codee

The First Study of the Impact of Codee on SiFive's LLVM RISC-V Development Ecosystem

Manuel Arenaz
manuel.arenaz@codee.com

Vadim Malenboim
vadim.malenboim@sifive.com

Wednesday, June 26th 2024

Introduction

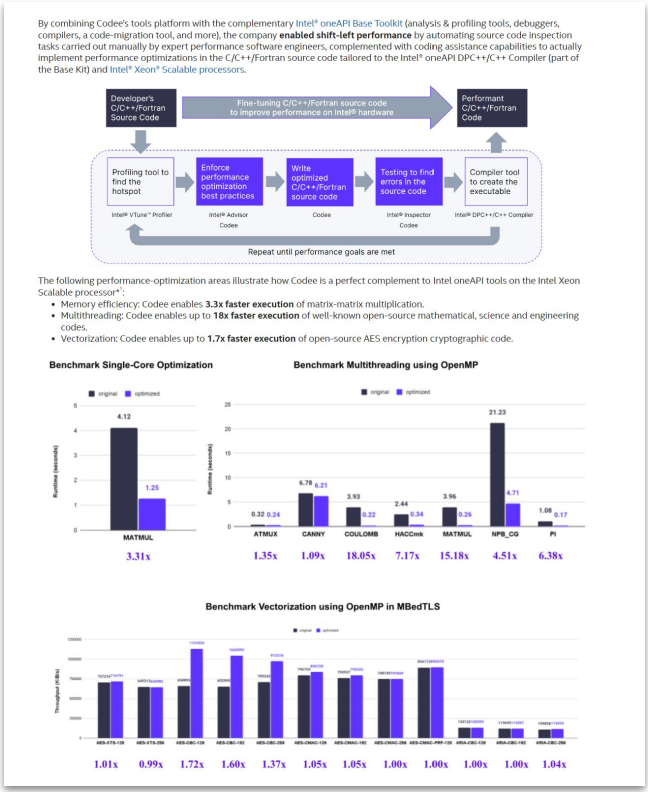
- **Why RISC-V?**
  - Instruction set architecture (ISA) is open, modular, and extensible.
  - Provides a foundation to create custom processors tailored to specific needs.
- **RISC-V is expanding rapidly across a diversity of industries.**
  - Embedded Systems, Edge Computing, AI/ML, High-Performance Computing (HPC), Storage and Networking, Automotive, Aerospace and Defense.
- **Challenges for the widespread adoption of RISC-V?**
  - Competes with the well-established ARM and x86 architectures.
  - The software ecosystems of ARM and x86 are very mature.
- **It is crucial for the success of RISC-V to build a robust ecosystem of software tools, libraries, and developer support.**

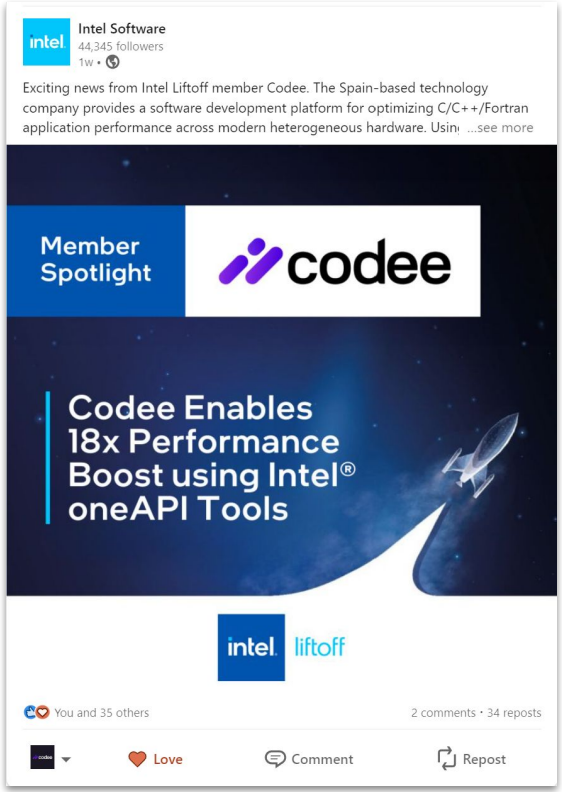
Codee: Static Code Analysis for Modernization and Optimization of Fortran/C/C++

- **Codee is the first static code analyzer specialized in modernization and optimization of Fortran/C/C++ code**
  - Modernization = Improving robustness, stability, maintainability
  - Performance = Improving speed, reducing code size and energy consumption
- **Codee is a complement for the software development ecosystem**
  - It does not replace the compiler, it is a complement to find opportunities overlooked
  - It does not replace the profiler, it receives as input the information about the hotspots of the code
  - It does not replace the debugger, it helps detect bugs and avoids introducing bugs (“shift left”)
  - It is designed to interoperate with IDEs and CI/CD frameworks
- **Codee helps write compiler-friendly hardware-friendly code, favoring maintainability and readability.**

Codee for x86 architecture

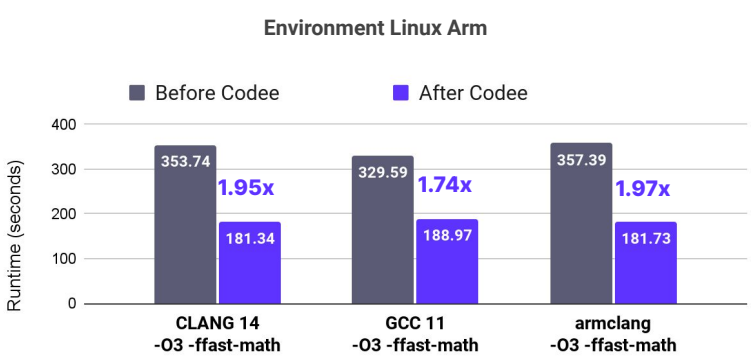
- It was shown to enable up to 18x performance boost on selected HPC workloads using Intel oneAPI Tools targeting Intel Xeon Scalable processors.

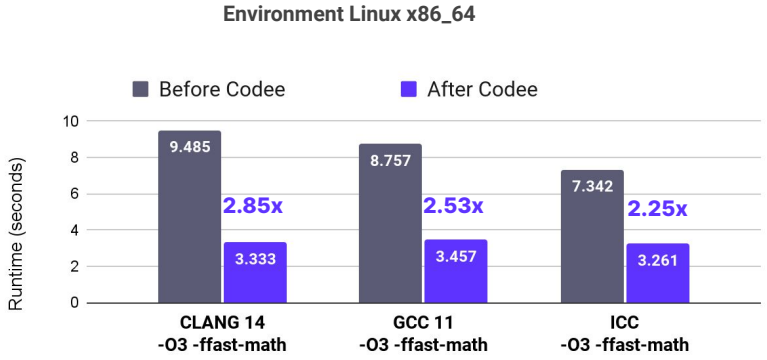




https://www.intel.com/content/www/us/en/developer/articles/case-study/codee-18x-boost-for-compute-intense-workloads.html#gs.3z5td4

Codee for ARM architecture





Codee brings 2x faster code on Arm environments through loop interchange and vectorization

Codee brings 3x faster code on x86 environments through loop interchange and vectorization

Reproducibility using resources in public Github repositories

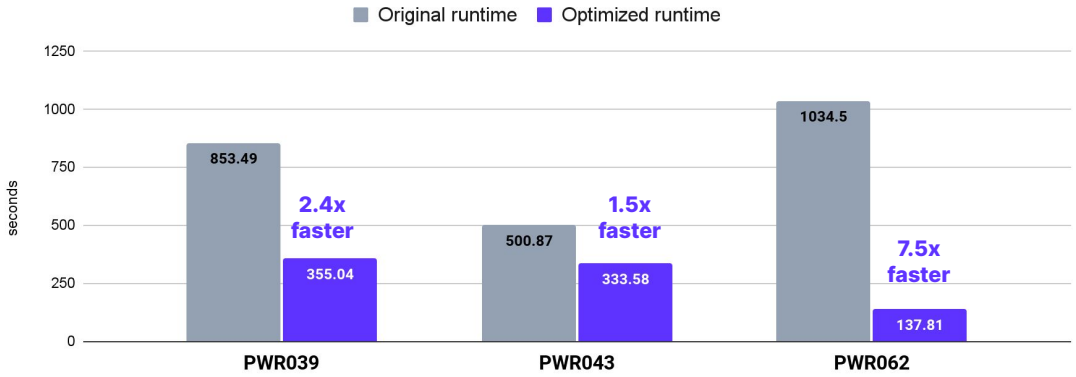
https://github.com/codee-com/open-catalog
https://github.com/codee-com/performance-demos
https://github.com/codee-com/performance-demos-fortran

Experimental Setup for SiFive's RISC-V architecture

- **Objective: Evaluate the impact of Codee in the scope of single-core optimizations**
  - More specifically, focus on vectorization efficiency
  - Also address memory efficiency, favoring sequential memory accesses in order to avoid cache misses.
- **Benchmarking machine equipped with:**
  - Operating system: SiFive LLVM-Linux 15.9.0-2023.03.0
  - Hardware: SiFive P470 Out-of-Order processor, running at 32Mhz on Xilinx VCU118 Ultrascale FPGA
  - Compiler: clang version 15.9.0 cross-compiler targeting SiFive's RISC-V P470 processor ISA.
  - Codee version 2023.1.6 revision number 019119d00ca6 (Oct 2023)
- **Benchmarking methodology:**
  - Average of 5 runs, setting up the compiler's optimization flags to -O3 -ffast-math.
  - Optimize the source code using Codee's detection capabilities and AutoFix'es.
  - Final compilation of the optimized source code with SiFive's cross-compiler.

Experimental Results using MATMUL

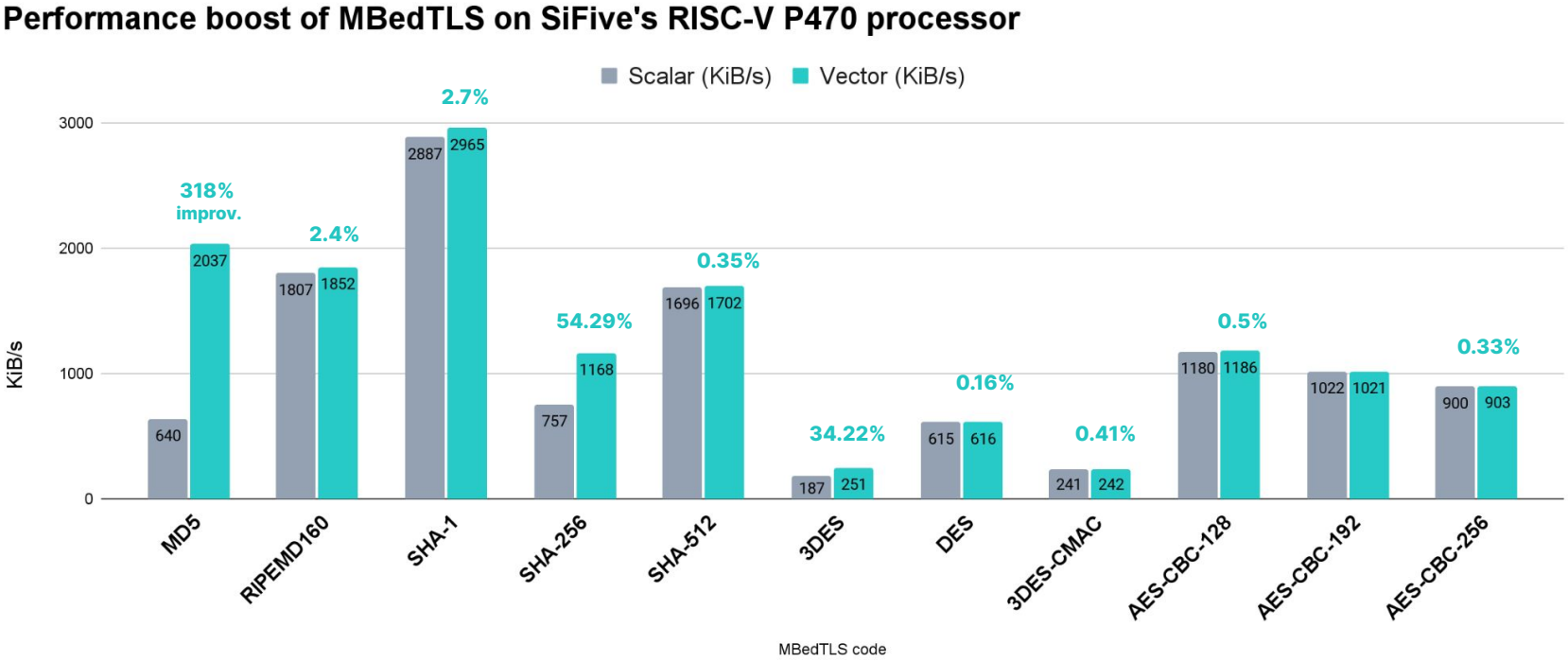
Performance boost of MATMUL on SiFive's RISC-V P470 processor



SiFive's LLVM/Clang compiler powered with Codee to enable higher vectorization efficiency

	SiFive Clang	Codee + Sifive C
PWR039 Loop at main.c:17	Not vectorized	Vectorized (new loop)
PWR043 Loop at matmul.c:29	Vectorized	Vectorized (higher efficiency)
PWR062 Loop at matmul.c:37	Vectorized	Vectorized (higher efficiency)

Experimental Results using MBedTLS embedded code



Conclusions and Next Steps

- **The RISC-V open architecture is competing with the well-established ARM and x86**
- **The LLVM ecosystem for RISC-V is evolving rapidly, but it is not as mature as ARM or x86 yet**
- **Codee is a solution for developers and managers to deliver “better” Fortran/C/C++ code**
  - New static code analyzer specialized in modernization and optimization of Fortran/C/C++
  - Modernization checkers: Improving robustness, stability, maintainability
  - Performance checkers: Improving speed, reducing code size and energy consumption
- **This is the first study of the impact of Codee on LLVM and RISC-V**
  - The results on SiFive’s P470 processor show up to 7.5x performance boost
  - Demonstrating that Codee brings a new and revolutionizing solution applicable to RISC-V
- **Codee makes the upstream LLVM+RISC-V ecosystem even better**
- **Future work: Plan to conduct a more comprehensive study of Codee for LLVM and RISC-V**

codee

Automated Code Inspection for Modernization and Optimization

www.codee.com

info@codee.com

Subscribe: codee.com/newsletter/

Spain

codee_com

/codee-com/