



Sandia
National
Laboratories

Exceptional service in the national interest

TRILINOS PERFORMANCE MONITORING WITH WATCHR

Brian Kelley & Chris Siefert

TUG 2023, November 1, 2023

Special Thanks: Elliott Ridgway for developing Watchr

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2023-11652PE

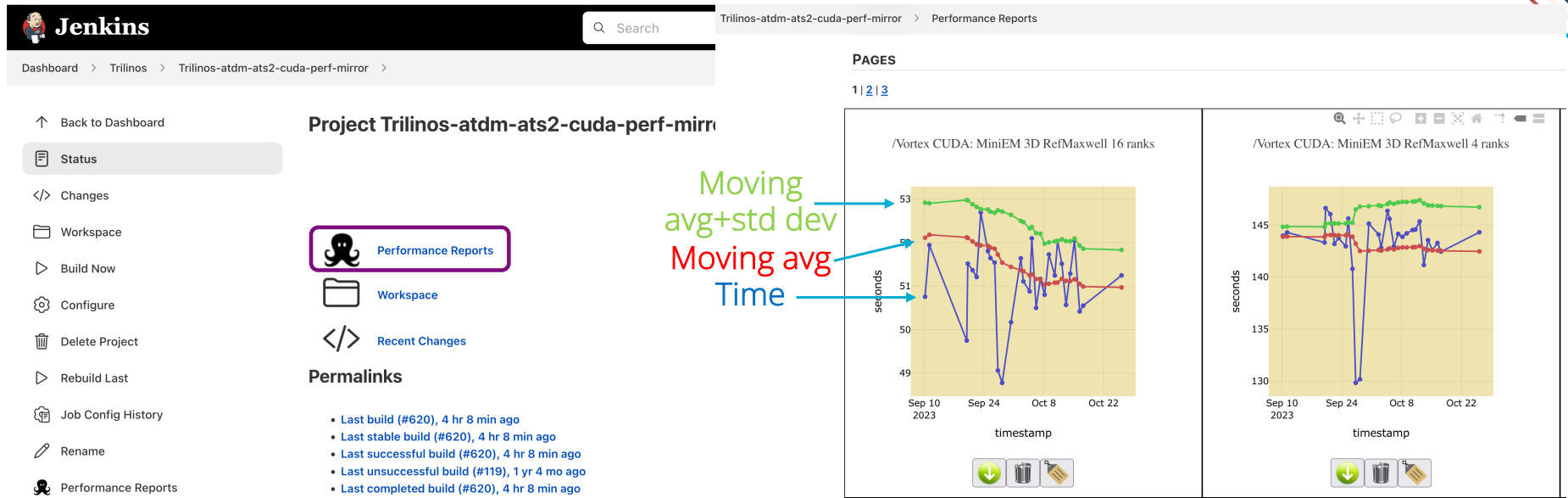




PERFORMANCE MONITORING WITH WATCHR

- Watchr is an open-source Jenkins plugin developed at Sandia
 - Created and developed by Elliott Ridgway until FY24, carried on by Eric Ho (both 1424)
 - And installed on Sandia Jenkins instances (like jenkins-son.sandia.gov)
- Adds a “Performance Reports” page to your build
- Perfect for nightly performance data
 - x-axis is build date
 - y-axis is elapsed time
- Teuchos::StackedTimer can output data in Watchr XML format

PERFORMANCE MONITORING WITH WATCHR





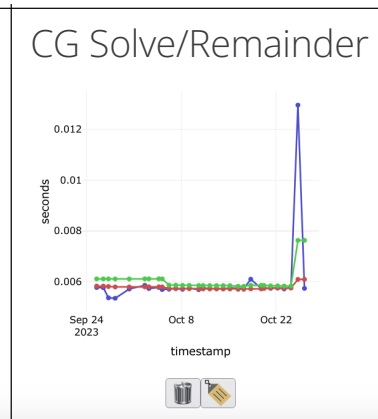
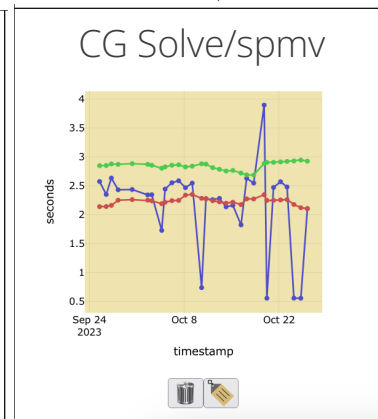
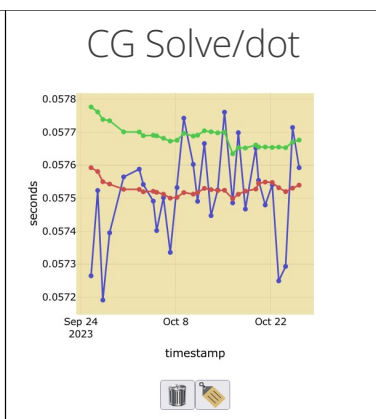
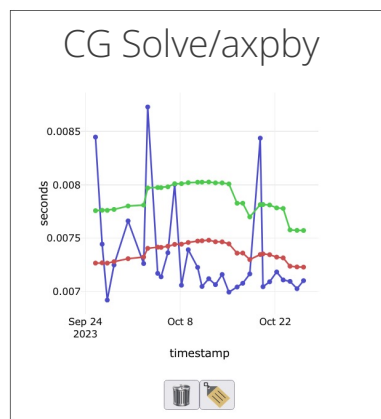
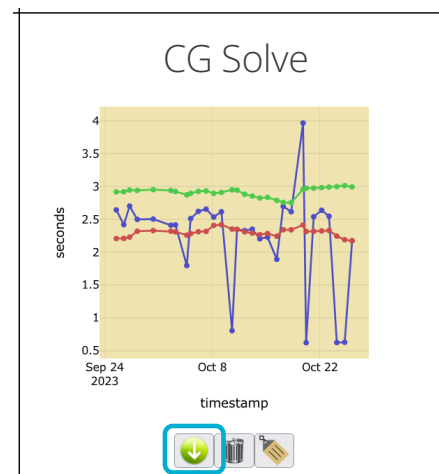
MOTIVATION

- It's useful to see how Trilinos code changes affect performance in different workloads
 - Changes can have unexpected impact, both positive and negative
- There are several architectures whose performance we are interested in
 - Intel CPU: CTS-1 (Broadwell), CTS-2 (Sapphire Rapids)
 - GPU: ATS-2 (Nvidia V100), Crusher/Frontier (AMD MI250X)
- And many different workloads, including:
 - Fundamental kernels like matrix-vector product (SpMV)
 - Specific linear systems from applications (SierraTF, SPARC)
 - MueLu multigrid setup and apply
 - Tpetra graph assembly
- This work doesn't reduce the amount of data, but it does let us assess performance at a glance and track down problematic changes
 - Mouse over a data point to see the build date and Trilinos version



WATCHR AND TEUCHOS STACKED TIMER

- Watchr supports hierarchical data
 - Timers can have sub-timers
 - Just like Teuchos::StackedTimer
 - StackedTimer can output Watchr XML-formatted data
- Real example: Tpetra CGSolve
- Conjugate gradient calls smaller kernels:
 - axpby
 - dot
 - spmv





GENERATING WATCHR DATA

- In code:
 - Use `Teuchos::StackedTimer` and `Teuchos::TimeMonitor` around code regions of interest
 - Call `stackedTimer->reportWatchrXML("my benchmark", comm);` to write Watchr XML
 - This has no effect if `$WATCHR_PERF_DIR` is not set (see below)
- At runtime, set some environment variables
 - [Required] Set `$WATCHR_PERF_DIR` to the absolute path of where XML files should be saved
 - [Optional] Set `$WATCHR_BUILD_NAME` to a description of your build (e.g. "GCC+OpenMP")
 - [Optional] Set `$TRILINOS_GIT_SHA` to the Trilinos version (run `"git rev-parse HEAD"`)



ADDING TIMERS TO CODE

```
#include "Teuchos_TimeMonitor.hpp"
#include "Teuchos_StackedTimer.hpp"

int main()
{
    using Teuchos::RCP;
    using Teuchos::TimeMonitor;
    using Teuchos::StackedTimer;
    // ... initialize MPI, Kokkos, Tpetra ...

    // Create and activate the base StackedTimer. Its name does not appear in Watchr plots.
    RCP<StackedTimer> timer = Teuchos::rcp(new StackedTimer("Global"));
    TimeMonitor::setStackedTimer(timer);
    {
        // Create a scoped sub-timer
        TimeMonitor t(*TimeMonitor::getNewTimer("Thing 1"));
        doThing1();
    }
    doThing2();
    timer->stopBaseTimer();
    timer->reportWatchrXML("My problem", comm);
}
```



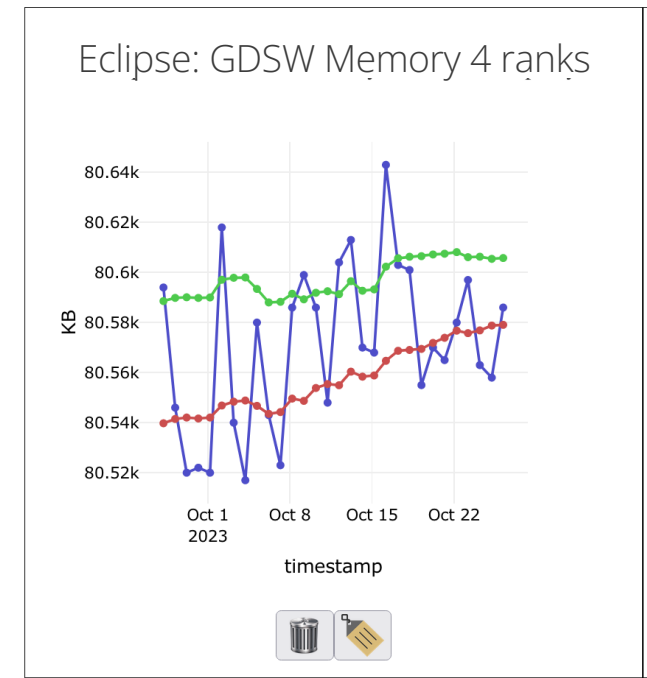
RUNNING WITH AN EXISTING DRIVER

- If your problem is a preconditioned linear solve (Belos + Ifpack2 or MueLu), configure Trilinos with:
 - -D Trilinos_TEST_CATEGORIES="PERFORMANCE"
 - -D Trilinos_ENABLE_TESTS=ON
 - -D Trilinos_ENABLE_EXAMPLES=ON
- and run a driver:
 - Ifpack2_tif_belos.exe <flags> --with_stacked_timer --problem-name="solve1"
 - Iterative solver from Belos, with preconditioner from Ifpack2
 - MueLu_Driver.exe <flags> --stacked-timer
 - Iterative solver from Belos, with MueLu preconditioner and smoother(s) from Ifpack2



NEW FEATURE: MEMORY USAGE

- Tpetra can now record and plot memory usage at certain point(s) in a program
- Currently used on Tpetra matrix communication (import, export)
 - GDSW (two-level overlapping domain decomposition) setup
- Will add to Belos+Ifpack2 problems soon





CURRENTLY RUNNING BENCHMARKS

- Panzer MiniEM
- Belos CG solve with MueLu defaults on large 7-point stencil problem
- Belos GMRES with Ifpack2 overlapping ILU(k)
- Ifpack2 block tridiagonal apply
- Distributed matrix-vector product (Tpetra::CrsMatrix::apply)
- Distributed matrix-matrix product (TpetraExt::MatrixMatrix::Multiply)
- Tpetra FECrsMatrix assembly
- Communication patterns for GDSW preconditioner (measuring memory usage, not time)



CONCLUSION

- Contact the Tpetra developers at tpetra-developers@software.sandia.gov
- Watchr on GitHub: <https://github.com/sandialabs/watchr-core>
- Main dashboard hosted on jenkins-son.sandia.gov
 - Only visible to Sandians
 - Actively monitored by Tpetra developers
- As Trilinos package developers, we want to support application needs
- **We encourage apps to give us proxy problems to add to the nightly performance runs**

QUESTIONS?

- Contact the Tpetra developers at tpetra-developers@software.sandia.gov

