Meet Monte Cimone: Exploring RISC-V High Performance Compute Clusters

Federico Ficarelli, Andrea Bartolini, Emanuele Parisi, Francesco Beneventi, Francesco Barchi, Daniele Gregori, Fabrizio Magugliani, Marco Cicala, Cosimo Gianfreda, Daniele Cesarini, Andrea Acquaviva, Luca Benini

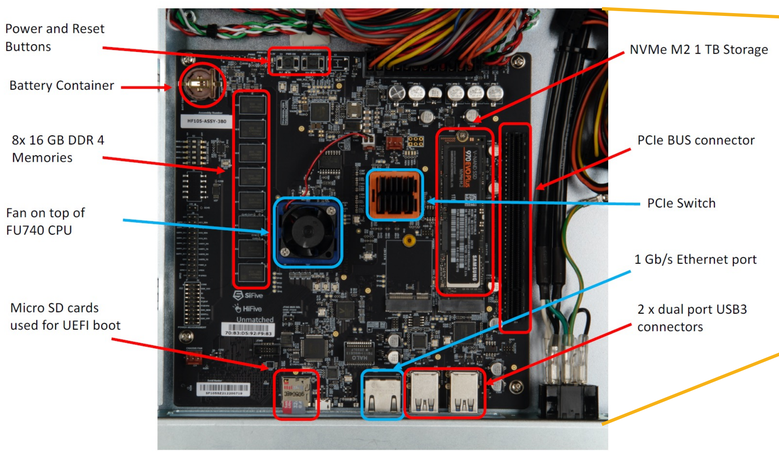
Mission: *Making High-performance RISC-V processors and accelerators ready for RISC-V-based HPC systems.*

Objective: *Monte Cimone, the first physical prototype and test-bed of a complete RISC-V (RV64) compute cluster, integrating not only all the key hardware elements besides processors, but also a* ***complete software environment for HPC****, as well as a full-featured* ***system monitoring infrastructure****. We demonstrate that it is possible to run* ***real-life HPC applications*** *on Monte Cimone today.*

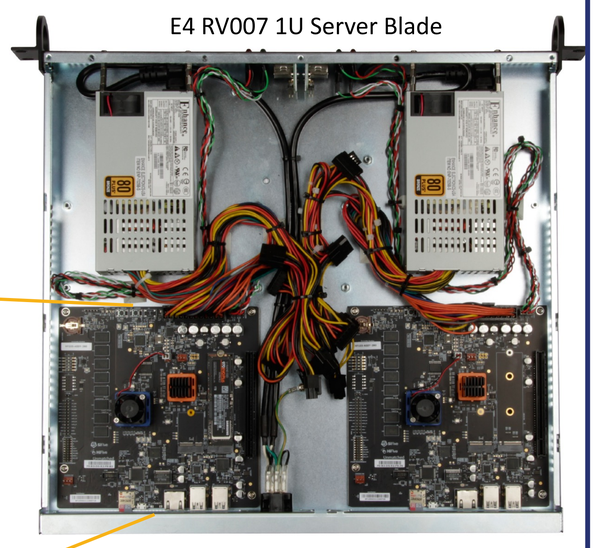
Monte Cimone Hardware Architecture:

We designed and set up **the first RISC-V-based cluster** containing **eight computing nodes** enclosed in **four computing blades**. Each computing node is based on the **U740 SoC from SiFive** and integrates:

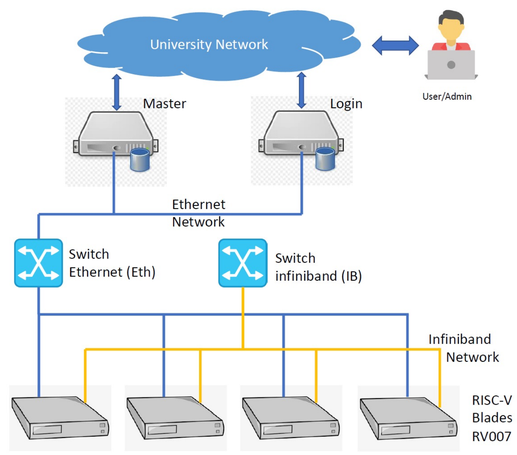
- **Four U74 RV64GCB** application cores, running up to **1.2 GHz**
- 16GB of **DDR4**
- 1 TB node-local **NVME storage**
- **PCIe expansion** cards



E4 RV007 1U Server Blade



Monte Cimone Software Stack:



The cluster is connected to a login node and master node running the job scheduler, network file system and system management software.

Monte Cimone: User-facing software stack

Package	Version
gcc	10.3.0
openmpi	4.1.1
openblas	0.3.18
fftw	3.3.10
netlib-lapack	3.9.1
netlib-scalapack	2.1.0
hpl	2.3
stream	5.10
quantumESPRESSO	6.8

We ported and assessed the maturity of a HPC software stack:

I. SLURM job scheduler, NAS filesystem, LDAP server, Spack package manager
II. Compilers toolchains, scientific and communication libraries,
III. A set of HPC benchmarks and applications,
IV. the ExaMon datacenter automation and monitoring framework.

Monte Cimone Efficiency vs ARMv8a, ppc64le:

We build the HPL benchmark and Stream benchmark following the same approach for the Monte Cimone cluster on two computing nodes, namely the Marconi100 (ppc64le, IBM Power9) compute node and the and Armida compute node (ARMv8a, Marvell ThunderX2).

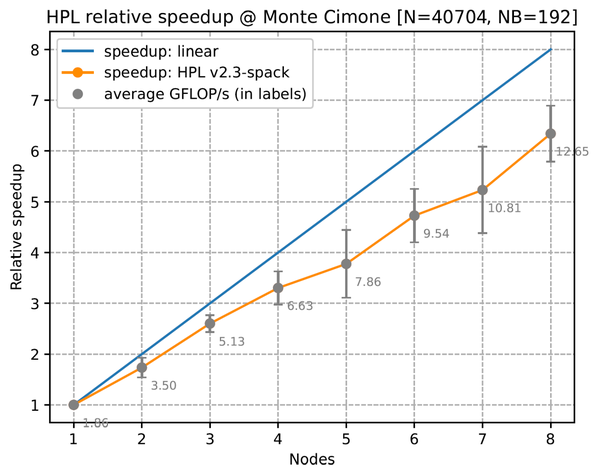
We compared the attained FPU utilization as a metric of efficiency against the one obtained by Monte Cimone while keeping the same benchmarking boundary conditions (e.g.: vanilla, unoptimized libraries and software stack deployed via Spack package manager).

System	HPL Efficiency [%FPU utilization]	Stream Efficiency [% Bandwidth Utilization]
Monte Cimone	46.5%	15.5%
Armida (ARMv8a, Marvell ThunderX2)	65.8%	63.2%
Marconi100 (ppc64le, IBM Power9)	59.7%	48.2%

HPL benchmark:

HPL peak theoretical value of 1.0 GFLOP/s/core, (from the micro-architecture specification)

- 4.0 GFLOP/s peak value for a single chip, the upstream HPL benchmark
- a sustained value of 1.86 ± 0.04 GFLOP/s on single node (on a N=40704 and NB=192 and a total runtime of 24105 ± 587s)



HPL multinode strong scaling w. 1Gb/s network :

- 39.5% of the entire machine's theoretical peak
- 85% of the extrapolated attainable peak in case of perfect linear scaling from the single-node case

quantumESPRESSO benchmark:

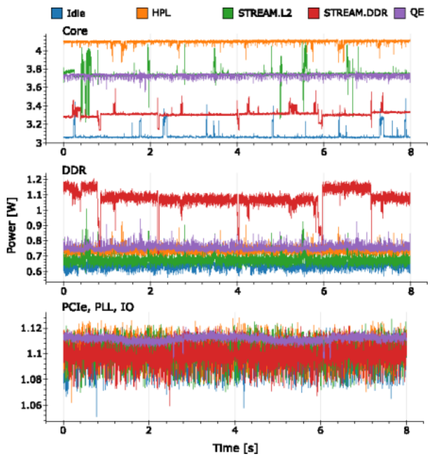
LAX test driver of the **quantumESPRESSO suite**, compiled with OpenMPI,

- performs a blocked (and optionally distributed) matrix diagonalization for a 512^2 input matrix (benchmark representative of the full-scale application workload).
- 1.44 ± 0.05 GFLOP/s (36% of the theoretical FPU efficiency) on a single node over a total test duration of 37.40 ±0.14 s

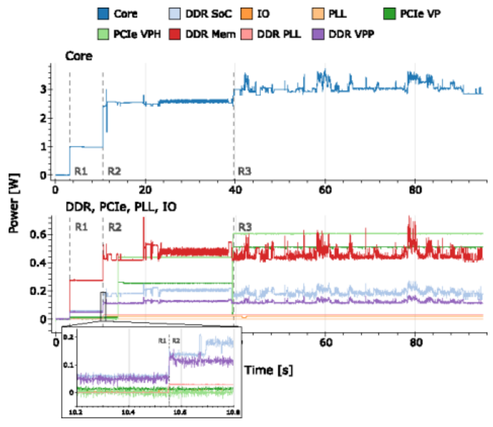
Power Characterization:

TABLE VI: Power consumption

Line	Idle		HPL		STREAM.L2		STREAM.DDR		QE		Boot R1	Boot R2
	[mW]	[%]	[mW]	[%]	[mW]	[%]	[mW]	[%]	[mW]	[%]	[mW]	[mW]
core	3075	64	4097	69	3714	68	3287	62	3825	67	984	2561
ddr_soc	139	3	177	3	170	3	232	4	176	3	59	197
io	20	0	20	0	20	0	20	0	20	0	5	20
pll	1	0	1	0	1	0	1	0	1	0	0	2
pcievp	521	11	527	9	524	10	522	10	530	9	12	231
pcievph	555	12	554	9	554	10	555	10	561	10	1	395
ddr_mem	404	8	440	7	401	7	592	11	434	8	275	467
ddr_pll	28	1	28	1	28	1	28	1	28	1	0	29
ddr_vpp	67	1	90	2	73	1	98	2	95	2	49	122
Total	4810	100	5935	100	5486	100	5336	100	5670	100	1385	4024



Snapshot of the power consumption of the core (top), the DDR (middle) and the PCIe, PLL and IO subsystems (bottom). The traces are obtained observing power consumption for 8 seconds during benchmark run and averaging raw data using 1 ms windows.



Power consumption for the Core (top), DDR, PCIe, PLL and IO (bottom) subsystems during system boot. Boot phases: power-on (R1), bootloader (R2), O.S. boot (R3). The Figure also shows the detail of PLL activation

.

We characterized the power consumption of various applications executed on Monte Cimone:

- **Idle**: 4.81W (64% of core power, 13% related to DDR and 23% of related to PCI subsystem)
- **HPL**: 5.935W (69% of core power, 14% related to DDR and 18% related to PCI subsystem)

During the boot process we measured for the **core complex**:

- 0.981W of leakage only power (32% of the Idle power)
- 0.514W of power consumed by the operating system during idle (17% of the Idle power)
- 1.577W of dynamic and clock tree power (51% of the core idle power).

STREAM benchmark:

TABLE V: STREAM, 4 threads

Test	STREAM.DDR 1945.5 MiB [MB/s]	STREAM.L2 1.1 MiB [MB/s]
copy	1206 ± 3.26	7079 ± 2.11
scale	1025 ± 4.94	3558 ± 3.72
add	1124 ± 4.93	4380 ± 3.72
triad	1122 ± 5.63	4365 ± 3.56

SiFive U740 RISC-V SoC peak DDR bandwidth 7760 MB/s. Possible causes to be investigate:

1. L2 prefetcher capable of tracking up to eight streams, then why it is not hiding the DDR latency?
2. Overall data size used by STREAM is currently limited by the RISC-V code model. The *medany* code model requires that every linked symbol resides within a ± 2GiB range from the pc register. Upstream STREAM benchmark uses statically-sized data arrays in a single translation unit preventing the linker to perform relaxed relocations, their overall size cannot exceed 2 GiB.
3. The upstream GCC 10.3.0 toolchain isn't capable of emitting the Zba and Zbb RISC-V bit manipulation standard extensions nor the underlying GNU assembler (shipped with GNU Binutils 2.36.1). Experimentally supported on GCC 12 and Binutils 2.37.x.

We acknowledge the European Processor Initiative (EPI) SGA2 (g.a.101036168), EuroHPC EU PILOT project (g.a. 101034126), EU Pilot for exascale EuroHPC EUPEX (g. a. 101033975), EU H2020-JTI-EuroHPC-2019-1 project REGALE (g.a. 956560)