Friedrich-Alexander-Universität
Technische Fakultät

Continuous Benchmarking for a Massively Parallel Multi Physics Framework

Christoph Alt[1,3] **Harald Köstler**[1,2]

[1]Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany
[2]Erlangen National High Performance Computing Center (NHR@FAU), Erlangen, Germany
[3]Paderborn University, Paderborn, Germany

March 6, 2024 — deRSE24 Würzburg

Introduction

Performance is important in High-Performance Computing (HPC)

- Saves Money and Energy
- Better scientifc results (higher accuracy, more complex scenarios)

Challenges

- Performance need to be evaluated regulary
- Can be tedious with huge parameter space (architecutres, input parameter, . . .)

Goals

- Automatic and systematic performance evaluation on different HPC architectures with continuous integration tools
- Easy accessible Visualization for Developers/Users

Example Application: waLBerla[1]

- HPC C++ Framework for computational fluid dynamics
- Lattice Boltzmann Method
- Coupling to other methods:
  - Particle Dynamics
  - Phase fields
  - Free-Surface Flows

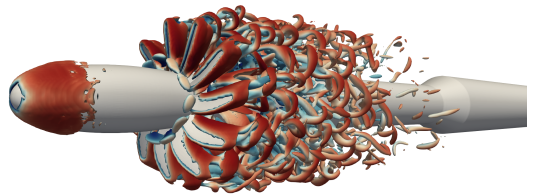


Figure: Flow around a rotor blades.

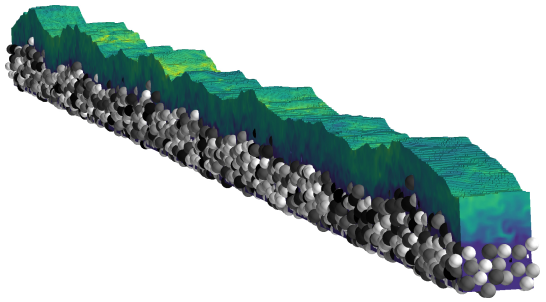
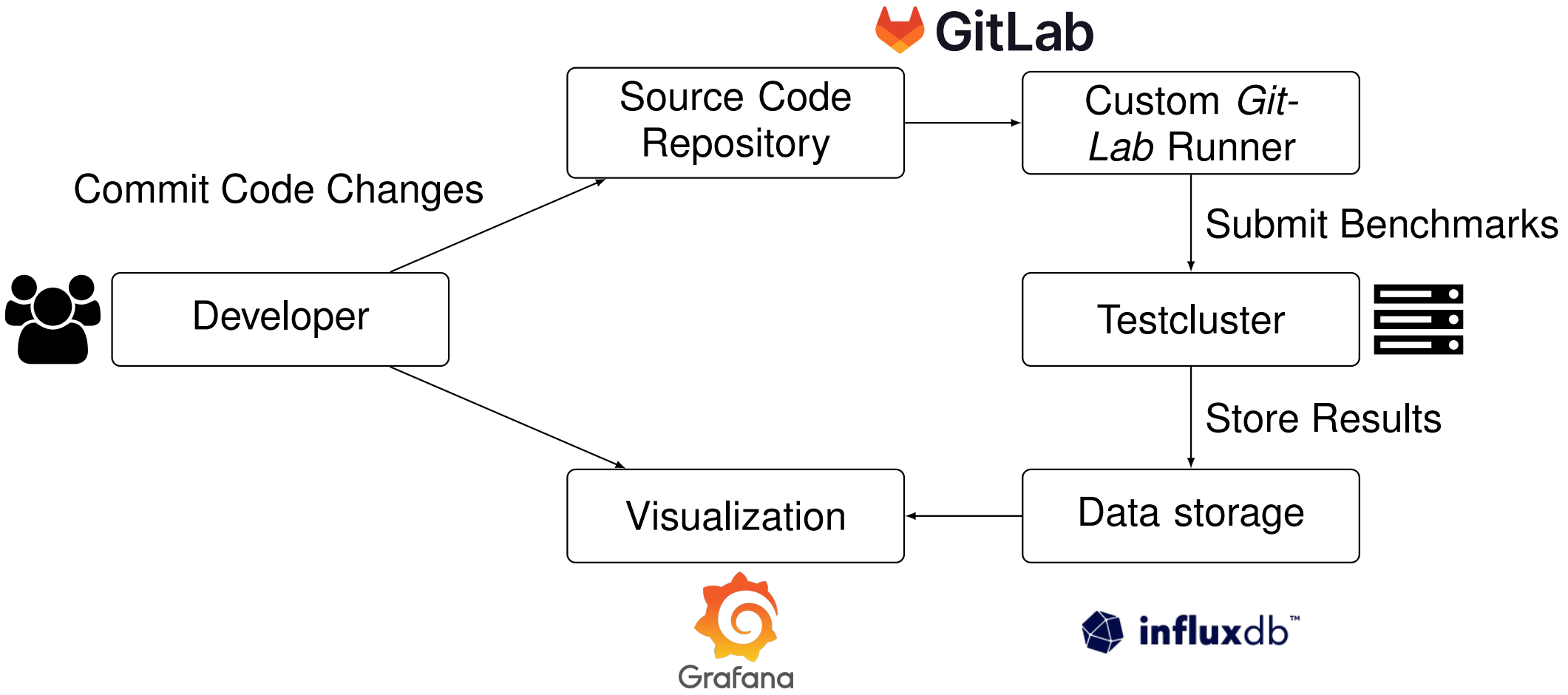


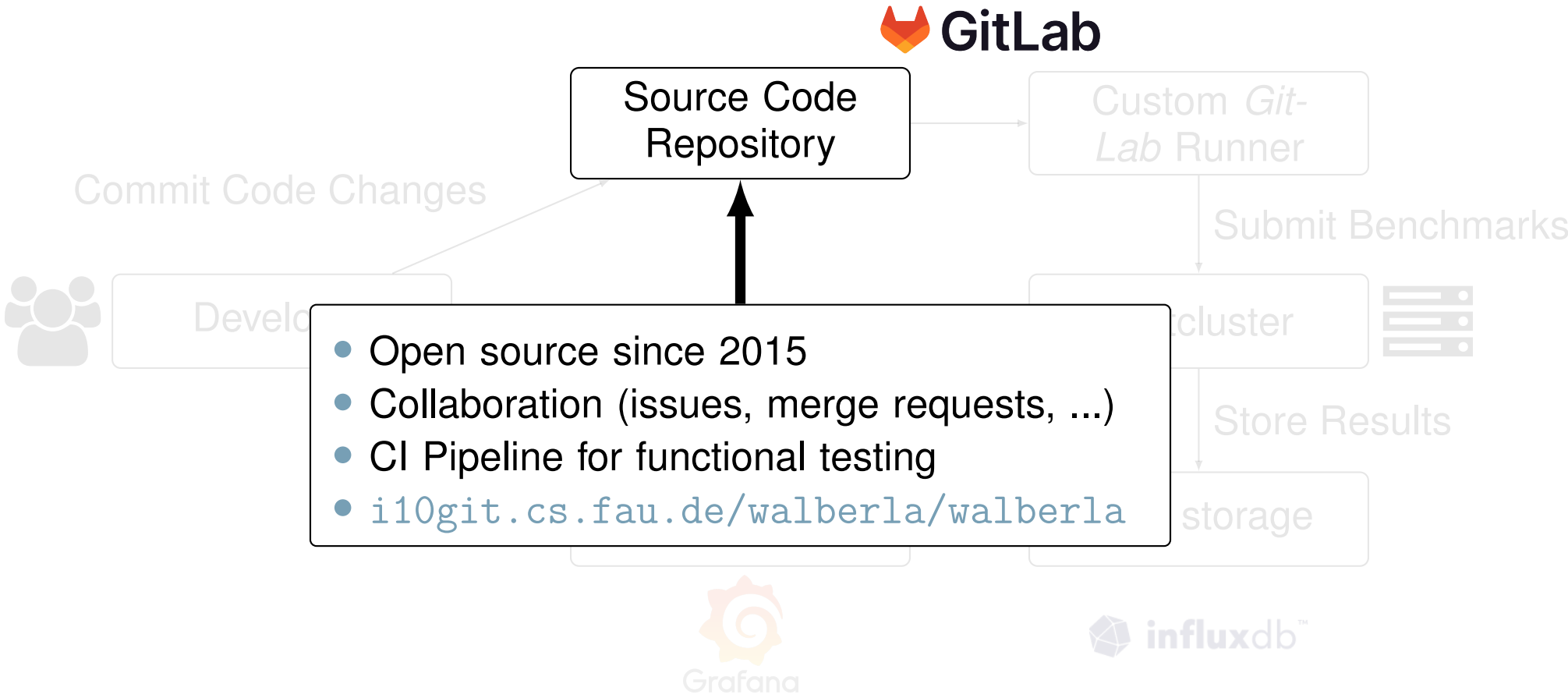
Figure: Upstream migrating dunes.

[1] M. Bauer et al. “waLBerla: A Block-Structured High-Performance Framework for Multiphysics Simulations”. In: *Computers & Mathematics with Applications*. Development and Application of Open-source Software for Problems with Numerical PDEs 81 (2021). DOI: 10.1016/j.camwa.2020.01.007.

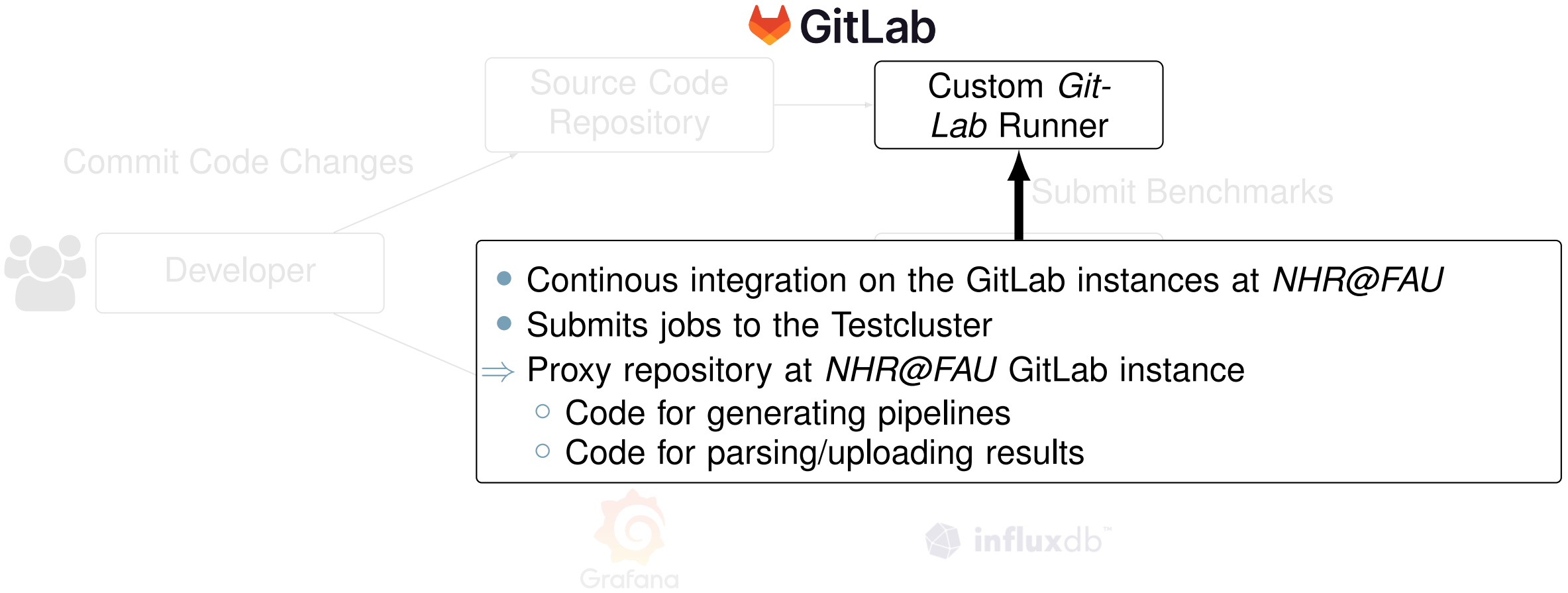
Overview


GitLab
Source Code Repository
Custom *Git-Lab* Runner
Commit Code Changes
Submit Benchmarks
Developer
Testcluster
Store Results
Visualization
Data storage
Grafana
influxdb™

waLBerla Repository



GitLab Cx Service at *NHR@FAU*[2]



[2] N. Team. *Continuous Integration / GitLab Cx*. https://doc.nhr.fau.de/sdt/ci/. [Accessed 19-02-2024].

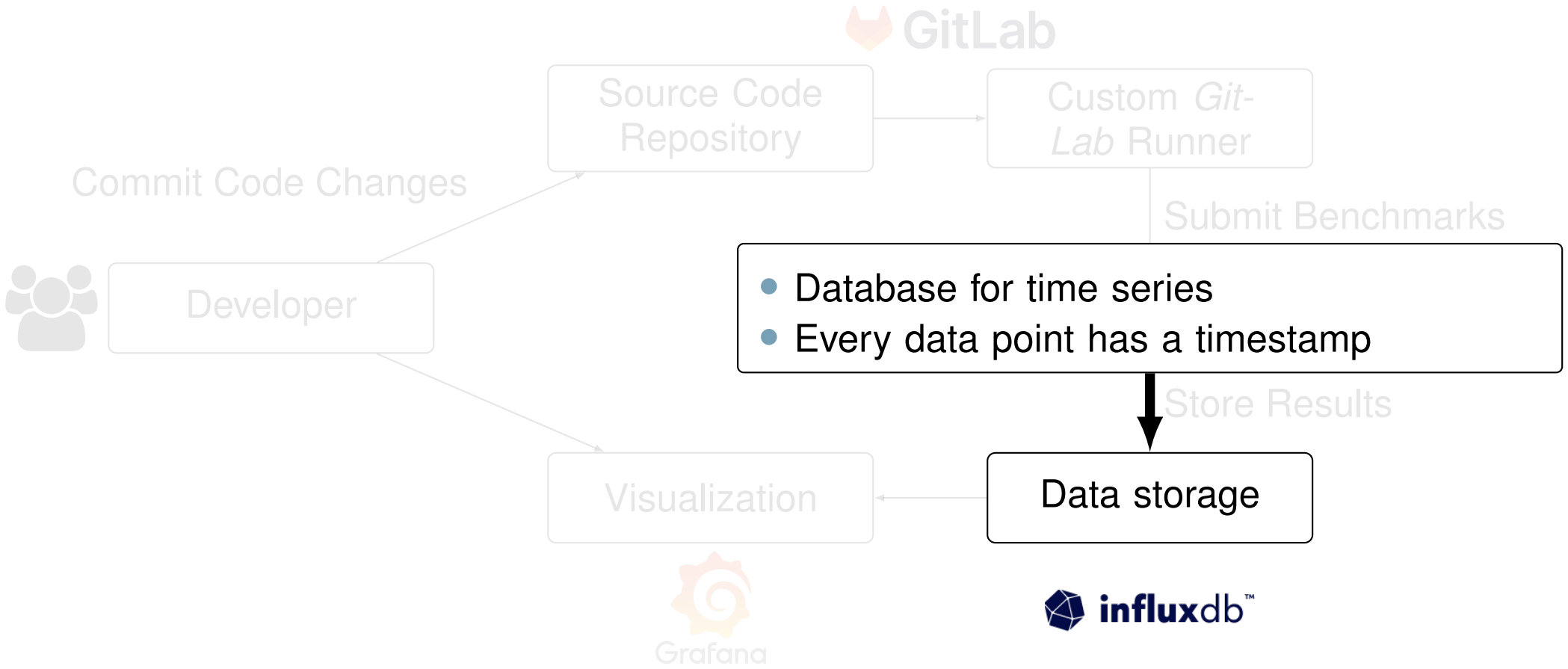
Target system - Testcluster

Table: Excerpt of the available compute nodes in the Testcluster at *NHR@FAU* [a]

Hostname	CPU	#Cores	Accelerators
`casclakesp2`	Dual Intel Xeon "Cascade Lake" Gold 6248 CPU	2x 20 cores	
`euryale`	Dual Intel Xeon "Broadwell" CPU E5-2620 v4	2x 8 cores	AMD RX 6900 XT
`genoa2`	Dual AMD EPYC 9354 "Genoa" CPU	2x 32 cores	Nvidia A40 Nvidia L40s
`icx36`	Dual Intel Xeon "Ice Lake" Platinum 8360Y CPU	2x 36 cores	
`medusa`	Dual Intel Xeon "Cascade Lake" Gold 6246 CPU	2x 12 cores	Nvidia Geforce RTX 2070 SUPER Nvidia Geforce RTX 2080 SUPER Nvidia Quadro RTX 5000 Nvidia Quadro RTX 6000
`naples1`	Dual AMD EPYC 7451 "Naples" CPU	2x 24 cores	
`rome1`	Single AMD EPYC 7452 "Rome" CPU	1x 32 cores	
`skylakesp2`	Intel Xeon "Skylake" Gold 6148 CPU	2x 20 cores	

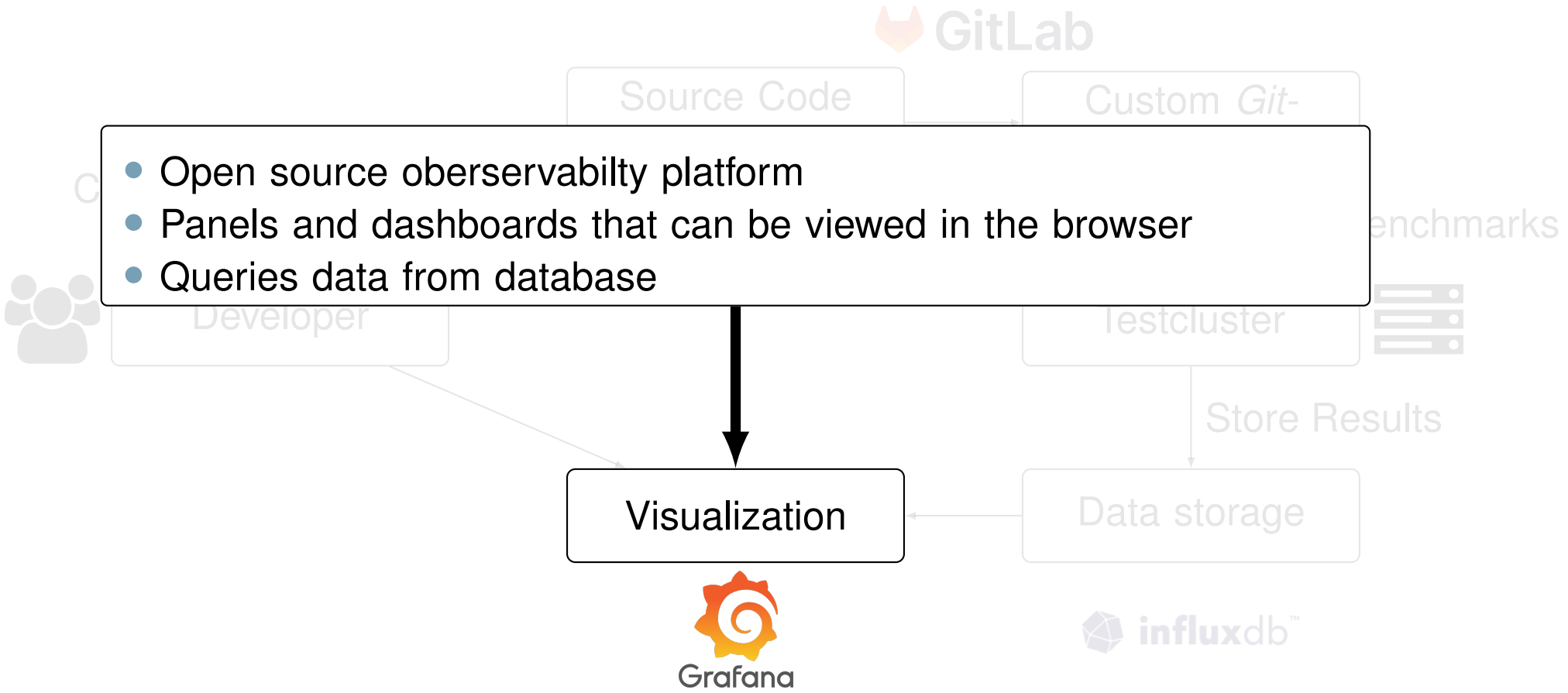
[a] https://doc.nhr.fau.de/clusters/testcluster/

InfluxDB[3]



[3] *InfluxDB*. https://www.influxdata.com/. [Accessed 15-02-2024].

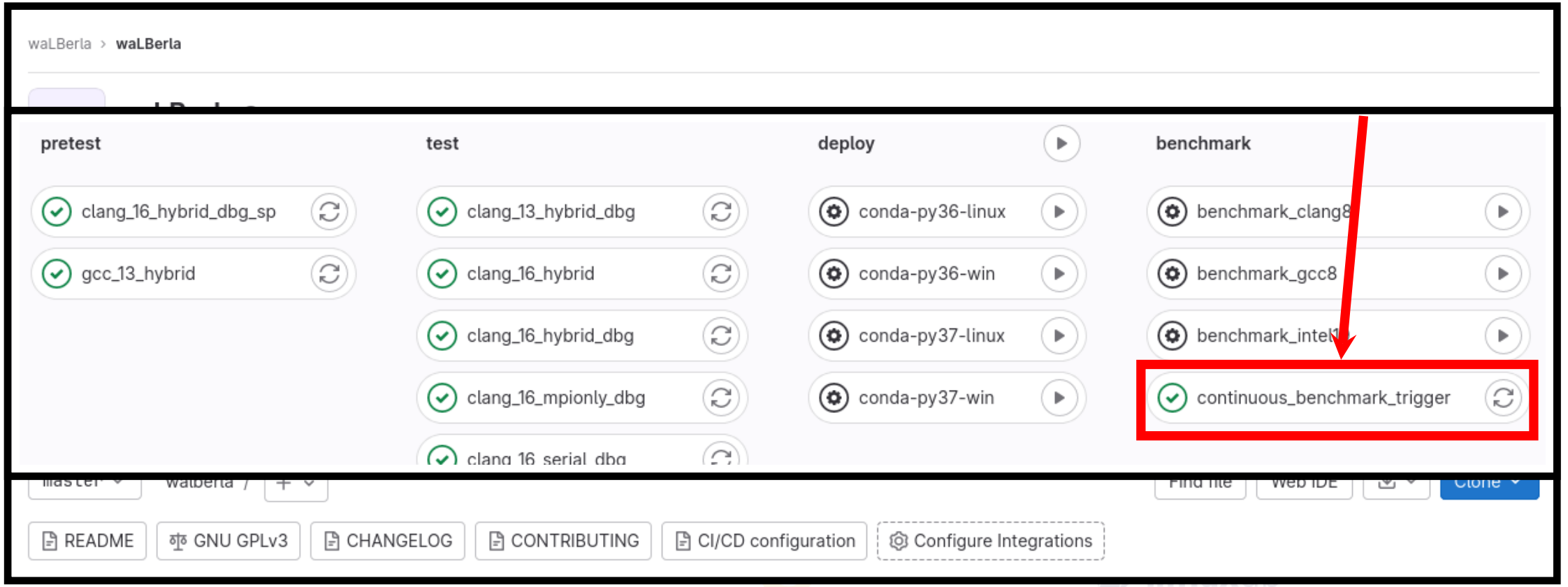
Grafana[4]



- Open source oberservabilty platform
- Panels and dashboards that can be viewed in the browser
- Queries data from database

[4] *Grafana*. https://grafana.com/grafana/. [Accessed 15-02-2024].

How is the Pipeline triggered?



How is the Pipeline triggered?

```
1  continuous_benchmark_trigger:
2    stage: benchmark
3    image: curlimages/curl
4    tags:
5      - docker
6    s
7
8
9
10
11
12
13
14   r
15
16
17
18       when: manual
```

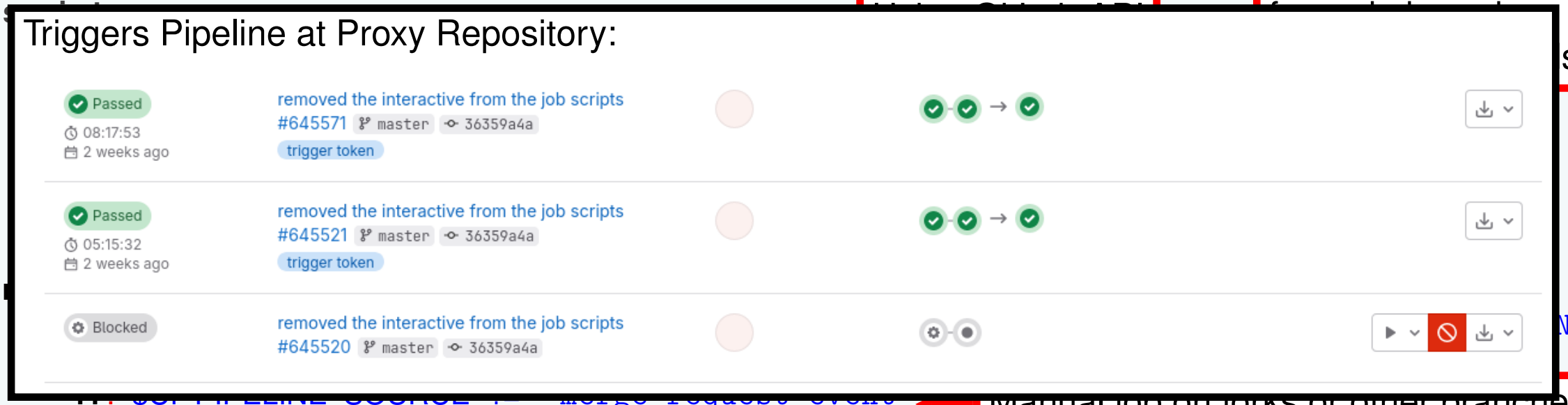
Exectue automatically

s

NCH'

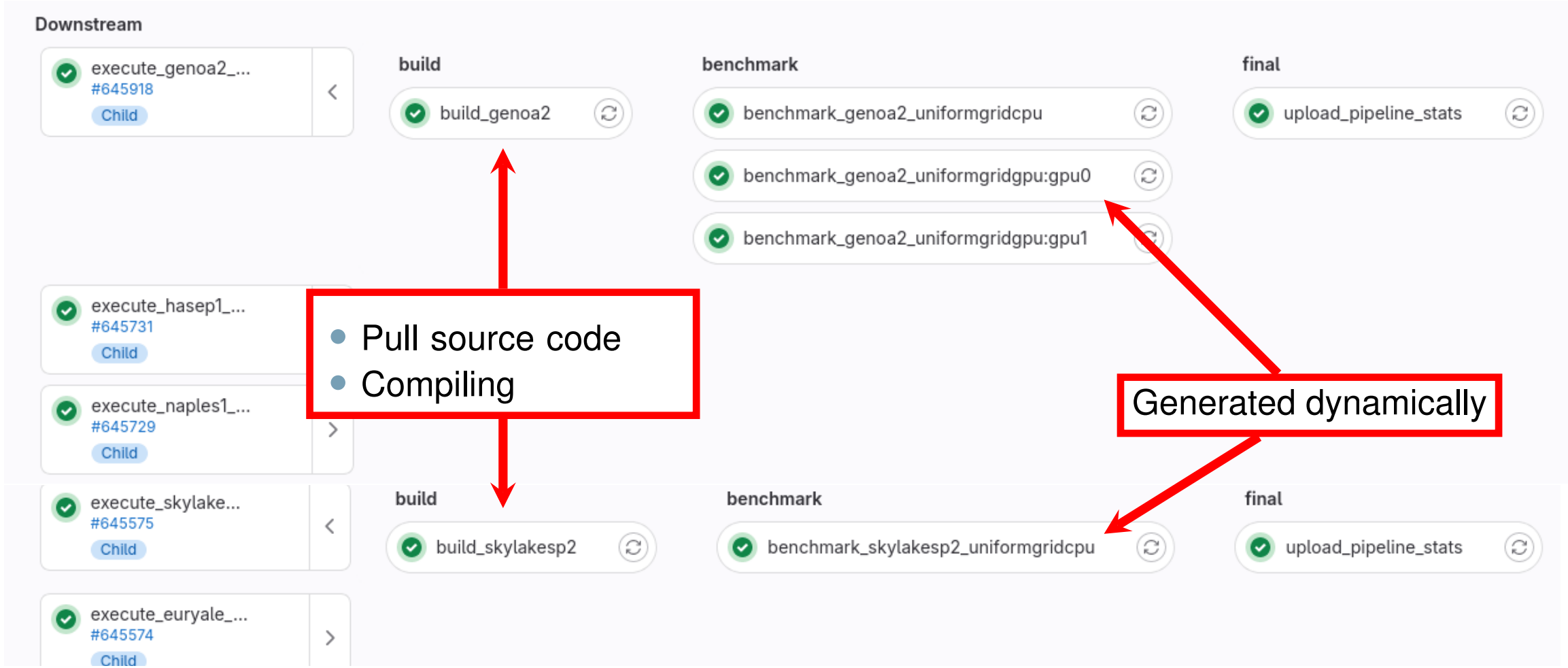
Manual job on forks or other branches

Triggers Pipeline at Proxy Repository:

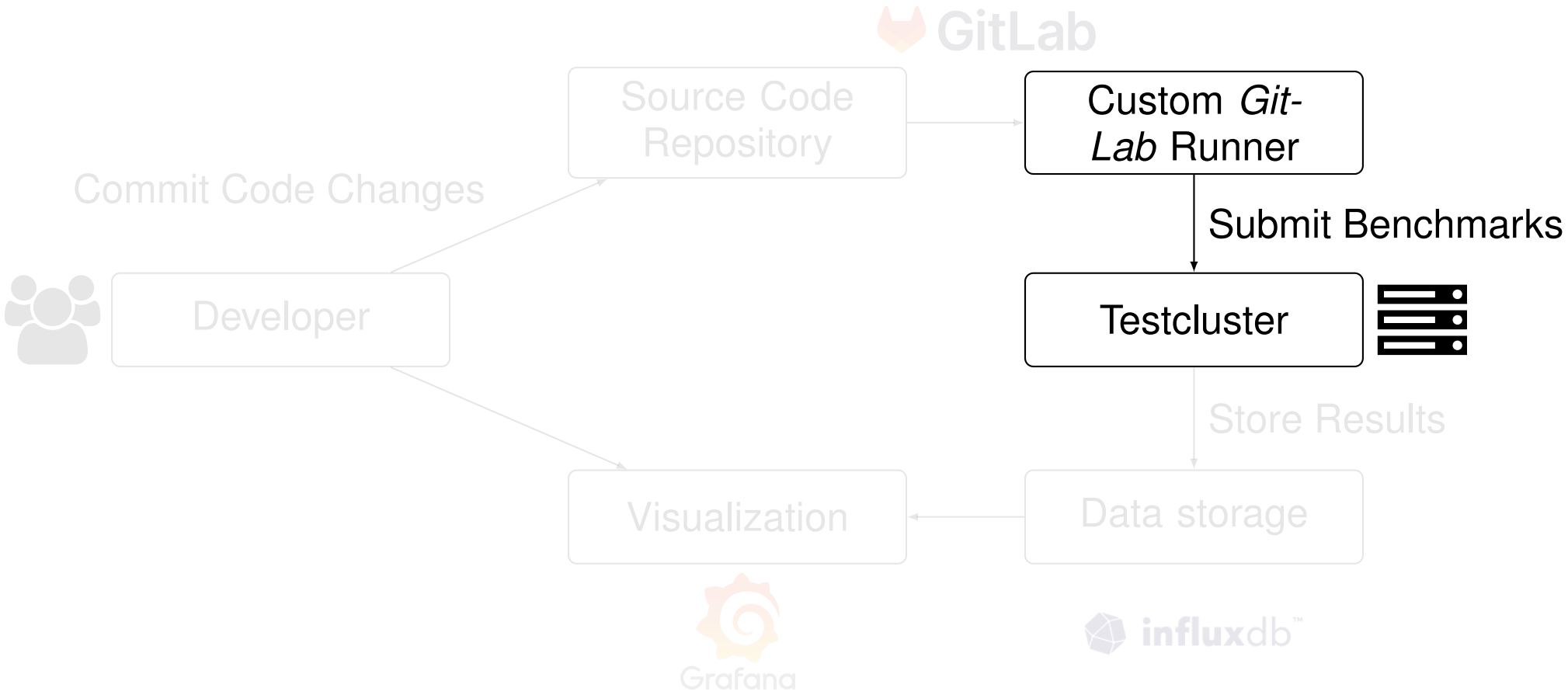


Dynamic Child Pipelines I

Dynamic Child Pipelines II



Benchmark Submissison



Benchmark Submission & Execution

GitLab CI job specification for job that submits a job[5]

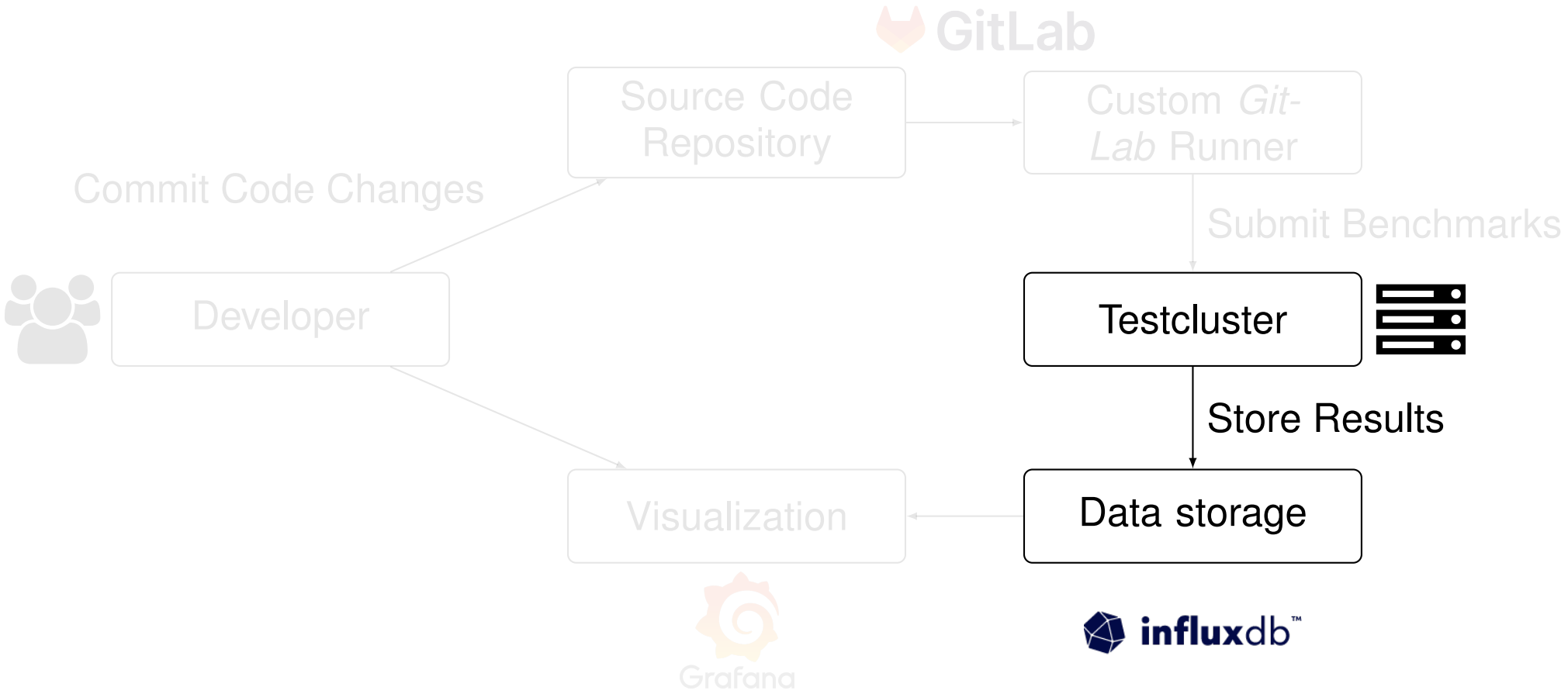
```
2484 Runing likwid-perfctr -g MEM_DP -o likwid_UniformGridCPU_d3q27_pull_w-mrt_16_MEM_DP.jso
     n likwid-mpirun -n 72 /scratch/walberla_ci/build/apps/benchmarks/UniformGridCPU/Uniform
     GridCPU_d3q27_pull_w-mrt /scratch/walberla_ci/build/apps/benchmarks/UniformGridCPU/simu
     lation_setup/benchmark_configs.py
2485 [ 0][INFO    ]------(0.000 sec) Total num of benchmark scenarios, saving result to cpu_
     benchmark.sqlite3
2486 [ 0][INFO    ]------(0.000 sec) Running single Node benchmarks
2487 [ 0][INFO    ]------(0.000 sec)
2488 [ 0][INFO    ]------(0.005 sec) Scenario:
2489 [ 0]                           {'DomainSetup': {'blocks': (6, 6, 2),
```

Using Likwid/Nvidia Nsight Compute for profiling

```
15       --job-name "${CI_JOB_NAME}" \
16       ${JOB_SCRIPT_FILE})
17     - cat ${CI_JOB_NAME}.o${job_id}.log
```

[5] T. Gruber, J. Eitzinger, G. Hager, and G. Wellein. *LIKWID*. Zenodo. 2022. DOI: 10.5281/ZENODO.7432487.

Data Upload


GitLab
Source Code Repository
Custom *Git-Lab* Runner
Commit Code Changes
Submit Benchmarks
Developer
Testcluster
Store Results
Visualization
Data storage
Grafana
influxdb™

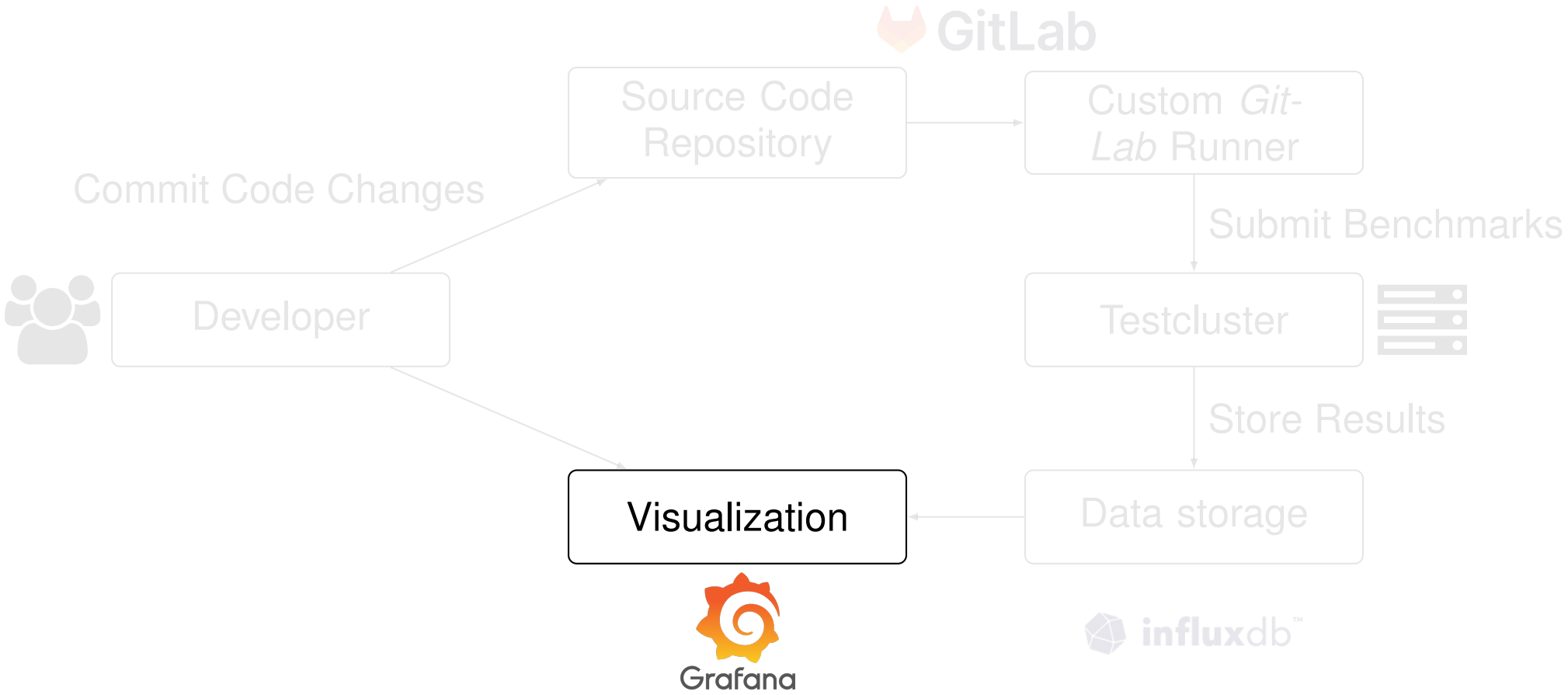
Data Upload to InfluxDB[6]

A single data point

```
{'measurement': 'UniformGridCPU',
 'time': 1707867380,
 'fields': {'AVX DP [MFLOP/s] STAT Sum': 53624.704,
            'Clock [MHz] STAT Avg': 2724.334,
            'DP [MFLOP/s] STAT Sum': 53628.097,
            'Energy [J] STAT Sum': 3802.983,
            'Memory bandwidth [MBytes/s] STAT Sum': 3225.023,
            'mlupsPerProcess': 23.08989552374622 },
 'tags': {'blocks_0': 8,
          'build_machine': 'optane1',
          'collisionSetup': 'srt',
          'mpi_num_processes': 128,
          'stencil': 'd3q27',
          'streamingPattern': 'pull',
          'walberla_version': '2b47f2a7a23bab5a4045710e942433e923ebf'}
}
```

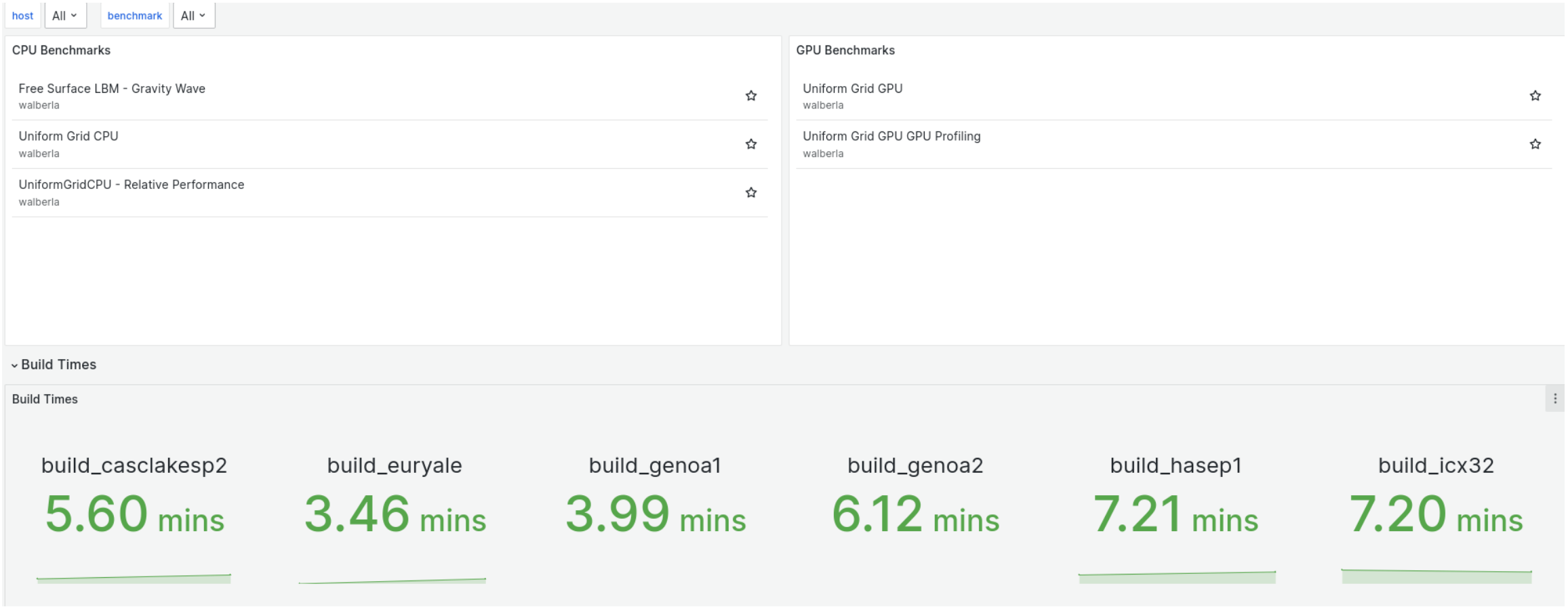
[6] *InfluxDB*. https://www.influxdata.com/. [Accessed 15-02-2024].

Visalization



Visualization with *Grafana*[7]

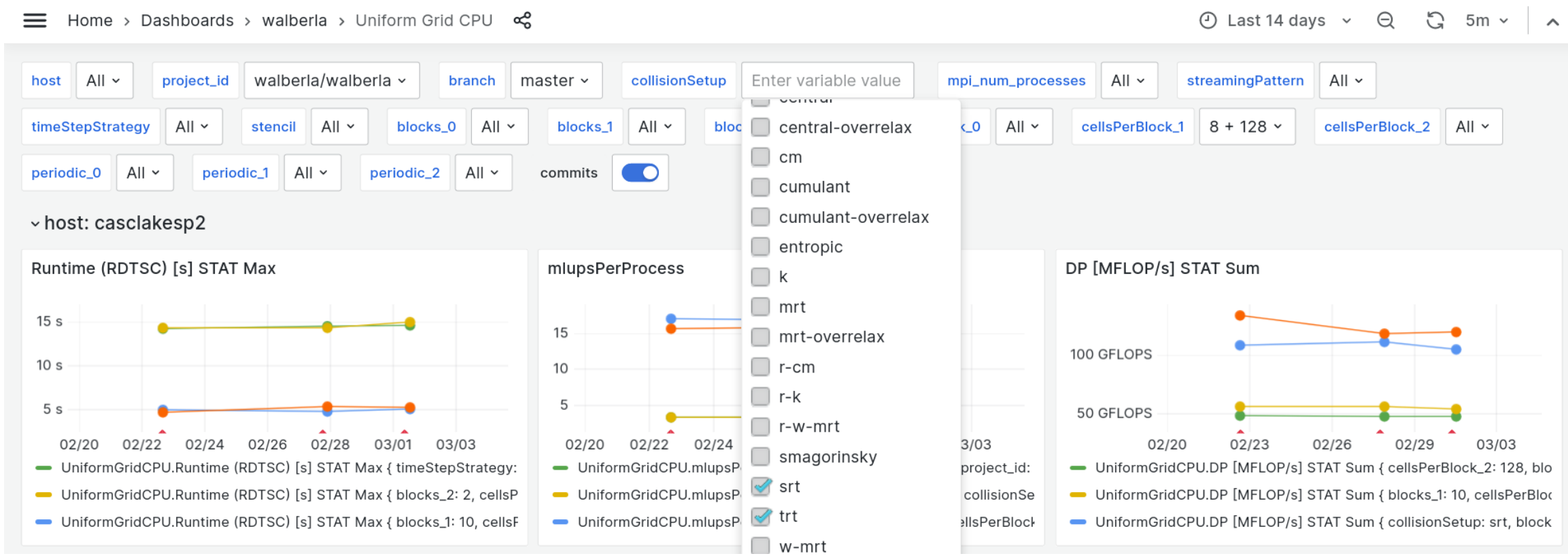
Overview Dashboard



[7] *Grafana*. https://grafana.com/grafana/. [Accessed 15-02-2024].

Visualization with *Grafana*[8]

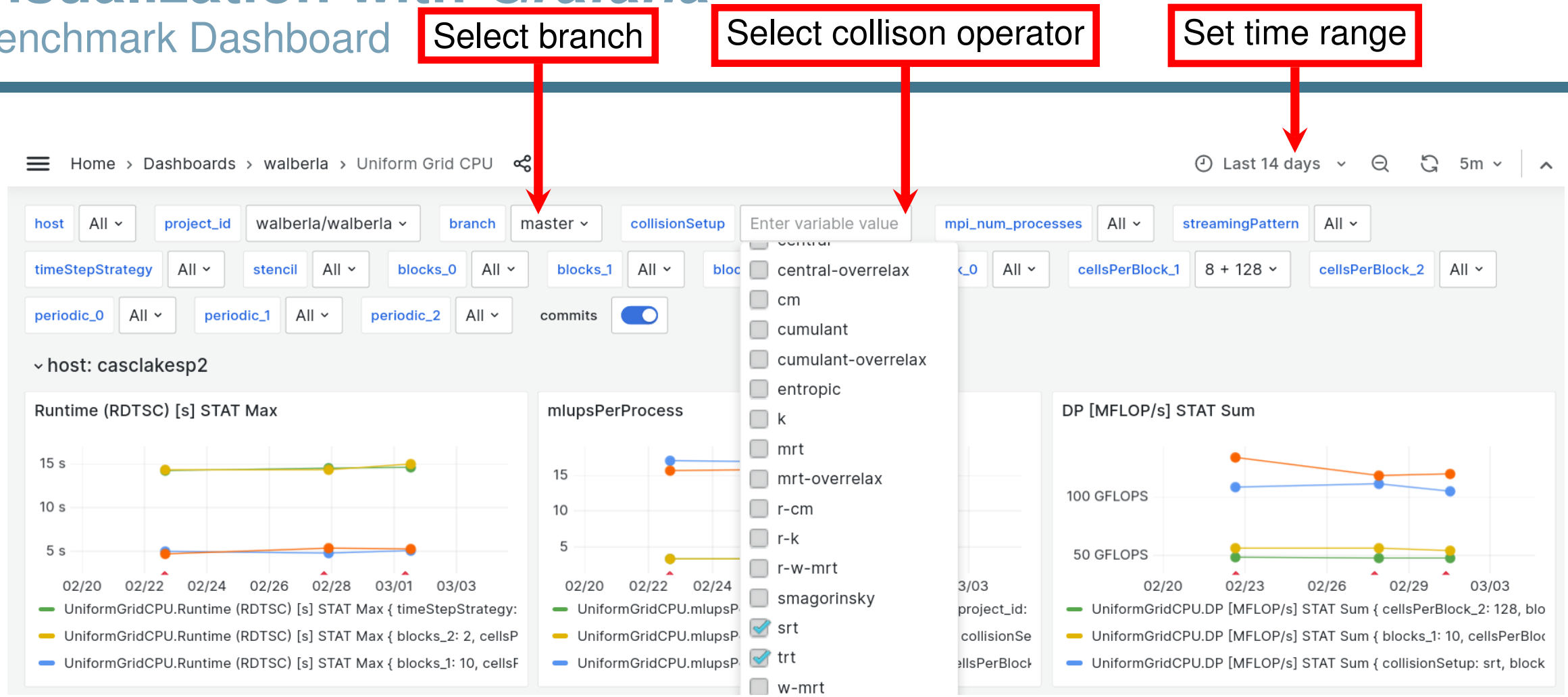
Benchmark Dashboard



[8] *Grafana*. https://grafana.com/grafana/. [Accessed 15-02-2024].

Visualization with *Grafana*[9]

Benchmark Dashboard

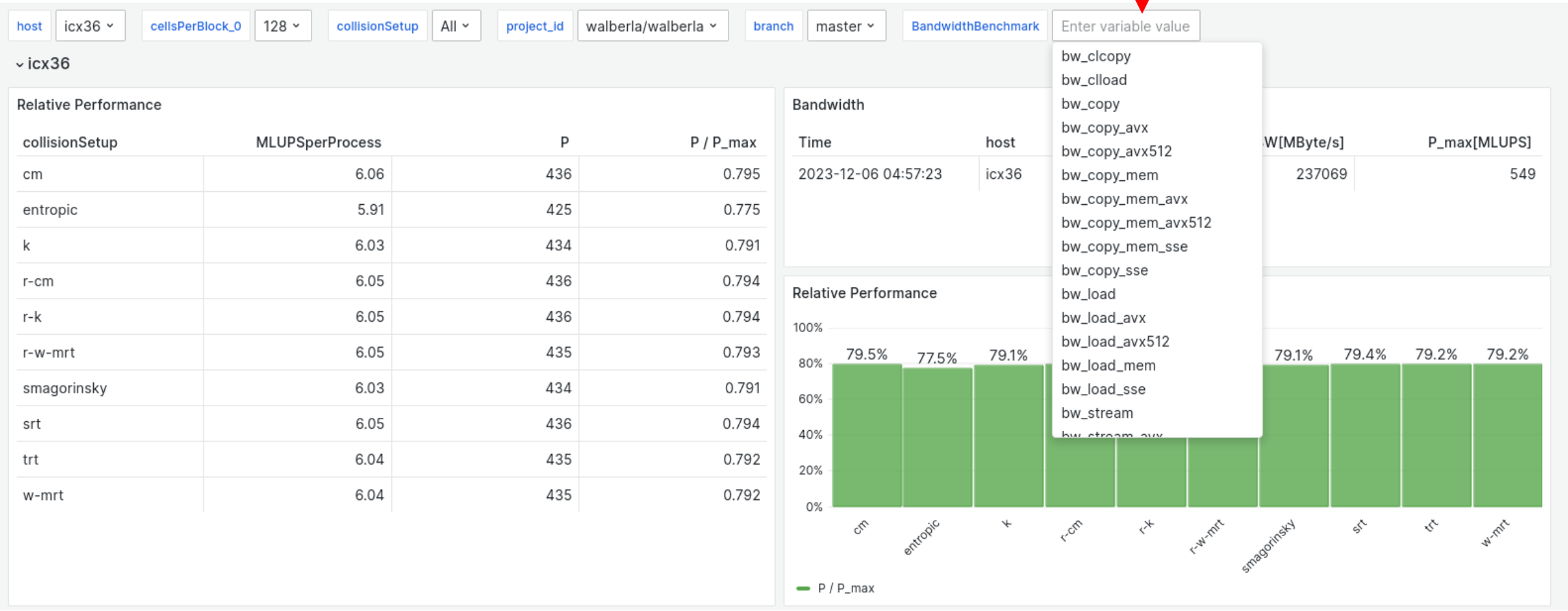


[9] *Grafana*. https://grafana.com/grafana/. [Accessed 15-02-2024].

Visualization with *Grafana*[10]

Roofline Analysis

Bandwidths measured with `likwid-bench`



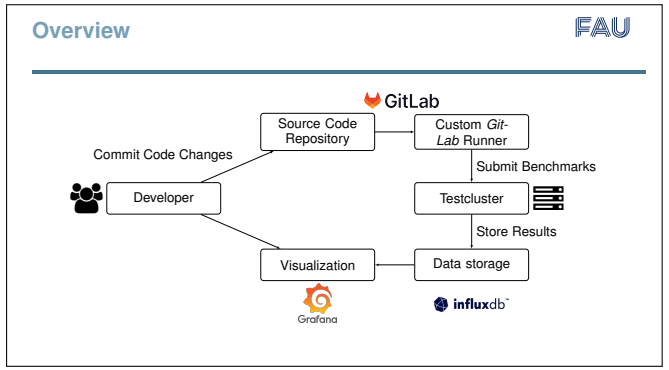
[10] *Grafana*. https://grafana.com/grafana/. [Accessed 15-02-2024].

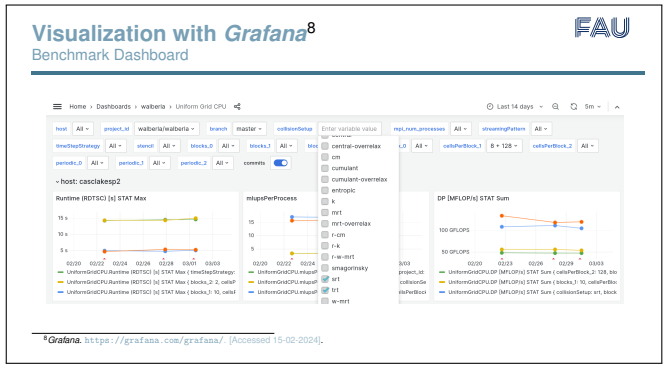
Conclusion and Outlook

- Conclusion:
  - Continuous benchmarking pipeline that automatically tests different architectures
  - Interactive visalization with Grafana

- Future improvements:
  - Integrate new benchmarks
  - Integrate new hardware architecutres (APUs, ...)
  - Multi-node setups (automate scaling runs)

Questions? Remarks?





Thank you for the Attention!

Grafana Dashboards

Collection of utility scripts

Proxy Repository

Paper Preprint

Friedrich-Alexander-Universität
Technische Fakultät

References I

- Bauer, M., S. Eibl, C. Godenschwager, N. Kohl, M. Kuron, C. Rettinger, F. Schornbaum, C. Schwarzmeier, D. Thönnes, H. Köstler, and U. Rüde. “waLBerla: A Block-Structured High-Performance Framework for Multiphysics Simulations”. In: *Computers & Mathematics with Applications*. Development and Application of Open-source Software for Problems with Numerical PDEs 81 (2021). DOI: 10.1016/j.camwa.2020.01.007.
- *Grafana*. https://grafana.com/grafana/. [Accessed 15-02-2024].
- Gruber, T., J. Eitzinger, G. Hager, and G. Wellein. *LIKWID*. Zenodo. 2022. DOI: 10.5281/ZENODO.7432487.
- *InfluxDB*. https://www.influxdata.com/. [Accessed 15-02-2024].
- Team, N. *Continuous Integration / GitLab Cx*. https://doc.nhr.fau.de/sdt/ci/. [Accessed 19-02-2024].