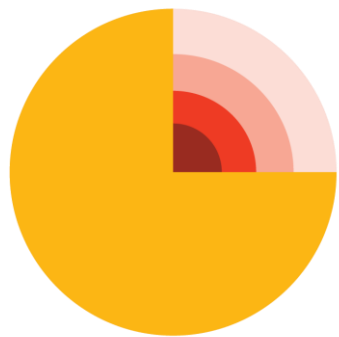




ChEESE

Partner: BSC - CNS

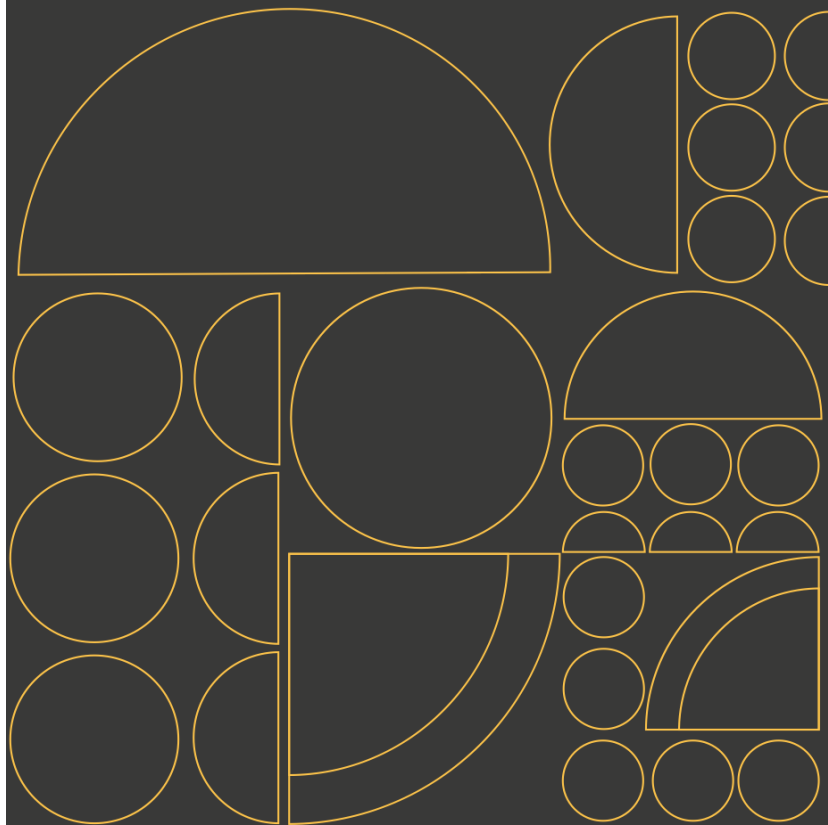


**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

Marta García
Victor López

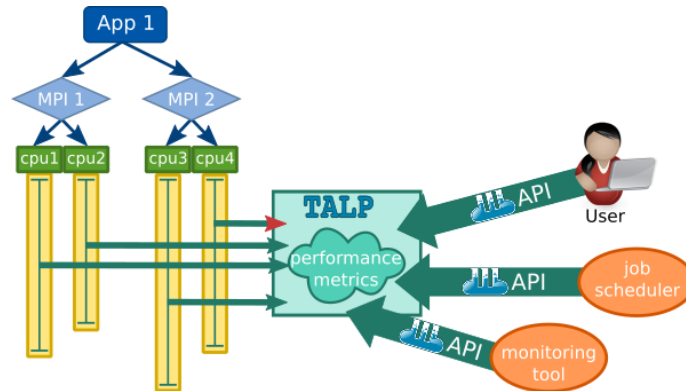


Project funded by EuroHPC under the grant agreement No 101093038.



Task 2.4 Status Update M18 (BSC)

- Task span M18-M48
- TALP generates profiling metrics reports defined by POP
- It will be extended as necessary to support the projects codes
- Support for user-defined code regions



<https://pm.bsc.es/dlb>

TALP metrics report

- Pure MPI
- Hybrid MPI + OpenMP

DLB[...]: ##### Monitoring Region POP Metrics

#####

DLB[...]: ### Name: MPI

Execution

DLB[...]: ### Elapsed Time : 12 s

DLB[...]: ### Parallel efficiency : 0.66

DLB[...]: ### - Communication eff. : 0.90

DLB[...]: ### - Load Balance : 0.73

DLB[...]: ### - LB_in : 0.73

DLB[...]: ### - LB_out : 1.00

DLB[...]: ##### Monitoring Region POP Metrics

#####

DLB[...]: ### Name:

Application

DLB[...]: ### Elapsed Time: 5 s

DLB[...]: ### Parallel efficiency: 0.30

DLB[...]: ### - MPI Parallel efficiency: 0.75

DLB[...]: ### - MPI Communication efficiency: 0.88

DLB[...]: ### - MPI Load Balance: 0.86

DLB[...]: ### - MPI Load Balance in: 0.86

DLB[...]: ### - MPI Load Balance out: 1.00

DLB[...]: ### - OpenMP Parallel efficiency: 0.40

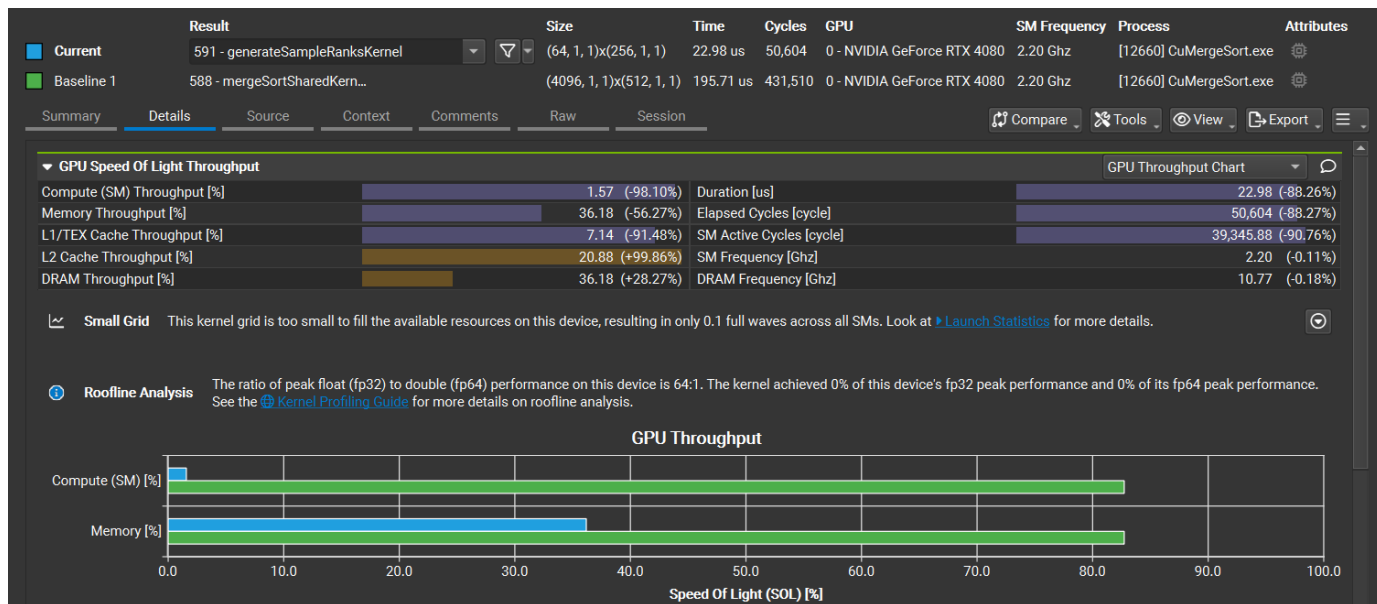
DLB[...]: ### - OpenMP Load Balance: 0.70

DLB[...]: ### - OpenMP Scheduling efficiency: 1.00

DLB[...]: ### - OpenMP Serialization efficiency: 0.57

TALP and GPU metrics

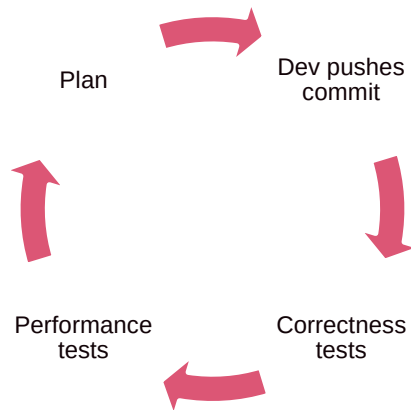
- Current status: early stage research



source: <https://docs.nvidia.com/nsight-compute/NsightCompute/>

TALP Pages (for continuous benchmarking)

- Similar CI/CD cycle:



- Monthly “big” tests to see if bigger runs see any improvement
- On Code Merge, test if performance is affected
- POP-Metrics give user hints on scalability bugs and provide a per-region metric to spot regions where optimization efforts have a larger impact

TALP Pages, usability in gitlab-ci.yml

```
include:
- remote: https://pm.bsc.es/gitlab/dlb/talp-pages/-/raw/add-v1/gitlab-templates/add-to-db/template.yml
  inputs:
    stage: deploy
    generating_job: talp-performance-run
    talp_output: ./talp.json
    environment: production
    gitlab_url: https://gitlab.com
    project_name: valentin.seitz1/sample-application
    job_name: talp-create-artifacts
- remote: https://pm.bsc.es/gitlab/dlb/talp-pages/-/raw/add-v1/gitlab-templates/generate-html/template.yml
  inputs:
    stage: deploy
    generating_job: talp-create-artifacts
    environment: production
    job_name: talp-gen-html

pages:
  stage: deploy
  script:
    - echo "building pages..."
  rules:
    # This ensures that only pushes to the default branch will trigger
    # a pages deploy
    - if: $CI_COMMIT_REF_NAME == $CI_DEFAULT_BRANCH
      needs:
        - job: talp-gen-html
          optional: true
  artifacts:
    paths:
      - public
    expire_in: 5min
```

Some job that
generates a talp.json



Case monthly job:

[Home](#)[Performance Report](#)[Time Series Report](#)

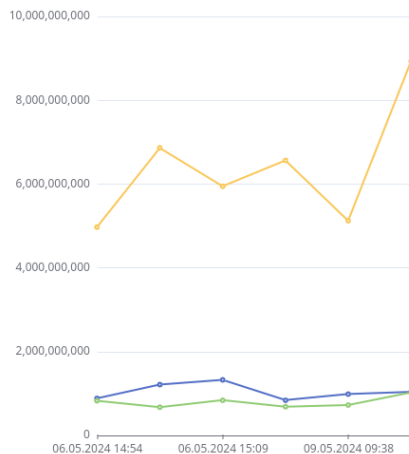
Monthly Regression: 30 nodes

Select Regions

[calc_e](#)[mpi_allreduce](#)[exchange](#)

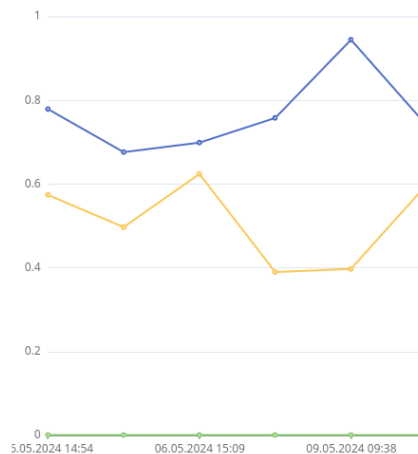
Elapsed Time

calc_e mpi_allreduce exchange



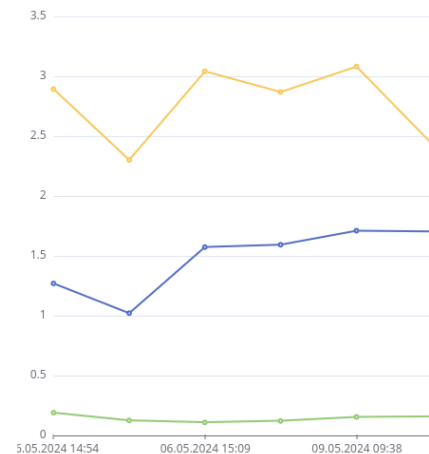
Parallel Efficiency

calc_e mpi_allreduce exchange



Average IPC

calc_e mpi_allreduce exchange



Case monthly job:

[Home](#)[Performance Report](#)[Time Series Report](#)

Monthly Regression: 30 nodes

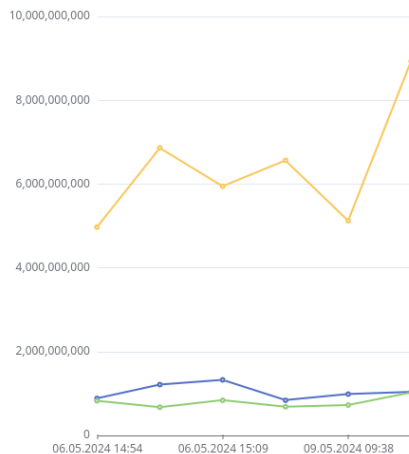
Select Regions

calc_e x mpi_allreduce x exchange x

Ability to select regions of interest

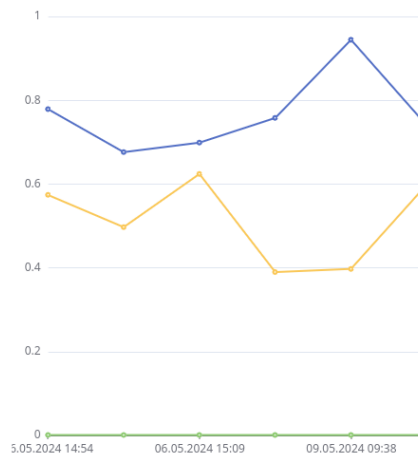
Elapsed Time

calc_e mpi_allreduce exchange



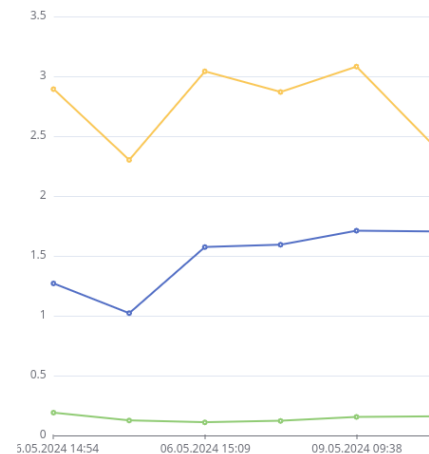
Parallel Efficiency

calc_e mpi_allreduce exchange



Average IPC

calc_e mpi_allreduce exchange



Case on Merge request job:

[Home](#)[Performance Report](#)[Time Series Report](#)

commit: f025c94

Select Regions

calc_e x mpi_allreduce x exchange x

Easy feedback to the devs on how the changes affect performance in a production like run

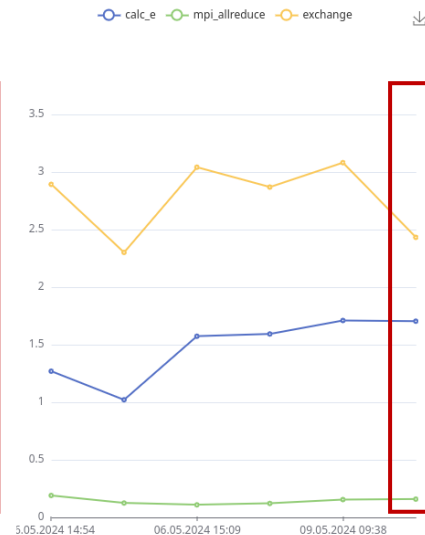
Elapsed Time



Parallel Efficiency



Average IPC



BEPPP - Barcelona Supercomputing Center



Case POP metrics for optimization potential:

calc_g for example suffers a low avg IPC → more in depth analysis might reveal optimization possibilities

Performance Report								
			Search:					
Name	Elapsed Time [s]	Average IPC	Parallel Efficiency	Communication Efficiency	Load Balance	LB In	LB Out	
MPI Execution	167.560521226	1.6334819381116237	0.6037819888683377	0.98	1.0	1.0	1.0	
app	153.482426591	2.766653091699078	0.7499946192281574	0.98	1.0	1.0	1.0	
calc_b	68.121775994	2.486681403822595	1.1772474561170319	1.0	1.0	1.0	1.0	
calc_i	63.228592986	3.370696866368961	1.2123868493721368	1.0	1.0	1.0	1.0	
initialize_mesh	35.46298392	2.537619981086314	0.8420539857735099	1.0	1.0	1.0	1.0	
initialize	32.295353057	3.015738585145399	1.4287471326482328	1.0	1.0	1.0	1.0	
calc_c	21.653503626	2.48448804262556	0.9783600133904361	0.98	1.0	1.0	1.0	
calc_l	20.264636507	1.5112202343277443	0.8922392800041579	1.0	1.0	1.0	1.0	
calc_d	17.649956336	3.284787689685336	0.8311903102603984	1.0	1.0	1.0	1.0	
calc_j	15.726633811	2.8814025453043333	0.8870878883901281	1.0	1.0	1.0	1.0	
initialize_timestep	14.136713008	1.9269123185318138	0.8690561228859646	1.0	1.0	1.0	1.0	
exchange	8.918168871	2.4282831561931753	0.5978473439282778	0.47	1.0	1.0	1.0	
calc_k	5.598190275	2.5415962018040146	1.289500437993429	1.0	1.0	1.0	1.0	
file_io	3.816951781	1.4886295267880318	1.1367320743365115	0.97	1.0	1.0	1.0	
calc_g	2.576887459	0.8871017273775169	1.000146336843625	0.91	1.0	1.0	1.0	
calc_a	2.184751881	0.4129924761480276	1.2253329826530492	1.0	1.0	1.0	1.0	
calc_h	1.601134431	0.3732263530206493	1.3383294354741504	1.0	1.0	1.0	1.0	
add	1.258311518	0.20183895991244213	1.0827008291465605	1.0	1.0	1.0	1.0	
lin_comb	1.146164902	0.41747730764653374	1.444563112291878	1.0	1.0	1.0	1.0	
calc_e	1.025941493	1.70035940275781	0.738705280156983	0.66	1.0	1.0	1.0	
mpi_allreduce	1.012698238	0.15461099514248505	0.0	0.0	1.0	1.0	1.0	
calc_f	0.959764646	1.8766987759190952	1.098657730009745	0.77	1.0	1.0	1.0	
copy	0.238513457	0.32682763696909206	1.0133126867193258	1.0	1.0	1.0	1.0	

Metrics overview

The coloring shown above follows the following rules:

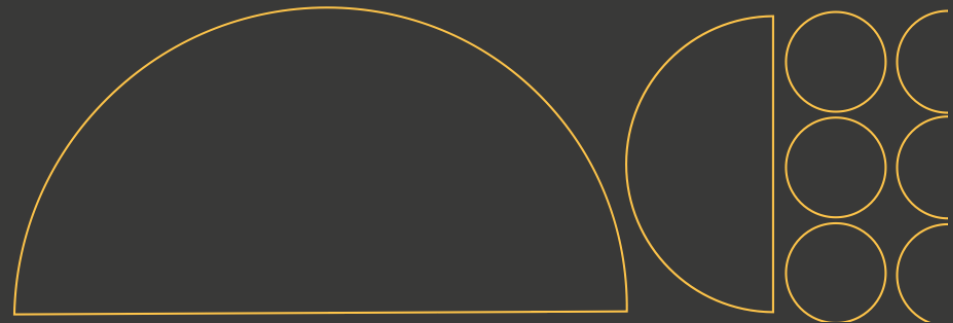
Average IPC Parallel Efficiency

<1	< 0.6
>1	> 0.6
<2	< 0.8
>2	> 0.8

For up to date overview of the computed metrics you can consult the documentation via:

Show in-depth explanation

Thank you!



@cheese-coe



<http://cheese2.eu>



@cheese-coe