codee

Codee for large codes: GROMACS case study

Codee is the first static code analyzer that is designed specifically to boost the performance of C/C++ code. Earlier generations of source code analysis are limited to bugs, coding standard enforcement or security, possibly even a combination of these, which while important do nothing to ensure that the code is written to take advantage of modern hardware capabilities offered by chip manufacturers in the low-power multicore processors.

> "GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles."

We have used the GROMACS project to benchmark how Codee behaves for large code bases. The highlights are summarized below:

340K lines of code across 2,481 C++ files

GROMACS contains 2,481 C++ files accounting for 340,449 lines of code, as reported by the *sloccount* tool.

940 of these C++ files correspond to headers, which are only analyzed when included from *.cpp* files.

99.6% of files successfully analyzed

We exclude 251 files (17 files that fail due to missing headers plus third-party code).

1,282 out of 1,287 *.cpp* files were successfully analyzed.

3 secs/file on average, for a total of 1 hour and 7 minutes analysis time

It took a total of 4.204 seconds to run for the 1,287 files[1].

The peak of RAM usage was 518 MBs according to the *valgrind* tool.

0.39% error rate

Codee successfully analyzed 1,926 loops found in the code. 984 out of those were reported as opportunities to increase performance.

Our R&D team reviewed[2] 52% of those opportunities and found only 2 of them to be erroneous.

1 On a laptop with an AMD Ryzen 7 4800HS CPU.
2 The review consisted of carefully examining loop variables' compute patterns and data scoping classification.

The above analysis can be reproduced by following the instructions in our GitHub repository.