Experience with new architectures: moving from HELIOS to Marconi

Serhiy Mochalskyy, Roman Hatzky

3rd Accelerated Computing For Fusion Workshop
November 28–29th, 2016, Saclay, France

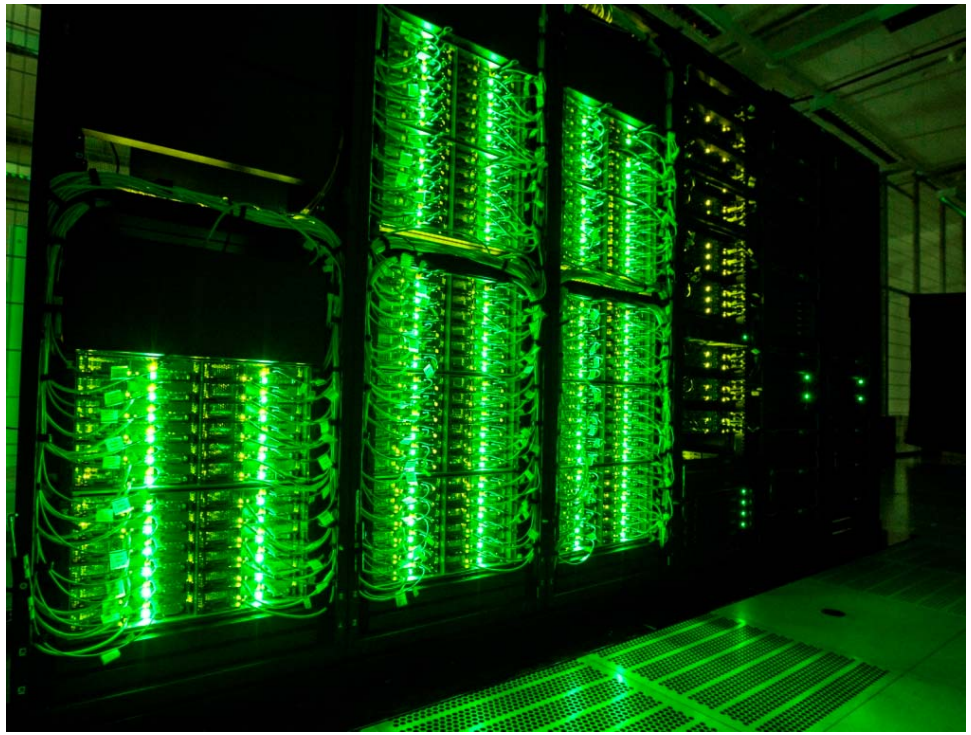
High Level Support Team
Max-Planck-Institut für Plasmaphysik
Boltzmannstr. 2, D-85748 Garching, Germany

Outline

- **Marconi general architecture**
- **Marconi vs HELIOS**
- **Roofline model**
- **Stream benchmark**
- **Intel MPI Benchmark**
- ***MPI_Barrier, MPI_Init, MPI_Alltoall* performance test**
- **Porting Starwall code on Marconi**
- **Summary**

Marconi general architecture

Marconi supercomputer – Bolonia, Italy



Model: Lenovo NeXtScale

1) A preliminary system went into production in July 2016: Intel Xeon processor E5-2600 v4 (Broadwell). 1512 computing nodes -> 2 Pflops. (HELIOS – 1.52 Pflops)

2) Till the end of 2016: the last generation of the Intel Xeon Phi (*Knights Landing*) ->11 Pflops.

3) July 2017: Intel Xeon processor *Skylake* -> 20 Pflops.

Marconi vs HELIOS

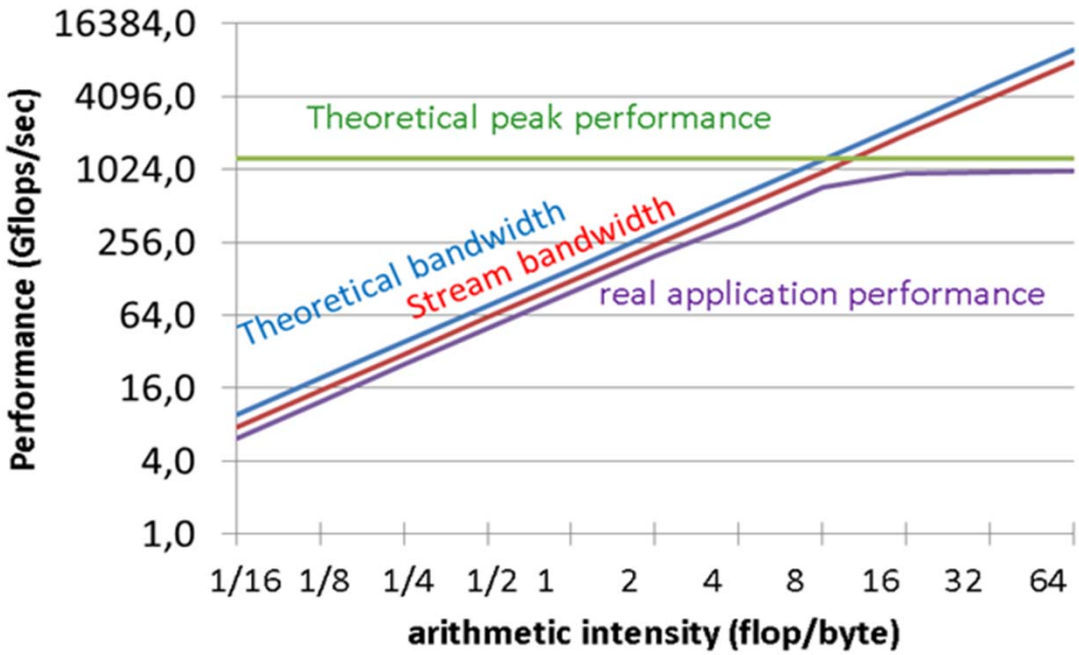
Comparison of CPU installed on Helios and Marconi

Processor	**Intel Sandy Bridge (HELIOS)**	**Intel Broadwell (Marconi)**
Number of cores	8	18
Memory	32 GB	64 GB
Frequency	2.6 GHz	2.3 GHz
FMA units	1	2
Peak performance	173 GFlop/s	633 GFlop/s
Memory bandwidth	68 GB/s	76.8 GB/s

- **~x1.62** increase in performance per core
- **~x3.6** increase in peak performance
- **~x1.13** increase in memory bandwidth

Marconi roofline model

Roofline model for Intel Broadwell installed on Marconi

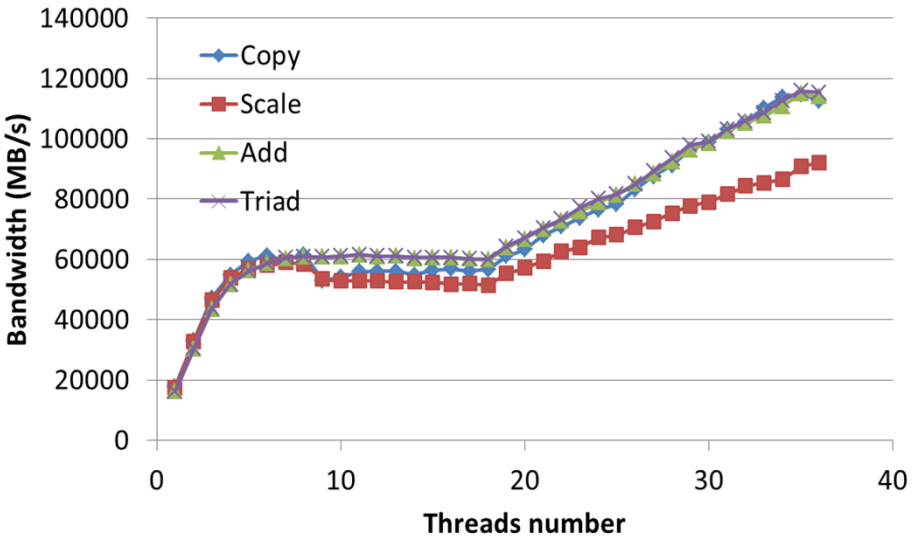


$$\text{Performance} = min \begin{cases} Peak\ performance\ (Gflops/s) \\ Arithmetic\ Intensity \cdot Memory\ bandwidth\ (Gflops/s) \end{cases}$$

- 80 % of the theoretical peak performance can be reached

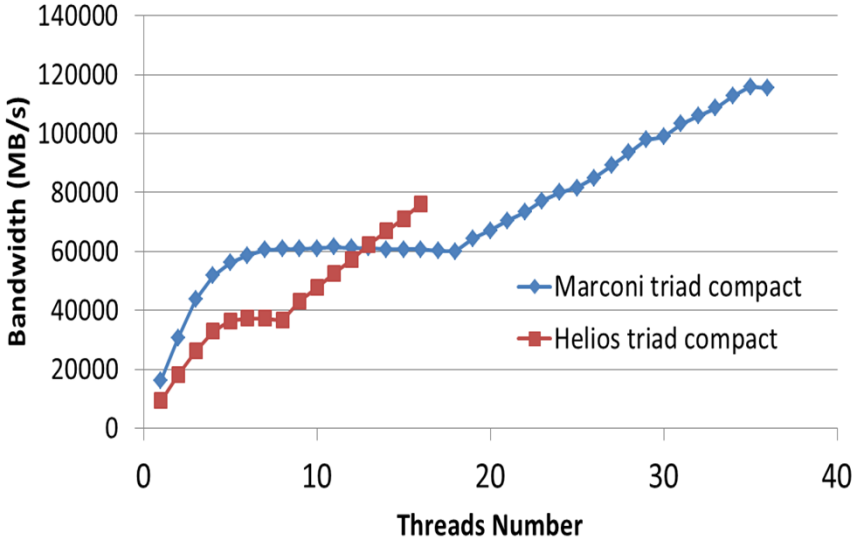
Stream Benchmark – compact pinning

Stream benchmark on Marconi



- For one CPU memory bandwidth ~61 Gbytes/s (79 % of theoretical)
- For one node memory bandwidth ~118 Gbytes/s (77 % of theoretical)

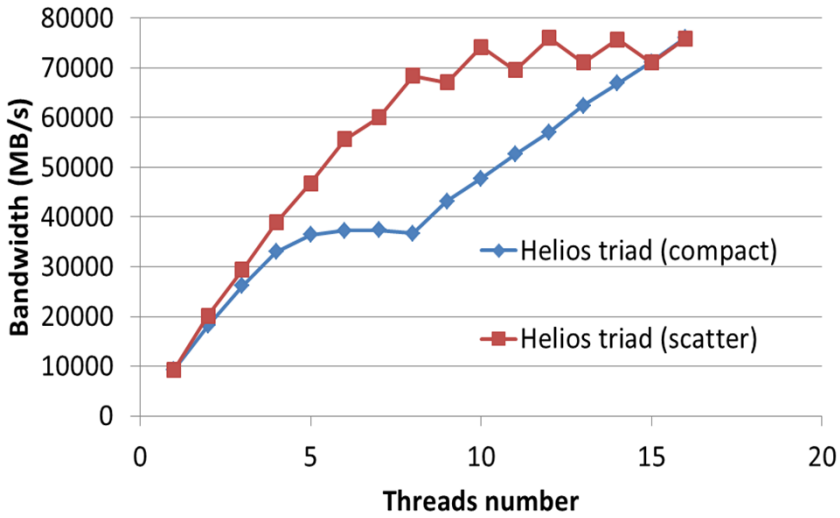
Marconi vs HELIOS



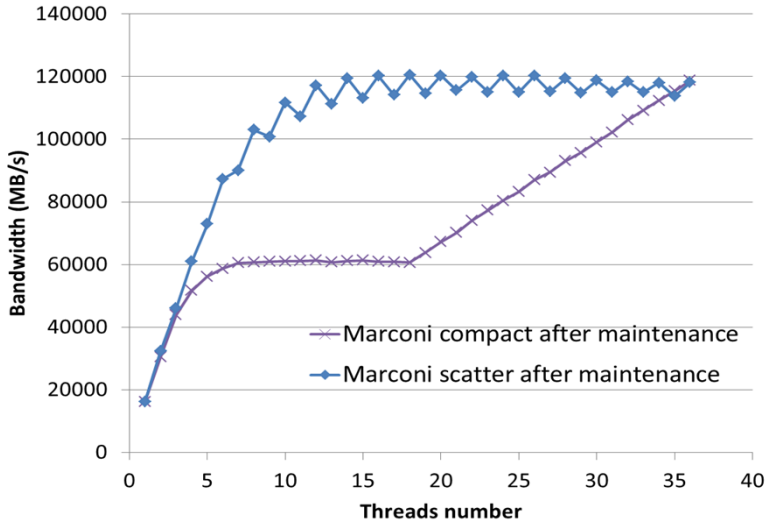
- Both supercomputers provide expected behavior
- Bandwidth ratio even higher than expected on Marconi x1.5 in comparison with Helios

Stream Benchmark – scatter vs compact pinning

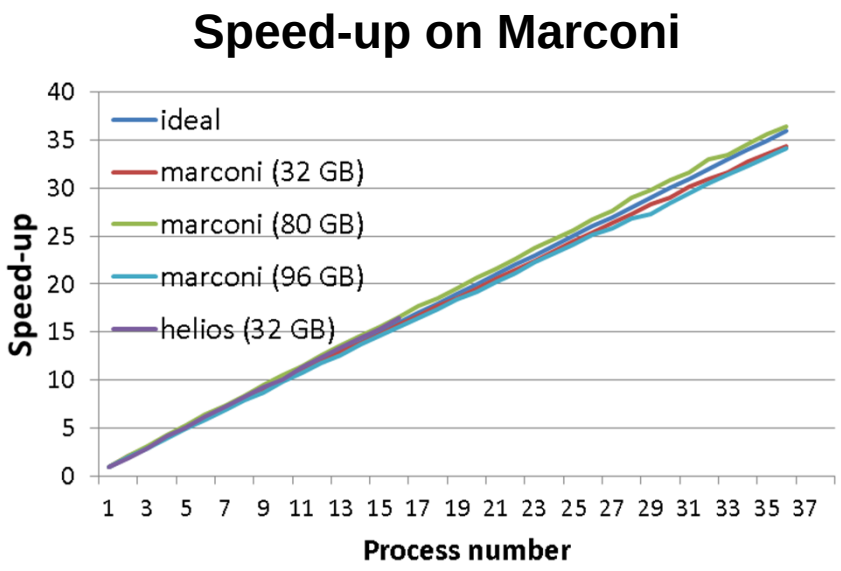
Stream benchmark on HELIOS

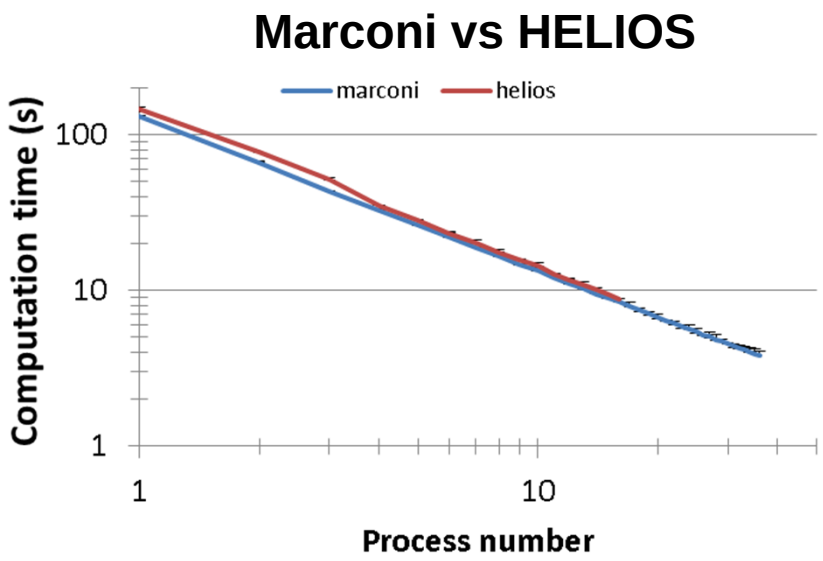


Stream benchmark on Marconi



Speed-up test within one node





- Good speed-up for all array sizes
- In spite of a lower CPU frequency, Marconi is faster than Helios for all core numbers (reason → 2 FMA)

Intel MPI benchmark (1) intra node

Ping Pong test for latency and memory bandwidth within one node

Intra node Marconi

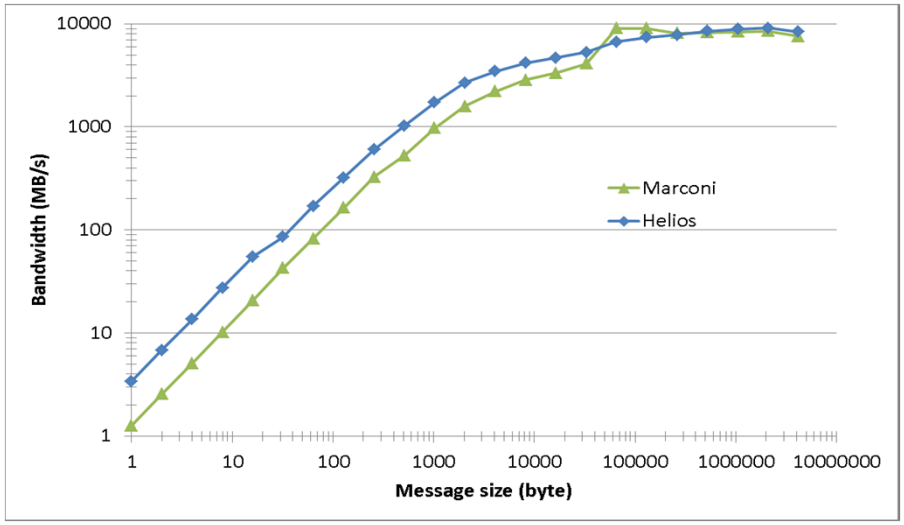
node0	CPU0	0.61
	CPU1	1.09
Latency (µs)		CPU0
		node0

Intra node HELIOS

node0	CPU0	0.25
	CPU1	0.64
Latency (µs)		CPU0
		node0

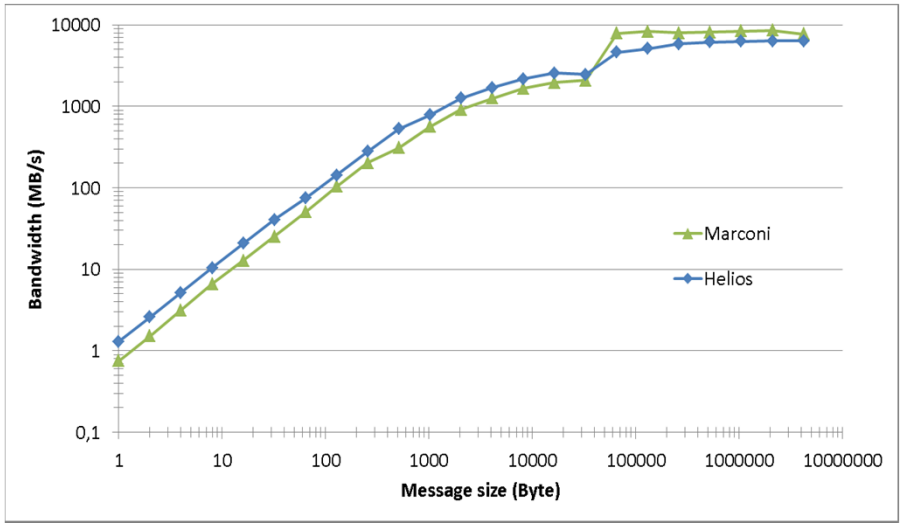
Marconi vs HELIOS

same CPU same node



Marconi vs HELIOS

different CPU same node



- The latency is lower on HELIOS but the bandwidth is higher on Marconi

Intel MPI benchmark (2) inter node

Ping Pong test for latency and memory bandwidth for two distinct nodes

Inter node Marconi

node0	CPU0	1.49
Latency (µs)		CPU0
		node1

node0	CPU0	352
Bandwidth (MB/s)		CPU0
		node1

Inter node HELIOS

node0	CPU0	1.13
Latency (µs)		CPU0
		node1

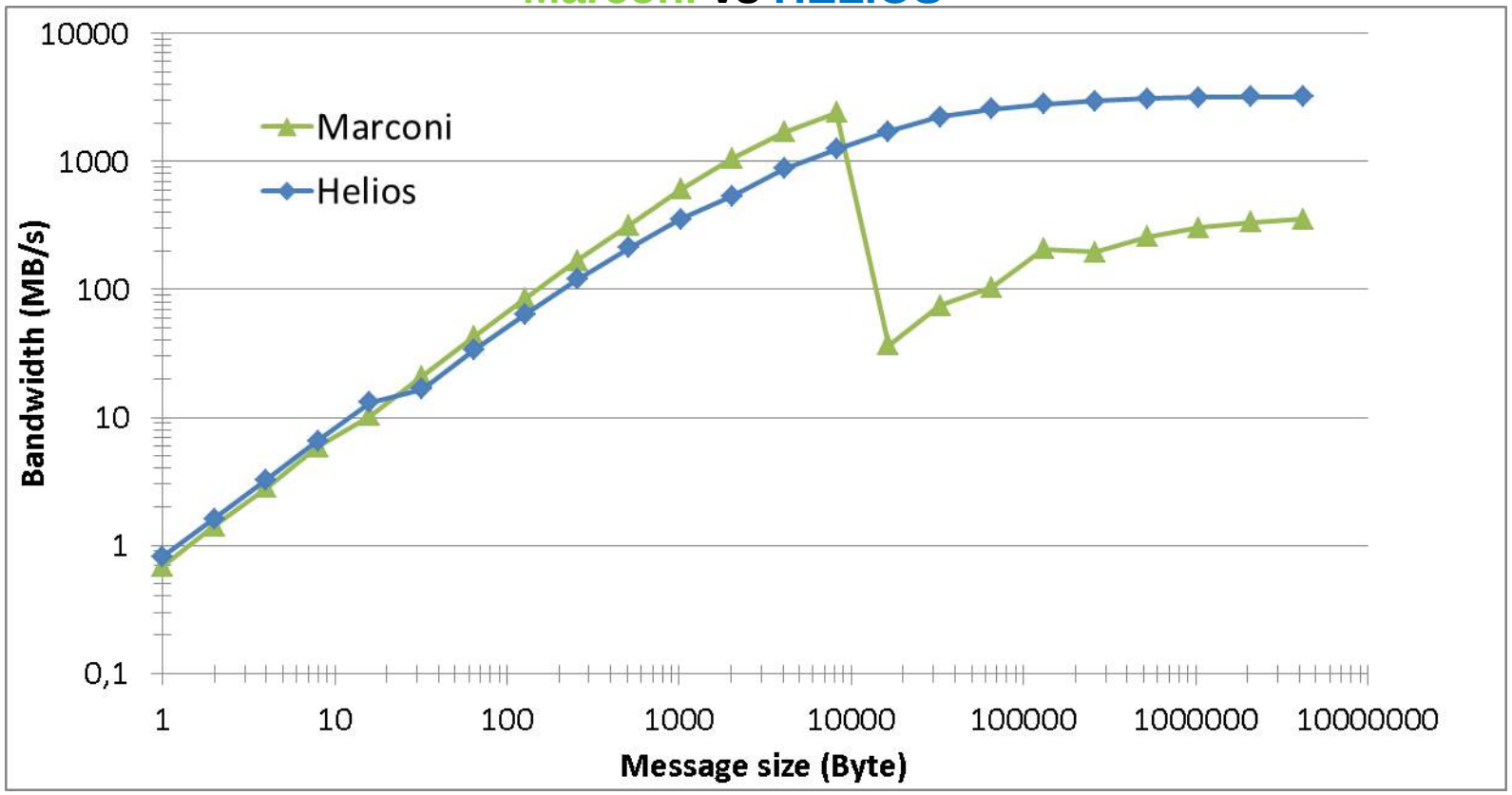
node0	CPU0	3202
Bandwidth (MB/s)		CPU0
		node1

- The Marconi inter node bandwidth is very low and “strange”

Intel MPI benchmark (3) inter node

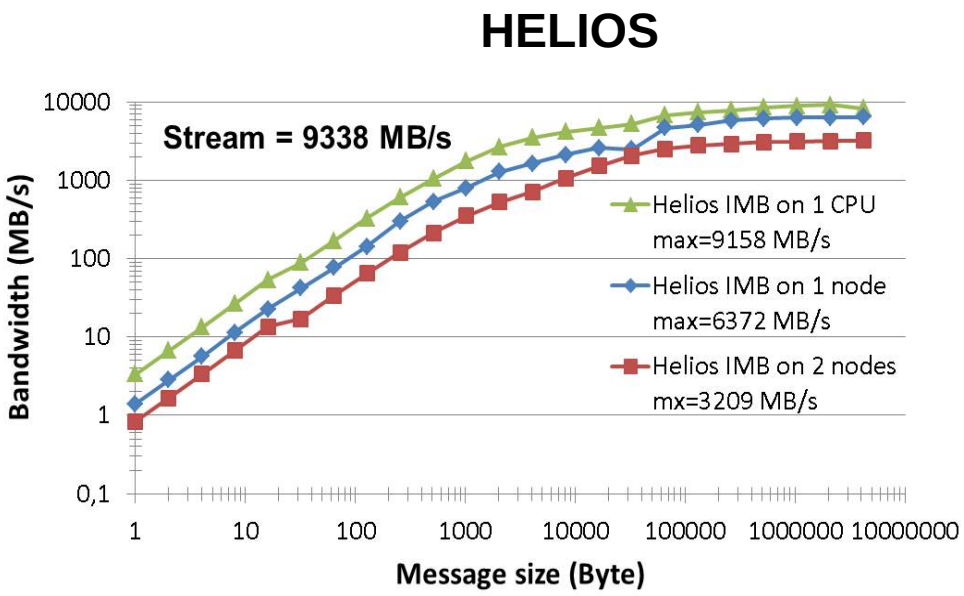
Ping Pong test for memory bandwidth of two distinct nodes

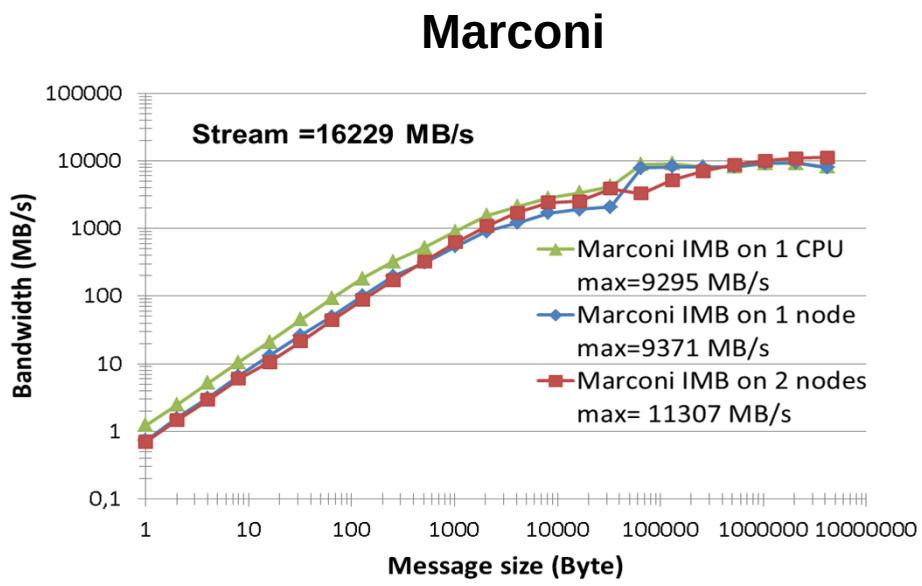
Marconi vs HELIOS



- The Marconi bandwidth broke down at a message size of 8kB

Intel MPI benchmark (4) summary

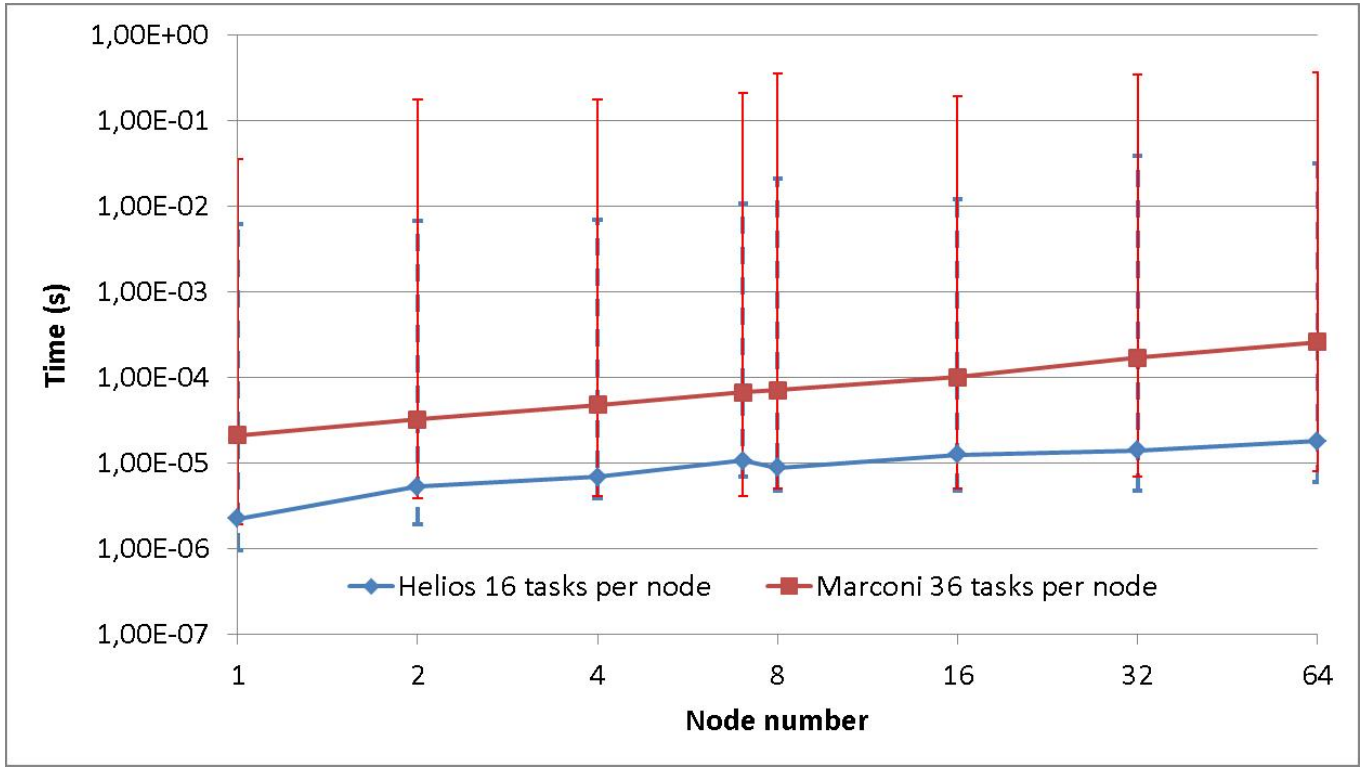




- HELIOS bandwidth shows expected behavior
- Marconi Stream bandwidth is much higher than Intel IMB
- **Marconi Intra node bandwidth is higher than intra node**

Basic MPI test on Marconi

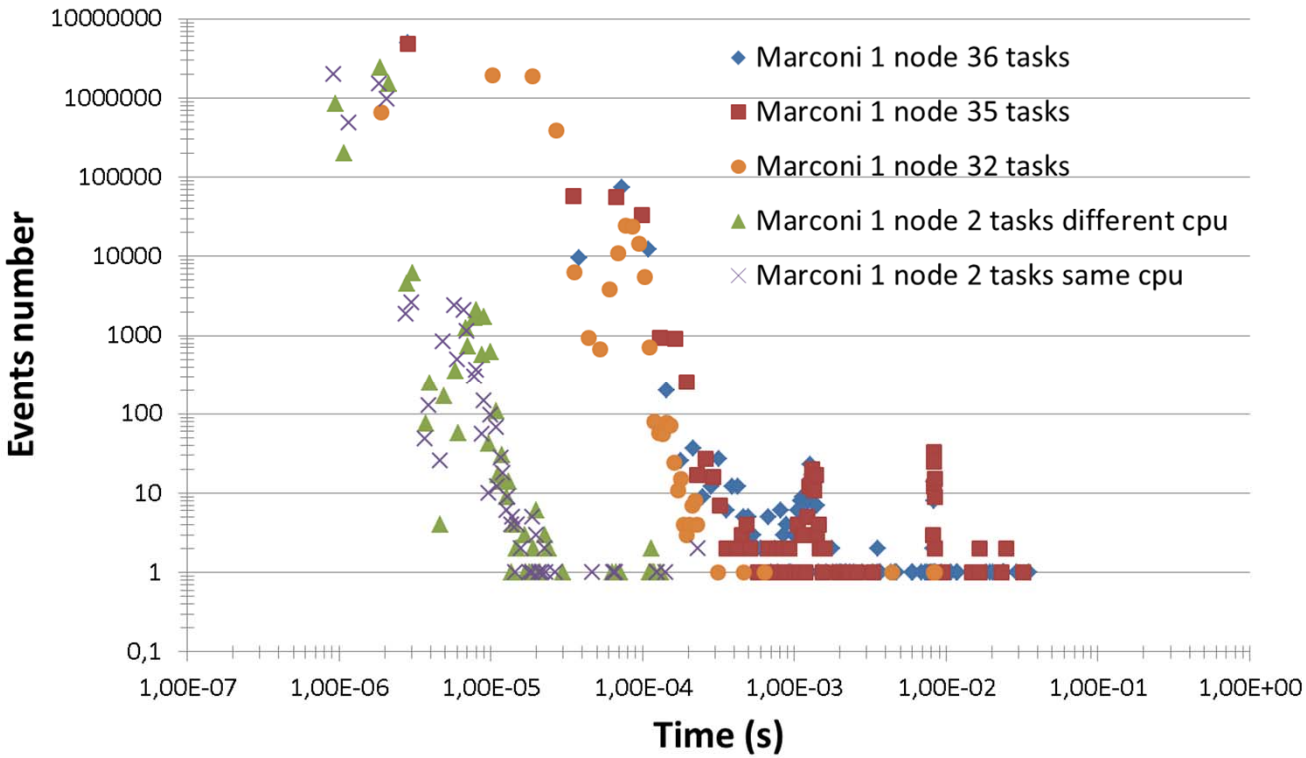
Execution of the *MPI_Barrier*: Marconi vs HELIOS



- Mean value is reasonable but large maximum peaks appear
- Such peaks appears even on one node
- With new update the max peaks on Marconi decrease by one order but they are still one order of magnitude slower than on Helios

Basic MPI test on Marconi

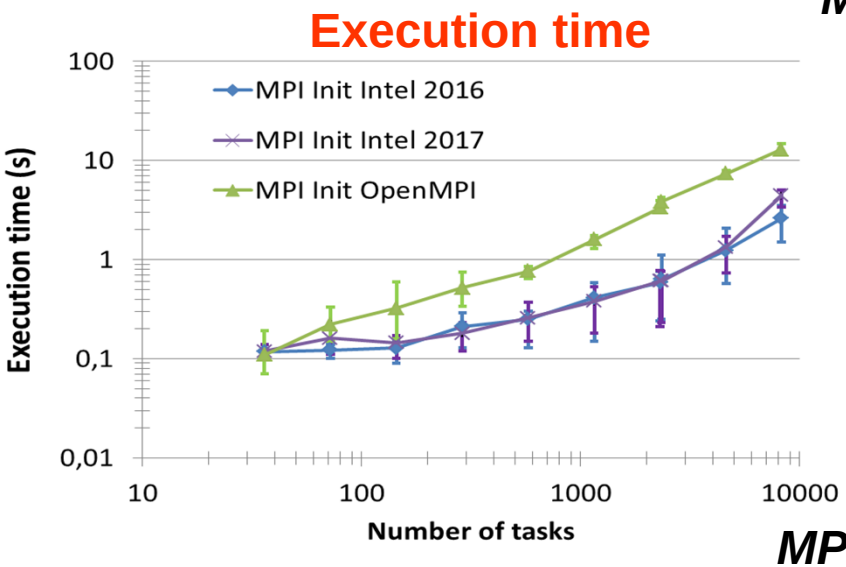
Histogram of execution of the *MPI_Barrier* on one node using different task number

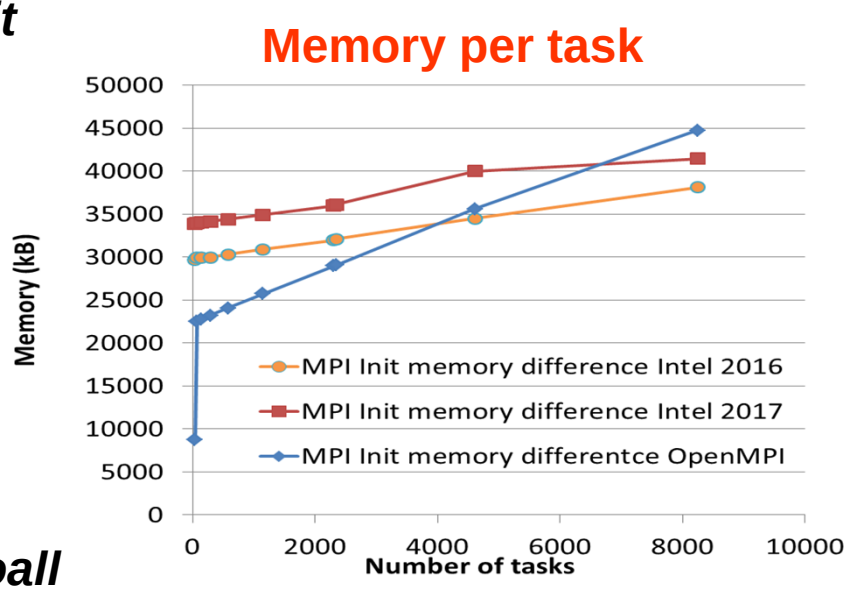


- Within one node the execution of *MPI_Barrier* remains much slower on Marconi for 32, 35 and 36 tasks but it is fast for 2 and 4 tasks

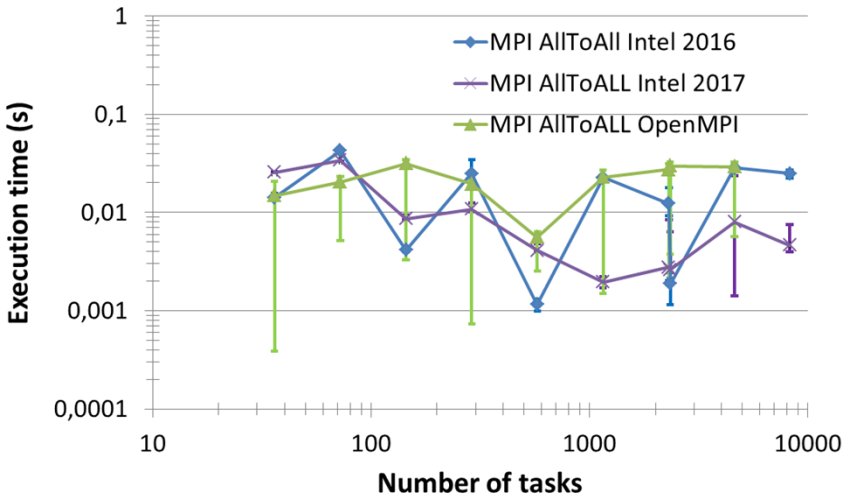
MPI_Init and *MPI_Alltoall* tests

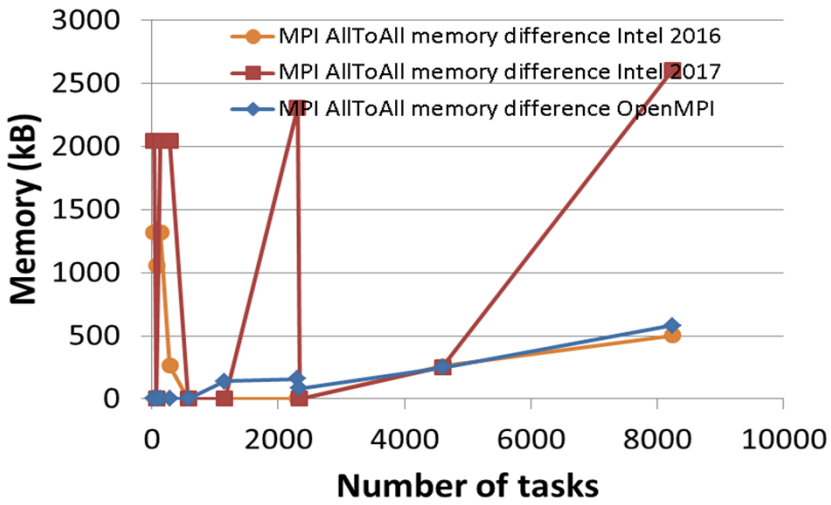
MPI_Init





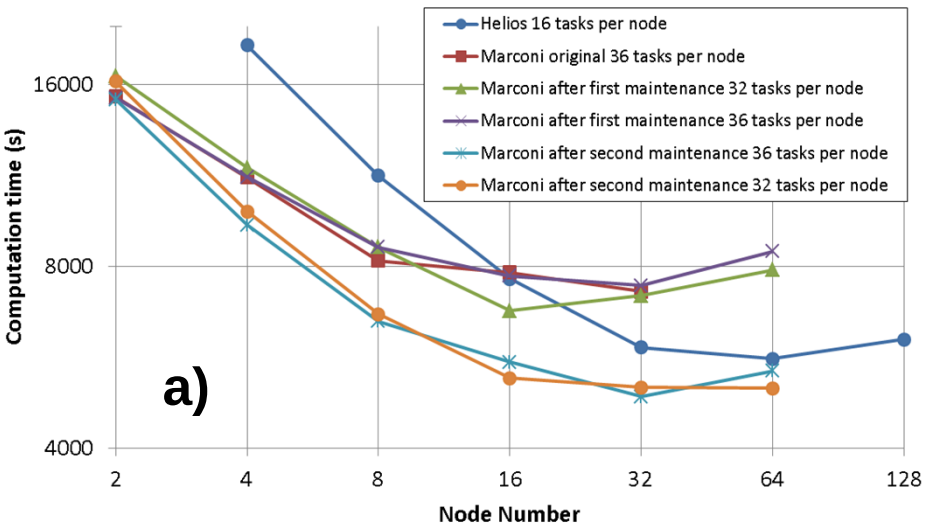
MPI_Alltoall

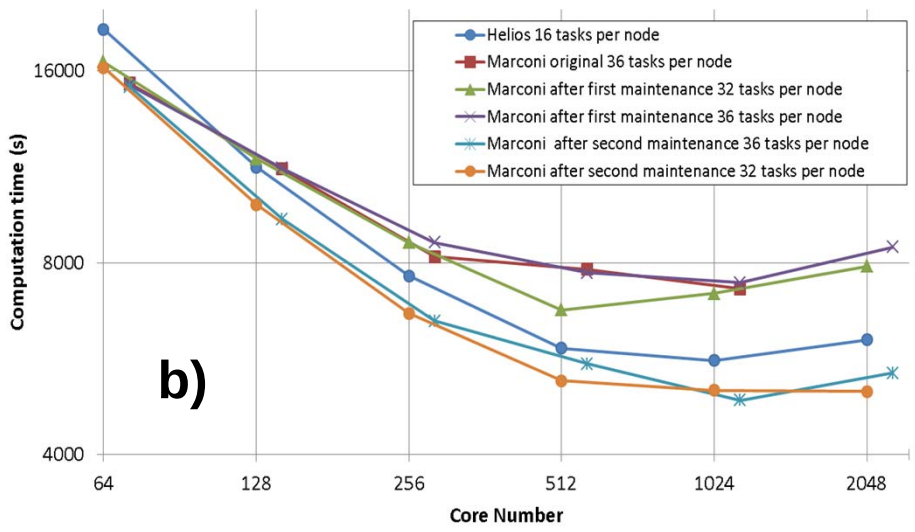




Porting Starwall code on Marconi

Scalability test Marconi vs HELIOS





- Due to larger memory Marconi can perform the test even on two nodes
- Marconi is faster for small number of nodes (even if one compares the same number of cores)
- Scalability breaks on Marconi at 16 nodes

Summary

- **Marconi supercomputer was tested during pre official operation phase.**
- **The roofline model was constructed and tested for the Intel Broadwell CPU.**
- **Different benchmarks were executed:**
  - Stream
  - Intel MPI benchmark
  - *MPI_Barrier, MPI_Init, MPI_Alltoall*
- **A problem with memory bandwidth was found.**
- **The performance and scalability of the Starwall code were tested.**

Thank you for your attention

Small bugs

- **PBS system**
- **Problem with file system: no free space**
- **Problem with operation system: hanging**
- **Problem with module loading: errors for some modules**
- **-*envlist* flag**

Bug with intel fortran16 compiler installed on Marconi

At the run time of the Fortran code (Starwall)

↓

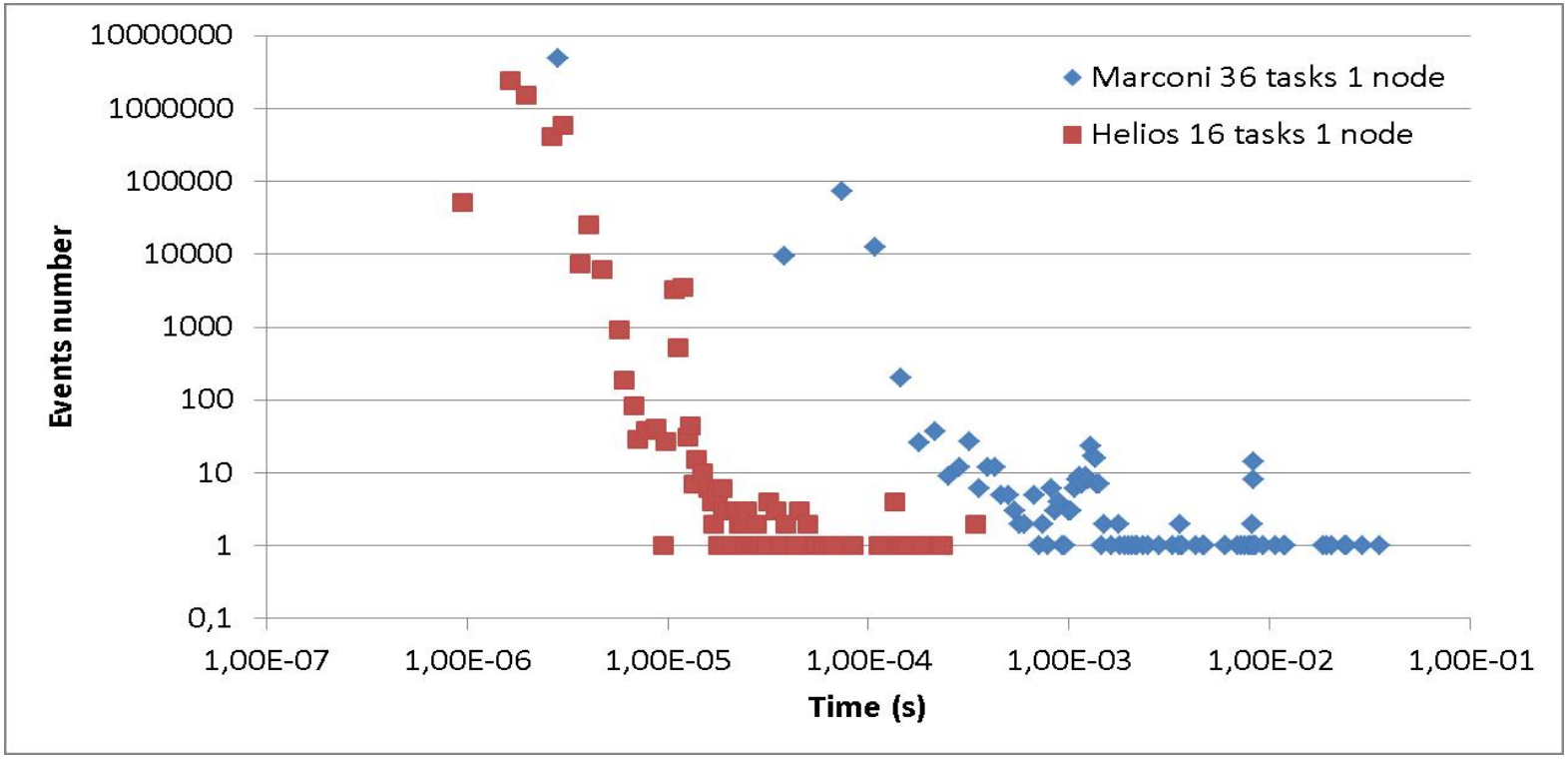
"buffer overflow detected" problem → **Temporary solution was to use auxiliary environment variables (export FOR_PRINT=ok.out export FOR_PRINT=/dev/null)**

↓

The bug was found in intel Fortran 16 compiler with PID number → **PID was limited to 5 digits as a temporary solution which should be corrected in the Intel 17**

Basic MPI test on Marconi (3)

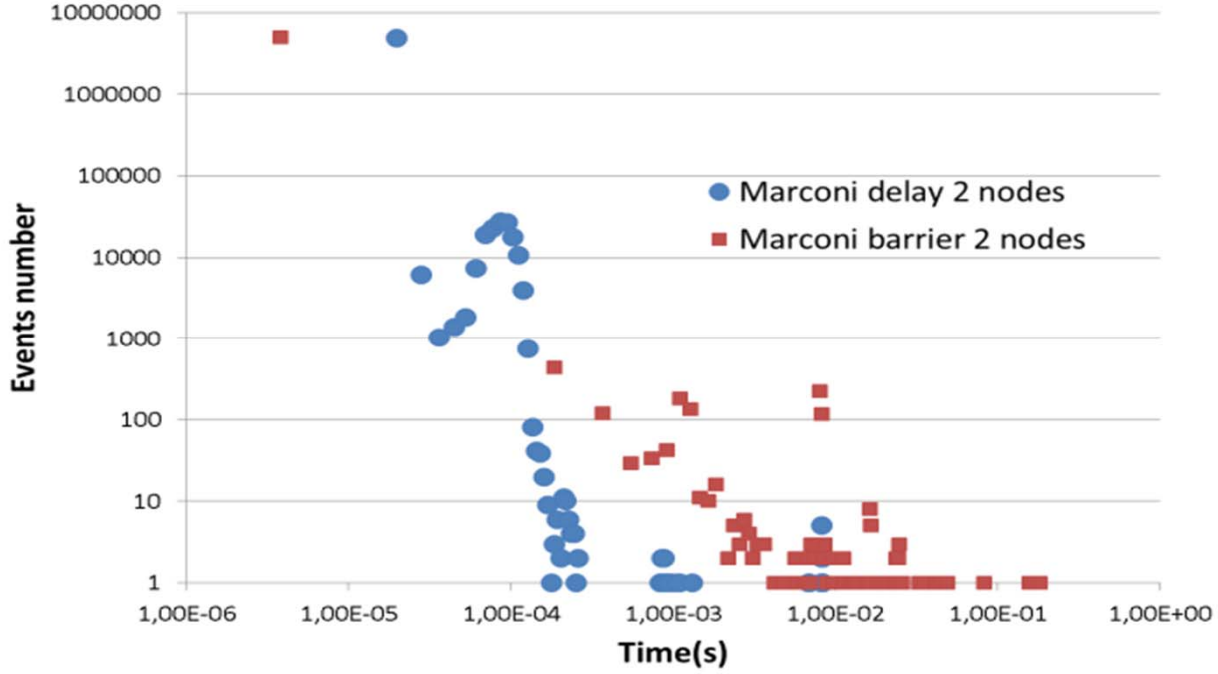
Execution of the MPI_BARRIER on one node-probability density function: Helios vs Marconi



- Within one node the execution of MPI_BARRIER remains much slower on Marconi in comparison with Helios

Basic test on Marconi (5)

Histogram of execution of the mathematical operation (delay) on one node

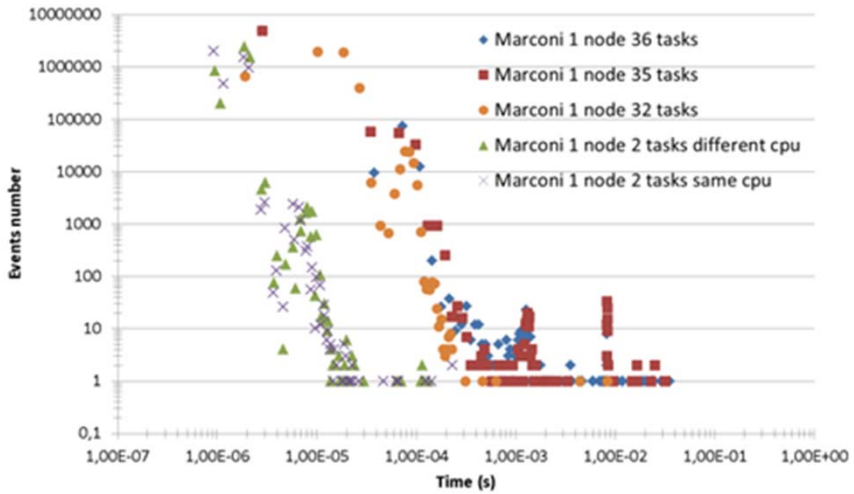


- Slow events appear for both MPI_BARRIER and “delay” operations but less pronounced for “delay”

Basic MPI test on Marconi

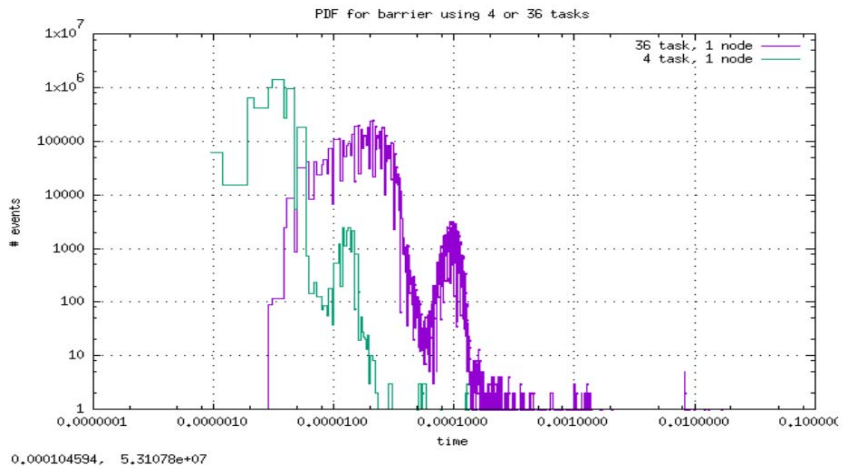
Histogram of execution of the MPI_BARRIER on one node using different task number

HLST results

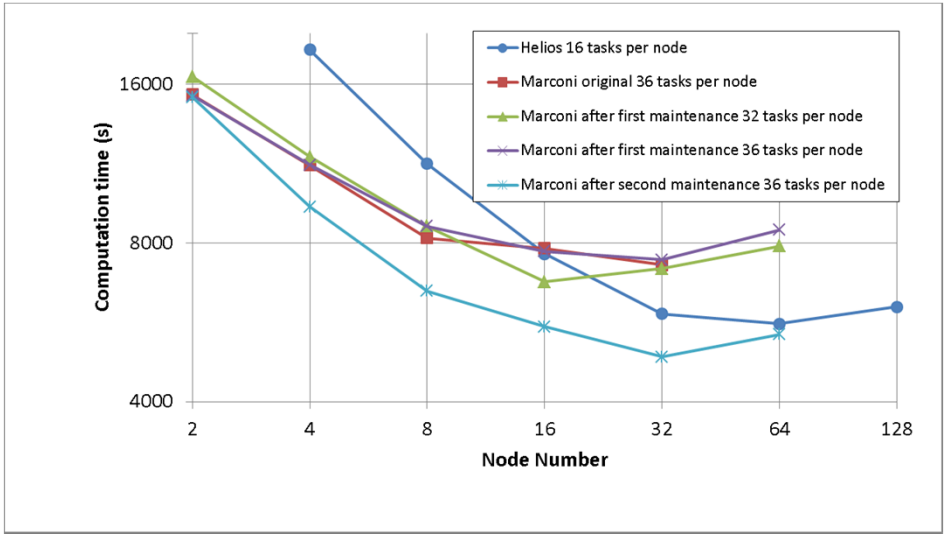


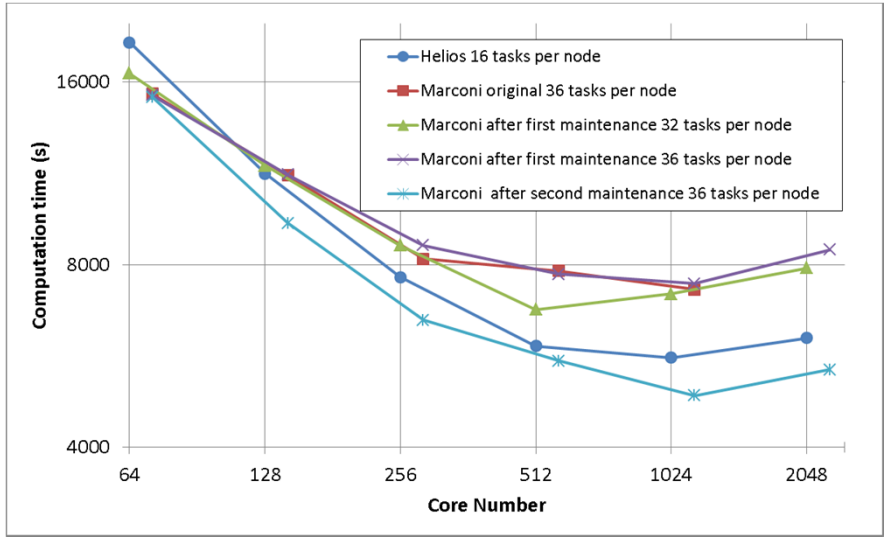
CINECA results after opening ticket

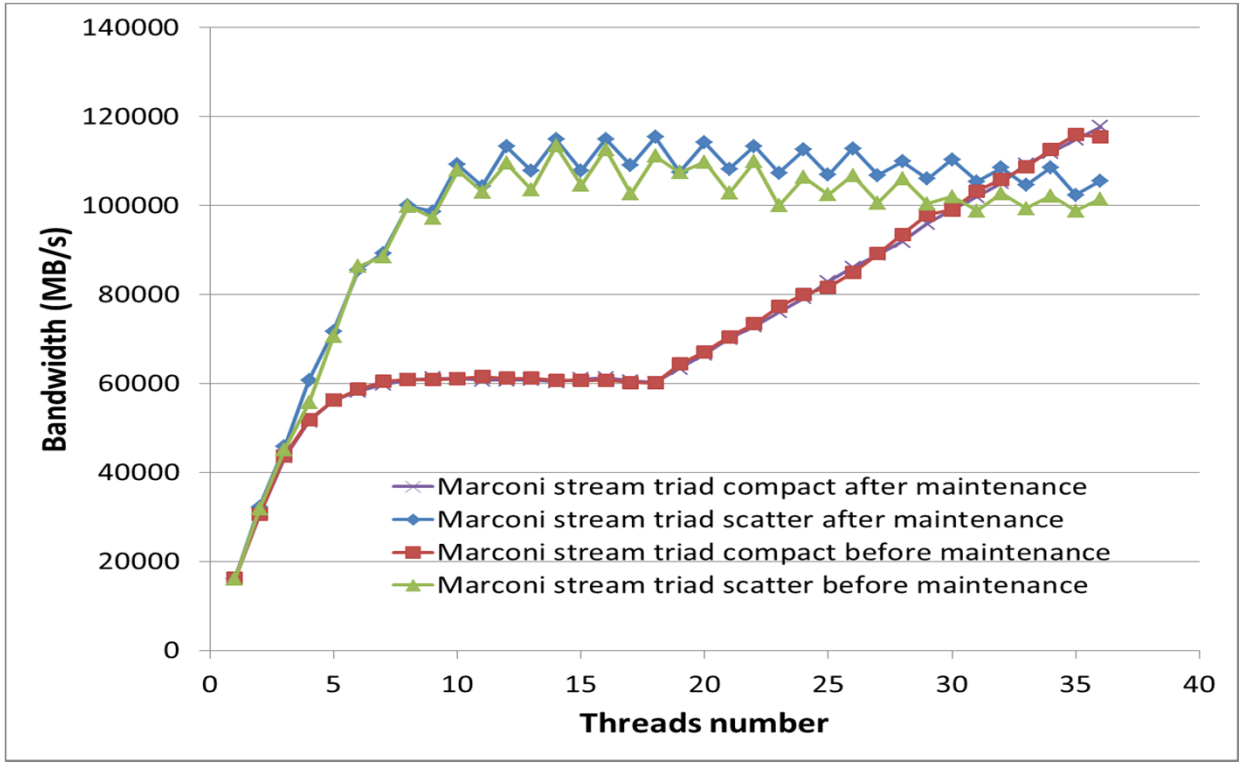
✓ Histogram using 4 or 36 tasks inside a node

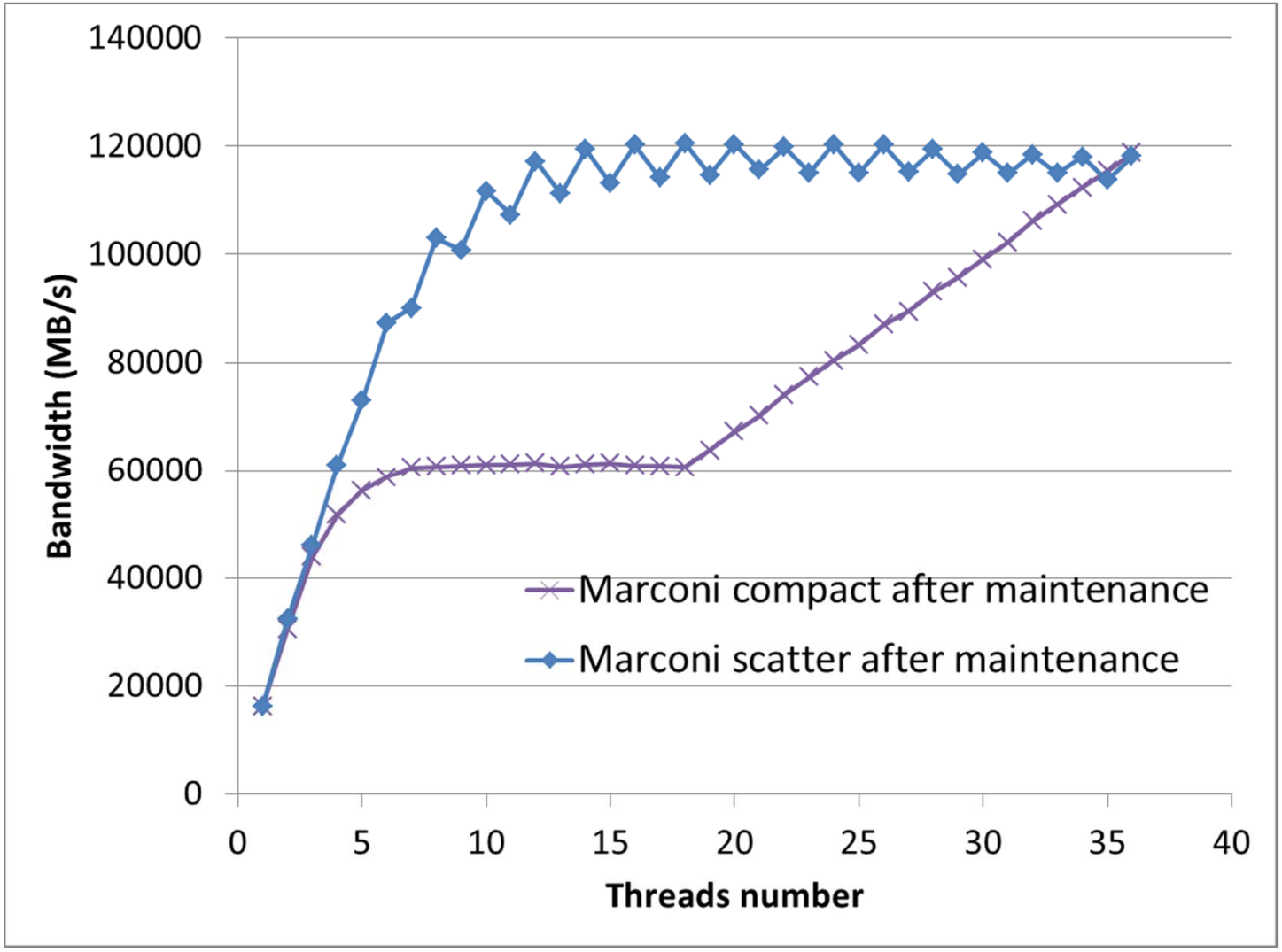


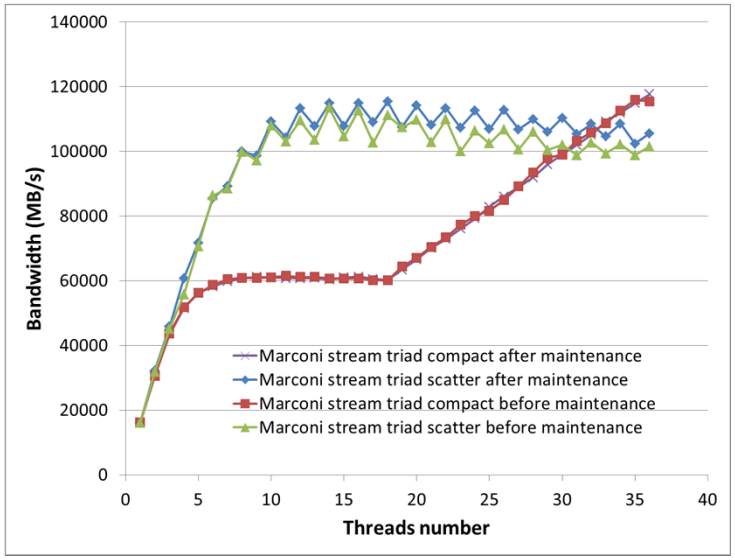
➢ Within one node the execution of MPI_BARRIER remains much slower on Marconi for 32, 35 and 36 tasks but it is very fast for 2 and 4 tasks

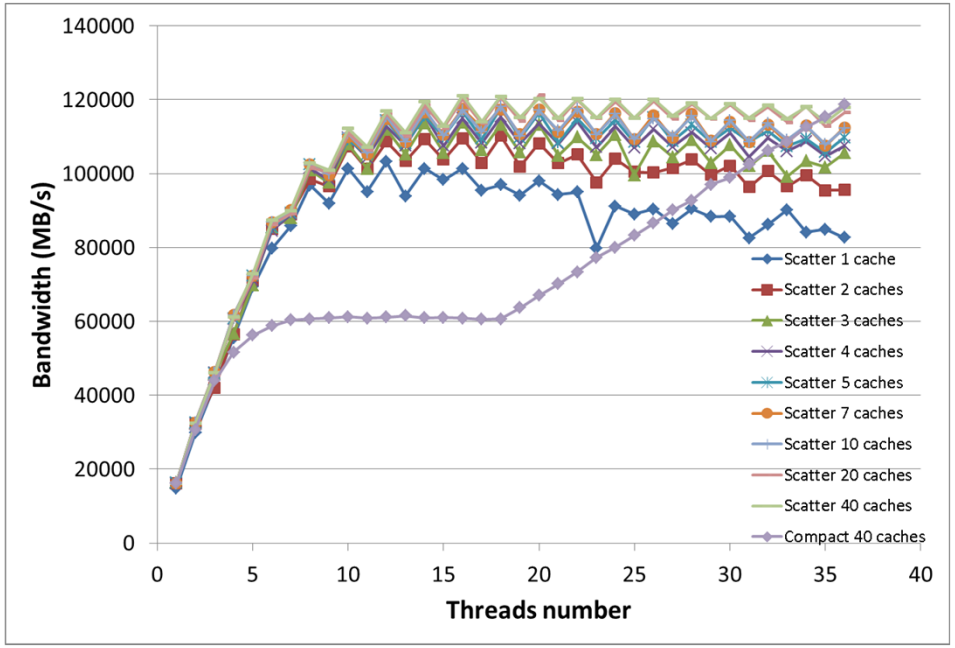

Helios 16 tasks per node
Marconi original 36 tasks per node
Marconi after first maintenance 32 tasks per node
Marconi after first maintenance 36 tasks per node
Marconi after second maintenance 36 tasks per node
Computation time (s)
16000
8000
4000
2
4
8
16
32
64
128
Node Number

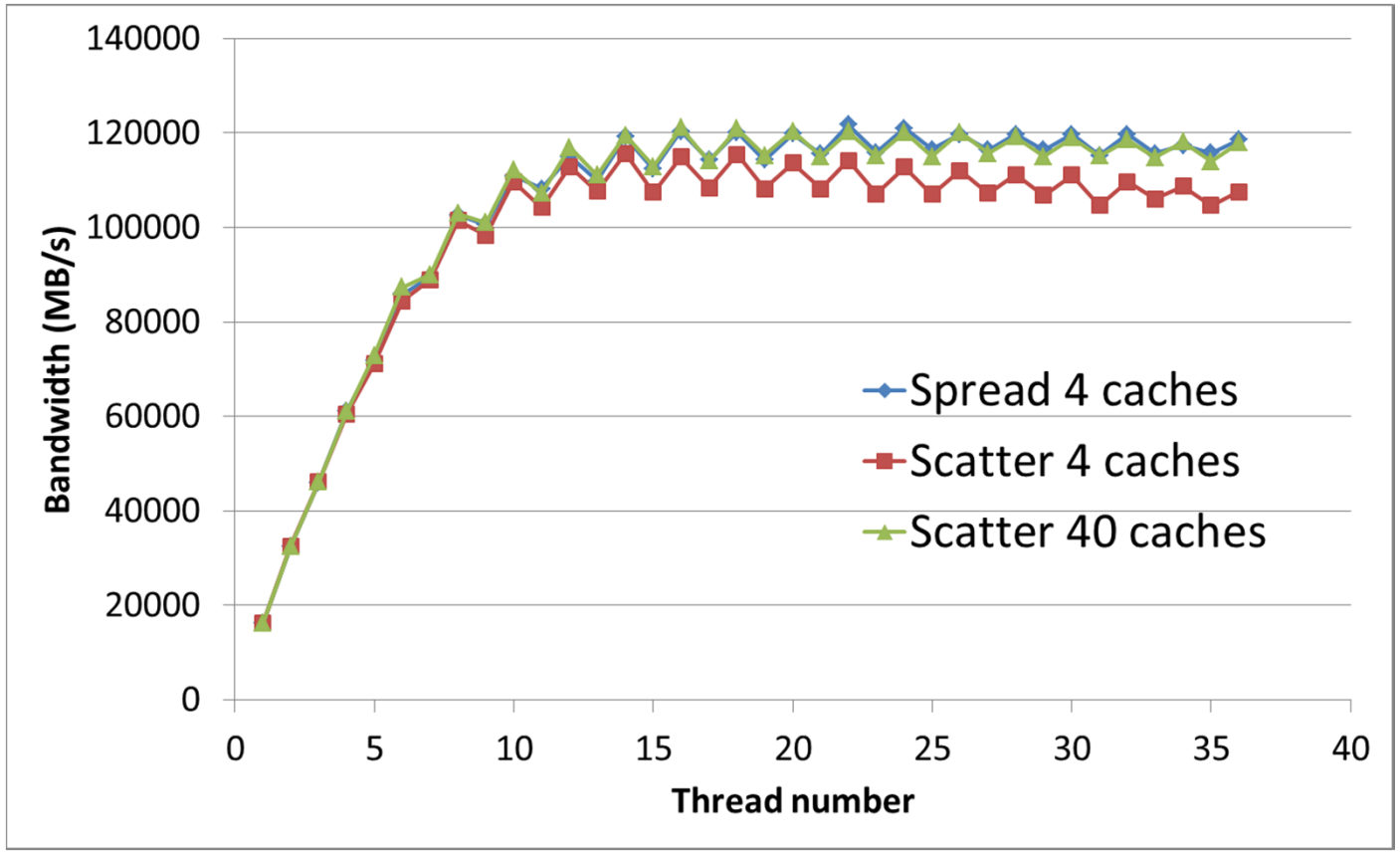

Helios 16 tasks per node
Marconi original 36 tasks per node
Marconi after first maintenance 32 tasks per node
Marconi after first maintenance 36 tasks per node
Marconi after second maintenance 36 tasks per node
Computation time (s)
16000
8000
4000
64
128
256
512
1024
2048
Core Number


140000
120000
100000
80000
60000
40000
20000
0
Bandwidth (MB/s)
0
5
10
15
20
25
30
35
40
Threads number
Marconi stream triad compact after maintenance
Marconi stream triad scatter after maintenance
Marconi stream triad compact before maintenance
Marconi stream triad scatter before maintenance


140000
120000
100000
80000
60000
40000
20000
0
Bandwidth (MB/s)
0
5
10
15
20
25
30
35
40
Threads number
Marconi compact after maintenance
Marconi scatter after maintenance


Bandwidth (MB/s)
Threads number
Marconi stream triad compact after maintenance
Marconi stream triad scatter after maintenance
Marconi stream triad compact before maintenance
Marconi stream triad scatter before maintenance


Bandwidth (MB/s)
Threads number
Scatter 1 cache
Scatter 2 caches
Scatter 3 caches
Scatter 4 caches
Scatter 5 caches
Scatter 7 caches
Scatter 10 caches
Scatter 20 caches
Scatter 40 caches
Compact 40 caches


140000
120000
100000
80000
60000
40000
20000
0
Bandwidth (MB/s)
0
5
10
15
20
25
30
35
40
Thread number
Spread 4 caches
Scatter 4 caches
Scatter 40 caches