



Performance Evaluation of OpenFOAM on Juelich Supercomputing Facilities (JURECA, JUWELS and JUSUF)

14 OCT. 2020 | ABOUZAR GHASEMI, PAUL GIBBON, STEVEN B. BEALE

OUTLINE

- Introduction
- Experimental setup
- Strong scaling study on JURECA and JUWELS
- Scalability on JUWELS for the cases without output
- Scalability on JUWELS for the cases with one output
- Strong scaling study on JUSUF
- Summary

Experimental setup

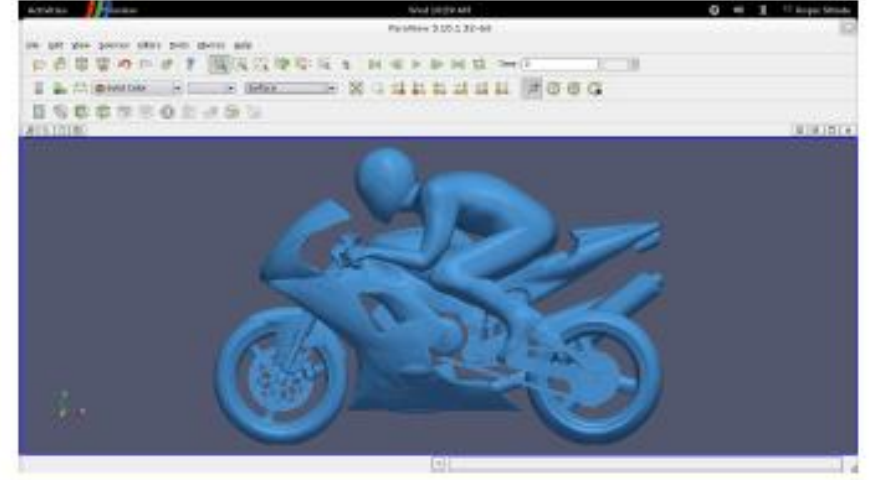
	Compiler	MPI	cases Node num.	Rank per node	Profiler
JURECA	Intel_2018.2	IntelMPI_2018.2	2,4,8,16, 32 48,64,96,128	24	VTune/APS, IPM
JUWELS	Intel_2018.2 Intel_2019.5	IntelMPI_2018.2 IntelMPI_2019.6,OpenMPI_4.0.2, PSMPI/5.4	2,4,8,16, 24 32,48,64	48	Vtune/APS,IPM
JUSUF	Intel_2019.5	IntelMPI_2019.6, PSMPI/5.2	1,6,12,24	128	IPM

-O3 -xHost -ftz -fp-speculation=safe -fp-model source -g -no-prec-div -sox

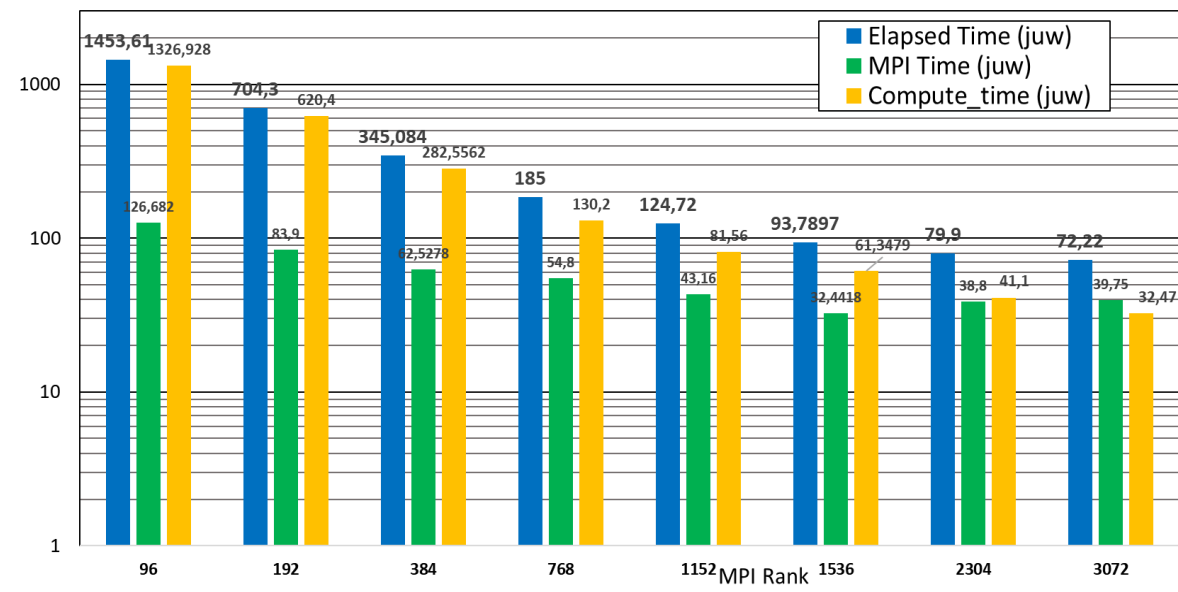
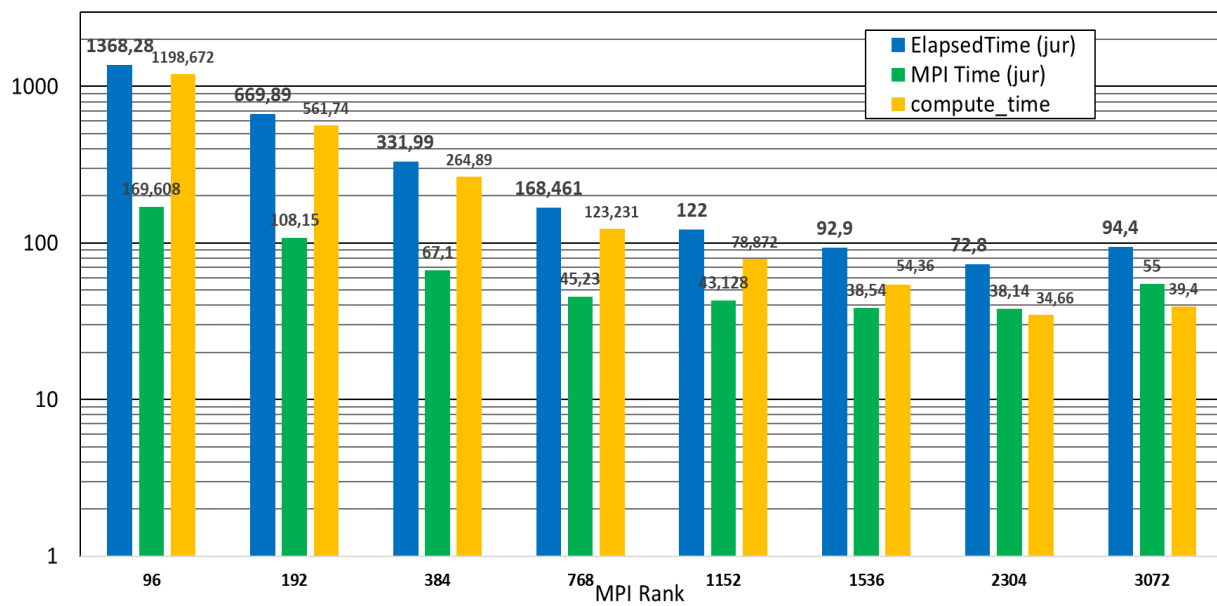
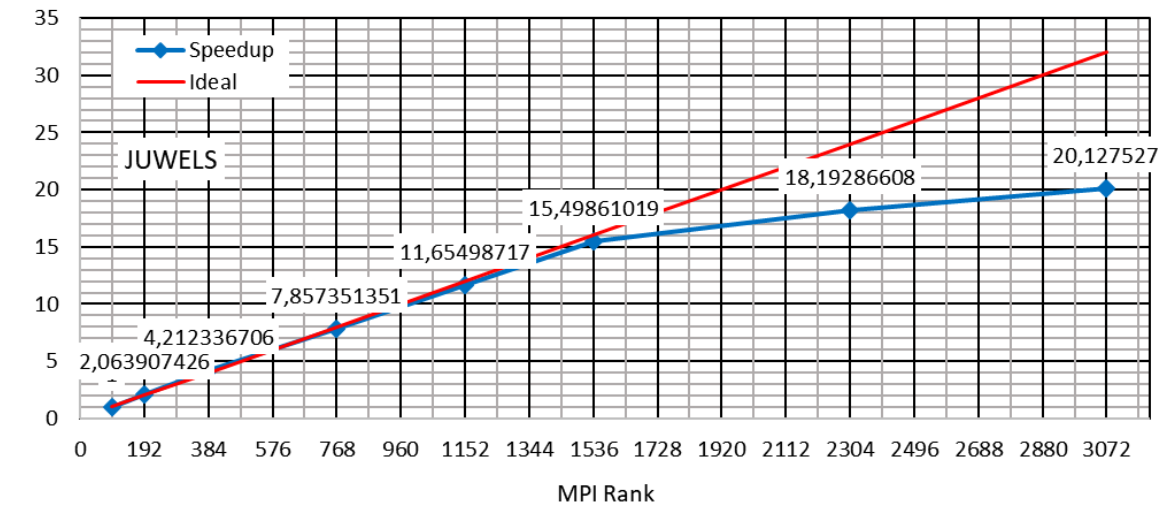
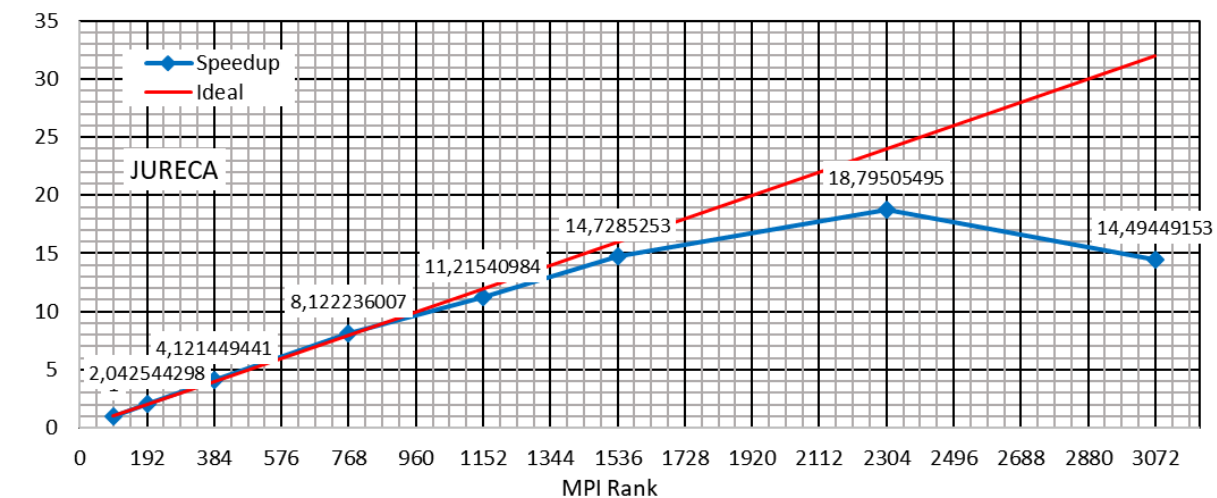
JURECA was built in 2015 and utilizes Haswell processor with 24 cores per node,
JUWELS was built in 2018, based on a Skylake processor and with 48 cores per node,
JUSUF was built in 2020, based on AMD EPYC 7742 with 128 cores per node.

Simulation with and without output

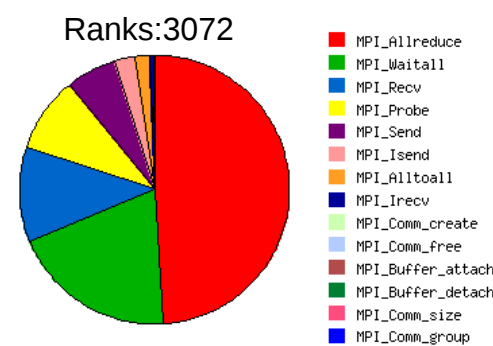
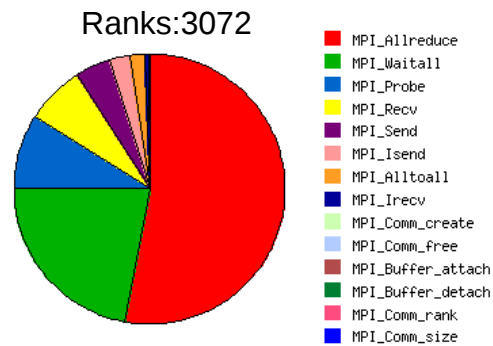
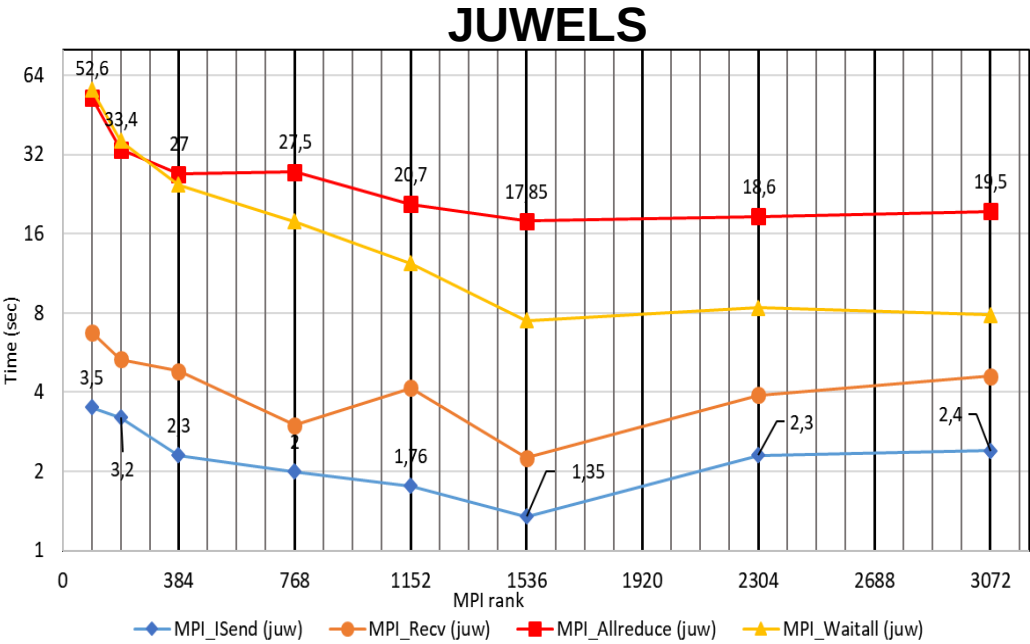
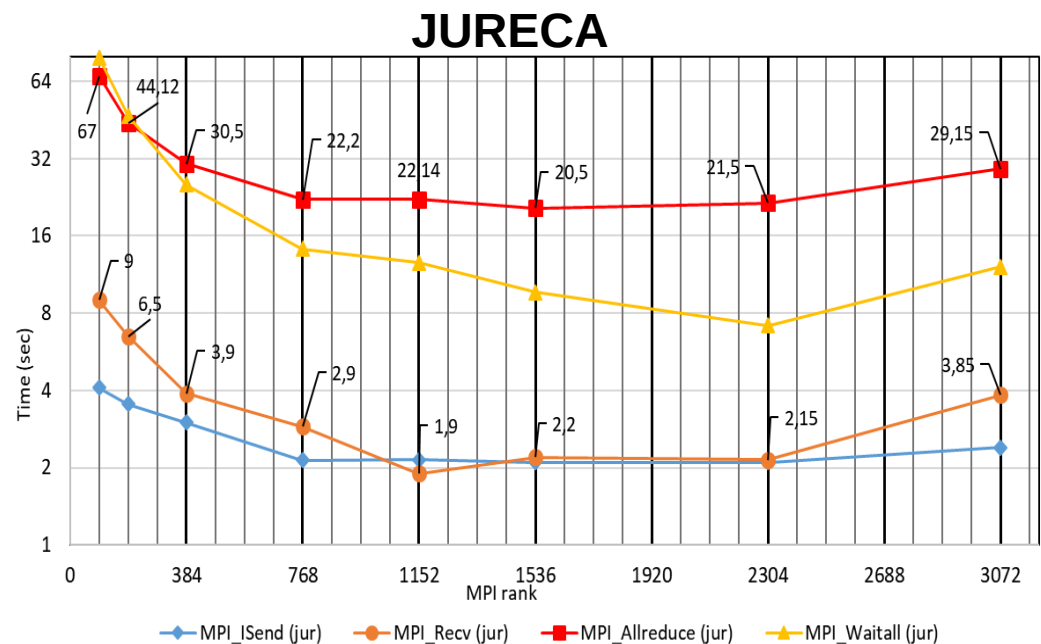
- ✓ OpenFoam-v1912
- ✓ SCOTCH/PT-SCOTCH is used for domain decomposition.
- ✓ **Strong scaling study** for the **motorbike test** case from OpenFoam tutorial for the cases with **51 M cells**.
- ✓ **Simulation with no output:** writeInterval larger than endTime (**writeInterval > endTime=200**).
- ✓ Simulation with **One output:** writeInterval= endTime=200
 - *#include "ensightWrite" for the fields (U p "(k|epsilon|omega)");*
 - *#include "adiosWrite" for the fields (U p "(k|epsilon|omega)");*
- ✓ MPI is linked internally within the executable at run time, rather than at link time. This is not consistent with profiling tool Score-p.



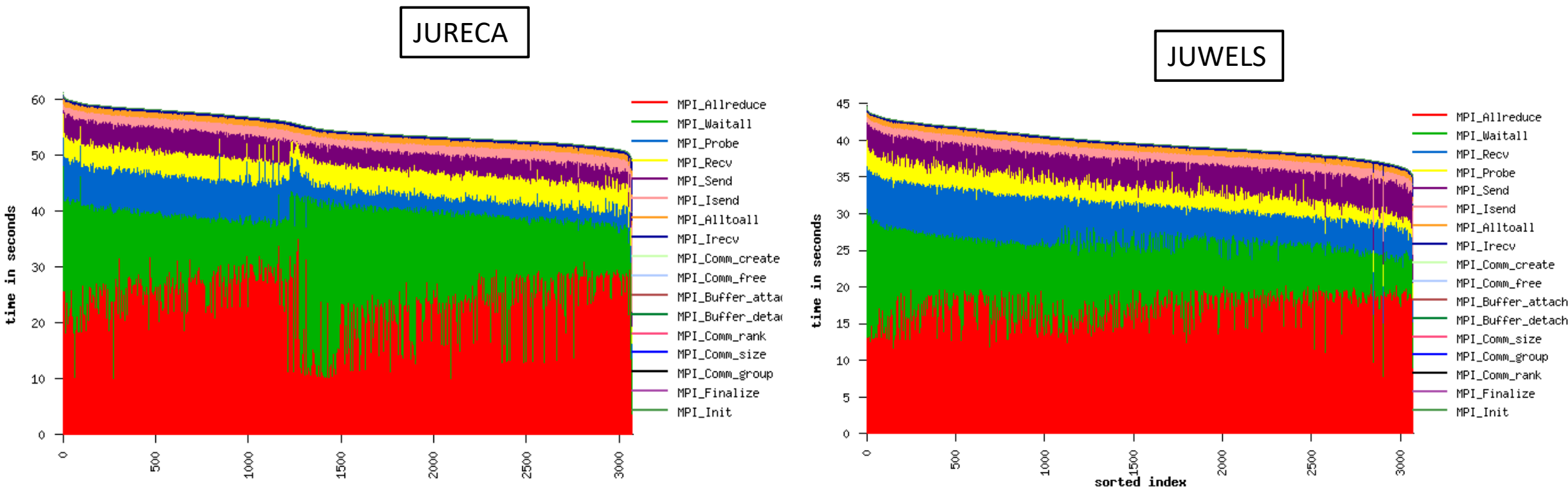
Strong scaling study on JURECA and JUWELS for the cases with no output



Time spent in most time consuming MPI functions

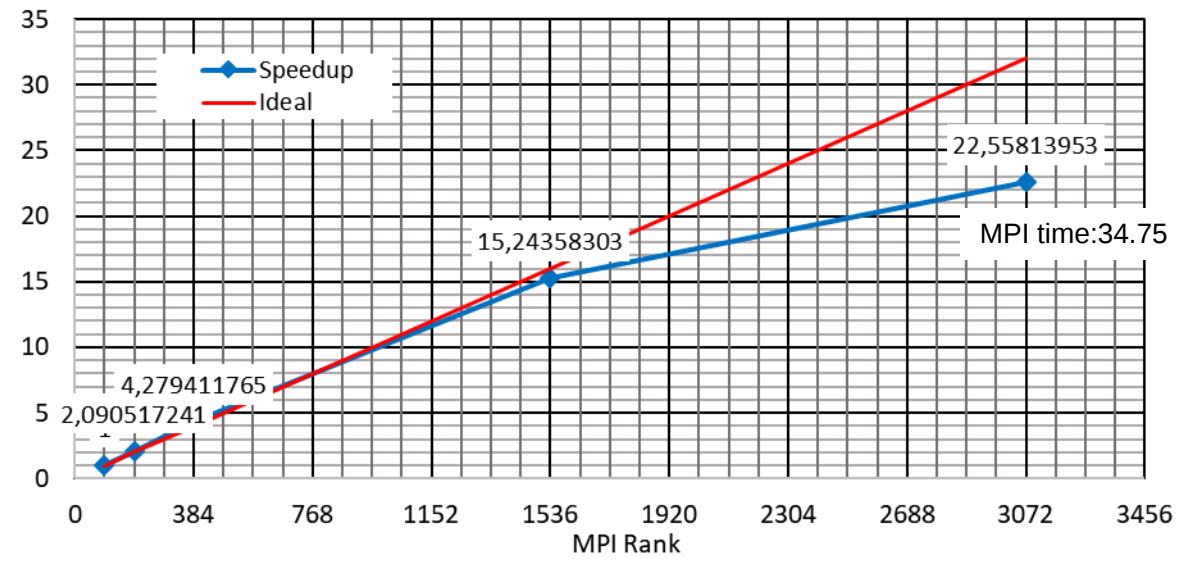


Load Imbalance for the cases with no output

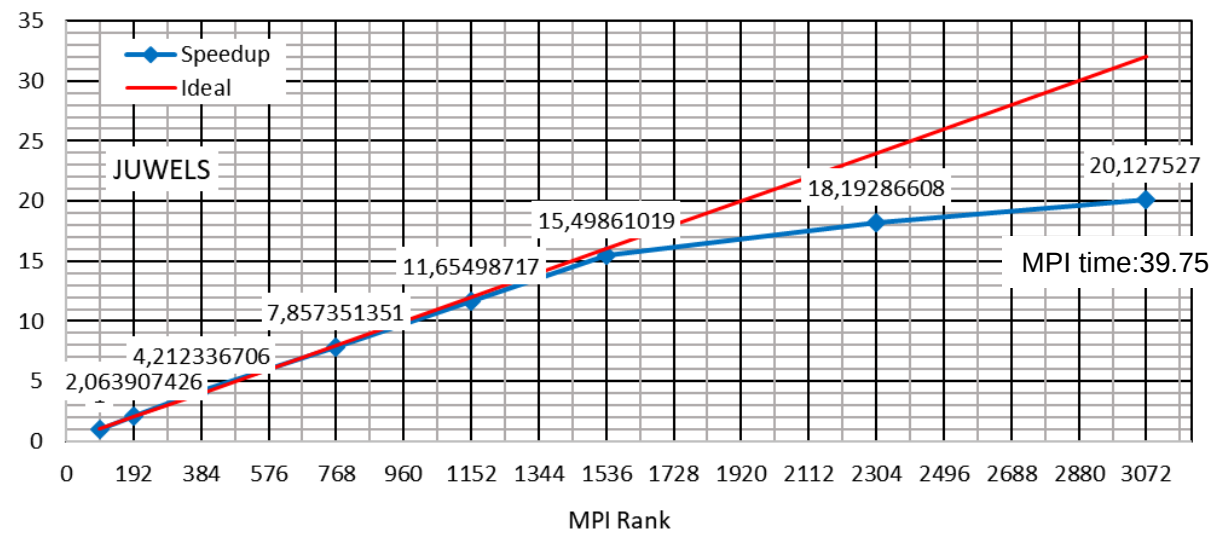


Case with 3072 MPI Ranks

Performance of IntelMPI_2019.6 and IntelMPI_2018.2 on JUWELS for the cases with no output

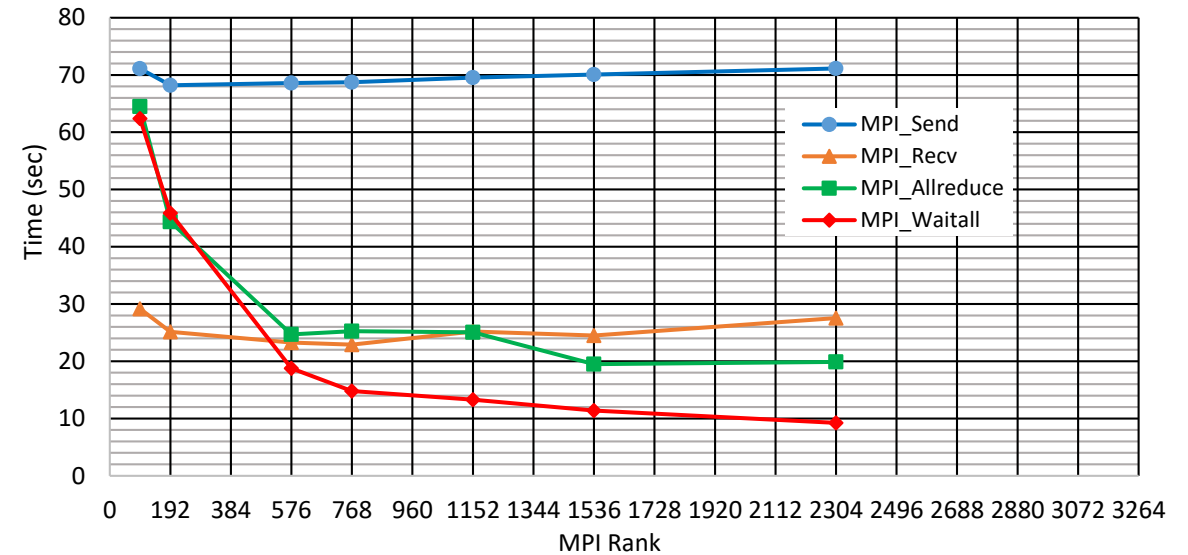
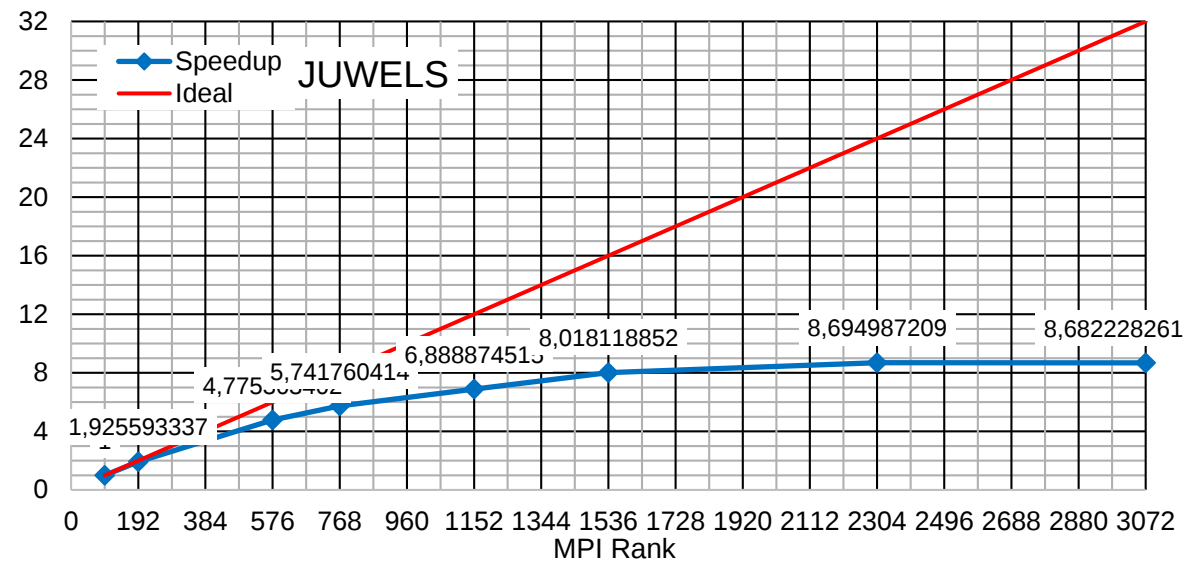


Intel_2019.5;
IntelMPI_2019.6



Intel_2018.2;
IntelMPI_2018.2

Strong scaling study on JUWELS for the cases with one output generated using “ensightWrite”



Intel_2018.2; IntelMPI_2018.2

```
#include "ensightWrite" for the fields (U p "(k|epsilon|omega)");
```

Impact of generation of one output on the elapsed time

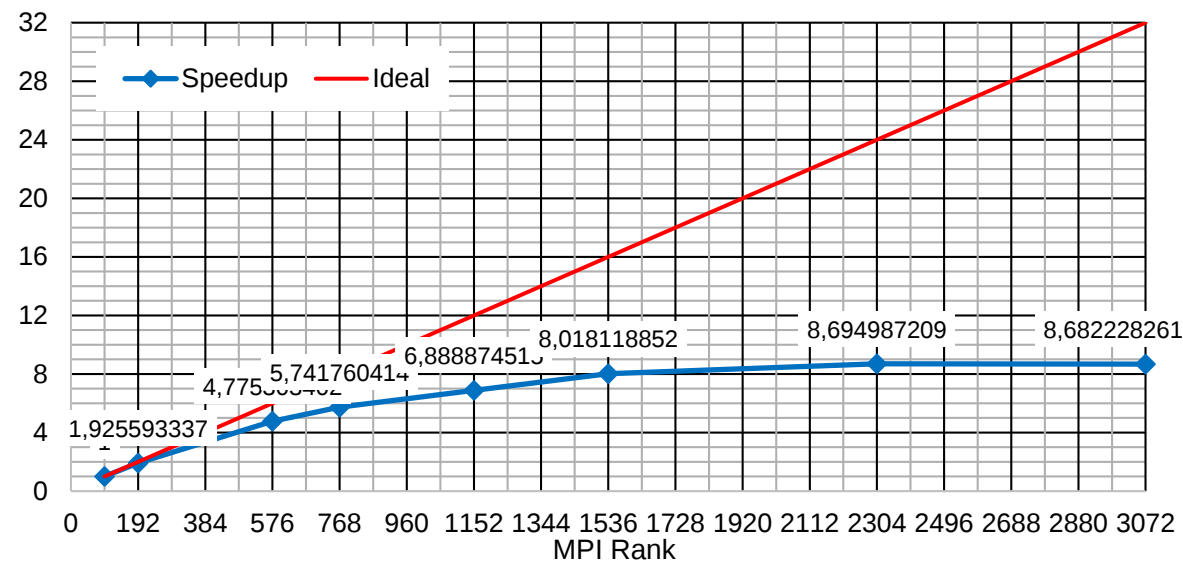
case with 1536 MPI ranks

N. Of ranks		Elapsed time	MPI time
No output	1536	93	38,5
1 output	1536	199,2	139

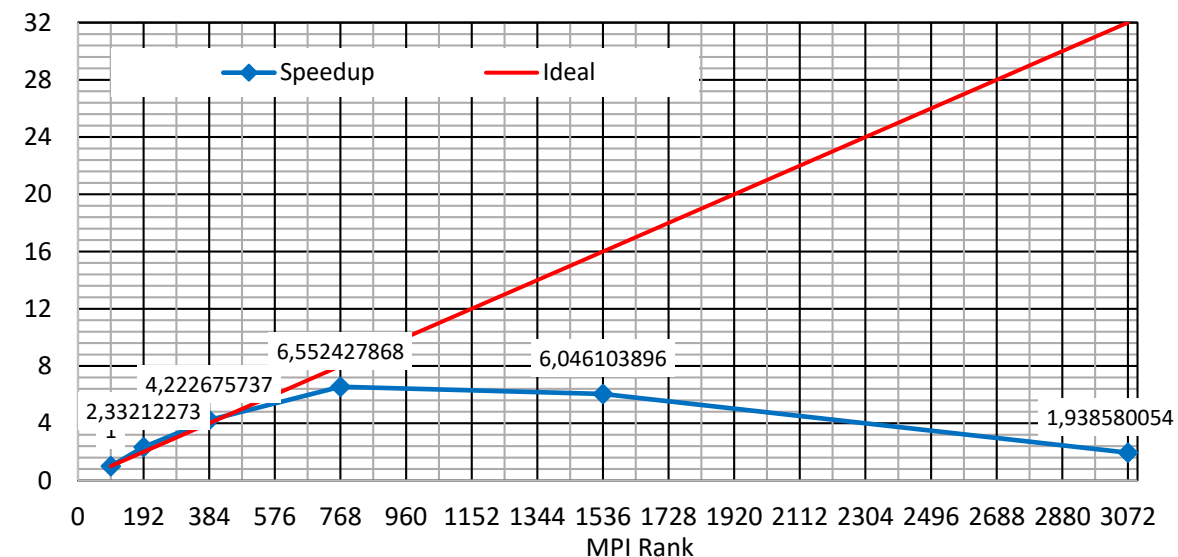
	Time (no Output)	Num. Calls (no Output)	Time (1 Output)	Num. Calls (1 Output)
Send	1.45	6572870	70.06	14667904
Isend	1.2	1946291961	1.2	1956809868
Allreduce	18.14	128792064	19.50	129796608
WaitAll	7.78	145659374	11.39	146629297
Recv	3.49	6572870	24.5	14688438

Intel_2018.2; IntelMPI_2018.2

Strong scaling study on JUWELS for the cases with one output



Intel_2018.2; IntelMPI_2018.2



Intel_2019.5; IntelMPI_2019.6

Performance of different MPI implementations for the cases with one output

N. of ranks	IntelMPI_2019.6		OpenMPI_4.0.2		PSMPI_5.4		IntelMPI_2018.2	
	Elapsed Time	MPI Time	Elapsed Time	MPI Time	Elapsed Time	MPI Time	Elapsed Time	MPI Time
768	285	152	295,6	156	-----	-----	278,2	144,2
1536	308	248	284,5	218,7	241.9	185	199,2	139
3072	960	930,6	471,3	437	219	189	192	159

IntelMPI_2018.2: UCX_TLS=ud_x,sm,self; with rc_x elapsed time is 5% less.

IntelMPI_2019.6: mpi-settings/large-job-mpi (UCX_TLS=ud_x,sm,self)

PSMPI_5.4: mpi-settings/UCX-RC (UCX_TLS=rc_x,sm,self; PSP_UCP=1)

OpenMPI_4.0.2: UCX_TLS=all

UCX_TLS=all → hanging

Time spent on MPI functions for different MPI implementations for the cases with one output

Case with 3072 MPI ranks

MPI Func.	IntelMPI_2019.6	OpenMPI_4.0.2	PSMPI_5.4	IntelMPI_2018.2
Recv	809,2	227,3	74,6	30,8
Send	71,8	85,8	50,6	71,6
Allreduce	23,6	52,3	36,4	25,6
Waitall	8,1	15,8	11,6	9,6

MPI_Recv calls: **30292190**

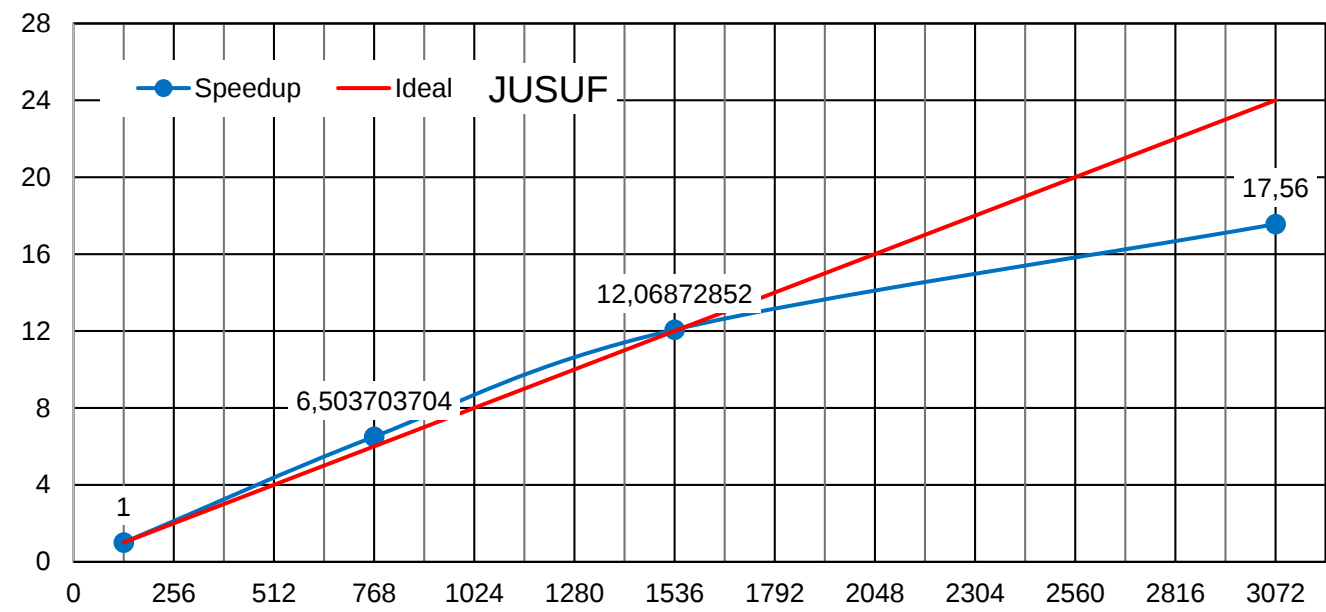
Generating one output via the Adaptable IO System (ADIOS)

N. of ranks	1 output (C)		No output (A)	
	Elapsed Time	MPI Time	Elapsed Time	MPI Time
1536	97,5	37,5	93	32,5
3072	80,9	44,3	71	39

One output at endTime (writeInterval= endTime=200)

#include "adiosWrite" write fields (U p "(k|epsilon|omega)");

Strong scaling study on JUSUF for the cases with no output



Intel2019,PSMPI5.4

Summary

- ✓ With no output OpenFoam scale acceptable (not ideal) up to 64 nodes on JUWELS for the problem investigated.
- ✓ Generating one output degrade the scaling to almost 24 nodes on JUWELS.
- ✓ Output generation using traditional method such as “ensightwrite” is a real bottleneck.
- ✓ With no output, IntelMPI2019.6 perform better than IntelMPI2018.2.
- ✓ With output, performance of IntelMPI2018.2 is much better than IntelMPI2019.6, PSMPI5.4 and OpenMPI4.02.
- ✓ Using ADIOS library the output generation bottleneck can be improved substantially.

Future study

Using “PETSc4FOAM” in OpenFoam in order to make use of external sparse linear solvers PETSc/Hypre On CPU and GPU.

Thanks for your attention!