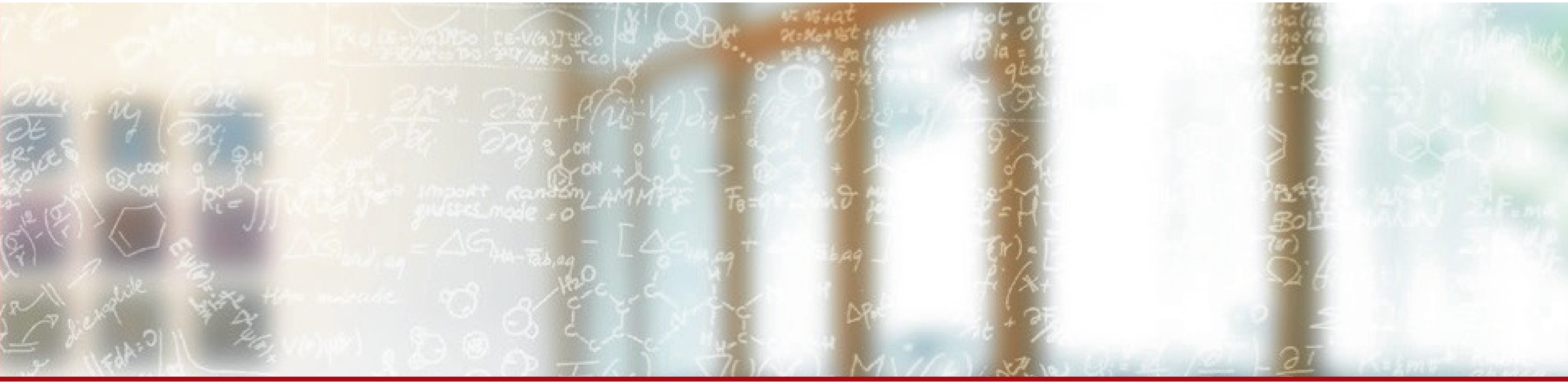
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

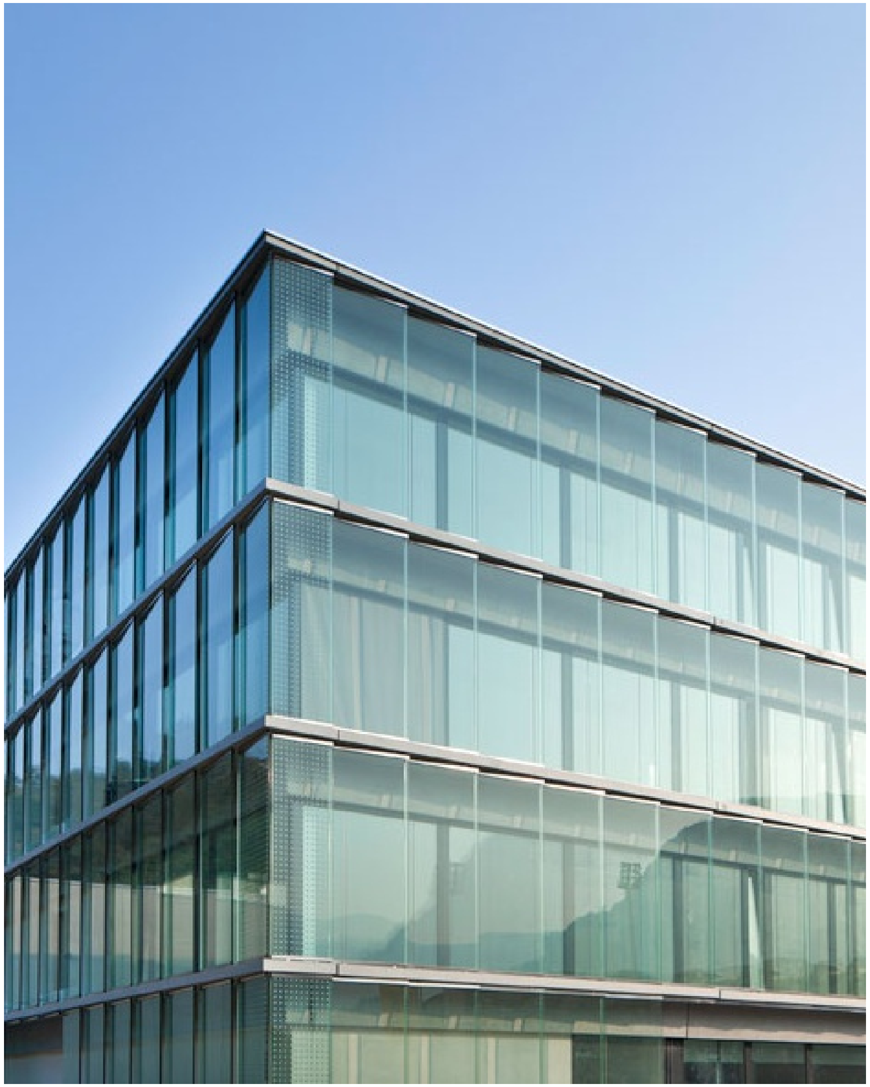
ETH zürich



CSCS Proposal writing webinar
Technical review

12th April 2015
CSCS

Agenda



- Tips for new applicants
  - CSCS overview
  - Allocation process
- Guidelines
  - Basic concepts
  - Performance tools
- Demo
- Q&A open discussion

Tips for new applicants

CSCS: Overview (Nov. 2014)

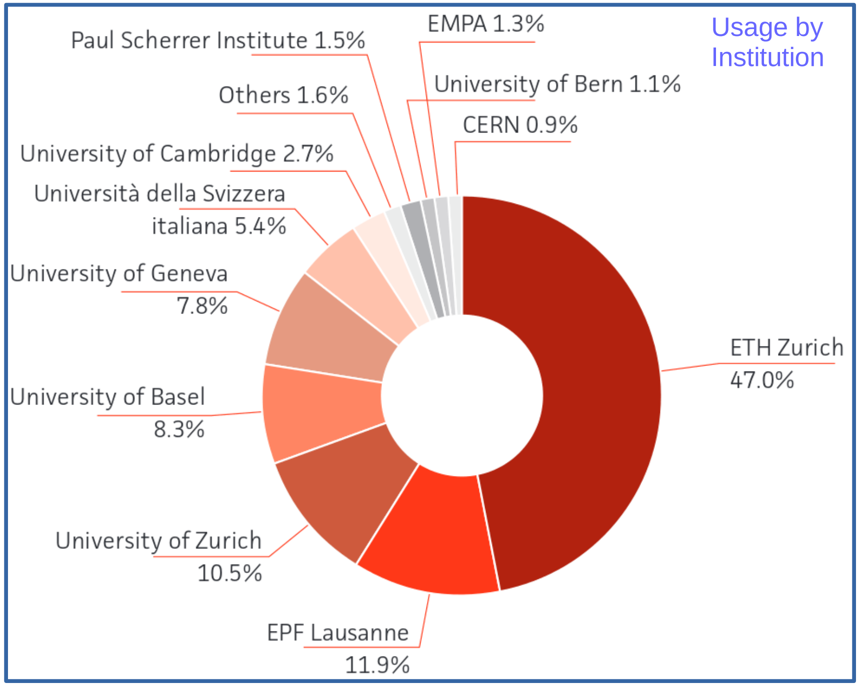
RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325

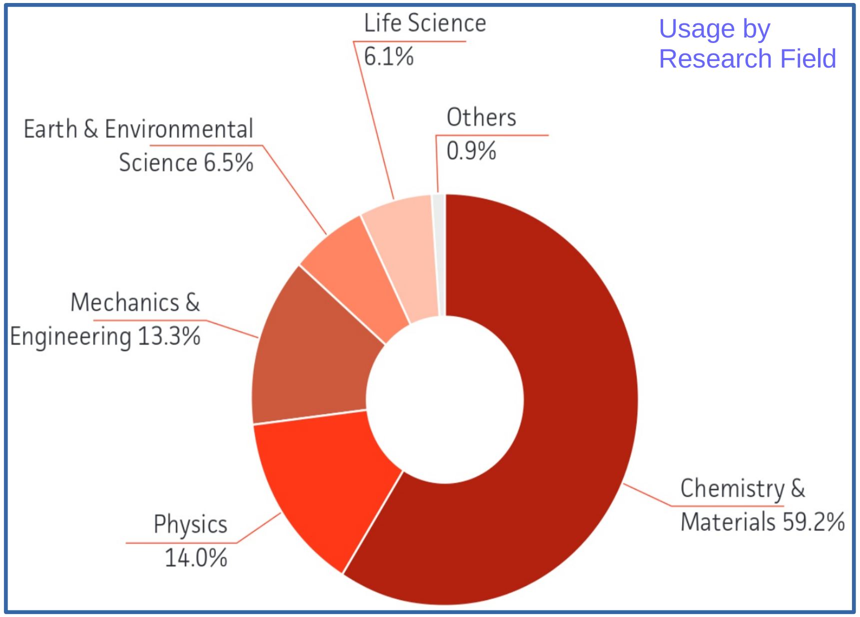
http://www.top500.org

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	5,271.81	GSI Helmholtz Center	L-CSC - ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150 Level 1 measurement data available	57.15
2	4,945.63	High Energy Accelerator Research Organization /KEK	Suiren - ExaScaler 32U256SC Cluster, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, PEZY-SC	37.83
3	4,447.58	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	35.39
4	3,962.73	Cray Inc.	Storm1 - Cray CS-Storm, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, Nvidia K40m Level 3 measurement data available	44.54
5	3,631.70	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
6	3,543.32	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	54.60
7	3,517.84	Center for Computational Sciences, University of Tsukuba	HA-PACS TCA - Cray CS300 Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x	78.77
8	3,459.46	SURFsara	Cartesius Accelerator Island - Bullx B515 cluster, Intel Xeon E5-2450v2 8C 2.5GHz, InfiniBand 4× FDR, Nvidia K40m	44.40
9	3,185.91	Swiss National Supercomputing Centre (CSCS)	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Level 3 measurement data available	1,753.66

http://www.green500.org

CSCS: Usage statistics (2013)





http://www.cscs.ch/publications/annual_reports/

CSCS: Piz Daint

CSCS petascale system:

- Hybrid Cray XC30, 5272 compute nodes, connected with Aries interconnect
- Each compute node hosts 1 Intel SandyBridge CPU and 1 NVIDIA K20X GPU
- For a total of 42176 cores and 5272 GPUs, 7.8 Pflops peak performance

http://user.cscs.ch/computing_resources/piz_daint_and_piz_daint_extension/

Submitting a project at CSCS

Submission deadline :		Will start in :
October 2014	→	April 2015
May 2015	→	October 2015
October 2015	→	April 2016

http://www.cscs.ch/user_lab/

Guidelines: benchmarking and scaling

Proposal format

Production Projects - Cscs - Mozilla Firefox

Production Projects - C...

www.cscs.ch/user_lab/allocation_scheme

Search

Scalability

The production projects should involve highly parallel and scalable simulations. We expect that you state the number of cores on which your jobs will run. CSCS will give preference to projects that involve highly scalable codes.

Project Proposal Format

The project proposal should be no longer than **10 A4 pages** including graphs and references, and must contain the following information:

- Abstract
- Background and significance
- Scientific goals and objectives
- Description of the research methods, algorithms, and code
- Parallelization approach
- Representative benchmarks and scaling
- Performance analysis (click here » to see how to do it)
- Resource justification
- Project plans: tasks and milestones
- Previous results (not mandatory)
- Development and debugging requirements (not mandatory)

Please note that lack of any of the mandatory parts may result in rejection of the proposal.

Benchmarking and Scalability

- There are many ways to measure the execution time:
  - The most simple one is to time the aprun command and to report the real time:

```
/usr/bin/time -p aprun \
        -n 8192 -N 8 -d 1 \
        ./bt-mz_E.8192 \
        &> myjob_output.txt

tail myjob_output.txt
        # BT-MZ Benchmark Completed.
        # real 161.40
        # user 0.06
        # sys 0.10
```

- It is the appplicant responsibility to show the scalability of his application:
  - A code scales if its execution time decreases when using increasing numbers of parallel processing elements (cores, processes, threads, etc…)
  - The idea is to find the scalability limit of your application – the point at which the execution time stops decreasing.

Typical user workflow: launching parallel jobs

Batch system:

- The job submission system used at CSCS is SLURM.

Submit your jobscript:

- cd $SCRATCH/
- cp /project/*/$USER/myinput .
- sbatch myjob.slurm
- squeue -u $USER
- scancel myjobid   # if needed

Adapt the jobscript to your needs:

- aprun [options] myexecutable
  - -n : Total number of MPI tasks
  - -N : Number of MPI tasks per compute node (<= 8)
  - -d : Number of OpenMP threads

```
#!/bin/bash

#SBATCH --ntasks=64          # -n
#SBATCH --ntasks-per-node=8  # -N
#SBATCH --cpus-per-task=1    # -d
#SBATCH --time=01:00:00      # 1hour max
#SBATCH --job-name="my64tasksjob"
#SBATCH --output=o
#SBATCH --error=o

export OMP_NUM_THREADS=1
/usr/bin/time -p aprun -n64 -N8 -d1 myexe
```

http://user.cscs.ch/get_started/run_batch_jobs

Cost study

- To run 50 steps, my code takes:
  - ➔ 1h/6~10m on 16 nodes,
  - ➔ 1h/40~2m on 128 nodes,
  - ➔ 1h/200<1m on 1024 nodes.

For my production job, I must run 2400 steps (x48).

How many compute node hours (CNH) should I ask ?

- User type #1:
  - ➔ (1024nodes*1h/200)*48
  - ➔ Realtime = 246 CNH
  - ➔ Human time < 15 minutes

- User type #2:
  - ➔ (16nodes*1h/6)*48
  - ➔ Realtime = 128 CNH
  - ➔ Human time = 8hours (>>15 min!)
    - ✓ BUT I used only half CNH
    - ✓ I can submit another 2400steps job!

- User type #3:
  - ➔ (128nodes*1h/40)*48
  - ➔ Realtime = 154 CNH
  - ➔ Human time = 1 hour 12 min
    - ✓ Faster than 8h !
    - ✓ I can submit another 2400steps job!

cscs

Cost study

- To run 50 steps, my code takes:
  - ➔ 1h/6~10m on 16 nodes,
  - ➔ 1h/40~2m on 128 nodes,
  - ➔ 1h/200<1m on 1024 nodes.

For my production job, I must run 2400 steps (x48).

How many compute node hours (CNH) should I ask ?

- User type #0:
  - ➔ (2048nodes*1h/100)*48
  - ➔ Realtime = 980 CNH !
  - ➔ Human time = 30 minutes !

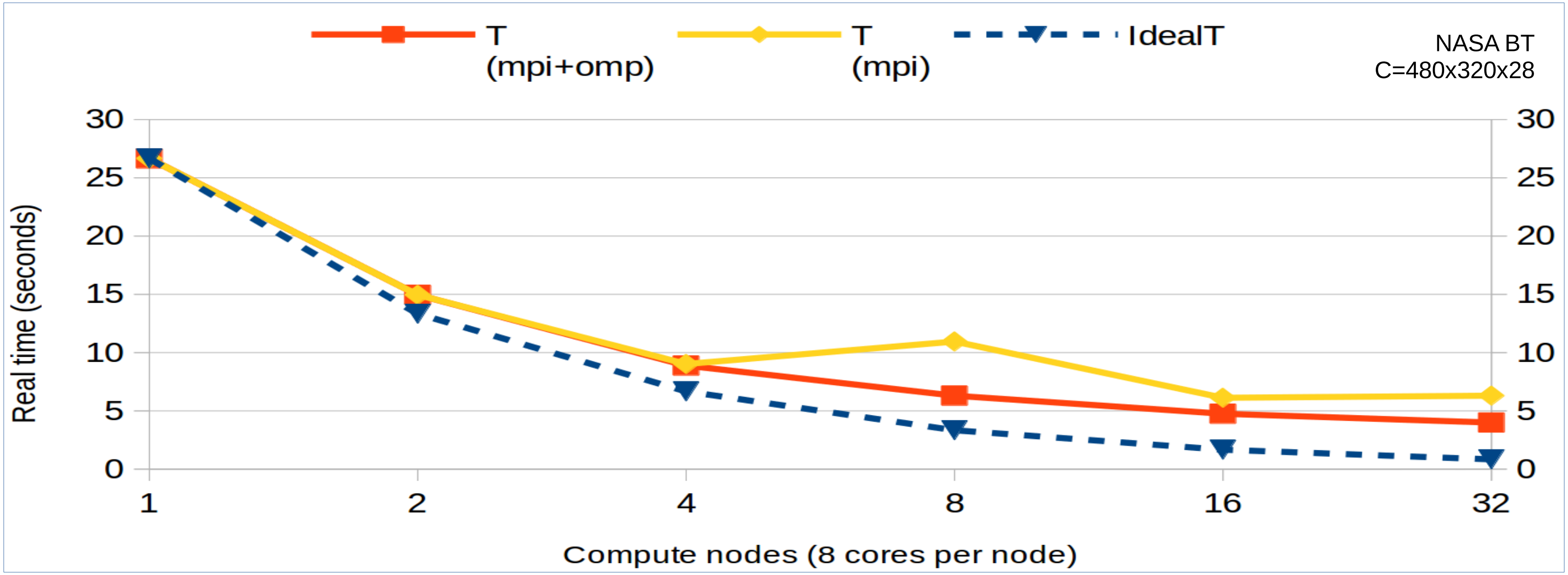
- User type #1:
  - ➔ (1024nodes*1h/200)*48
  - ➔ Realtime = 246 CNH
  - ➔ Human time < 15 minutes

- User type #2:
  - ➔ (16nodes*1h/6)*48
  - ➔ Realtime = 128 CNH
  - ➔ Human time = 8hours (>>15 min!)
    - ✓ BUT I used only half CNH
    - ✓ I can submit another 2400steps job!

- User type #3:
  - ➔ (128nodes*1h/40)*48
  - ➔ Realtime = 154 CNH
  - ➔ Human time = 1 hour 12 min
    - ✓ Faster than 8h !
    - ✓ I can submit another 2400steps job!

Benchmarking: Mpi+OpenMP code on PizDaint (small problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	8	8	1	26.65	26.65
2	16	8	1	14.95	14.95
4	16	4	2	8.88	9.02
8	32	4	2	6.3	10.95
16	64	4	2	4.75	6.12
32	64	2	4	3.99	6.3



Speedup and Efficiency

- It can sometimes be difficult to read a scaling curve
  - It is standard to compare the execution time with a reference time
  - Speedup is defined by the following formula:
    - where T_0 is the reference time, and
    - T_n is the execution on n compute nodes

$$Speedup_n = \frac{t_0}{t_n}$$

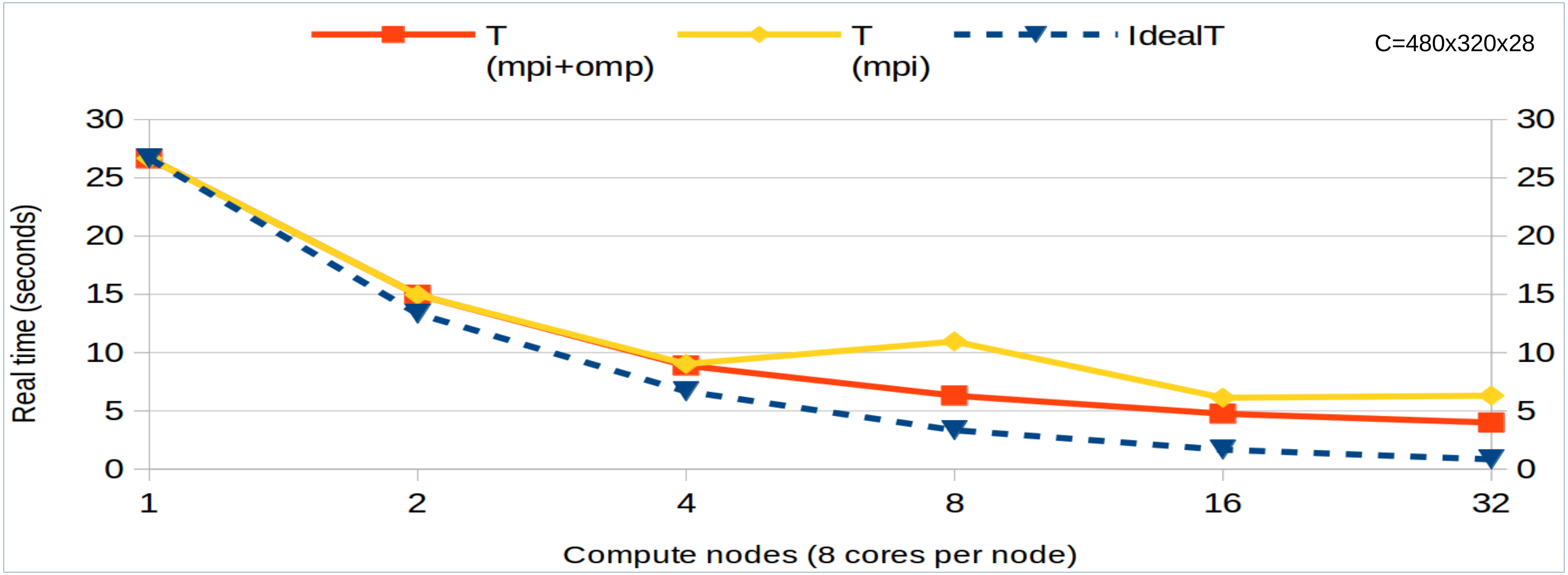
    - Linear speedup or ideal speedup is obtained when $Speedup_n = n$.
    - When running a code with linear speedup, doubling the number of processors doubles the speed. As this is ideal, it is considered very good scalability.
  - Efficiency is a performance metric defined as

$$Efficiency_n = \frac{S_n}{n}$$

    - Codes with an efficiency > 0.5 are considered scalable.

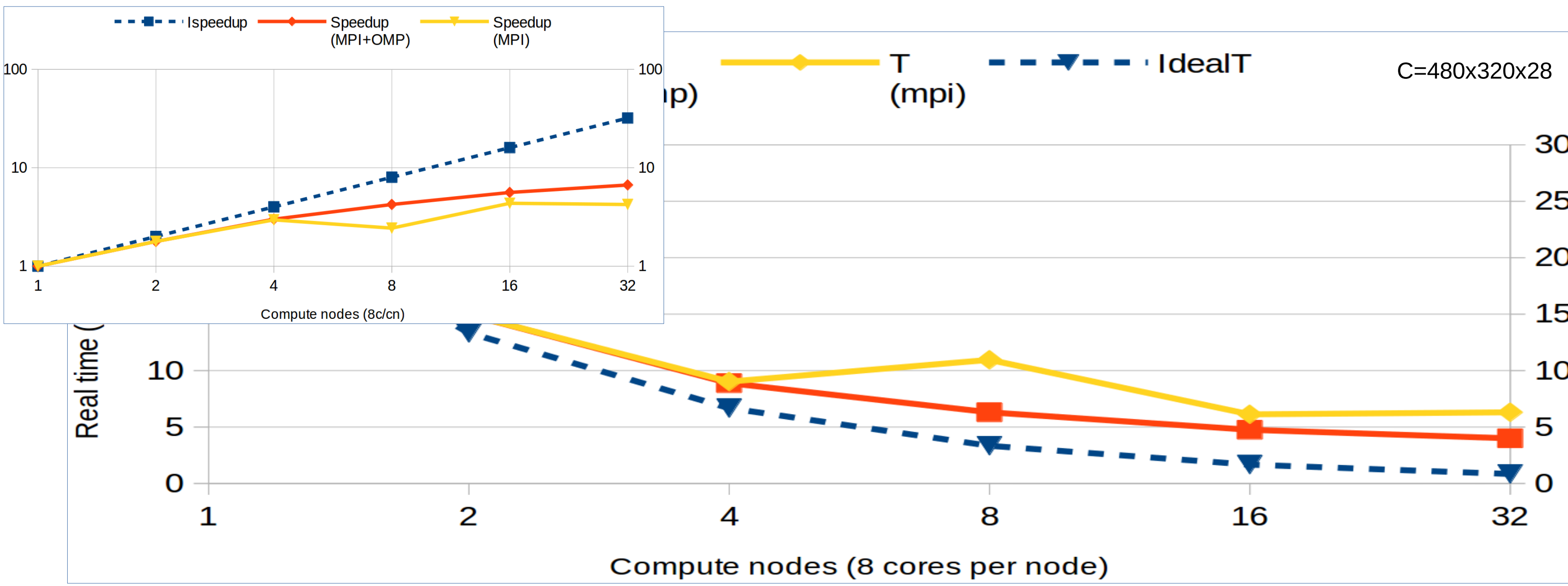
Scalability quiz: Mpi+OpenMP code on PizDaint (small problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	8	8	1	26.65	26.65
2	16	8	1	14.95	14.95
4	16	4	2	8.88	9.02
8	32	4	2	6.3	10.95
16	64	4	2	4.75	6.12
32	64	2	4	3.99	6.3



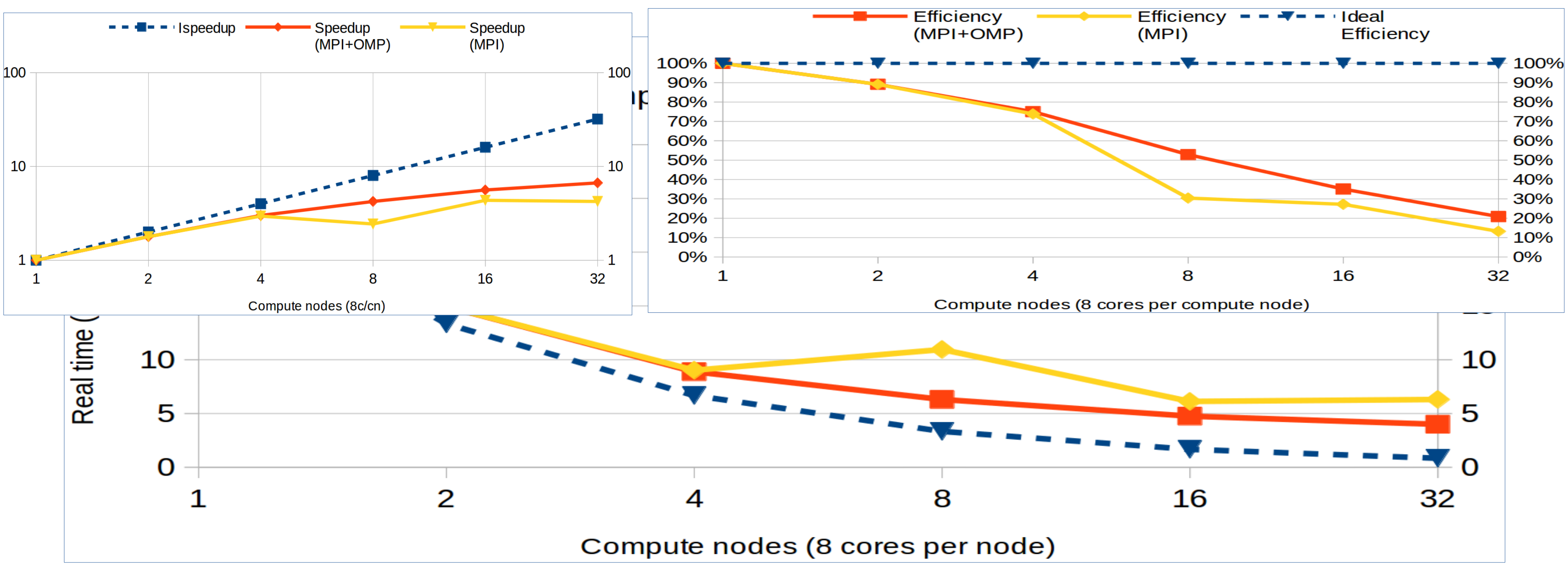
Scalability quiz: Mpi+OpenMP code on PizDaint (small problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	8	8	1	26.65	26.65
2	16	8	1	14.95	14.95
4	16	4	2	8.88	9.02
8	32	4	2	6.3	10.95
16	64	4	2	4.75	6.12
32	64	2	4	3.99	6.3



Scalability quiz: Mpi+OpenMP code on PizDaint (small problem size)

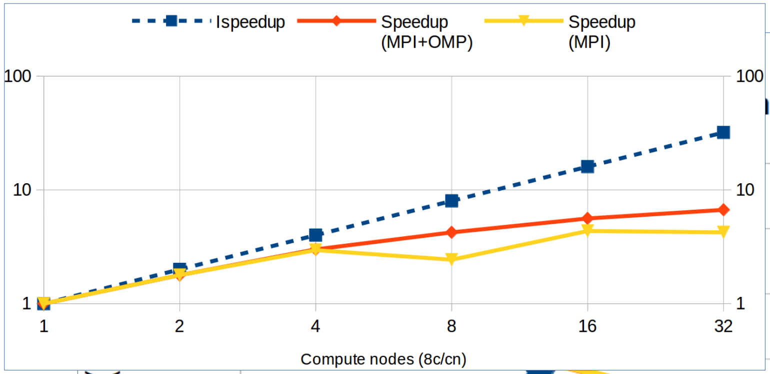
CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	8	8	1	26.65	26.65
2	16	8	1	14.95	14.95
4	16	4	2	8.88	9.02
8	32	4	2	6.3	10.95
16	64	4	2	4.75	6.12
32	64	2	4	3.99	6.3

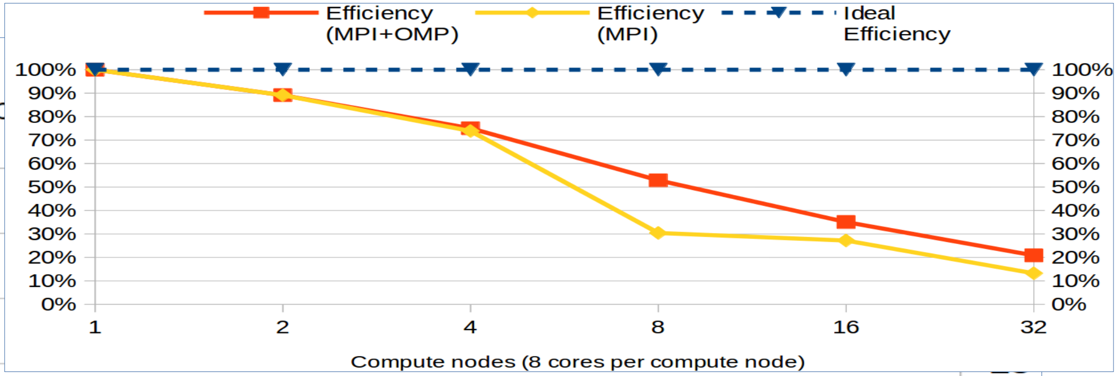


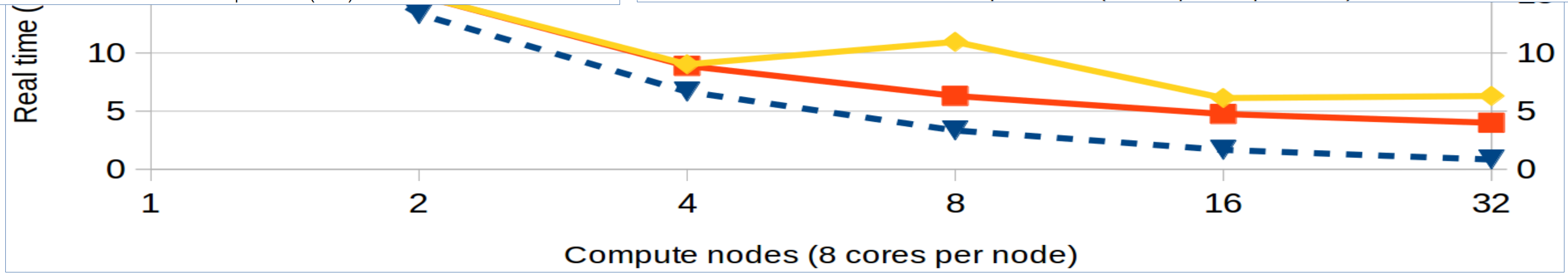
Scalability quiz: Mpi+OpenMP code on PizDaint (small problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	8	8	1	26.65	26.65
2	16	8	1	14.95	14.95
4	16	4	2	8.88	9.02
8	32	4	2	6.3	10.95
16	64	4	2	4.75	6.12
32	64	2	4	3.99	6.3

➔ Use MPI&OpenMP



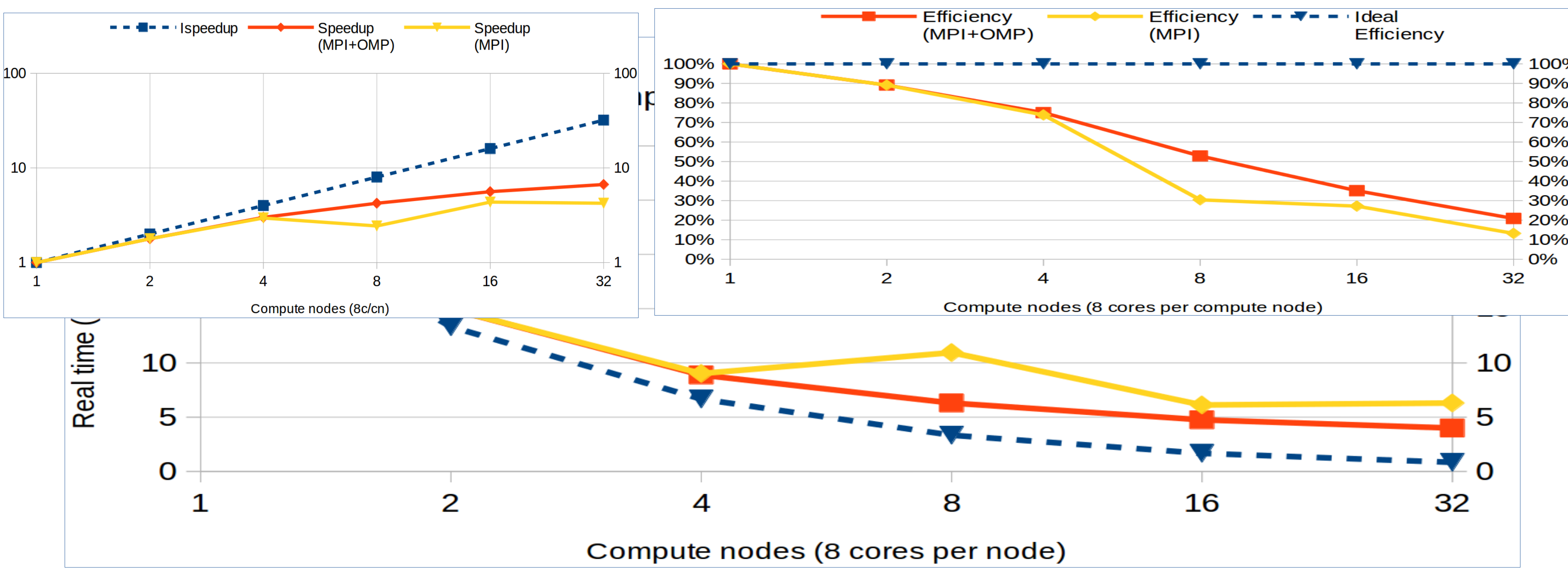




Scalability quiz: Mpi+OpenMP code on PizDaint (small problem size)

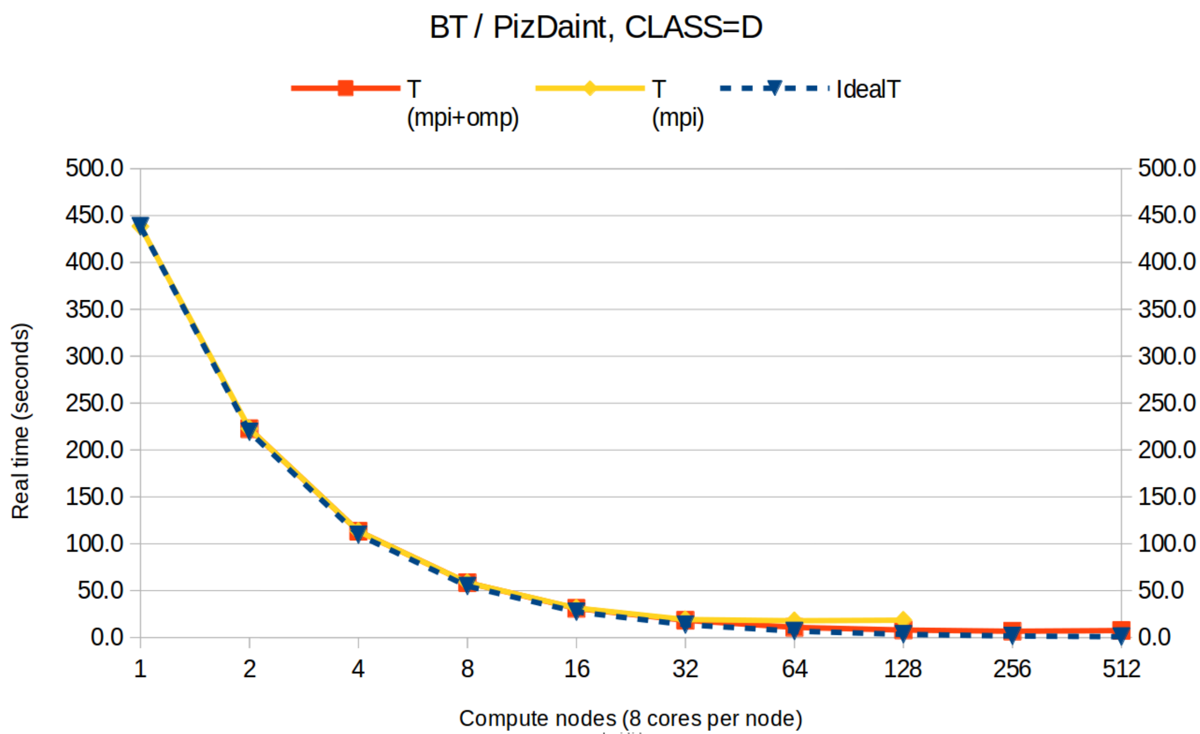
CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	8	8	1	26.65	26.65
2	16	8	1	14.95	14.95
4	16	4	2	8.88	9.02
8	32	4	2	6.3	10.95
16	64	4	2	4.75	6.12
32	64	2	4	3.99	6.3

- ➔ Use MPI&OpenMP
- ➔ Scales up to 8 CN



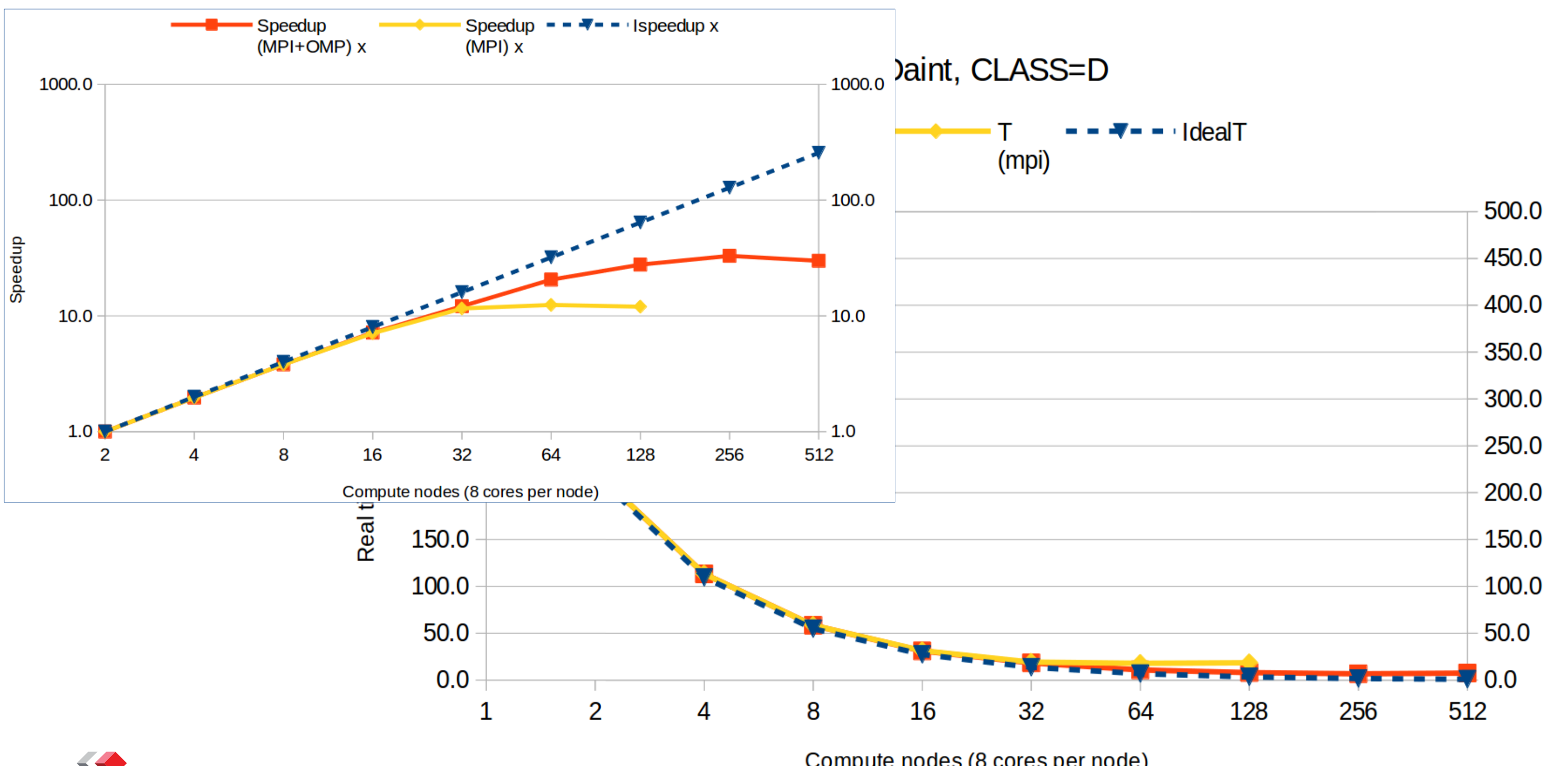
Scalability quiz: Mpi+OpenMP code on PizDaint (medium problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	x	x	x	x	438.4
2	8	4	2	222.6	222.6
4	32	8	1	113.3	113.3
8	32	4	2	58.5	58.5
16	64	4	2	31.1	31.5
32	128	4	2	18.4	19.2
64	64	1	8	10.8	17.9
128	128	1	8	8.0	18.6
256	256	1	8	6.8	x
512	512	1	8	7.5	x



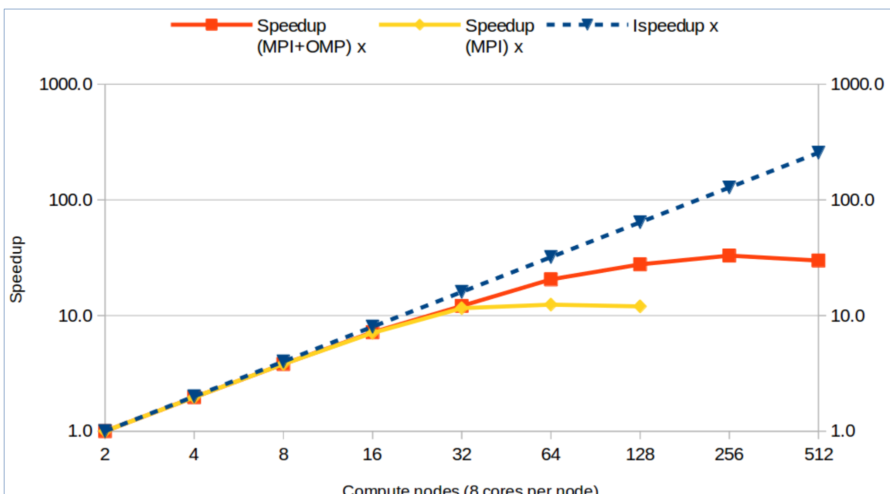
Scalability quiz: Mpi+OpenMP code on PizDaint (medium problem size)

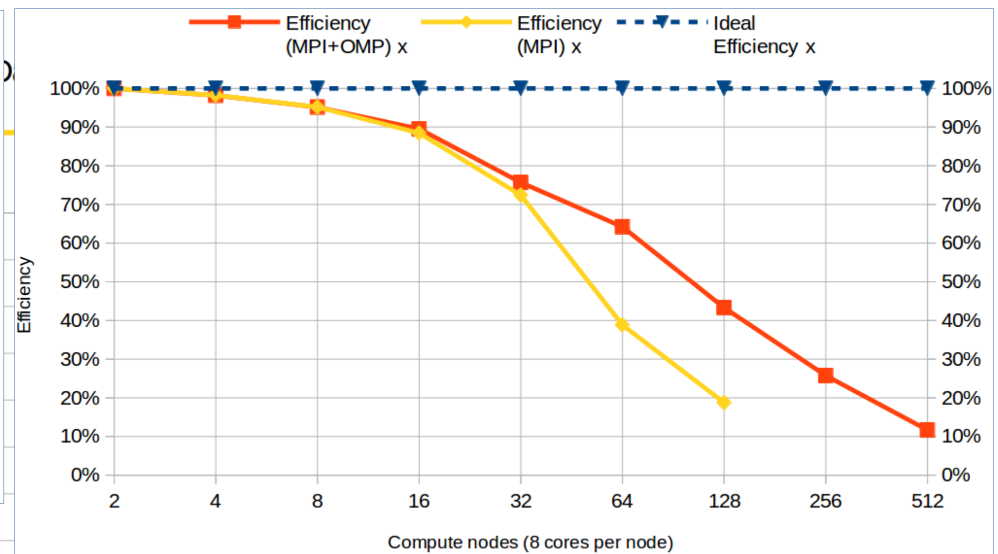
CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	x	x	x	x	438.4
2	8	4	2	222.6	222.6
4	32	8	1	113.3	113.3
8	32	4	2	58.5	58.5
16	64	4	2	31.1	31.5
32	128	4	2	18.4	19.2
64	64	1	8	10.8	17.9
128	128	1	8	8.0	18.6
256	256	1	8	6.8	x
512	512	1	8	7.5	x

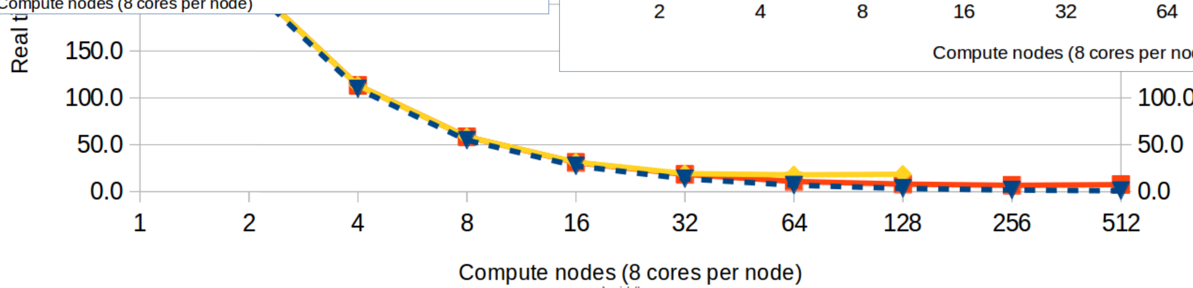


Scalability quiz: Mpi+OpenMP code on PizDaint (medium problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	x	x	x	x	438.4
2	8	4	2	222.6	222.6
4	32	8	1	113.3	113.3
8	32	4	2	58.5	58.5
16	64	4	2	31.1	31.5
32	128	4	2	18.4	19.2
64	64	1	8	10.8	17.9
128	128	1	8	8.0	18.6
256	256	1	8	6.8	x
512	512	1	8	7.5	x



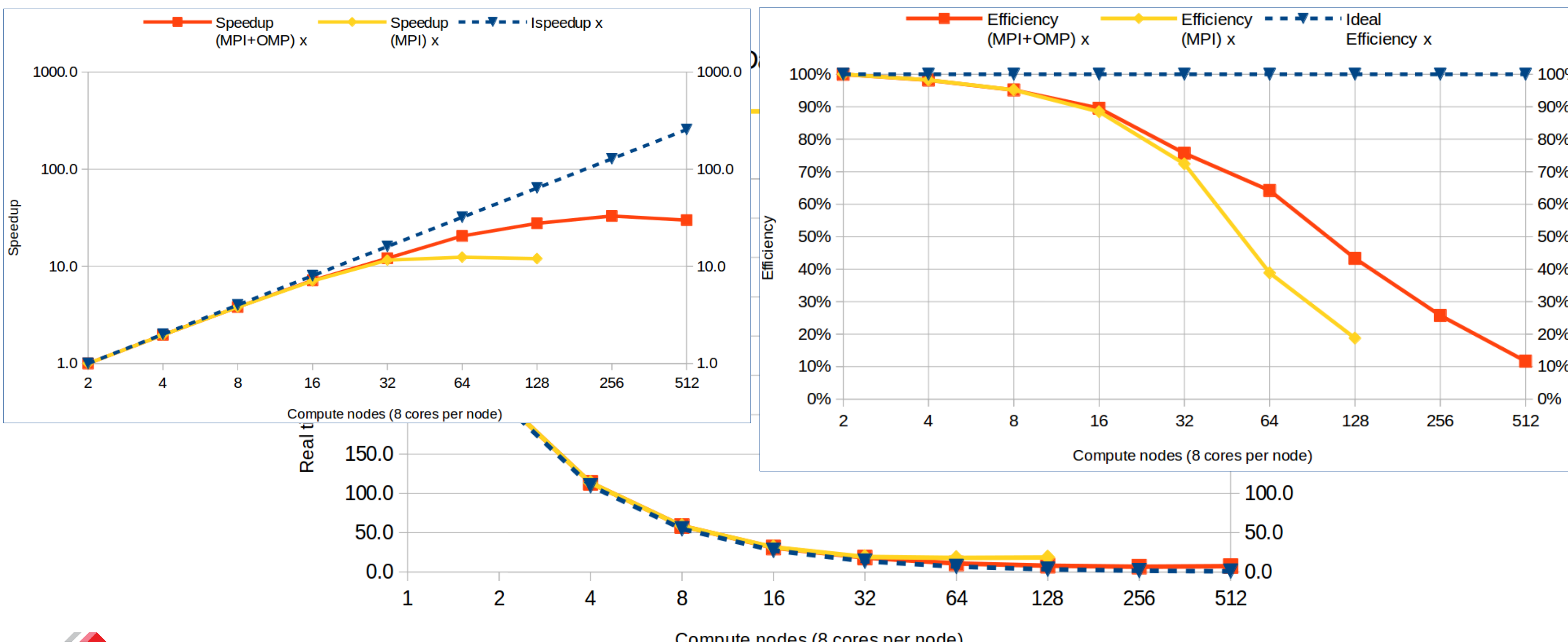




Scalability quiz: Mpi+OpenMP code on PizDaint (medium problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	x	x	x	x	438.4
2	8	4	2	222.6	222.6
4	32	8	1	113.3	113.3
8	32	4	2	58.5	58.5
16	64	4	2	31.1	31.5
32	128	4	2	18.4	19.2
64	64	1	8	10.8	17.9
128	128	1	8	8.0	18.6
256	256	1	8	6.8	x
512	512	1	8	7.5	x

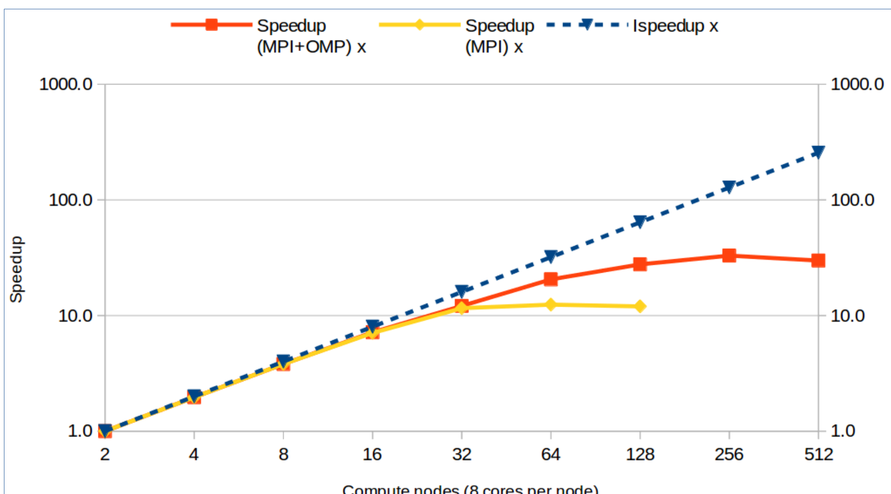
➔ Use MPI&OpenMP

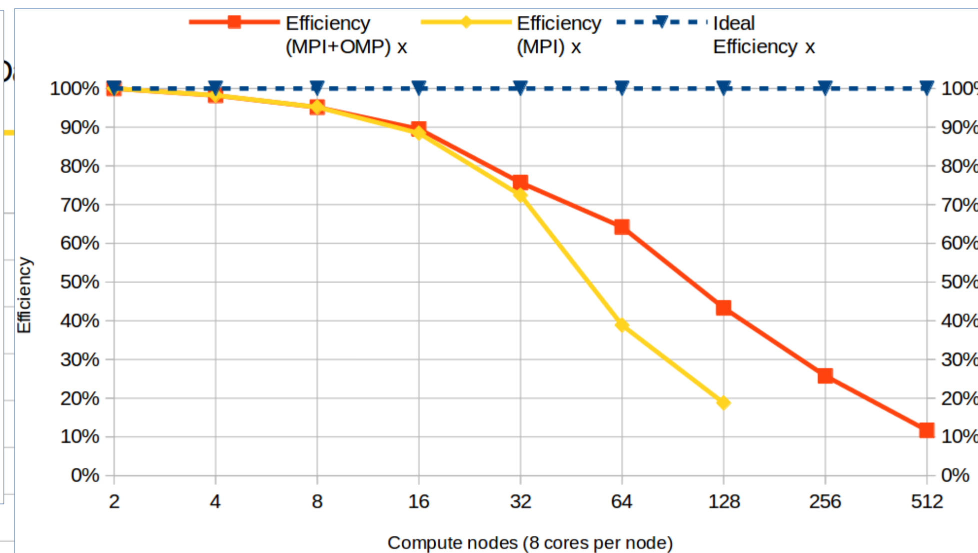


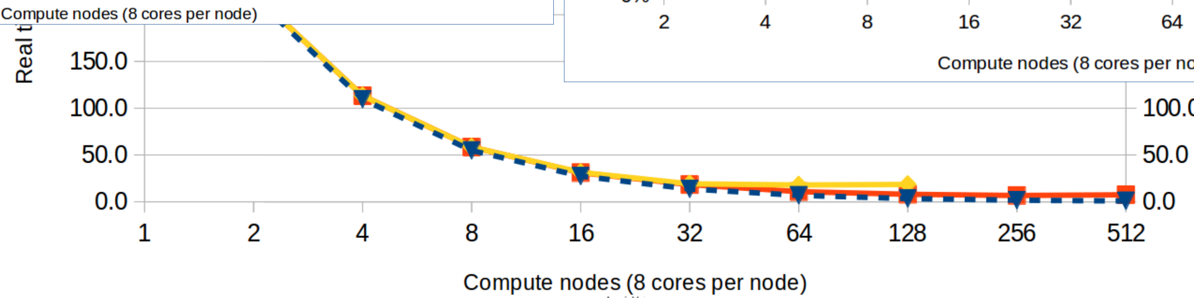
Scalability quiz: Mpi+OpenMP code on PizDaint (medium problem size)

CN	-n	-N	-d	Real time (mpi+omp)	Real time (mpi)
1	x	x	x	x	438.4
2	8	4	2	222.6	222.6
4	32	8	1	113.3	113.3
8	32	4	2	58.5	58.5
16	64	4	2	31.1	31.5
32	128	4	2	18.4	19.2
64	64	1	8	10.8	17.9
128	128	1	8	8.0	18.6
256	256	1	8	6.8	x
512	512	1	8	7.5	x

- ➔ Use MPI&OpenMP
- ➔ Scales up to 64 CN







How much can you ask ?

- Allocation units are in node hours:
  - Compare different job sizes (using the time command),
  - Report in your proposal the execution timings (without tools) for each job size,
  - Multiply the optimal job size (nodes) by the execution time (hours) to find the amount to request (compute node hours).

- Projects will always be charged full node hours:
  - even though not all the CPU cores are used,
  - even though the GPU is not used.

- Performance data must come from jobs run on Piz Daint:
  - For a problem similar to that proposed in the project description
    - use a problem state that best matches your intended production runs,
    - scaling should be measured based on the overall performance of the application,
    - compare results from same machine, same computational model,
    - no simplified models or preferential configurations.

- There are many ways for presenting mediocre performance results in the best possible light:
  - We know them,
  - Contact us if you need help: help@cscs.ch

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Guidelines: performance report

Typical user workflow: compilation

0. Connect to CSCS:
   - ssh -X myusername@ela.cscs.ch
   - ssh daint

1. Setup your Programming Environment:
   - module swap PrgEnv-cray PrgEnv-gnu
     - module list
     - module avail
     - module avail cray-netcdf
     - module show cray-netcdf
     - module load cray-netcdf/4.3.2
     - module rm cray-netcdf/4.3.2

2. Compile your code:
   - make clean
   - make bt-mz MAIN=bt CLASS=C NPROCS=64

3. Submit your job:
   - cd bin
   - sbatch myjob.slurm

4 compilers available:
- CCE
- GNU
- INTEL
- PGI

CRAY specifics:
You must use the CRAY wrappers to compile
- ftn (Fortran),
- cc (C)
- CC (C++)

The wrappers will detect the modules you have loaded and will automatically use the compiler and libraries that you need.

http://user.cscs.ch/compiling_optimizing/compiling_your_code

Typical user workflow: compilation with perftool

1. Setup your Programming Environment:
   - module swap PrgEnv-cray PrgEnv-gnu
   - module use /project/csstaff/proposals
   - module load perflite/622

   or
   - module load craype-accel-nvidia35
   - module load perflite/622cuda

   or
   - module load craype-accel-nvidia35
   - module load perflite/622openacc

2. Recompile your code with the tool:
   - cd $SCRATCH/proposals.git/vihps/NPB3.3-MZ-MPI
   - make clean
   - make bt-mz MAIN=bt CLASS=C NPROCS=64

3. Submit your job:
   - cd bin
   - sbatch myjob.slurm

4. Read and attach the full report to your proposal:
   - cat *.rpt

perflite is Cray's performance tool,
it provides performance statistics automatically,
and supports all compilers (cce, intel, gnu, pgi).

For MPI/OpenMP codes

For MPI/OpenMP + Cuda codes

For MPI/OpenMP + OpenACC codes

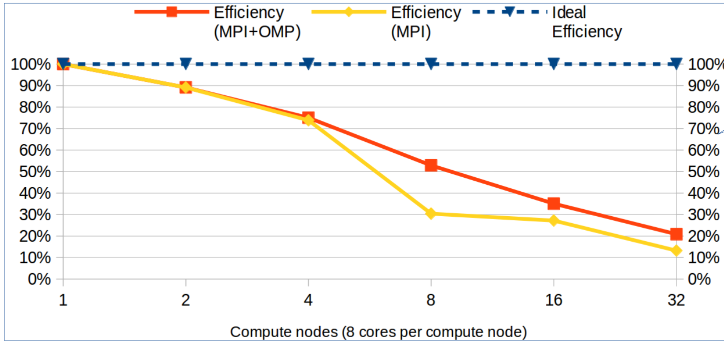
No need to modify the src code,
just recompile

No need to modify the jobscript,
Just rerun

Performance report at the end of the job

http://user.cscs.ch/compiling_optimizing/performance_report

Performance: Mpi+OpenMP code on PizDaint (BT CLASS=C, 50 steps)



The code stops scaling above 8 nodes. Why ?

What can we learn from the performance reports ?

8CN: aprun -n32 -N4 -d2 ./bt-mz_C.32+pat622

```
Table 1:  Profile by Function Group and Function
(top 7 functions shown)

 Samp% |     Samp |  Imb. |  Imb. |Group
       |          |  Samp | Samp% | Function
       |          |       |       |  PE=HIDE
       |          |       |       |   Thread=HIDE

 100.0% | 15260.0 |    -- |    -- |Total
|-------------------------------------------------------
|  93.4% | 14250.3 |    -- |    -- |USER
||------------------------------------------------------
||  24.2% |  3686.6 | 187.4 |   5.0% |binvcrhs_
||  14.2% |  2165.4 | 109.6 |   5.0% |z_solve_._omp_fn.0
||  12.7% |  1941.8 | 189.2 |   9.2% |y_solve_._omp_fn.0
||  12.5% |  1913.9 |  94.1 |   4.8% |x_solve_._omp_fn.0
||  12.5% |  1912.9 |  93.1 |   4.8% |matmul_sub_
||  11.0% |  1686.1 | 136.9 |   7.8% |compute_rhs_._omp_fn.0
||   3.8% |   586.8 |  50.2 |   8.1% |matvec_sub_
||======================================================
|   4.2% |   639.8 |    -- |    -- |MPI
||------------------------------------------------------
||   3.8% |   572.5 | 297.5 |  35.3% |mpi_waitall
|=======================================================
```

16CN: aprun -n64 -N4 -d2 ./bt-mz_C.64+pat622

```
Table 1:  Profile by Function Group and Function
(top 10 functions shown)

 Samp% |   Samp |  Imb. |  Imb. |Group
       |        |  Samp | Samp% | Function
       |        |       |       |  PE=HIDE
       |        |       |       |   Thread=HIDE

 100.0% | 8526.5 |    -- |    -- |Total
|-----------------------------------------------------
|  83.5% | 7121.8 |    -- |    -- |USER
||----------------------------------------------------
||  21.6% | 1843.2 | 139.8 |   7.2% |binvcrhs_
||  12.7% | 1079.1 | 147.9 |  12.2% |z_solve_._omp_fn.0
||  11.4% |  972.9 | 172.1 |  15.3% |y_solve_._omp_fn.0
||  11.2% |  952.1 | 141.9 |  13.2% |x_solve_._omp_fn.0
||  11.0% |  941.8 |  62.2 |   6.3% |matmul_sub_
||  10.2% |  869.6 | 151.4 |  15.1% |compute_rhs_._omp_fn.0
||   3.4% |  286.8 |  44.2 |  13.6% |matvec_sub_
||====================================================
|  11.2% |  956.9 |    -- |    -- |MPI
||----------------------------------------------------
||   9.7% |  825.1 | 376.9 |  31.9% |mpi_waitall
||====================================================
|   2.6% |  224.7 | 618.3 |  74.5% |PTHREAD
||----------------------------------------------------
||   2.6% |  224.7 | 618.3 |  74.5% |pthread_join
||====================================================
|   2.6% |  221.1 |    -- |    -- |ETC
||----------------------------------------------------
||   1.6% |  133.7 | 100.3 |  43.6% |gomp_barrier_wait_end
|=====================================================
```

32CN: aprun -n64 -N2 -d4 ./bt-mz_C.64+pat622

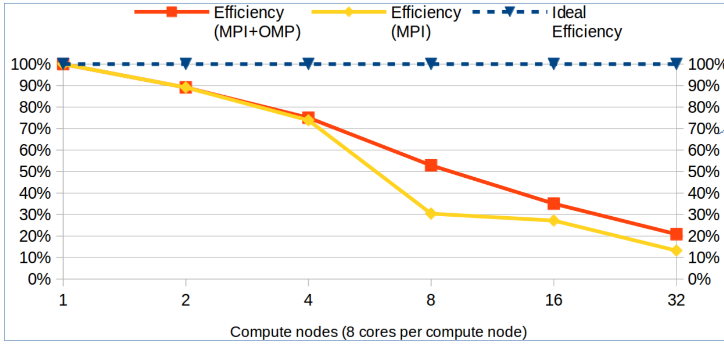
```
Table 1:  Profile by Function Group and Function
(top 10 functions shown)

 Samp% |   Samp |  Imb. |  Imb. |Group
       |        |  Samp | Samp% | Function
       |        |       |       |  PE=HIDE
       |        |       |       |   Thread=HIDE

 100.0% | 5949.8 |    -- |    -- |Total
|-----------------------------------------------------
|  63.5% | 3778.4 |    -- |    -- |USER
||----------------------------------------------------
||  16.6% |  987.3 |  73.7 |   7.1% |binvcrhs_
||   9.4% |  561.4 |  92.6 |  14.4% |z_solve_._omp_fn.0
||   8.7% |  515.4 |  86.6 |  14.6% |y_solve_._omp_fn.0
||   8.6% |  514.3 |  66.7 |  11.7% |x_solve_._omp_fn.0
||   8.5% |  503.6 |  69.4 |  12.3% |matmul_sub_
||   7.5% |  443.5 |  67.5 |  13.4% |compute_rhs_._omp_fn.0
||   2.6% |  157.1 |  28.9 |  15.8% |matvec_sub_
||====================================================
|  13.6% |  806.3 |    -- |    -- |ETC
||----------------------------------------------------
||  11.3% |  671.8 | 212.2 |  24.4% |gomp_barrier_wait_end
||====================================================
|  11.9% |  707.0 | 639.0 |  48.2% |PTHREAD
||----------------------------------------------------
||  11.9% |  707.0 | 639.0 |  48.2% |pthread_join
||====================================================
|  11.0% |  656.2 |    -- |    -- |MPI
||----------------------------------------------------
||   8.4% |  501.7 | 416.3 |  46.1% |mpi_waitall
|=====================================================
```

Setup your Programming Environment:
- module swap PrgEnv-cray PrgEnv-gnu
- module use /project/csstaff/proposals
- module load perflite/622

Recompile your code with the tool:
- cd $SCRATCH/proposals.git/vihps/NPB3.3-MZ-MPI
- make clean
- make bt-mz MAIN=bt CLASS=C NPROCS=64

Submit your job:
- cd bin
- sbatch myjob.slurm
- cat *.rpt

Performance: Mpi+OpenMP code on PizDaint (BT CLASS=C, 50 steps)



The code stops scaling above 8 nodes. Why ?

What can we learn from the performance reports ?

8CN: aprun -n32 -N4 -d2 ./bt-mz_C.32+pat622

```
Table 1:  Profile by Function Group and Function
(top 7 functions shown)

  Samp% |    Samp |  Imb. |  Imb. |Group
        |         |  Samp | Samp% | Function
        |         |       |       |  PE=HIDE
        |         |       |       |   Thread=HIDE

 100.0% | 15260.0 |    -- |    -- |Total
|-------------------------------------------------------
|  93.4% | 14250.3 |    -- |    -- |USER
||------------------------------------------------------
||  24.2% |  3686.6 | 187.4 |  5.0% |binvcrhs_
||  14.2% |  2165.4 | 109.6 |  5.0% |z_solve_._omp_fn.0
||  12.7% |  1941.8 | 189.2 |  9.2% |y_solve_._omp_fn.0
||  12.5% |  1913.9 |  94.1 |  4.8% |x_solve_._omp_fn.0
||  12.5% |  1912.9 |  93.1 |  4.8% |matmul_sub_
||  11.0% |  1686.1 | 136.9 |  7.8% |compute_rhs_._omp_fn.0
||   3.8% |   586.8 |  50.2 |  8.1% |matvec_sub_
||======================================================
|   4.2% |   639.8 |    -- |    -- |MPI
||------------------------------------------------------
||   3.8% |   572.5 | 297.5 | 35.3% |mpi_waitall
|=======================================================
```

16CN: aprun -n64 -N4 -d2 ./bt-mz_C.64+pat622

```
Table 1:  Profile by Function Group and Function
(top 10 functions shown)

  Samp% |   Samp |  Imb. |  Imb. |Group
        |        |  Samp | Samp% | Function
        |        |       |       |  PE=HIDE
        |        |       |       |   Thread=HIDE

 100.0% | 8526.5 |    -- |    -- |Total
|-------------------------------------------------------
|  83.5% | 7121.8 |    -- |    -- |USER
||------------------------------------------------------
||  21.6% | 1843.2 | 139.8 |  7.2% |binvcrhs_
||  12.7% | 1079.1 | 147.9 | 12.2% |z_solve_._omp_fn.0
||  11.4% |  972.9 | 172.1 | 15.3% |y_solve_._omp_fn.0
||  11.2% |  952.1 | 141.9 | 13.2% |x_solve_._omp_fn.0
||  11.0% |  941.8 |  62.2 |  6.3% |matmul_sub_
||  10.2% |  869.6 | 151.4 | 15.1% |compute_rhs_._omp_fn.0
||   3.4% |  286.8 |  44.2 | 13.6% |matvec_sub_
||======================================================
|  11.2% |  956.9 |    -- |    -- |MPI
||------------------------------------------------------
||   9.7% |  825.1 | 376.9 | 31.9% |mpi_waitall
||======================================================
|   2.6% |  224.7 | 618.3 | 74.5% |PTHREAD
||------------------------------------------------------
||   2.6% |  224.7 | 618.3 | 74.5% |pthread_join
||======================================================
|   2.6% |  221.1 |    -- |    -- |ETC
||------------------------------------------------------
||   1.6% |  133.7 | 100.3 | 43.6% |gomp_barrier_wait_end
|=======================================================
```

32CN: aprun -n64 -N2 -d4 ./bt-mz_C.64+pat622

```
Table 1:  Profile by Function Group and Function
(top 10 functions shown)

  Samp% |   Samp |  Imb. |  Imb. |Group
        |        |  Samp | Samp% | Function
        |        |       |       |  PE=HIDE
        |        |       |       |   Thread=HIDE

 100.0% | 5949.8 |    -- |    -- |Total
|-------------------------------------------------------
|  63.5% | 3778.4 |    -- |    -- |USER
||------------------------------------------------------
||  16.6% |  987.3 |  73.7 |  7.1% |binvcrhs_
||   9.4% |  561.4 |  92.6 | 14.4% |z_solve_._omp_fn.0
||   8.7% |  515.4 |  86.6 | 14.6% |y_solve_._omp_fn.0
||   8.6% |  514.3 |  66.7 | 11.7% |x_solve_._omp_fn.0
||   8.5% |  503.6 |  69.4 | 12.3% |matmul_sub_
||   7.5% |  443.5 |  67.5 | 13.4% |compute_rhs_._omp_fn.0
||   2.6% |  157.1 |  28.9 | 15.8% |matvec_sub_
||======================================================
|  13.6% |  806.3 |    -- |    -- |ETC
||------------------------------------------------------
||  11.3% |  671.8 | 212.2 | 24.4% |gomp_barrier_wait_end
||======================================================
|  11.9% |  707.0 | 639.0 | 48.2% |PTHREAD
||------------------------------------------------------
||  11.9% |  707.0 | 639.0 | 48.2% |pthread_join
||======================================================
|  11.0% |  656.2 |    -- |    -- |MPI
||------------------------------------------------------
||   8.4% |  501.7 | 416.3 | 46.1% |mpi_waitall
|=======================================================
```

Setup your Programming Environment:

- module swap PrgEnv-cray PrgEnv-gnu
- module use /project/csstaff/proposals
- module load perflite/622

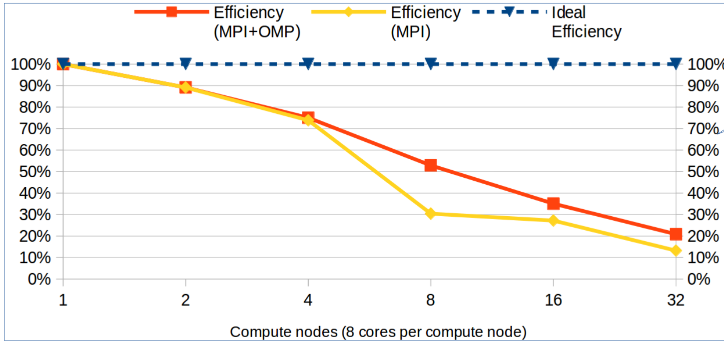
Recompile your code with the tool:

- cd $SCRATCH/proposals.git/vihps/NPB3.3-MZ-MPI
- make clean
- make bt-mz MAIN=bt CLASS=C NPROCS=64

Submit your job:

- cd bin
- sbatch myjob.slurm
- cat *.rpt

Performance: Mpi+OpenMP code on PizDaint (BT CLASS=C, 50 steps)



The code stops scaling above 8 nodes. Why ?

What can we learn from the performance reports ?

CN	USER	MPI	ELSE
8	93%	4%	3%
16	83%	11%	6%
32	63%	11%	26%

8CN: aprun -n32 -N4 -d2 ./bt-mz_C.32+pat622

```
Table 1:  Profile by Function Group and Function
(top 7 functions shown)

 Samp% |    Samp |  Imb. |  Imb. |Group
       |         |  Samp | Samp% | Function
       |         |       |       |  PE=HIDE
       |         |       |       |   Thread=HIDE

 100.0% | 15260.0 |    -- |    -- |Total
|-----------------------------------------------------
|  93.4% | 14250.3 |    -- |    -- |USER
||----------------------------------------------------
||  24.2% |  3686.6 | 187.4 |  5.0% |binvcrhs_
||  14.2% |  2165.4 | 109.6 |  5.0% |z_solve_._omp_fn.0
||  12.7% |  1941.8 | 189.2 |  9.2% |y_solve_._omp_fn.0
||  12.5% |  1913.9 |  94.1 |  4.8% |x_solve_._omp_fn.0
||  12.5% |  1912.9 |  93.1 |  4.8% |matmul_sub_
||  11.0% |  1686.1 | 136.9 |  7.8% |compute_rhs_._omp_fn.0
||   3.8% |   586.8 |  50.2 |  8.1% |matvec_sub_
||====================================================
|   4.2% |   639.8 |    -- |    -- |MPI
||----------------------------------------------------
||   3.8% |   572.5 | 297.5 | 35.3% |mpi_waitall
|=====================================================
```

Setup your Programming Environment:

- module swap PrgEnv-cray PrgEnv-gnu
- module use /project/csstaff/proposals
- module load perflite/622

Recompile your code with the tool:

- cd $SCRATCH/proposals.git/vihps/NPB3.3-MZ-MPI
- make clean
- make bt-mz MAIN=bt CLASS=C NPROCS=64

Submit your job:

- cd bin
- sbatch myjob.slurm
- cat *.rpt

16CN: aprun -n64 -N4 -d2 ./bt-mz_C.64+pat622

```
Table 1:  Profile by Function Group and Function
(top 10 functions shown)

 Samp% |   Samp |  Imb. |  Imb. |Group
       |        |  Samp | Samp% | Function
       |        |       |       |  PE=HIDE
       |        |       |       |   Thread=HIDE

 100.0% | 8526.5 |    -- |    -- |Total
|-----------------------------------------------------
|  83.5% | 7121.8 |    -- |    -- |USER
||----------------------------------------------------
||  21.6% | 1843.2 | 139.8 |  7.2% |binvcrhs_
||  12.7% | 1079.1 | 147.9 | 12.2% |z_solve_._omp_fn.0
||  11.4% |  972.9 | 172.1 | 15.3% |y_solve_._omp_fn.0
||  11.2% |  952.1 | 141.9 | 13.2% |x_solve_._omp_fn.0
||  11.0% |  941.8 |  62.2 |  6.3% |matmul_sub_
||  10.2% |  869.6 | 151.4 | 15.1% |compute_rhs_._omp_fn.0
||   3.4% |  286.8 |  44.2 | 13.6% |matvec_sub_
||====================================================
|  11.2% |  956.9 |    -- |    -- |MPI
||----------------------------------------------------
||   9.7% |  825.1 | 376.9 | 31.9% |mpi_waitall
||====================================================
|   2.6% |  224.7 | 618.3 | 74.5% |PTHREAD
||----------------------------------------------------
||   2.6% |  224.7 | 618.3 | 74.5% |pthread_join
||====================================================
|   2.6% |  221.1 |    -- |    -- |ETC
||----------------------------------------------------
||   1.6% |  133.7 | 100.3 | 43.6% |gomp_barrier_wait_end
|=====================================================
```

32CN: aprun -n64 -N2 -d4 ./bt-mz_C.64+pat622

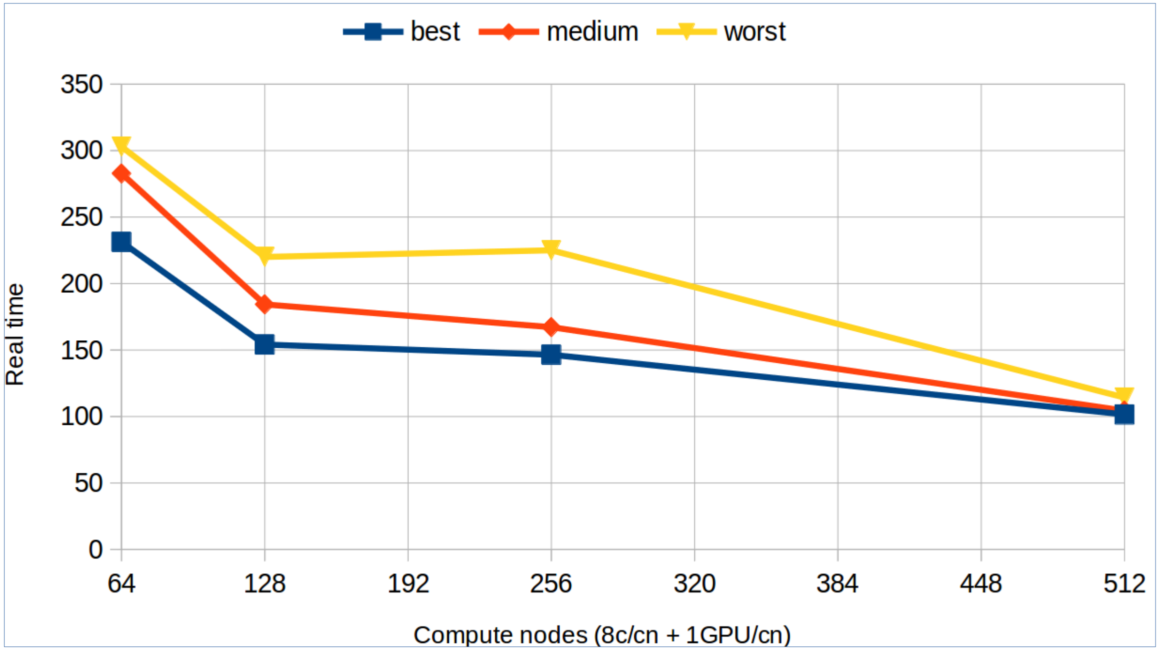
```
Table 1:  Profile by Function Group and Function
(top 10 functions shown)

 Samp% |   Samp |  Imb. |  Imb. |Group
       |        |  Samp | Samp% | Function
       |        |       |       |  PE=HIDE
       |        |       |       |   Thread=HIDE

 100.0% | 5949.8 |    -- |    -- |Total
|-----------------------------------------------------
|  63.5% | 3778.4 |    -- |    -- |USER
||----------------------------------------------------
||  16.6% |  987.3 |  73.7 |  7.1% |binvcrhs_
||   9.4% |  561.4 |  92.6 | 14.4% |z_solve_._omp_fn.0
||   8.7% |  515.4 |  86.6 | 14.6% |y_solve_._omp_fn.0
||   8.6% |  514.3 |  66.7 | 11.7% |x_solve_._omp_fn.0
||   8.5% |  503.6 |  69.4 | 12.3% |matmul_sub_
||   7.5% |  443.5 |  67.5 | 13.4% |compute_rhs_._omp_fn.0
||   2.6% |  157.1 |  28.9 | 15.8% |matvec_sub_
||====================================================
|  13.6% |  806.3 |    -- |    -- |ETC
||----------------------------------------------------
||  11.3% |  671.8 | 212.2 | 24.4% |gomp_barrier_wait_end
||====================================================
|  11.9% |  707.0 | 639.0 | 48.2% |PTHREAD
||----------------------------------------------------
||  11.9% |  707.0 | 639.0 | 48.2% |pthread_join
||====================================================
|  11.0% |  656.2 |    -- |    -- |MPI
||----------------------------------------------------
||   8.4% |  501.7 | 416.3 | 46.1% |mpi_waitall
|=====================================================
```

Demo: MPI+OpenMP / 32CN

GPU codes

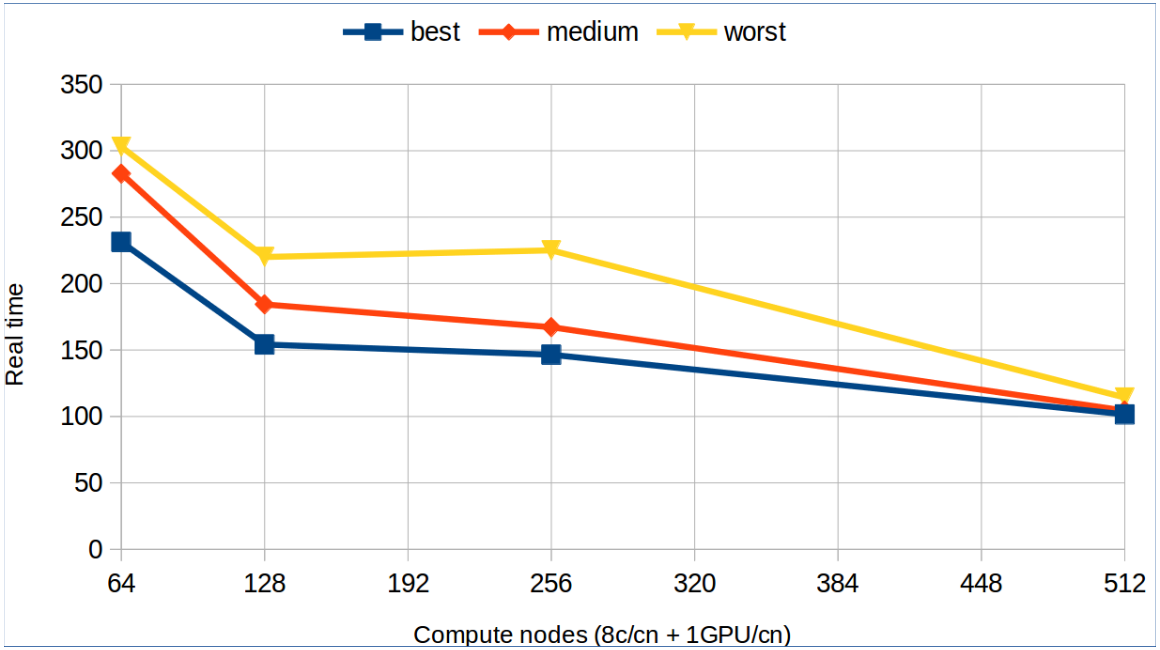
Scalability: Mpi+OpenMP+CUDA code on PizDaint

CN	-n	-N	-d	Real Time
64	256	4	2	231
128	512	4	2	154
256	1024	4	2	146
512	512	1	8	102
1024	1024	1	8	182



Scalability: Mpi+OpenMP+CUDA code on PizDaint

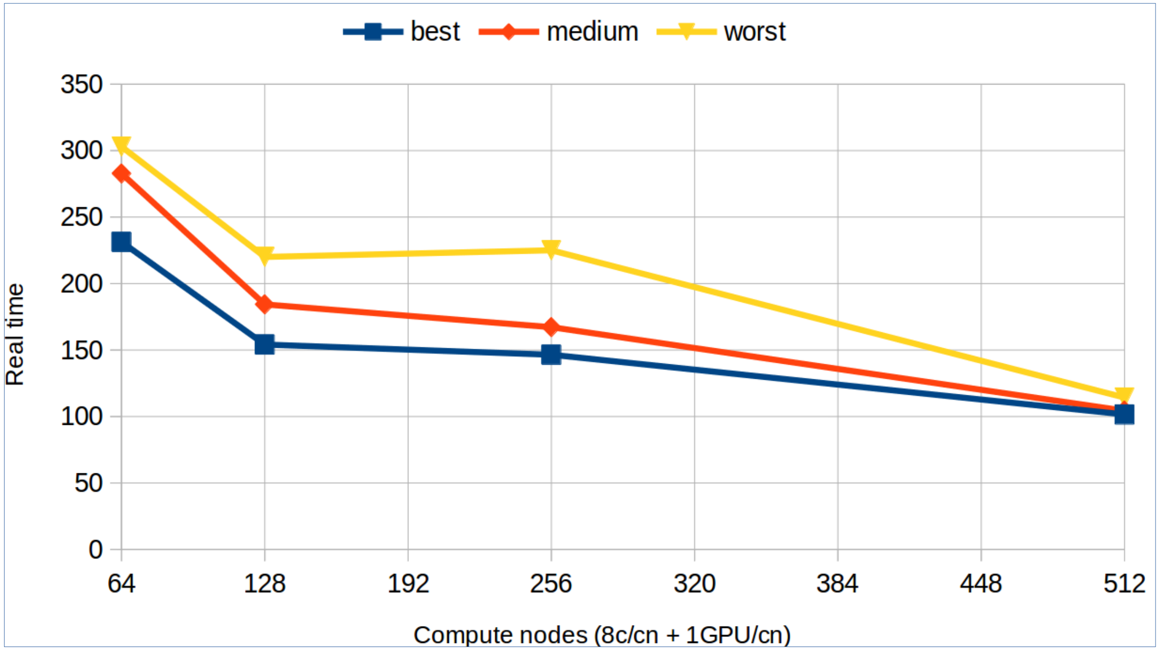
CN	-n	-N	-d	Real Time
64	256	4	2	231
128	512	4	2	154
256	1024	4	2	146
512	512	1	8	102
1024	1024	1	8	182



CN	Speedup	Ispeedup
64	1	1
128	1.5	2
256	1.6	4
512	2.3	8
1024	1.3	16

Scalability: Mpi+OpenMP+CUDA code on PizDaint

CN	-n	-N	-d	Real Time
64	256	4	2	231
128	512	4	2	154
256	1024	4	2	146
512	512	1	8	102
1024	1024	1	8	182

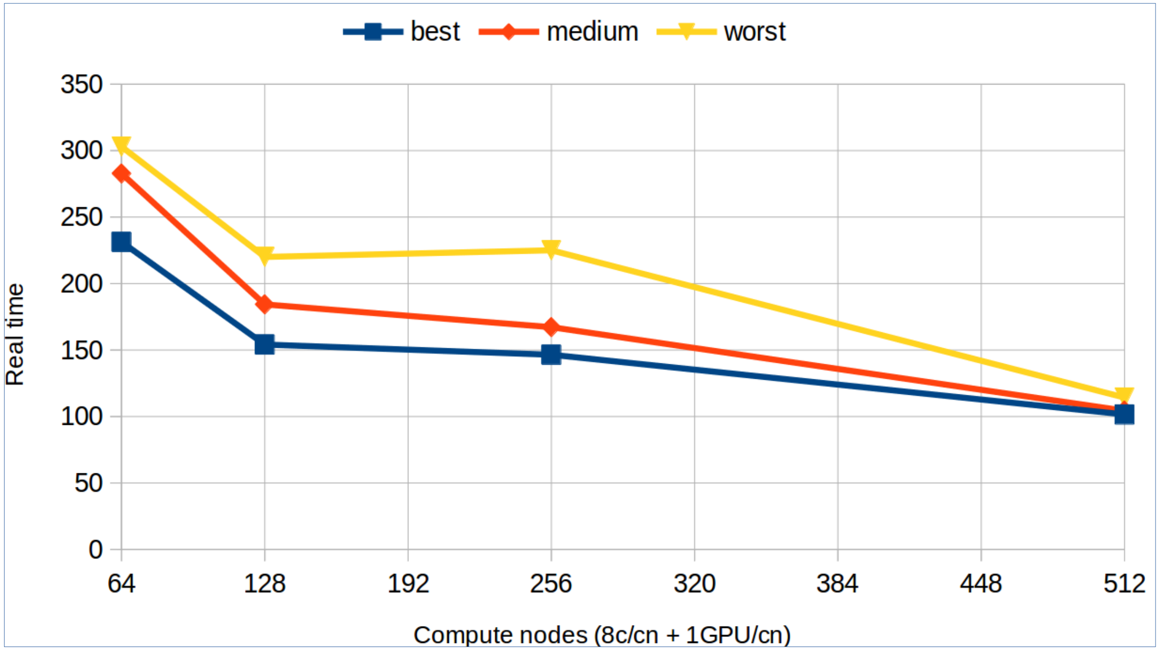


CN	Speedup	Ispeedup
64	1	1
128	1.5	2
256	1.6	4
512	2.3	8
1024	1.3	16

CN	Efficiency
64	100%
128	75%
256	39%
512	28%
1024	8%

Scalability: Mpi+OpenMP+CUDA code on PizDaint

CN	-n	-N	-d	Real Time
64	256	4	2	231
128	512	4	2	154
256	1024	4	2	146
512	512	1	8	102
1024	1024	1	8	182



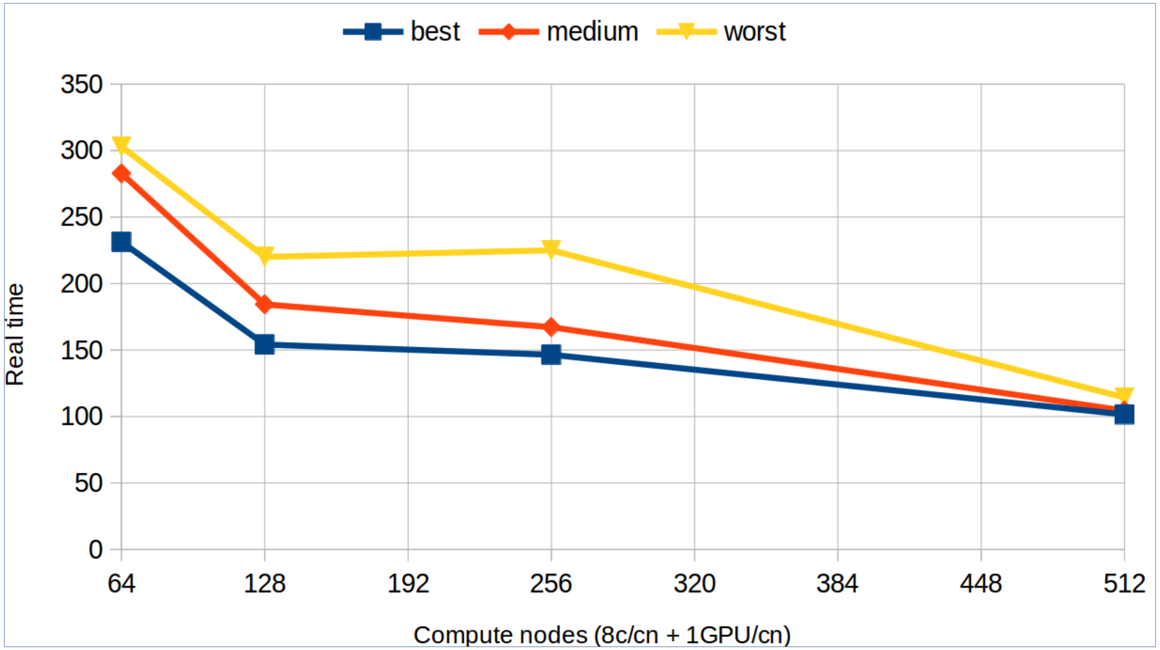
CN	Speedup	Ispeedup
64	1	1
128	1.5	2
256	1.6	4
512	2.3	8
1024	1.3	16

CN	Efficiency
64	100%
128	75%
256	39%
512	28%
1024	8%

→ Scales up to <256 CN

Scalability: Mpi+OpenMP+CUDA code on PizDaint

CN	-n	-N	-d	Real Time
64	256	4	2	231
128	512	4	2	154
256	1024	4	2	146
512	512	1	8	102
1024	1024	1	8	182



CN	Speedup	Ispeedup
64	1	1
128	1.5	2
256	1.6	4
512	2.3	8
1024	1.3	16

CN	Efficiency
64	100%
128	75%
256	39%
512	28%
1024	8%

- Scales up to <256 CN
- What can we learn from the tool ?

Performance: Mpi+OpenMP+CUDA code on PizDaint

128CN: aprun -n128 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |        Time | Imb. Time |   Imb. |       Calls |Group
        |             |           |  Time% |             | Function
        |             |           |        |             |  PE=HIDE
        |             |           |        |             |   Thread=HIDE

 100.0% | 242.303181 |        -- |     -- | 21286787.3 |Total
|-----------------------------------------------------------------------
|  34.9% |  84.471554 |        -- |     -- |    53714.0 |MPI
||----------------------------------------------------------------------
||  28.5% |  69.008385 |  5.581379 |  7.5% |    17106.0 |mpi_waitall_
||   4.3% |  10.528301 |  0.697655 |  6.3% |    12577.9 |mpi_isend_
||======================================================================
|  31.9% |  77.380701 |  7.712314 |  9.1% |        1.0 |USER
||----------------------------------------------------------------------
||  31.9% |  77.380701 |  7.712314 |  9.1% |        1.0 |main
||======================================================================
|  16.8% |  40.822133 |        -- |     -- |   151572.5 |CUDA
||----------------------------------------------------------------------
||  12.8% |  31.095332 | 26.188607 | 46.1% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.880492 |  3.258596 | 46.0% |     7198.0 |cudaEventSynchroniz
||======================================================================
|  10.8% |  26.201472 |        -- |     -- |     7140.0 |MPI_SYNC
||----------------------------------------------------------------------
||   3.9% |   9.506842 |  8.881698 | 93.4% |     2711.0 |mpi_bcast_(sync)
||   3.6% |   8.744357 |  3.631879 | 41.5% |     3906.0 |mpi_allreduce_(sync
||   1.8% |   4.367063 |  1.808994 | 41.4% |      262.0 |mpi_alltoall_(sync)
||======================================================================
|   5.1% |  12.477306 |        -- |     -- | 19062275.1 |PTHREAD
||----------------------------------------------------------------------
||   5.1% |  12.476856 |  3.100570 | 20.1% | 19062267.7 |pthread_mutex_lock
|=======================================================================
```

512CN: aprun -n512 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |        Time | Imb. Time |   Imb. |       Calls |Group
        |             |           |  Time% |             | Function
        |             |           |        |             |  PE=HIDE
        |             |           |        |             |   Thread=HIDE

 100.0% | 209.075899 |        -- |     -- | 15732046.4 |Total
|-----------------------------------------------------------------------
|  29.9% |  62.581357 |        -- |     -- |   275745.7 |CUDA
||----------------------------------------------------------------------
||  26.7% |  55.804788 | 61.774329 |  52.6% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.379988 |  3.367774 |  50.0% |    14750.0 |cudaEventSynchronize
||======================================================================
|  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |USER
||----------------------------------------------------------------------
||  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |main
||======================================================================
|  22.7% |  47.487831 |        -- |     -- |    98947.0 |MPI
||----------------------------------------------------------------------
||  16.0% |  33.532853 |  4.743034 |  12.4% |    32234.0 |mpi_waitall_
||   2.5% |   5.189488 |  0.562764 |   9.8% |    25381.2 |mpi_isend_
||   1.8% |   3.693452 |  0.141373 |   3.7% |      133.0 |mpi_alltoallv_
||======================================================================
|  17.4% |  36.332651 |        -- |     -- |     7152.0 |MPI_SYNC
||----------------------------------------------------------------------
||  11.1% |  23.206718 | 22.498565 |  96.9% |     2709.0 |mpi_bcast_(sync)
||   2.9% |   6.030251 |  2.064512 |  34.2% |     3900.0 |mpi_allreduce_(sync)
||   2.0% |   4.128165 |  4.128102 | 100.0% |        1.0 |mpi_init_thread_(sync)
||======================================================================
|   3.0% |   6.228465 |        -- |     -- | 14276828.2 |PTHREAD
||----------------------------------------------------------------------
||   3.0% |   6.228044 |  4.239491 |  40.6% | 14276820.9 |pthread_mutex_lock
|=======================================================================
```

Performance: Mpi+OpenMP+CUDA code on PizDaint

128CN: aprun -n128 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |        Time | Imb. Time |  Imb.  |       Calls |Group
        |             |           | Time%  |             | Function
        |             |           |        |             |  PE=HIDE
        |             |           |        |             |   Thread=HIDE

 100.0% | 242.303181 |        -- |     -- | 21286787.3 |Total
|-----------------------------------------------------------------------
|  34.9% |  84.471554 |        -- |     -- |    53714.0 |MPI
||----------------------------------------------------------------------
||  28.5% |  69.008385 |  5.581379 |   7.5% |    17106.0 |mpi_waitall_
||   4.3% |  10.528301 |  0.697655 |   6.3% |    12577.9 |mpi_isend_
||======================================================================
|  31.9% |  77.380701 |  7.712314 |   9.1% |        1.0 |USER
||----------------------------------------------------------------------
||  31.9% |  77.380701 |  7.712314 |   9.1% |        1.0 |main
||======================================================================
|  16.8% |  40.822133 |        -- |     -- |   151572.5 |CUDA
||----------------------------------------------------------------------
||  12.8% |  31.095332 | 26.188607 |  46.1% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.880492 |  3.258596 |  46.0% |     7198.0 |cudaEventSynchroniz
||======================================================================
|  10.8% |  26.201472 |        -- |     -- |     7140.0 |MPI_SYNC
||----------------------------------------------------------------------
||   3.9% |   9.506842 |  8.881698 |  93.4% |     2711.0 |mpi_bcast_(sync)
||   3.6% |   8.744357 |  3.631879 |  41.5% |     3906.0 |mpi_allreduce_(sync
||   1.8% |   4.367063 |  1.808994 |  41.4% |      262.0 |mpi_alltoall_(sync)
||======================================================================
|   5.1% |  12.477306 |        -- |     -- | 19062275.1 |PTHREAD
||----------------------------------------------------------------------
||   5.1% |  12.476856 |  3.100570 |  20.1% | 19062267.7 |pthread_mutex_lock
|=======================================================================
```

CN	USER	MPI*	CUDA	ELSE
128	31.9%	45.7%	16.8%	5.6%
256	28.2%	41.4%	26.2%	4.2%
512	26.6%	40.1%	29.9%	3.4%

512CN: aprun -n512 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |        Time | Imb. Time |  Imb.  |       Calls |Group
        |             |           | Time%  |             | Function
        |             |           |        |             |  PE=HIDE
        |             |           |        |             |   Thread=HIDE

 100.0% | 209.075899 |        -- |     -- | 15732046.4 |Total
|-----------------------------------------------------------------------
|  29.9% |  62.581357 |        -- |     -- |   275745.7 |CUDA
||----------------------------------------------------------------------
||  26.7% |  55.804788 | 61.774329 |  52.6% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.379988 |  3.367774 |  50.0% |    14750.0 |cudaEventSynchronize
||======================================================================
|  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |USER
||----------------------------------------------------------------------
||  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |main
||======================================================================
|  22.7% |  47.487831 |        -- |     -- |    98947.0 |MPI
||----------------------------------------------------------------------
||  16.0% |  33.532853 |  4.743034 |  12.4% |    32234.0 |mpi_waitall_
||   2.5% |   5.189488 |  0.562764 |   9.8% |    25381.2 |mpi_isend_
||   1.8% |   3.693452 |  0.141373 |   3.7% |      133.0 |mpi_alltoallv_
||======================================================================
|  17.4% |  36.332651 |        -- |     -- |     7152.0 |MPI_SYNC
||----------------------------------------------------------------------
||  11.1% |  23.206718 | 22.498565 |  96.9% |     2709.0 |mpi_bcast_(sync)
||   2.9% |   6.030251 |  2.064512 |  34.2% |     3900.0 |mpi_allreduce_(sync)
||   2.0% |   4.128165 |  4.128102 | 100.0% |        1.0 |mpi_init_thread_(sync)
||======================================================================
|   3.0% |   6.228465 |        -- |     -- | 14276828.2 |PTHREAD
||----------------------------------------------------------------------
||   3.0% |   6.228044 |  4.239491 |  40.6% | 14276820.9 |pthread_mutex_lock
|=======================================================================
```

Performance: Mpi+OpenMP+CUDA code on PizDaint

128CN: aprun -n128 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |       Time | Imb. Time |   Imb. |      Calls |Group
        |            |           |  Time% |            | Function
        |            |           |        |            |  PE=HIDE
        |            |           |        |            |   Thread=HIDE

 100.0% | 242.303181 |        -- |     -- | 21286787.3 |Total
|-----------------------------------------------------------------------
|  34.9% |  84.471554 |        -- |     -- |    53714.0 |MPI
||----------------------------------------------------------------------
||  28.5% |  69.008385 |  5.581379 |   7.5% |    17106.0 |mpi_waitall_
||   4.3% |  10.528301 |  0.697655 |   6.3% |    12577.9 |mpi_isend_
||======================================================================
|  31.9% |  77.380701 |  7.712314 |   9.1% |        1.0 |USER
||----------------------------------------------------------------------
||  31.9% |  77.380701 |  7.712314 |   9.1% |        1.0 |main
||======================================================================
|  16.8% |  40.822133 |        -- |     -- |   151572.5 |CUDA
||----------------------------------------------------------------------
||  12.8% |  31.095332 | 26.188607 |  46.1% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.880492 |  3.258596 |  46.0% |     7198.0 |cudaEventSynchroniz
||======================================================================
|  10.8% |  26.201472 |        -- |     -- |     7140.0 |MPI_SYNC
||----------------------------------------------------------------------
||   3.9% |   9.506842 |  8.881698 |  93.4% |     2711.0 |mpi_bcast_(sync)
||   3.6% |   8.744357 |  3.631879 |  41.5% |     3906.0 |mpi_allreduce_(sync
||   1.8% |   4.367063 |  1.808994 |  41.4% |      262.0 |mpi_alltoall_(sync)
||======================================================================
|   5.1% |  12.477306 |        -- |     -- | 19062275.1 |PTHREAD
||----------------------------------------------------------------------
||   5.1% |  12.476856 |  3.100570 |  20.1% | 19062267.7 |pthread_mutex_lock
|=======================================================================
```

CN	USER	MPI*	CUDA	ELSE
128	31.9%	45.7%	16.8%	5.6%
256	28.2%	41.4%	26.2%	4.2%
512	26.6%	40.1%	29.9%	3.4%

512CN: aprun -n512 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |       Time | Imb. Time |   Imb. |      Calls |Group
        |            |           |  Time% |            | Function
        |            |           |        |            |  PE=HIDE
        |            |           |        |            |   Thread=HIDE

 100.0% | 209.075899 |        -- |     -- | 15732046.4 |Total
|-----------------------------------------------------------------------
|  29.9% |  62.581357 |        -- |     -- |   275745.7 |CUDA
||----------------------------------------------------------------------
||  26.7% |  55.804788 | 61.774329 |  52.6% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.379988 |  3.367774 |  50.0% |    14750.0 |cudaEventSynchronize
||======================================================================
|  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |USER
||----------------------------------------------------------------------
||  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |main
||======================================================================
|  22.7% |  47.487831 |        -- |     -- |    98947.0 |MPI
||----------------------------------------------------------------------
||  16.0% |  33.532853 |  4.743034 |  12.4% |    32234.0 |mpi_waitall_
||   2.5% |   5.189488 |  0.562764 |   9.8% |    25381.2 |mpi_isend_
||   1.8% |   3.693452 |  0.141373 |   3.7% |      133.0 |mpi_alltoallv_
||======================================================================
|  17.4% |  36.332651 |        -- |     -- |     7152.0 |MPI_SYNC
||----------------------------------------------------------------------
||  11.1% |  23.206718 | 22.498565 |  96.9% |     2709.0 |mpi_bcast_(sync)
||   2.9% |   6.030251 |  2.064512 |  34.2% |     3900.0 |mpi_allreduce_(sync)
||   2.0% |   4.128165 |  4.128102 | 100.0% |        1.0 |mpi_init_thread_(sync)
||======================================================================
|   3.0% |   6.228465 |        -- |     -- | 14276828.2 |PTHREAD
||----------------------------------------------------------------------
||   3.0% |   6.228044 |  4.239491 |  40.6% | 14276820.9 |pthread_mutex_lock
|=======================================================================
```

Performance: Mpi+OpenMP+CUDA code on PizDaint

128CN: aprun -n128 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |        Time | Imb. Time |   Imb. |        Calls |Group
        |             |           |  Time% |              | Function
        |             |           |        |              |  PE=HIDE
        |             |           |        |              |   Thread=HIDE

 100.0% | 242.303181 |        -- |     -- | 21286787.3 |Total
|-----------------------------------------------------------------------------
|  34.9% |  84.471554 |        -- |     -- |    53714.0 |MPI
||----------------------------------------------------------------------------
||  28.5% |  69.008385 |  5.581379 |   7.5% |    17106.0 |mpi_waitall_
||   4.3% |  10.528301 |  0.697655 |   6.3% |    12577.9 |mpi_isend_
||============================================================================
|  31.9% |  77.380701 |  7.712314 |   9.1% |        1.0 |USER
||----------------------------------------------------------------------------
||  31.9% |  77.380701 |  7.712314 |   9.1% |        1.0 |main
||============================================================================
|  16.8% |  40.822133 |        -- |     -- |   151572.5 |CUDA
||----------------------------------------------------------------------------
||  12.8% |  31.095332 | 26.188607 |  46.1% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.880492 |  3.258596 |  46.0% |     7198.0 |cudaEventSynchroniz
||============================================================================
|  10.8% |  26.201472 |        -- |     -- |     7140.0 |MPI_SYNC
||----------------------------------------------------------------------------
||   3.9% |   9.506842 |  8.881698 |  93.4% |     2711.0 |mpi_bcast_(sync)
||   3.6% |   8.744357 |  3.631879 |  41.5% |     3906.0 |mpi_allreduce_(sync
||   1.8% |   4.367063 |  1.808994 |  41.4% |      262.0 |mpi_alltoall_(sync)
||============================================================================
|   5.1% |  12.477306 |        -- |     -- | 19062275.1 |PTHREAD
||----------------------------------------------------------------------------
||   5.1% |  12.476856 |  3.100570 |  20.1% | 19062267.7 |pthread_mutex_lock
|=============================================================================
```

512CN: aprun -n512 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

  Time% |        Time | Imb. Time |   Imb. |        Calls |Group
        |             |           |  Time% |              | Function
        |             |           |        |              |  PE=HIDE
        |             |           |        |              |   Thread=HIDE

 100.0% | 209.075899 |        -- |     -- | 15732046.4 |Total
|-----------------------------------------------------------------------------
|  29.9% |  62.581357 |        -- |     -- |   275745.7 |CUDA
||----------------------------------------------------------------------------
||  26.7% |  55.804788 | 61.774329 |  52.6% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.379988 |  3.367774 |  50.0% |    14750.0 |cudaEventSynchronize
||============================================================================
|  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |USER
||----------------------------------------------------------------------------
||  26.6% |  55.544639 | 29.564967 |  34.8% |        1.0 |main
||============================================================================
|  22.7% |  47.487831 |        -- |     -- |    98947.0 |MPI
||----------------------------------------------------------------------------
||  16.0% |  33.532853 |  4.743034 |  12.4% |    32234.0 |mpi_waitall_
||   2.5% |   5.189488 |  0.562764 |   9.8% |    25381.2 |mpi_isend_
||   1.8% |   3.693452 |  0.141373 |   3.7% |      133.0 |mpi_alltoallv_
||============================================================================
|  17.4% |  36.332651 |        -- |     -- |     7152.0 |MPI_SYNC
||----------------------------------------------------------------------------
||  11.1% |  23.206718 | 22.498565 |  96.9% |     2709.0 |mpi_bcast_(sync)
||   2.9% |   6.030251 |  2.064512 |  34.2% |     3900.0 |mpi_allreduce_(sync)
||   2.0% |   4.128165 |  4.128102 | 100.0% |        1.0 |mpi_init_thread_(sync)
||============================================================================
|   3.0% |   6.228465 |        -- |     -- | 14276828.2 |PTHREAD
||----------------------------------------------------------------------------
||   3.0% |   6.228044 |  4.239491 |  40.6% | 14276820.9 |pthread_mutex_lock
|=============================================================================
```

CN	USER	MPI*	CUDA	ELSE
128	31.9%	45.7%	16.8%	5.6%
256	28.2%	41.4%	26.2%	4.2%
512	26.6%	40.1%	29.9%	3.4%

Performance: Mpi+OpenMP+CUDA code on PizDaint

128CN: aprun -n128 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

 Time% |       Time | Imb. Time |  Imb.  |      Calls |Group
       |            |           |  Time% |            | Function
       |            |           |        |            |  PE=HIDE
       |            |           |        |            |   Thread=HIDE

100.0% | 242.303181 |        -- |     -- | 21286787.3 |Total
|-----------------------------------------------------------------------
|  34.9% |  84.471554 |        -- |     -- |    53714.0 |MPI
||----------------------------------------------------------------------
||  28.5% |  69.008385 |  5.581379 |  7.5% |    17106.0 |mpi_waitall_
||   4.3% |  10.528301 |  0.697655 |  6.3% |    12577.9 |mpi_isend_
||======================================================================
|  31.9% |  77.380701 |  7.712314 |  9.1% |        1.0 |USER
||----------------------------------------------------------------------
||  31.9% |  77.380701 |  7.712314 |  9.1% |        1.0 |main
||======================================================================
|  16.8% |  40.822133 |        -- |     -- |   151572.5 |CUDA
||----------------------------------------------------------------------
||  12.8% |  31.095332 | 26.188607 | 46.1% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.880492 |  3.258596 | 46.0% |     7198.0 |cudaEventSynchroniz
||======================================================================
|  10.8% |  26.201472 |        -- |     -- |     7140.0 |MPI_SYNC
||----------------------------------------------------------------------
||   3.9% |   9.506842 |  8.881698 | 93.4% |     2711.0 |mpi_bcast_(sync)
||   3.6% |   8.744357 |  3.631879 | 41.5% |     3906.0 |mpi_allreduce_(sync
||   1.8% |   4.367063 |  1.808994 | 41.4% |      262.0 |mpi_alltoall_(sync)
||======================================================================
|   5.1% |  12.477306 |        -- |     -- | 19062275.1 |PTHREAD
||----------------------------------------------------------------------
||   5.1% |  12.476856 |  3.100570 | 20.1% | 19062267.7 |pthread_mutex_lock
|=======================================================================
```

512CN: aprun -n512 -N1 -d8 cp2k+pat622

```
Table 1:  Profile by Function Group and Function

 Time% |       Time | Imb. Time |  Imb.  |      Calls |Group
       |            |           |  Time% |            | Function
       |            |           |        |            |  PE=HIDE
       |            |           |        |            |   Thread=HIDE

100.0% | 209.075899 |        -- |     -- | 15732046.4 |Total
|-----------------------------------------------------------------------
|  29.9% |  62.581357 |        -- |     -- |   275745.7 |CUDA
||----------------------------------------------------------------------
||  26.7% |  55.804788 | 61.774329 | 52.6% |        4.0 |cudaGetDeviceCount
||   1.6% |   3.379988 |  3.367774 | 50.0% |    14750.0 |cudaEventSynchronize
||======================================================================
|  26.6% |  55.544639 | 29.564967 | 34.8% |        1.0 |USER
||----------------------------------------------------------------------
||  26.6% |  55.544639 | 29.564967 | 34.8% |        1.0 |main
||======================================================================
|  22.7% |  47.487831 |        -- |     -- |    98947.0 |MPI
||----------------------------------------------------------------------
||  16.0% |  33.532853 |  4.743034 | 12.4% |    32234.0 |mpi_waitall_
||   2.5% |   5.189488 |  0.562764 |  9.8% |    25381.2 |mpi_isend_
||   1.8% |   3.693452 |  0.141373 |  3.7% |      133.0 |mpi_alltoallv_
||======================================================================
|  17.4% |  36.332651 |        -- |     -- |     7152.0 |MPI_SYNC
||----------------------------------------------------------------------
||  11.1% |  23.206718 | 22.498565 | 96.9% |     2709.0 |mpi_bcast_(sync)
||   2.9% |   6.030251 |  2.064512 | 34.2% |     3900.0 |mpi_allreduce_(sync)
||   2.0% |   4.128165 |  4.128102 |100.0% |        1.0 |mpi_init_thread_(sync)
||======================================================================
|   3.0% |   6.228465 |        -- |     -- | 14276828.2 |PTHREAD
||----------------------------------------------------------------------
||   3.0% |   6.228044 |  4.239491 | 40.6% | 14276820.9 |pthread_mutex_lock
|=======================================================================
```

CN	USER	MPI*	CUDA	ELSE
128	31.9%	45.7%	16.8%	5.6%
256	28.2%	41.4%	26.2%	4.2%
512	26.6%	40.1%	29.9%	3.4%

- To allow multiple CPU cores to simultaneously utilize a single GPU, the CUDA proxy must be enabled.
- Performance tools are only supported with the CUDA proxy disabled

CRAY_CUDA_MPS=1

CRAY_CUDA_MPS=0

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Conclusion

Recipe for a successfull technical review

Project Description Sections

☑ I hereby confirm that my application is complete and contains all of the following sections:

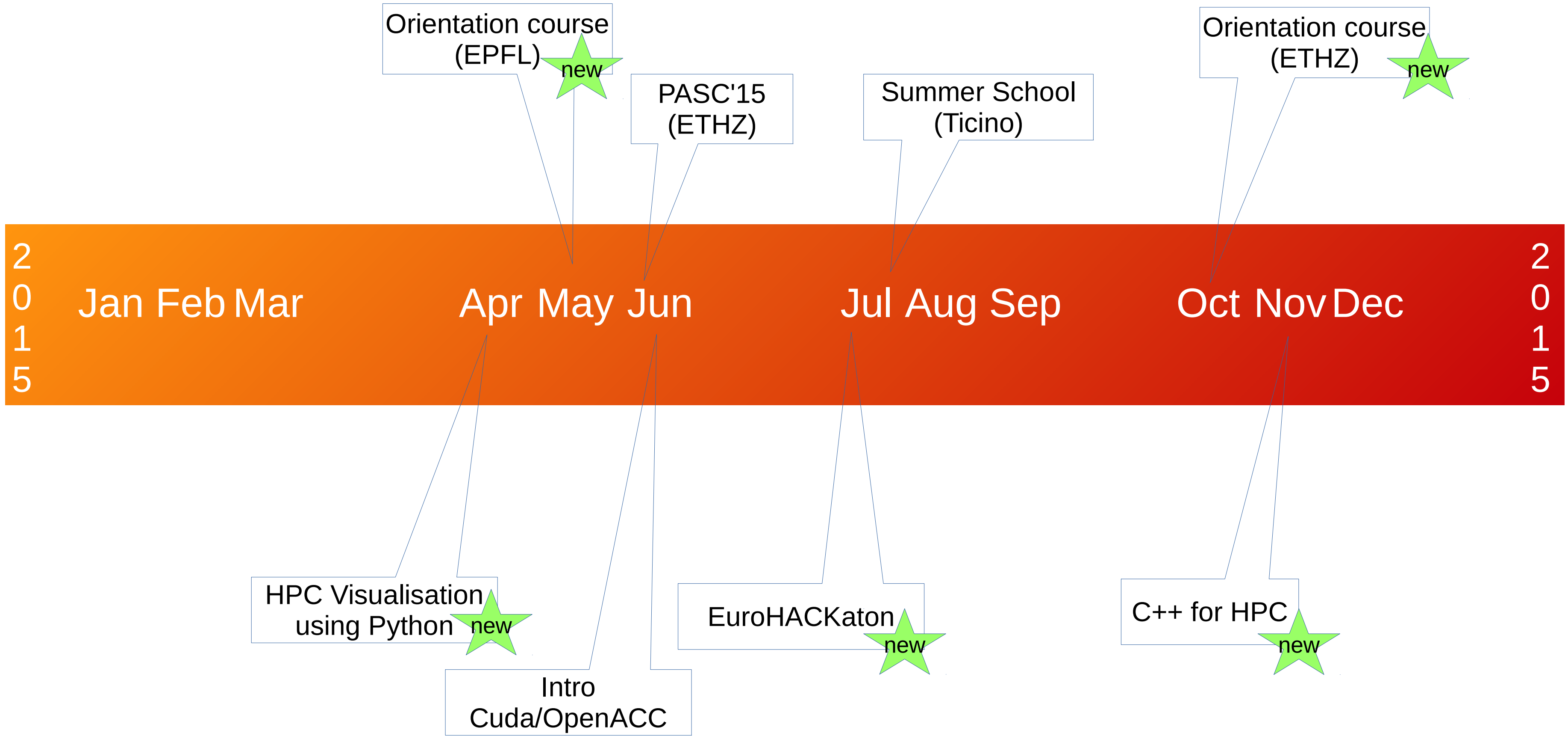
- ✓ Abstract
- ✓ Scientific goals and objectives
- ✓ Background and significance
- ✓ Research methods, algorithms, and codes
- ✓ Performance analysis
- ✓ Representative benchmarks and scaling
- ✓ Resources justification
- ✓ Parallelization approach
- ✓ Project plans: tasks and milestones

If the answer to any of the following question is No…

- Does your code run on Piz Daint ?
- Did you provide scalability data from Piz Daint ?
- Did you attach the .rpt performance report file(s) ?
- Is your resource request consistent with your scientific goals ?
- Does the project plan fit the allocated time frame ?

… there is a risk that your proposal will not be accepted

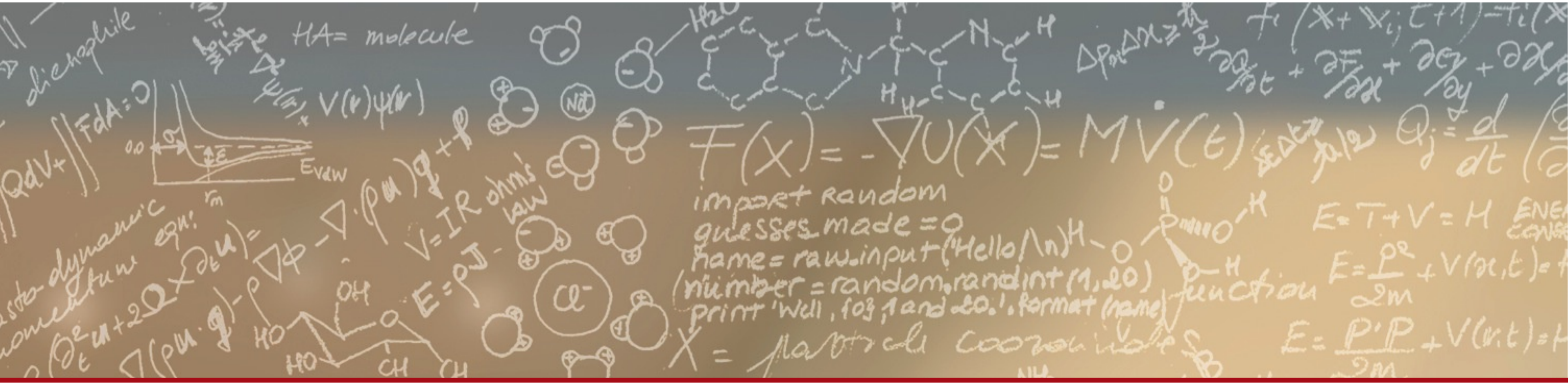
Future events (2015)



Online registration ===> http://www.cscs.ch/events

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.

Questions ?