

Software Complexity is Growing & Requires Higher Performance

- Software is critical across all modern industries. And there are industries with time-critical software where performance is key for competitive advantage
- Examples include Renewable Energies, Mobility, Electronics, Healthcare and, definitely, High Performance Computing (HPC)

Application Performance Critical Success Factors

- Time-critical applications written in C/C++/Fortran require the use of specialized development tools in order to achieve peak performance
- Compilers automate the optimization of C/C++/Fortran code but do not provide peak performance
- Codee provides additional performance taking advantage of optimizations overlooked by the compiler

Benefits

- Quick assessment
- Automate code optimization
- Deliver faster application
- Save software development cost

Overview

Codee is a software development platform that provides tools to help improve the performance of C/C++/Fortran applications.

Codee Static Code Analyzer provides a systematic predictable approach to enforce C/C++/Fortran performance optimization best practices for the target environment: hardware, compiler and operating system. It provides innovative Coding Assistant capabilities to enable semi-automatic source code rewriting through the software development lifecycle.

Codee provides integrations with IDEs and CI/CD frameworks to make it possible to Shift Left Performance.

Key Features



Screening report to do a quick assessment of the potential performance gain of the C/C++/Fortran code.



Checks report to list the checkers of the open catalog of performance optimization best practices found in the C/C++/Fortran code.



Coding assistant features to enable semi-automatic source code rewriting, always under the control and upon invocation by the programmer. Includes annotation with compiler directives targeting microcontrollers, microprocessors and accelerators.



Integration with compiler optimization reports, including support for GCC, Clang, Intel oneAPI and Microsoft Visual Studio.



Interoperability with CI/CD frameworks through data exchange file formats.



Open catalog of performance optimization best practices for C/C++/Fortran.



Performance Optimization Process for C/C++/Fortran Codes

The C/C++/Fortran code optimization areas are ultimately the same on most types of processors:

- Memory efficiency
- Vectorization
- Multi-threading

Developers that are experts in C/C++/Fortran code optimization use programming techniques specialized in performance to speed up runtime.

Codee provides automated code inspection capabilities specialized in C/C++/Fortran codes with <2% false positives, a very low error rate.

Open catalog of performance optimization best practices:

www.codee.com/catalog/

1	Scalar	Optimize compute instructions
2	Control	Optimize the conditional instructions
3	Memory	Optimize memory efficiency
4	Vector	Optimize using vector instructions
5	Multi	Optimize using multiple threads
6	Offload	Optimize using accelerators
7	Quality	Compliance with performance best practice

Reproducible Codee Benchmarking: Speeding up C/C++/Fortran Codes

Codee discovers performance optimization opportunities in C/C++/Fortran source code, enabling to benefit from the memory efficiency, vectorization, multithreading and offloading capabilities available in modern computers.

Reproduce the impact on performance for selected codes from embedded and high-performance computing benchmarks. Follow the step-by-step instructions in the repository to measure the performance gains on your environment.

Performance demos repository:

github.com/codee-com/performance-demos/

github.com/codee-com/performance-demos-fortran/

(*) Benchmarking conducted using Codee 2023.1.1 (July 2023) on a machine equipped with AMD Ryzen 7 4800H CPU, 16GB of DDR4 memory, GNU/Linux Ubuntu 20.04 Operating System, and GCC 11.1 compilers. The compiler optimization levels are as follows: MBedTLS release with -O2 and MULTI/single-core benchmarks with -O3.




```
$ CC=gcc-11 RUNS=10 RUNS_WARMUP=5
./benchmark-single-core-optimizations.sh
...
Code      Original  Optimized  Speedup
=====  =====  =====  =====
MATMUL    5.47      1.45      73.58% (3.79x)
```

```
$ CC=gcc-11 RUNS=10 RUNS_WARMUP=5
./benchmark-mbedtls-vector.sh
...
Code      Original  Optimized  Speedup
=====  =====  =====  =====
AES-XTS-128 609984 KiB/s 771191 KiB/s 26.43%
AES-XTS-256 534216 KiB/s 650501 KiB/s 21.77%
AES-CBC-128 686260 KiB/s 995623 KiB/s 45.08%
AES-CBC-192 638178 KiB/s 894059 KiB/s 40.10%
AES-CBC-256 591336 KiB/s 807582 KiB/s 36.57%
AES-CMAC-128 676441 KiB/s 885630 KiB/s 30.92%
AES-CMAC-192 633305 KiB/s 811924 KiB/s 28.20%
AES-CMAC-256 588881 KiB/s 736039 KiB/s 24.99%
AES-CMAC-PRF-128 678421 KiB/s 881149 KiB/s 29.88%
ARIA-CBC-128 151316 KiB/s 159032 KiB/s 5.10%
ARIA-CBC-192 133866 KiB/s 139109 KiB/s 3.92%
ARIA-CBC-256 117773 KiB/s 123772 KiB/s 5.09%
```

```
$ CC=gcc-11 RUNS=10 RUNS_WARMUP=5 ./benchmark-omp-multi.sh
...
Code      Original  Optimized  Speedup
=====  =====  =====  =====
ATMUX     0.78      0.12      85.00% (6.70x)
CANNY    10.96     6.27      42.80% (1.75x)
COULOMB   8.26      1.06      87.20% (7.81x)
HACCmk    36.49     8.15      77.67% (4.48x)
MATMUL    5.23      0.98      81.20% (5.32x)
NPB_CG   33.17     9.94      70.03% (3.34x)
PI        3.28      0.43      87.02% (7.70x)
```

Release binary distribution

The Codee software is distributed as a binary release for the platforms listed in the following table:

	 Linux (*1) (*4)	 Windows (*2) (*4)	 MacOS (*3) (*4)
x86_64	✓	✓	✓
aarch64	✓		
ppc64le	✓		

(*1) The installation of Codee on Linux requires the C++ Runtime Library *libstdc++* version 6.0.21 or later (it is often installed by default in every Linux desktop distribution so no additional steps are often required).

(*2) The installation of Codee on Windows requires the [Microsoft Visual C++ Redistributable](#) package.

(*3) The installation of Codee on MacOS requires the MacOS X High Sierra version 10.15.4 or later.

(*4) By default, Codee does not require a compatible C/C++/Fortran compiler installed in the system. Note Codee can operate in "Compiler-Driven Mode" where it integrates with a selected compiler installed in the system (see the details in the user manual and the list of supported environments later in this document).

Technical Specifications

SUPPORTED PROGRAMMING LANGUAGES

- C/C++ (based on Clang 14.0 which supports C17 and C++17, and provides partial support for C++20)
- Fortran (based on Flang 14.0 which provides partial support for Fortran 90 and Fortran 2018)

SUPPORTED ENVIRONMENTS FOR INTEGRATION WITH COMPILER OPTIMIZATION REPORTS

GNU compiler	gcc version 6.2 up to 12.2	gfortran version 6.2 up to 12.2
LLVM compiler	clang/clang-cl version 7.0 up to 15.0	
Intel oneAPI compiler	icx version 2022.2.0 up to 2023.0	ifx version 2022.2.0 up to 2023.0
Intel oneAPI classic compiler	icc version 16.0 up to 2021.5	lfort version 2022.2.0 up to 2023.0
Msvc compiler	cl version 19.14 up to 19.32	

The environments (compiler, operating system and processor architecture) tested during the Codee release process are as follows:

	Linux Ubuntu 20.04 x86_64	Linux Ubuntu 20.04 aarch64	Linux Ubuntu 20.04 ppc64le	Windows version 10 x86_64 (*1)
GCC 11.1 (gcc, g++, gfortran)	✓	✓	✓	N/A
GCC 9.4 (gcc, g++, gfortran)	✓	✓	✓	N/A
GCC 8.4 (gcc, g++, gfortran)	✓	✓	✓	N/A
GCC 6.4 (gcc, g++, gfortran)	✓			N/A
Clang 14.0 (clang, clang++)	✓	✓		✓
Clang 10.0 (clang, clang++)	✓	✓	✓	N/A
Intel Compiler 2021.7 (icc, icpc, ifort)	✓	N/A	N/A	N/A
MSVS Compiler 19.33 (cl)	N/A	N/A	N/A	✓

(*1) The supported environments on Windows correspond to the native tooling installed with [Microsoft Visual Studio with C++](#).

SUPPORTED INTEGRATIONS WITH INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

- **SARIF** reports are available for exploration of the Checks Report e.g. using VSCode.

SUPPORTED INTEGRATIONS WITH BUILD SYSTEMS

- **CMake** produces JSON compilation database since version 3.5.
- **Ninja** produces JSON compilation database since version 1.2.
- **WAF** produces JSON compilation database.
- **Make** does not support JSON compilation database directly, and the tool *bear* since version 3 is required.
- **FPM (Fortran Package Manager)** does not support JSON compilation database directly, and the tool *bear* since version 3 is required.

SUPPORTED INTEGRATIONS WITH COMPILER DIRECTIVES

- **OpenMP compiler pragmas** version 3.0 or higher up to 4.5.
- **OpenACC compiler pragmas** version 2.0 or higher up to 2.7.
- **GCC / Clang / ICC native compiler pragmas** (supported vendor-specific native pragmas).



Codee's mission is to accelerate the digitalised world empowering businesses to develop fast and improve the performance of software. The innovative Codee Software Platform facilitates to shift left performance thanks to its automated code inspection specifically designed to improve the performance of software. We envision every developer being capable of writing fast software for modern low-power hardware. Codee provides a systematic, predictable approach to performance optimization that enables the inexperienced developers to write faster codes at the level of experts and alleviates the scarcity of qualified senior developers. The benefits are huge to C/C++/Fortran applications in the scope of time-critical industries such as Renewable Energy, Aerospace, Automotive, Telecommunications, and Semiconductors.

For more information:
www.codee.com

Email: info@codee.com
Request a demo: codee.com/request-demo/

Appentra Solutions SL
P.º Marítimo, 22b, 15679
Cambre, A Coruña, Spain

