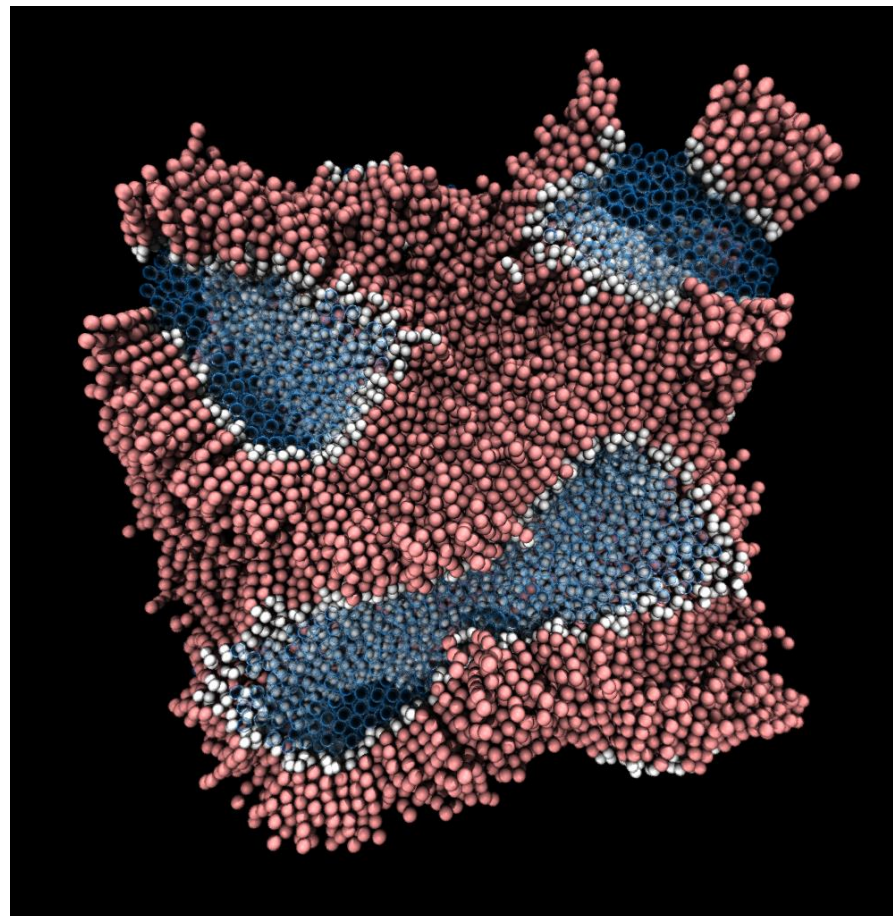


Helping build a developer community through good software practice

Dr Mark Abraham

HPC Carpentry for Chemists, University of Mauritius, August 20, 2021

Lessons from GROMACS evolution - biomolecular simulation



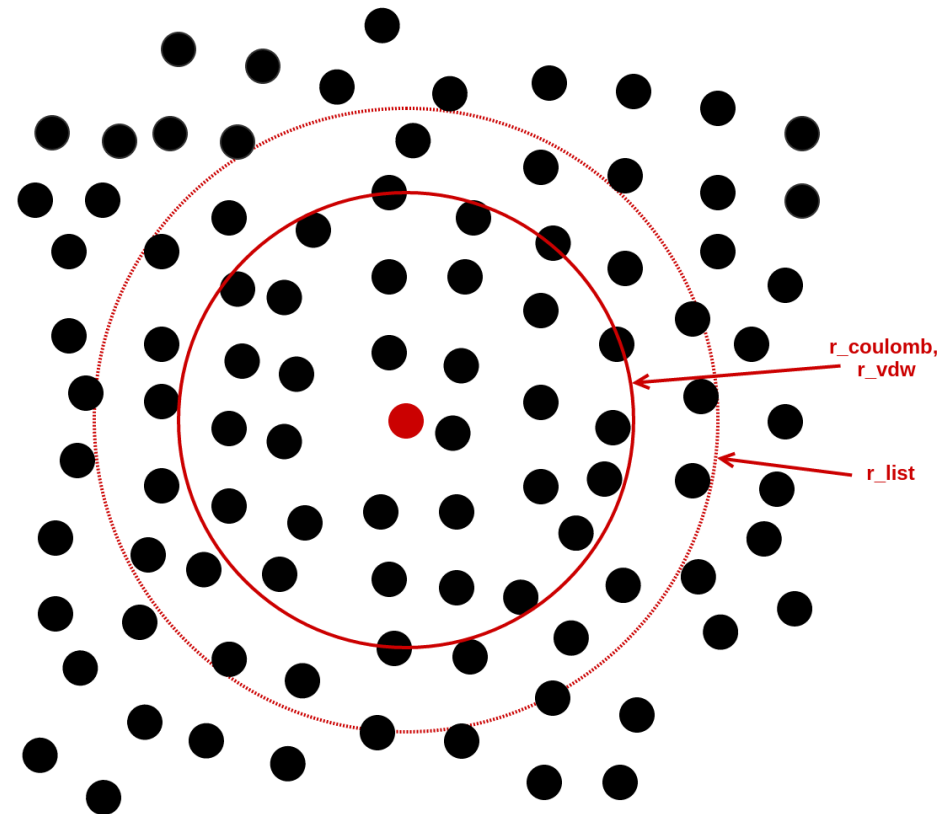
Humble origins

- Originally a **hardware** project at the University of Groningen in the early 1990s – custom-designed 32-processor ring using PVM
- Design and simulation methodology largely adopted from existing GROMOS molecular simulation package
- **Key decision:** write in a language that suits the problem you want to solve and where you want to spend your time

Early software practice

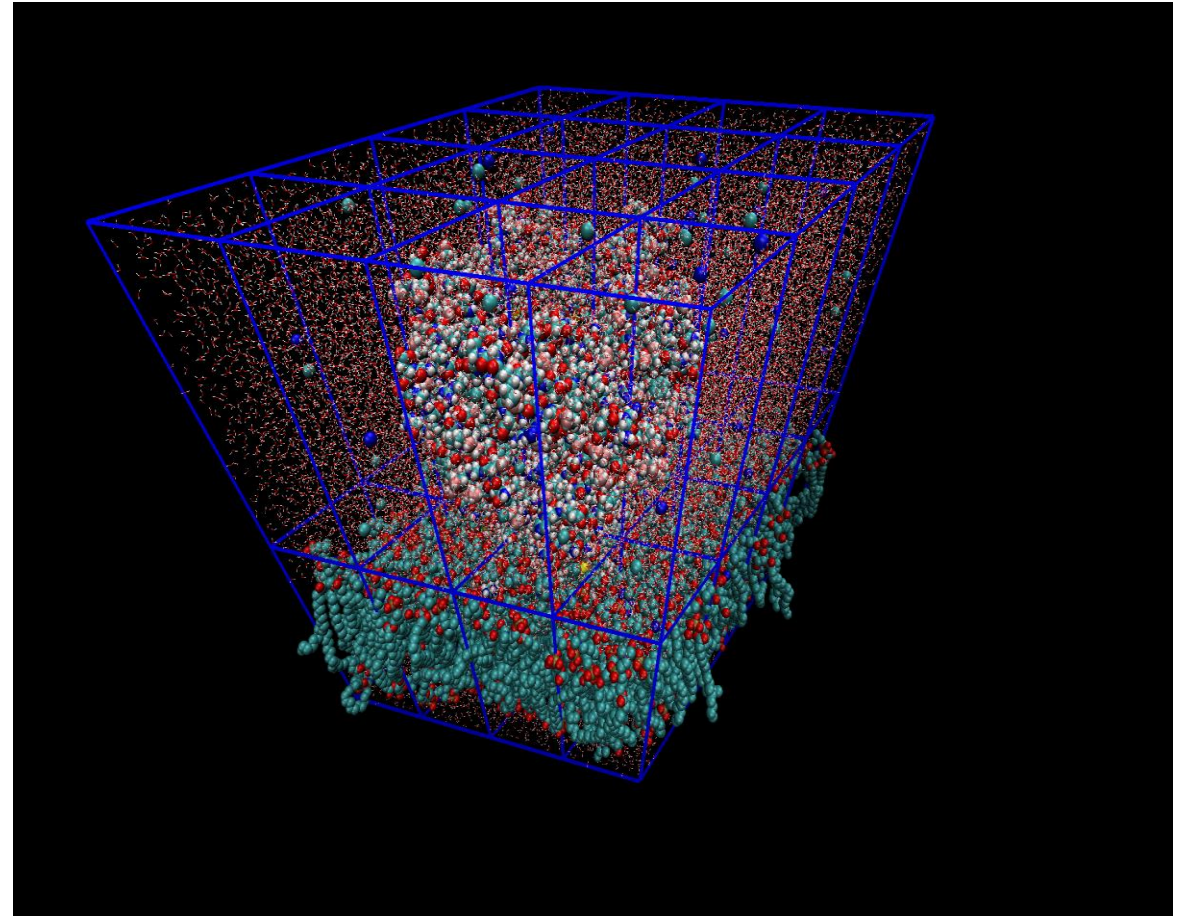
- First public release – 2.0 some time in the 1990s
- First paper – “GROMACS: a message-passing parallel molecular dynamics implementation” *Comp. Phys. Comm.* 1995
- **Key decision:** use version control – first CVS commit in 1997, already 750 files and 145k LOC
- **Key decision:** first FOSS release – 3.0 in August 2001, GPL 2.0
- **Key decision:** provide preparation and analysis tools along with the core simulation engine

N-body neighbor lists



Software practice evolves

- **Key decision:** Use issue tracking, Bugzilla introduced summer of 2005
- **Key decision:** Develop a regression test suite 2007
- **Key decision:** Switch to git for distributed version control (June 2009)
- **Key decision:** Rewrite parallelism layer, keeping the old one working

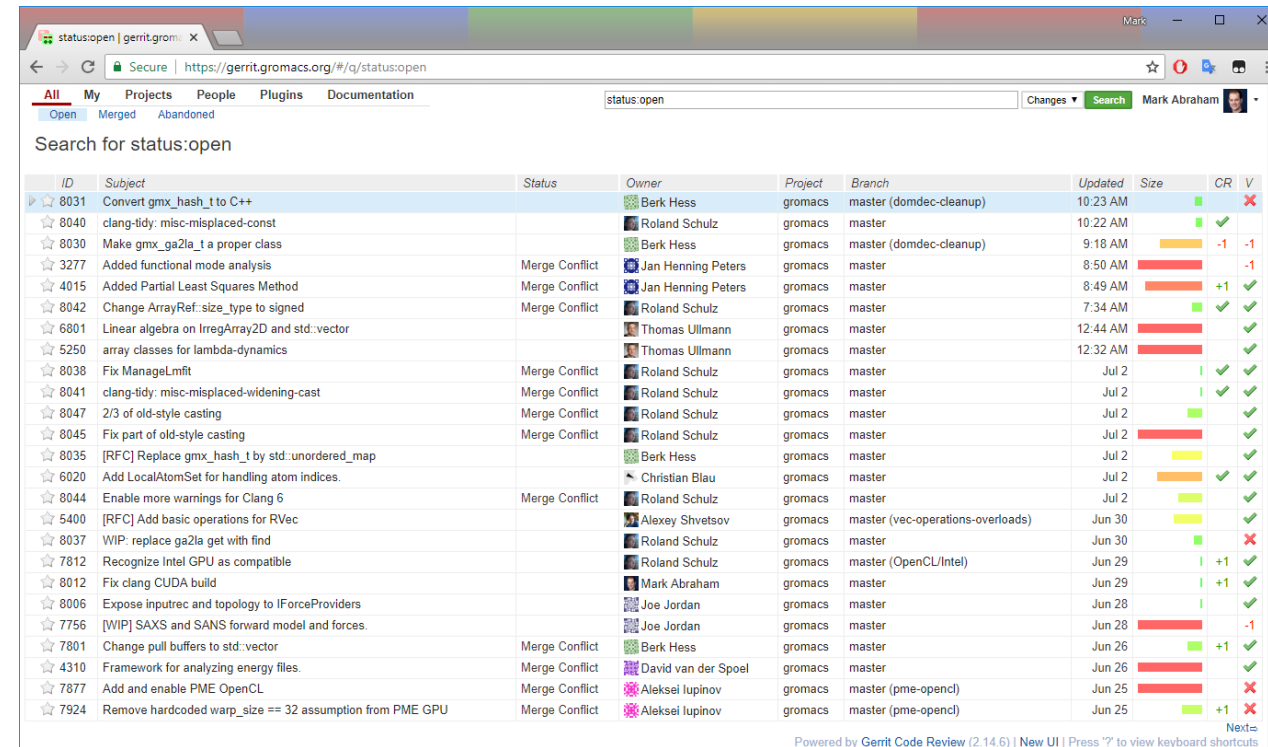


Software practice evolves...

- Because of the high performance, portability, and usability, usage grew
- User community included also many with desire to modify the code and contribute back
- Many new contributors permitted to push to the central git repository over 2008-2010
- Note: not all contributions need to be new code!
- **Key decision:** unite documentation git repository with source code repository (2013)

Growth problems!

- A large community of developers who are not dog-fooding each other's code will break everything, all the time
- Many new code paths led to many bugs, some of them subtle
- **Key decision:** introduce open code review (2011) before changes are accepted – <https://gerrit.gromacs.org>
- **Key decision:** everybody's changes must be reviewed by others, including changes from the long-term developers (who were mostly now busy professors)



ID	Subject	Status	Owner	Project	Branch	Updated	Size	CR	V
8031	Convert gmx_hash_t to C++		Berk Hess	gromacs	master (domdec-cleanup)	10:23 AM			X
8040	clang-tidy: misc-misplaced-const		Roland Schulz	gromacs	master	10:22 AM			
8030	Make gmx_ga2la_t a proper class		Berk Hess	gromacs	master (domdec-cleanup)	9:18 AM		-1	-1
3277	Added functional mode analysis	Merge Conflict	Jan Henning Peters	gromacs	master	8:50 AM			
4015	Added Partial Least Squares Method	Merge Conflict	Jan Henning Peters	gromacs	master	8:49 AM		+1	
8042	Change ArrayRef::size_type to signed	Merge Conflict	Roland Schulz	gromacs	master	7:34 AM			
6801	Linear algebra on IrregArray2D and std::vector		Thomas Ullmann	gromacs	master	12:44 AM			
5250	array classes for lambda-dynamics		Thomas Ullmann	gromacs	master	12:32 AM			
8038	Fix ManageLmfit	Merge Conflict	Roland Schulz	gromacs	master	Jul 2			
8041	clang-tidy: misc-misplaced-widening-cast	Merge Conflict	Roland Schulz	gromacs	master	Jul 2			
8047	2/3 of old-style casting	Merge Conflict	Roland Schulz	gromacs	master	Jul 2			
8045	Fix part of old-style casting	Merge Conflict	Roland Schulz	gromacs	master	Jul 2			
8035	[RFC] Replace gmx_hash_t by std::unordered_map		Berk Hess	gromacs	master	Jul 2			
6020	Add LocalAtomSet for handling atom indices		Christian Blau	gromacs	master	Jul 2			
8044	Enable more warnings for Clang 6	Merge Conflict	Roland Schulz	gromacs	master	Jul 2			
5400	[RFC] Add basic operations for RVec		Alexey Shvetsov	gromacs	master (vec-operations-overloads)	Jun 30			
8037	WIP: replace ga2la get with find		Roland Schulz	gromacs	master	Jun 30			
7812	Recognize Intel GPU as compatible		Roland Schulz	gromacs	master (OpenCL/Intel)	Jun 29		+1	
8012	Fix clang CUDA build		Mark Abraham	gromacs	master	Jun 29		+1	
8006	Expose inputrec and topology to IForceProviders		Joe Jordan	gromacs	master	Jun 28			
7756	[WIP] SAXS and SANS forward model and forces		Joe Jordan	gromacs	master	Jun 28			
7801	Change pull buffers to std::vector	Merge Conflict	Berk Hess	gromacs	master	Jun 26		+1	
4310	Framework for analyzing energy files	Merge Conflict	David van der Spoel	gromacs	master	Jun 26			
7877	Add and enable PME OpenCL	Merge Conflict	Aleksel Iupinov	gromacs	master (pme-opencl)	Jun 25			
7924	Remove hardcoded warp_size == 32 assumption from PME GPU	Merge Conflict	Aleksel Iupinov	gromacs	master (pme-opencl)	Jun 25		+1	

More problems!



The screenshot shows the Jenkins dashboard with a list of builds. The left sidebar contains navigation links like 'Check File Fingerprint', 'Manage Jenkins', 'Query and Trigger Gerrit Patches', 'My Views', 'Plugin Usage', 'Credentials', and 'New View'. Below these are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing various executors like 'master', 'bs-win2012r2', 'bs_gpu01', 'bs_jetson_tk1' and their current status). The main table lists builds with columns for job name, duration, build number, and status. Jobs include 'clang_static_analyzer_PreSubmit', 'Coverage_Nightly_master', 'Coverage_OnDemand', 'cpccheck_PreSubmit', 'Documentation_Nightly_2016', 'Documentation_Nightly_master', 'Documentation_PreSubmit', 'Gromacs_Nightly_master', 'Gromacs_OnDemand_workflow', 'Gromacs_PostSubmit_2016', 'Gromacs_PostSubmit_2018', 'Gromacs_PostSubmit_master', 'Gromacs_PreSubmit_2016', 'Gromacs_PreSubmit_2018', 'Gromacs_PreSubmit_master', 'Matrix_Nightly_master', 'Matrix_OnDemand', 'Matrix_PostSubmit_2016', 'Matrix_PostSubmit_2018', 'Matrix_PostSubmit_master', 'Matrix_PreSubmit_2016', 'Matrix_PreSubmit_2018', and 'Matrix_PreSubmit_master'.

Job Name	Duration	Build Number	Status
clang_static_analyzer_PreSubmit	57 min	#5688	1 day 2 hr - #5671
Coverage_Nightly_master	9 hr 18 min	#610	N/A
Coverage_OnDemand	1 yr 10 mo	#34	1 yr 10 mo - #33
cpccheck_PreSubmit	16 min	#5672	N/A
Documentation_Nightly_2016	11 hr	#365	N/A
Documentation_Nightly_master	11 hr	#1050	N/A
Documentation_PreSubmit	57 min	#5638	1 day 2 hr - #5621
Gromacs_Nightly_master	8 hr 30 min	#266	7 mo 27 days - #34
Gromacs_OnDemand_workflow	9 days 9 hr	#554	4 days 2 hr - #558
Gromacs_PostSubmit_2016	3 mo 14 days	#83	11 mo - #32
Gromacs_PostSubmit_2018	16 days	#239	10 days - #238
Gromacs_PostSubmit_master	1 mo 5 days	#596	2 hr 57 min - #673
Gromacs_PreSubmit_2016	3 mo 7 days	#986	4 mo 27 days - #975
Gromacs_PreSubmit_2018	9 days 10 hr	#763	1 mo 12 days - #690
Gromacs_PreSubmit_master	1 hr 43 min	#7713	15 hr - #7704
Matrix_Nightly_master	8 hr 29 min	#233	N/A
Matrix_OnDemand	4 days 2 hr	#402	1 mo 2 days - #395
Matrix_PostSubmit_2016	3 mo 14 days	#82	N/A
Matrix_PostSubmit_2018	16 days	#240	18 days - #239
Matrix_PostSubmit_master	10 days	#640	2 hr 56 min - #678
Matrix_PreSubmit_2016	3 mo 7 days	#214	4 mo 27 days - #204
Matrix_PreSubmit_2018	9 days 10 hr	#739	19 days - #732
Matrix_PreSubmit_master	1 hr 42 min	#4617	15 hr - #4609

- Portability requires ongoing testing
- Developers want to use their laptops, but HPC clusters and distributed computing environments look very different
- **Key decision:** Use continuous-integration testing, which must pass before code is considered for acceptance (2012), including static and dynamic analyzers and linting
- **Key decision:** require unit test for new functionality, add coverage also when modifying old functionality

Recent developments

- **Key decision:** Refactor to support a long-term stable API, callable from C++ and Python, easing pressure for non-core developers to want to contribute to the core, and reducing burden on core developers
- **Key decision:** switch to GitLab so everything is under one roof
- **Key decision:** use containerization to reduce CI maintenance burden

Lessons for your developer community

- Rome wasn't built in a day – pick the battles that suit the current state of your project, the needs of your users, and the resources you can afford
- Automation of processes pays off well
- Pre-commit CI testing makes bug hunting easier – you know you can go back in time, build the old code and it will work
- If you need more performance, consider your algorithms and data structures, match them to the capabilities of the target hardware.
- Plan first how you will test that your code is correct!

Thank you!



Swedish
Research
Council

enccs.se