



Technische
Universität
Braunschweig



Towards sustainability of the legacy research CFD code VirtualFluids

deRSE 2024

Sören Peters, 07.03.2024

Software Overview

- Computational Fluid Dynamics (CFD) solver based on Lattice Boltzmann Method (LBM)
- turbulent and multicomponent flow problems
- pre- and postprocessing capabilities for simulations with more than 100 billion degrees of freedom.
- developed at the Institute for Computational Modeling in Civil Engineering (iRMB) at TU Braunschweig
- under development for 25 years by various researchers



Software Overview



- Main Technologies: C++ 17, CMake, CUDA, Python
- ~125000 LOC
- Platforms: Linux, Windows (x64) (MacOS)
- Compiler: GCC, Clang, MSVC, (AppleClang, Intel)
- High-Performance Computing on Shared- and Distributed Memory Architectures

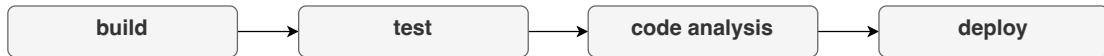
- Project in Braunschweig and Erlangen fostering the sustainability of research software by helping researchers adopt practices and tools from the software engineering community.¹



¹<https://suresoft.dev/>

Continuous Integration

- automated builds, tests and code quality checks
- Treat Warnings as Errors!
- test the functionality of the software
- deploy documentation (doxygen) pages ¹



¹<https://irmb.gitlab-pages.rz.tu-bs.de/VirtualFluids/index.html>





- Unit-Tests based on GoogleTest ¹
- Regression tests using Fieldcompare ²
- HPC tests with HPC-Rocket ³
- Performance Tests

¹<https://github.com/google/googletest>

²<https://joss.theoj.org/papers/10.21105/joss.04905>

³<https://zenodo.org/doi/10.5281/zenodo.7568958>

Static Code Analysis

- utilizing static code analysis to track code quality
 - cppcheck¹
 - clang-tidy¹
 - lizard²
 - gcov³
 - Sanitizer



¹<http://cppcheck.net/>

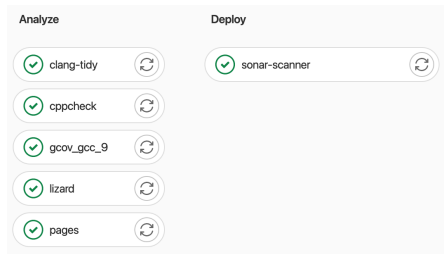
²<https://clang.llvm.org/extra/clang-tidy/>

³<https://pypi.org/project/lizard/>

⁴<https://gcc.gnu.org/onlinedocs/gcc/Gcov.html>

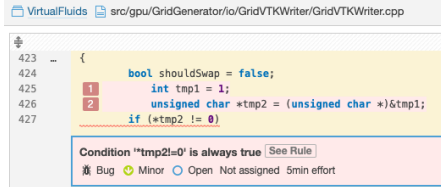
Static Code Analysis

- static analysis is executed in the ci pipeline
- results are then transferred to Sonarqube



Static Code Analysis

- local host of a sonarqube¹ instance to visualise the code quality



The screenshot shows a SonarQube interface with a file path: VirtualFluids / src/gpu/GridGenerator/io/GridVTKWriter/GridVTKWriter.cpp. The code snippet is as follows:

```
423 ... {  
424     bool shouldSwap = false;  
425     int tmp1 = 1;  
426     unsigned char *tmp2 = (unsigned char *)&tmp1;  
427     if (*tmp2 != 0)
```

Two lines of code are highlighted with red boxes and numbers 1 and 2. Below the code, a message box states: "Condition '*tmp2!=0' is always true" with a "See Rule" link. Below this, there are icons for Bug, Minor, Open, Not assigned, and 5min effort.

¹<https://github.com/SonarSource/sonarqube>

Virtualisation



- VirtualFluids development container provided in GitLab container registry ²
 - Based on NVIDIA container for GPU usage ³
- Development based on vscode and devcontainer ¹
 - low entry barrier -> ready to develop
- on HPC-Cluster using Singularity and HPC-rocket ⁴

¹ <https://irmb.gitlab-pages.rz.tu-bs.de/VirtualFluids/getting-started-with-docker.html>

² https://git.rz.tu-bs.de/irmb/virtualfluids/container_registry

³ <https://hub.docker.com/r/nvidia/cuda>

³ <https://zenodo.org/records/7568959>



GitLab Workflow

- Switched from internal institute git server to instance of the university ¹
- Public Repository containing Main and Develop Branch
- Forking Strategy - Merge Requests into develop
- Once in a while – merge in main and create a tag -> Code Publication



¹<https://git.rz.tu-bs.de/irmb/virtualfluids>





- Merge Develop into Main and create a Tag on Main
- Main Branch and Tag are mirrored to GitHub ¹
- Create Release on GitHub
- Zenodo is connected to GitHub ²
 - Create Release on GitHub -> triggers publication on zenodo ³
 - Meta data are taken from CFF file ⁴

¹<https://github.com/irmb/virtualfluids>

²<https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content>

³<https://zenodo.org/doi/10.5281/zenodo.10283048>

⁴<https://citation-file-format.github.io/>



Conclusion

- Use the Tools!
- CI + Container + Tests = <3
- But: You need to maintain and you need people who feel responsible

