ChEESE

TALP-Pages

Valentin Seitz[1], Jordy Trilaksono[2], Marta Garcia-Gasulla[1]

valentin.seitz@bsc.es

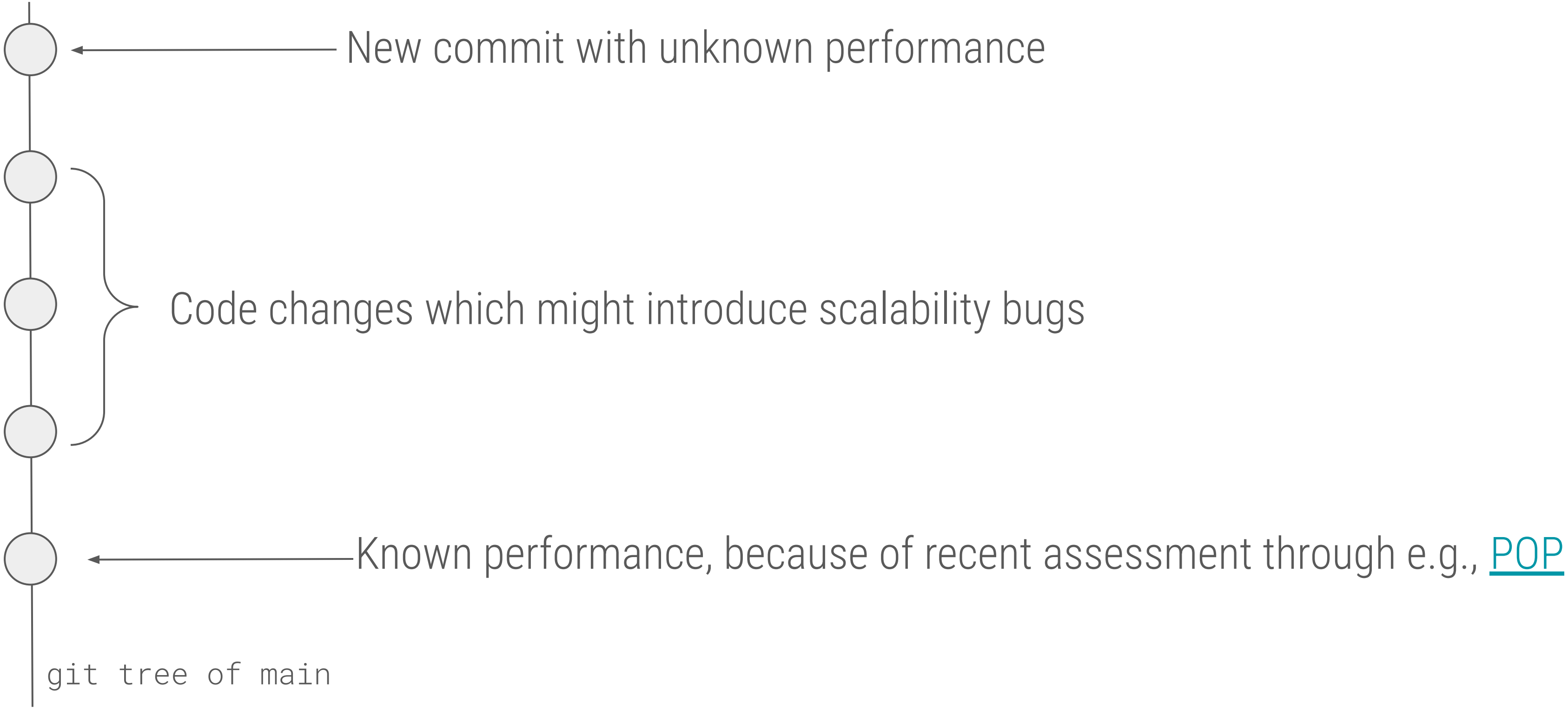
1 Barcelona Supercomputing Center
Centro Nacional de Supercomputación

dynamic load balancing

in collaboration with

2 MAX PLANCK INSTITUTE FOR PLASMA PHYSICS

Typical workflow for HPC software



New commit with unknown performance

Code changes which might introduce scalability bugs

Known performance, because of recent assessment through e.g., [POP](#)

`git tree of main`

Problems

Performance regression

- Do we introduce problematic code?
- When did we introduce problematic code?

Performance insight

- What is the bottleneck when scaling the application?
- Where should we try to improve to get better performance?

Available solutions

- Mostly based on **timings** or **figure of merit**
  Good for regression, bad for insight

- Require **additional infrastructure** (e.g. time-series database, web-server)
  Bad for teams with focus on developing scientific software

References

Anzt, et al.: Towards Continuous Benchmarking: An Automated Performance Evaluation Framework for High Performance Software, 10.1145/3324989.3325719

Alt, et al.: A continuous benchmarking infrastructure for high-performance computing applications, 10.1080/17445760.2024.2360190

TALP-Pages

- Based on TALP[1] we compute POP-like metrics that provide insight into bottlenecks
- No additional infrastructure required (only a DLB installation)
- Our development goal: Easy to use and integrate

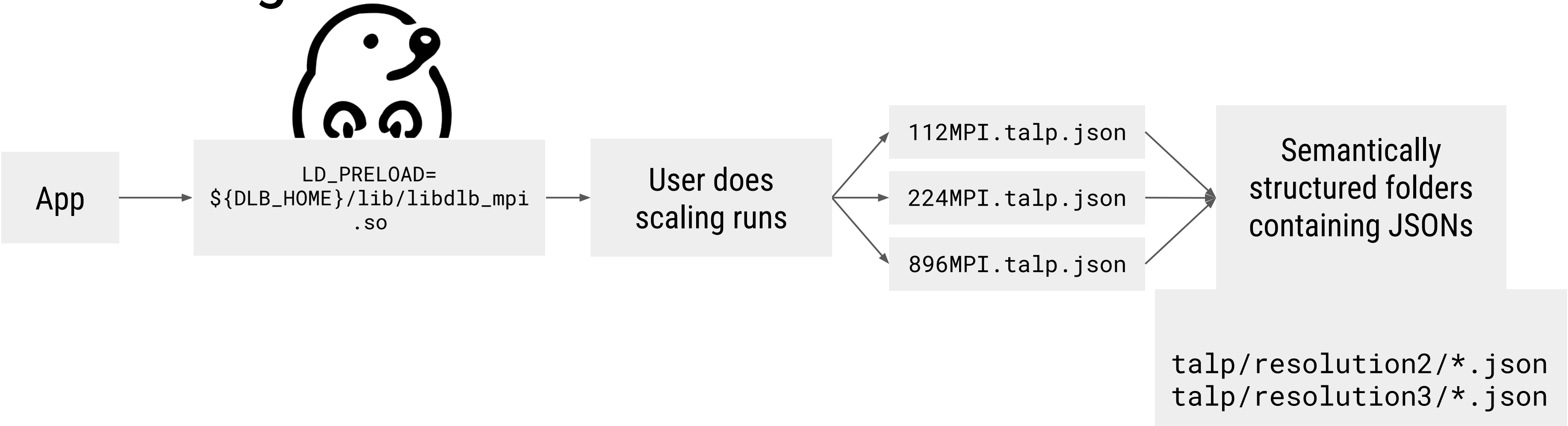
Installation:

```
pip install talp-pages
```

References

[1] Lopez, et. al. TALP: A Lightweight Tool to Unveil Parallel Efficiency of Large-scale Executions, https://doi.org/10.1145/3452412.3462753

TALP-Pages: Manual Workflow



App

```
LD_PRELOAD=
${DLB_HOME}/lib/libdlb_mpi
.so
```

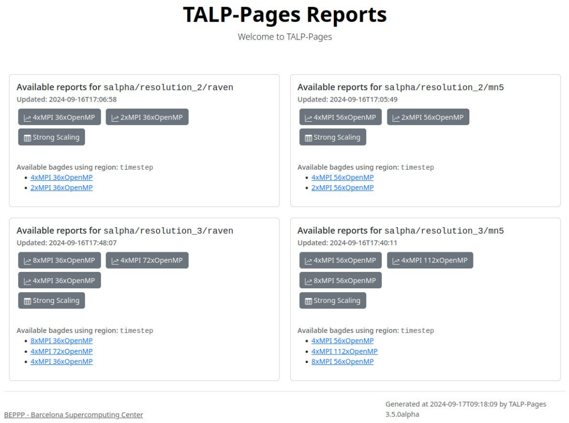
User does scaling runs

```
112MPI.talp.json
224MPI.talp.json
896MPI.talp.json
```

Semantically structured folders containing JSONs

```
talp/resolution2/*.json
talp/resolution3/*.json
```

```
talp pages -i talp -o public/talp
```



TALP-Pages Reports

Welcome to TALP-Pages

Available reports for `salpha/resolution_2/raven`

Updated: 2024-09-16T17:06:58

4xMPI 36xOpenMP | 2xMPI 36xOpenMP | Strong Scaling

Available bagdes using region: `timestep`

- 4xMPI 36xOpenMP
- 2xMPI 36xOpenMP

Available reports for `salpha/resolution_2/mn5`

Updated: 2024-09-16T17:05:49

4xMPI 56xOpenMP | 2xMPI 56xOpenMP | Strong Scaling

Available bagdes using region: `timestep`

- 4xMPI 56xOpenMP
- 2xMPI 56xOpenMP

Available reports for `salpha/resolution_3/raven`

Updated: 2024-09-16T17:48:07

8xMPI 36xOpenMP | 4xMPI 72xOpenMP | 4xMPI 36xOpenMP | Strong Scaling

Available bagdes using region: `timestep`

- 8xMPI 36xOpenMP
- 4xMPI 72xOpenMP
- 4xMPI 36xOpenMP

Available reports for `salpha/resolution_3/mn5`

Updated: 2024-09-16T17:40:11

4xMPI 56xOpenMP | 4xMPI 112xOpenMP | 8xMPI 56xOpenMP | Strong Scaling

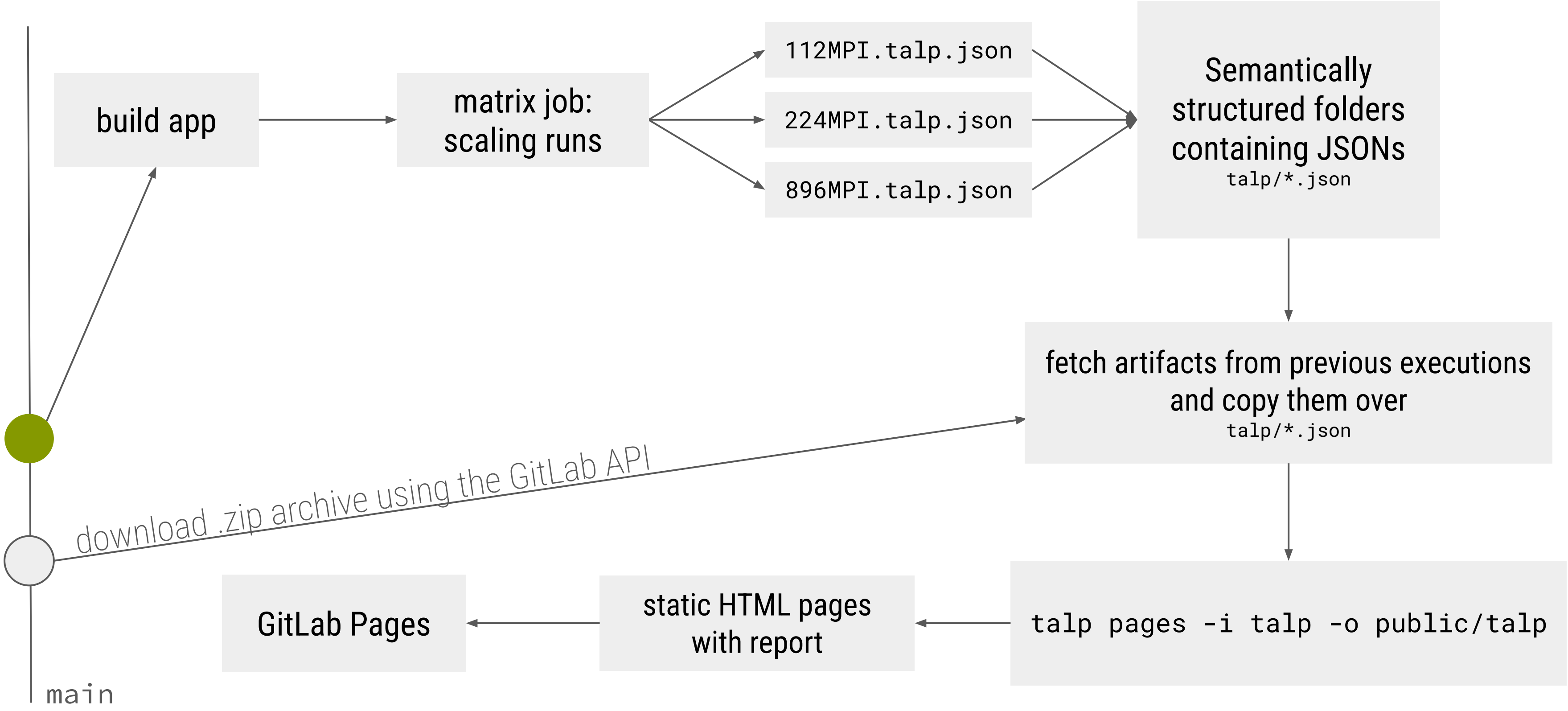
Available bagdes using region: `timestep`

- 4xMPI 56xOpenMP
- 4xMPI 112xOpenMP
- 8xMPI 56xOpenMP

BEPPP - Barcelona Supercomputing Center

Generated at 2024-09-17T09:18:09 by TALP-Pages 3.5.0alpha

TALP-Pages: GitLab workflow



build app

matrix job: scaling runs

```
112MPI.talp.json
224MPI.talp.json
896MPI.talp.json
```

Semantically structured folders containing JSONs `talp/*.json`

fetch artifacts from previous executions and copy them over `talp/*.json`

download .zip archive using the GitLab API

```
talp pages -i talp -o public/talp
```

static HTML pages with report

GitLab Pages

TALP-Pages

Two visualization modes available:

- **Scaling efficiency tables** with POP-like metrics (to gain **insight**)
- **Time evolution** plots of POP-like metrics (to track **regression**)

Integration into GENE-X

- Code changes
- GitLab Pipeline changes

Annotate regions (optional, but recommended)

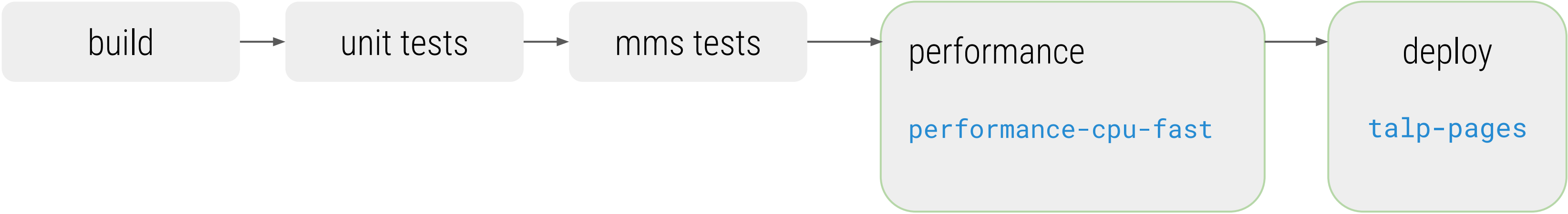
Add monitoring regions through TALP-API

```
#include "dlb_talp.h"

namespace annotation
{
   // Wrapper functions to annotate the start of a profiler region
   inline void start_region(const std::string& region_name) {
       auto handle = DLB_MonitoringRegionRegister(region_name.c_str());
       DLB_MonitoringRegionStart(handle);
   }

   // Wrapper functions to annotate the end of a profiler region
   inline void end_region(const std::string& region_name) {
       auto handle = DLB_MonitoringRegionRegister(region_name.c_str());
       DLB_MonitoringRegionStop(handle);
   }
}
```

GitLab Pipeline modifications



- Added a performance stage with a matrix job to run a fast test set
- Added a talp-pages job to the deploy stage

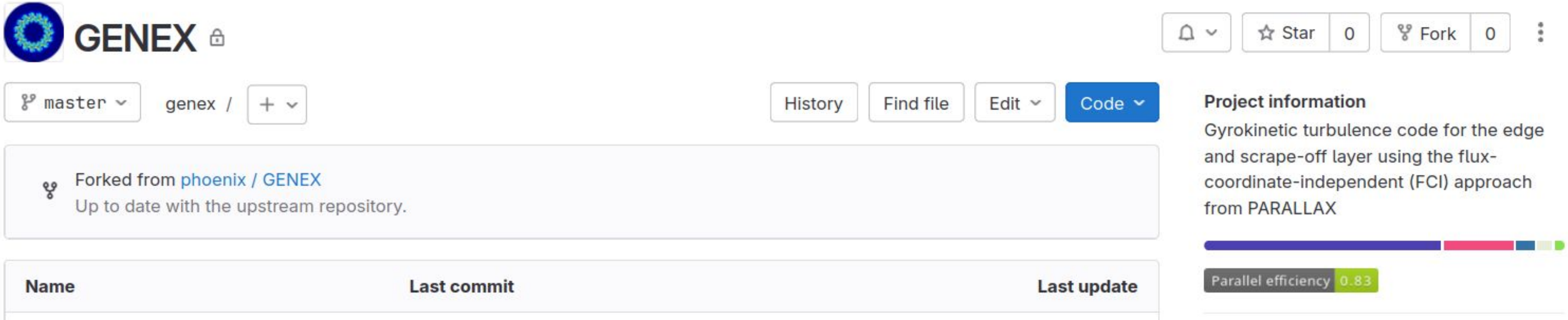
GitLab Pipeline modifications

```yaml
talp-pages:
 stage: deploy
 needs:
   - job: performance-cpu-fast
     optional: true
     artifacts: true
 script:
   - talp metadata -i talp --log-level=DEBUG # Add metadata like the git commit and git branch
   - talp_download_artifacts_gitlab ... --output-file downloaded/talp.zip # download previous artifacts
   ... # Unzip & copy over the downloaded artifacts into the ./talp folder
   - talp pages -i ./talp -o ./public/talp --regions initialize timestep --region-for-badge timestep
 artifacts:
   paths:
     - public/talp # For the pages job to deploy to GitLab Pages
     - talp # For the next talp_download_artifacts_gitlab to have the previous JSONs present
```

Results

- Scaling efficiency table
- Time evolution of performance metrics

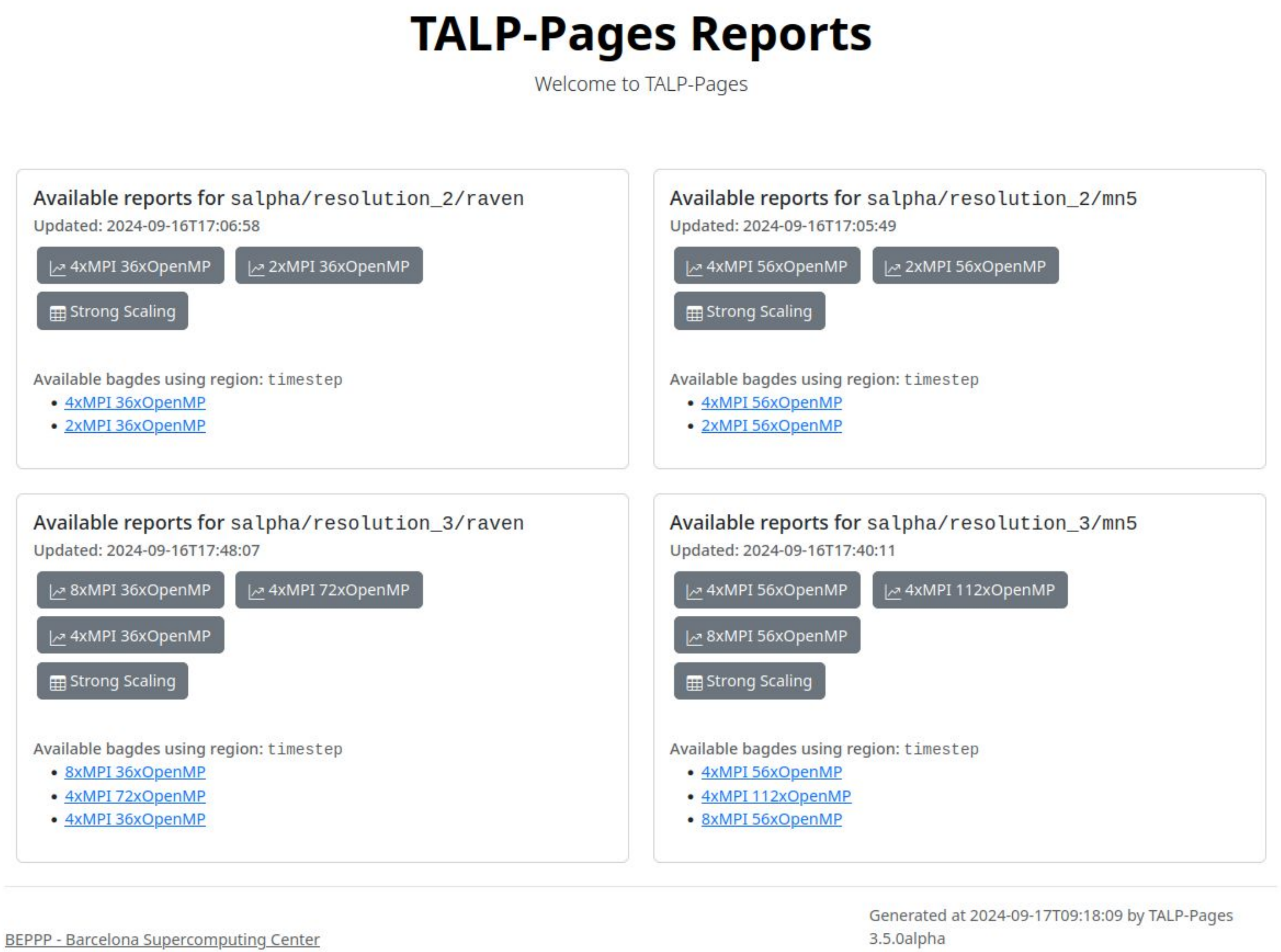
GENE-X repository view



Forked from phoenix / GENEX
Up to date with the upstream repository.

Project information
Gyrokinetic turbulence code for the edge and scrape-off layer using the flux-coordinate-independent (FCI) approach from PARALLAX

By clicking onto the efficiency badge the user get redirected to the GitLab Pages instance



TALP-Pages Reports

Welcome to TALP-Pages

Available reports for `salpha/resolution_2/raven`

Updated: 2024-09-16T17:06:58

Available bagdes using region: `timestep`

- 4xMPI 36xOpenMP
- 2xMPI 36xOpenMP

Available reports for `salpha/resolution_2/mn5`

Updated: 2024-09-16T17:05:49

Available bagdes using region: `timestep`

- 4xMPI 56xOpenMP
- 2xMPI 56xOpenMP

Available reports for `salpha/resolution_3/raven`

Updated: 2024-09-16T17:48:07

Available bagdes using region: `timestep`

- 8xMPI 36xOpenMP
- 4xMPI 72xOpenMP
- 4xMPI 36xOpenMP

Available reports for `salpha/resolution_3/mn5`

Updated: 2024-09-16T17:40:11

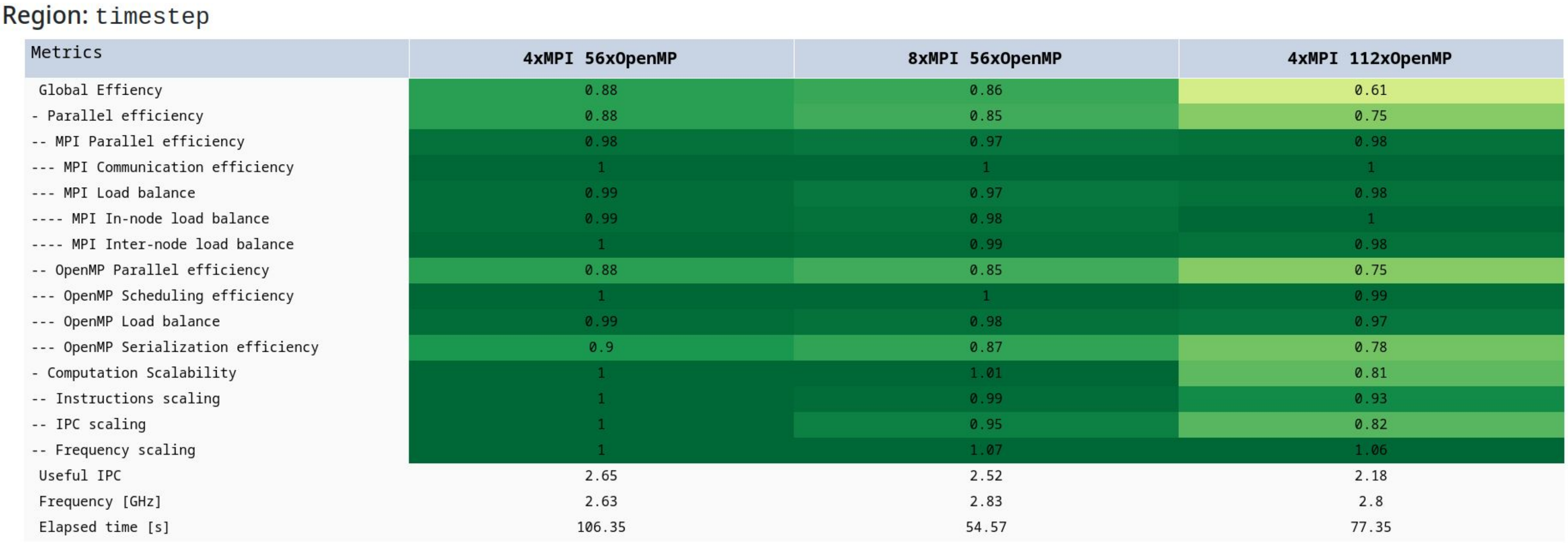
Available bagdes using region: `timestep`

- 4xMPI 56xOpenMP
- 4xMPI 112xOpenMP
- 8xMPI 56xOpenMP

BEPPP - Barcelona Supercomputing Center

Generated at 2024-09-17T09:18:09 by TALP-Pages 3.5.0alpha

Results: Scaling efficiency table of GENE-X

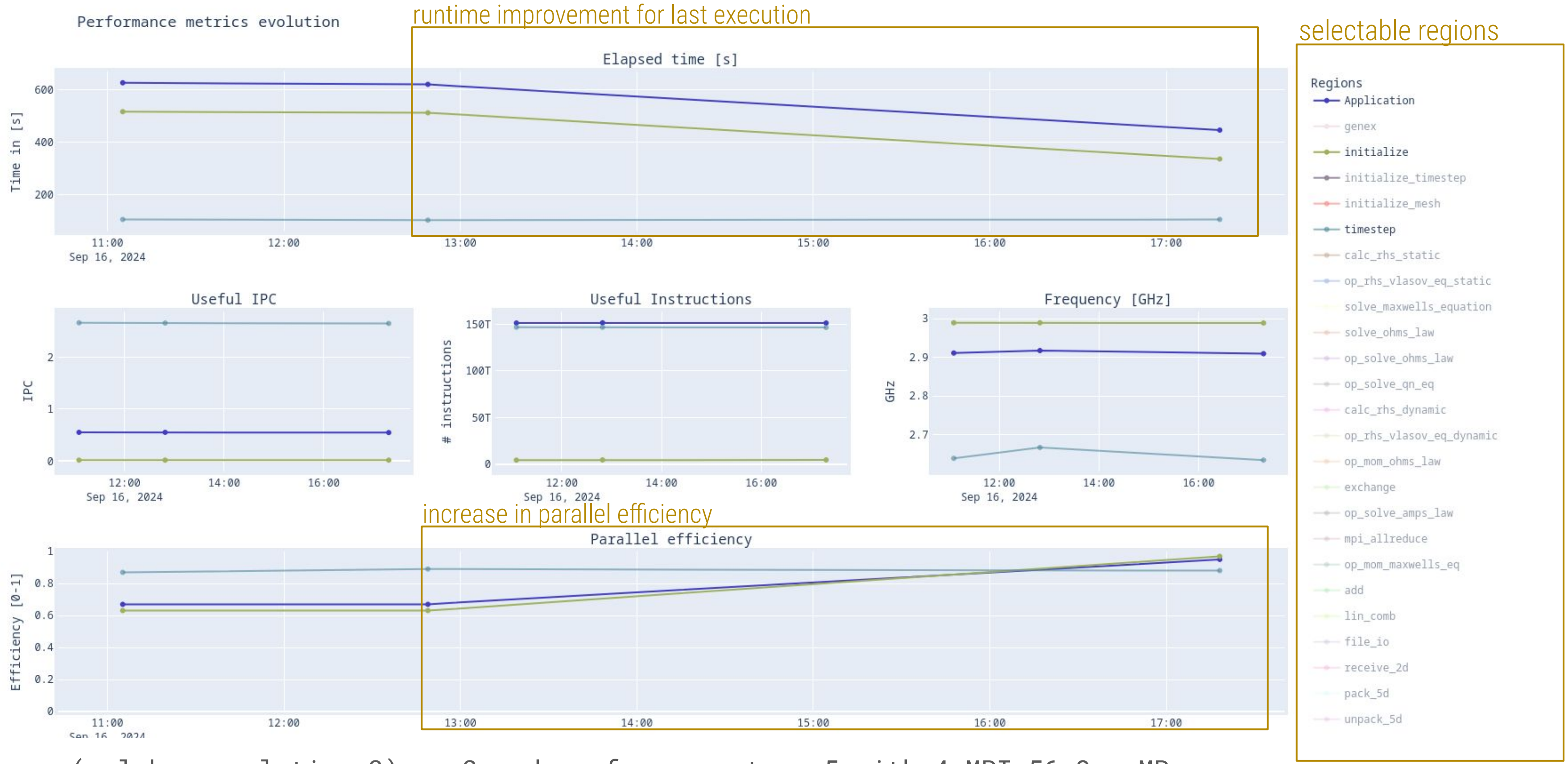


Region: timestep

Metrics	4xMPI 56xOpenMP	8xMPI 56xOpenMP	4xMPI 112xOpenMP
Global Effiency	0.88	0.86	0.61
- Parallel efficiency	0.88	0.85	0.75
-- MPI Parallel efficiency	0.98	0.97	0.98
--- MPI Communication efficiency	1	1	1
--- MPI Load balance	0.99	0.97	0.98
---- MPI In-node load balance	0.99	0.98	1
---- MPI Inter-node load balance	1	0.99	0.98
-- OpenMP Parallel efficiency	0.88	0.85	0.75
--- OpenMP Scheduling efficiency	1	1	0.99
--- OpenMP Load balance	0.99	0.98	0.97
--- OpenMP Serialization efficiency	0.9	0.87	0.78
- Computation Scalability	1	1.01	0.81
-- Instructions scaling	1	0.99	0.93
-- IPC scaling	1	0.95	0.82
-- Frequency scaling	1	1.07	1.06
Useful IPC	2.65	2.52	2.18
Frequency [GHz]	2.63	2.83	2.8
Elapsed time [s]	106.35	54.57	77.35

Case: (salpha, resolution_3) on marenostrum 5 general purpose partition

Results: Time evolution of GENE-X (with optimization)



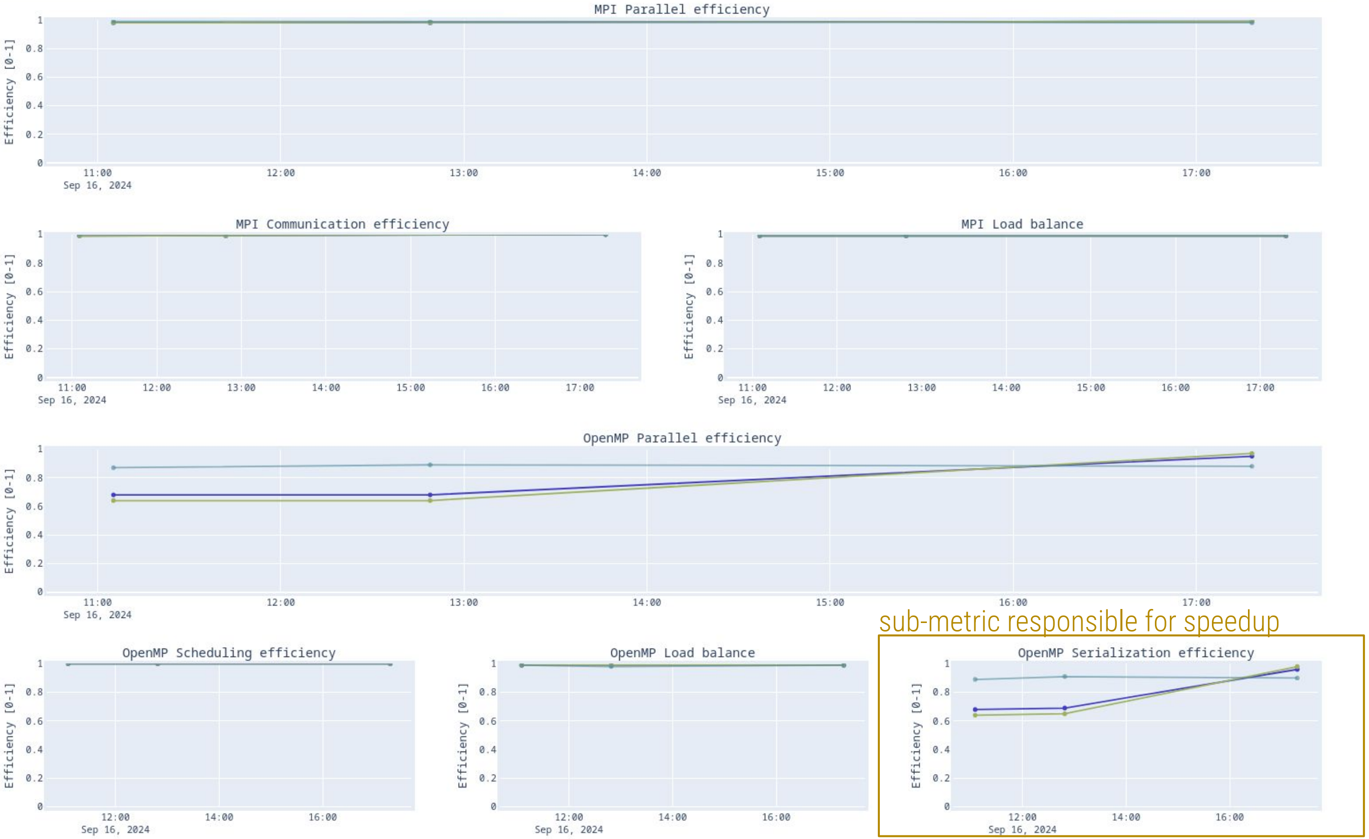
runtime improvement for last execution

selectable regions

increase in parallel efficiency

Case: (salpha,resolution_3) on 2 nodes of marenostrum 5 with 4xMPI 56xOpenMP

Results: Time evolution of GENE-X (with optimization)



sub-metric responsible for speedup

Case: (salpha,resolution_3) on 2 nodes of marenostrum 5 with 4xMPI 56xOpenMP

Summary

TALP-Pages: a framework for continuous performance monitoring

- **Minimal code changes** if existing annotation in place
- **Insight** and **regression** tightly coupled to code development workflow
- **No additional infrastructure** required

Outlook

- Release stable version
- GitHub integration
- Add more visualizations: e.g., On Pull-Request, empirical roofline model, textual feedback
- Add other output formats (PDF, SVG)

ChEESE

15. International Parallel Tools Workshop
19. September 2024 in Dresden

valentin.seitz@bsc.es