



Contribution ID: 70

Type: **Talk (15min + 5min)**

## Integrated Continuous Benchmarking

*Wednesday, March 6, 2024 3:30 PM (20 minutes)*

When developing research software, it is often relevant to track its performance over time. It is even vital when targeting high-performance computing (HPC). Changes to the software itself, the used toolchains, or the system setup should not compromise how fast users obtain their results. Ideally, performance or scalability should only ever increase. Hence benchmarking should be an integral part of testing, in particular for HPC codes. At the same time, up-to-date benchmarks that are publicly available can advertise the code and inform users how to set-up the software in the most ideal way or whether they are achieving the expected performance.

To limit the burden on developers, the aforementioned steps should be automated within continuous integration (CI) practices, introducing continuous benchmarking (CB) to it. For HPC, an added complexity is the requirement of more than the usual CI backends, with access to longer running steps and more resources than available on a single node. Reusing test cases that are easily run by hand is another simplification for developers that may not be familiar with the research field. We show our solution to CB that we use at the Juelich Supercomputing Centre (JSC), where we combine the already implemented benchmarking via the Juelich Benchmarking Environment (JUBE) with properly authenticated CI steps running on the supercomputing systems at JSC. The combined results, including the evolution over time, are then further processed and displayed on pages published via CI.

### Slot length

**Primary author:** BRÖMMEL, Dirk

**Co-authors:** FRITZ, Jakob (FJZ, JSC); SPECK, Robert

**Presenter:** BRÖMMEL, Dirk

**Session Classification:** Continuous Integration - Current Research and Beyond!

**Track Classification:** Continuous Integration: Basic