

HPC Software Engineering

“Do as I say, not as I did”

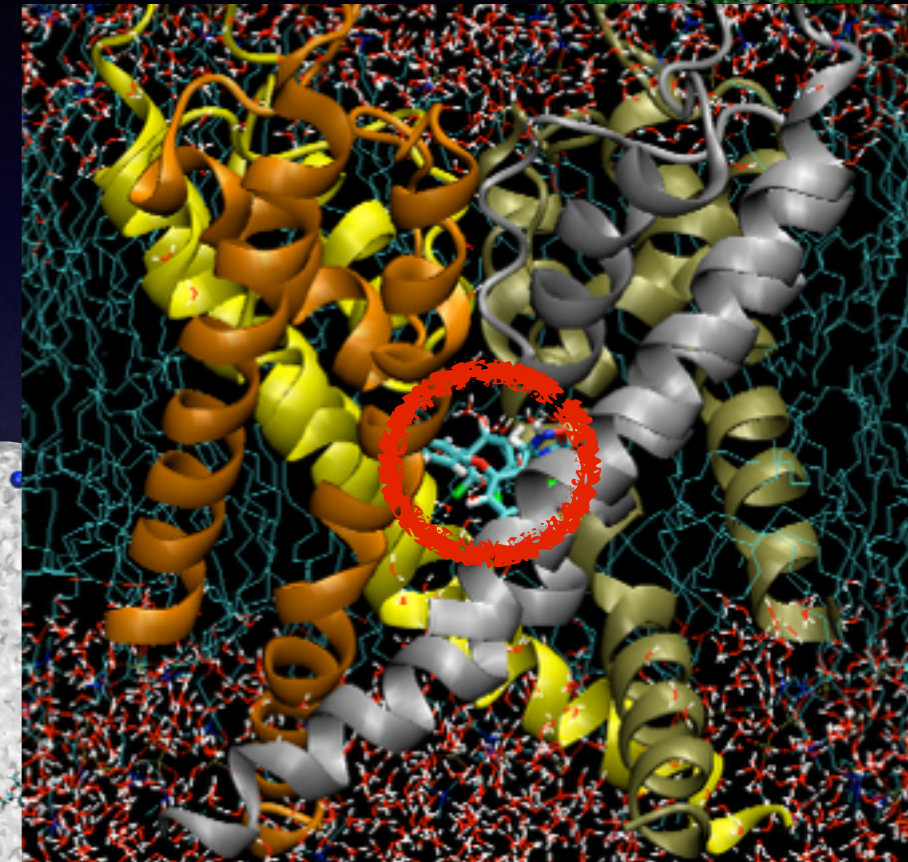
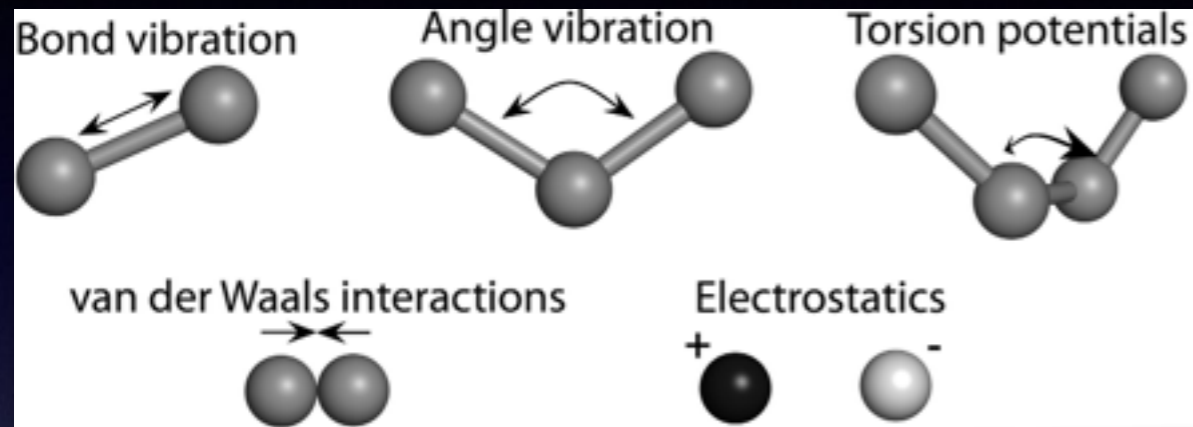
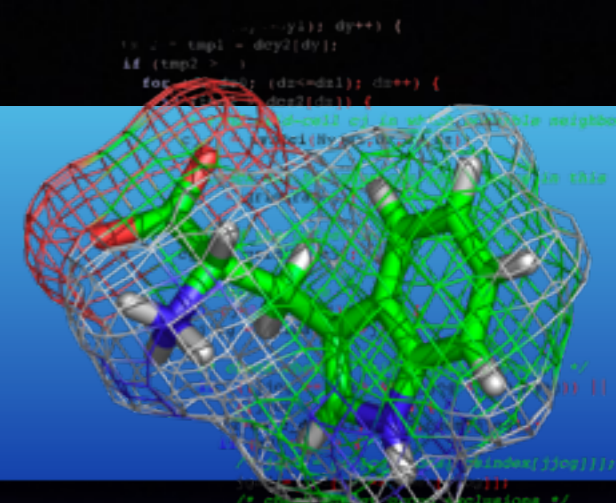
XSEDE/PRACE/RIKEN/Compute Canada HPC Summer School Budapest 2014

Erik Lindahl

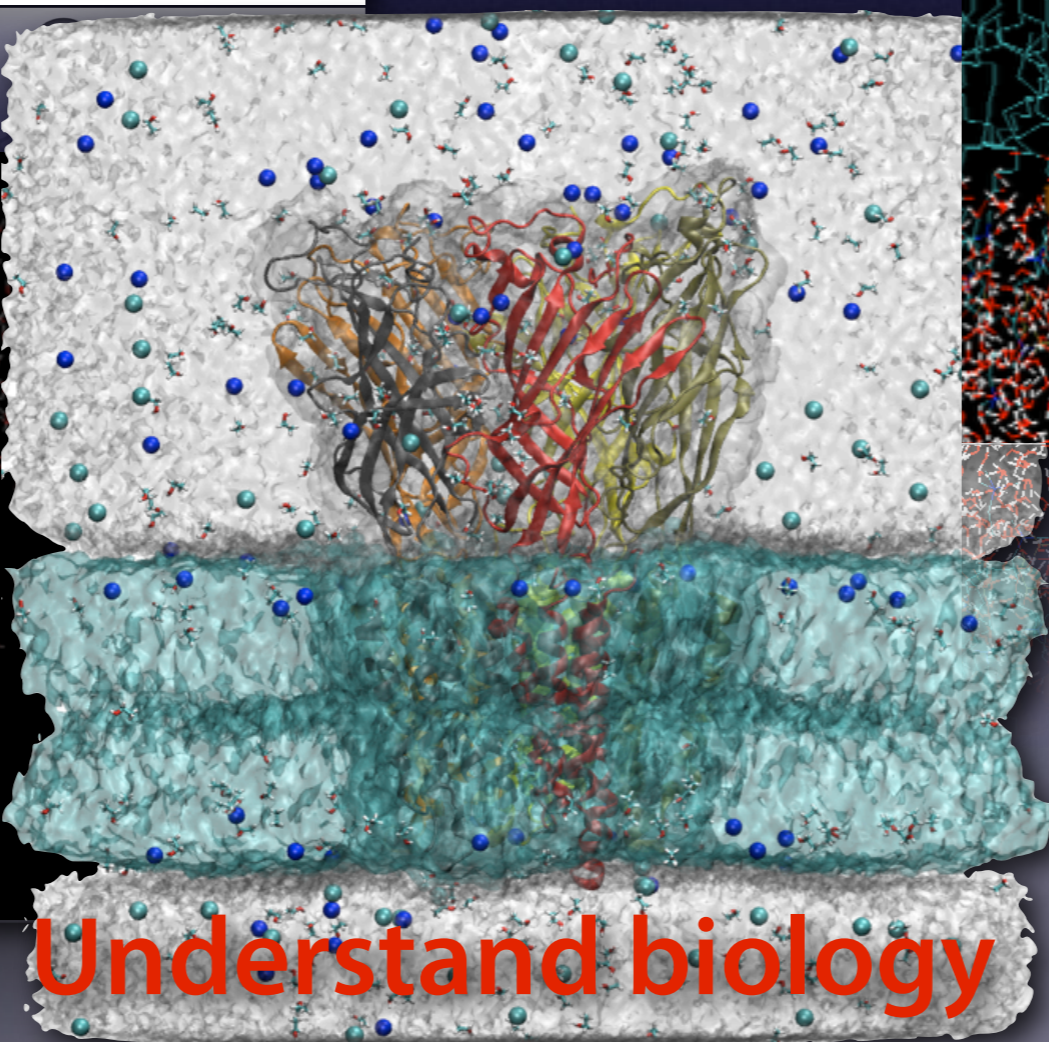
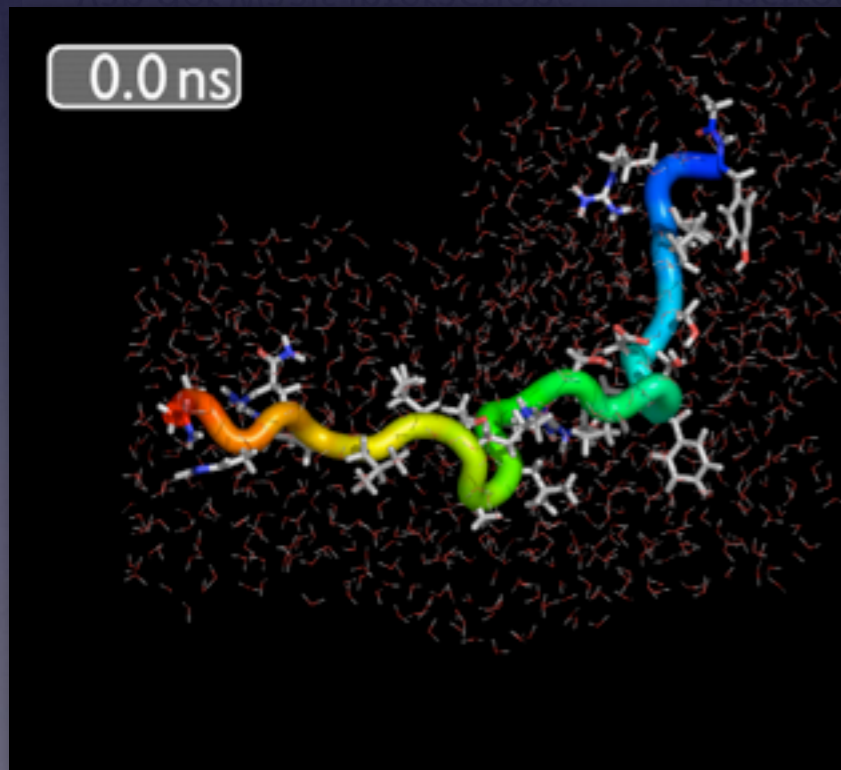
SciLifeLab
STOCKHOLM



Molecular Dynamics



0.0 ns



Free Energy & Drug Design

Protein Folding

Understand biology

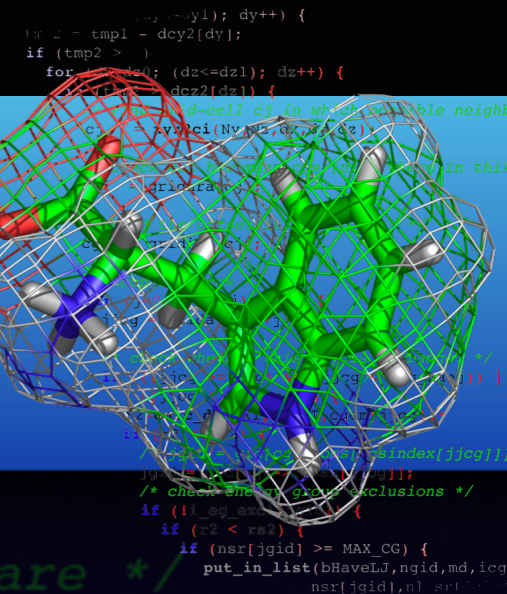
Open Source & Free Software Development Models

**How do you write high(er)
quality software?**

**Tools & Recommendations for HPC
software engineering**

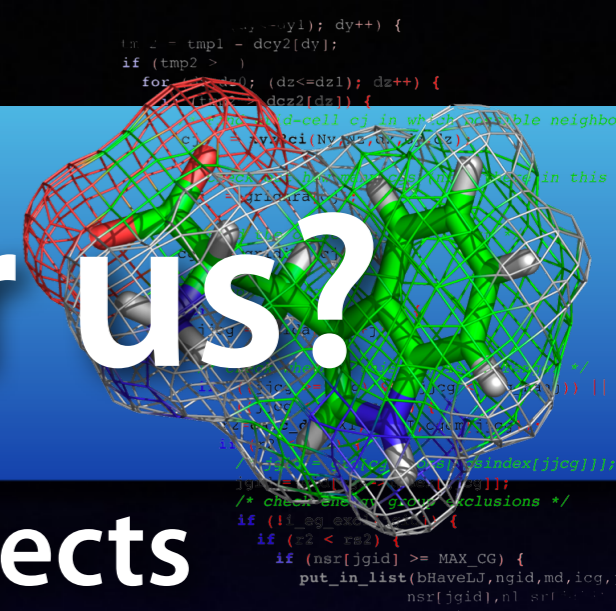
**How do you handle large projects with
many developers?**

GROMACS



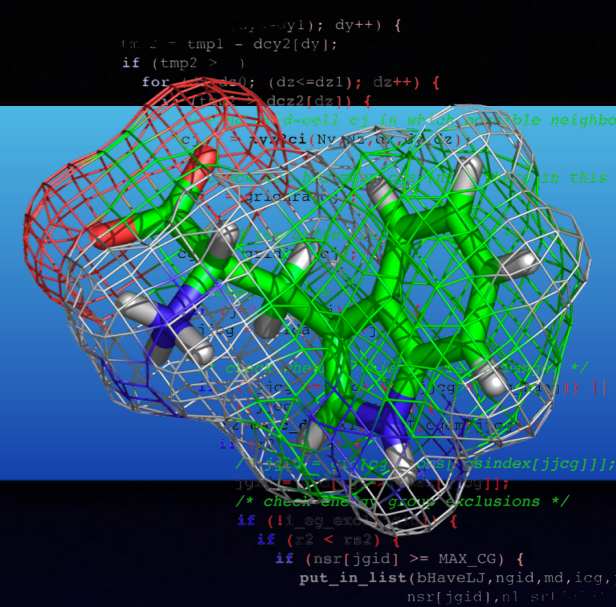
- Simulation *hardware* project in Groningen, 1995, with software spin-off
- Early development based on our own needs
- Turned GPL in 2001
- Organic growth of development
 - Roughly 10-15 core developers
 - Another 15-20 active contributors
- Currently 2,063,361 lines of C/C++/Asm code
- Lots of old code. Lots of new code. Lots of complicated code written by scientists.

What does OSS do for us?



- We can reuse code from other OSS projects
- Peer review. Bugs *will* be found and hung out. Embarrassing short-term, strength long-term.
- Easy to handle IP even with 10-15 European, US, Asian, Australian universities involved.
- We get code contributions, and lots of them.
- Fully open development: git.gromacs.org
- Bad: Easy for any ignorant student to download the code and publish crap results

The Community



April 2011 Archives by thread

- Messages sorted by: [\[subject \]](#) [\[author \]](#) [\[date \]](#)
- [More info on this list...](#)

Starting: [Fri Apr 1 00:19:43 CEST 2011](#)
Ending: [Sat Apr 30 22:55:40 CEST 2011](#)
Messages: 983

[gmx-users] OPLS-AA atom types

Justin B. Yongye ybyongye@yahoo.com
[Tue Apr 5 14:17:56 CEST 2011](#)

- [Previous message: \[gmx-users\] g_sasa](#)
- [Next message: \[gmx-users\] OPLS-AA atom types](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

- [\[gmx-users\] Taking difference of...](#)
- [\[gmx-users\] VDW parameters: M...](#)
 - [\[gmx-users\] VDW param...](#)
- [\[gmx-users\] RE: VDW paramet...](#)
 - [\[gmx-users\] RE: VDW pa...](#)
- [\[gmx-users\] VDW parameters: M...](#)
- [\[gmx-users\] Installtion of groma...](#)
 - [\[gmx-users\] Installtion of](#)
 - [\[gmx-users\] Installtion of](#)
- [\[gmx-users\] dipole moment of a...](#)
 - [\[gmx-users\] dipole mome...](#)
 - [\[gmx-users\] dipole...](#)
 - [\[gmx-users\] c...](#)

Hello GROMACS USERS,
Does any one have a d

I would like to build

```
TYPE NAME
135 CT
136 CT
137 CT
157 CT
```

I can deduce that 135

Thanks,
Austin-

(mirrored text from the question)

```
123 CG
133 CG
139 CG
132 CG
```

[gmx-users] OPLS-AA atom types

Justin A. Lemkul jalemkul@vt.edu
[Tue Apr 5 14:19:41 CEST 2011](#)

- [Previous message: \[gmx-users\] OPLS-AA atom types](#)
- [Next message: \[gmx-users\] OPLS-AA atom types](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

Austin B. Yongye wrote:

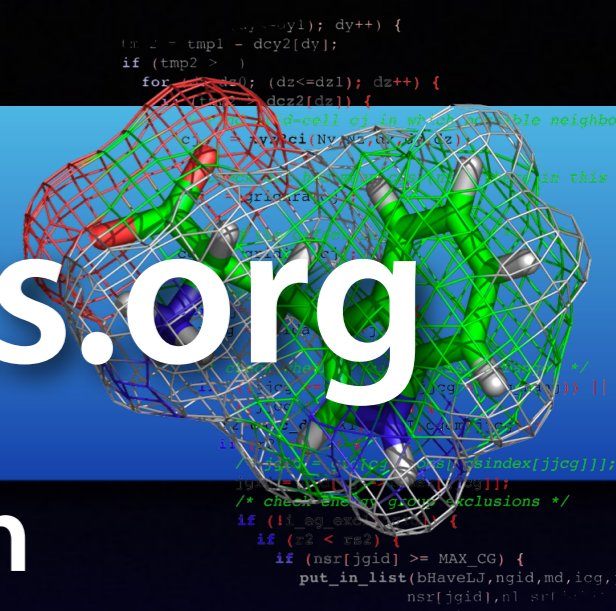
```
> Hello GROMACS USERS,
> Does any one have a document/manual/link that describes the name-number atom type mappings?
```

The correspondence between the atom types and bonded types is all contained in `ffnonbonded.itp`.

-Justin

Answer provided - in 105 seconds!

Tutorials: www.gromacs.org



400-page manual & online documentation Tutorials and How-Tos provided by users:

Adding a Residue to a Force Field

Analysing Trajectory Information

Beginners

Carbon Nanotube

Changing a 3 Point Water Model to a 4 Point Water Model

Checkpointing Jobs

Constant pH Simulation

Diffusion Constant

Dihedral PCA

Dihedral Restraints

Distance Restraints

Doing Restarts

Electrostatics

Essential Dynamics

Extending Simulations

Free Energy Calculations

Plotting Data

Build a Linux Cluster

Making Commands Non-Interactive

Making Disulfide Bonds

Membrane Simulations

Micelle Clustering

Mixed Solvents Multiple Chains

Multiple Topology Entries

Non-Water Solvation

Normal Mode Analysis

Parameterization of novel molecules

pKa calculations

Polymers

Position Restraints

Potential of Mean Force

QMMM

Reading XTC From Fortran

Reducing Trajectory Storage Volume

REMD

Removing fastest degrees of freedom

Speeding Up Simulations

Steps to Perform a Simulation

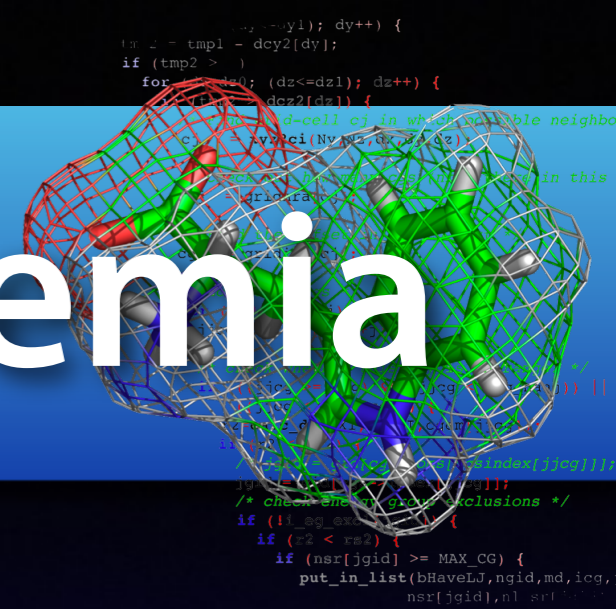
Tabulated Potentials

Water Solvation

Trajectory Visualization

Van Der Waals

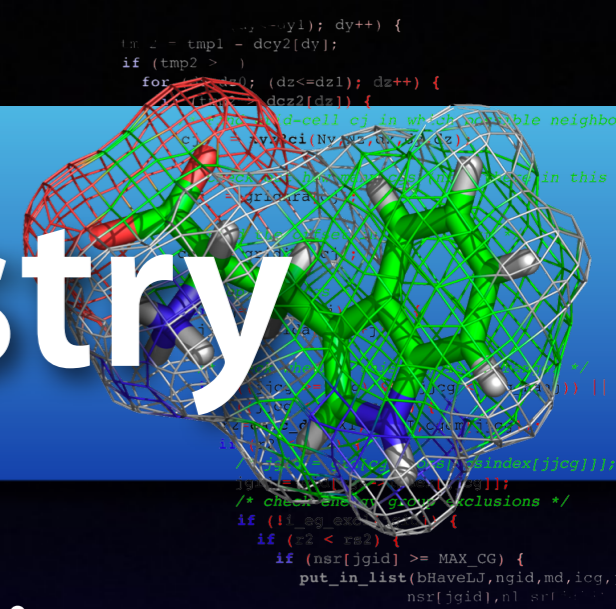
Contributions - Academia



- QM/MM interface (Groenhof, Göttingen)
- Advanced integrators (Shirts, Virginia)
- Coarse-graining (Junghans, Mainz)
- Coarse-graining (Marrink, Groningen)
- Steered simulation (Grubmüller, Göttingen)
- Free energy code (Shirts, Mobley, etc.)

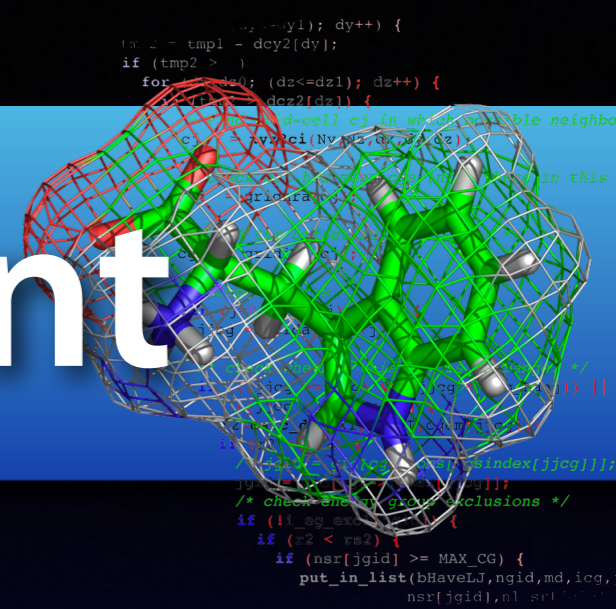
Just to mention a few...

Contributions - Industry



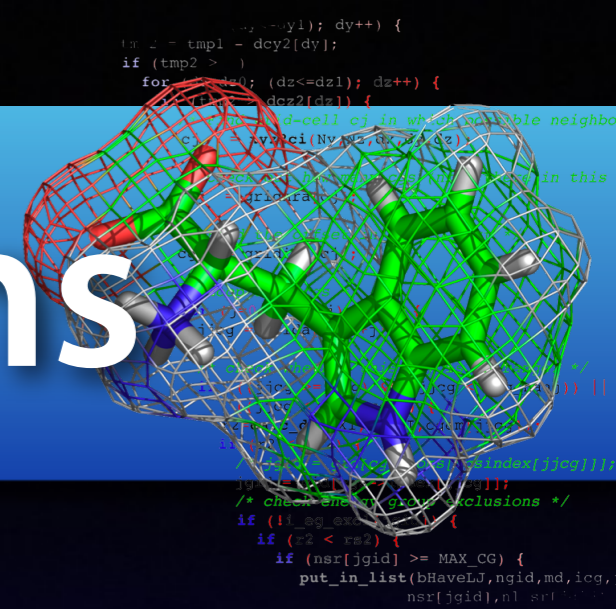
- Hewlett-Packard: IA64 asm acceleration
- IBM: BlueGene asm acceleration
- Cray: Vectorization, 3D-torus optimization
- Intel: New kernels that work better with AVX
- Intel: Larrabee->MIC->Xeon Phi work
- DE Shaw: Improved constraints, assistance with new domain decomposition algorithms

Funding Development



- Sneak it into other projects
- Add a bit of method development in all application-focused projects
- Funding as research infrastructure
 - National, EU, International
- Fund collaborative parts of the project
- Easier to fund algorithm work, hard to fund better implementations/interfaces
- Virtually impossible to fund support/GUI

License Considerations



GPLV2

LGPLV2.1

GPLV3

BSD

Dual license?

Exceptions/encryption?

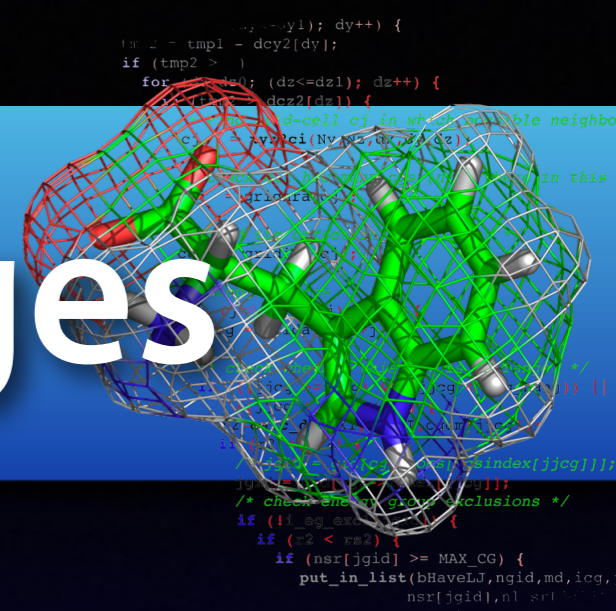
Academia-friendly?

Business-friendly?

EU-friendly?

Licenses are tools - decide what you want to achieve, and pick one that helps you!

Development Challenges



- Constant balance between things we are funded for, things fun to write, and useful things
- Students & postdocs need to publish papers
- Regression tests - make sure you can trust us
 - Large QA efforts required
- Discussion & conflicts are open on mailing lists
- Release schedules with 10+ groups involved

Political Indicents **Mighty**
"Govt might is a curse on the people"
"Antieros Causa Sedita Is itulalteiron"
...from an OSS Perspective

HPC Software Engineering

The mistakes we've made:

Overwriting new code with old by mistake

Constantly forgetting to check for compiler bugs

Software releases delayed for months

Unix contributions breaking windows

Lacking documentation

Forgetting to fix bugs

Results changing btw releases

Fights on the mailing list

Religious code convictions



The Picture until early 2011

Source code repository:

CVS

Build Chain:

Automake/Autoconf/libtool

Bug Tracking:

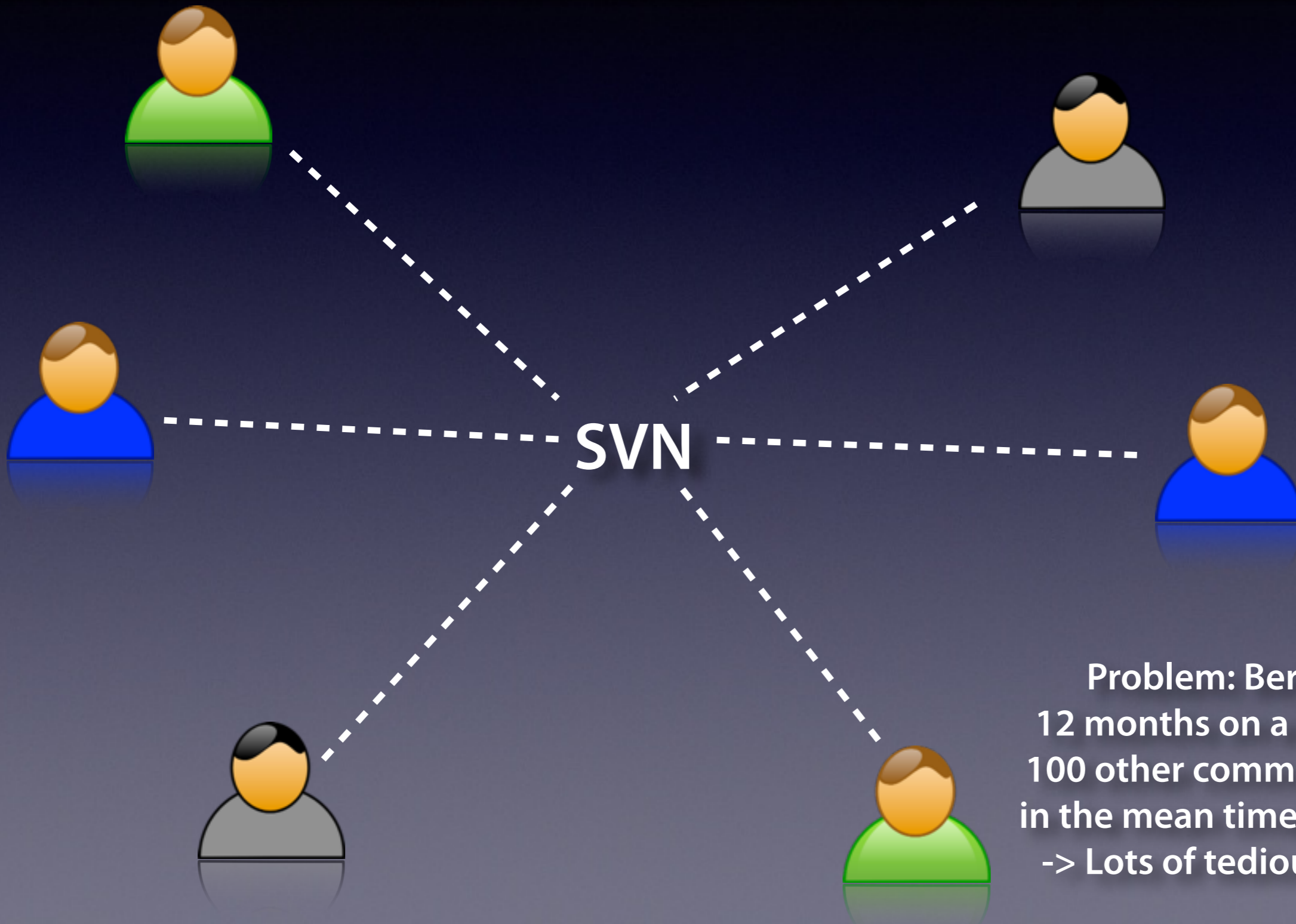
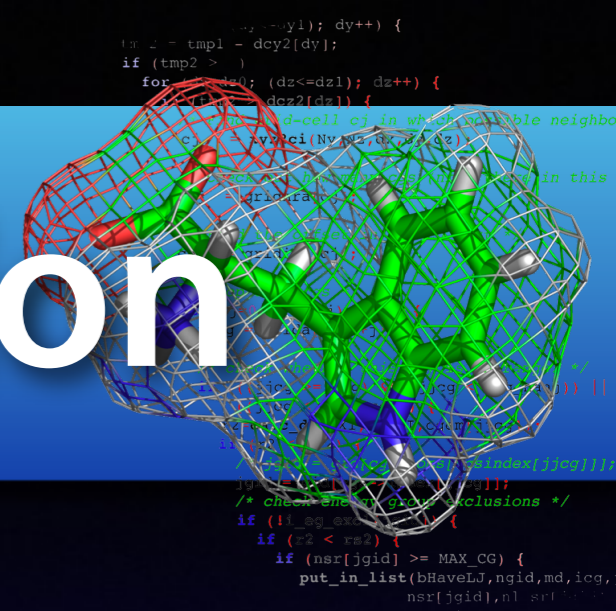
Bugzilla

Testing:



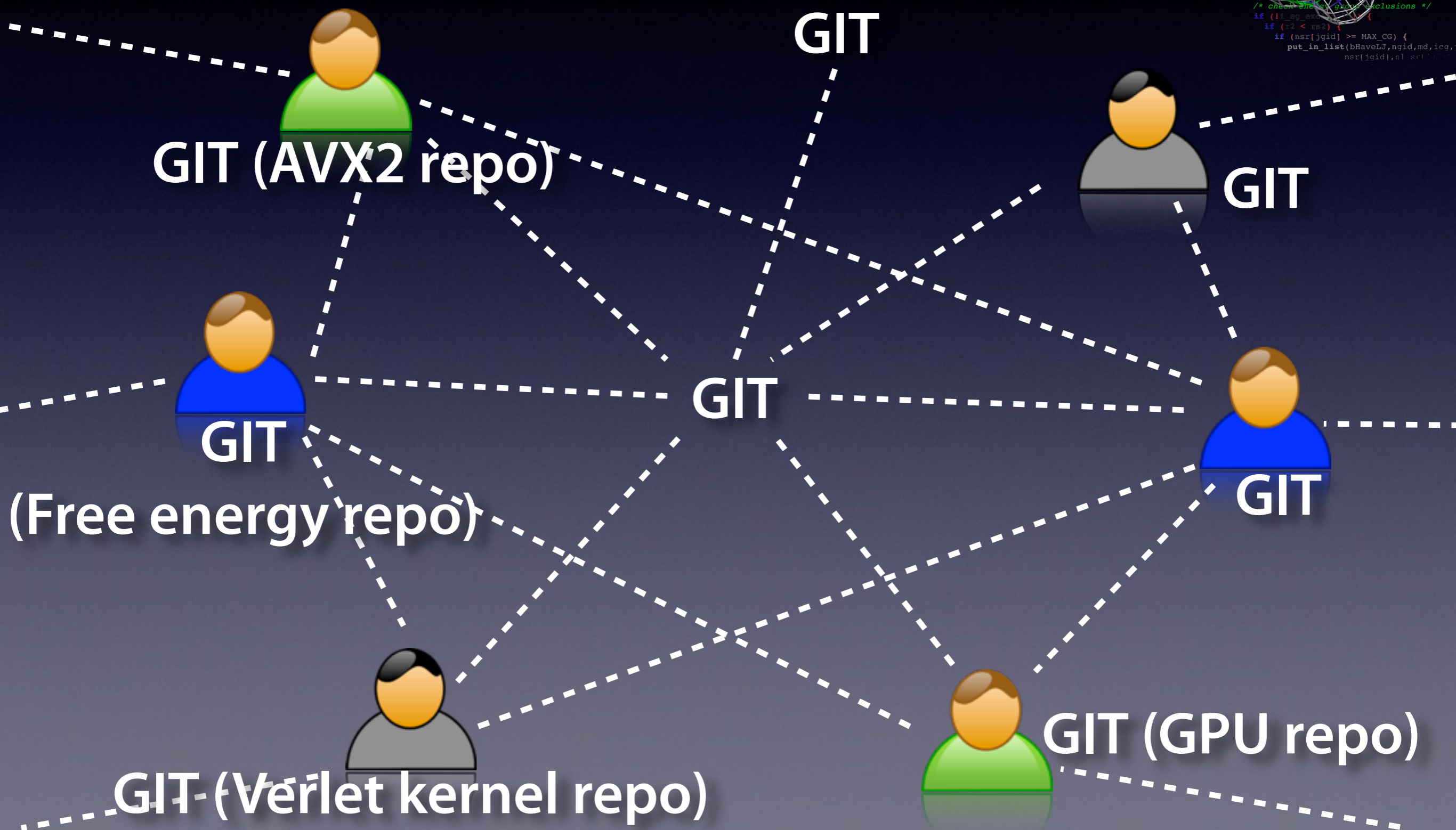
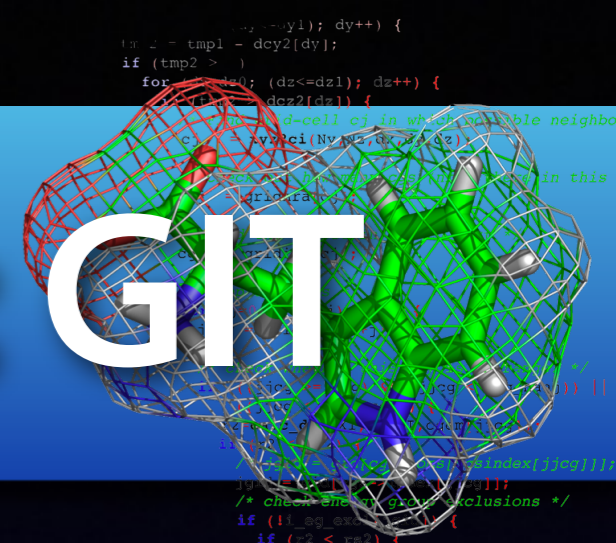
Track Your Source Code Edits

The CVS/SVN limitation



Problem: Berk has worked 12 months on a GPU branch, but 100 other commits has happened in the mean time. How to commit?
-> Lots of tedious manual work!

Better source control: GIT



*Start your free repo on
github.com if you don't
want to administrate
your own Git server!*



Local branches

Several repositories, but public & private

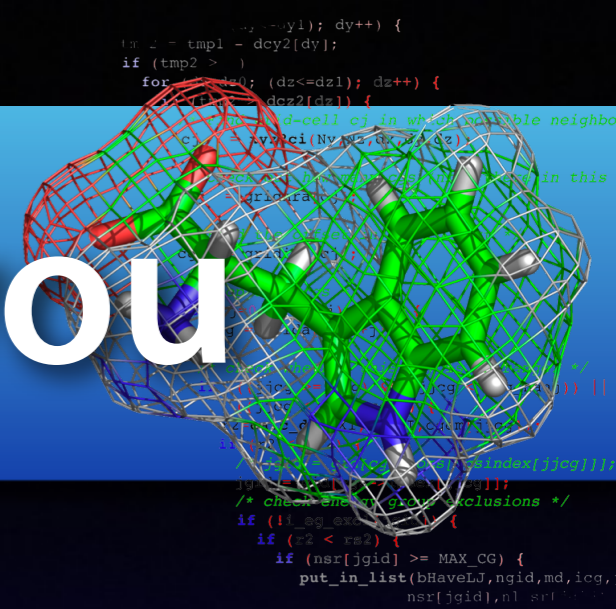
Easy to have separate branches that exchange patches

No real "master" repository

Enable both push and pull patches

<http://git-scm.com>

What git will give you



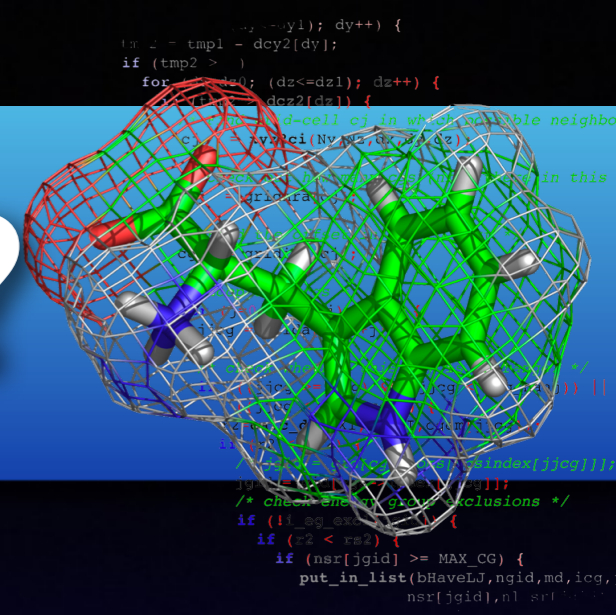
- Handle multiple developers
- Handle multiple feature branches in parallel with a stable production-quality one
- Develop based on features, not source files
- Pull/push patches between branches
- Revert a specific stupid thing I did 6 months ago, without changing subsequent patches
- Bisect changes to find which of (say) 1,500 patches caused a bug

Drawback: Git is a VERY powerful tool, but the advanced features can be difficult to understand

How do you build your code?

Does your code compile on
windows (MSVC)?
K computer (Fujitsu compilers)?
ARM? ARM64?
PowerPC (big endian)?
OpenPower (little endian?)

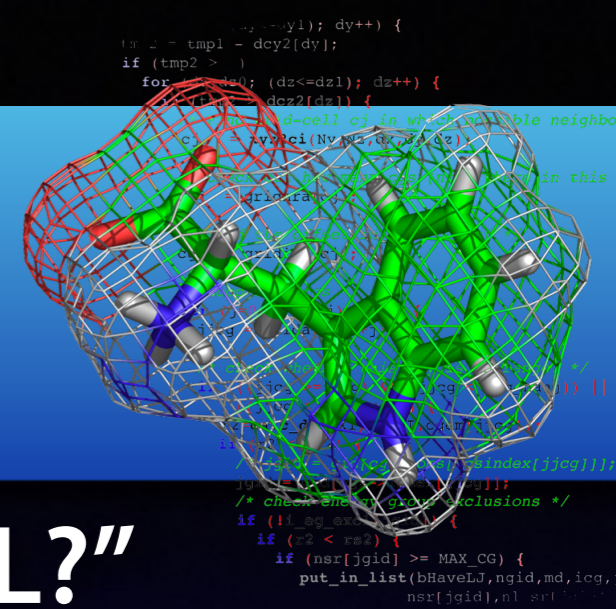
What is a build chain?



The typical user progression:

- Issue compiler commands manually
- Start using Makefiles, edit Makefiles, give up
- Automate the generation of makefiles

Configuration



- “Where is the X11 library? MKL? LibXML?”
- “What version is the FFTW library?”
- “Is the Intel Math Kernel Library installed?”
- “Do we use that buggy gcc version?”
- “Does this compiler understand AVX2 assembly?”
- “Which flags should be used for this compiler?”
- “Is this a big or small endian system?”
- “Is a long integer 4 or 8 bytes on this host?”
- “How do we build a shared library here?”
- What C compiler is used with CUDA?

CMake: Cross-platform replacement for Autoconf, Automake, Libtool

(instead of ./configure; make; make install)



The image shows a screenshot of a web browser displaying the CMake website. The browser's address bar shows the URL `http://www.cmake.org/`. The website header features the CMake logo (a 3D pyramid) and the text "CMake". Navigation links for "PROJECT", "RESOURCES", "HELP", and "OPEN SOURCE" are visible. A search bar with the text "Kitware" and a search button is also present. The main content area includes a welcome message: "Welcome to **CMake**, the cross-platform, open-source build system. CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice." Below this is a "News" section with a "More News >" link. The news items listed are: "04.19.2012 CMake 2.8.8 is Now Available", "03.02.2012 CDash 2.0.2 Now Available", "01.12.2012 Kitware Collaborates with NREL on Software Process for Radiance P...", "01.02.2012 CMake 2.8.7 Now Available", and "10.31.2011 Kitware Courses Move to Webinar Format". On the right side, there is a large banner for "CMake 2.8.8 Available" with the text "See what's new with the CMake 2.8.8 release" and a "Download Now >" button. The banner features the CMake logo. At the bottom of the page, there are logos for several partner organizations: a head with a brain and a caduceus, "ASC", "NATIONAL LIBRARY OF MEDICINE", "itk", and "Sandia National Laboratories".

~100 CMake tests for features/bugs/libraries/compilers

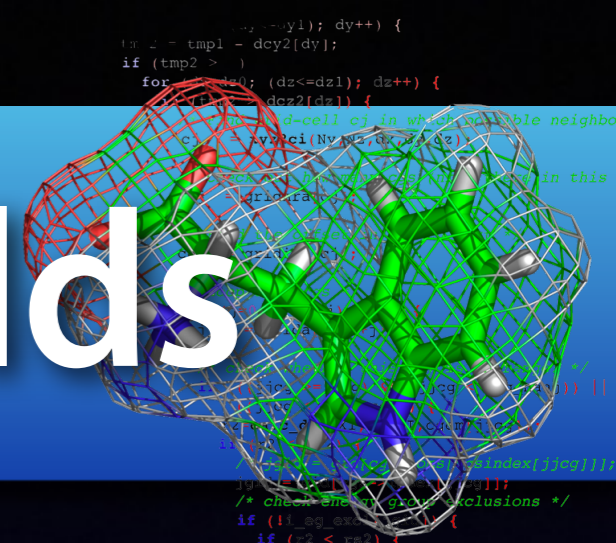
- CheckCCompilerFlag.cmake
- CheckCXXCompilerFlag.cmake
- cmake_uninstall.cmake.in
- FindEXTRA.cmake
- FindFFTW.cmake
- FindVMD.cmake
- gmxBuildTypeProfile.cmake
- gmxBuildTypeReference.cmake
- gmxBuildTypeReleaseWithAssert.cmake
- gmxBuildTypeThreadSanitizer.cmake
- gmxCFlags.cmake
- gmxDetectClang30.cmake
- gmxDetectGpu.cmake
- gmxDetectSimd.cmake
- gmxDetectTargetArchitecture.cmake
- gmxFindFlagsForSource.cmake
- gmxGCC4403BugWorkaround.cmake
- gmxGenerateVersionInfo.cmake
- gmxManageBlueGene.cmake
- gmxManageFFTLibraries.cmake
- gmxManageGPU.cmake
- gmxManageLinearAlgebraLibraries.cmake
- gmxManageMPI.cmake
- gmxManageNvccConfig.cmake
- gmxManageOpenMP.cmake
- gmxManageSharedLibraries.cmake
- gmxManageSuffixes.cmake
- gmxOptionUtilities.cmake
- gmxSetBuildInformation.cmake
- gmxTestAVXMaskload.cmake
- gmxTestCatamount.cmake
- gmxTestCompilerProblems.cmake
- gmxTestCXX11.cmake
- gmxTestdlopen.cmake
- gmxTestFloatFormat.cmake
- gmxTestInlineASM.cmake
- gmxTestIsfinite.cmake
- gmxTestLargeFiles.cmake

```
MACRO(GMX_TEST_AVX_GCC_MASKLOAD_BUG VARIABLE AVX_CFLAGS)
  IF(NOT DEFINED ${VARIABLE})
    MESSAGE(STATUS "Checking for gcc AVX maskload bug")
    # some compilers like clang accept both cases,
    # so first try a normal compile to avoid flagging those as buggy.
    TRY_COMPILE(${VARIABLE}_COMPILEOK "${CMAKE_BINARY_DIR}"
      "${CMAKE_SOURCE_DIR}/cmake/TestAVXMaskload.c"
      COMPILE_DEFINITIONS "${AVX_CFLAGS}" )
    IF(${VARIABLE}_COMPILEOK)
      SET(${VARIABLE} 0 CACHE INTERNAL "Work around GCC bug in AVX maskload argument" FORCE)
      MESSAGE(STATUS "Checking for gcc AVX maskload bug - not present")
    ELSE()
      TRY_COMPILE(${VARIABLE}_COMPILEOK "${CMAKE_BINARY_DIR}"
        "${CMAKE_SOURCE_DIR}/cmake/TestAVXMaskload.c"
        COMPILE_DEFINITIONS "${AVX_CFLAGS} -DGMX_SIMD_X86_AVX_GCC_MASKLOAD_BUG" )
      IF(${VARIABLE}_COMPILEOK)
        SET(${VARIABLE} 1 CACHE INTERNAL "Work around GCC bug in AVX maskload argument" FORCE)
        MESSAGE(STATUS "Checking for gcc AVX maskload bug - found, will try to work around")
      ELSE()
        MESSAGE(WARNING "Cannot compile AVX code - assuming gcc AVX maskload bug not present.")
        MESSAGE(STATUS "Checking for gcc AVX maskload bug - not present")
      ENDIF()
    ENDIF()
  ENDIF()
ENDMACRO()
```

Optional components (FFT libs) and extensive regression tests can be downloaded automatically

Generators: Makefiles, Eclipse, Xcode, VisualStudio, nmake, CodeBlocks, KDevelop3, etc.

Out-of-source builds



Don't put the build objects inside the source code directory!

/home/lindahl/code/Gromacs-5.0

source code

Mac mixed precision build

Mac mixed precision installation

Linux SSE4.1 mixed build

Linux SSE4.1 mixed install

Linux AVX2 double build

Linux AVX2 double install

Make a small change, run "make" in three build directories, done.

Living with your code for years: Documentation

“Documentation is like sex. When it’s good, it’s great. When it’s bad, it’s better than nothing”

[Linus Torvalds]

If documentation is not in the source, it won't be updated



The image shows a screenshot of a web browser displaying the Doxygen website. The browser's address bar shows the URL www.stack.nl/~dimitri/doxygen/. The website has a blue header with the Doxygen logo and a navigation menu with links for Home, Downloads, Documentation, Extensions, and Support. A sidebar on the left contains a list of links including About, Downloads, Changelog, Documentation, Get Involved, Wish list, Examples, Links, Extensions, Support, and Donate. The main content area features a large heading 'Doxygen' and a sub-heading 'Generate documentation from source code'. Below this, there is a paragraph explaining that Doxygen is the de facto standard tool for generating documentation from annotated C++ sources, and it also supports other languages like C, Objective-C, C#, PHP, Java, Python, IDL, Fortran, VHDL, and Tcl. A list of three ways Doxygen can help is provided, followed by a paragraph stating that Doxygen is developed under Mac OS X and Linux and is highly portable.

Doxygen: Main Page

www.stack.nl/~dimitri/doxygen/

Umeå Univer...mi i Skolan How to Do ...n — Medium Computation...gy Council Varför denna...naljapen.se

Donate

€ (EUR)

Home Downloads Documentation Extensions Support

Doxygen

About

- Downloads
- Changelog
- Documentation
- Get Involved
- Wish list
- Examples
- Links
- Extensions
- Support
- Donate

Doxygen

Generate documentation from source code

Doxygen is the de facto standard tool for generating documentation from annotated C++ sources, but it also supports other popular programming languages such as C, Objective-C, C#, PHP, Java, Python, IDL (Corba, Microsoft, and UNO/OpenOffice flavors), Fortran, VHDL, Tcl, and to some extent D.

Doxygen can help you in three ways:

1. It can generate an on-line documentation browser (in HTML) and/or an off-line reference manual (in \LaTeX) from a set of documented source files. There is also support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages. The documentation is extracted directly from the sources, which makes it much easier to keep the documentation consistent with the source code.
2. You can [configure](#) doxygen to extract the code structure from undocumented source files. This is very useful to quickly find your way in large source distributions. Doxygen can also visualize the relations between the various elements by means of include dependency graphs, inheritance diagrams, and collaboration diagrams, which are all generated automatically.
3. You can also use doxygen for creating normal documentation (as I did for the doxygen user manual and web-site).

Doxygen is developed under Mac OS X and Linux, but is set-up to be highly portable. As a result, it runs on most other Unix flavors as well. Furthermore, executables for Windows are available.

Doxygen example - our SIMD module: [gromacs/src/gromacs/simd/]

```
49 #ifndef GMX_SIMD_SIMD_H
50 #define GMX_SIMD_SIMD_H
51
52 /*! \libinternal \file
53 *
54 * \brief Definitions, capabilities, and wrappers for SIMD module.
55 *
56 * The macros in this file are intended to be used for writing
57 * architecture-independent SIMD intrinsics code.
58 * To support a new architecture, adding a new sub-include with macros here
59 * should be (nearly) all that is needed.
60 *
61 * The defines in this top-level file will set default Gromacs real precision
62 * operations to either single or double precision based on whether
63 * GMX_DOUBLE is defined. The actual implementation - including e.g.
64 * conversion operations specifically between single and double
65 * in impl_reference.h.
66 *
67 * \author Erik Lindahl <erik.lindahl@scilifelab.se>
68 *
69 * \inlibraryapi
70 * \ingroup module_simd
71 */
72
73 #ifdef HAVE_CONFIG_H
74 #include <config.h>
75 #endif
76
77 #include <stddef.h>
78
```

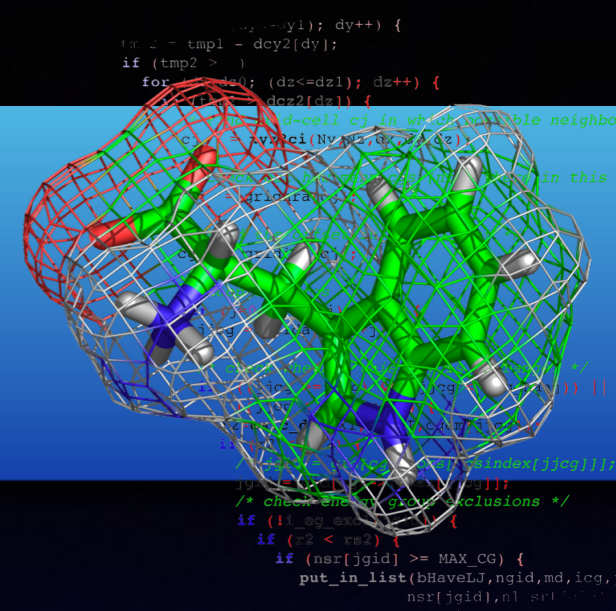
```
155 /*! \brief
156 * Align a float pointer for usage with SIMD instructions.
157 *
158 * You should typically \a not call this function directly (unless you explicitly
159 * want single precision even when GMX_DOUBLE is set), but use the
160 * \ref gmx_simd_align_r macro to align memory in default Gromacs real precision.
161 *
162 * \param p Pointer to memory, allocate at least \ref GMX_SIMD_FLOAT_WIDTH extra elements.
163 *
164 * \return Aligned pointer (>=p) suitable for loading/storing float fp SIMD.
165 *         If \ref GMX_SIMD_HAVE_FLOAT is not set, p will be returned unchanged.
166 *
167 * Start by allocating an extra \ref GMX_SIMD_FLOAT_WIDTH float elements of memory,
168 * and then call this function. The returned pointer will be greater or equal
169 * to the one you provided, and point to an address inside your provided memory
170 * that is aligned to the SIMD width.
171 */
172 static gmx_inline float *
173 gmx_simd_align_f(float *p)
174 {
175     #ifdef GMX_SIMD_HAVE_FLOAT
176         return (float *)(((size_t)((p)+GMX_SIMD_FLOAT_WIDTH-1)) & ~(size_t)
177             (GMX_SIMD_FLOAT_WIDTH*sizeof(float)-1));
178     #else
179         return p;
180     #endif
181 }
```

The best comments don't explain what your code does, they explain WHY you do it this way!

< Demo of doxygen documentation >

Finding & Preventing Bugs

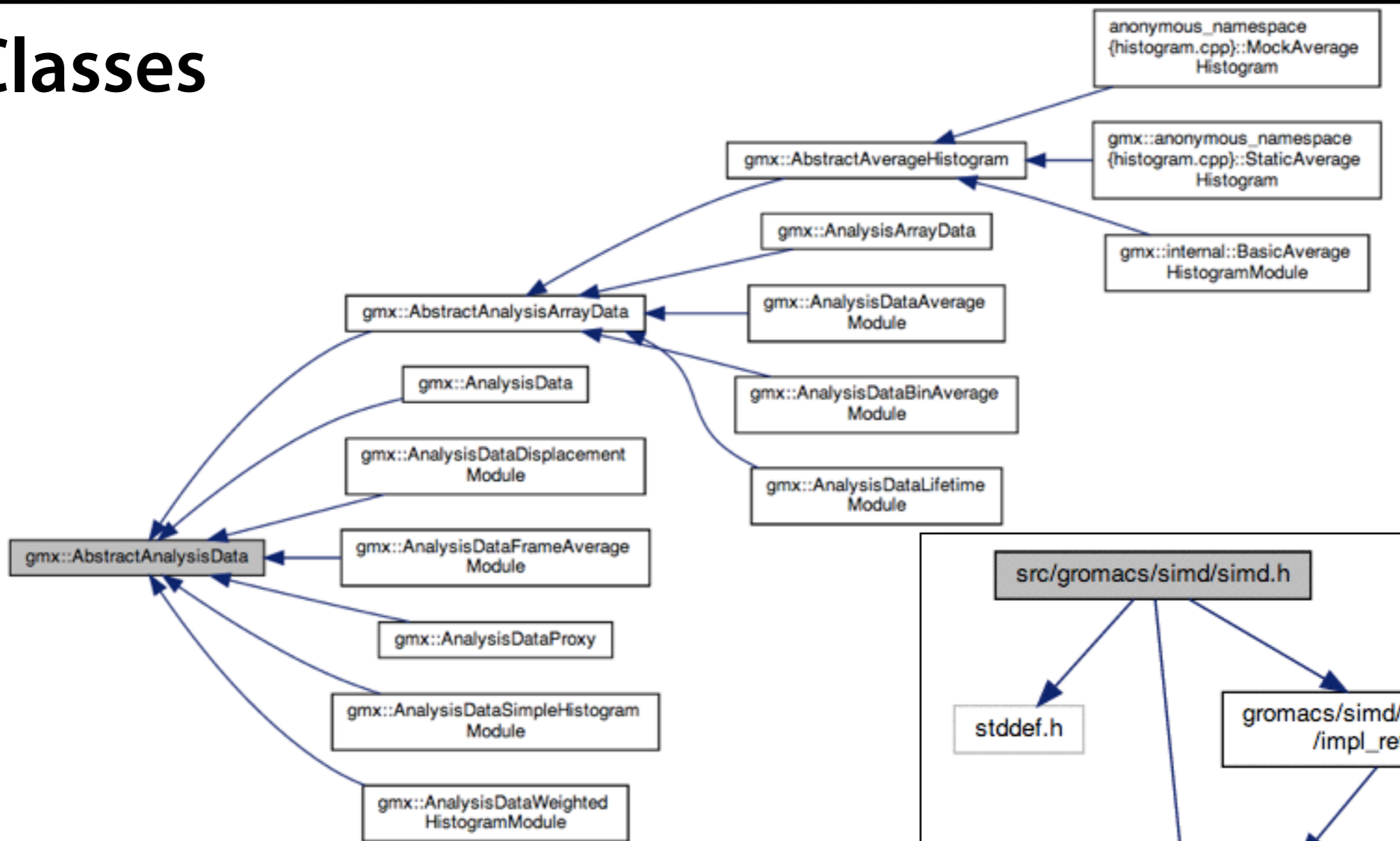
Modularization



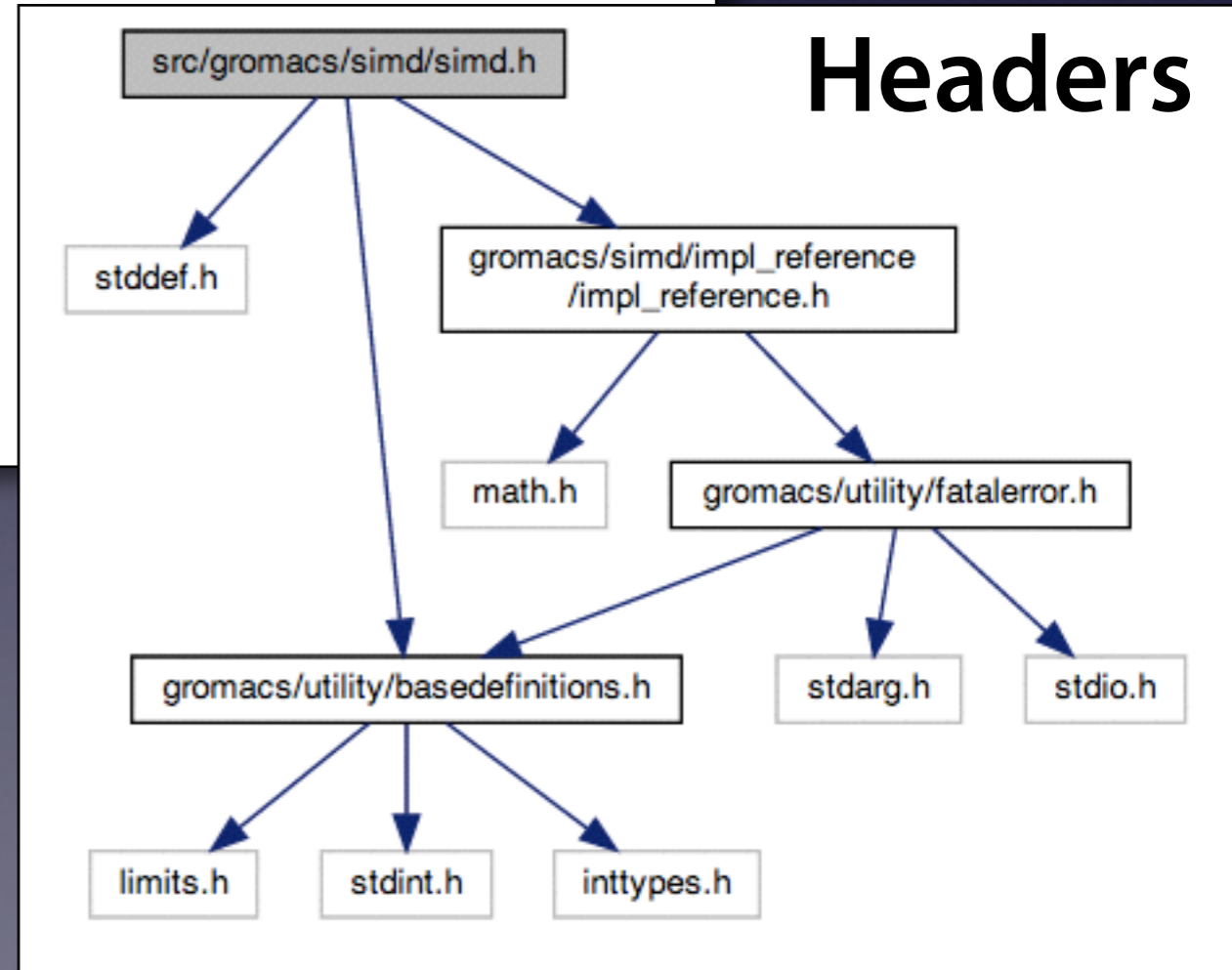
- Avoid code inter-dependencies
- Have modules doing clearly separate tasks
- Make sure all code is thread-safe!
- Have a clear (documented) API for each module
- Write unit tests, not only regression tests
- Write unit test first, then the code implementation

Modularization: Say 'no' to circular dependencies

Classes



Headers



This is hard, but Doxygen helps you detect it!



Aggressive unit testing: "Trust, but verify"

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#)

[Summary](#) [People](#)

Project Information

★ Starred by 2339 users

[Project feeds](#)

Code license

[New BSD License](#)

Labels

Cplusplus, Testing, Framework, Tests, Unittests, Cpp, Google

Members

[j...@google.com](#),
[zhanyong...@gmail.com](#),
[w...@google.com](#),
[ko...@google.com](#),
[sbe...@google.com](#),
[billydon...@google.com](#)
[8 committers](#)

Featured

Downloads

[gtest-1.7.0.zip](#)

[Show all »](#)

Wiki pages

[DevGuide](#)

[Documentation](#)

[V1.6 Documentation](#)

[V1.7 Documentation](#)

[Show all »](#)

Links

Blogs

[Musings on C++ Testing Tools](#)

[g+1](#) [1k](#)

Google's framework for writing C++ tests on a variety of platforms (Linux, Mac OS X, Windows, Cygwin, Windows CE, and Symbian). Based on the xUnit architecture. Supports automatic test discovery, a rich set of assertions, user-defined assertions, death tests, fatal and non-fatal failures, value- and type-parameterized tests, various options for running the tests, and XML test report generation.

Getting Started

After downloading Google Test, unpack it, read the README file and the documentation wiki pages (listed on the right side of this front page).

Who Is Using Google Test?

In addition to many internal projects at Google, Google Test is also used by the following notable projects:

- The [Chromium projects](#) (behind the Chrome browser and Chrome OS)
- The [LLVM](#) compiler
- [Protocol Buffers](#) (Google's data interchange format)

If you know of a project that's using Google Test and want it to be listed here, please let googletestframework@googlegroups.com know.

Google Test-related open source projects

[Google Test UI](#) is test runner that runs your test binary, allows you to track its progress via a progress bar, and displays a list of test failures. Clicking on one shows failure text. Google Test UI is written in C#.

[GTest TAP Listener](#) is an event listener for Google Test that implements the [TAP protocol](#) for test result output. If your test runner understands TAP, you may find it useful.

Example Gromacs unit tests: The idea is that you should test *everything*

```
185 TEST_P(FFTTest1D, Real)
186 {
187     const int rx = GetParam();
188     const int cx = (rx/2+1);
189     ASSERT_LE(cx*2, static_cast<int>(sizeof(inputdata)/sizeof(real)));
190
191     in_ = std::vector<real>(inputdata, inputdata+cx*2);
192     out_ = std::vector<real>(cx*2);
193     real* in = &in_[0];
194     real* out = &out_[0];
195
196     gmx_fft_init_1d_real(&fft_, rx, flags_);
197
198     gmx_fft_1d_real(fft_, GMX_FFT_REAL_TO_COMPLEX, in, out);
199     checker_.checkSequenceArray(cx*2, out, "forward");
200     gmx_fft_1d_real(fft_, GMX_FFT_COMPLEX_TO_REAL, in, out);
201     checker_.checkSequenceArray(cx*2, out, "backward");
202 }

204 TEST_F(SimdFloatingpointTest, gmxSimdGetMantissaR)
205 {
206     GMX_EXPECT_SIMD_REAL_EQ(setSimdRealFrom3R(1.219097320577810839026256,
207                                               1.166738027848349235071623,
208                                               1.168904015004464724825084), gmx_simd_get_mantissa_r(rSimd_Exp));
209     #if (defined GMX_SIMD_HAVE_DOUBLE) && (defined GMX_DOUBLE)
210     GMX_EXPECT_SIMD_REAL_EQ(setSimdRealFrom3R(1.241261238952345623563251,
211                                               1.047294723759123852359232,
212                                               1.856066204750275957395734), gmx_simd_get_mantissa_r(rSimd_ExpDouble));
213     #endif
214 }

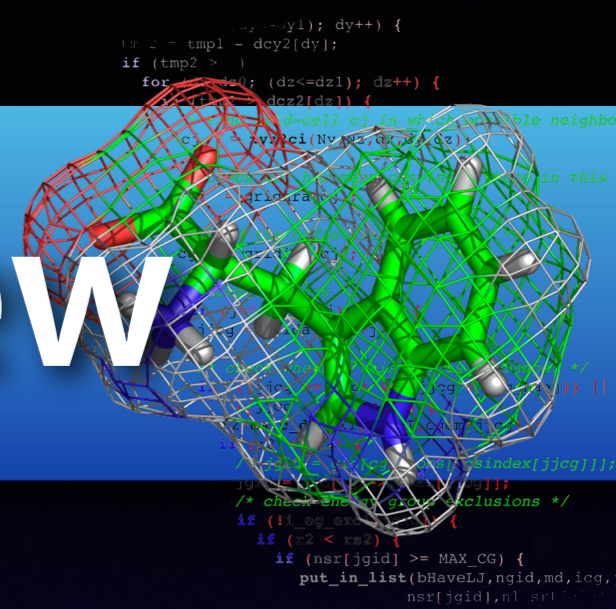
216 TEST_F(SimdFloatingpointTest, gmxSimdSetExponentR)
217 {
218     gmx_simd_real_t x0 = setSimdRealFrom3R(0.5, 11.5, 99.5);
219     gmx_simd_real_t x1 = setSimdRealFrom3R(-0.5, -11.5, -99.5);
220
221     GMX_EXPECT_SIMD_REAL_EQ(setSimdRealFrom3R(pow(2.0, 60.0), pow(2.0, -41.0), pow(2.0, 54.0)),
222                             gmx_simd_set_exponent_r(setSimdRealFrom3R(60.0, -41.0, 54.0)));
223     #if (defined GMX_SIMD_HAVE_DOUBLE) && (defined GMX_DOUBLE)
224     GMX_EXPECT_SIMD_REAL_EQ(setSimdRealFrom3R(pow(2.0, 587.0), pow(2.0, -462.0), pow(2.0, 672.0)),
225                             gmx_simd_set_exponent_r(setSimdRealFrom3R(587.0, -462.0, 672.0)));
226     #endif
227     /* Rounding mode in gmx_simd_set_exponent_r() must be consistent with gmx_simd_round_r() */
228     GMX_EXPECT_SIMD_REAL_EQ(gmx_simd_set_exponent_r(gmx_simd_round_r(x0)), gmx_simd_set_exponent_r(x0));
229     GMX_EXPECT_SIMD_REAL_EQ(gmx_simd_set_exponent_r(gmx_simd_round_r(x1)), gmx_simd_set_exponent_r(x1));
230 }
231
```

Do you think it's overkill to test that hardware rounding works? In March, this very test caught that IBM BlueGene uses different rounding modes for SIMD and normal floating-point to integer conversions...

Commits - how code makes it into Gromacs

Who is allowed to write to *your* code repository?

Gerrit Code Review



gerrit - Gerrit Code Review - Google Project Hosting

http://code.google.com/p/gerrit/

erik.lindahl@gmail.com | My favorites | Profile | Sign out

gerrit
Gerrit Code Review

Project Home Downloads Issues Source

Summary People

Project Information

Web based code review and project management for Git based projects.

Objective

Gerrit is a web based code review system, facilitating online code reviews for projects using the Git version control system.

Gerrit makes reviews easier by showing changes in a side-by-side display, and allowing inline comments to be added by any reviewer.

Gerrit simplifies Git based project maintainership by permitting any authorized user to submit changes to the master Git repository, rather than requiring all approved changes to be merged in by hand by the project maintainer. This functionality enables a more centralized usage of Git.

News

- Jun 25, 2012 - Gerrit 2.2.2.2, 2.3.1, 2.4.2 [Released](#)
- Jun 14, 2012 - Gerrit 2.4.1 [Released](#)
- May 25, 2012 - Gerrit 2.4 final [Released](#)
- May 23, 2012 - A new page dedicated to furthering Gerrit [MultiMaster](#) support
- May 23, 2012 - A new page dedicated to tips for [Scaling](#) Gerrit installations
- May 23, 2012 - Gerrit 2.4-rc2 [Released](#)
- May 21, 2012 - Hackathon [Report](#)

Members

sop@google.com,
mf...@codeaurora.org,
ziv...@gmail.com,
spea...@spearce.org,
edwin.ke...@gmail.com
[21 contributors](#)

Featured

Downloads

[gerrit-2.4.2.war](#)
[Show all »](#)

Nobody can commit directly to central Git repo anymore!
... which means we can allow almost anybody to commit

status:open | gerrit.gromacs Code Review

https://gerrit.gromacs.org/#/q/status:open,n,z

status:open

Search Register Sign

Search for status:open

| Subject | Status | Owner | Project | Branch | Updated | CR | V |
|--|--------------------------|---------------------|-----------------|----------------------------|----------|----|---|
| ▶ New quote | | Mark Abraham | gromacs | release-5-0 | 3:30 PM | +1 | ✓ |
| Improve module dependency graph layout | | Teemu Murtola | gromacs | master (doxygen) | 2:42 PM | ✓ | ✓ |
| Module dependency cycle checker for 'doc-check' | | Teemu Murtola | gromacs | master (doxygen) | 2:41 PM | ✓ | ✓ |
| Updated reference with fixed potential-shift discorr | | Erik Lindahl | regressiontests | release-4-6 | 2:11 PM | ✓ | ✓ |
| Fixed shift and switch modifiers, particularly for free-energy | | Berk Hess | gromacs | release-4-6 | 2:10 PM | +1 | ✓ |
| Remove mdrun -seppot | | Mark Abraham | gromacs | master | 2:09 PM | +1 | ✓ |
| Move atomprop.* to topology/ | Submitted, Merge Pending | Teemu Murtola | gromacs | master (legacyheaders) | 1:48 PM | ✓ | ✓ |
| Update tests using v-rescale | | Mark Abraham | regressiontests | release-5-0 | 1:25 PM | | |
| Use RNG correctly for v-rescale thermostat | | Mark Abraham | gromacs | release-5-0 | 1:25 PM | ✓ | |
| Move some verlet headers to mdlib | | Roland Schulz | gromacs | master | 11:36 AM | | ✗ |
| RFC: Used IWYU to partially clean up includes | | Roland Schulz | gromacs | master | 10:49 AM | | ✓ |
| Check to ensure not reading past end of file. | | Magnus Lundborg | tng | master | 9:47 AM | | |
| Fixed wrong journal reference in manual | | David van der Spoel | gromacs | master | 9:44 AM | ✓ | ✓ |
| Add StringFormatter and formatAndJoin to stringutil | | Mark Abraham | gromacs | master (g-tune-pme-reform) | 5:57 AM | ✓ | ✓ |
| RFC: Make all include paths same format | | Roland Schulz | | | | | |
| Replace all command line parsing with Options | | Teemu Murtola | | | | | |
| Move mtop_util.* and topsort.* to topology/ | | Teemu Murtola | | | | | |
| Remove more uses of typedefs.h | | Teemu Murtola | | | | | |
| Enable 4-letter residue names in PDB output | | Erik Lindahl | | | | | |
| Updated C-/N-terminal partial charges in Amber03.ff. | | Rossen Apostolov | | | | | |
| Convert repl_ex.c to C++ | | Mark Abraham | | | | | |
| [RFC] Framework for analyzing energy files. | | David van der Spoel | | | | | |
| Improve FileNameOption error handling | | Teemu Murtola | | | | | |
| Fix ref error in pull | | Roland Schulz | | | | | |
| Issue a warning for using gmx_rms -prev with large trajectories. | | Rossen Apostolov | | | | | |

Multiple patches in-flight
Gerrit/git do dependency tracking, patches can be rebased onto others by hitting a rebase button

Extensive comments on code during review

Roland has approved Mark's patch. Anybody can add comments. When two trusted developers say OK, the patch is committed.

Change #, SHA-1, trid or owner:email

Change-Id: I1fb8eddb7c8b029dc3686be80f3f083108fc28c

Owner: Mark Abraham

Project: gromacs

Branch: release-5-0

Topic:

Uploaded: May 25, 2014 9:28 PM

Updated: Jun 2, 2014 1:25 PM

Submit Type: Rebase if Necessary

Status: Review in Progress

Commit Message: [Permalink](#)

Use RNG correctly for v-rescale thermostat

Two integers were passed in the wrong order. I suspect from the construction of the RNG that the only effect of this is to permit a rare re-use of a random number in a different RNG stream (i.e. no effect in practice).

Change-Id: I1fb8eddb7c8b029dc3686be80f3f083108fc28c

| Reviewer | Code-Review | Verified |
|---------------|-------------|----------|
| Mark Abraham | | |
| Roland Schulz | ✓ | |

- Need Verified

▶ Dependencies

Reference Version: Base

▶ Patch Set 1: 7aff98680e3b1d29b7a3786799606bad068768f6 (github)

▼ Patch Set 2: 9817980d60eab742f9d3e7468d210de82ac80bcb (github)

Author: Mark Abraham <mark.j.abraham@gmail.com> May 25, 2014 9:38 PM

Committer: Mark Abraham <mark.j.abraham@gmail.com> Jun 2, 2014 9:21 AM

Parent(s): ab9ac88415a51482e2e99a9e6e8a44242d365805 Add quote on the kT-kj/mol conversion factor

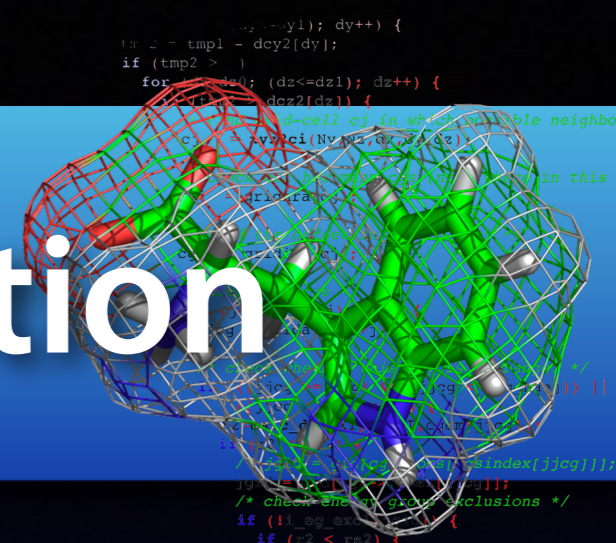
Download: checkout | pull | cherry-pick | patch | Anonymous HTTP

git fetch https://gerrit.gromacs.org/gromacs refs/changes/05/3505/2 && git checkout FETCH_HEAD

Maintaining quality & avoiding breaking stuff

How do I make sure that **I** don't make mistakes?

Jenkins Continuous Integration

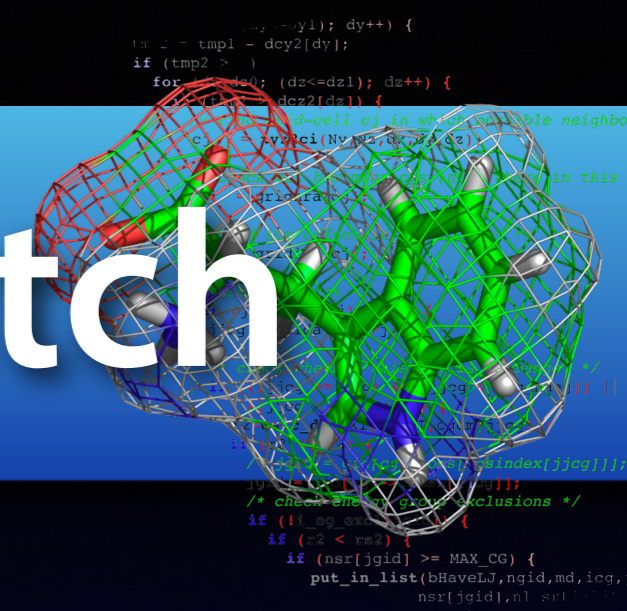


Every single commit is tested automatically on our build farm, including both builds and regression tests.

Results are integrated into the gerrit review

- Catches Cmake build errors
- Catches Google test unit test failures

CI tests - for every patch



- **Unit Tests:** *Do modules reproduce reference values?*
- **Regression tests:** *Are previous simulation results identical?*
- **Clang AddressSanitizer:** *Catch silent memory errors*
- **Clang/GCC ThreadSanitizer:** *Thread synchronization errors*
- **Clang Static Analyzer:** *Logical execution dependency errors*
- **Uncrustify:** *Proper code formatting, no tabs, brace standards?*
- **Doxygen:** *All classes/methods/arguments/variables documented?*

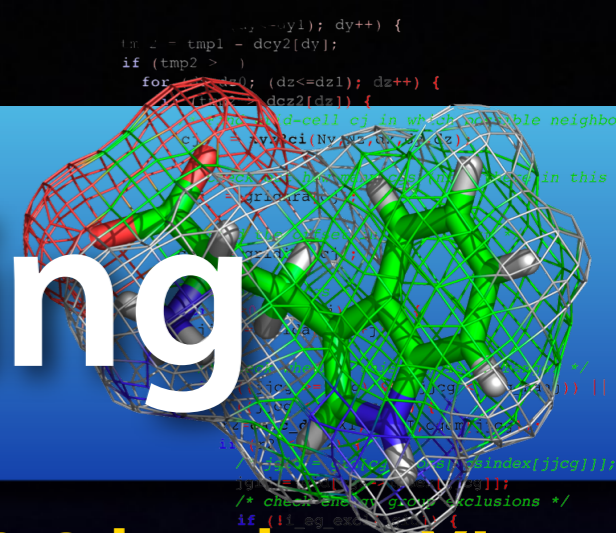
Book-keeping

Bugtracking

Feature tracking

Developer discussions

Redmine issue tracking



Gromacs - Issues - Gromacs development

http://redmine.gromacs.org/projects/gromacs/issues?set_filter=14

Home Projects Help

Gromacs

Overview Activity Roadmap **Issues** Documents Files Repository

Issues

Filters: Status: open Tracker: is Bug

| # | Project | Tracker | Status | Priority | Subject | Assignee | Updated | Target version |
|-----|---------|---------|--------|----------|---|-----------|---------------------|----------------|
| 962 | Gromacs | Bug | New | Normal | segv/hang of EM with foreign lambda's | Berk Hess | 06/25/2012 11:30 am | 4.5.6 |
| 959 | Gromacs | Bug | New | Normal | Issue with nonhomogeneous boundaries and domain decomposition | | 06/20/2012 03:22 am | 4.6 |
| 958 | Gromacs | Bug | New | Normal | MPI on Windows | | 06/27/2012 07:01 am | |
| 957 | Gromacs | Bug | New | Normal | Spurious parameters for Argon in OPLS-AA and Charmm27 | | 06/15/2012 01:18 am | |
| 956 | Gromacs | Bug | New | Normal | Unit cell expands in X/Y during semiisotropic simulation of an octane slab with 8 threads and -pd but not with 2 threads or when using -dd on 8 | | 06/25/2012 04:19 pm | |

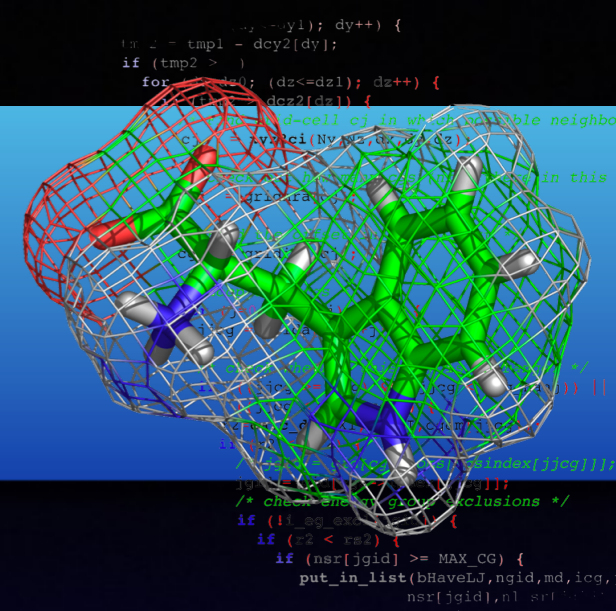
- Version 1.2.3 has bug X!
- Windows builds broke
- How is the work going on refactoring module Y?
- Should we improve scaling by method Z or W?
- Why did we decide to modify that loop in file F in git change Icfca5a?

Automatic referencing in commit messages!

```
Closes #926 - Raw assembly code has been removed.
Refs #923 - Old kernels removed, new will be added shortly.
Fixes #914 - Cmake now does architecture-specific optimization.
Fixes #912, #913
Fixes #857 - We detect rdtscp support with CPUID and use it if possible.
Fixes #750
Closes #537, #574 - Altivec is now deprecated.

Change-Id: Icfca5a940762f8d82ae67b59c65b2d2ac683256d
```

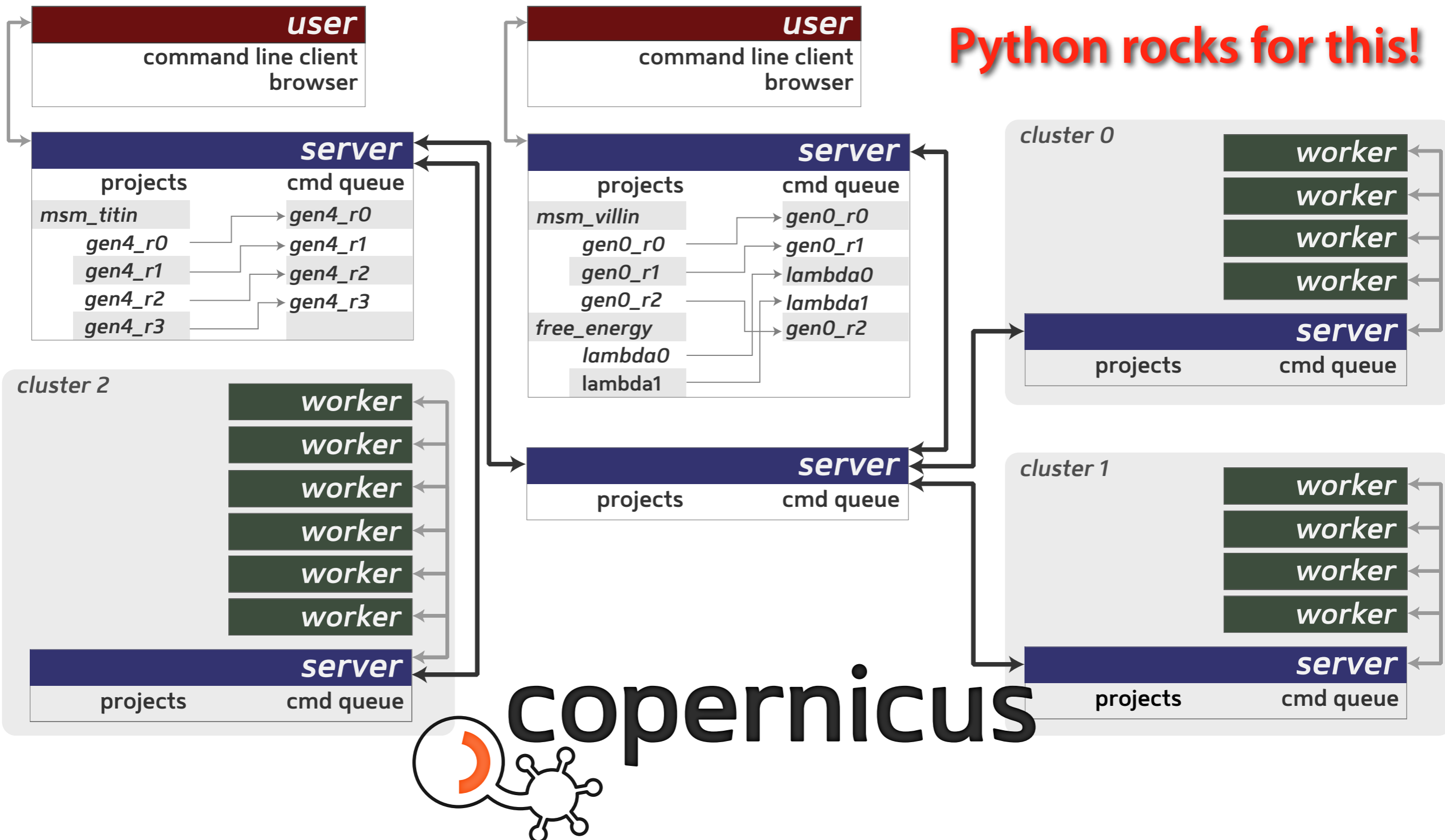
Languages?



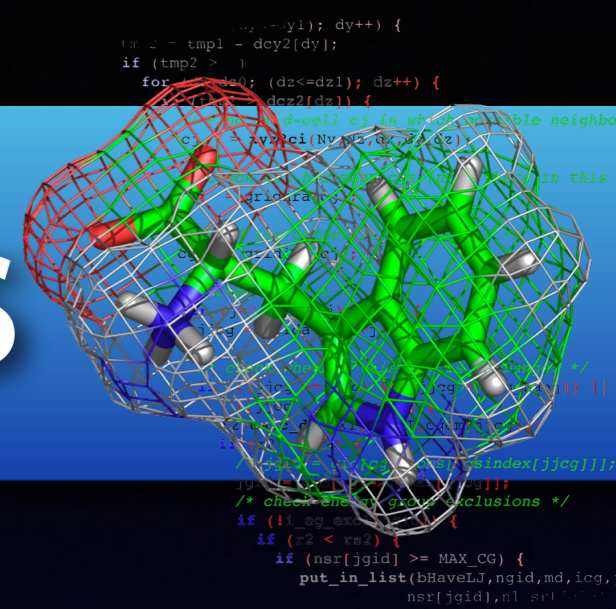
- **“REAL PROGRAMMERS CAN WRITE FORTRAN IN ANY LANGUAGE”**
- **“C combines the flexibility and power of assembly language with the user-friendliness of assembly language.”**
- **“C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off.”**
- **The actual C++ nightmare: *You accidentally create a dozen instances of yourself and shoot them all in the foot. Providing emergency medical care is impossible since you can't tell which are bitwise copies and which are just pointing at others and saying, "That's me over there."***

New level (for us) of network-centric programming

Python rocks for this!



Language hierarchies



User-level code in Python

*High-level code in C++
(most C++ features allowed)*

*Low-level code in C++
(very few C++ features allowed)*

C Inline ASM intrinsics

Copernicus

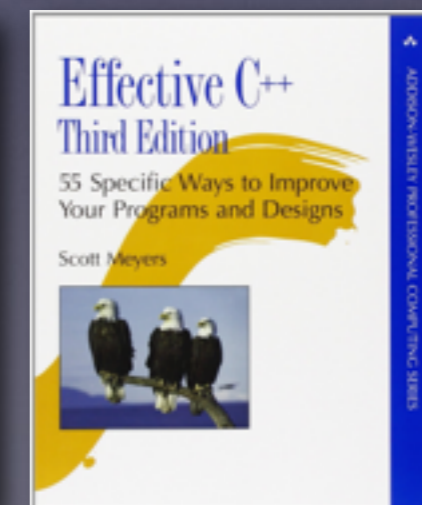
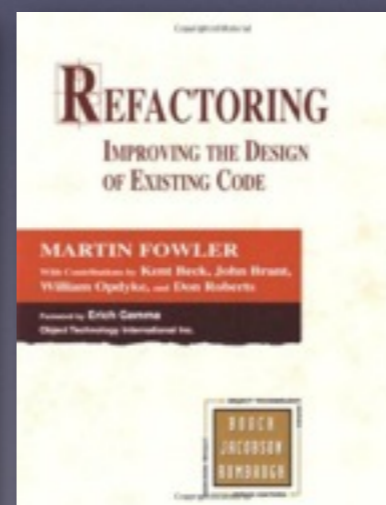
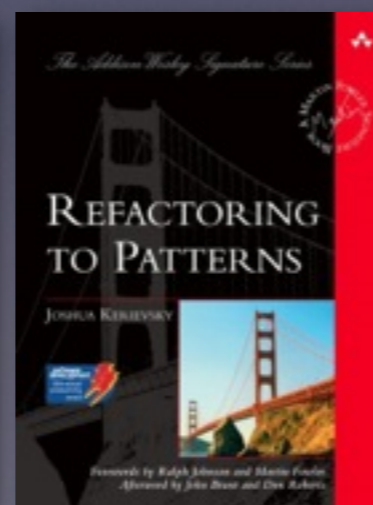
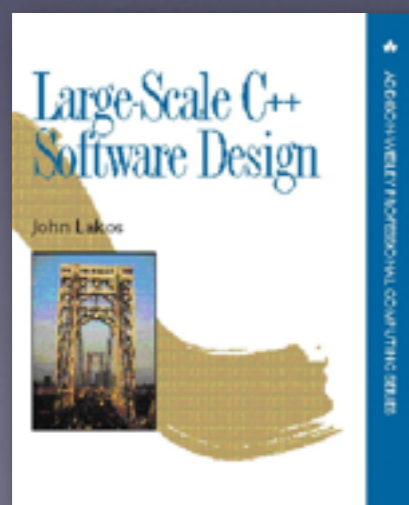
**Object instantiation/destruction
handles memory (de-)allocation
Exception-based error handling**

**Performance-sensitive code
Direct memory manipulation**

**`a = _mm_add_ps(b,c);`
(translated into single instruction)**

Some good reading

- Large-scale C++ software design [John Lakos]
- Design Patterns - Elements of Reusable Object-oriented software [Gamma, Helm, Johnson, Vlissides] “Gang of four”
- Refactoring to Patterns [Joshua Kerievsky]
- Refactoring - improving the design of existing code [Martin Fowler]
- Effective C++ - 55 specific ways to improve your programs and design [Scott Meyers]
- Patterns for concurrent, parallel, and distributed systems:
<http://www.cs.wustl.edu/~schmidt/patterns-ace.html>
- What everybody should know about floating-point math:
<http://randomascii.wordpress.com/category/floating-point/>



Use the source, Luke

<http://www.gromacs.org>

<git://git.gromacs.org>

<http://gerrit.gromacs.org>

<http://redmine.gromacs.org>

<http://jenkins.gromacs.org>

(there are lots of other programs out there too!)