Monte Cimone v2: Down the Road of RISC-V High-Performance Computers

Emanuele Venieri[1], Simone Manoni[1], Giacomo Madella[1], Federico Ficarelli[2], Daniele Gregori[3] , Daniele Cesarini[2], Luca Benini[1,4], Andrea Bartolini[1]

[1]University of Bologna, Italy, [2]CINECA, Italy, [3]E4 Company S.p.A., [4]ETH Zürich Switzerland

emanuele.venieri2@unibo.it

Outline

- RISC-V & HPC
- Monte Cimone
- Performance evaluation
- BLAS vectorization
- Results

RISC-V & HPC

- HPC platform requirements:
  - High core count + vector/matrix extensions
  - Large amount of system memory
  - PCIe support (accelerators, network cards, fast storage)
  - Software: OS, job schedulers, monitoring systems, packages and libraries

- RISC-V is mature in embedded domain
- How to foster RISC-V ISA uptake in HPC?

RISC-V HPC timeline

RVV 0.7.1

RVV 1.0

RVA23

· · · AME/IME

2020 2021 2022 2023 2024 2025

time

SiFive Freedom U740

E4 RV007 Server Blades

Monte Cimone v1

Milk-V Pioneer Box

E4 RSV10 dual socket server

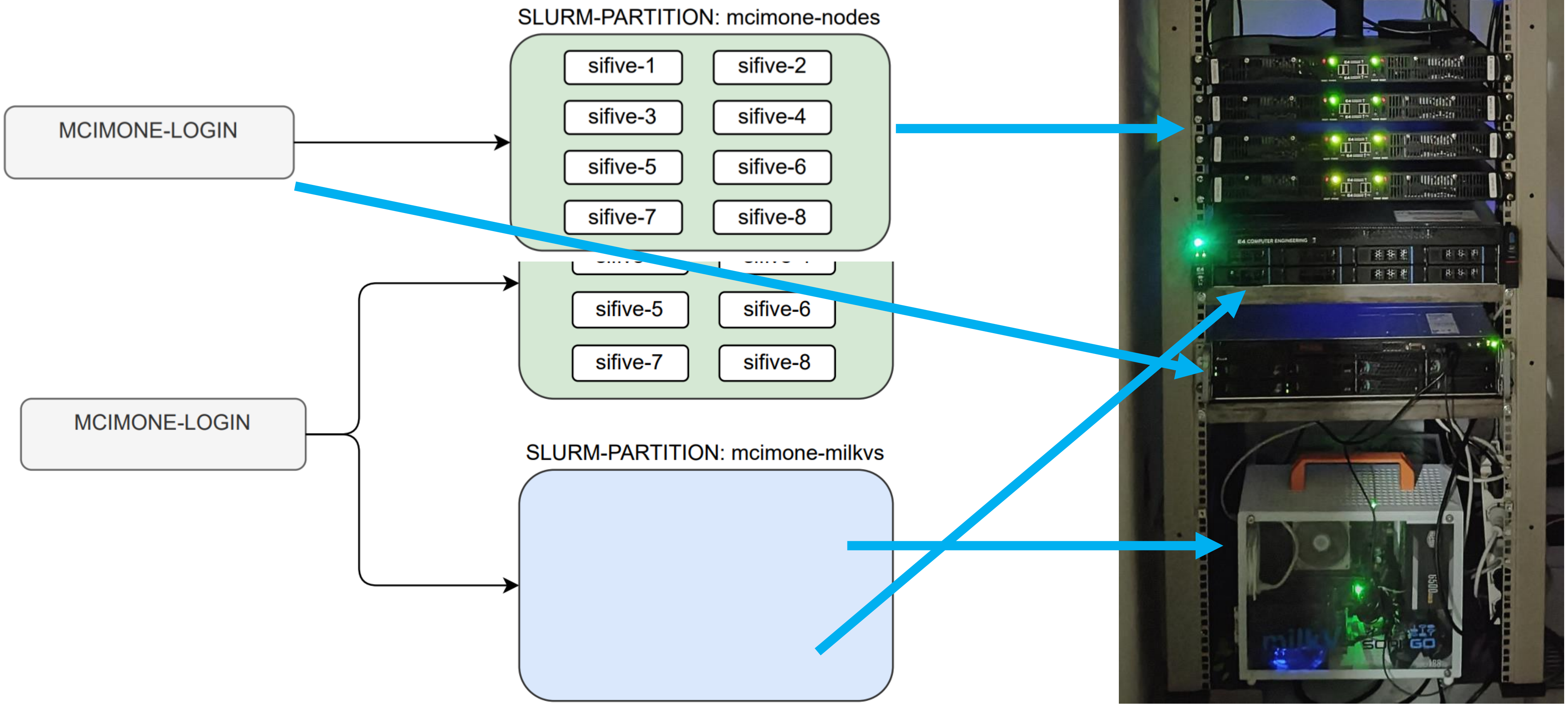
Monte Cimone v2

- 3-levels cache system
- ECC DDR4 memory
- 32 PCIe Gen 4 lanes

Monte Cimone v2 bring up

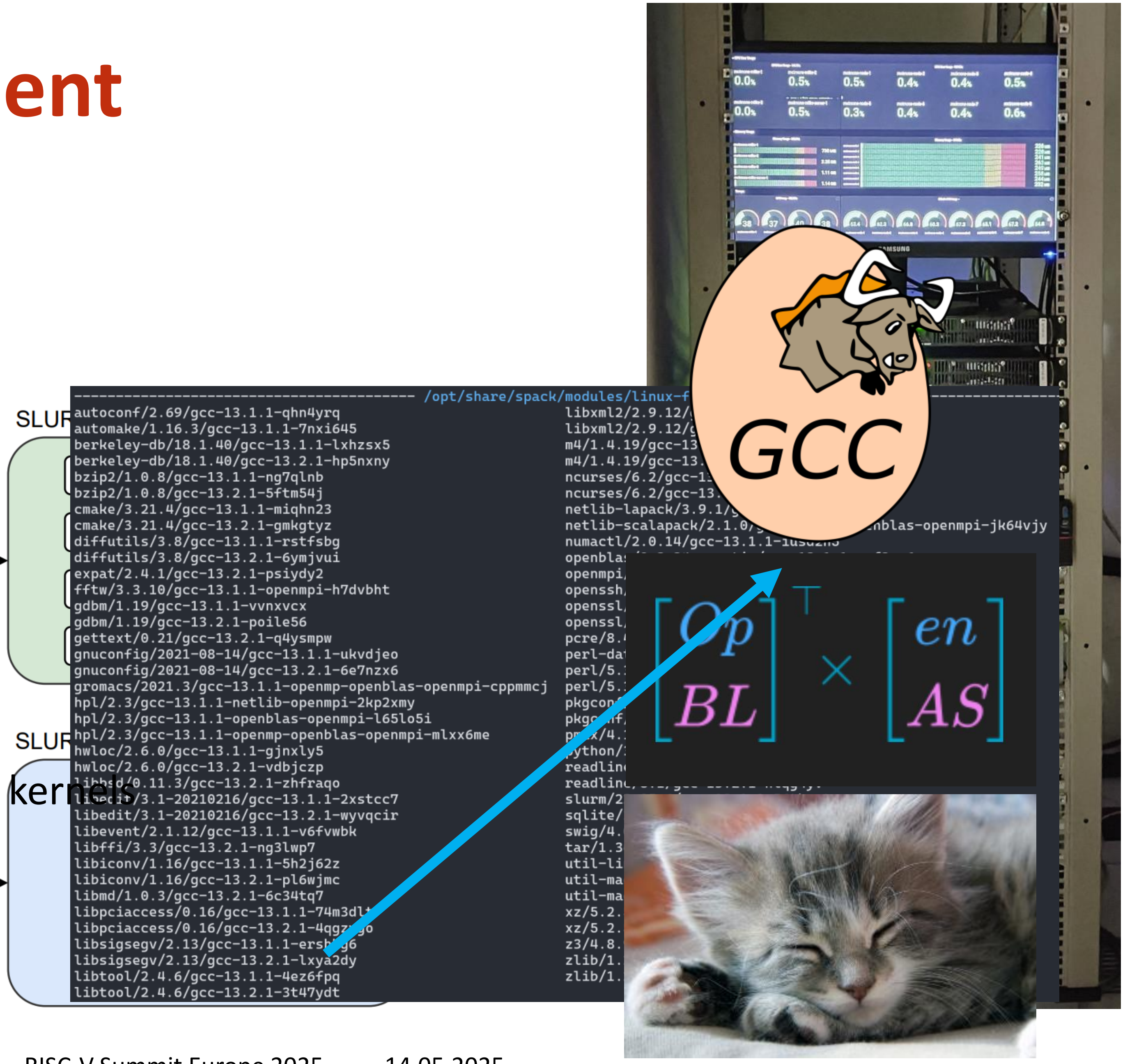
Starting from Monte Cimone v1

New SLURM partition

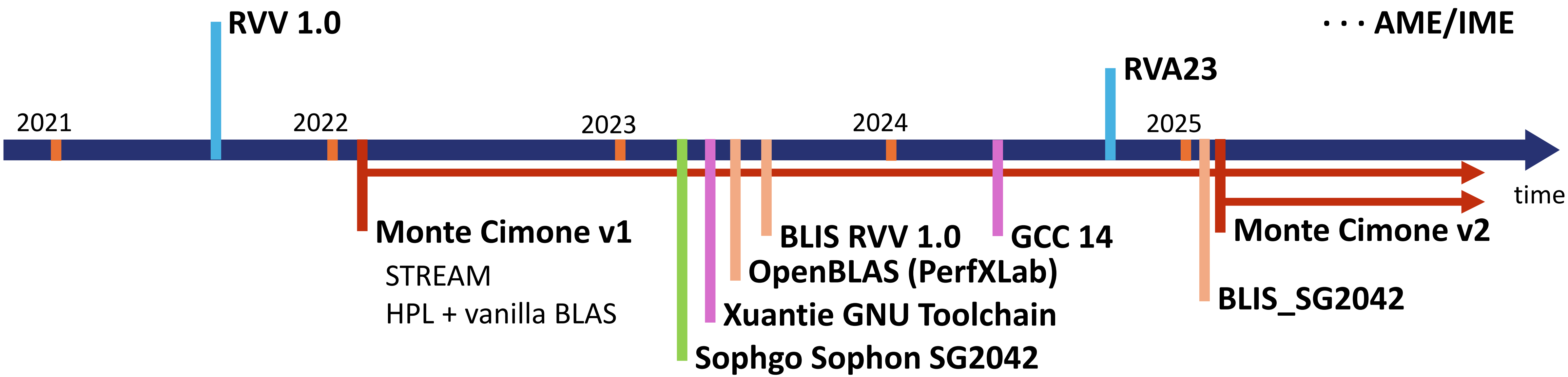


Cluster environment

- SLURM
- Examon
- Shared NFS
- Spack + new tools
  - New compilers:
    - Xuantie GNU Toolchain
    - GCC 14
  - New BLAS libraries:
    - OpenBLAS
    - BLIS with our optimized kernels



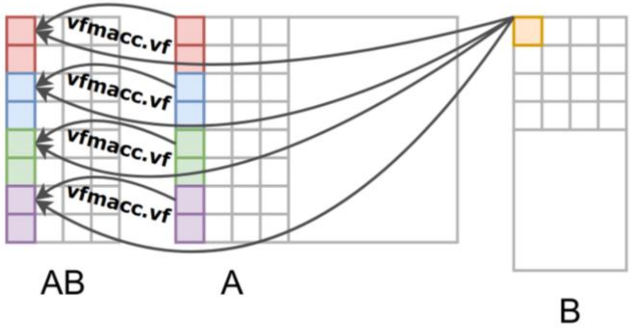
Monte Cimone v2 benchmark



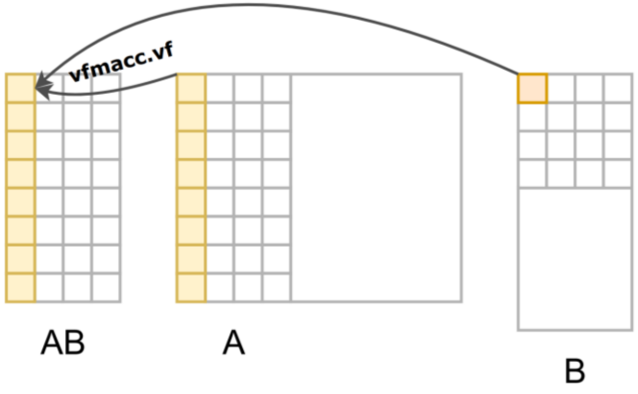
- **Performance assessment of new hardware**:
  - STREAM
  - HPL + OpenBLAS w/ Xuantie GNU Toolchain
- **BLIS library adapting and optimization:**
  - Developing an alternative for the community compilable with stock GCC 14
  - Improving found suboptimal micro-kernel

BLIS kernels optimization

- BLIS comes with RVV 1.0 generic micro-kernels written in assembly for DGEMM
- SG2042 has an RVV 0.7.1 vector unit, GCC 14 supports it as theadvector
- Steps of our BLIS micro-kernel optimization:
  - Porting from RVV 1.0 to 0.7.1
  - First tests: suboptimal implementation
  - Optimization using register grouping



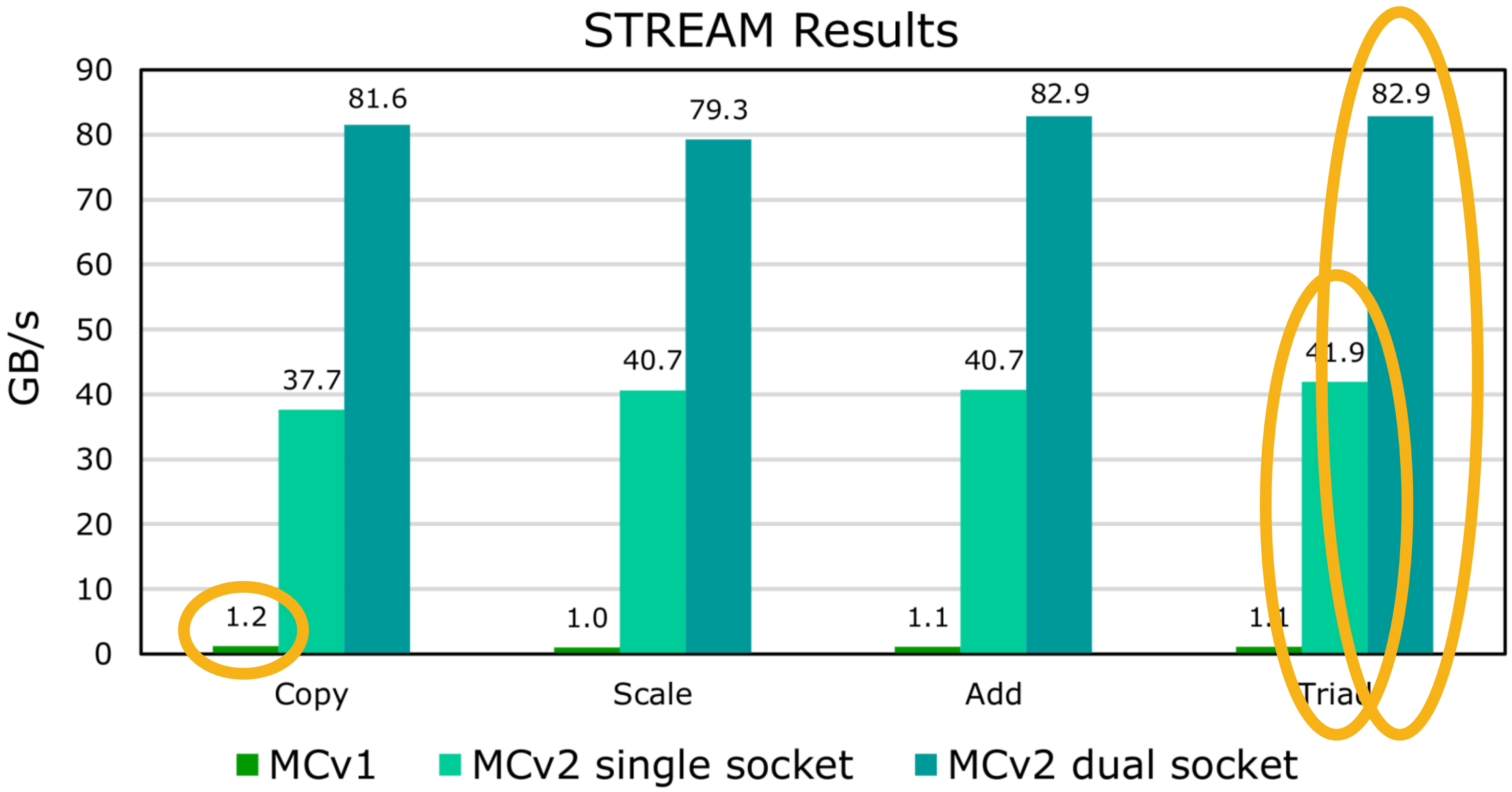
BLIS vectorization strategy



Our optimization strategy

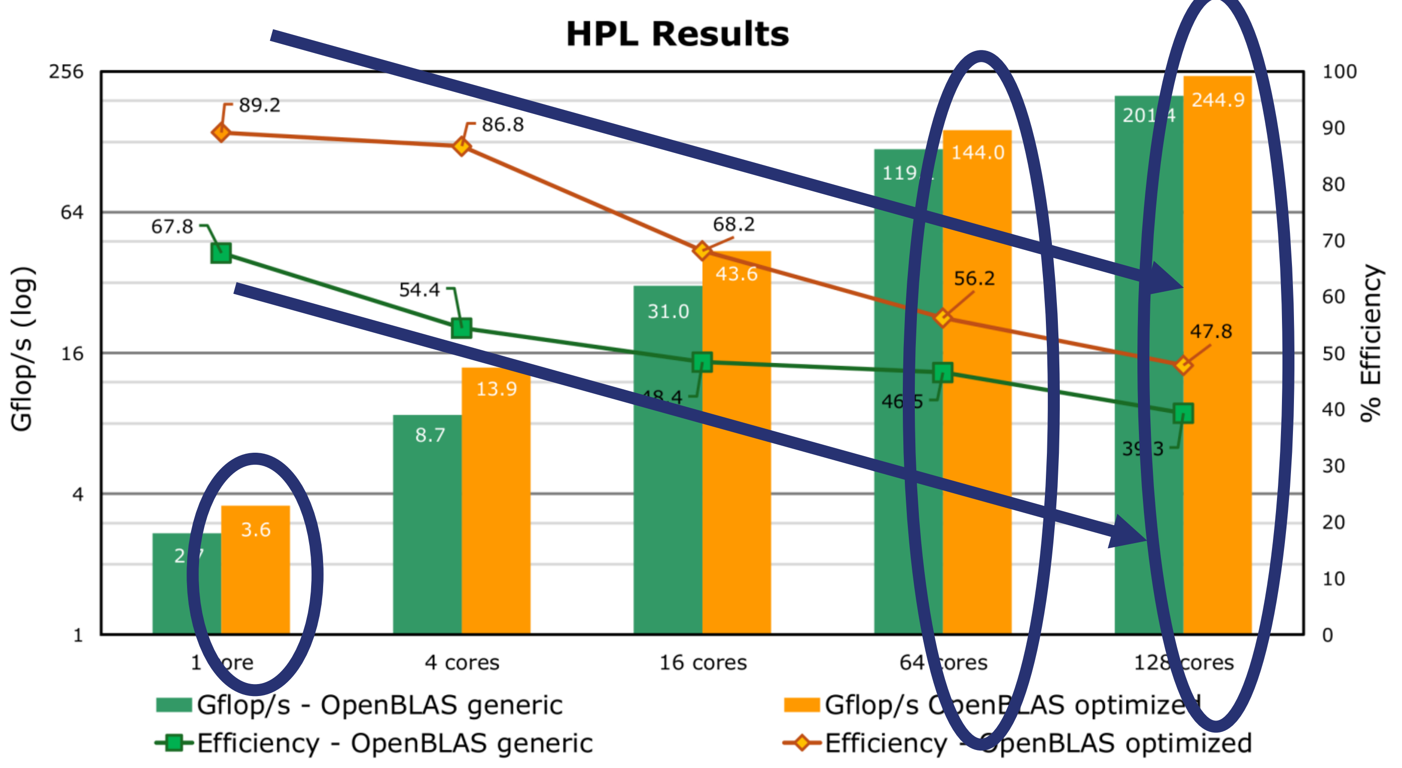
STREAM Results

- **Monte Cimone v1 node**:
  - Max: 1.2 GB/s
- **Milk-V Pioneer Box:**
  - 64 threads
  - 4 memory channels
  - Max: 41.9 GB/s
- **E4 RSV10 dual socket:**
  - 64 threads
  - 8 memory channels
  - Max: 82.9 GB/s



HPL Results

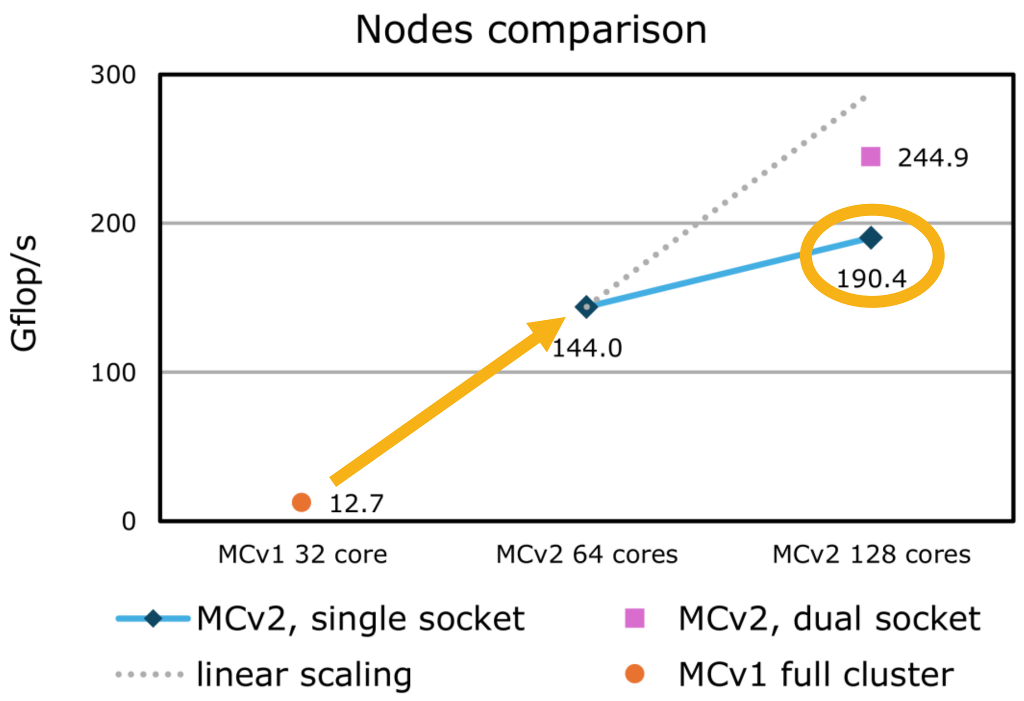
- **Reference results of SG2042:**
  - HPL + vanilla OpenBLAS
  - HPL + optimized OpenBLAS
- **Single nodes performance:**
  - 1 core:
    - 3.6 Gflop/s
  - 64 cores:
    - 144 Gflop/s
  - 128 cores:
    - 244.9 Gflop/s



- Decreasing efficiency with incremental core count
- Optimized OpenBLAS experience the same bottlenecks as vanilla ones
- The bottleneck was identified as the memory subsystem

HPL Results

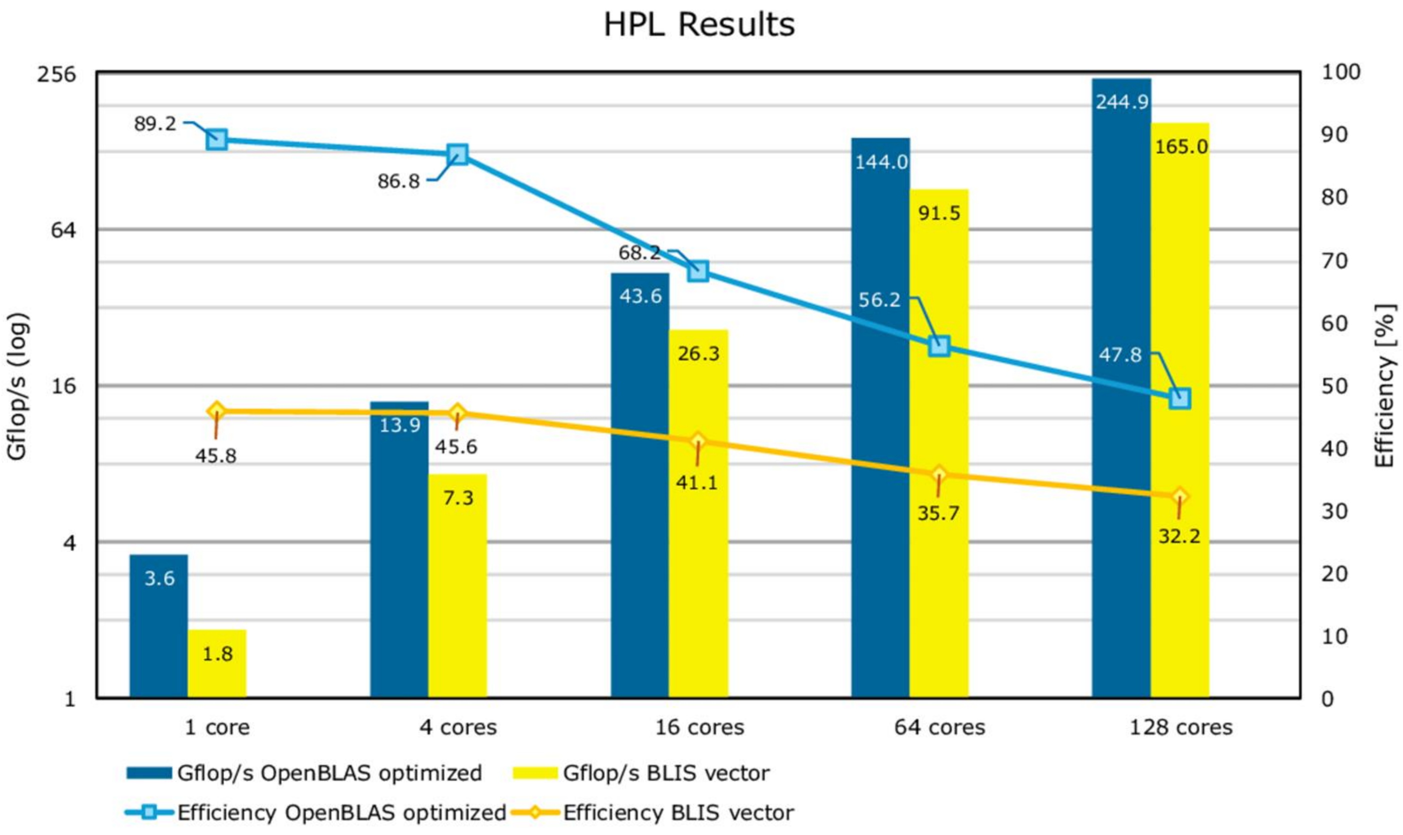
- **MCv1 full cluster**:
  - Linear scaling in multi node runs
  - HPL: 12.65 Gflop/s
- **MCv2 single node, single socket**:
  - 64 cores
  - HPL: 144.0 Gflop/s
- **MCv2 dual node, single socket**:
  - 128 cores
  - Ethernet 1Gbit/s network
  - HPL: 190.4 Gflop/s



- **MCv2 single node, dual socket**:
  - 128 cores
  - On board communication
  - HPL: 244.9 Gflop/s

HPL BLIS Results

- The available vectorized BLIS micro-kernel performed poorly
- Boost in performance post optimization:
  - 49% w.r.t. original vectorized BLIS (dual socket and 128 core runs)
  - Performance on par with OpenBLAS
  - Effectiveness of register grouping



Conclusions

- MCv1 vs. MCv2: rapidly evolving RISC-V ecosystem
  - Analysis of the first server style platforms
  - In two years a performance improvement obtained in average in eight years (Top500)
- BLIS library optimization:
  - Feasability of enhancing RISC-V software ecosystem
  - Contributed to the HPC community

This activity has been supported by the HE EU Graph-Massivizer (g.a. 101093202), DECICE (g.a. 101092582), and DARE (g.a. 101143421) projects, as well as the Italian Research Center on High Performance Computing, Big Data, and Quantum Computing.