The Eruption of RISC-V in HPC: Earth Sciences Codes on Long Vector Architectures

Pablo Vizcaino [1],*Fabio Banchelli[1], David Jurado[1], Marta Garcia-Gasulla[1] and Filippo Mantovani[1]

[1]Barcelona Supercomputing Center (Spain)

Abstract

In this research poster, we present the performance study and optimization of two solid earth physics applications, Seissol and Fall3D, on a long vector architecture RISC-V chip. We focus our optimizations on increasing the applications' vectorization and taking advantage of the machine's full vector length. The techniques used in our research include merging many small instances of linear algebra kernels to expose more data-level parallelism (batching), redefining data structures, and rewriting loops to facilitate the vectorization done by the compiler. We highlight the portability of our solutions, making them architecture-agnostic, which improved the performance among different machines. We present speedups ranging from 6× to 30× in the RISC-V prototype, and substantial speedups on an Intel supercomputer (Marenostrum4) and the NEC SX-Aurora architecture.

Context

European Centers of Excellence bring together researchers focused on shared scientific goals. One such center, ChEESe, is advancing solid earth physics and geohazard mitigation services related to earthquakes, volcanoes, and tsunamis, with ten European flagship codes like Exahype, Salvus, Ashee, Seissol, and Fall3D, using EuroHPC's most powerful supercomputers and emerging European architectures.

The Barcelona Supercomputing Center (BSC) is at the forefront of this co-design effort, testing these codes on EPAC-VEC, a novel RISC-V processor developed within the European Processor Initiative (EPI).

EPAC-VEC boasts a unique architecture composed by a scalar core designed by Semidynamics and the Vector Processing Unit developed by BSC, which can process up to 256 double-precision data elements per instruction, significantly speeding up certain computations. The EPI project also includes the development of an LLVM-based compiler optimized for C/C++/Fortran with support for SIMD parallelization and auto-vectorization.

Our contribution to the summit includes the vectorization and optimization of two applications from the ChEESe Center of Excellence, specifically Seissol and Fall3D, evaluating the performance of both applications and the EPAC-VEC architecture.

Seissol

The first application studied is SeisSol, a high-performance computational seismology software to simulate complex earthquake scenarios. It supports various rheologies (e.g. isotropic elastic, anisotropic elastic, acoustic), boundary conditions and dynamic rupture laws. SeisSol uses high-order discontinuous Galerkin discretization with an Arbitrary DERivative (ADER) local time stepping scheme on unstructured adaptive tetrahedral meshes.

Fall3D

The second code we study is Fall3D, an Eulerian model for atmospheric passive transport and deposition based on the advection–diffusion–sedimentation (ADS) equation. The application was developed for inert volcanic particles (tephra), but its capabilities have been extended by incorporating new features such as running ensemble forecasts and dealing with multiple atmospheric species (i.e. volcanic ash and gases, mineral dust, and radionuclides). Additionally, Fall3D is a multi-purpose model, since it can be used to compute both the airborne concentration (e.g. at flight levels) and the fallout deposit (ground accumulation).

Contribution to the summit

In our evaluation, we use the software development vehicle methodology [1] to study both solid earth applications on the EPAC-VEC prototype. This process is carried out in three steps. First, we compile and run the applications on commercial RISC-V platforms, to ensure that no dependencies forbid us from porting them to this architecture and to profile them on scalar machines. Then, we leverage a software emulator to study the vectorization done by the EPI compiler. Finally, we test the vectorized applications on a hardware implementation of the EPAC chip in an FPGAs,

*Corresponding author: pablo.vizcaino@bsc.es

providing cycle-accurate measurements.

This methodology gave us the following results:

Application profiling

Running Seissol on the commercial RISC-V boards shows that the execution time is divided between the integration of local cells and neighboring cells. These two functions are composed of many kernels performing Matrix-Matrix multiplications (DGEMM), which after further investigation, we found to be using very small matrices with only up to 20 elements per side.

For Fall3D, our profiling showed us that the application was divided into solving an equation in the X, Y, and Z dimensions, which took roughly the same amount of computation time. In this step, we also identified loops with potential vectorization inside these equation-solving functions.

Vectorization analysis

For Seissol, the small matrices limit the vectorization and constrain the full usage of the long vectors present in the EPAC-VEC architecture. We implemented a batched-matrix method that applies the application's kernels to multiple matrices simultaneously, operating more than one matrix in each vector. Using emulation tools, we confirmed the exploitation of the vector-length, but discovered long-latency strided memory operations required to load the multiple matrices. Given the small nature of these matrix multiplications, we completely unrolled the inner-most loops which helped mitigate this effect.

With Fall3D we found that the compiler only vectorized less than 1% of the instructions. Additionally, a function called in numerous points in the application was not being vectorized at all. After further investigation, we found that some high-level constructs of Fortran were hindering the vectorization, which we then transformed into vector-friendly loops. For other parts of the code, we also applied techniques to increase the vector-length and the number of vector instructions, such as loop collapsing and swapping.

Performance measuring

We ran Seissol in the EPAC-VEC prototype and compared the performance between the OpenBLAS library and our batched DGEMM kernels. For the smallest matrices, batching and vectorizing them granted a 30× speedup, down to 4× for the largest ones. Moreover, we noticed that unrolling the largest matrices introduced vector register spilling instructions, which we measured to take half of the execution time. Further work in Seissol includes removing or minimizing these spill instructions, even if it requires lowering the unroll.

In Fall3D we studied the effect of our code optimization on the EPAC-VEC prototype. While the vanilla autovectorized version of the code gave us a 1.25× speedup, after optimizing the code we reached a 6× speedup. Now, the limiting factor is the vector length, which remains suboptimal in some parts of the code due to the size of the input test. Further investigation on increasing the input size and its effect on the performance is required, but we expect to improve the vectorization speedup even more.

Performance portability

For both applications, we highlight that while the optimizations done to the code are centered around a better vectorization, they are not RISC-V specific and do not rely on unique compiler intrinsics of our architecture. To showcase the portability of our solutions, we compile and run the applications with our optimizations on other architectures.

The autovectorized batched kernels of Seissol running on the Marenostrum4 supercomputer (using Intel's AVX512 vectors) got up to a 13× speedup to the system's OpenBLAS implementation, showing that our optimization works with shorter vectors too.

In Fall3D, our changes provided a more moderate 1.23× speedup on Marenostrum4, with an absolute performance 4× as fast as the RISC-V prototype. We also evaluated the application on the NEC SX-Aurora architecture, which like the EPAC-VEC leverages long vectors capable of operating 256 double precision elements each. In this case, our same optimizations resulted in a 2.25× speedup when compared against the system's openBLAS library, and an absolute performance on-par with Intel's.

Conclusions

In this research poster, we showcase the study of two complex solid earth physics applications running on a long vector RISC-V prototype. Using the methodology and analysis tools of the Software Development Vehicles within the EPI project, we highlight the limitations of the applications that hinder the vectorization, and present various optimizations to circumvent them. Finally, we show that our optimizations remain portable across different architectures.

References

[1] Filippo Mantovani et al. "Software Development Vehicles to enable extended and early co-design: a RISC-V and HPC case of study". In: *International Conference on High Performance Computing*. Springer. 2023, pp. 526–537.