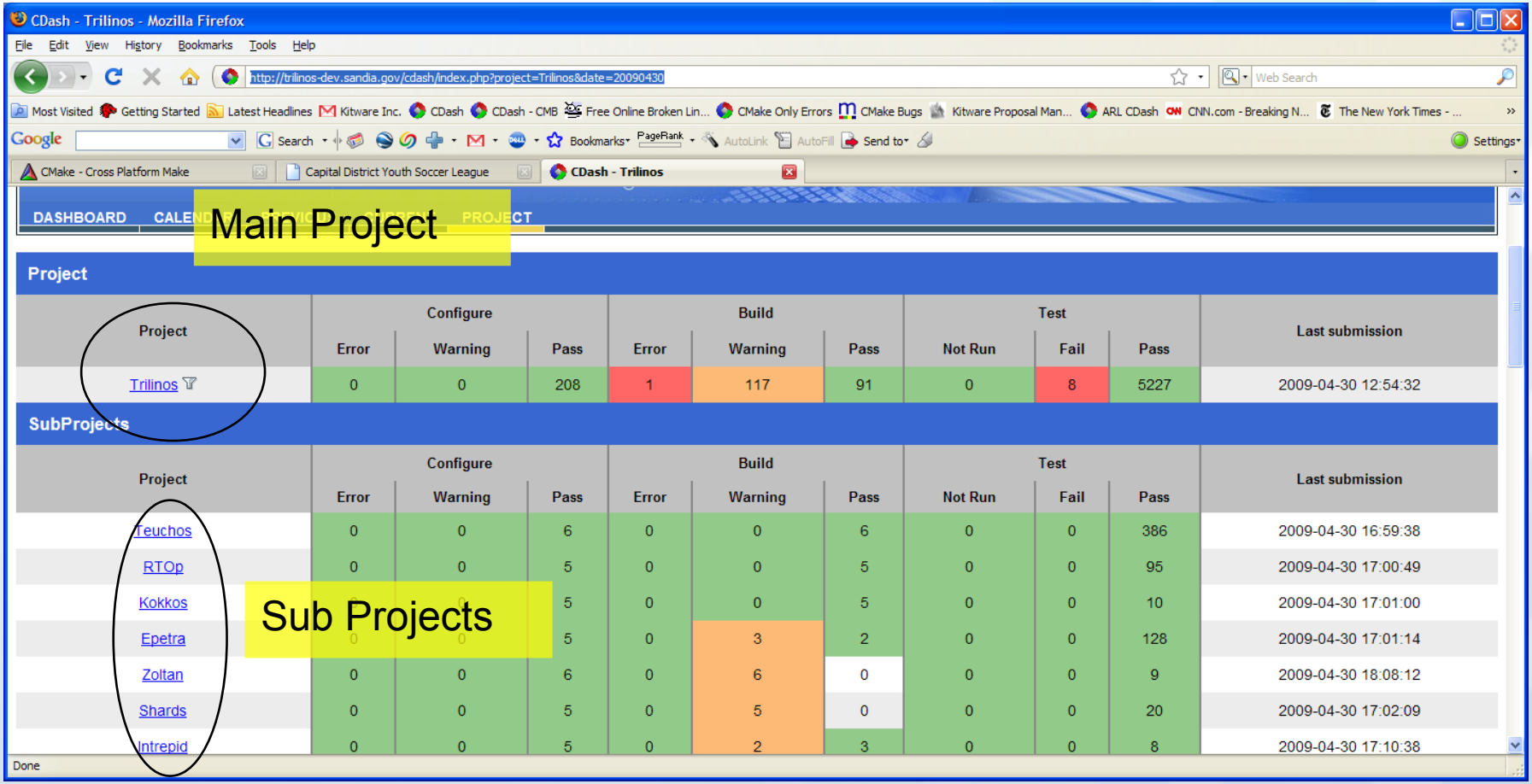
Video of ParaView Nightly Testing



CDash Dashboard www.cdash.org

Kitware

CDash Subproject Support



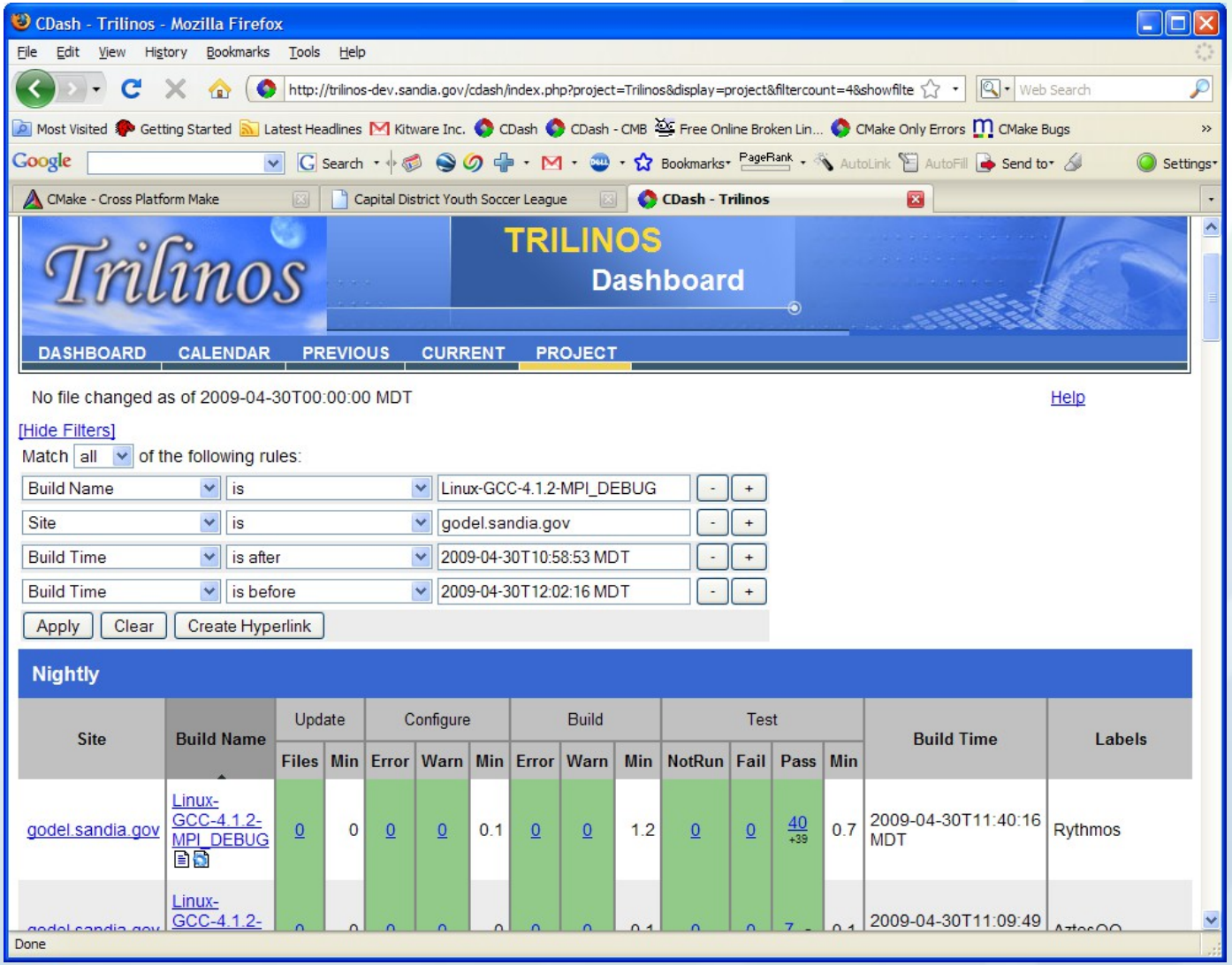
Project	Configure Error	Configure Warning	Configure Pass	Build Error	Build Warning	Build Pass	Test Not Run	Test Fail	Test Pass	Last submission
Trilinos	0	0	208	1	117	91	0	8	5227	2009-04-30 12:54:32

SubProjects

Project	Configure Error	Configure Warning	Configure Pass	Build Error	Build Warning	Build Pass	Test Not Run	Test Fail	Test Pass	Last submission
Teuchos	0	0	6	0	0	6	0	0	386	2009-04-30 16:59:38
RTOp	0	0	5	0	0	5	0	0	95	2009-04-30 17:00:49
Kokkos	0	0	5	0	0	5	0	0	10	2009-04-30 17:01:00
Epetra	0	0	5	0	3	2	0	0	128	2009-04-30 17:01:14
Zoltan	0	0	6	0	6	0	0	0	9	2009-04-30 18:08:12
Shards	0	0	5	0	5	0	0	0	20	2009-04-30 17:02:09
Intrepid	0	0	5	0	2	3	0	0	8	2009-04-30 17:10:38

Kitware

CDash: Query Filters



Kitware

CTest Command Wrappers

Build Time:2009-05-04T01:53:37 MDT

Found 1 Warnings

Errors are here.

Warning while building C++ object file "CMakeFiles/Kokkos_BaseSparseSolve.dir/cxx_main.cpp.o" in target Kokkos_BaseSparseSolve.	
Source File	packages/kokkos/test/BaseSparseSolve/cxx_main.cpp
Label	Kokkos
Command	"/Users/bmpersc/bin/gcc-4.3.3/bin/g++" "-mmacosx-version-min=10.5" "-ansi" "-pedantic" "-Wall" "-Wno-long-long" "-Wwrite-strings" "-g" "-O0" "-D_GLIBCXX_DEBUG" "-I/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/BUILD/packages/kokkos/src" "-I/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src" "-I/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/../BaseSparseMultiply" "-o" "CMakeFiles/Kokkos_BaseSparseSolve.dir/cxx_main.cpp.o" "-c" "/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/cxx_main.cpp"
Directory	/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/BUILD/packages/kokkos/test/BaseSparseSolve
Exit Condition	0
Standard Output	
Standard Error	/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp: In member functio /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/cxx_main.cpp:262: instanti /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:646: warning: sugg /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:693: warning: sugg /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp: In member functio /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/cxx_main.cpp:287: instanti /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:541: warning: sugg /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:583: warning: sugg

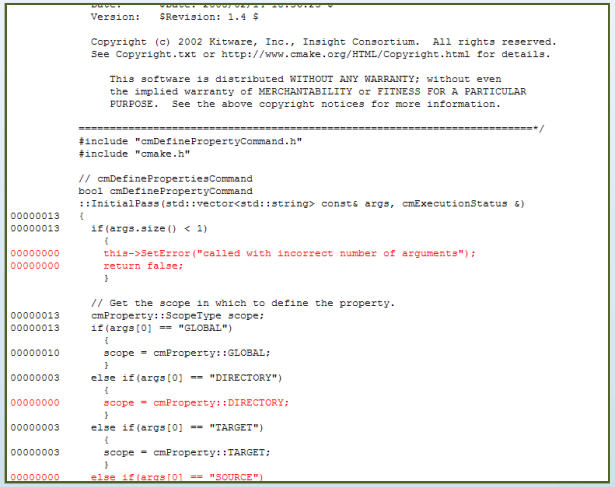
Kitware

CDash 1.5.0 © 2009 Kitware Inc.
[report problems]

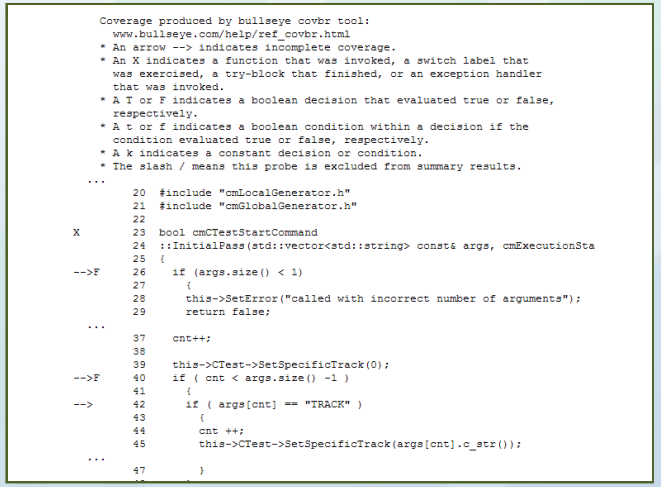
Kitware

Coverage: GCov/Bullseye

./Source/CTest/cmCTestUpdateHandler.cxx	68.21%	45	1
./Source/cmMakefileLibraryTargetGenerator.cxx	68.48%	60	2
./Source/cmTargetLinkLibrariesCommand.cxx	69.17%	17	1
./Source/cmGetPropertyCommand.cxx	69.31%	36	2
./Source/cmExportInstallFileGenerator.cxx	69.32%	16	2
./Source/kwsys/ProcessUNIX.c	69.33%	371	11
./Source/cmVariableWatch.cxx	69.44%	8	1
./Source/cmSystemTools.h	69.64%	1	5
./Source/cmComputeLinkDepends.cxx	69.89%	78	5
./Source/CTest/cmCTestStartCommand.cxx	70.00%	12	0
./Source/cmMakefileExecutableTargetGenerator.cxx	70.83%	16	1
./Source/cmLinkLibrariesCommand.cxx	70.83%	7	0
./Source/cmMakeDepend.cxx	71.01%	44	1
./Source/CTest/cmCTestBuildCommand.cxx	71.74%	26	0
./Source/cmsys/auto_ptr.hxx	71.88%	1	1
./Source/kwsys/testCommandLineArguments.cxx	71.88%	7	1
./Source/CTest/cmCTestSVN.cxx	72.07%	57	2
./Source/cmScriptGenerator.cxx	72.34%	20	1



```
            Version:   $Revision: 1.4 $

            Copyright (c) 2002 Kitware, Inc., Insight Consortium.  All rights reserved.
            See Copyright.txt or http://www.cmake.org/HTML/Copyright.html for details.

               This software is distributed WITHOUT ANY WARRANTY; without even
               the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
               PURPOSE.  See the above copyright notices for more information.

          ============================================================================*/
          #include "cmDefinePropertyCommand.h"
          #include "cmake.h"

          // cmDefinePropertiesCommand
          bool cmDefinePropertyCommand
          ::InitialPass(std::vector<std::string> const& args, cmExecutionStatus &)
00000013  {
00000013    if(args.size() < 1)
              {
00000000      this->SetError("called with incorrect number of arguments");
00000000      return false;
              }


            // Get the scope in which to define the property.
00000013    cmProperty::ScopeType scope;
00000013    if(args[0] == "GLOBAL")
              {
00000010      scope = cmProperty::GLOBAL;
              }
00000003    else if(args[0] == "DIRECTORY")
              {
00000000      scope = cmProperty::DIRECTORY;
              }
00000003    else if(args[0] == "TARGET")
              {
00000003      scope = cmProperty::TARGET;
              }
00000000    else if(args[0] == "SOURCE")
```



```
            Coverage produced by bullseye covbr tool:
              www.bullseye.com/help/ref_covbr.html
            * An arrow --> indicates incomplete coverage.
            * An X indicates a function that was invoked, a switch label that
              was exercised, a try-block that finished, or an exception handler
              that was invoked.
            * A T or F indicates a boolean decision that evaluated true or false,
              respectively.
            * A t or f indicates a boolean condition within a decision if the
              condition evaluated true or false, respectively.
            * A k indicates a constant decision or condition.
            * The slash / means this probe is excluded from summary results.
  ...
                20  #include "cmLocalGenerator.h"
                21  #include "cmGlobalGenerator.h"
                22
X               23  bool cmCTestStartCommand
                24  ::InitialPass(std::vector<std::string> const& args, cmExecutionSta
                25  {
-->F            26    if (args.size() < 1)
                27      {
                28      this->SetError("called with incorrect number of arguments");
                29      return false;
  ...
                37    cnt++;
                38
                39    this->CTest->SetSpecificTrack(0);
-->F            40    if ( cnt < args.size() -1 )
                41      {
-->             42      if ( args[cnt] == "TRACK" )
                43        {
                44        cnt ++;
                45        this->CTest->SetSpecificTrack(args[cnt].c_str());
  ...
                47        }
```

Kitware

Valgrind/Purify

Dynamic analysis started on 2009-05-03 03:36:06

Site Name: dash17.kitware
Build Name: Linux-g++4.0

Name	Status	Memory Leak	Uninitialized Memory Read	Potential Memory Leak	Uninitialized Memory Conditional	Mismatched Deallocate	Freeing Invalid Memory	Invalid Pointer Read	Invalid Pointer Write	Labels
QtChart-TestBarSeriesColors	Passed		1	25						
QtChart-TestChartWidget	Passed		1	26						
Mace	Passed			2						
TestHyperOctreeContourFilter	Passed			2	1					
TestUncertaintyTubeFilter	Passed			2						
TestMultiBlock	Passed			2						
TemporalStatistics	Passed			3						
TestGenericCutter	Passed			2						
TestActorLightingFlag	Passed			2						
TestLabelPlacer	Passed			2						
TestOpacity	Passed			2						
TestTextActor3DAlphaBlending	Passed			2						
TestAreaSelections	Passed			2						
TestTranslucentImageActorDepthPeeling	Passed		2	2						
TestGenericVertexAttributesGLSLDepthPeelingPass	Passed			2						

Find: asio Next Previous Highlight all Match case

Dynamic analysis started on 2009-05-04 03:37:17

Site Name: dash17.kitware
Build Name: Linux-g++4.0
TestMultiBlockPassed

```
==3002== Memcheck, a memory error detector.
==3002== Copyright (C) 2002-2007, and GNU GPL'd, by Julian Seward et al.
==3002== Using LibVEX rev 1732, a library for dynamic binary translation.
==3002== Copyright (C) 2004-2007, and GNU GPL'd, by OpenWorks LLP.
==3002== Using valgrind-3.2.3, a dynamic binary instrumentation framework.
==3002== Copyright (C) 2000-2007, and GNU GPL'd, by Julian Seward et al.
==3002== For more details, rerun with: -v
==3002==
==3002==
==3002== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 119 from 2)
==3002== malloc/free: in use at exit: 30,294 bytes in 327 blocks.
==3002== malloc/free: 37,724 allocs, 37,397 frees, 5,207,986 bytes allocated.
==3002== For counts of detected errors, rerun with: -v
==3002== searching for pointers to 327 not-freed blocks.
==3002== checked 2,298,764 bytes.
==3002==
 ==3002== 64 bytes in 1 blocks are still reachable in loss record 15 of 34
==3002==    at 0x401DC87: realloc (vg_replace_malloc.c:306)
==3002==    by 0x62F83E5: (within /usr/lib/libX11.so.6.2.0)
==3002==    by 0x62F908E: (within /usr/lib/libX11.so.6.2.0)
==3002==    by 0x62F95F0: XrmGetStringDatabase (in /usr/lib/libX11.so.6.2.0)
==3002==    by 0x659FB22: (within /usr/lib/libXt.so.6.0.0)
==3002==    by 0x65A0ED4: _XtDisplayInitialize (in /usr/lib/libXt.so.6.0.0)
==3002==    by 0x6596DC7: XtOpenDisplay (in /usr/lib/libXt.so.6.0.0)
==3002==    by 0x437DD13: vtkXRenderWindowInteractor::Initialize() (vtkXRenderWindowInteractor.cxx:317)
==3002==    by 0x42EFDD0: vtkRenderWindow::Render() (vtkRenderWindow.cxx:265)
==3002==    by 0x441E401: vtkXOpenGLRenderWindow::Render() (vtkXOpenGLRenderWindow.cxx:1846)
==3002==    by 0x8081AB6: TestMultiBlock(int, char**) (TestMultiBlock.cxx:142)
==3002==    by 0x805B2E8: main (GraphicsCxxTests.cxx:306)
==3002==
```

Kitware

Email Notification

A submission to CDash for the project CMake has failing tests.
You have been identified as one of the authors who have checked
in changes that are part of this submission or you are listed in the default contact list.

Details on the submission can be found at http://www.cdash.org/CDash/buildSummary.php?buildid=322849

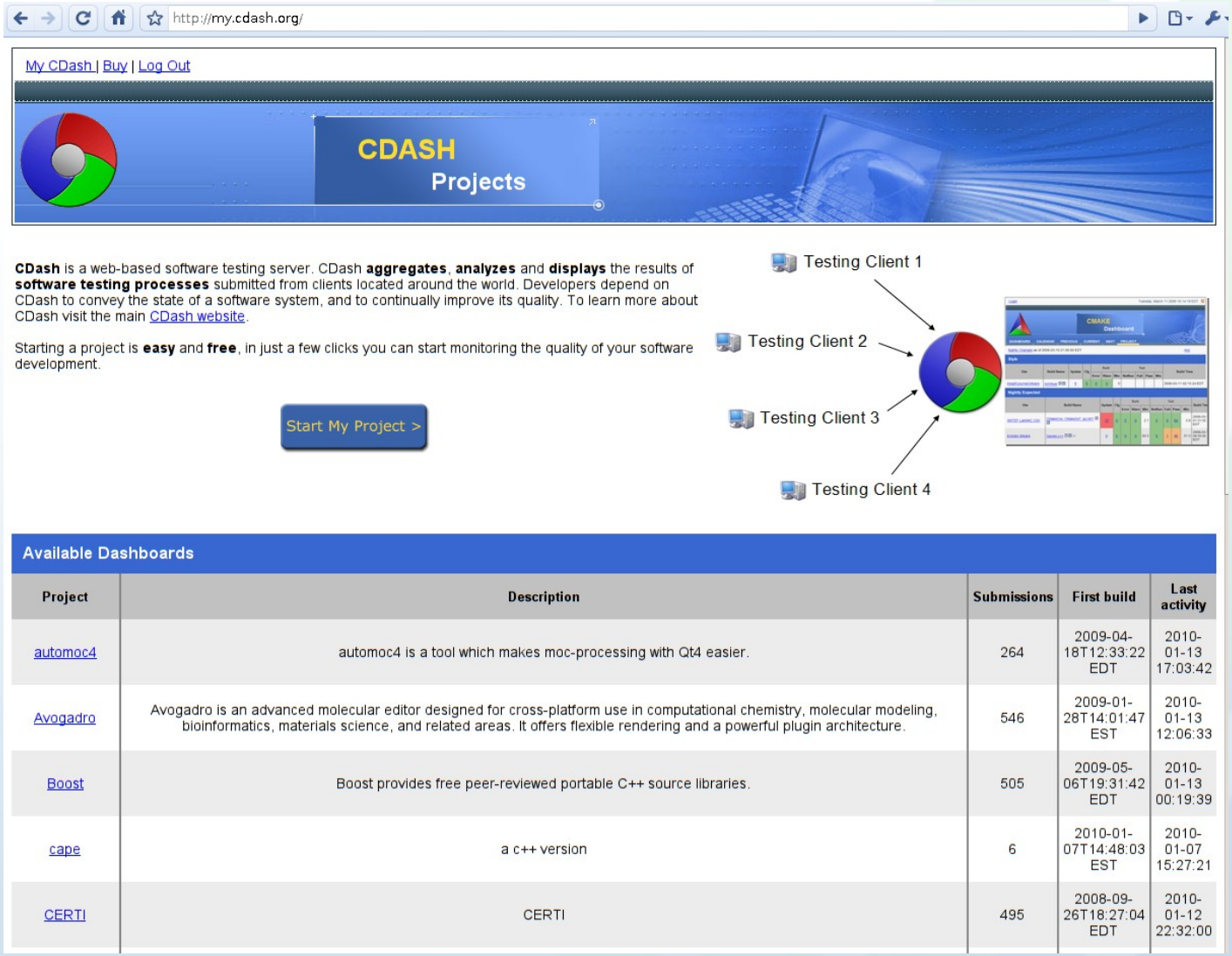
Project: CMake
Site: destiny.kitware
Build Name: HP-UX-aCC
Build Time: 2009-04-29T14:28:00 EDT
Type: Continuous
Tests failing: 85

Tests failing (first 5)
SystemInformationNew (http://www.cdash.org/CDash/testDetails.php?test=21959894&build=322849)
CommandLineTest (http://www.cdash.org/CDash/testDetails.php?test=21959897&build=322849)
FindPackageTest (http://www.cdash.org/CDash/testDetails.php?test=21959898&build=322849)
FindModulesExecuteAll (http://www.cdash.org/CDash/testDetails.php?test=21959899&build=322849)
StringFileTest (http://www.cdash.org/CDash/testDetails.php?test=21959900&build=322849)

-CDash on www.cdash.org

Kitware

Kitware Hosted CDash http://my.cdash.org/

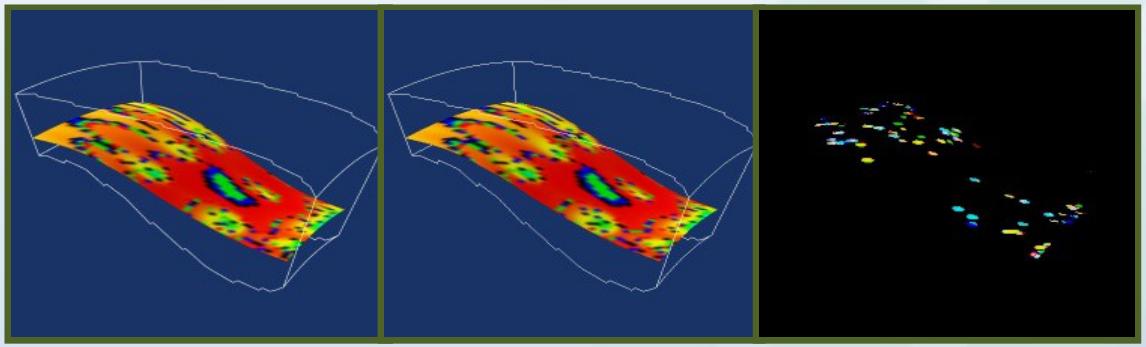


Project	Description	Submissions	First build	Last activity
automoc4	automoc4 is a tool which makes moc-processing with Qt4 easier.	264	2009-04-18T12:33:22 EDT	2010-01-13 17:03:42
Avogadro	Avogadro is an advanced molecular editor designed for cross-platform use in computational chemistry, molecular modeling, bioinformatics, materials science, and related areas. It offers flexible rendering and a powerful plugin architecture.	546	2009-01-28T14:01:47 EST	2010-01-13 12:06:33
Boost	Boost provides free peer-reviewed portable C++ source libraries.	505	2009-05-06T19:31:42 EDT	2010-01-13 00:19:39
cape	a c++ version	6	2010-01-07T14:48:03 EST	2010-01-07 15:27:21
CERTI	CERTI	495	2008-09-26T18:27:04 EDT	2010-01-12 22:32:00

Kitware

CDash Testing

- Purify / valgrind
- Coverage (gcov, bullseye)
- Configuration coverage
  - Make sure different OSes, libraries and options are covered
- Image difference testing



Testing with CMake/CTest

- Testing command in CMake
  - add_test (testName exeName arg1 arg2 …)
  - Executable is expected to return 0 for passed
  - Other passing criteria can be specified
- CTest – an executable that is distributed with CMake, can run tests in a project
  - Used for continuous integration testing
  - Client for CDash

CTest/CDash: Current State

- Alexander Neundorf has already done a lot of work getting KDE dashboards up
- Currently testing full KDE modules
- If all developers start receiving email
  - Is a KStars dev interested in kalzium errors?
  - How long until they filter/unsubscribe?
  - Marble devs only receive Marble reports

Build Name	Update		Configure			Build			Test			
	Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min
FreeBSD-cmake-2.8.0	12	0.1	0	0	0.8	0	1	20.6	147	466	0	0.1
Linux-c++	12	0.4	0	0	0.2	3 +3	22 -11	10.5	2 +2	14 -1	720 -1	66.1
Linux-gcc-4.3.2-cmake-2.6.2	12	0.2	0	0	0.2	0	8 -20	12.2	147	466	0	0.1
SunOS-sunstudio-12.1-cmake-2.6.2	12	0.2	0	0	0.6	0	190 -10	12.7	147	466	0	0.2

Kitware

KDE and Subprojects

- KDE modules are quite large
  - Typically more than 15 components
  - Developers don't normally work on all components in a module
  - Without subprojects testing is too noisy
- Divide KDE modules?
- Split modules into subprojects?
  - New feature in CMake 2.8

Subproject Requirements

- Add LABELS property to build system
  - Source
  - Tests

```
set_target_properties(avogadro PROPERTIES
  VERSION ${Avogadro_VERSION_FULL} SOVERSION 1
  LABELS "libavogadro")
```

- Create custom targets for subprojects
- CMake 2.8 CTest driver scripts
  - Only need CMake 2.8 on submission hosts
  - Continue using >=CMake 2.6.2 to build

What Subprojects Get Us

- Build KDE modules in parts
- Test KDE modules in parts
- Developers receive emails about issues
  - Only for the subproject they committed to
  - CDash summaries for each part
    - Kalzium
    - Marble
  - Not everyone who committed to kdeedu!

CTest Driver Script

- Update the source directory
  - CTest added Git support in 2.8
- Configure the module
- Loop over each subproject in the module
  - Build the custom subproject target
  - Run the tests for the subproject
- Submit the result of each part
- Uses new CTest script functionality

Example CTest Driver Script

```
cmake_minimun_required(VERSION 2.8)
set(CTEST_SOURCE_DIRECTORY $ENV{HOME}/kdeedu)
set(CTEST_BINARY_DIRECTORY $ENV{HOME}/kdeedubuild)
set(CTEST_CMAKE_GENERATOR "Unix Makefiles")
set(CTEST_BUILD_COMMAND "make -j9")

# Empty the binary directory - clean build
ctest_empty_binary_directory("${CTEST_BINARY_DIRECTORY}")
```

Example CTest Driver Script

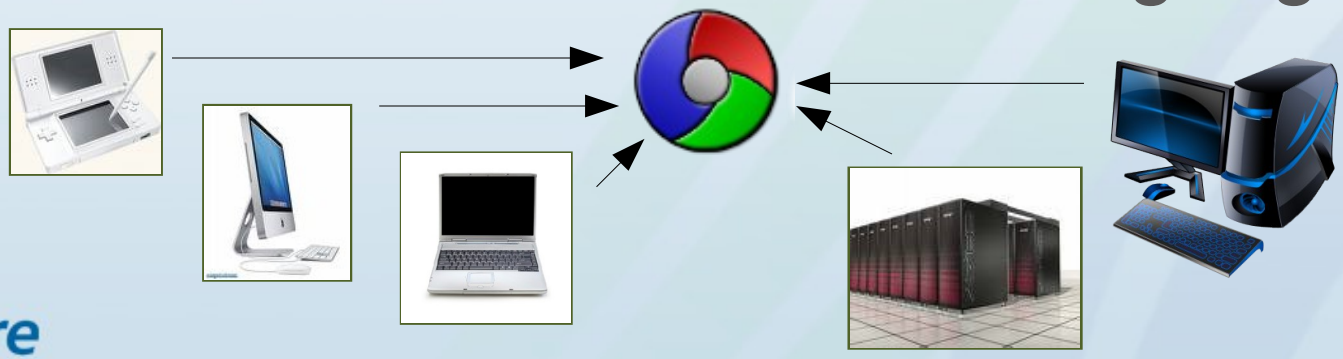
```
# Write initial cache
file(WRITE
  "${CTEST_BINARY_DIRECTORY}/CMakeCache.txt" "
  CMAKE_BUILD_TYPE:String=DebugFull
  CMAKE_CXX_COMPILER:FILEPATH=/usr/ccache/bin/c++
")
# Now start the update and configure steps
set(CTEST_UPDATE_COMMAND "svn")
ctest_start(Nightly)
ctest_update(SOURCE "${CTEST_SOURCE_DIRECTORY}")
ctest_submit(PARTS Update Notes)
ctest_configure(BUILD "${CTEST_BINARY_DIRECTORY}" APPEND)
ctest_submit(PARTS Configure)
```

Example CTest Driver Script

```
include(
  "${CTEST_SOURCE_DIRECTORY}/CTestConfig.cmake")
foreach(sub ${CTEST_PROJECT_SUBPROJECTS})
  set_property(GLOBAL PROPERTY SubProject ${sub})
  set_property(GLOBAL PROPERTY Label ${sub})
  set(CTEST_BUILD_TARGET ${sub})
  ctest_build(BUILD "${CTEST_BINARY_DIRECTORY}" APPEND)
  ctest_submit(PARTS Build)
  ctest_test(BUILD "${CTEST_BINARY_DIRECTORY}"
    INCLUDE_LABEL $sub)
  ctest_submit(PARTS Test)
endforeach()
```

Using CTest Driver Scripts

- Invoke the CTest binary in script mode
  - ctest -S /path/to/driver_script.cmake
- Can easily be run from a cron job, etc.
- Can submit parts of the submission
- Allows for much tighter control, scripting
- Access to most of the CMake language



Kitware

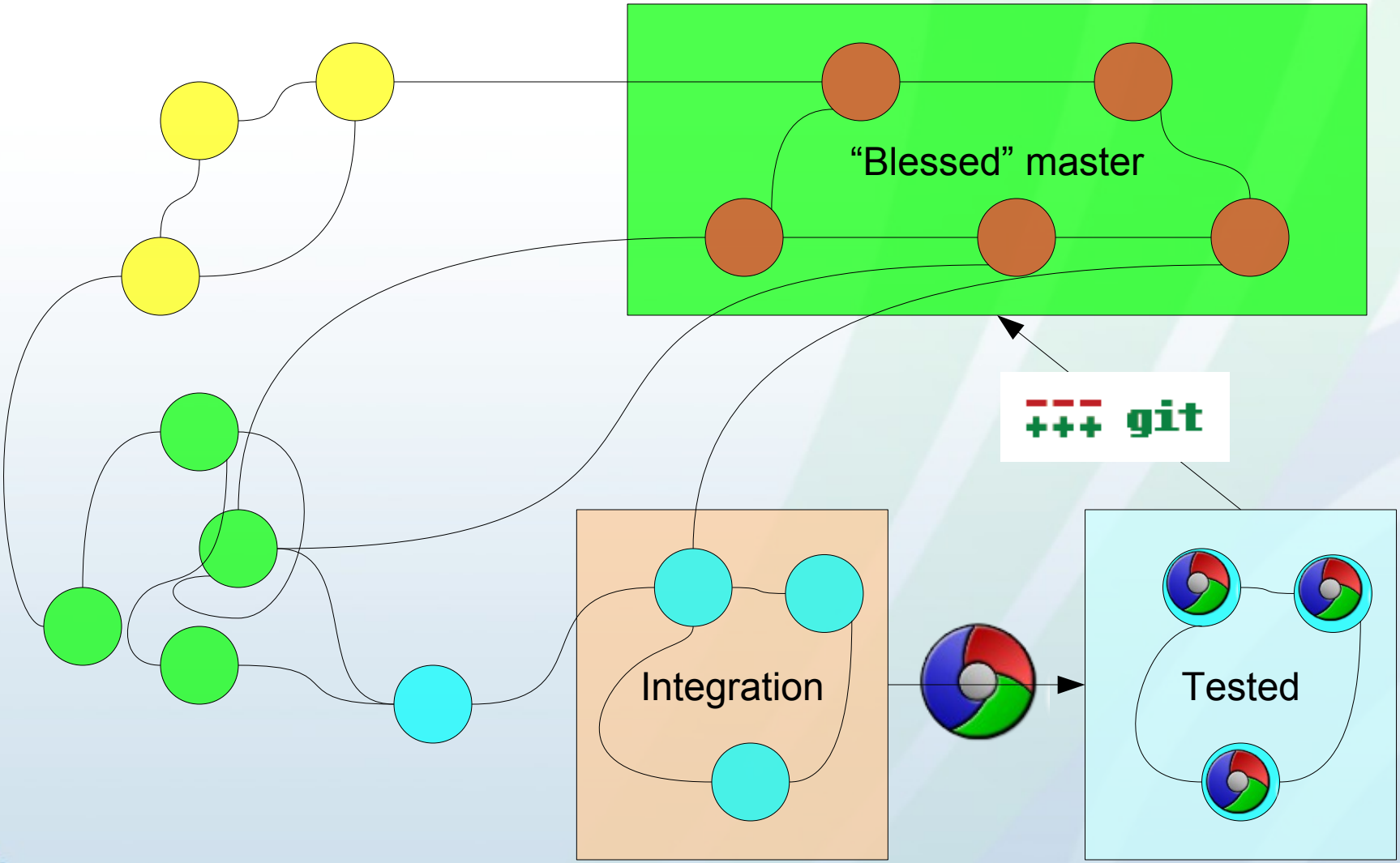
CTest and Git

- CTest must understand version control
  - What files have changed?
  - Who changed those files?
  - What to checkout for a nightly dashboard?
- CMake 2.8 added support for Git, Mercurial and Bazaar (distributed VCS)
- Also abstracted support
  - Easier to add others in the future

The Future: CTest and Git

- Is it possible to test changes before they are committed?
  - With distributed version control it is!
- Developer pushes experimental branch
- Requests CDash server tests this branch on all supported architectures
- If everything is green, proceed to merge
- Development tree is more stable!

DVCS: New Workflows



Kitware

Only merged after testing

Cross Compiling

- New chapter in the new CMake book
- Far more important to the KDE community with embedded devices, e.g. N900+
- Significant work has already been done
- I need to read up on this!

Kitware is Hiring

- Looking for talented developers
- Opportunities for internships this summer
- Work on interesting, challenging problems
  - Exceptional skill in C++
  - Other languages such as Python, Tcl, Java
  - Work on Linux, Unix, Mac OS X and/or Windows

http://www.kitware.com/company/jobopps.html

Thank You

- Any questions?
- I will be available afterwards
- If you would like to learn more
  - Attend the tutorial session, Monday @ 2pm
- CMake is more than just a build system!

http://www.cmake.org/

mhanwell@kde.org

CMake Tutorial

- Summary of CMake
- CMake input files
- CMake GUIs
- CTest – testing your code
- CPack – packaging
- CDash – collating test results