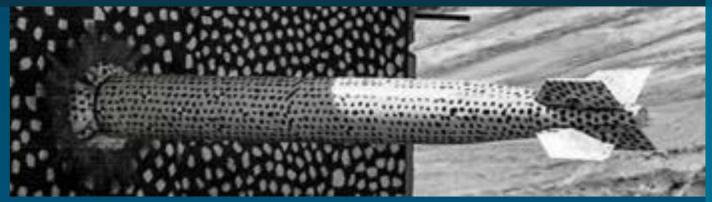
Sandia National Laboratories

SAND2020-6699PE

Automated performance testing and analysis in Albany Land Ice



June 17th, 2020

PRESENTED BY

Jerry Watkins and Irina Tezaur

Climate modeling meeting

SAND

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.

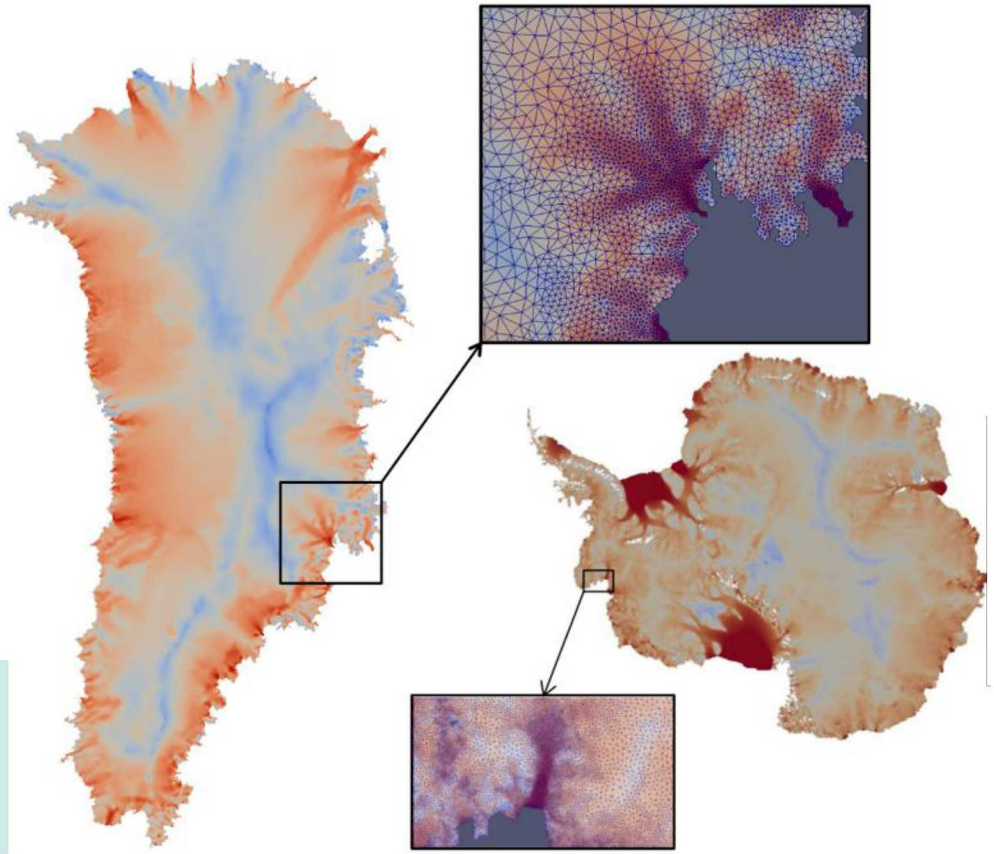
ProSPect – project under SciDAC

ProSPect = Probabilistic Sea Level Projections from Ice Sheet and Earth System Models
5 year SciDAC4 project (2017-2022).

Role: to **develop** and **support** a robust and scalable land ice solver based on the First-Order (FO) Stokes model → *Albany Land Ice*

Requirements for *Albany Land Ice* (formerly *FELIX*):

- ***First-order Stokes*** *model*
- ***Unstructured*** meshes
- ***Scalable, fast*** and ***robust***
- ***Verified*** and ***validated***
- ***Portable*** to new architecture machines
- ***Advanced analysis*** capabilities: deterministic inversion, model calibration, uncertainty quantification, sensitivity analysis



As part of ***DOE E3SM Earth System Model***, solver will provide actionable predictions of 21st century sea-level change (including uncertainty bounds).

https://doe-prospect.github.io/

Overview for performance

Application Performance and Portability:

- Performance quantified by wall-clock time (s) - steady state solution of the first-order Stokes equations on high resolution (up to ~1 km) ice sheet meshes (Antarctica, Greenland)
- Portability characterized by the correct and efficient execution of the above on various HPC platforms (NERSC Cori – CPUs and ORNL Summit - GPUs)

Two key problems:

1. Need ability to identify performance variations
   - Active development in both Trilinos/Albany cause changes in performance to go unnoticed
2. Need ability to showcase performance improvements to stakeholders
   - Manual performance analysis across HPC platforms is time consuming and imprecise

Progress in performance and portability is hindered by the above

Solution:

Extend nightly testing framework to include performance testing and analysis (CMDV-SM)

- Requires **automated data collection, processing and visualization** for large scale simulations

Data Collection

Need automated builds, tests and data extracting tools (nightly testing) and resources

- Builds are automated on clusters using cron/Jenkins
- Cmake is used to define tests
- Data is collected and monitored in CDash

```
#create test
add_test(${testName}_fea_1ws_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_FiniteElementAssembly_SingleWorkset.yaml ${ALBANY_EXTRA_ARGS})
add_test(${testName}_fea_mem_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_FiniteElementAssembly_Memoization.yaml ${ALBANY_EXTRA_ARGS})
add_test(${testName}_muk_ls_mem_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_MueLuKokkos_LineSmoothing_Memoization.yaml ${ALBANY_EXTRA_A
IF (NOT ALBANY_ENABLE_CUDA)
  add_test(${testName}_mu_dls_1ws_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_MueLu_DecoupledLineSmoothing_SingleWorkset.yaml ${ALBANY_
  add_test(${testName}_ml_ls_mem_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_ML_LineSmoothing_Memoization.yaml ${ALBANY_EXTRA_ARGS})
  add_test(${testName}_mu_ls_mem_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_MueLu_LineSmoothing_Memoization.yaml ${ALBANY_EXTRA_ARGS})
  add_test(${testName}_mu_dls_mem_np${ALBANY_NUM_PROCS} ${ALBANY_SFAD8_EXE} input_albany_MueLu_DecoupledLineSmoothing_Memoization.yaml ${ALBANY_EX
ENDIF()
```

		Configure		Build		Test			
Site	Build Name	Error	Warn	Error	Warn	Not Run	Fail	Pass	Start Time
waterman.sandia.gov	waterman-CUDA-Albany-Perf-Tests	0	1	0	0	0	0	4	17 hours ago
blake.sandia.gov	blake-serial-sfad-Albany-PerfTests	0	1	0	0	0	0	19	13 hours ago

Data Processing

Need to store **relevant** data

- Automated python scripts used to extract and store data in json file
- All output compressed and stored for future extraction (other data might become relevant)
- All data available open source on github: https://github.com/ikalash/ikalash.github.io
- Easy access, common format makes it easy to collaborate (e.g. universities, RAPIDS)

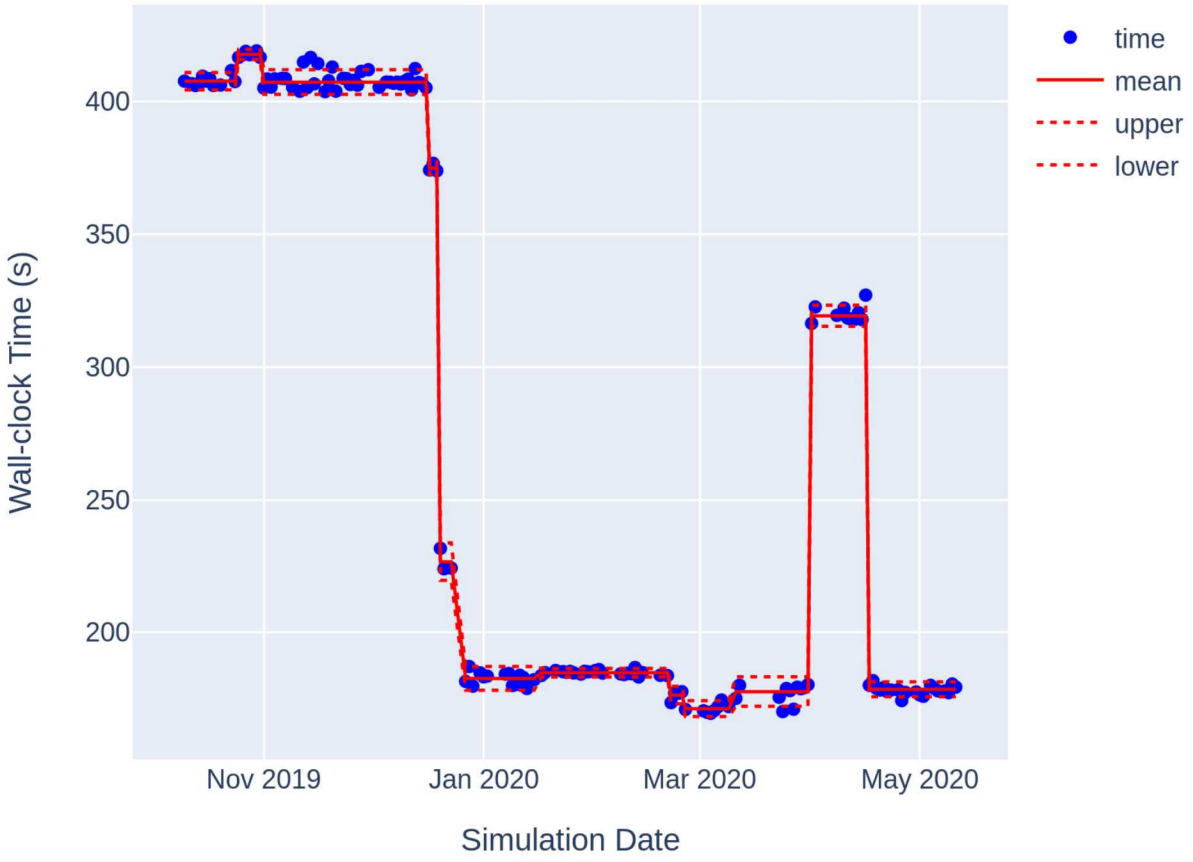

```
"ant-2-20km_ml_ls_np384": {
  "Albany cxx compiler": "Intel 18.0.1.20171018",
  "Albany git branch": "master",
  "Albany git commit id": "1204aa2",
  "Simulation start time": "05:31:33",
  "Trilinos git commit id": "81c48c3",
  "case": "ant-2-20km_ml_ls",
  "date": 20200616,
  "max host memory": 3013.199219,
  "max kokkos memory": 2451.87,
  "np": 384,
  "passed": true,
  "timers": {
    "Albany Fill: Jacobian:": 66.9928,
    "Albany Fill: Residual:": 72.4239,
    "Albany Jacobian Fill: Evaluate:": 60.7375,
    "Albany Jacobian Fill: Export:": 4.66407,
    "Albany Residual Fill: Evaluate:": 68.0752,
    "Albany Residual Fill: Export:": 3.65503,
    "Albany Total Time:": 160.121,
    "Albany: Setup Time:": 5.52318,
    "Albany: Total Fill Time:": 139.417,
    "NOX Total Linear Solve:": 10.7768,
    "NOX Total Preconditioner Construction:": 3.55561
  }
},
```

Data Visualization (performance variation)

Performance variations are identified by a changepoint detection algorithm

- Developed by Stanford ICME student, Kyle Shan
- A generalized likelihood ratio test is used
  - Null hypothesis – $\{x\}_{i=1}^{n}$ belongs to a single distribution
  - Alternate hypothesis – there exists some ν such that $\{x\}_{i=1}^{\nu-1}$ and $\{x\}_{i=}^{n}$ are separate distributions
- Jupyter notebooks used to visualize data
- Exported as html for website: https://ikalash.github.io/

Total simulation time for a 2-20km resolution Antarctica mesh, executed nightly



Data Visualization (performance testing)

A daily email report provides a summary of nightly performance tests

- Pass (green): no performance drop detected
- Warning (yellow): changepoint detected but waiting on more observations
- Fail (red): changepoint detected for a few consecutive days

Status

Name	Run Test	Performance Tests (Passes/Warnings/Fails)
AIS-8km-l5-np16	Passed	2/0/0
AIS-4km-l10-np64	Passed	0/2/0
AIS-2km-l20-np256	Failed	0/0/0
AIS-1km-l40-np2048	Passed	1/0/1

AIS-8km-l5-np16 Timers (s)

Timer	Measured	Mean	Std
NOX Lin Solve	200.1	195.2	10.4
Alb. Total Fill	450.1	445.2	10.4

AIS-4km-l10-np64 Timers (s)

Timer	Measured	Mean	Std
NOX Lin Solve	250.1	195.2	10.4
Alb. Total Fill	550.1	445.2	10.4

AIS-2km-l20-np256 test failed...

AIS-1km-l40-np2048 Timers (s)

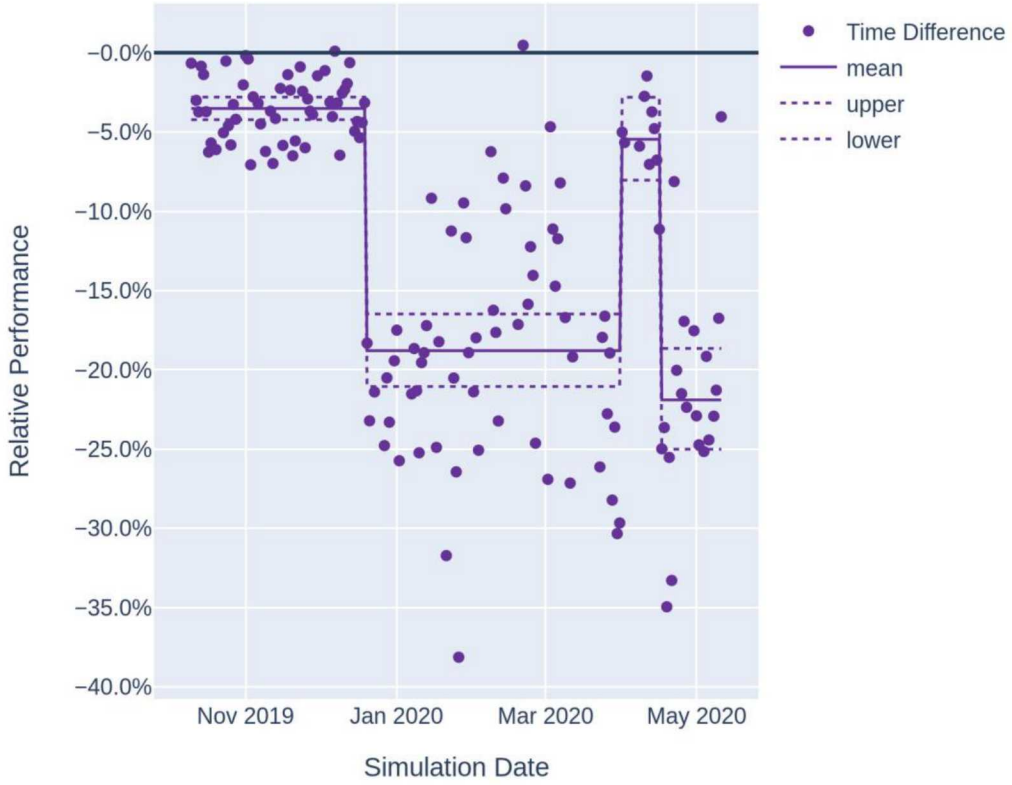
Timer	Measured	Mean	Std
NOX Lin Solve	202.1	195.2	10.4
Alb. Total Fill	460.1	445.2	10.4

Data Visualization (performance analysis)

Comparisons between data sets can be performed using various data streams

- A t-test of the mean relative difference of two data streams can be used to quantify performance improvements with confidence intervals
- Infrastructure is provided in a jupyter notebook
- Examples:
  - Difference between solvers
  - Highlight performance improvements
  - Difference between architectures (WIP)

A comparison between using/not using block decoupling in Ifpack2 to obtain tridiagonal blocks. Measurements of preconditioner construction made between 4/17-5/11 show a relative difference of **-23.2% (99% CI: -25.0, -18.6)** favoring the block decoupling approach.



Challenges – Extensibility, Maintenance

Adding/maintaining a good performance test via ctest is difficult (particularly for next gen hardware)

Adding/maintaining performance tests at scale is difficult (hope the cluster isn't down/over allocated/has a bad node, someone has to monitor the build)

Adding/maintaining python scripts is difficult (one small change results in the person who created the scripts to be available to fix issues)

Hopefully maintenance cost < manual assessment cost

- So far, working pretty well

DEMO