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

David Jurado (*)
david.jurado@bsc.es

Fabio Banchelli
fabio.banchelli@bsc.es

Filippo Mantovani
filippo.mantovani@bsc.es

Marta Garcia-Gasulla
marta.garcia@bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

(*) Corresponding author

RISC-V vector extension and scientific codes

European Processor Initiative to deliver a RISC-V based vector accelerator.
We provide a HW and SW ecosystem to enable co-design before the chip arrives.

Software Development Vehicles to Enable Extended and Early Co-design: A RISC-V and HPC Case of Study DOI: 10.1007/978-3-031-40843-4_39

We also develop an evaluation methodology to study potential optimizations
Our optimization techniques are CPU-agnostic and provide benefits in other architectures.

Short reasons for long vectors in HPC CPUs: a study based on RISC-V
DOI: 10.1145/3624062.3624231

We contribute to the porting and evaluation of scientific codes in RISC-V systems with a strong focus on long vector architectures.

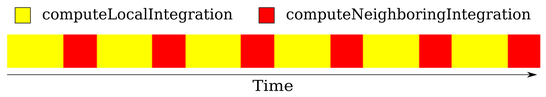
Batched DGEMMs for scientific codes on long vector architectures
DOI: 10.48550/arXiv.2501.06175

Exploiting long vectors with a CFD code: a co-design show case
DOI: 10.1109/IPDPS57955.2024.00047

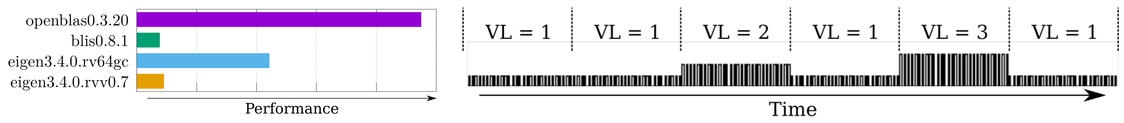
SeisSol

SeisSol Software package for simulating wave propagation and dynamic rupture based on the arbitrary high-order accurate derivative discontinuous Galerkin method.

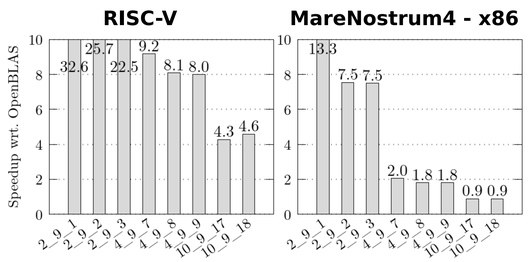
Structure Iterative pattern with two phases.
Both call DGEMM-based kernels.



BLAS libs Available BLAS libraries are either i) not vectorized at all; or ii) auto-vectorization is not able to leverage long vectors.

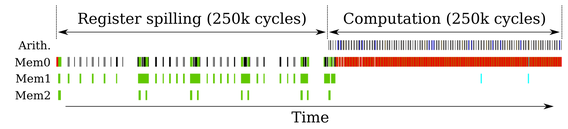


Batched DGEMMs Multiplication of small matrices does not expose enough work to take advantage of long-vectors -> Batch multiple DGEMMs at once.

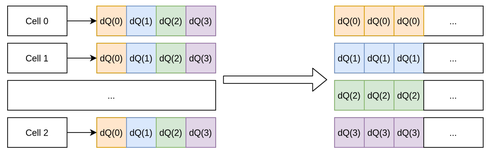


Batched DGEMM implementation in pure C
Code is architecture agnostic

RISC-V: always better than reference.
Able to leverage full vector length



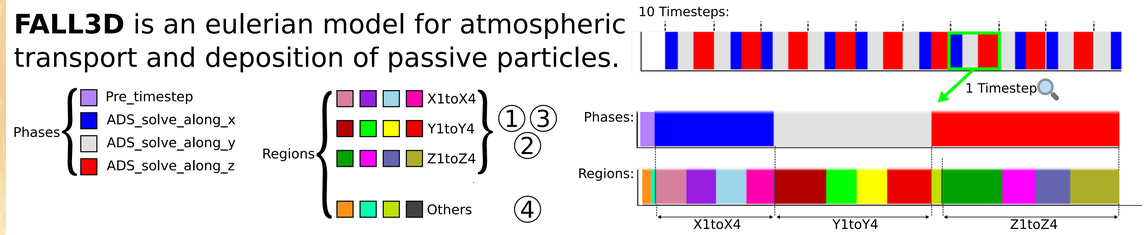
Performance of current batched DGEMM implementation is limited by register spilling.



Integration with SeisSol requires modifying data structures (AoS to SoA)

MiniFALL3D

FALL3D is an eulerian model for atmospheric transport and deposition of passive particles.



KT_RHS function

Fine grained execution
Called 88k times
Called by all ADS phases
Cumulative 57% of Timestep

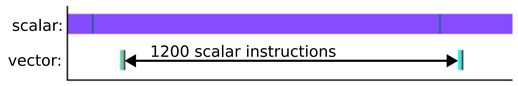
Not vector-friendly ⓪

```
do i = ips, ipe
  ! Do work...
  if (i.eq.ips) flux(1) = F_m+P_m
  if (i.eq.ips) flux(2) = F_p+P_p
end do
```

Vector-friendly ①

```
flux(1) = !...
do i = ips, ipe
  ! Do work...
end do
flux(2) = !...
```

Fortran constructs Compiler acts conservatively, it introduces extra scalar array copies.



array operations → do-loops ②
function → subroutine ③

Other Regions

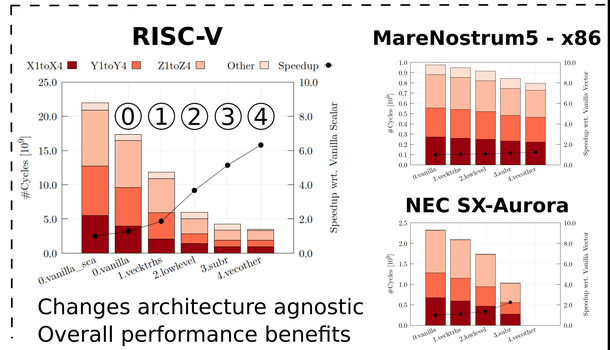
Loop collapsing, do-loops, and loop swap.

Not vectorized

```
1 CB%w(:,:,:) = (1.0_rp-stime)*my_w1(:,:,:) + stime*my_w2(:,:,:)
```

Vectorized (with VL=256) ④

```
1 ijksize = (my_kbe_1h-my_kbs_1h+1)
2   * (my_ipe-my_ips+1) * (my_jpe-my_jps+1)
3 ! 1D-to-3D pointer assignments
4 cb_w_1d(1:ijksize) => CB%w(:,:,:)
5 my_w1_1d(1:ijksize) => my_w1(:,:,:)
6 my_w2_1d(1:ijksize) => my_w2(:,:,:)
7 ! 1D do-loop
8 do i = 1, ijksize
9   cb_w_1d(i) = (1.0_rp-stime)*my_w1_1d(i) + stime*my_w2_1d(i)
10 done
```



Changes architecture agnostic
Overall performance benefits