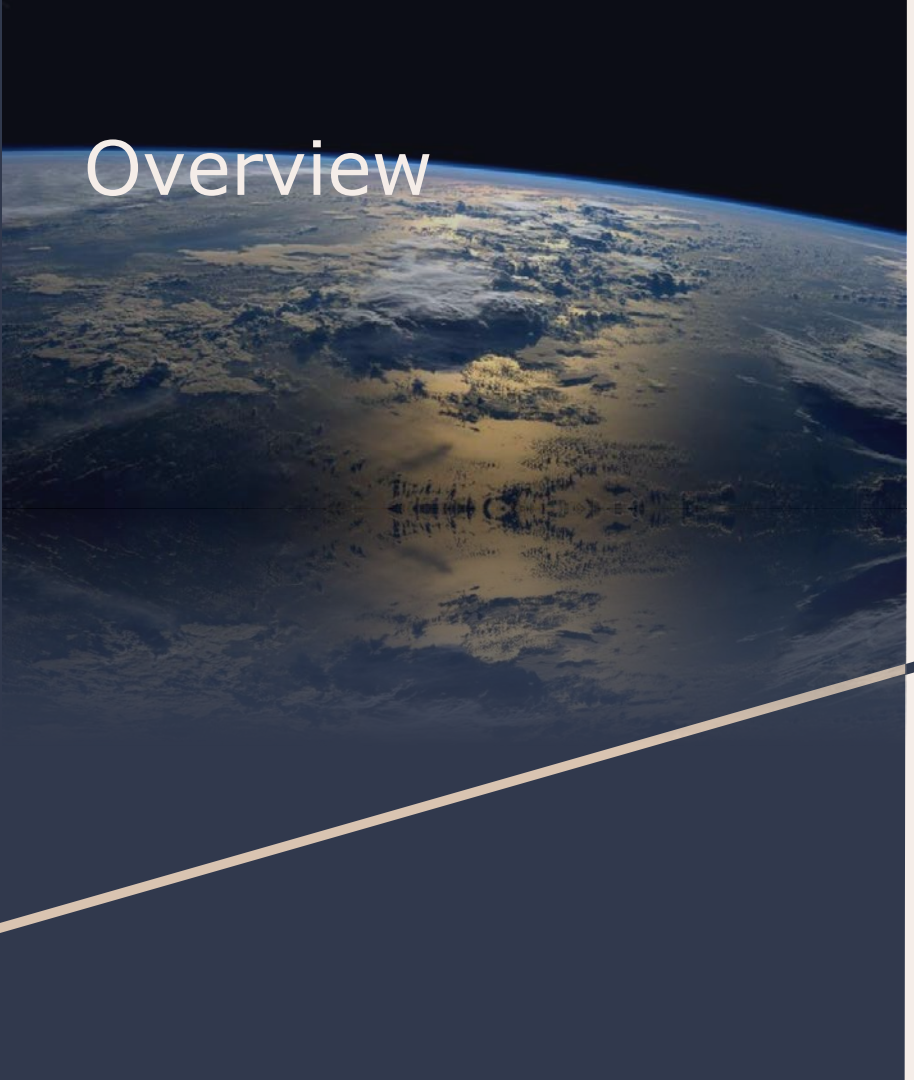
Integration of the Energy Exascale Earth System Model (E3SM) into the Pavilion Test Harness

Timothy Goetsch
New Mexico Institute of Mining and Technology
tgoetsch@lanl.gov

Mentors: Jennifer Green & David Shrader

LA-UR-20-26133



Overview

- The Issue
- The Harness
- The Application
- The Test Definition
- The Results
- Conclusion
- Moving Forward

The Issue

HPC Support Teams:

Support client software

Develop software

Maintain cluster software

Develop procedures

https://meet.google.com/smb-zbjj-wvf

The Issue

Manual Testing is:

- Time Consuming
- Difficult to Port
- Content Management

The Harness

Programming & Runtime

Environments Team

github.com/hpc/pavilion2

The Harness: Pavilion Features

YAML-based Configs

YAML-based Configs

Extensible Python Plugins

Extensible Python Plugins

Portable Tests

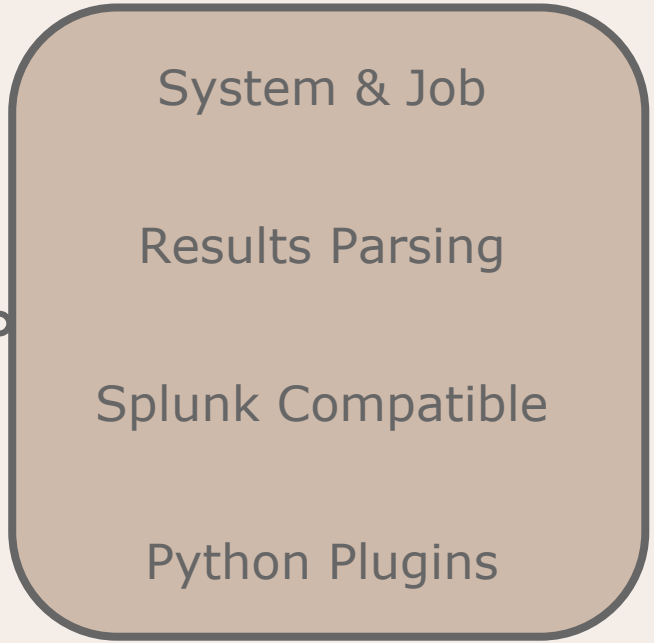
The Harness: Pavilion Features

YAML-based Configs

Extensible Python Plugins

Portable Tests

Automated Results Analysis



The Application

E3SM
Energy Exascale
Earth System Model

The Application: Developers

E3SM
Energy Exascale Earth System Model

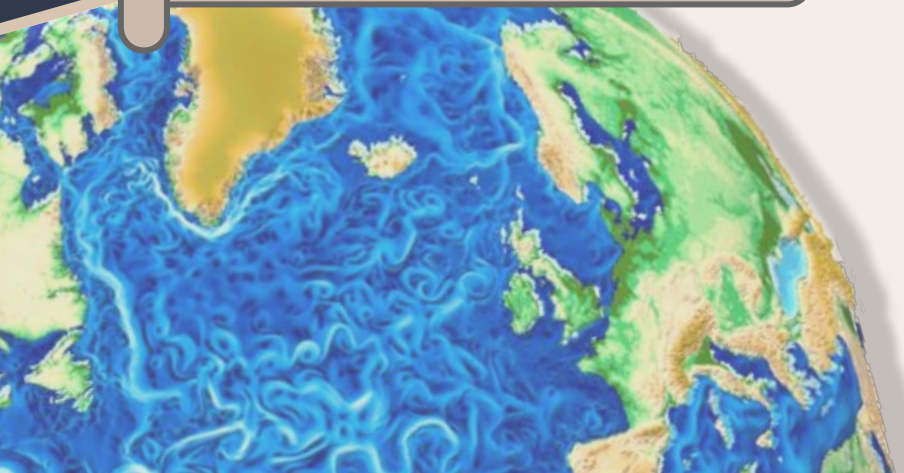
"E3SM is the first end-to-end multi-scale Earth system model, meaning that we can focus model resolution and computer resources toward specific locations to help answer specific questions that are important to the DOE"

- Dr. Stephen Price, LANL

- Over 100 scientists and software engineers
- Multiple DOE laboratories & universities
- Secretary of Energy's Achievement Award in 2015
- E3SM supplies configurations in support of customers running on *LANL HPC Systems*

The Application

E3SM
Energy Exascale
Earth System Model

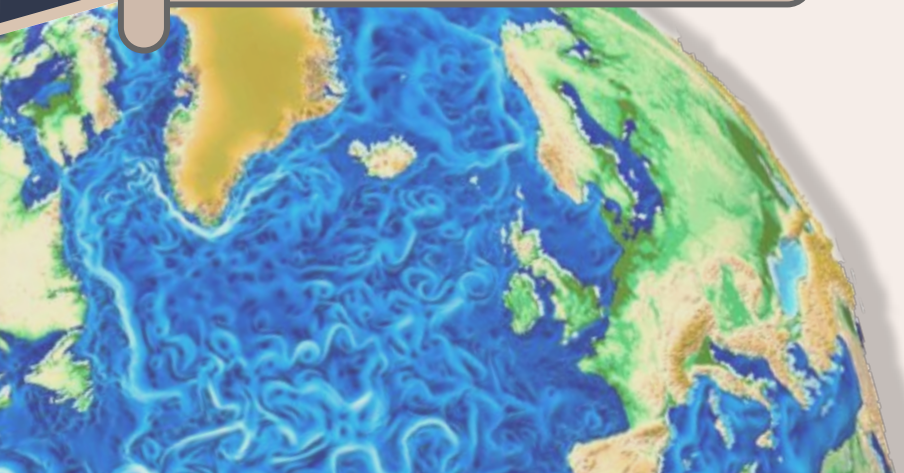


Purpose

Earth System Modeling
Earth System Simulation
Earth System Prediction

The Application

E3SM
Energy Exascale
Earth System Model



Purpose

Earth System Modeling
Earth System Simulation
Earth System Prediction

Facilitate

Scientific Understanding

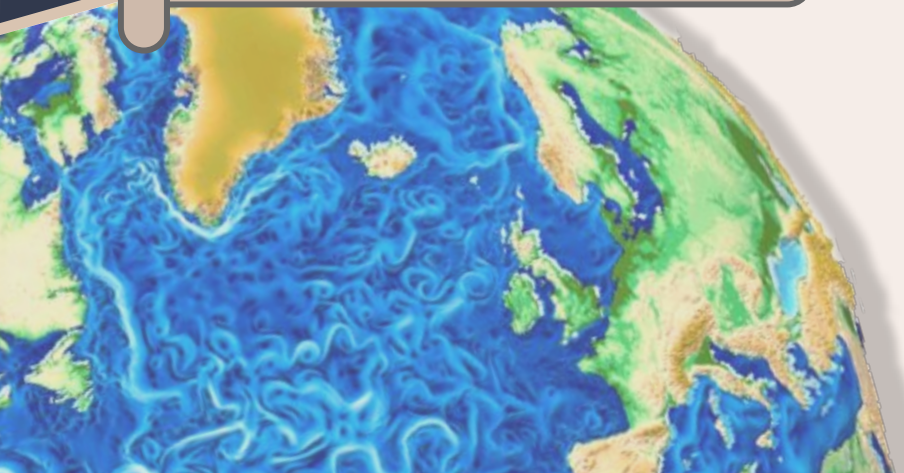
Water Cycle

Bio-Geo-Chemistry

Cryo-sphere

The Application

E3SM
Energy Exascale
Earth System Model



Informs Decisions on

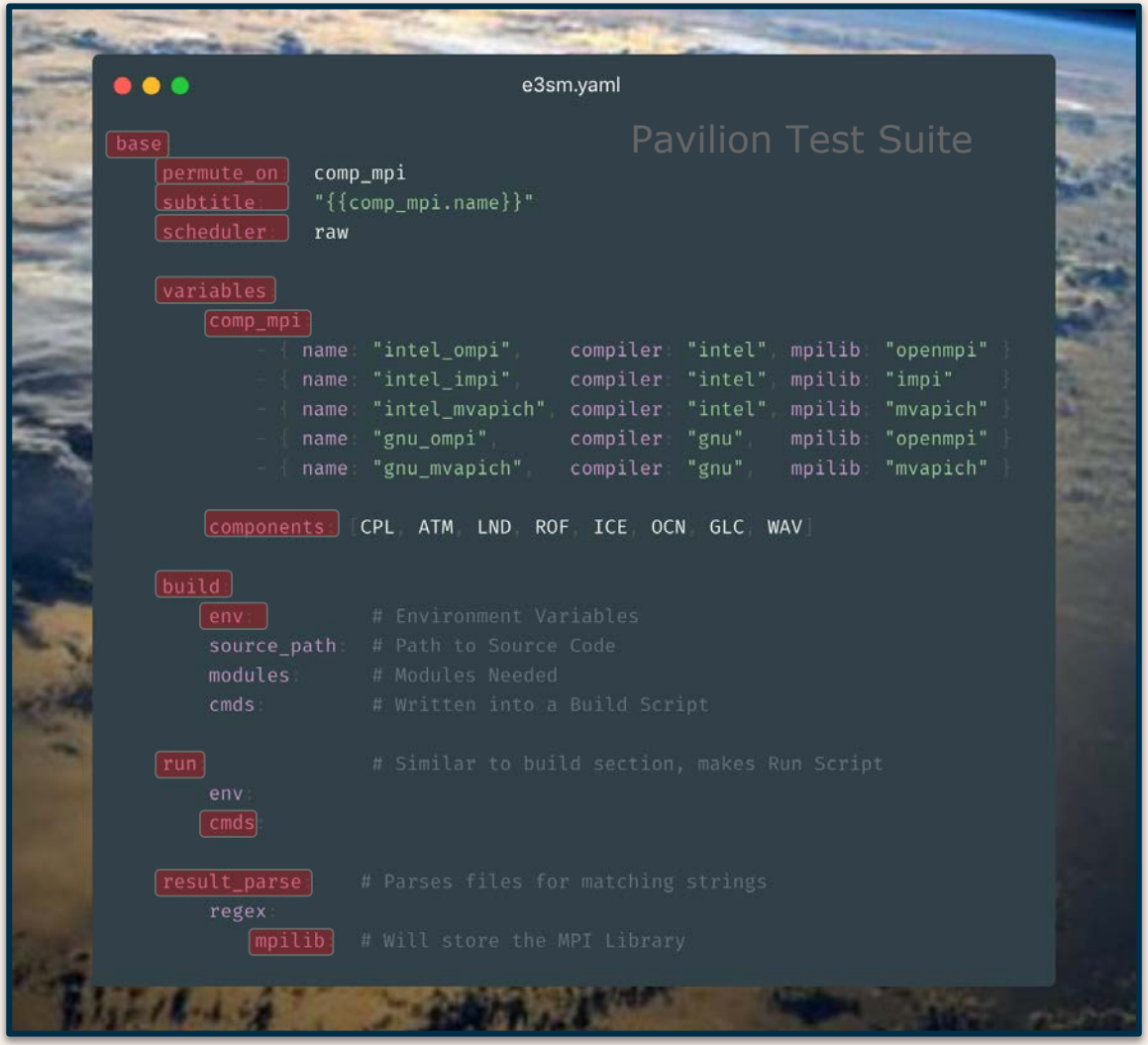
Energy Sector Vulnerabilities

National Nuclear
Stockpile Stewardship

Other Exascale-Class Problems

The Test Definition

The Test Definition



The Test Definition

PAVILION

E3SM

e3sm.yaml

```
variables.comp_mpi:
    - { name: "intel_ompi",
        compiler: "intel",
        mpilib: "openmpi"
      }
    - { name: "intel_impi",
        compiler: "intel",
        mpilib: "impi"
}
```

e3sm.yaml

```
cmds:
    - './create_newcase --handle-preexisting-dirs -r -v -case {{case_root}}/{{e3sm_case}}
      -compiler {{comp_mpi.compiler}} -mach {{sys_name}} -mpilib {{comp_mpi.mpilib}}
      -project {{project}} -compset {{compset}} -res {{grid}}'
```

The Test Definition

PAVILION

E3SM

e3sm.bash

```
./xmlchange -file env_mach_pes.xml -id NTASKS_CPL -val $NPROCS_CPL
./xmlchange -file env_mach_pes.xml -id NTASKS_ATM -val $NPROCS_ATM
./xmlchange -file env_mach_pes.xml -id NTASKS_LND -val $NPROCS_LND
./xmlchange -file env_mach_pes.xml -id NTASKS_ROF -val $NPROCS_ROF
./xmlchange -file env_mach_pes.xml -id NTASKS_ICE -val $NPROCS_ICE
./xmlchange -file env_mach_pes.xml -id NTASKS_OCN -val $NPROCS_OCN
./xmlchange -file env_mach_pes.xml -id NTASKS_GLC -val $NPROCS_GLC
./xmlchange -file env_mach_pes.xml -id NTASKS_WAV -val $NPROCS_WAV
```

e3sm.yaml

```
"[~./xmlchange -file env_mach_pes.xml -id NTASKS_{{components}} -val ${NPROCS_{{components}}}\n~]"
```

The Test Definition

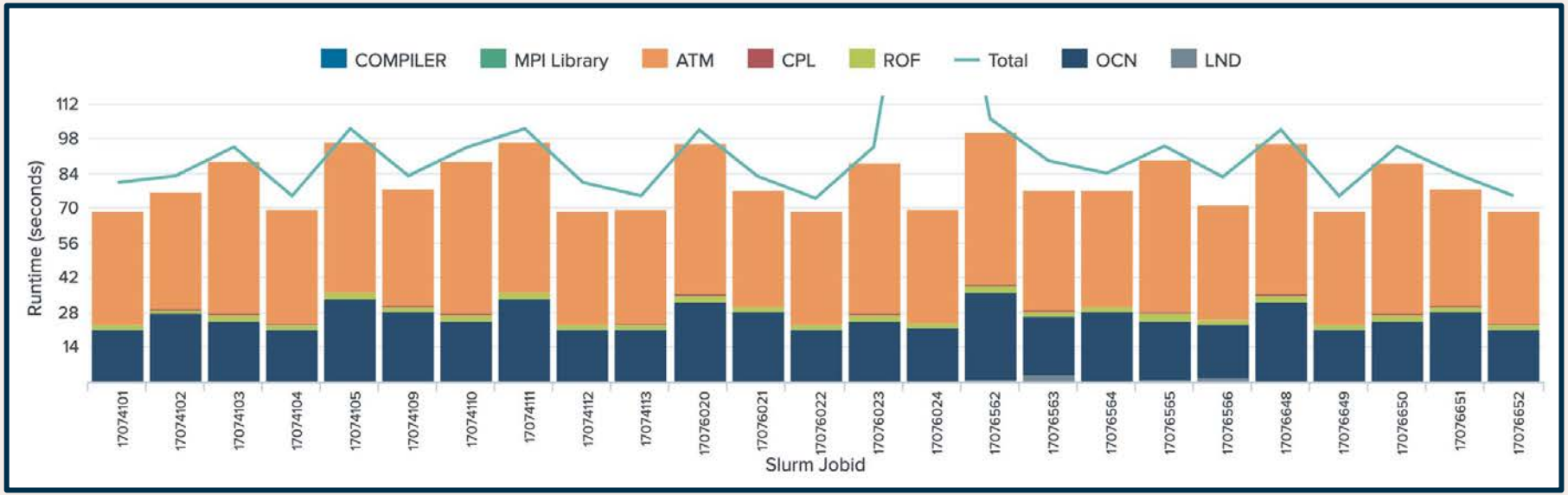
e3sm.yaml

```
results_parse:
    regex:
        tot_run_seconds:
            files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
            regex: 'TOT Run Time:\s+((?:\d+\.)?\d*) seconds'
            action: 'store'
        cpl_run_seconds:
            files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
            regex: 'CPL Run Time:\s+((?:\d+\.)?\d*) seconds'
            action: 'store'
        atm_run_seconds:
            files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
            regex: 'ATM Run Time:\s+((?:\d+\.)?\d*) seconds'
            action: 'store'
```

The Results

splunk>

slurm_jobid	COMPILER	MPI Library	ATM	CPL	ROF	Total	OCN	LND
17074101	intel	impi	45.344	0.341	1.802	80.092	20.692	0.677
17074102	intel	mvapich	46.500	0.434	1.612	82.728	27.130	0.734
17074103	gnu	ompi	61.240	0.400	2.692	94.350	23.994	0.702
17074104	intel	ompi	45.276	0.613	1.826	74.688	20.986	0.694
17074105	gnu	mvapich	60.203	0.418	2.652	101.701	32.882	0.676



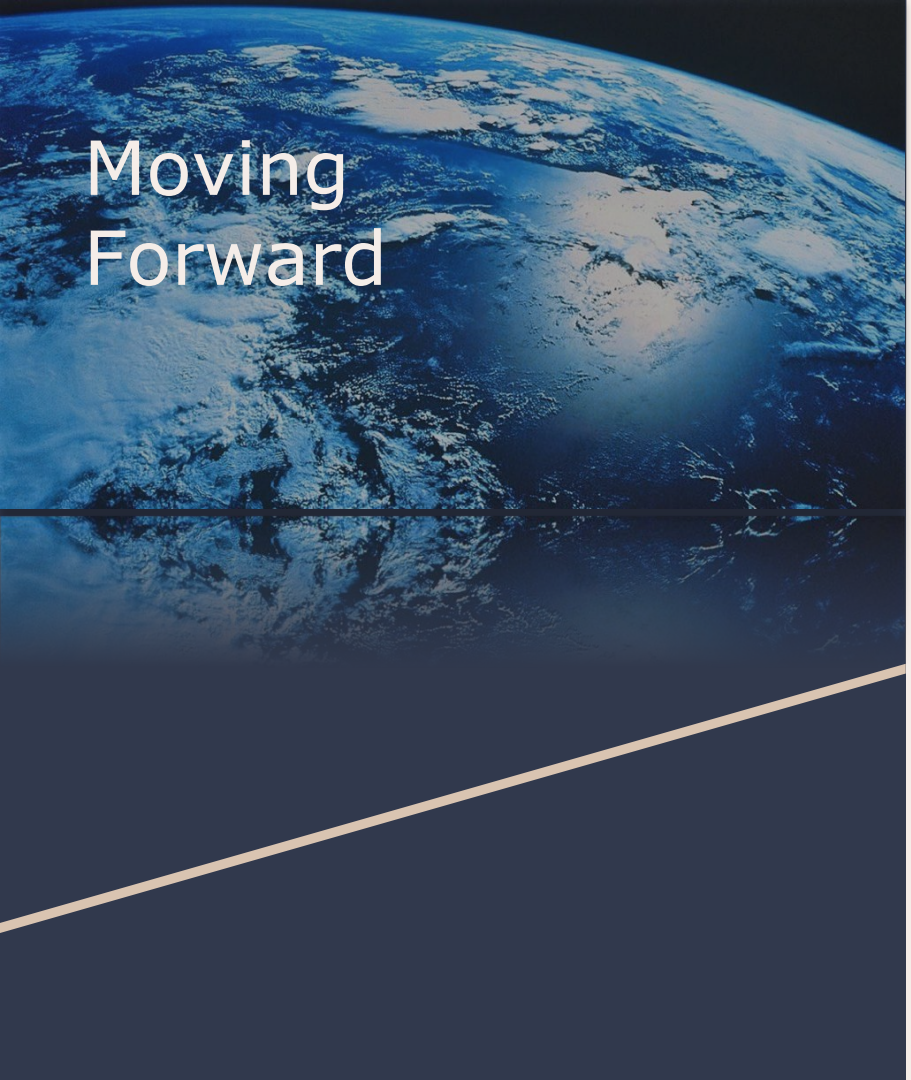
Conclusion

Production Tests

Inputs & Validations

Performance & Analysis Tools

Performance Characteristics



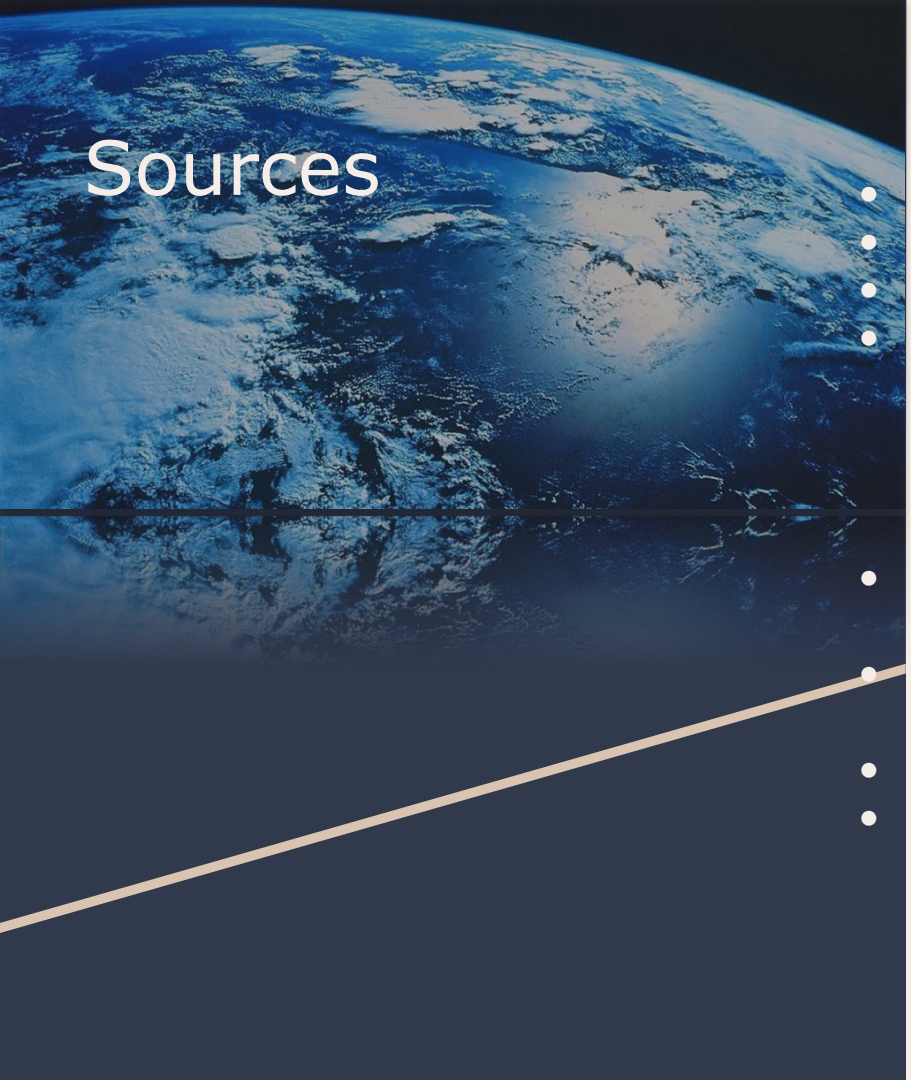
Moving Forward

What's Next

Working with E3SM Developers

Performance Analysis

Identify bottlenecks revealed with profiling



Sources

INFORMATION

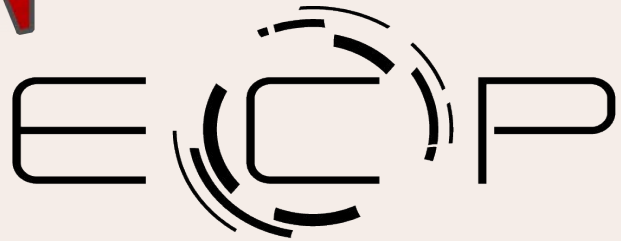
https://github.com/hpc/pavilion2
https://e3sm.org
https://esgf-node.llnl.gov/projects/e3sm/
https://www.splunk.com/

ART

https://www.pinclipart.com/pindetail/wxJwh_https-openclipart-org-http-goo-gl-zvscfc-cartoon/
https://www.nationalobserver.com/2019/02/06/news/2018-was-worlds-fourth-hottest-year-record
http://aharchaou.com/on-global-warming/
https://meetings.pices.int/publications/presentations/2018-Climate-Change/S16-1400-Ringler_R.pdf

Los Alamos
NATIONAL LABORATORY
EST. 1943

Acknowledgements



Questions

The Test Definition

Photo by USGS on Unsplash

The Test Definition

```
base:
    permute_on:   comp_mpi
    subtitle:     "{{comp_mpi.name}}"
    scheduler:    raw

    variables:
        comp_mpi:
            - { name: "intel_ompi",    compiler: "intel", mpilib: "openmpi" }
            - { name: "intel_impi",    compiler: "intel", mpilib: "impi"    }
            - { name: "intel_mvapich", compiler: "intel", mpilib: "mvapich" }
            - { name: "gnu_ompi",      compiler: "gnu",   mpilib: "openmpi" }
            - { name: "gnu_mvapich",   compiler: "gnu",   mpilib: "mvapich" }

        components: [CPL, ATM, LND, ROF, ICE, OCN, GLC, WAV]

    build:
        env:
        source_path:
        modules:
        cmds:

    run:
        env:
        cmds:

    result_parse:
        regex:
            mpilib:
```

Photo by USGS on Unsplash

The Test Definition

Pavilion Test Suite

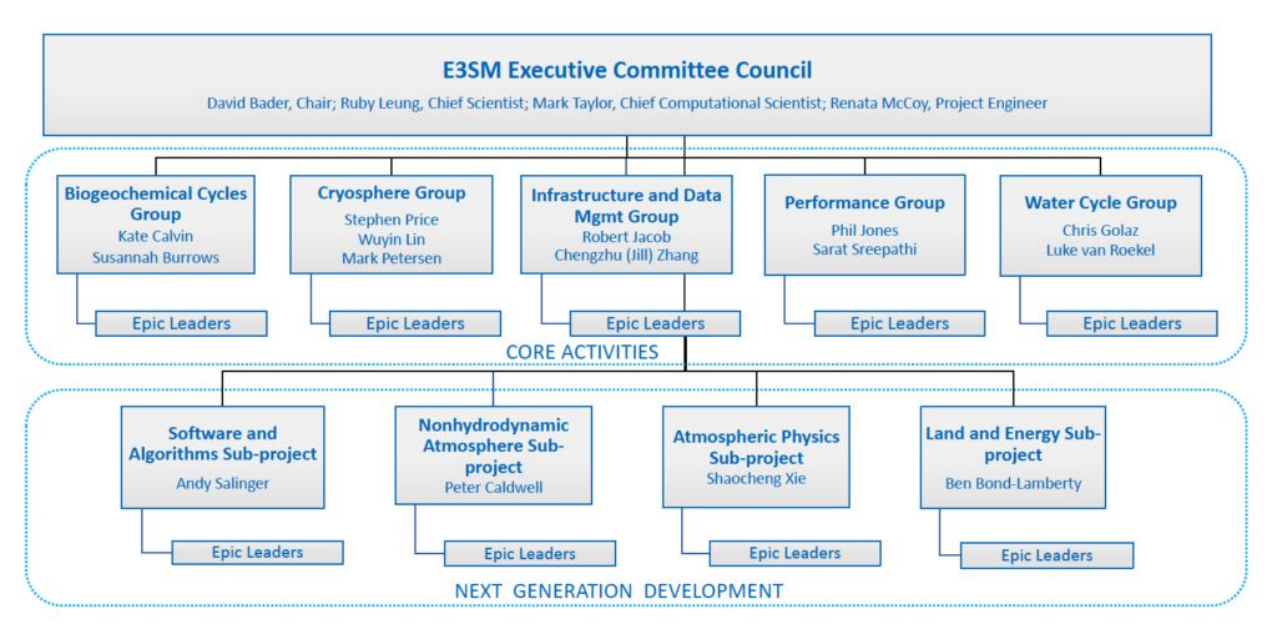
```
result_parse:
  regex:
    tot_run_seconds:
      files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
      regex: 'TOT Run Time:\s+((?:\d+\.)?\d+) seconds'
      action: 'store'
    cpl_run_seconds:
      files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
      regex: 'CPL Run Time:\s+((?:\d+\.)?\d+) seconds'
      action: 'store'
    atm_run_seconds:
      files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
      regex: 'ATM Run Time:\s+((?:\d+\.)?\d+) seconds'
      action: 'store'
```


base:
permute_on: comp_mpi
subtitle: "{{comp_mpi.name}}"
scheduler: raw
variables:
comp_mpi:
- { name: "intel_ompi", compiler: "intel", mpilib: "openmpi" }
- { name: "intel_impi", compiler: "intel", mpilib: "impi" }
- { name: "intel_mvapich", compiler: "intel", mpilib: "mvapich" }
- { name: "gnu_ompi", compiler: "gnu", mpilib: "openmpi" }
- { name: "gnu_mvapich", compiler: "gnu", mpilib: "mvapich" }
components: CPL ATM LND ROF ICE OCN GLC WAV
build:
env: # Environment Variables
source_path: # Path to Source Code
modules: # Modules Needed
cmds: # Written into a Build Script
run: # Similar to build section, makes Run Script
env:
cmds:
result_parse: # Parses files for matching strings
regex:
mpilib: # Will store the MPI Library

The Application

E3SM
Energy Exascale
Earth System Model

The Application



JEN PLAYING AROUND


```
variables.comp_mpi:
    - { name: "intel_ompi",
        compiler: "intel",
        mpilib: "openmpi"
      }

    - { name: "intel_impi",
        compiler: "intel",
        mpilib: "impi"
      }

    - { name: "intel_mvapich",
        compiler: "intel",
        mpilib: "mvapich"
      }
```

The Test Definition

Results Parsing

```
tot_run_seconds:
  files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
  regex: 'TOT Run Time:\s+((?:\d+\.)?\d+) seconds'
  action: 'store'
cpl_run_seconds:
  files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
  regex: 'CPL Run Time:\s+((?:\d+\.)?\d+) seconds'
  action: 'store'
atm_run_seconds:
  files: '{{case_root}}/{{e3sm_case}}/timing/e3sm_timing*{{e3sm_case}}*'
  regex: 'ATM Run Time:\s+((?:\d+\.)?\d+) seconds'
  action: 'store'
```

The Test Definition

Took a bash script, ported it to a YAML-based test definition.

Dealt with permission stuff, differences between pavilion work-flow and plain bash-scripts

Talk about process that got me to the code stuff.
Removed repeated code
Easy to change configurations

The Test Definition


```
variables.comp_mpi:
    - { name: "intel_ompi",
        compiler: "intel",
        mpilib: "openmpi"
      }

    - { name: "intel_impi",
        compiler: "intel",
        mpilib: "impi"
      }

    - { name: "intel_mvapich",
        compiler: "intel",
        mpilib: "mvapich"
      }
```

The Test Definition


```
variables.comp_mpi:
  ...
  - { name: "intel_ompi",
        compiler: "intel",
        mpilib: "openmpi"
    }

  - { name: "intel_impi",
        compiler: "intel",
        mpilib: "impi"
    }

  - { name: "intel_mvapich",
        compiler: "intel",
        mpilib: "mvapich"
    }
...
```

The Test Definition


```
base:
    build:
        cmds:
        ...
        - './create_newcase
            --handle-preexisting-dirs r -v
            -case {{case_root}}/{{e3sm_case}}
            -compiler {{comp_mpi.compiler}}
            -mach {{sys_name}}
            -mpilib {{comp_mpi.mpilib}}
            -project {{project}}
            -compset {{compset}}
            -res {{grid}}
          '
       …
```

The Test Definition

Before Pavilion

```
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_CPL  -val $NPROCS_CPL &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_ATM  -val $NPROCS_ATM &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_LND  -val $NPROCS_LND &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_ROF  -val $NPROCS_ROF &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_ICE  -val $NPROCS_ICE &>>$DEBUGOUT ); then die
                                                      "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_OCN  -val $NPROCS_OCN &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_GLC  -val $NPROCS_GLC &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
if ! $(./xmlchange -file env_mach_pes.xml  -id NTASKS_WAV  -val $NPROCS_WAV &>>$DEBUGOUT ); then
                                                    die "xmlchange failed" ; fi
```

Under Pavilion

```
- 'set -e'
- "[~./xmlchange -file env_mach_pes.xml -id NTASKS_{{e3sm_tests}} -val ${NPROCS_{{e3sm_tests}}}\n~]"
```

Clipart