Cluster of emerging technology: evaluation of a production HPC system based on A64FX

Fabio Banchelli
fabio.banchelli@bsc.es

Kilian Peiro
kilian.peiro@bsc.es

Guillem Ramirez-Gargallo
guillem.ramirez@bsc.es

Joan Vinyals
joan.vinyals@bsc.es

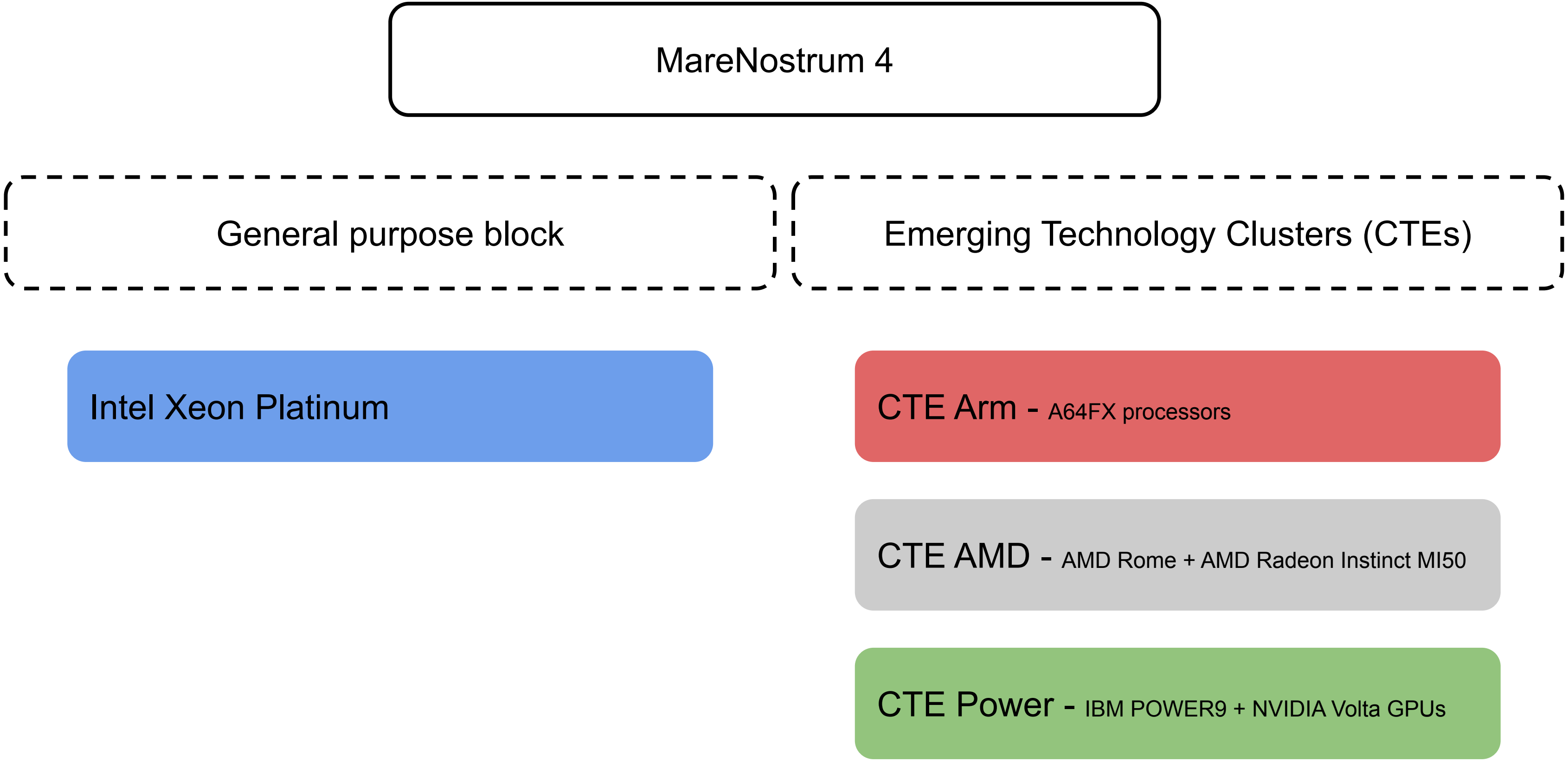
David Vicente
david.vicente@bsc.es

Marta Garcia-Gasulla
marta.garcia@bsc.es

Filippo Mantovani
filippo.mantovani@bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Emerging Technology Clusters deployed at BSC



MareNostrum 4

General purpose block

Emerging Technology Clusters (CTEs)

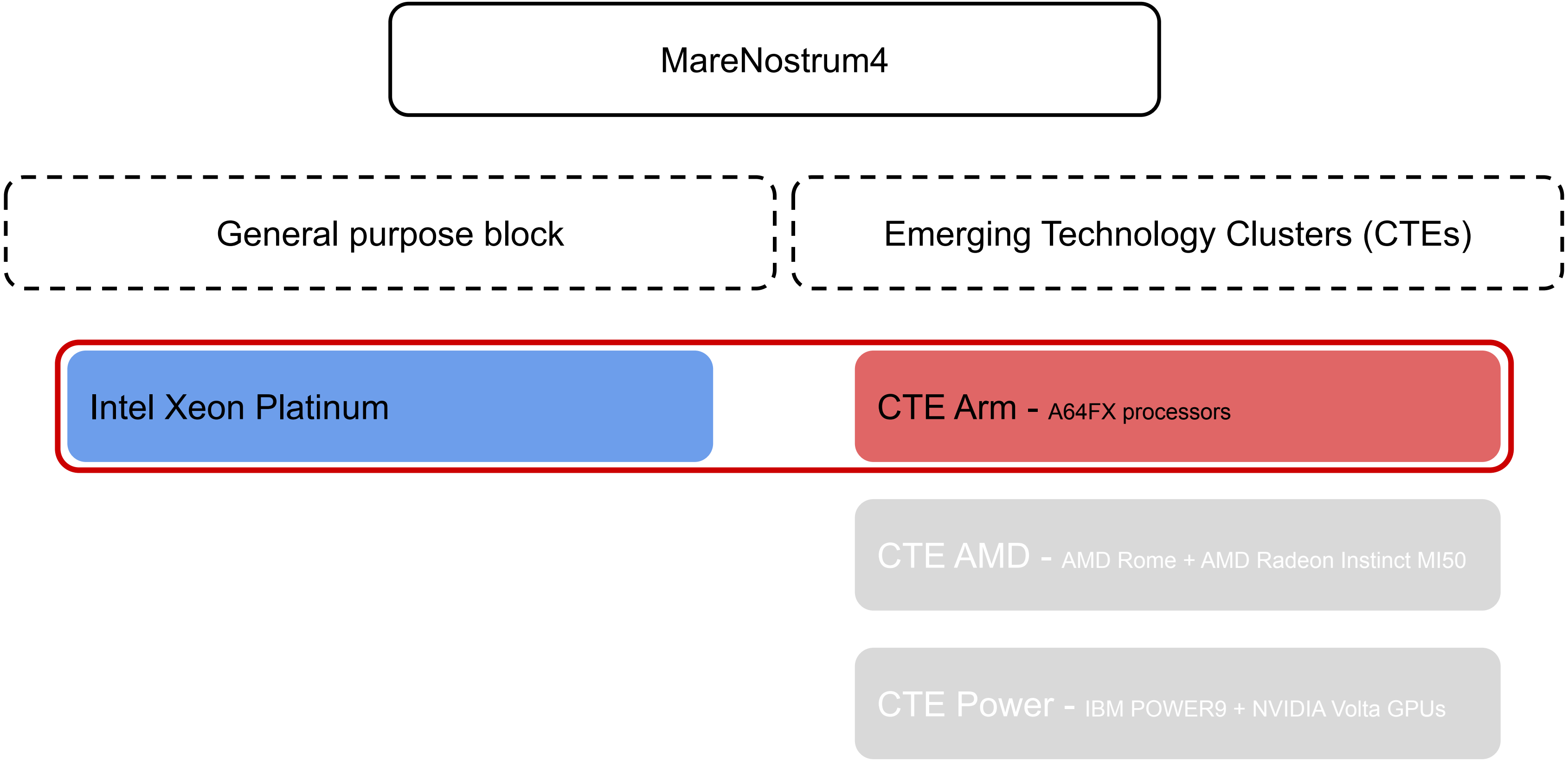
Intel Xeon Platinum

CTE Arm - A64FX processors

CTE AMD - AMD Rome + AMD Radeon Instinct MI50

CTE Power - IBM POWER9 + NVIDIA Volta GPUs

Machine under study: CTE-Arm



MareNostrum4

General purpose block

Emerging Technology Clusters (CTEs)

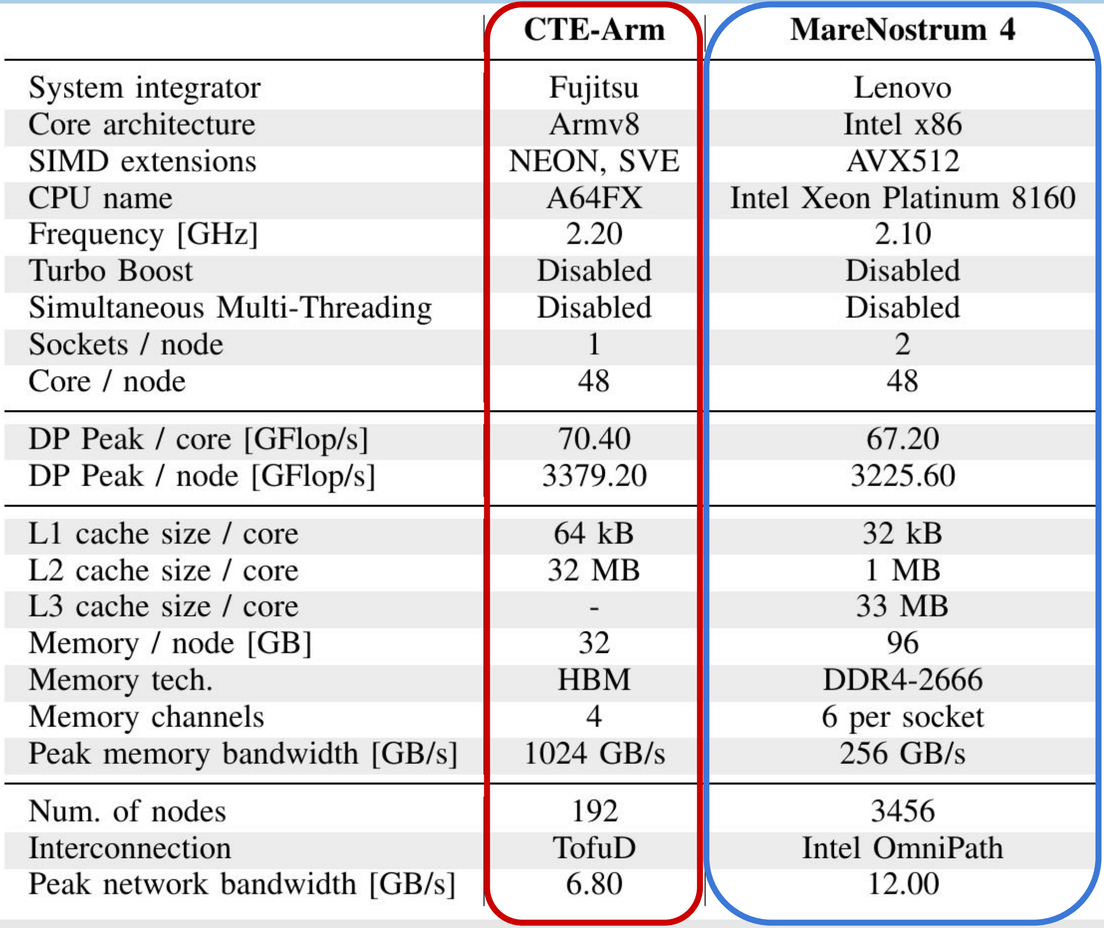
Intel Xeon Platinum

CTE Arm - A64FX processors

CTE AMD - AMD Rome + AMD Radeon Instinct MI50

CTE Power - IBM POWER9 + NVIDIA Volta GPUs

Hardware specification



	CTE-Arm	MareNostrum 4
System integrator	Fujitsu	Lenovo
Core architecture	Armv8	Intel x86
SIMD extensions	NEON, SVE	AVX512
CPU name	A64FX	Intel Xeon Platinum 8160
Frequency [GHz]	2.20	2.10
Turbo Boost	Disabled	Disabled
Simultaneous Multi-Threading	Disabled	Disabled
Sockets / node	1	2
Core / node	48	48
DP Peak / core [GFlop/s]	70.40	67.20
DP Peak / node [GFlop/s]	3379.20	3225.60
L1 cache size / core	64 kB	32 kB
L2 cache size / core	32 MB	1 MB
L3 cache size / core	-	33 MB
Memory / node [GB]	32	96
Memory tech.	HBM	DDR4-2666
Memory channels	4	6 per socket
Peak memory bandwidth [GB/s]	1024 GB/s	256 GB/s
Num. of nodes	192	3456
Interconnection	TofuD	Intel OmniPath
Peak network bandwidth [GB/s]	6.80	12.00

Scope of our work

Test if CTE-Arm matches specifications

- Run simple codes that stress one specific aspect of the system at a time
- Repeat tests across all nodes of the system

Emulate end-user experience

- Run complex scientific applications: Alya, NEMO, Gromacs, OpenIFS, and WRF
- Try to compile and run codes “as is”

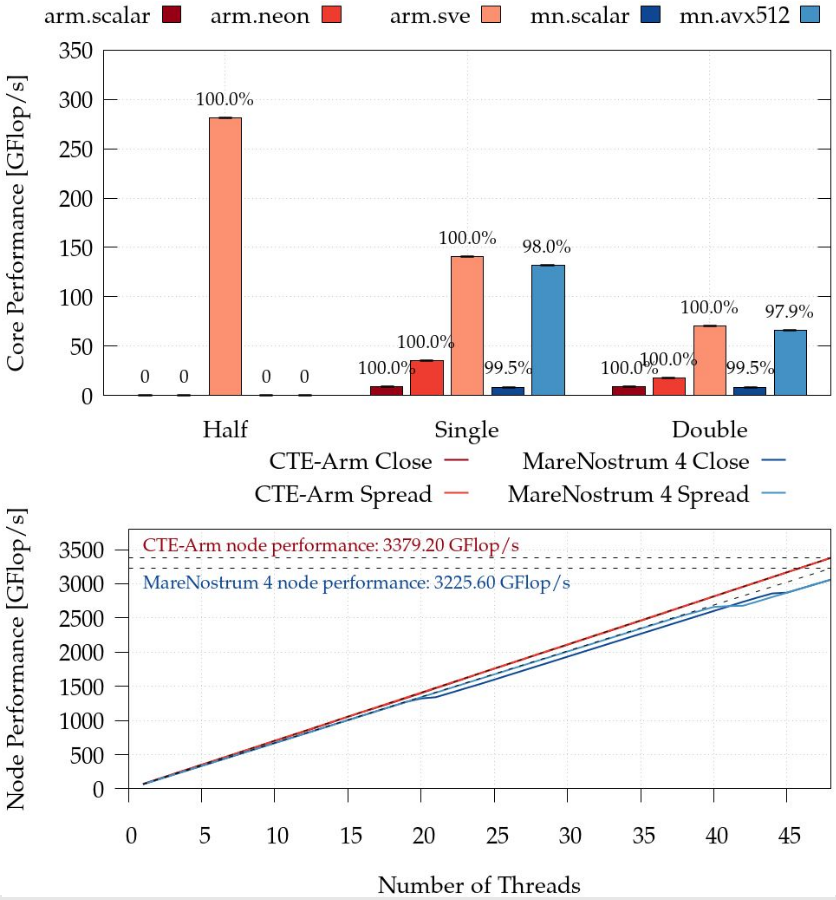
Micro-Benchmarks

Floating-Point Throughput

Custom kernel that performs

Fused-Multiply-Add instructions back-to-back

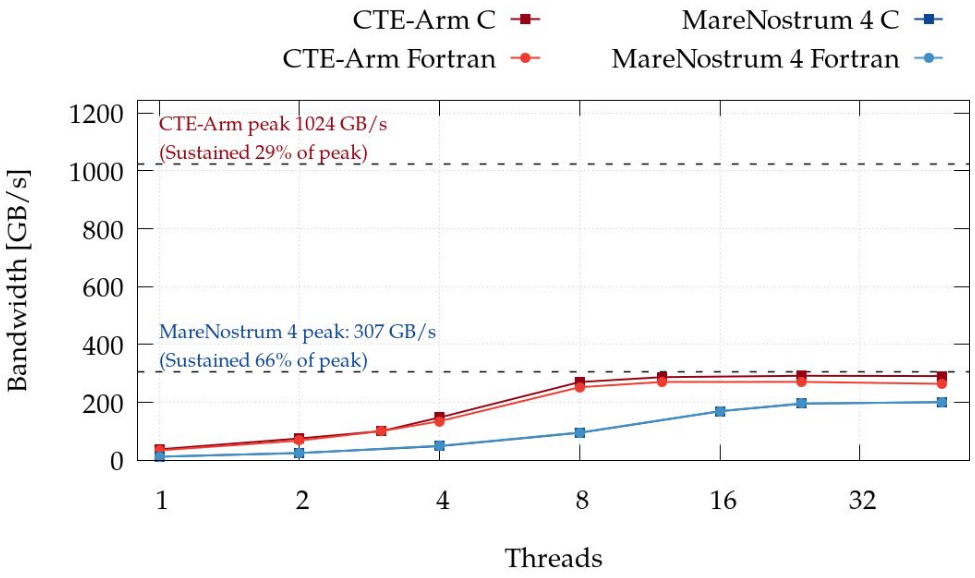
- Measured performance matches theoretical peak
- Performance degrades in MN4 when adding more threads
- We verified our measurements are consistent across all nodes of CTE-Arm



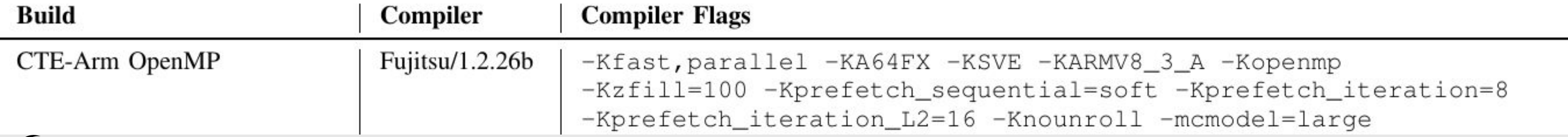
Memory Bandwidth

STREAM benchmark using OpenMP

- In CTE-Arm, we measure 29% of the theoretical peak
- We tried different combinations of compiler flags with no improvements
- Behavior is consistent for all kernels of STREAM



STREAM - Triad

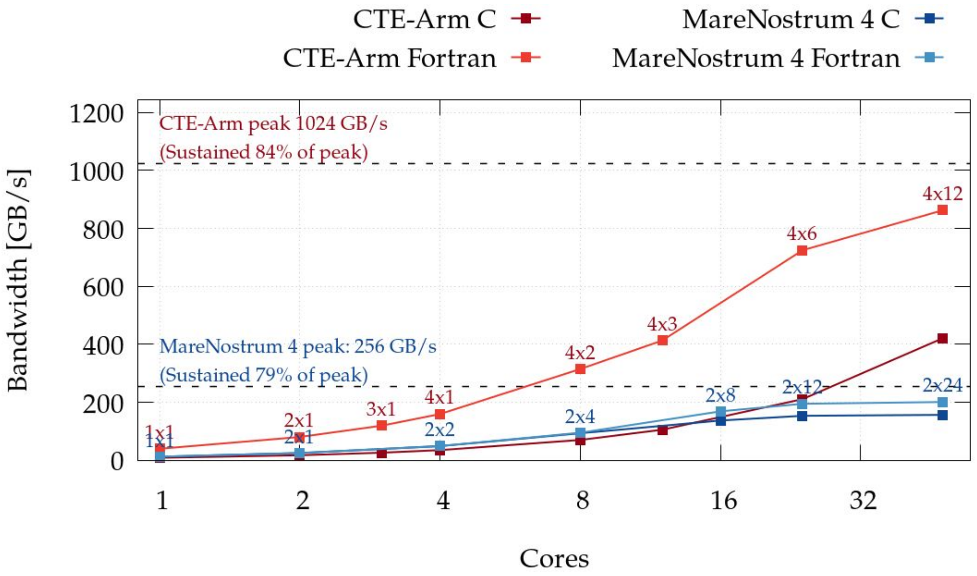


Build	Compiler	Compiler Flags
CTE-Arm OpenMP	Fujitsu/1.2.26b	`-Kfast,parallel -KA64FX -KSVE -KARMV8_3_A -Kopenmp -Kzfill=100 -Kprefetch_sequential=soft -Kprefetch_iteration=8 -Kprefetch_iteration_L2=16 -Knounroll -mcmodel=large`

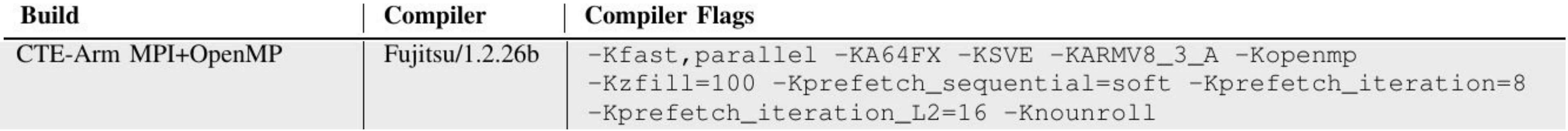
Memory Bandwidth

STREAM benchmark using MPI + OpenMP

- Processes pinned to different Core Memory Groups (CMGs)
- Noticeable bandwidth difference between C and Fortran codes
- The Fortran version reaches 84% of the theoretical peak



STREAM - Triad

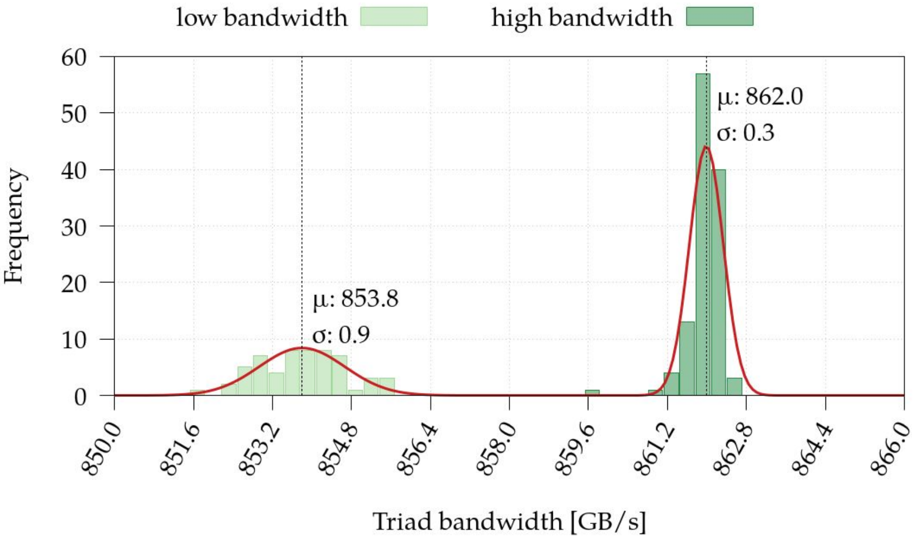


Build	Compiler	Compiler Flags
CTE-Arm MPI+OpenMP	Fujitsu/1.2.26b	`-Kfast,parallel -KA64FX -KSVE -KARMV8_3_A -Kopenmp -Kzfill=100 -Kprefetch_sequential=soft -Kprefetch_iteration=8 -Kprefetch_iteration_L2=16 -Knounroll`

Memory Bandwidth

STREAM benchmark across nodes

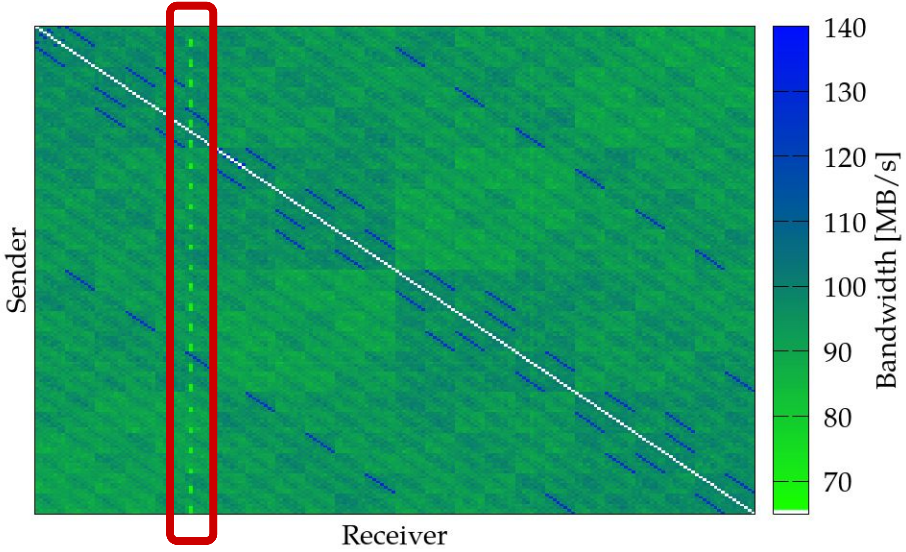
- We repeated our STREAM study on all node of the CTE-Arm cluster
- We observe that there are “fast” and “slow” nodes
- Consistent difference in bandwidth of ~10GB/s (2%)



Network Bandwidth

MPI Ping-pong between nodes

- For messages above 16 KiB, we measure 93% of theoretical peak
- Network topology has a noticeable impact on network bandwidth
- Job scheduler allocates close nodes to reduce this impact
- We detected a “slow” receiver node

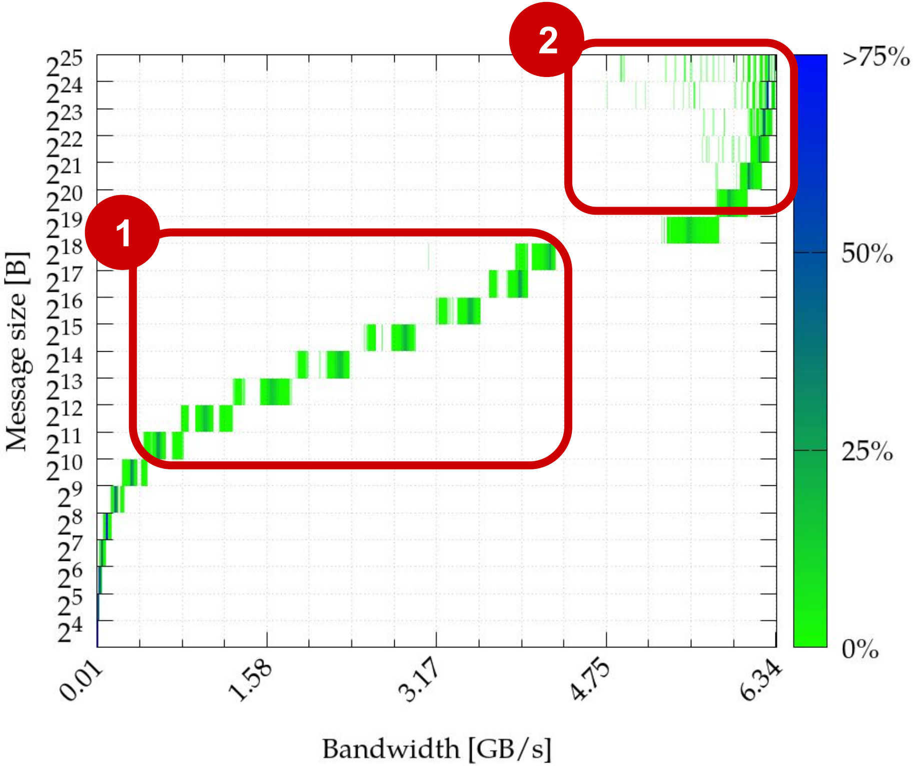


Message size 256 B

Network Bandwidth

- Histogram of network bandwidth across node pairs
  - Light green → Low occurrences
  - Dark blue → High occurrences

1. Bimodal distribution for messages between 1 KiB and 256 KiB
2. Messages bigger than 1 MiB have more variability

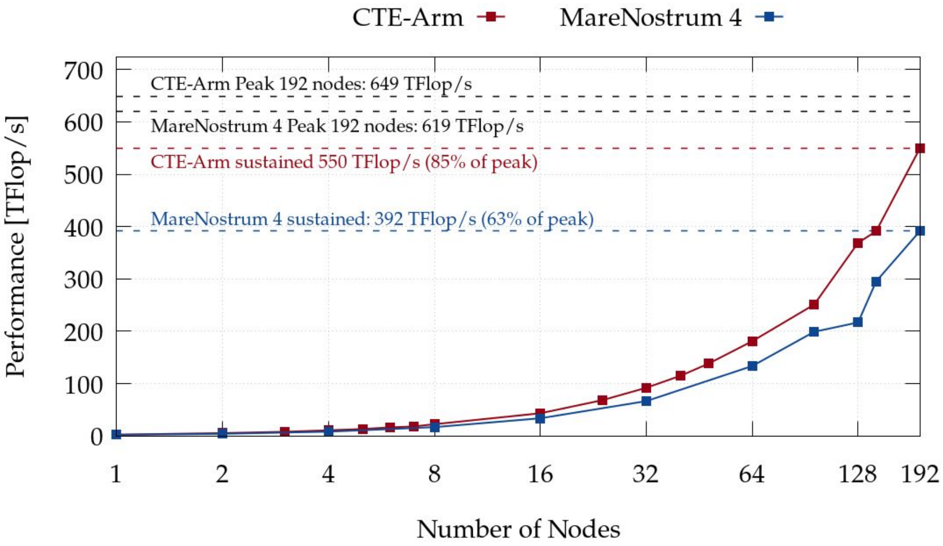


HPC Benchmarks

Linpack

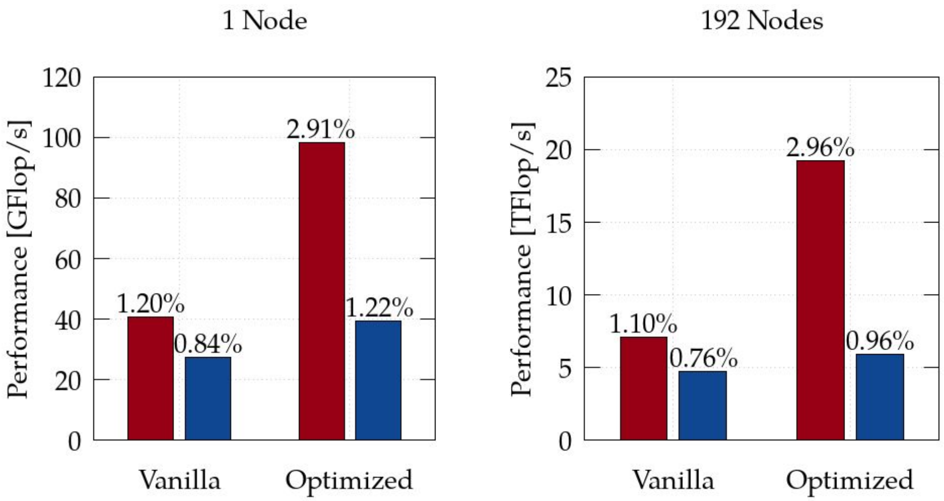
- Vendor provided binaries
- CTE-Arm → 4 MPI ranks mapped to different CMGs
- MareNostrum4 → 1 MPI rank per node

- We observe a higher performance of CTE-Arm compared to MareNostrum4
- CTE-Arm also achieves a higher percentage of the peak



HPCG

- Vanilla → compiled “as-is” from the HPCG repository
- Optimized → vendor provided binary
- CTE-Arm achieves higher efficiency than MareNostrum4
- With 192 nodes, CTE-Arm reaches 2.96% efficiency, close to the 3.62% of Fugaku in the Top500 of Nov. 2020



Scientific Applications

Software environment in CTE-Arm

Compilers

- Fujitsu Compiler 1.2.26.b
- GNU 8.3.1 (SVE enabled by Fujitsu)
- GNU 10.2.0 (SVE enabled upstream)
- Arm compiler 20.3

- None of the applications could be compiled with Fujitsu Compiler
- Used GNU 8.3.1 (SVE enabled by Fujitsu)

MPI library

- Custom OpenMPI from Fujitsu

- Only installed custom MPI library supports TofuD
- Using custom OpenMPI + GNU “not out of the box”

Software environment in CTE-Arm

Fujitsu Compiler

- Alya
  - Prohibitive compilation times
- NEMO
  - Compilation errors
- Gromacs
  - CMake and compilation errors
- OpenIFS
  - Required modifications to the code in order to compile
  - Runtime errors
- WRF
  - Compilation errors

Software environment in CTE-Arm

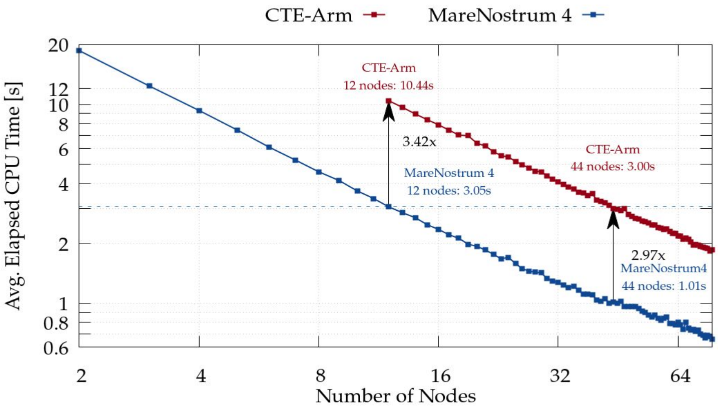
Fujitsu Compiler

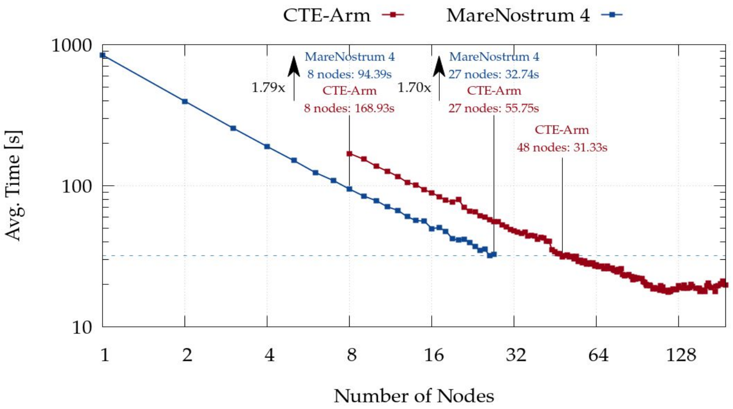
- Alya
  - Prohibitive compilation times
- NEMO
  - Compilation errors
- Gromacs
  - CMake and compilation errors
- OpenIFS
  - Required modifications to the code in order to compile
  - Runtime errors
- WRF
  - Compilation errors

In modern clusters, one software toolchain is not enough

Performance Results

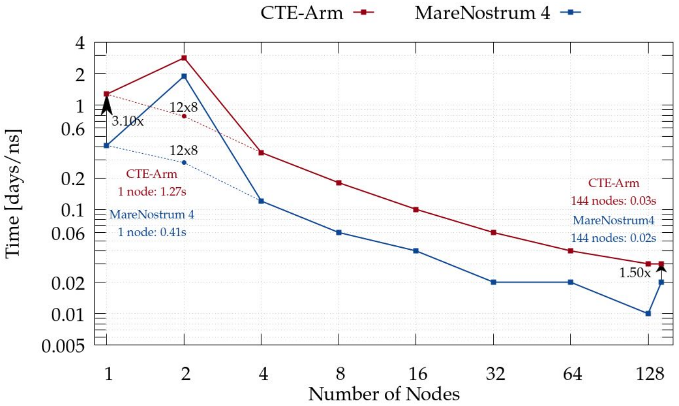
Alya

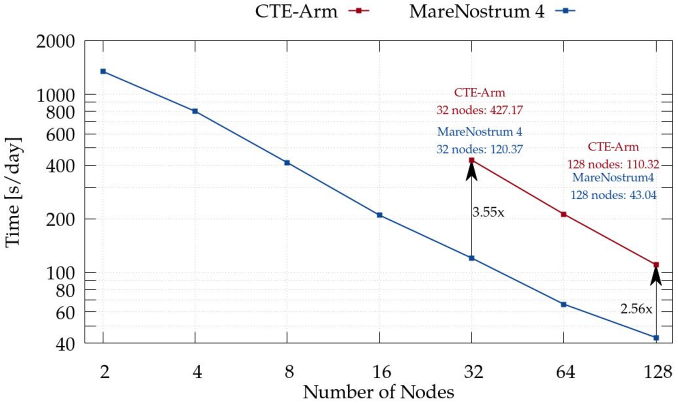




NEMO

Gromacs

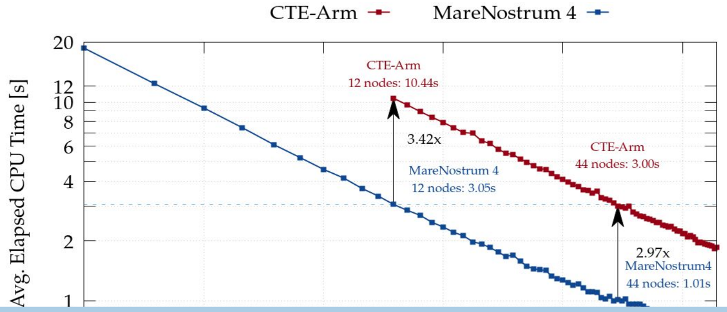




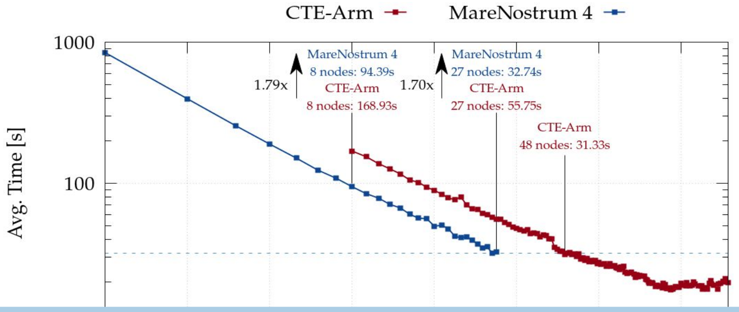
OpenIFS

Performance Results

Alya

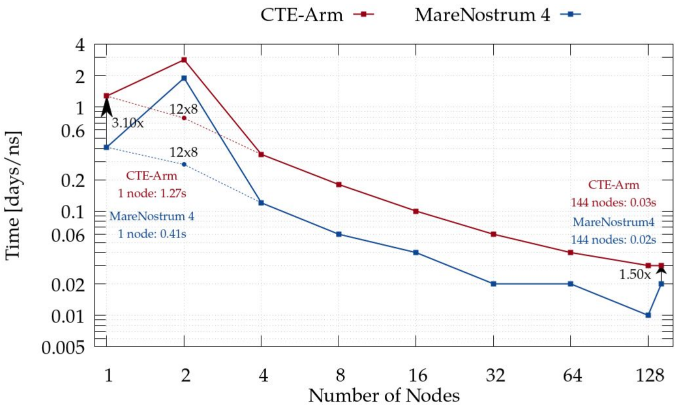


NEMO

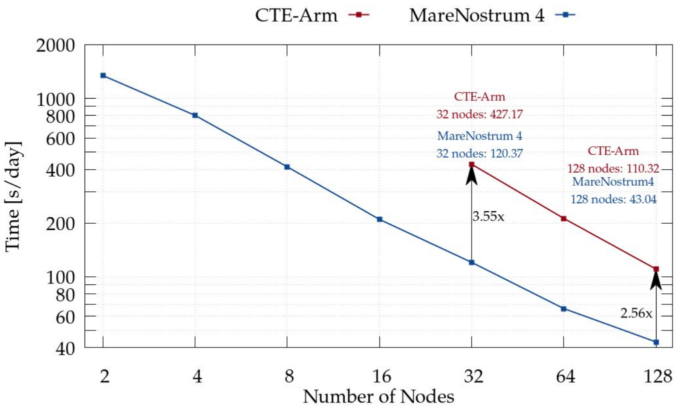


If we compile and run "out-of-the-box", CTE-Arm is between 1.50x and 3.42x slower than MareNostrum 4

Gromacs

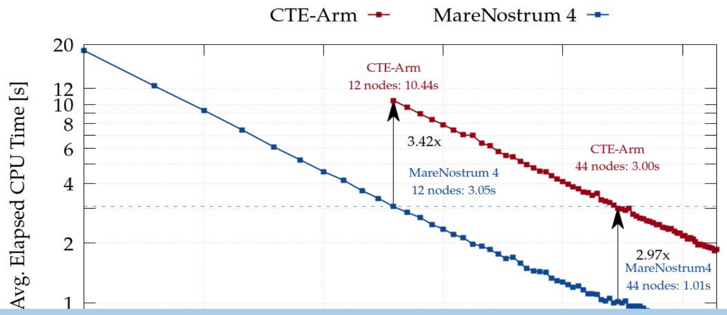


OpenIFS

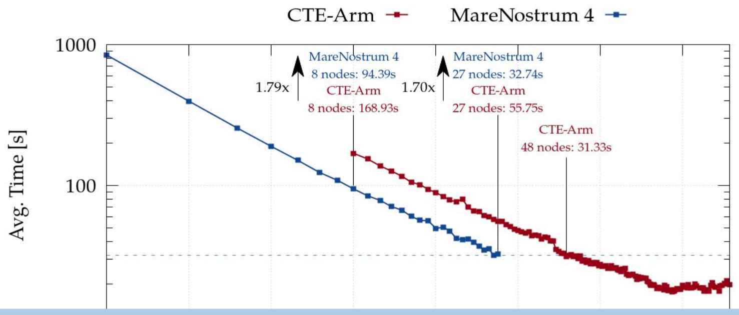


Performance Results

Alya



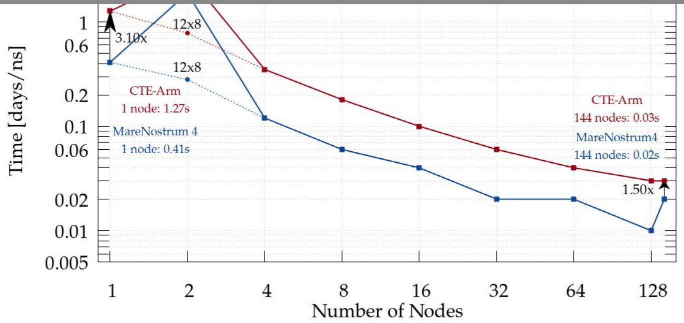
NEMO



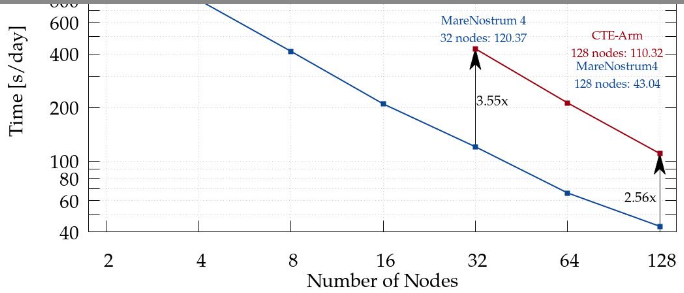
If we compile and run “out-of-the-box”, CTE-Arm is between 1.50x and 3.42x slower than MareNostrum 4

We know that the performance is there, we just need a toolchain that takes advantage of it

Gromacs



OpenIFS



Conclusions

Summary

- Simple kernels like micro-benchmarks reach close to peak performance

- Classical benchmarks (HPL & HPCG) yield good performance and efficiency

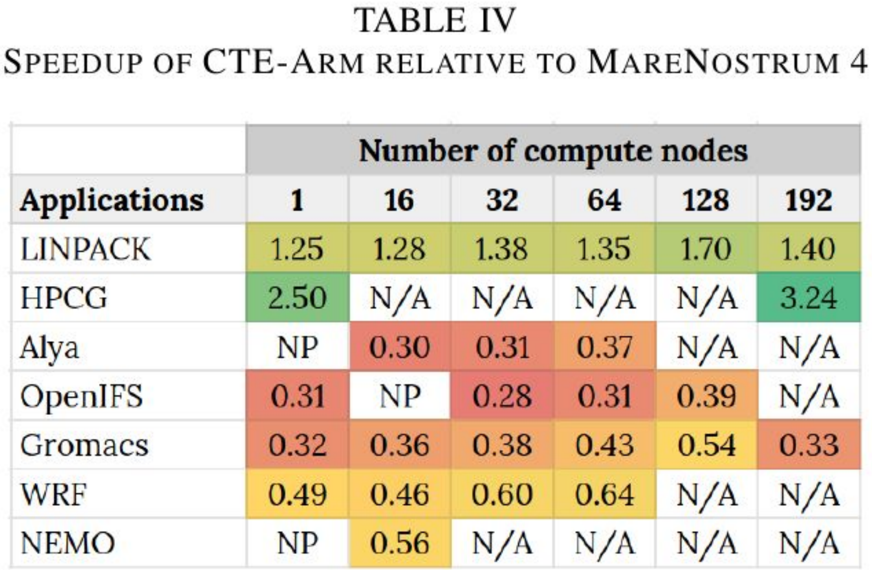


TABLE IV
SPEEDUP OF CTE-ARM RELATIVE TO MARENOSTRUM 4

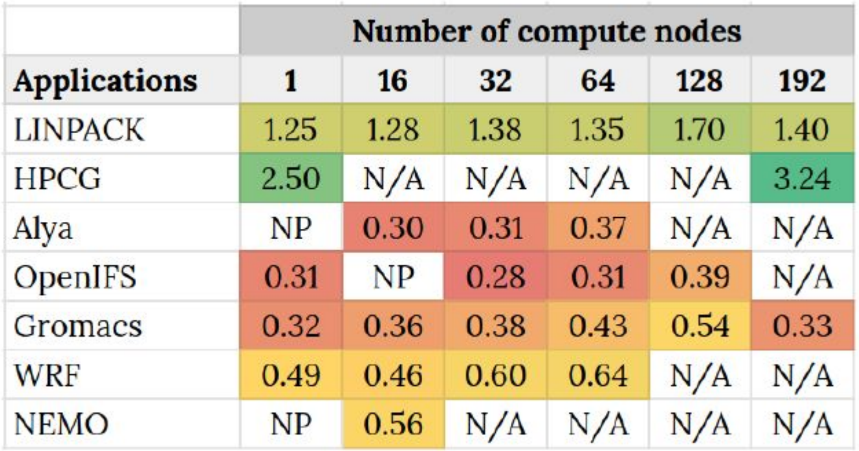
	Number of compute nodes					
Applications	**1**	**16**	**32**	**64**	**128**	**192**
LINPACK	1.25	1.28	1.38	1.35	1.70	1.40
HPCG	2.50	N/A	N/A	N/A	N/A	3.24
Alya	NP	0.30	0.31	0.37	N/A	N/A
OpenIFS	0.31	NP	0.28	0.31	0.39	N/A
Gromacs	0.32	0.36	0.38	0.43	0.54	0.33
WRF	0.49	0.46	0.60	0.64	N/A	N/A
NEMO	NP	0.56	N/A	N/A	N/A	N/A

Summary

- Software ecosystem still maturing
  - Unable to compile and run applications with Fujitsu compiler
  - GNU compiler required some tweaks to work

- Applications have to be optimized in order to leverage the hardware

TABLE IV
SPEEDUP OF CTE-ARM RELATIVE TO MARENOSTRUM 4



	Number of compute nodes					
Applications	**1**	**16**	**32**	**64**	**128**	**192**
LINPACK	1.25	1.28	1.38	1.35	1.70	1.40
HPCG	2.50	N/A	N/A	N/A	N/A	3.24
Alya	NP	0.30	0.31	0.37	N/A	N/A
OpenIFS	0.31	NP	0.28	0.31	0.39	N/A
Gromacs	0.32	0.36	0.38	0.43	0.54	0.33
WRF	0.49	0.46	0.60	0.64	N/A	N/A
NEMO	NP	0.56	N/A	N/A	N/A	N/A

Cluster of emerging technology: evaluation of a production HPC system based on A64FX

Fabio Banchelli
fabio.banchelli@bsc.es

Kilian Peiro
kilian.peiro@bsc.es

Guillem Ramirez-Gargallo
guillem.ramirez@bsc.es

Joan Vinyals
joan.vinyals@bsc.es

David Vicente
david.vicente@bsc.es

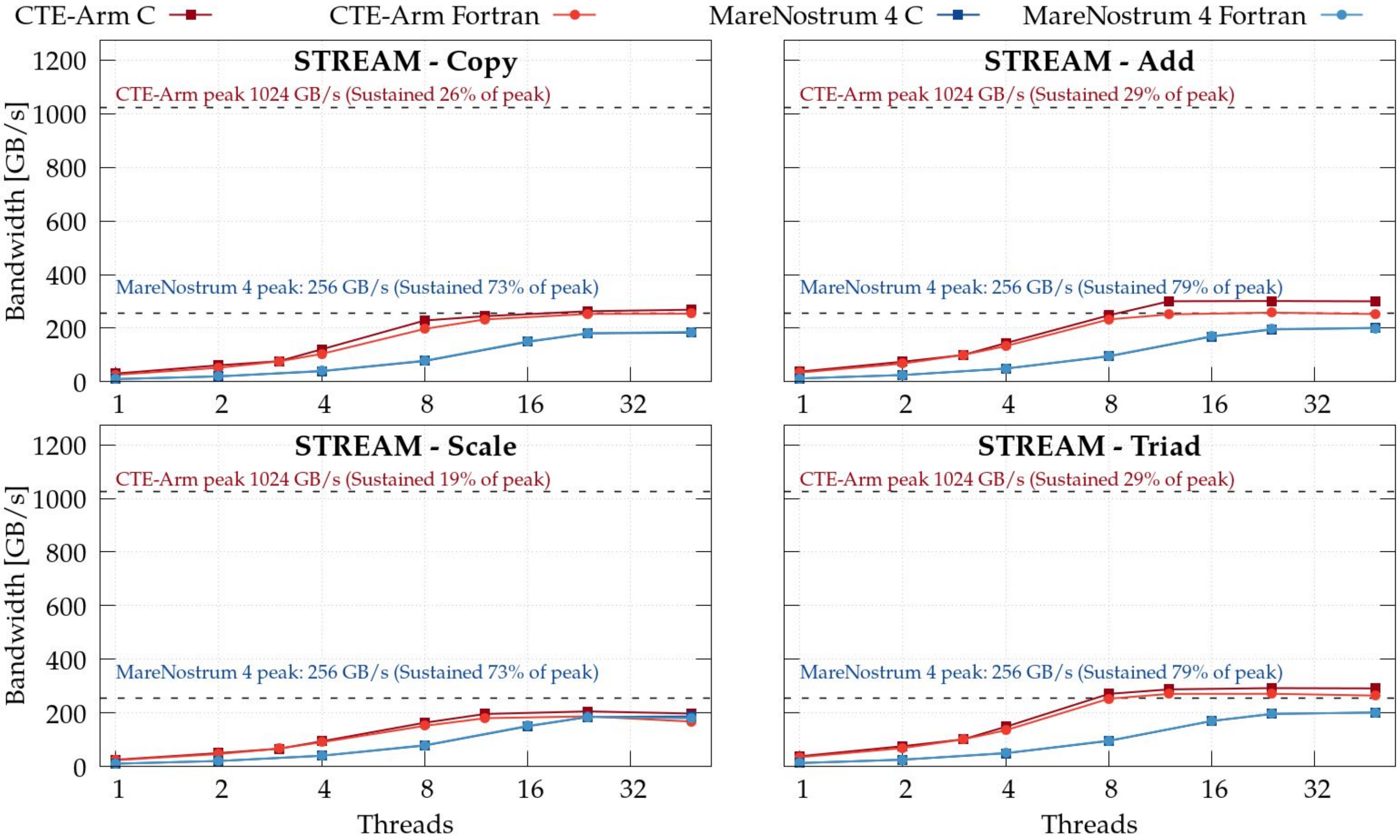
Marta Garcia-Gasulla
marta.garcia@bsc.es

Filippo Mantovani
filippo.mantovani@bsc.es

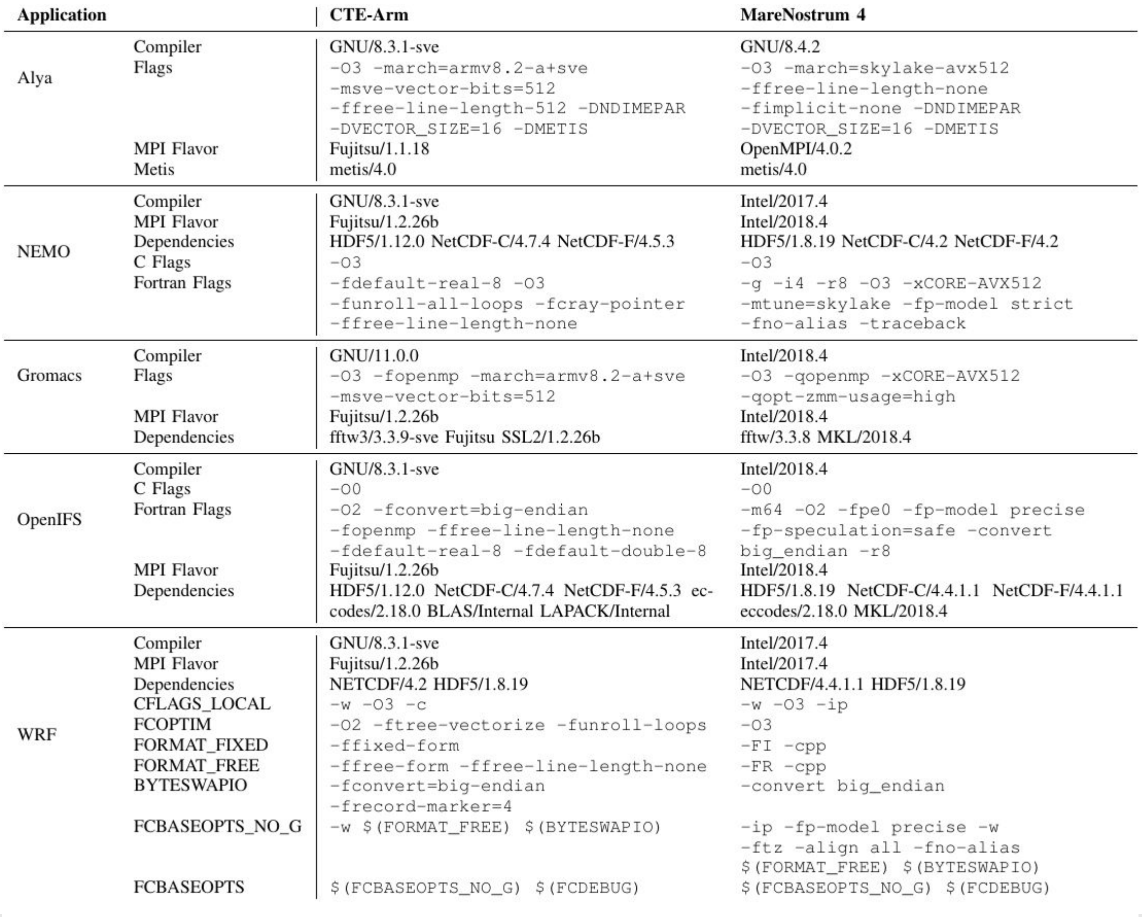
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Back-up Slides

STREAM OpenMP



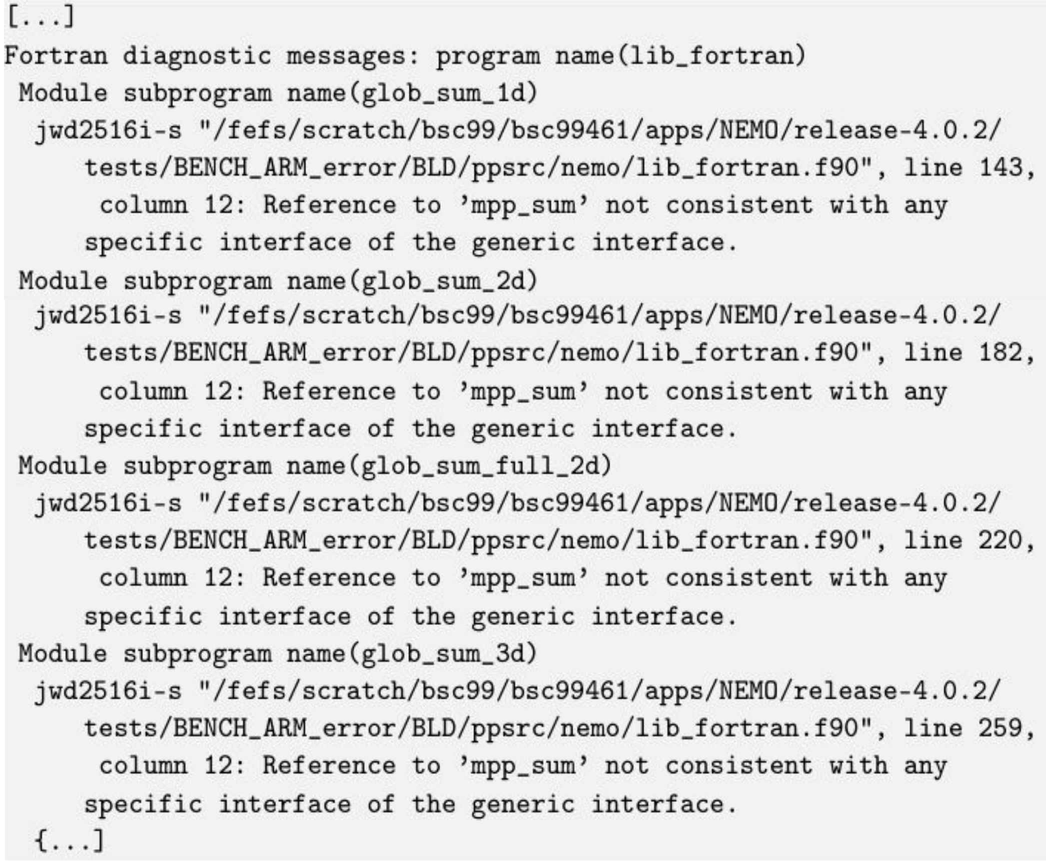
Build configurations



Application		CTE-Arm	MareNostrum 4
Alya	Compiler	GNU/8.3.1-sve	GNU/8.4.2
	Flags	-O3 -march=armv8.2-a+sve -msve-vector-bits=512 -ffree-line-length-512 -DNDIMEPAR -DVECTOR_SIZE=16 -DMETIS	-O3 -march=skylake-avx512 -ffree-line-length-none -fimplicit-none -DNDIMEPAR -DVECTOR_SIZE=16 -DMETIS
	MPI Flavor	Fujitsu/1.1.18	OpenMPI/4.0.2
	Metis	metis/4.0	metis/4.0
NEMO	Compiler	GNU/8.3.1-sve	Intel/2017.4
	MPI Flavor	Fujitsu/1.2.26b	Intel/2018.4
	Dependencies	HDF5/1.12.0 NetCDF-C/4.7.4 NetCDF-F/4.5.3	HDF5/1.8.19 NetCDF-C/4.2 NetCDF-F/4.2
	C Flags	-O3	-O3
	Fortran Flags	-fdefault-real-8 -O3 -funroll-all-loops -fcray-pointer -ffree-line-length-none	-g -i4 -r8 -O3 -xCORE-AVX512 -mtune=skylake -fp-model strict -fno-alias -traceback
Gromacs	Compiler	GNU/11.0.0	Intel/2018.4
	Flags	-O3 -fopenmp -march=armv8.2-a+sve -msve-vector-bits=512	-O3 -qopenmp -xCORE-AVX512 -qopt-zmm-usage=high
	MPI Flavor	Fujitsu/1.2.26b	Intel/2018.4
	Dependencies	fftw3/3.3.9-sve Fujitsu SSL2/1.2.26b	fftw/3.3.8 MKL/2018.4
OpenIFS	Compiler	GNU/8.3.1-sve	Intel/2018.4
	C Flags	-O0	-O0
	Fortran Flags	-O2 -fconvert=big-endian -fopenmp -ffree-line-length-none -fdefault-real-8 -fdefault-double-8	-m64 -O2 -fpe0 -fp-model precise -fp-speculation=safe -convert big_endian -r8
	MPI Flavor	Fujitsu/1.2.26b	Intel/2018.4
	Dependencies	HDF5/1.12.0 NetCDF-C/4.7.4 NetCDF-F/4.5.3 eccodes/2.18.0 BLAS/Internal LAPACK/Internal	HDF5/1.8.19 NetCDF-C/4.4.1.1 NetCDF-F/4.4.1.1 eccodes/2.18.0 MKL/2018.4
WRF	Compiler	GNU/8.3.1-sve	Intel/2017.4
	MPI Flavor	Fujitsu/1.2.26b	Intel/2017.4
	Dependencies	NETCDF/4.2 HDF5/1.8.19	NETCDF/4.4.1.1 HDF5/1.8.19
	CFLAGS_LOCAL	-w -O3 -c	-w -O3 -ip
	FCOPTIM	-O2 -ftree-vectorize -funroll-loops	-O3
	FORMAT_FIXED	-ffixed-form	-FI -cpp
	FORMAT_FREE	-ffree-form -ffree-line-length-none	-FR -cpp
	BYTESWAPIO	-fconvert=big-endian -frecord-marker=4	-convert big_endian
	FCBASEOPTS_NO_G	-w $(FORMAT_FREE) $(BYTESWAPIO)	-ip -fp-model precise -w -ftz -align all -fno-alias $(FORMAT_FREE) $(BYTESWAPIO)
	FCBASEOPTS	$(FCBASEOPTS_NO_G) $(FCDEBUG)	$(FCBASEOPTS_NO_G) $(FCDEBUG)

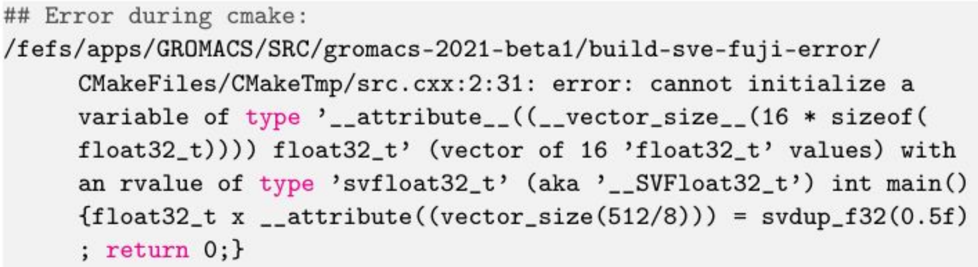
NEMO Error

Compilation error



```
[...]
Fortran diagnostic messages: program name(lib_fortran)
 Module subprogram name(glob_sum_1d)
  jwd2516i-s "/fefs/scratch/bsc99/bsc99461/apps/NEMO/release-4.0.2/
     tests/BENCH_ARM_error/BLD/ppsrc/nemo/lib_fortran.f90", line 143,
      column 12: Reference to ’mpp_sum’ not consistent with any
     specific interface of the generic interface.
 Module subprogram name(glob_sum_2d)
  jwd2516i-s "/fefs/scratch/bsc99/bsc99461/apps/NEMO/release-4.0.2/
     tests/BENCH_ARM_error/BLD/ppsrc/nemo/lib_fortran.f90", line 182,
      column 12: Reference to ’mpp_sum’ not consistent with any
     specific interface of the generic interface.
 Module subprogram name(glob_sum_full_2d)
  jwd2516i-s "/fefs/scratch/bsc99/bsc99461/apps/NEMO/release-4.0.2/
     tests/BENCH_ARM_error/BLD/ppsrc/nemo/lib_fortran.f90", line 220,
      column 12: Reference to ’mpp_sum’ not consistent with any
     specific interface of the generic interface.
 Module subprogram name(glob_sum_3d)
  jwd2516i-s "/fefs/scratch/bsc99/bsc99461/apps/NEMO/release-4.0.2/
     tests/BENCH_ARM_error/BLD/ppsrc/nemo/lib_fortran.f90", line 259,
      column 12: Reference to ’mpp_sum’ not consistent with any
     specific interface of the generic interface.
  {...]
```

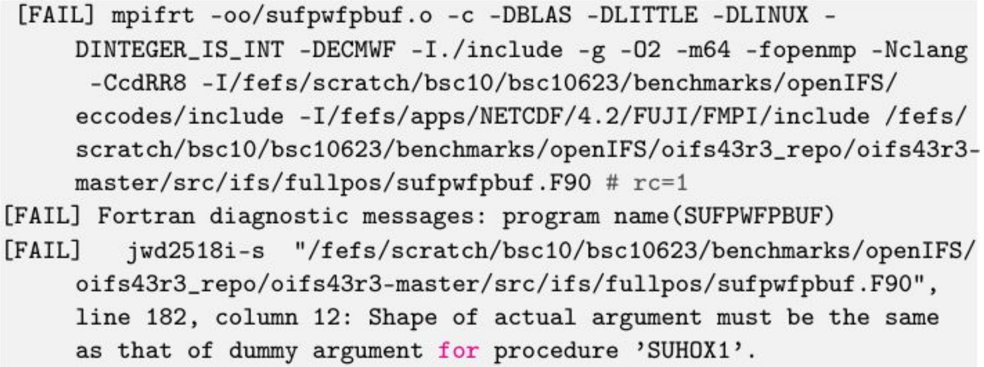
Gromacs Error



```
## Error during cmake:
/fefs/apps/GROMACS/SRC/gromacs-2021-beta1/build-sve-fuji-error/
     CMakeFiles/CMakeTmp/src.cxx:2:31: error: cannot initialize a
     variable of type '__attribute__((__vector_size__(16 * sizeof(
     float32_t)))) float32_t' (vector of 16 'float32_t' values) with
     an rvalue of type 'svfloat32_t' (aka '__SVFloat32_t') int main()
     {float32_t x __attribute((vector_size(512/8))) = svdup_f32(0.5f)
     ; return 0;}
```

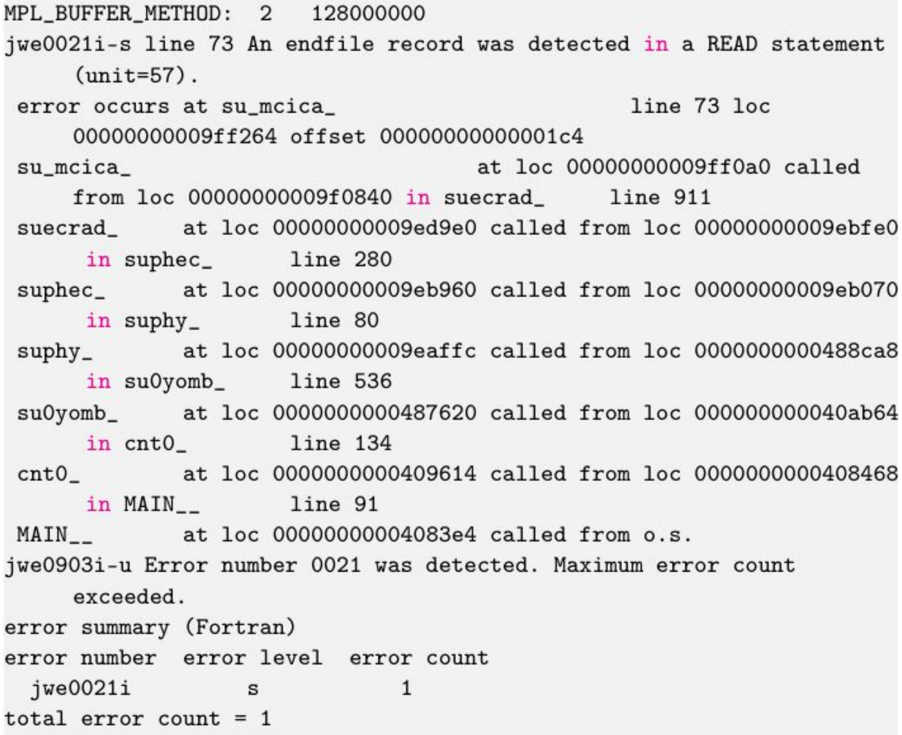
CMake error

OpenIFS Error



```
 [FAIL] mpifrt -oo/sufpwfpbuf.o -c -DBLAS -DLITTLE -DLINUX -
     DINTEGER_IS_INT -DECMWF -I./include -g -O2 -m64 -fopenmp -Nclang
      -CcdRR8 -I/fefs/scratch/bsc10/bsc10623/benchmarks/openIFS/
     eccodes/include -I/fefs/apps/NETCDF/4.2/FUJI/FMPI/include /fefs/
     scratch/bsc10/bsc10623/benchmarks/openIFS/oifs43r3_repo/oifs43r3-
     master/src/ifs/fullpos/sufpwfpbuf.F90 # rc=1
[FAIL] Fortran diagnostic messages: program name(SUFPWFPBUF)
[FAIL]   jwd2518i-s  "/fefs/scratch/bsc10/bsc10623/benchmarks/openIFS/
     oifs43r3_repo/oifs43r3-master/src/ifs/fullpos/sufpwfpbuf.F90",
     line 182, column 12: Shape of actual argument must be the same
     as that of dummy argument for procedure 'SUHOX1'.
```

Compilation error - Solved with minor modifications to the source code



```
MPL_BUFFER_METHOD:  2   128000000
jwe0021i-s line 73 An endfile record was detected in a READ statement
     (unit=57).
 error occurs at su_mcica_                     line 73 loc
     00000000009ff264 offset 00000000000001c4
 su_mcica_                          at loc 00000000009ff0a0 called
     from loc 00000000009f0840 in suecrad_     line 911
 suecrad_     at loc 00000000009ed9e0 called from loc 00000000009ebfe0
      in suphec_      line 280
 suphec_      at loc 00000000009eb960 called from loc 00000000009eb070
      in suphy_       line 80
 suphy_       at loc 00000000009eaffc called from loc 0000000000488ca8
      in su0yomb_     line 536
 su0yomb_     at loc 0000000000487620 called from loc 000000000040ab64
      in cnt0_        line 134
 cnt0_        at loc 0000000000409614 called from loc 0000000000408468
      in MAIN__       line 91
 MAIN__       at loc 00000000004083e4 called from o.s.
jwe0903i-u Error number 0021 was detected. Maximum error count
     exceeded.
error summary (Fortran)
error number  error level  error count
  jwe0021i         s           1
total error count = 1
```

Runtime error