EUROfusion

TSSV3 annual meeting

Alejandro Soba on behalf of the BSC-ACH team

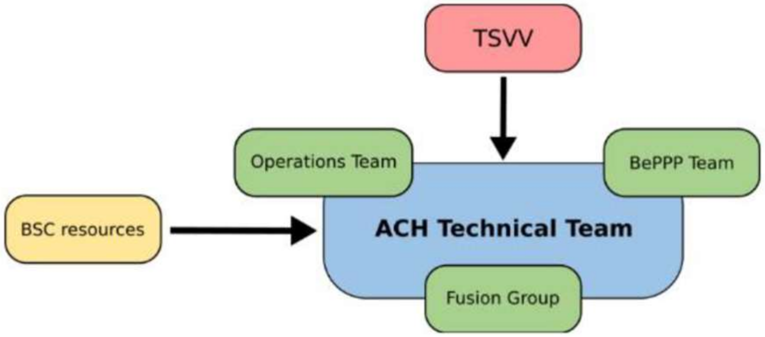
HPC ACHs | 29 May 2024

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

BSC-ACH team

ACH at CIEMAT / Barcelona Supercomputing Center (BSC), one of the 5 ACHs (IPP, EPFL, VTT, IFPiLM)

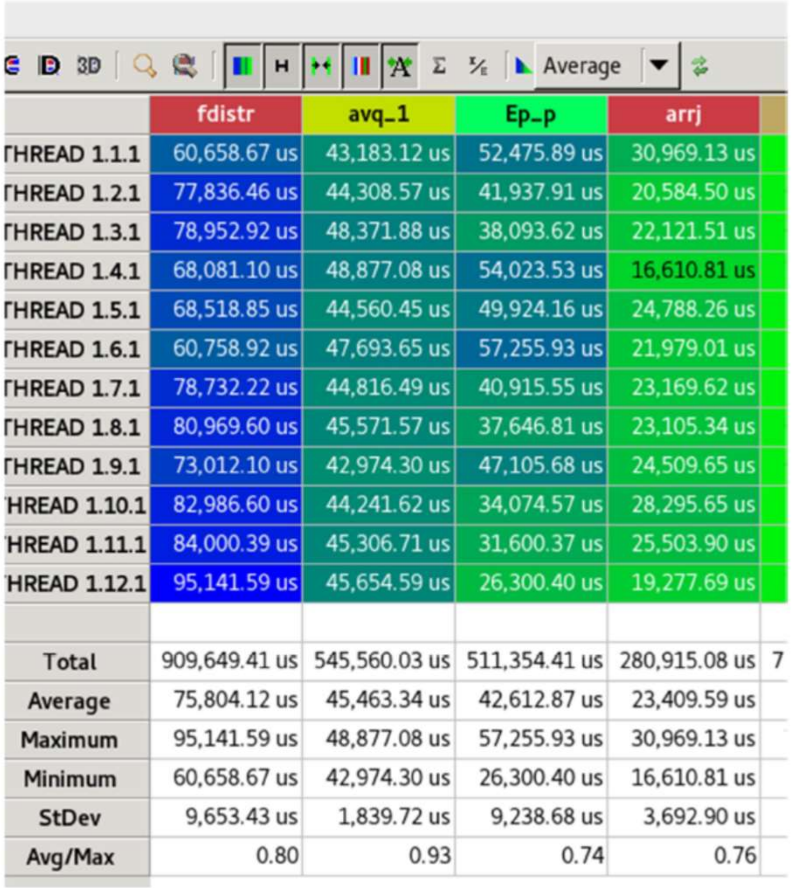
- BSC hosts one of ACHs in HPC
- Involves three groups at BSC:7 people in Fusion Group
- 2 people in Operations
- 2 people in Best Practices for Performance and Programmability (BePPP)
- Total effort: 328 pm in 2021-2025
- From 2024 on, at full size of ~100 pm/year
- We work with: JOREK, GENE-X, BIT1, ERO2, SOLPS, SPICE2, X-TORK, STELLA, SPEC, BOUT++ and KNOSOS.



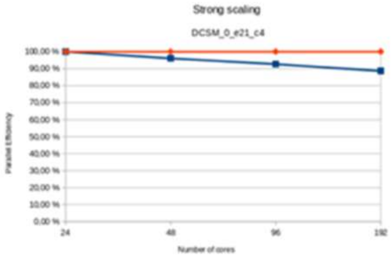
BIT1-firsts analysis. Paraver tool

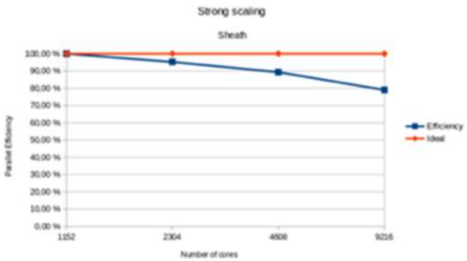
BIT1

- BIT1 is an electrostatic 1D3V PIC direct Monte Carlo code for plasma simulations used to edge plasma simulations.
- The first Analysis of performance was done by the Operation group.
- MareNostrum 4



	fdistr	avq_1	Ep_p	arrj	
THREAD 1.1.1	60,658.67 us	43,183.12 us	52,475.89 us	30,969.13 us	
THREAD 1.2.1	77,836.46 us	44,308.57 us	41,937.91 us	20,584.50 us	
THREAD 1.3.1	78,952.92 us	48,371.88 us	38,093.62 us	22,121.51 us	
THREAD 1.4.1	68,081.10 us	48,877.08 us	54,023.53 us	16,610.81 us	
THREAD 1.5.1	68,518.85 us	44,560.45 us	49,924.16 us	24,788.26 us	
THREAD 1.6.1	60,758.92 us	47,693.65 us	57,255.93 us	21,979.01 us	
THREAD 1.7.1	78,732.22 us	44,816.49 us	40,915.55 us	23,169.62 us	
THREAD 1.8.1	80,969.60 us	45,571.57 us	37,646.81 us	23,105.34 us	
THREAD 1.9.1	73,012.10 us	42,974.30 us	47,105.68 us	24,509.65 us	
HREAD 1.10.1	82,986.60 us	44,241.62 us	34,074.57 us	28,295.65 us	
HREAD 1.11.1	84,000.39 us	45,306.71 us	31,600.37 us	25,503.90 us	
HREAD 1.12.1	95,141.59 us	45,654.59 us	26,300.40 us	19,277.69 us	
Total	909,649.41 us	545,560.03 us	511,354.41 us	280,915.08 us	7
Average	75,804.12 us	45,463.34 us	42,612.87 us	23,409.59 us	
Maximum	95,141.59 us	48,877.08 us	57,255.93 us	30,969.13 us	
Minimum	60,658.67 us	42,974.30 us	26,300.40 us	16,610.81 us	
StDev	9,653.43 us	1,839.72 us	9,238.68 us	3,692.90 us	
Avg/Max	0.80	0.93	0.74	0.76	

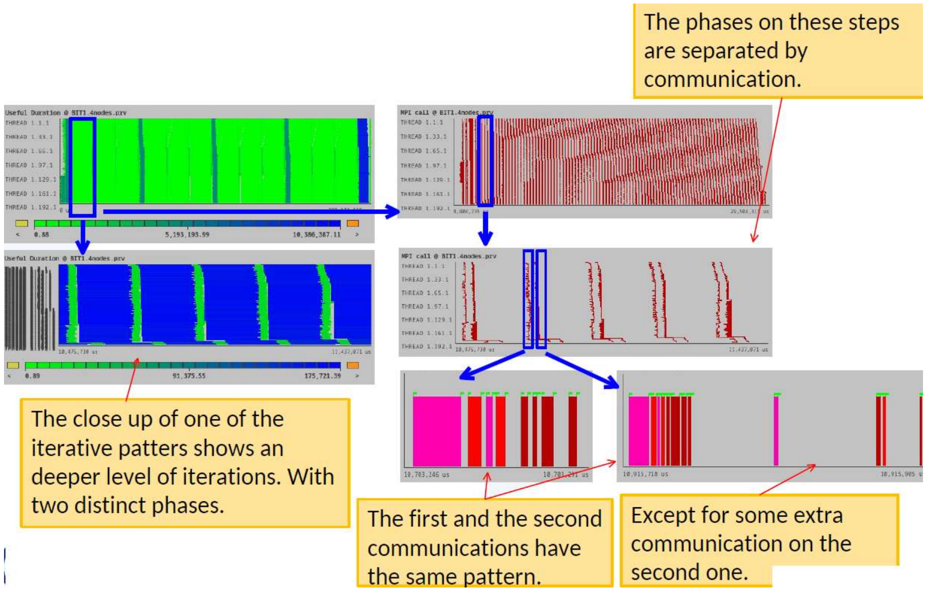




BIT1-firsts analysis

BIT1-MN4

- Intel Compiler 2020.1
- MPI: IMPI 2018.4
- Input file: Sheath_c8.inp

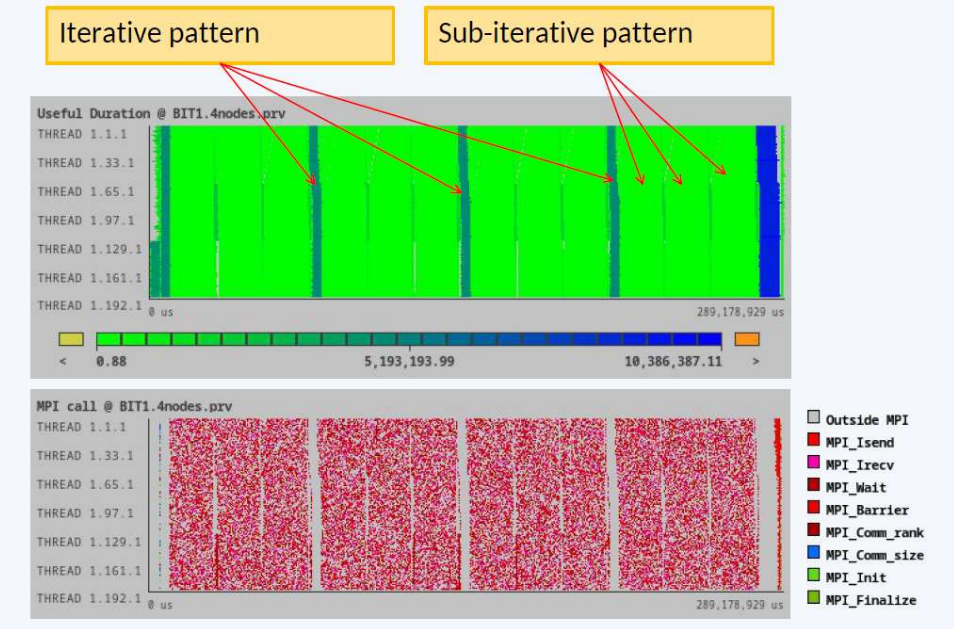


The phases on these steps are separated by communication.

The close up of one of the iterative patters shows an deeper level of iterations. With two distinct phases.

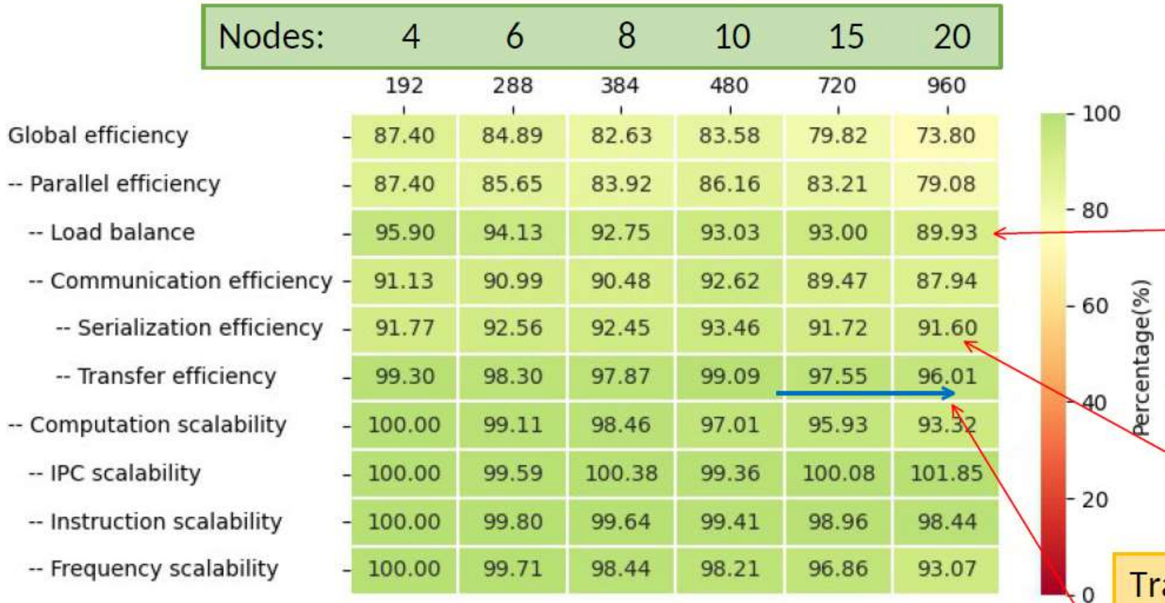
The first and the second communications have the same pattern.

Except for some extra communication on the second one.



BIT1-firsts analysis of efficiency

Input file: Sheath_c8.inp



Nodes:	4	6	8	10	15	20
	192	288	384	480	720	960
Global efficiency	87.40	84.89	82.63	83.58	79.82	73.80
-- Parallel efficiency	87.40	85.65	83.92	86.16	83.21	79.08
-- Load balance	95.90	94.13	92.75	93.03	93.00	89.93
-- Communication efficiency	91.13	90.99	90.48	92.62	89.47	87.94
-- Serialization efficiency	91.77	92.56	92.45	93.46	91.72	91.60
-- Transfer efficiency	99.30	98.30	97.87	99.09	97.55	96.01
-- Computation scalability	100.00	99.11	98.46	97.01	95.93	93.32
-- IPC scalability	100.00	99.59	100.38	99.36	100.08	101.85
-- Instruction scalability	100.00	99.80	99.64	99.41	98.96	98.44
-- Frequency scalability	100.00	99.71	98.44	98.21	96.86	93.07

Load balance gets worse as we scale.

Serialization is generally low.

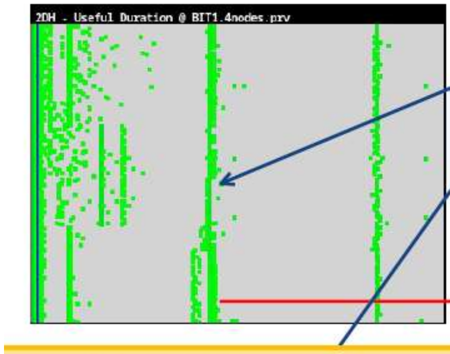
Transfer starts to show a tendency to drop. Might be a problem with more resources.

Parallel efficiency is low, mainly due to serialization. But the Load Balance and Transfer Efficiency tendency to drop, cause a worse Parallel efficiency when scaling.

BIT1-firsts analysis of load balance

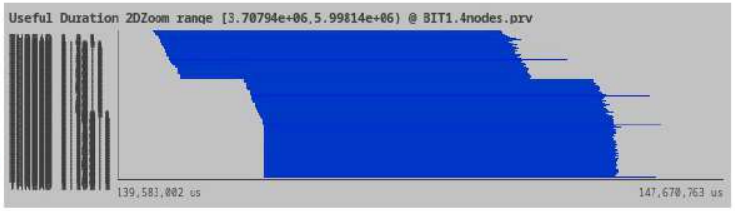
Input file:

Sheath_c8.inp

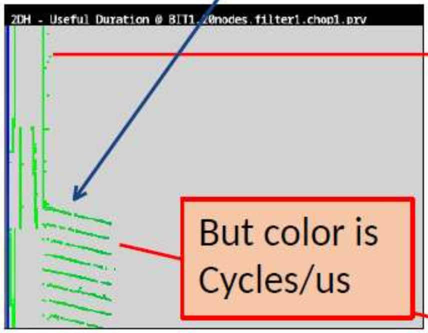


Same region does not have load imbalance with 4 nodes. But does at 20 nodes.

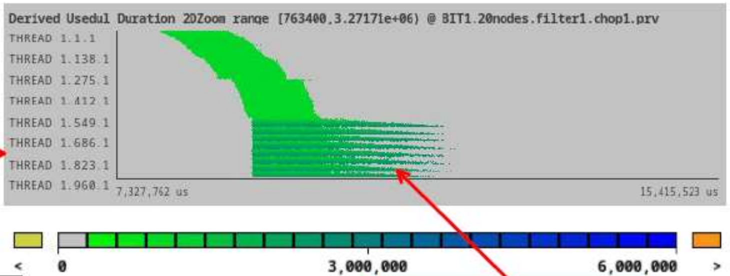
This useful dur. range


Useful Duration 2DZoom range [3.70794e+06,5.99814e+06) @ BIT1.4nodes.prv
139,581,002 us
147,670,763 us

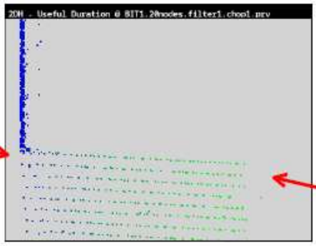
Sheath_c8, 20 node execution



This useful dur. range



But color is Cycles/us



Load Balance is correlated with Cycles/us. Input / Output?

Load imbalance appears in this region. At the 8 nodes with higher ranks.

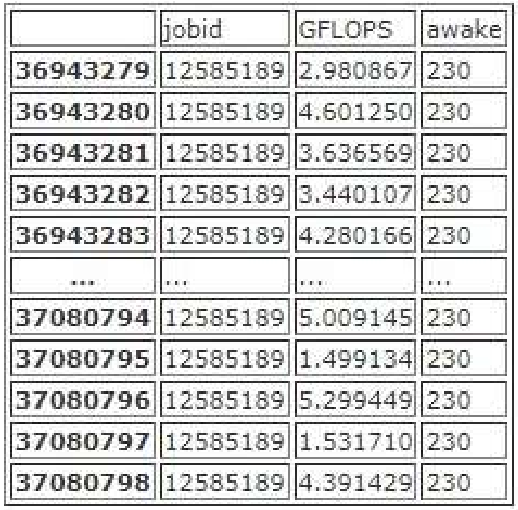
BIT1-Efficiency in MARCONI

- In several execution of BIT1 in MARCONI cluster, tested using **HPCMD** monitoring tool, during several months of 2023 was detected a low efficiency measure in Gflops.
- BIT1 resource-intensive jobs: utilizing 192 nodes for 24 hours, resulting in 4600 node-hours. Inputs filed N_aELM_A, N_CU, 9_CU.
- Vectorial operations were not founded in the reported files

- Table of example by sockets shows an efficiency of 0.19%

Job_comp_eff (%) = GF / (peak) * 100

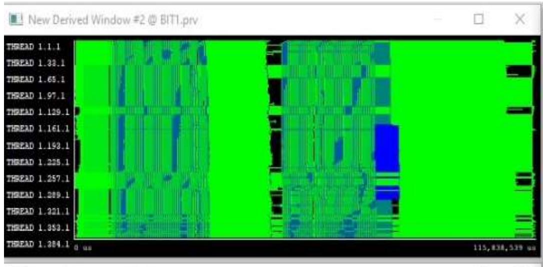
Peak in MARCONI is aprox. 3200 Gflops

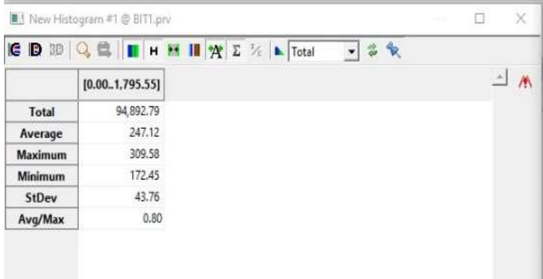


	jobid	GFLOPS	awake
36943279	12585189	2.980867	230
36943280	12585189	4.601250	230
36943281	12585189	3.636569	230
36943282	12585189	3.440107	230
36943283	12585189	4.280166	230
...	...	...	...
37080794	12585189	5.009145	230
37080795	12585189	1.499134	230
37080796	12585189	5.299449	230
37080797	12585189	1.531710	230
37080798	12585189	4.391429	230

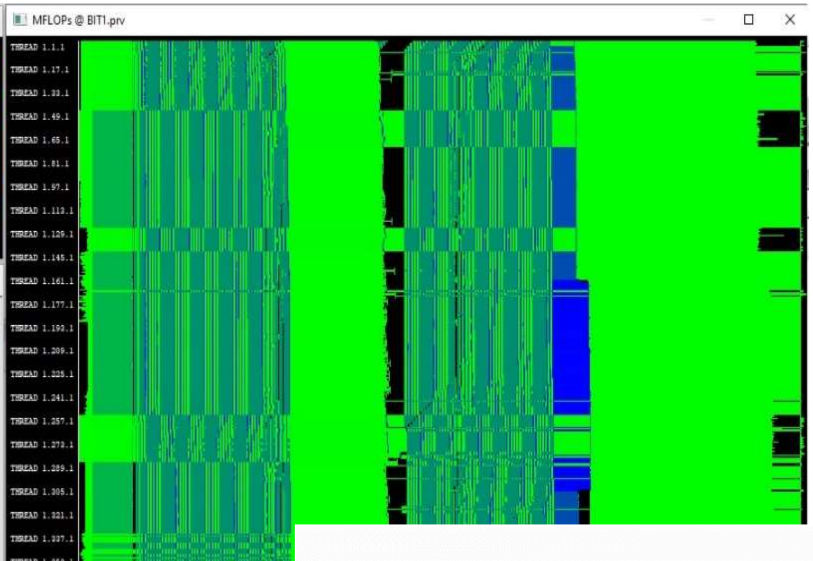
BIT1-Efficiency in MN4

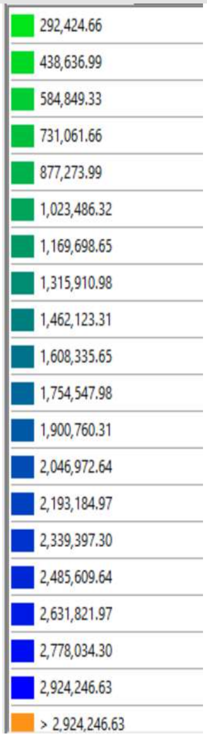
Extrae/Paraver analysis


New Derived Window #2 @ BIT1.prv

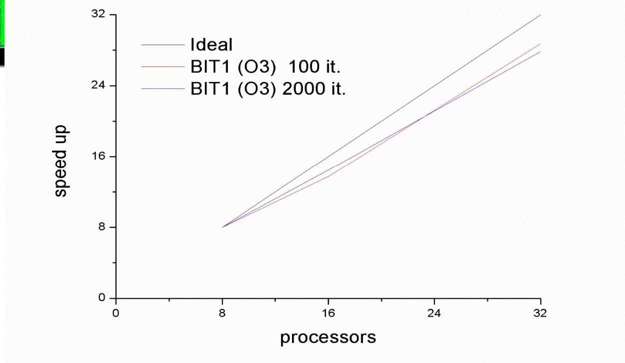

New Histogram #1 @ BIT1.prv

	[0.00..1,795.55]
Total	94,892.79
Average	247.12
Maximum	309.58
Minimum	172.45
StDev	43.76
Avg/Max	0.80


MFLOPs @ BIT1.prv


🟩 292,424.66
🟩 438,636.99
🟩 584,849.33
🟩 731,061.66
🟩 877,273.99
🟩 1,023,486.32
🟩 1,169,698.65
🟩 1,315,910.98
🟩 1,462,123.31
🟦 1,608,335.65
🟦 1,754,547.98
🟦 1,900,760.31
🟦 2,046,972.64
🟦 2,193,184.97
🟦 2,339,397.30
🟦 2,485,609.64
🟦 2,631,821.97
🟦 2,778,034.30
🟦 2,924,246.63
🟧 > 2,924,246.63

8 nodes (100 it.) [0.34%]
(2000 it) [0.61%]

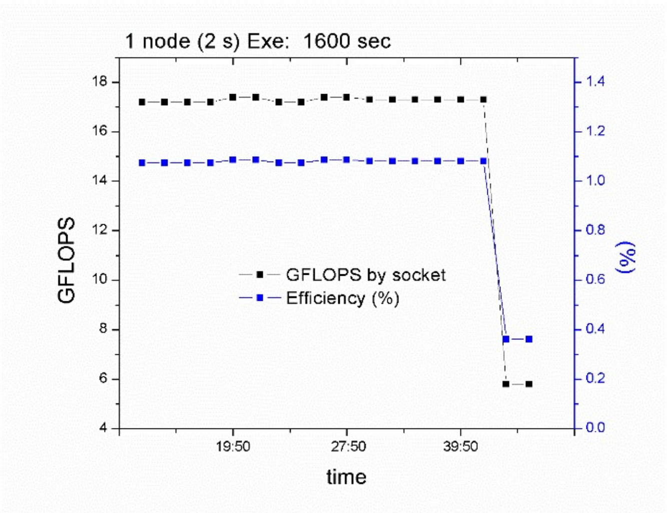


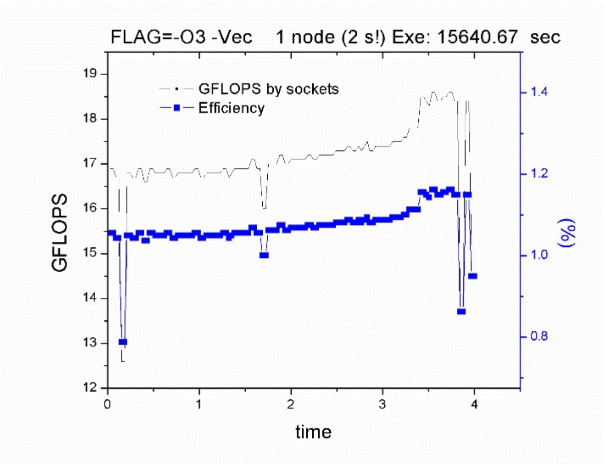
BIT1 with -O3 flag. We identify Vectorial instructions using Objdump tool. (Not with –O2 flag)

BIT1-Efficiency in MARCONI

HPCMD tool in MARCONI

Compilation FLAGs	case	Nodes/sockets	Time (sec)
-O2	DCSM_0_e21_c8.inp	1 /2	~15200
-O3 - Vect	DCSM_0_e21_c8.inp	1 /2	~1600
-O3 - Vect	DCSM_0_e21_c8.inp	1 /2	~15400
-O2	BIT1_N_CU.inp	64/128	~16800
-O3	BIT1_N_CU.inp	64/128	~1890
-O3	BIT1_N_CU.inp	64/128	~17200
-O3 - Vect	BIT1_N_CU.inp	64/128	~1895
-O2	BIT1_N_CU.inp	128/256	~1075
-O3 - Vect	BIT1_N_CU.inp	128/256	~9700
-O3	BIT1_N_CU.inp	192/384	~1600

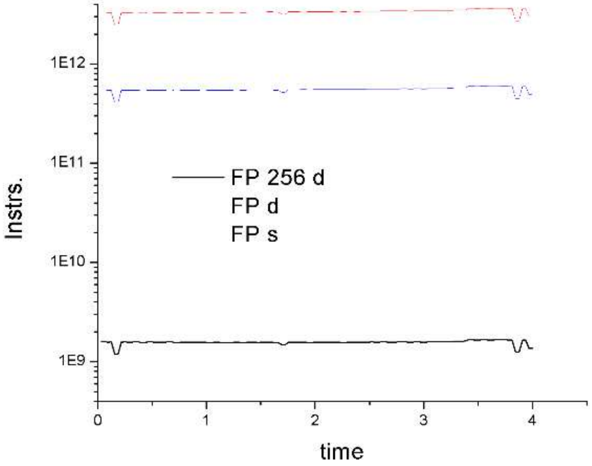


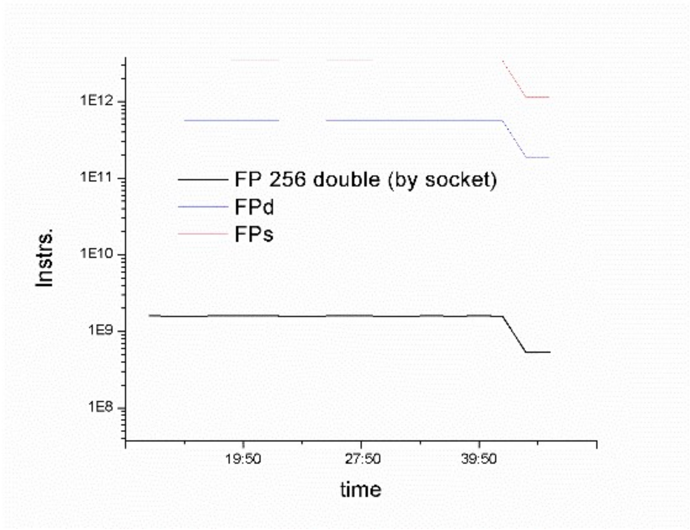


BIT1-Efficiency in MARCONI

HPCMD tool in MARCONI

With –O3 –Vect flags we identify vectorial operations

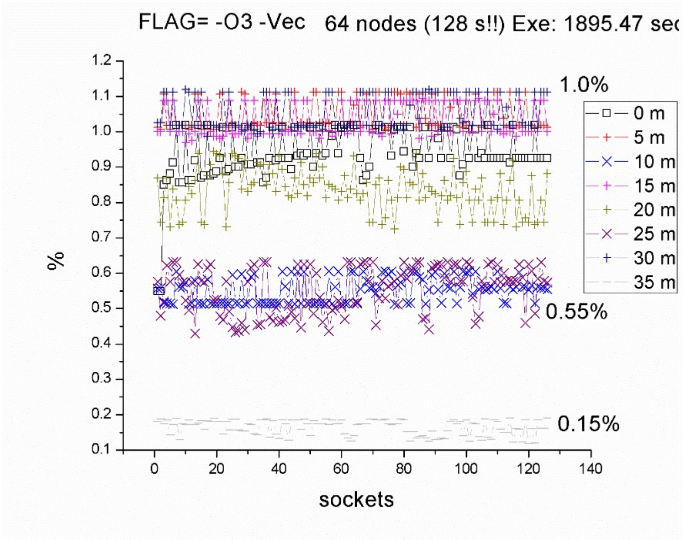


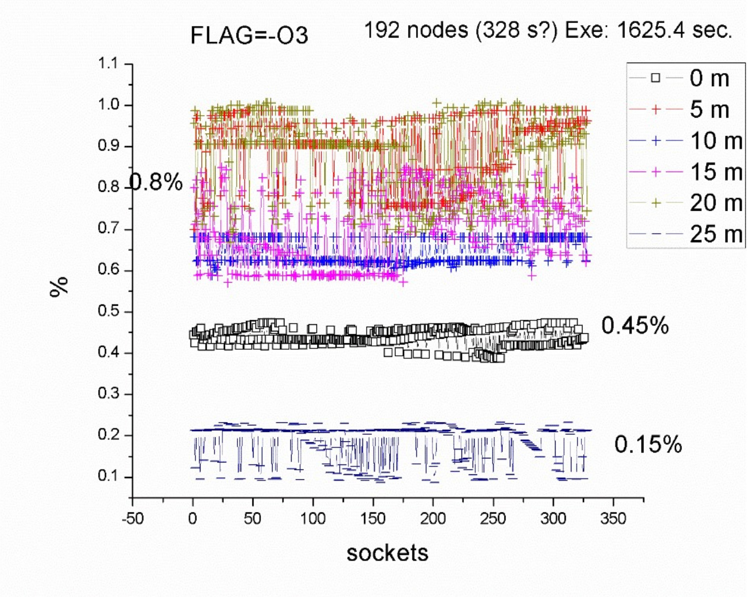


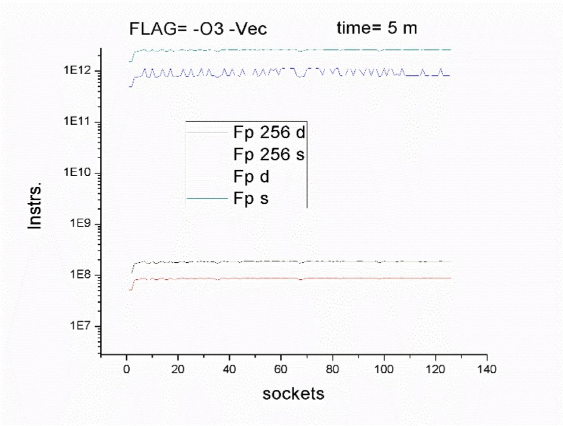
BIT1-Efficiency in MARCONI

HPCMD tool in MARCONI

Cases with 64 nodes and 192 nodes.

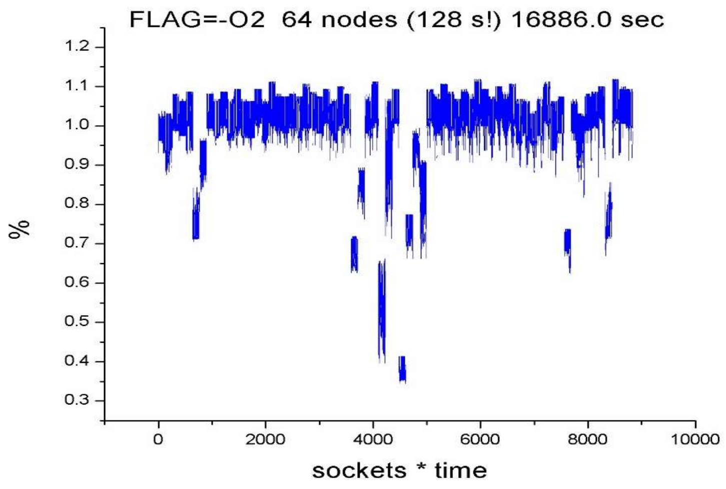




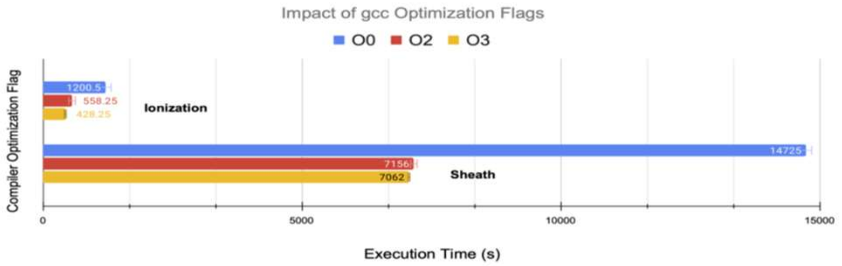


BIT1-Efficiency in MARCONI

HPCMD tool in MARCONI



More time steps Continues the stable efficiency.

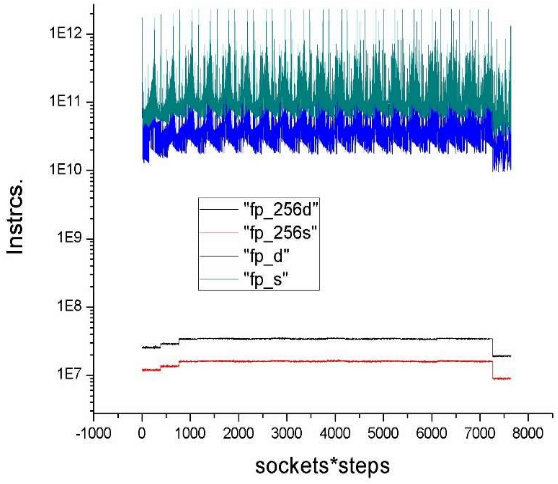


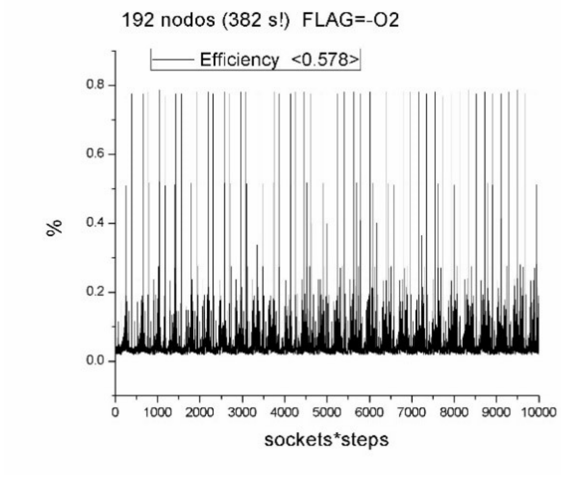
Not relevant influence of the O2 or O3 option.

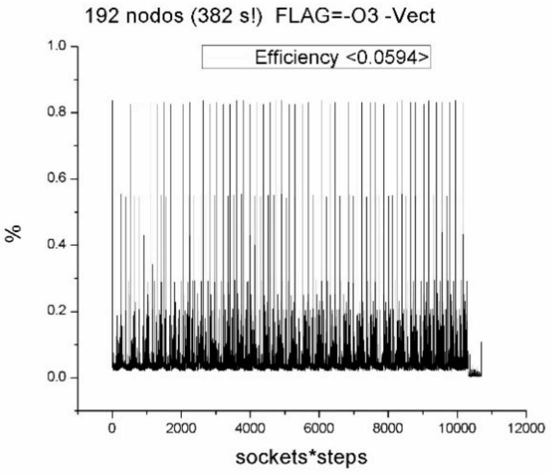
BIT1-Efficiency in MARCONI

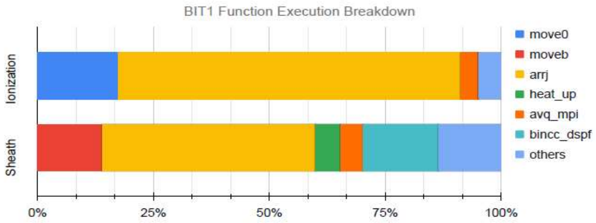
HPCMD tool in MARCONI

Restart files! We reinitiate from
time step>1 171 000 000







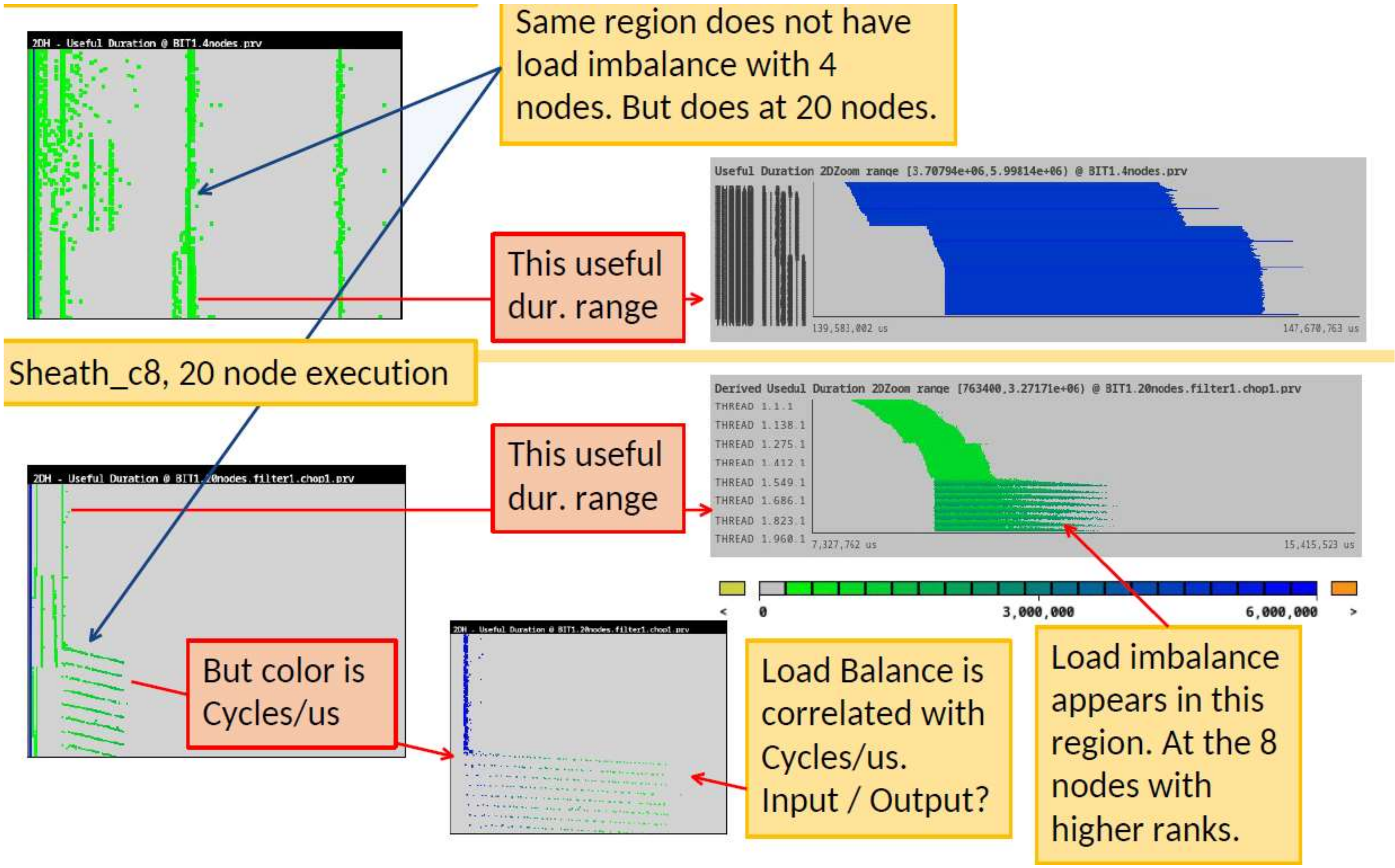


Influence of the reordering function grows with time steps

BIT1-Efficiency in MN4

Previous test

in MN4


2DH - Useful Duration @ BIT1.4nodes.prv
Useful Duration 2DZoom range [3.70794e+06,5.99814e+06) @ BIT1.4nodes.prv
139,583,002 us
147,670,763 us
Derived Usedul Duration 2DZoom range [763400,3.27171e+06) @ BIT1.20nodes.filter1.chop1.prv
THREAD 1.1.1
THREAD 1.138.1
THREAD 1.275.1
THREAD 1.412.1
THREAD 1.549.1
THREAD 1.686.1
THREAD 1.823.1
THREAD 1.960.1
7,327,762 us
15,415,523 us
<
0
3,000,000
6,000,000
>
2DH - Useful Duration @ BIT1.20nodes.filter1.chop1.prv
2DH - Useful Duration @ BIT1.20nodes.filter1.chop1.prv

Same region does not have load imbalance with 4 nodes. But does at 20 nodes.

This useful dur. range

Sheath_c8, 20 node execution

This useful dur. range

But color is Cycles/us

Load Balance is correlated with Cycles/us. Input / Output?

Load imbalance appears in this region. At the 8 nodes with higher ranks.

BIT1-GPU tests

- The four most time-consuming subroutines are "fdistr", "avq_1", "Ep_p" and "arrj".
- dfdiag.c, that call subroutine fdistr, is a diagnostic function. In order to generate a “clean code” we eliminate this call. Ep_p is part of this calls.
- Subroutine avq_1() (exist several instances of avq)
- Arranger.c is arrj() we use the arranger_1.c. Is the reordering particles function.
- We starts using OPENAcc paradigm.

Starting using OPENACC directives in CTE-Power

CTE-POWER 9

2 login node and 52 compute nodes, each of them:

- **2 x IBM Power9 8335-GTH @ 2.4GHz (3.0GHz on turbo, 20 cores and 4 threads/core, total 160 threads per node)**
- **512GB of main memory distributed in 16 dimms x 32GB @ 2666MHz**
- **2 x SSD 1.9TB as local storage**
- **2 x 3.2TB NVME**
- **4 x GPU NVIDIA V100 (Volta) with 16GB HBM2.**
- **Single Port Mellanox EDR**
- **GPFS via one fiber link 10 GBit**

BIT1-GPU tests

```
#pragma acc data
copyin(iisp,nsp,ng,limit1,tesca,eesca,qesca,bmag,sn,cs,np[0:nsp][0:ng+1],vx[0:nsp][0:ng+1][0:limit1],vy[0:nsp][0:ng+1][0:…
#pragma acc data
copy(Vxef[0:ng],Vyef[0:ng],Vzef[0:ng],Tx[0:nsp][0:ng],Qh[0:nsp][0:ng],Qtot[0:nsp][0:ng],T_dif[0:nsp][0:ng],Vx[0:nsp][0:ng…
{
                                        void avq_1(int iisp)
 #pragma acc data copyin(iisp,nsp,ng,nc,limit1,np[0:nsp][0:ng+1],x[0:nsp][0:ng+1][0:limit1])
 copy(srho[0:nsp][0:ng])
 {
 #pragma acc parallel loop
 for (j=0; j<=nc; j++)
 {
     #pragma acc loop seq
     for (i=np[iisp][j]-1; i>=0; i--)     /* Accumulate n and V */
     {
         srho[iisp][j]   += 1.0-x[iisp][j][i];  }   }
```

BIT1-GPU tests

BIT1 is a one-dimensional PIC code, which means that it treats a big number of particles in big domains, and for each of them a relatively low number of calculi must be performed.
Not so good to use in a GPU.

Using OPENAcc, the computing time increase **four times** aprox. IN CTE-power & Leonardo
Problem: we can´t copy to the GPU in each time step this amount of data.

Solution: copy the data out of the time loop, and them, make the whole operation into the GPUs avoiding MPI communications.

New problem: in a small case, this can be do it. But in the big cases, there are not enough GPUs memory to do it.

Solution: Divide the data into several GPUs, and incorporate GPUs communications.

But: I afraid that we will have the same problem that we have in MPI case.

BIT1

Conclusions

Efficiency in MARCONI and MN4:

- new FLAGS to turn on vectorization, but not relevant inf efficiency measures. We are exploring other possibilities.
- For “few” time steps (less than millions) the behavior is acceptable
- For high time steps (>1 170 000 000) very low efficiency. Disbalance of nonlinear plasmas.
- We get an input file to be run with big disbalance from first stages to analyze and try to improve efficiency.
- A new load balanced version of BIT1 is under development.
- Last week we received a new input file with in-balance from the very beginning.

GPU porting

- The first tests were not satisfactory. Even using a version avoiding spurious communication and diagnostics.
- A version of BIT1 for work in GPU will need a considerable reformulation of the engineering of the code to reach improvements.
- We need to test MN5
- Other studies shows improvements using inputs files for a few nodes.
- We are exploring options with better manage of memory in OpenAcc.