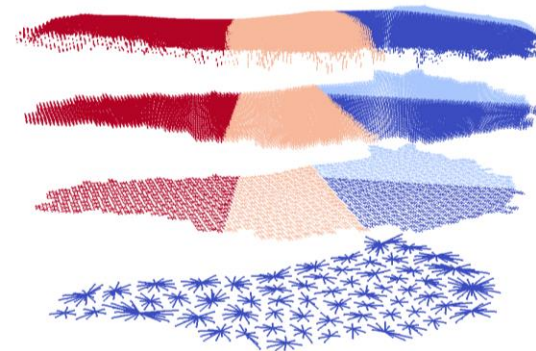
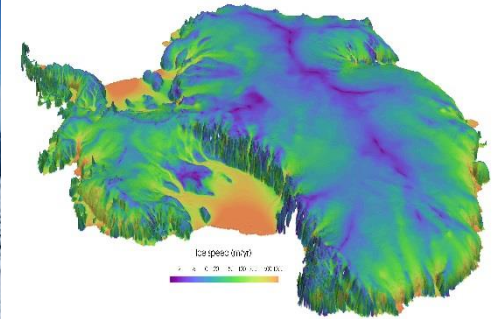


# Performance Variation of the Albany Land-Ice Code at Sandia

Kyle Shan, ICME Xplore – Winter 2020

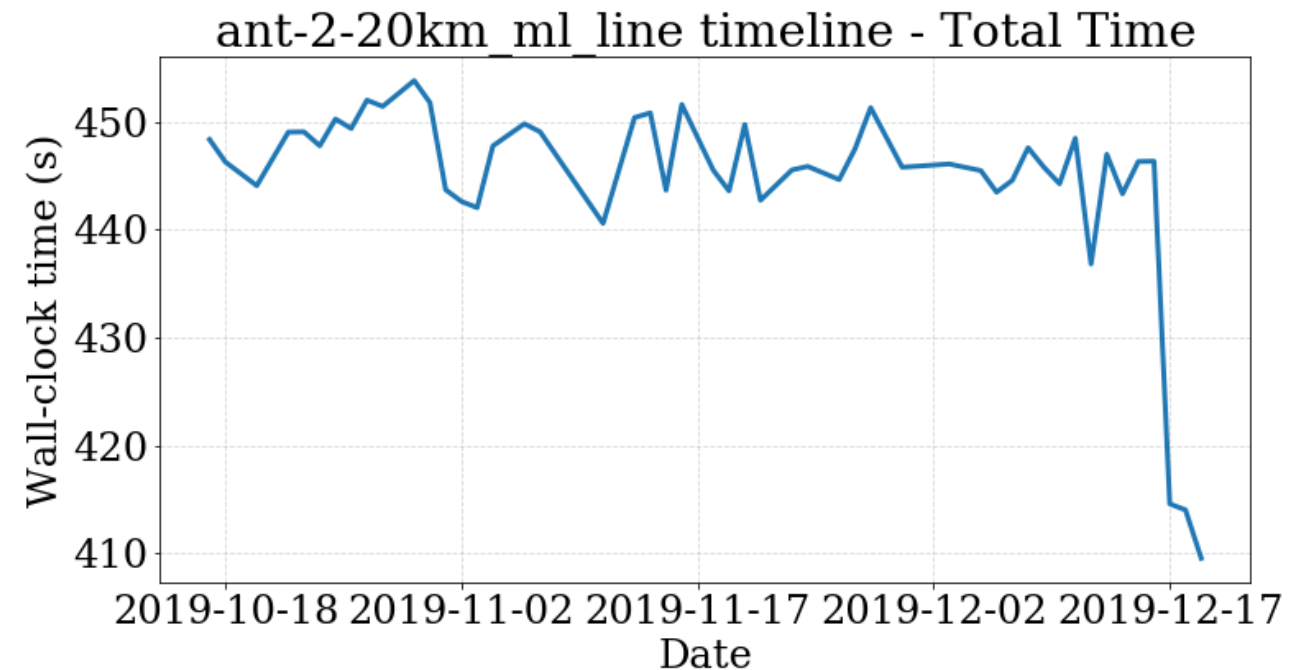
# Background

- Land ice modeling is a critical component of Earth System Models, which predict changes in sea level over decades/centuries
- Albany Land-Ice Code being developed to run on various HPC architectures
- Ongoing improvements to various code aspects – changes in performance may occur (either intended or unintended)
- **Desired solution: automated review of nightly test cases to identify performance changes**



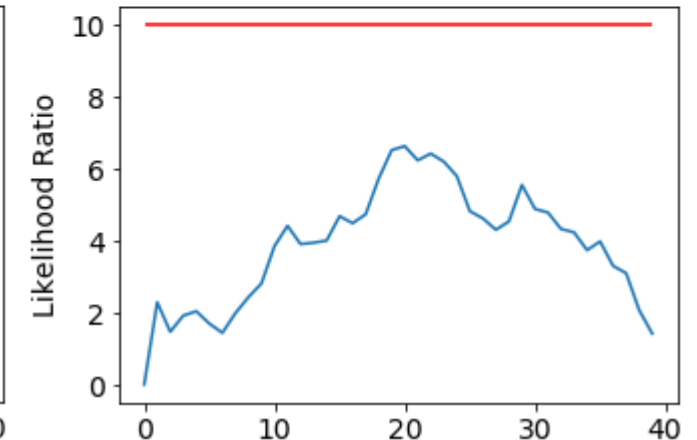
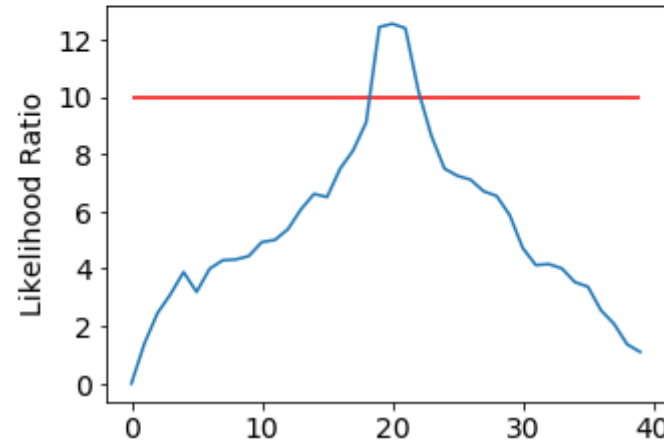
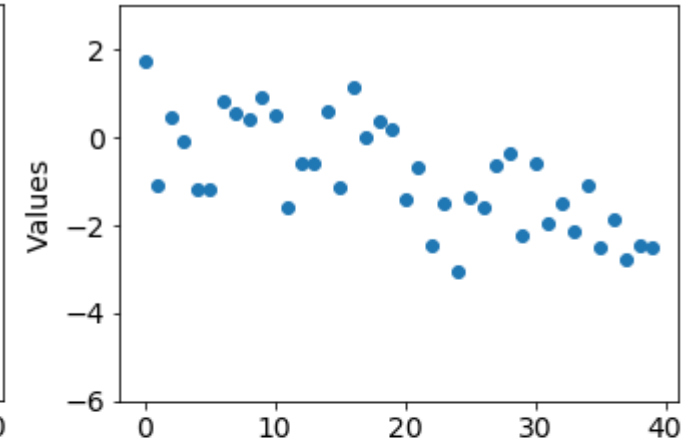
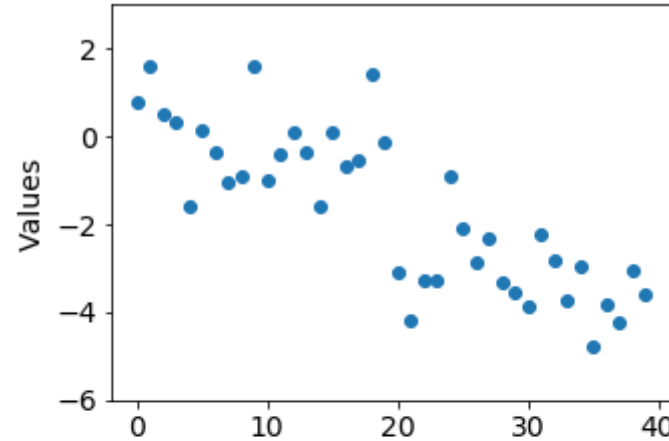
# Data

- Nightly test runs of current code base with various ice sheets and/or solvers, stored in daily .json file
- Timers measure individual components of the model (e.g. residual/Jacobian fill, preconditioner construction, solver)
- Metadata: Trilinos/Albany repository commit ID, compiler



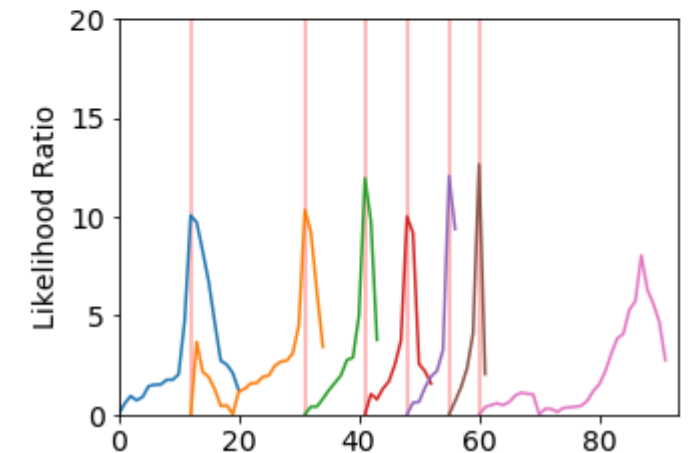
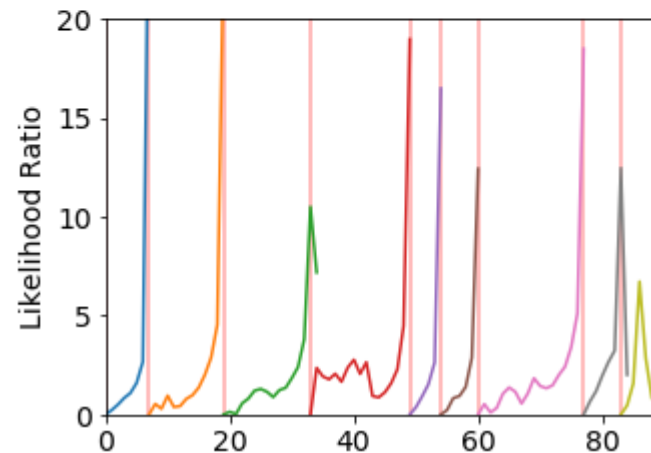
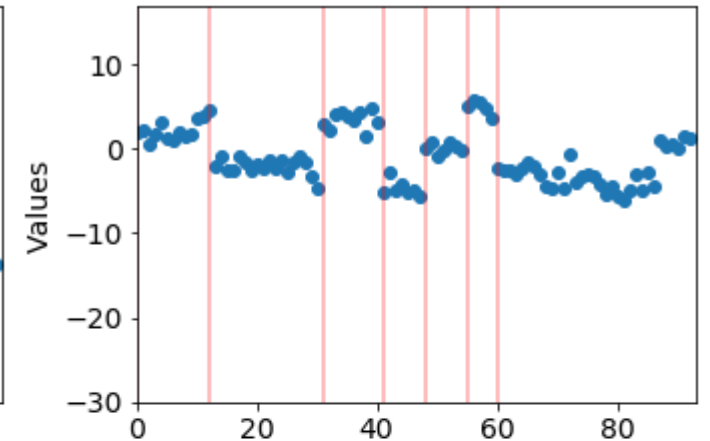
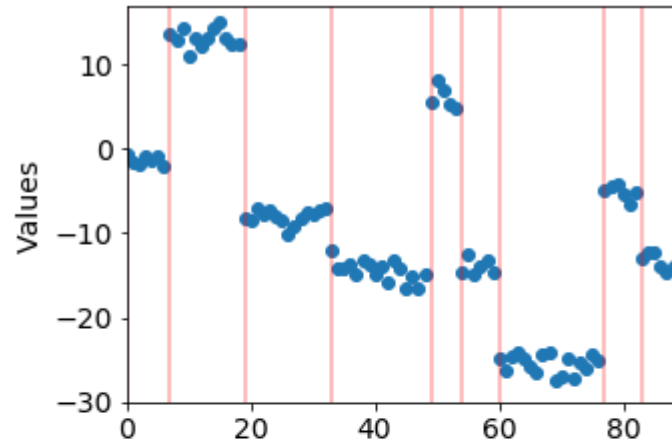
# Changepoint Detection

- Given a time series  $x_1, \dots, x_n$ , where  $x_1, \dots, x_\nu$  have distribution  $f_0$  and  $x_{\nu+1}, \dots, x_n$  have distribution  $f_1$ , can we identify  $\nu$ ?
- Generalized Likelihood Ratio (GLR): pick  $\nu$  that maximizes likelihood of the changepoint model versus the single-distribution model
- Small amount of data – assume parameterized distribution
- Set a minimum threshold to ensure that a change point exists



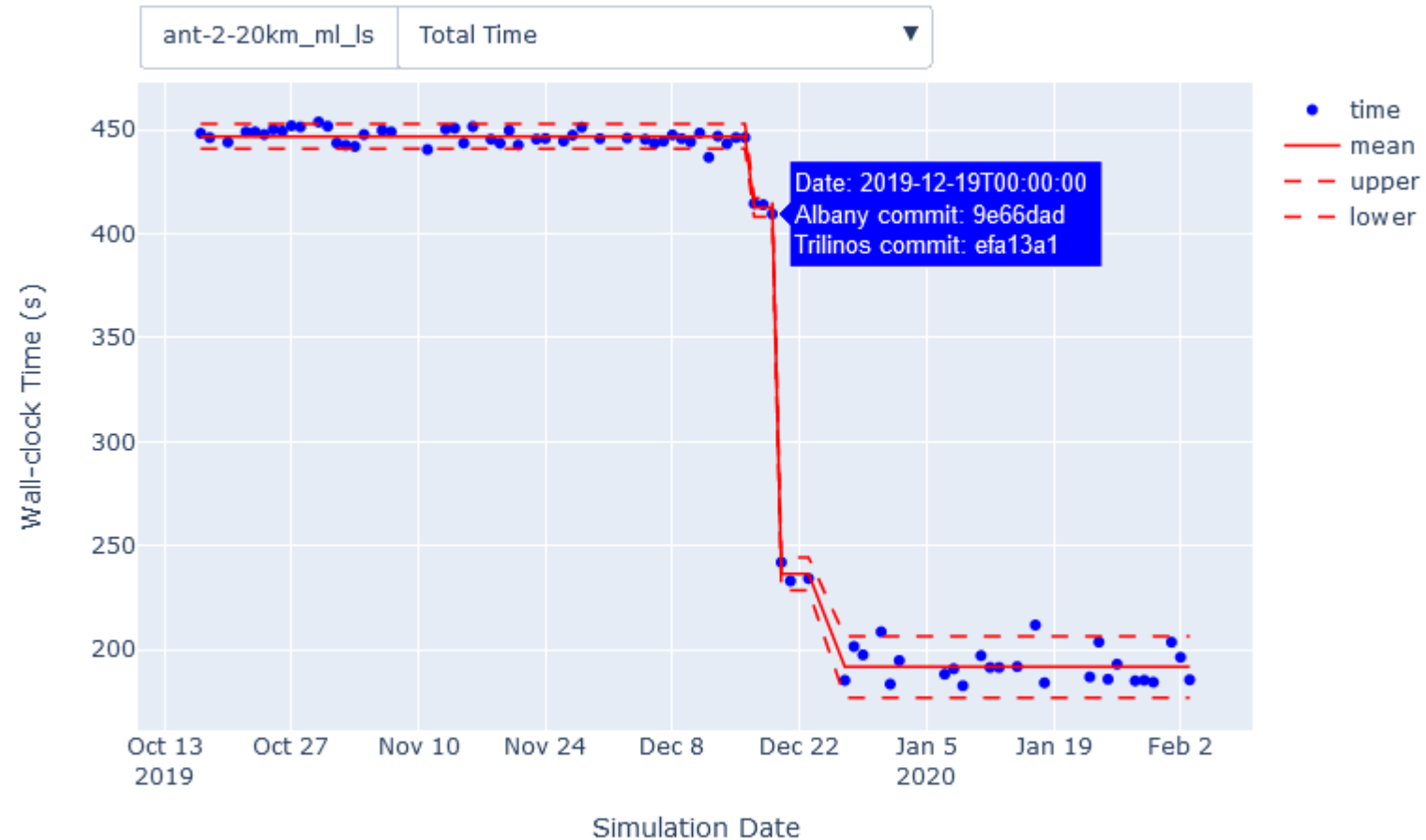
# Multiple Changepoints

- Our dataset contains an unknown number of changepoints
- Perform analysis sequentially; once a changepoint is detected, prior data is no longer considered
- Detection delay and accuracy depends on magnitude of changes and threshold
- Missed changepoints can decrease future accuracy

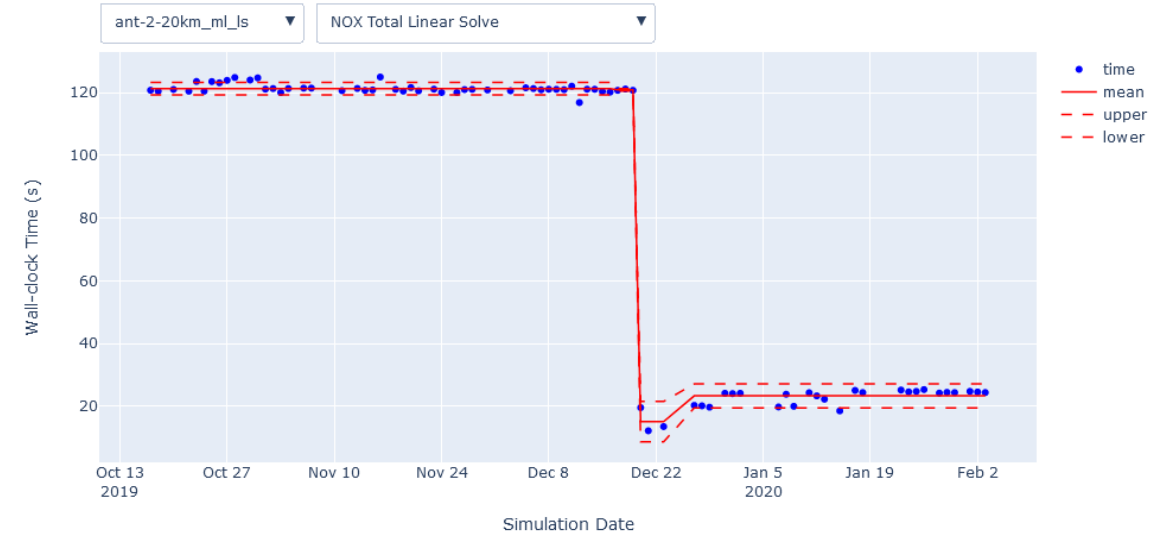
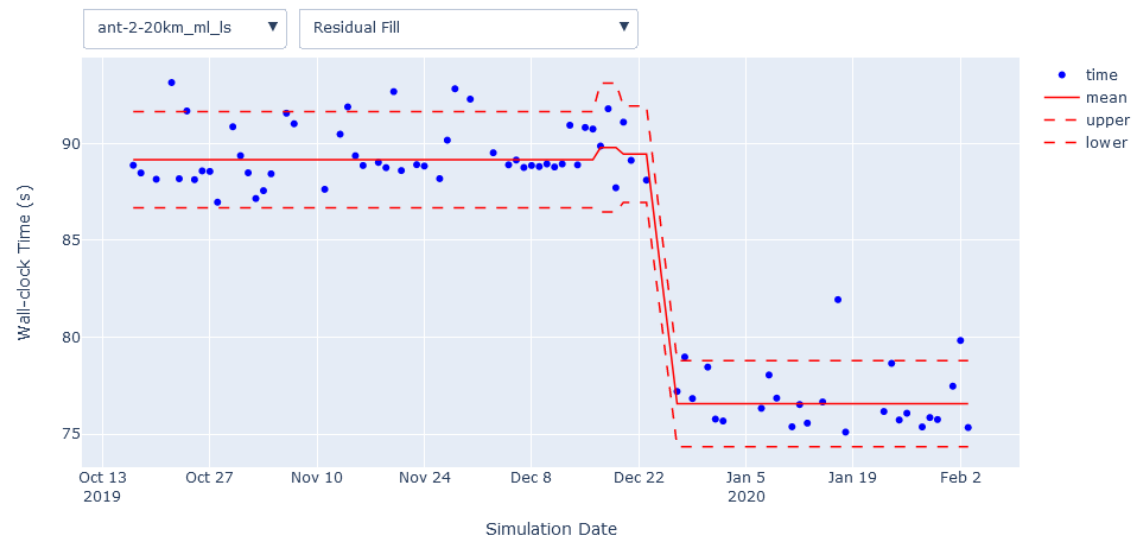
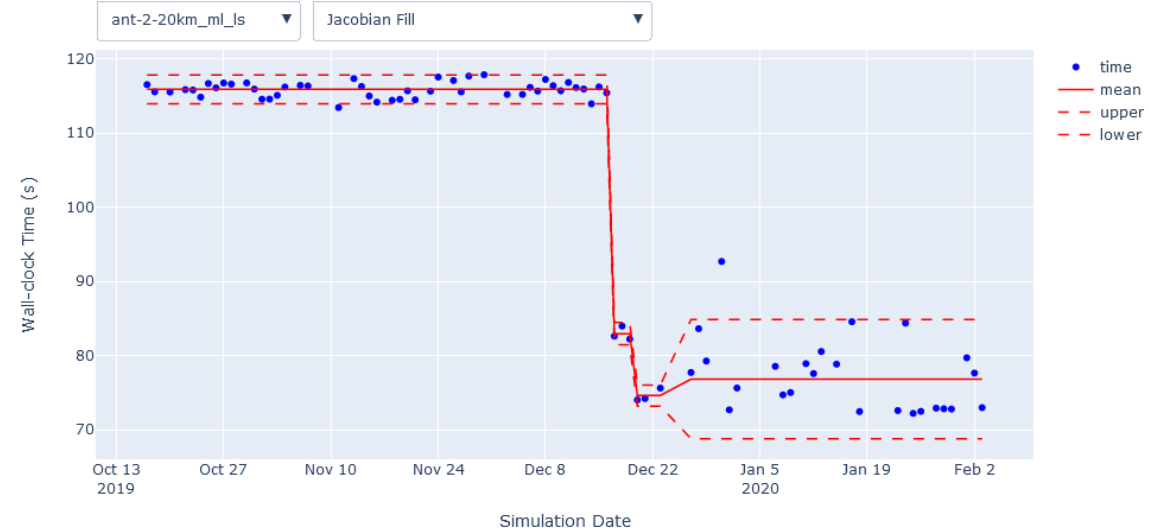
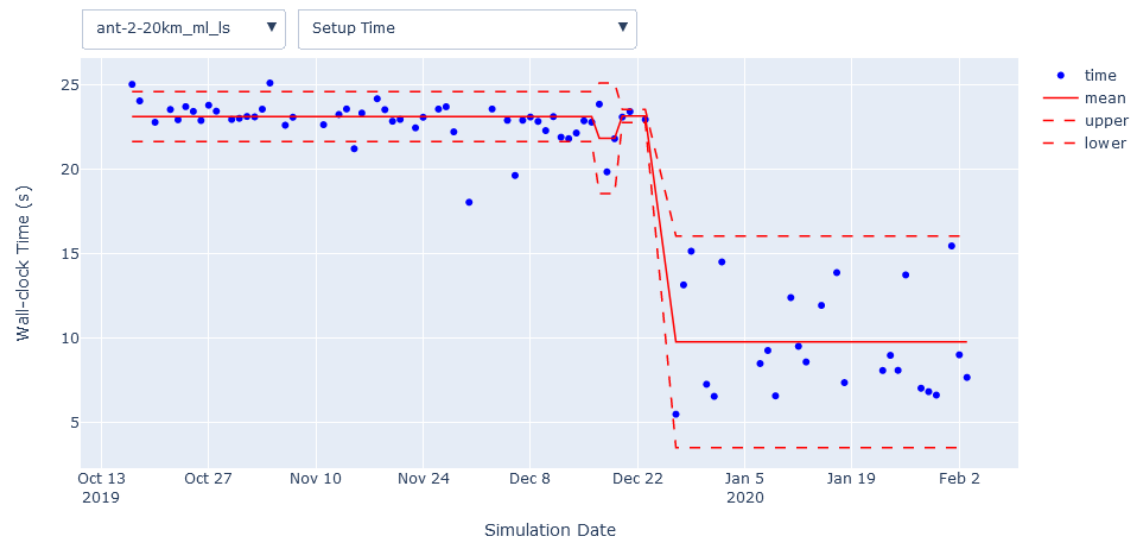


# Current Progress

- Apply GLR to test cases/timers individually, and then merge changepoints over all timers
- Jupyter notebook loads most recent data from .json files and performs sequential test
- Plot results in interactive chart with Plotly for Python; exportable to offline HTML file



# Current Progress



# Future Work

- Combine detections from multiple data sources and methods to improve detection rate
- Compare timers from different computing architectures and/or solvers
- Incorporate suggestions to improve automated report format
- Write documentation
- Integrate with Sandia's current system