

Continuous Performance Monitoring of the Grid Library on a Supercomputer

Simon Bürger, Antonin Portelli

September 19th, 2024



THE UNIVERSITY
of EDINBURGH

Our code base

- *Grid*: data parallel C++ container classes mapping efficiently to SIMD architectures including GPUs
<https://github.com/paboyle/Grid>
- *Hadrons*: Grid-based workflow management system for lattice field theory simulations
<https://github.com/aportelli/Hadrons>

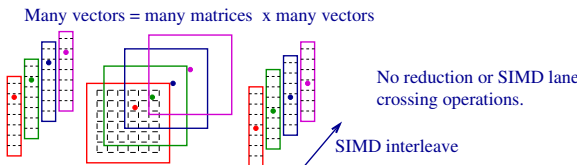


Figure: Grid architecture, from [hep-lat/1512.03487]

Tursa Computing cluster

- part of the national *STFC DiRAC HPC facility*
- 178 compute nodes, each equipped with
 - two *AMD EPYC* processors
 - four *NVIDIA Ampere A100* accelerators
- interconnect
 - NVLink on each node, running at 600 GB/s
 - Four HDR-200 infiniband interfaces per node
- jobs are run via Slurm scheduler from dedicated login nodes
- find out more at <https://dirac.ac.uk>

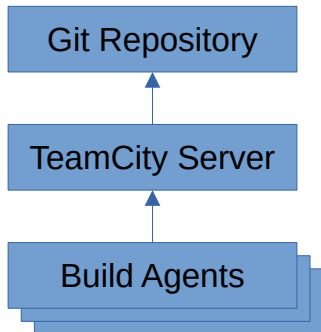


DiRAC

Goals

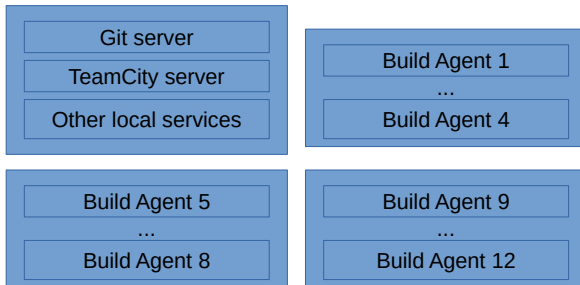
- **Continuous integration (“CI”)**: Notice breaking changes as soon as possible, avoid infamous “works on my machine”
- **Continuous deployment (“CD”)**: Improve reproducibility and simplify user experience
- **Performance monitoring**: Detect performance degradations caused by any part of the system

Jetbrains TeamCity Architecture



- Scalable to arbitrary number of build servers
- Communication via https

Dedicated CI/CD hardware



- Four AMD EPYC servers, same environment as login nodes
- Head server: docker containers, externally visible
- Agents: run on bare metal, not externally visible

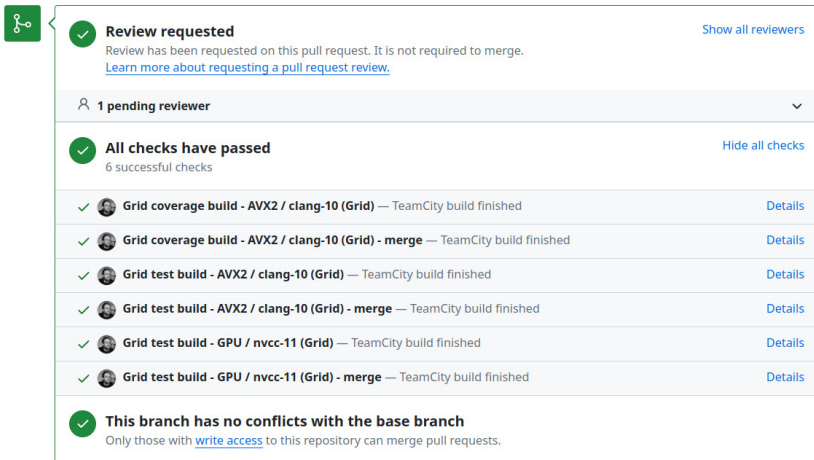
Triggers for the CI system


- Each commit, including pending pull requests
→ build Grid and Hadrons and run unittests
- Once per day:
→ deploy new production binaries if there were any changes
→ run benchmarks, using latest production binaries
- Benchmarking run regardless of code changes, thus monitoring the runtime environment as well.

Integration into the HPC cluster Tursa



- Problem: Build servers do not have GPUs
- Solution: GPU-based unittests and benchmarking jobs are submitted via SLURM to the computing cluster
- Finished jobs report back to TeamCity via a REST API
- Build agents are not waiting for SLURM queue

What does it look like? On a GitHub pull request:















 **Review requested** [Show all reviewers](#)

Review has been requested on this pull request. It is not required to merge.
[Learn more about requesting a pull request review.](#)

 **1 pending reviewer** 

All checks have passed [Hide all checks](#)
6 successful checks

-   **Grid coverage build - AVX2 / clang-10 (Grid)** — TeamCity build finished [Details](#)
-   **Grid coverage build - AVX2 / clang-10 (Grid) - merge** — TeamCity build finished [Details](#)
-   **Grid test build - AVX2 / clang-10 (Grid)** — TeamCity build finished [Details](#)
-   **Grid test build - AVX2 / clang-10 (Grid) - merge** — TeamCity build finished [Details](#)
-   **Grid test build - GPU / nvcc-11 (Grid)** — TeamCity build finished [Details](#)
-   **Grid test build - GPU / nvcc-11 (Grid) - merge** — TeamCity build finished [Details](#)

This branch has no conflicts with the base branch
Only those with [write access](#) to this repository can merge pull requests.



Projects

Changes

Agents 12

Queue 0



Log in...

Search... Q <<

All Projects

Favorite Builds

PROJECTS

v Lattice

v Benchmarks

Run All

> TursaCpu

v TursaGpu

v Benchmark_Grid

> 4 x NVIDIA A100-...

> 4 x NVIDIA A100-80

> 8 x NVIDIA A100-40

> 8 x NVIDIA A100-80

> Grid

> Hadrons

Lattice / Benchmarks / TursaGpu / Benchmark_Grid

4 x NVIDIA A100-40

Overview

Change Log

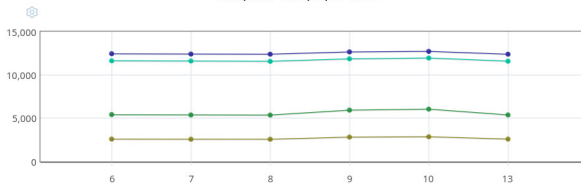
Problems

Statistics

Build Chains

Flaky Tests 0

Compute, GFlops per node



GFlops per nodes All None

Comparison Point

Wilson Fermion

Staggered Fermion

DomainWall Fermion

☐ Average☐ Show failed

Stability

- All build artifacts are backed up to a cloud storage provider
- Configuration of TeamCity itself is tracked in a git repository

Hardening

- Build servers are not publicly visible, communication to main server is “one-way”
- Build agents run as dedicated user with limited permissions on the cluster, e.g., no access to research data

Future outlook

- Solution is scalable to multiple HPC clusters

This CI/CD solution was funded by the STFC DiRAC Facility

Questions?

for a live demo, come talk to me later, or visit
<https://ci.dev.dirac.ed.ac.uk>
(choose “Log in as guest”)