

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 automated code inspection specifically designed to improve the performance of C/C++/Fortran applications. It provides a systematic predictable approach to optimize C/C++/Fortran source code for the target environment.

Codee scans the C/C++/Fortran source code without executing it, and produces a report for the developer indicating the issue, its location, why it penalizes performance, and how to fix it in order to make the code run faster. It is designed to interoperate with CI/CD systems.

Key Features



Quick assessment of the potential performance improvement by screening the C/C++/Fortran code.



Auto mode to automatically optimize the source code and verify its correctness, targeting vectorization and memory efficiency issues on microprocessors.



Guided mode to semi-automatically optimize the source code for microprocessors and accelerators following best practice recommendations for computations, memory usage and data transfers.



Integration with compiler vectorization reports, including support for GCC, Clang, Intel Compiler 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 applications.



Performance Optimization Roadmap 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:

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

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

Codee uses as a reference our open catalog of performance optimization best practices:

[See the catalog of best practices](#)

Discover the performance issues hidden in your C/C++/Fortran source code using Codee!

Release binary distribution

The Codee software is distributed as a binary release for the platforms specified 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) The installation of Codee requires a compatible C/C++/Fortran compiler installed (see the list of supported environments later in this document).

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 vectorization, multithreading and offloading capabilities available in modern computers.

Reproduce the impact on the performance of selected codes representative of embedded and high-performance computing benchmarks. Follow the instructions in the following GitHub repository:

[Performance demos repository](#)

```

$ ./benchmark-mbedTLS-vector.sh
...
Algorithm      Original      Optimized     Speedup
=====
AES-XTS-128    613506 KiB/s  774841 KiB/s  26.30%
AES-XTS-256    542305 KiB/s  652312 KiB/s  20.29%
AES-CMAC-128   660392 KiB/s  887986 KiB/s  34.46%
AES-CMAC-192   601832 KiB/s  822032 KiB/s  36.59%
AES-CMAC-256   556624 KiB/s  738438 KiB/s  32.66%
AES-CMAC-PRF-128 648914 KiB/s  874481 KiB/s  34.76%
ARIA-CBC-128   152955 KiB/s  159800 KiB/s  4.48%
ARIA-CBC-192   132988 KiB/s  138426 KiB/s  4.09%
ARIA-CBC-256   119809 KiB/s  122843 KiB/s  2.53%
    
```

```

$ ./benchmark-omp-multi.sh
...
Code           Original     Optimized     Speedup
=====
ATMUX          0.29         0.13          53.93% (2.17x)
CANNY          11.72        6.99          40.32% (1.68x)
COULOMB        8.35         1.07          87.22% (7.82x)
HACcmk         36.86        10.74         70.86% (3.43x)
MATMUL         6.36         1.24          80.55% (5.14x)
NPB_CG         44.33        20.25         54.32% (2.19x)
PI             3.29         0.46          86.11% (7.20x)
    
```

(*) Benchmarking conducted on a laptop equipped with Linux operating system, x86 processor AMD Ryzen 7 4800H processor, and 16GB of RAM.

Technical Specifications

SUPPORTED PROGRAMMING LANGUAGES

- C/C++ (based on Clang 13.0 which supports for C17 and C++17, and provides partial support for C++20)
- Fortran (based on Flang rev 0b95852b0d00 which provides partial support for Fortran 90 and Fortran 2018)

SUPPORTED ENVIRONMENTS FOR INTEGRATION WITH COMPILER VECTORIZATION REPORTS

The supported compiler vectorization reports are as follows (check latest versions in the Codee tools):

- GNU compiler: gcc version 6.2 up to 12.2.
- LLVM compiler: clang/clang-cl version 7.0 up to 15.0.
- Intel compiler: icc version 16.0 up to 2021.5.
- MSVC compiler: cl version 19.14 up to 19.32.

Codee is integrated with compiler vectorization reports, which makes its output sensitive to the target compiler version. 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 BUILD SYSTEMS

Codee is integrated with build systems that produce a JSON Compilation Database (typically named e.g. compile_commands.json file):

- **CMake** produces JSON compilation database since version 3.5.
- **Ninja** produces JSON compilation database since version 1.2.
- **Make** does not support JSON compilation database directly, and a tool like *bear* 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



Phone: +34 881 01 55 56
Email: info@codee.com
Request a demo:
www.codee.com/contact-us/

Appentra Solutions SL
Centro de Investigación TIC.
Campus de Elviña S/N
15071, A Coruña, Spain