Performative Benchmarking of Unidata MetPy with ASV

Jaye Norman 7/23/2025

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Why should we benchmark?

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

C++ification

- Linfeng has created C++ to help MetPy run faster through bottlenecks like CAPE calculation
- We needed a way to quantify his and future changes

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Other Benefits

- Benchmarking also identifies bottlenecks and shows us when merges change the performance of the code

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

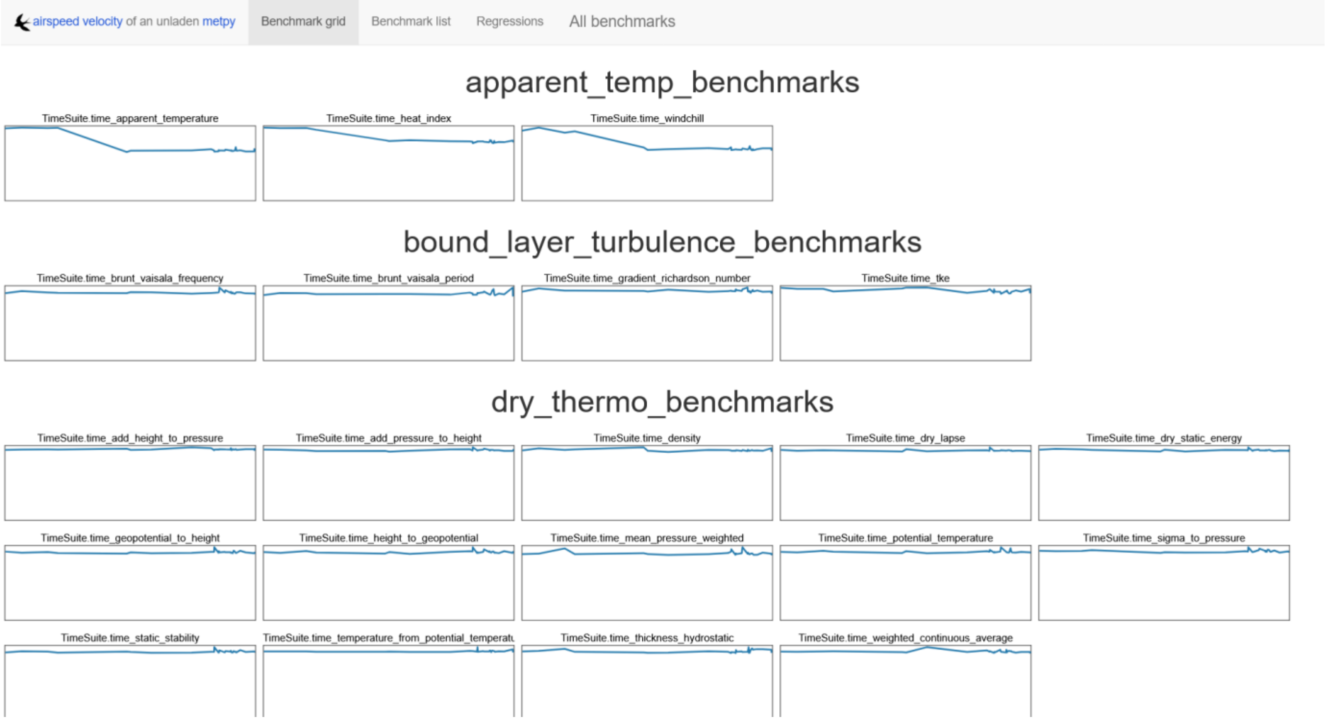
Where can I see the results?

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

unidata.github.io/MetPy-benchmark

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

ASV Webpage



apparent_temp_benchmarks

TimeSuite.time_apparent_temperature
TimeSuite.time_heat_index
TimeSuite.time_windchill

bound_layer_turbulence_benchmarks

TimeSuite.time_brunt_vaisala_frequency
TimeSuite.time_brunt_vaisala_period
TimeSuite.time_gradient_richardson_number
TimeSuite.time_tke

dry_thermo_benchmarks

TimeSuite.time_add_height_to_pressure
TimeSuite.time_add_pressure_to_height
TimeSuite.time_density
TimeSuite.time_dry_lapse
TimeSuite.time_dry_static_energy
TimeSuite.time_geopotential_to_height
TimeSuite.time_height_to_geopotential
TimeSuite.time_mean_pressure_weighted
TimeSuite.time_potential_temperature
TimeSuite.time_sigma_to_pressure
TimeSuite.time_static_stability
TimeSuite.time_temperature_from_potential_temperatu
TimeSuite.time_thickness_hydrostatic
TimeSuite.time_weighted_continuous_average

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

What software did we use?

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

GitHub

- Provides a home for the MetPy source code
- The MetPy repo holds the benchmarking functions and configs for CI/CD
- The MetPy-benchmark repo holds the results of the benchmark runs

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Airspeed Velocity

- ASV is an open-source python benchmarking package
- Creates environments based on historical states of a repo, runs the benchmark functions, and returns them in a pretty html format

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins

- Unidata’s Jenkins instance is used for CI/CD workflows
- Runs on a machine owned by UCAR and ensures that the machine specs are the same between each benchmark run

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Docker

- Within Jenkins, the benchmarks run in a Docker container from a Dockerfile
- Improves consistency between runs and is portable to many devices for local benchmarking

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

GitHub Actions

- The MetPy-benchmark repo has a GHA that uses ASV to generate the html from the results
- Action also deploys files to a static page

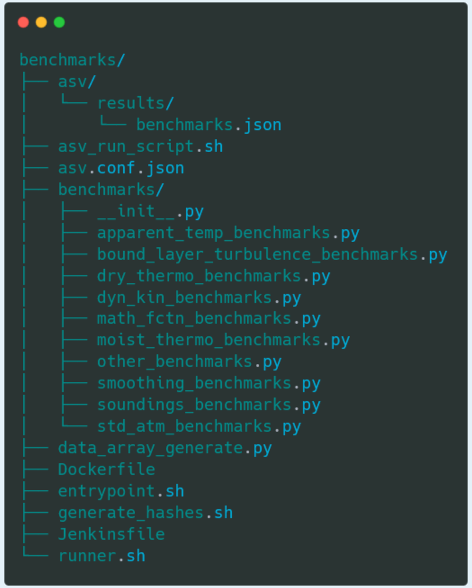
This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

How does benchmarking with ASV work?

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

File Tree

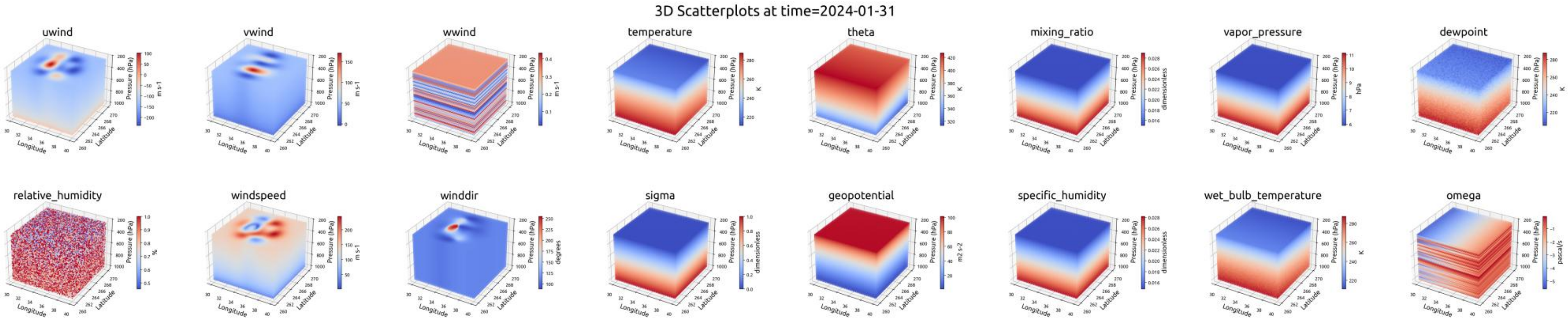
- asv.conf.json: configuration file for ASV
- asv/results: where the results are stored
- benchmarks/.py files: benchmarking snippets



```
benchmarks/
├── asv/
│   └── results/
│       └── benchmarks.json
├── asv_run_script.sh
├── asv.conf.json
├── benchmarks/
│   ├── __init__.py
│   ├── apparent_temp_benchmarks.py
│   ├── bound_layer_turbulence_benchmarks.py
│   ├── dry_thermo_benchmarks.py
│   ├── dyn_kin_benchmarks.py
│   ├── math_fctn_benchmarks.py
│   ├── moist_thermo_benchmarks.py
│   ├── other_benchmarks.py
│   ├── smoothing_benchmarks.py
│   ├── soundings_benchmarks.py
│   └── std_atm_benchmarks.py
├── data_array_generate.py
├── Dockerfile
├── entrypoint.sh
├── generate_hashes.sh
├── Jenkinsfile
└── runner.sh
```

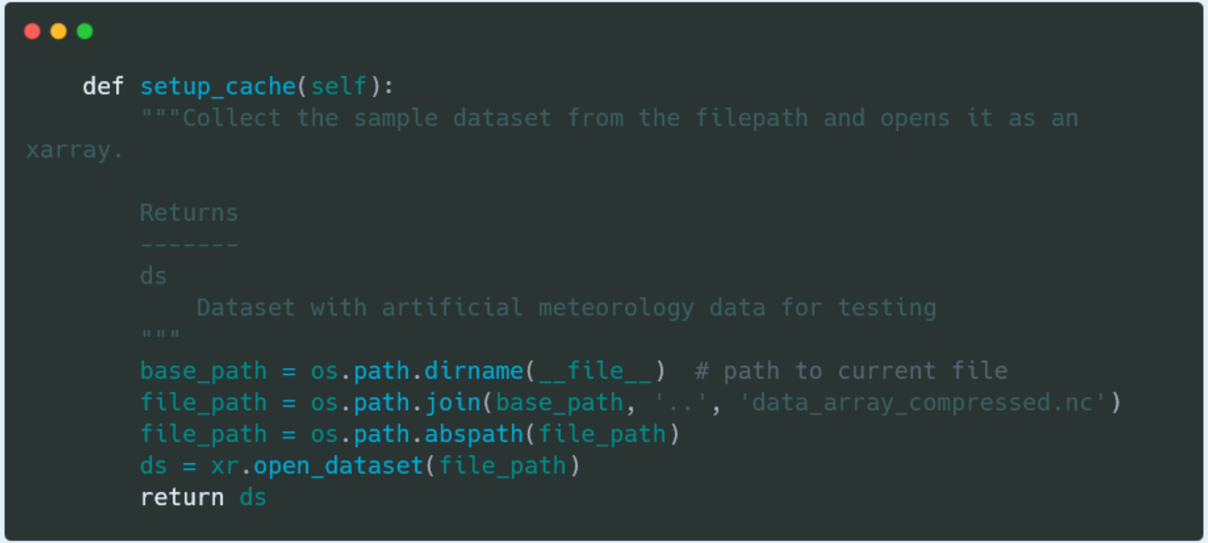
This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Benchmark Dataset


3D Scatterplots at time=2024-01-31

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Benchmark setup_cache

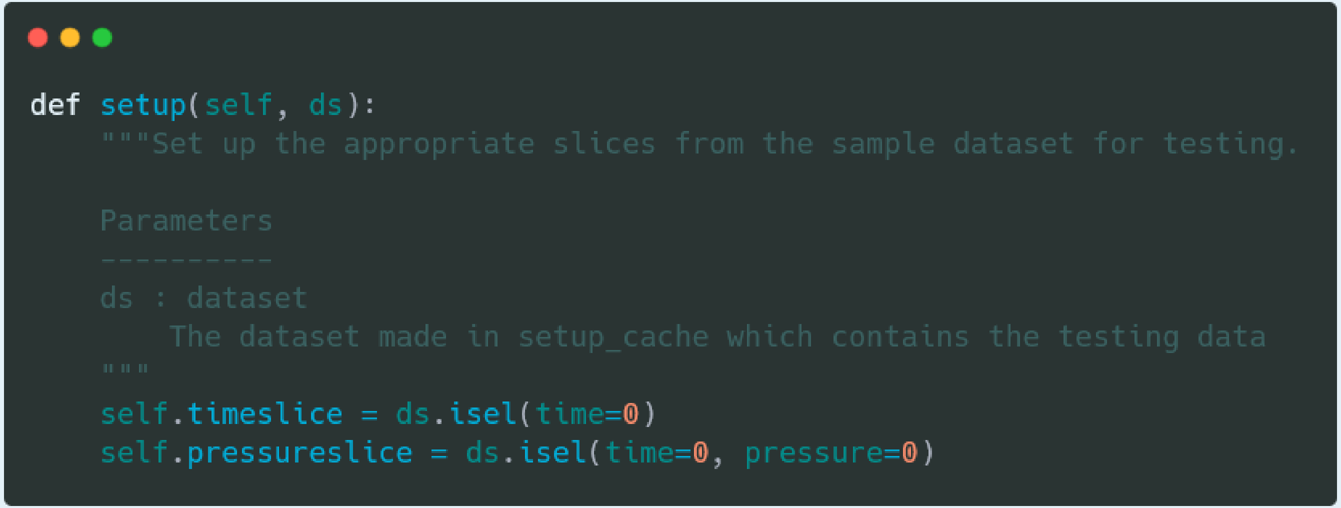


```
    def setup_cache(self):
        """Collect the sample dataset from the filepath and opens it as an
xarray.

        Returns
        -------
        ds
            Dataset with artificial meteorology data for testing
        """
        base_path = os.path.dirname(__file__)  # path to current file
        file_path = os.path.join(base_path, '..', 'data_array_compressed.nc')
        file_path = os.path.abspath(file_path)
        ds = xr.open_dataset(file_path)
        return ds
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Benchmark setup

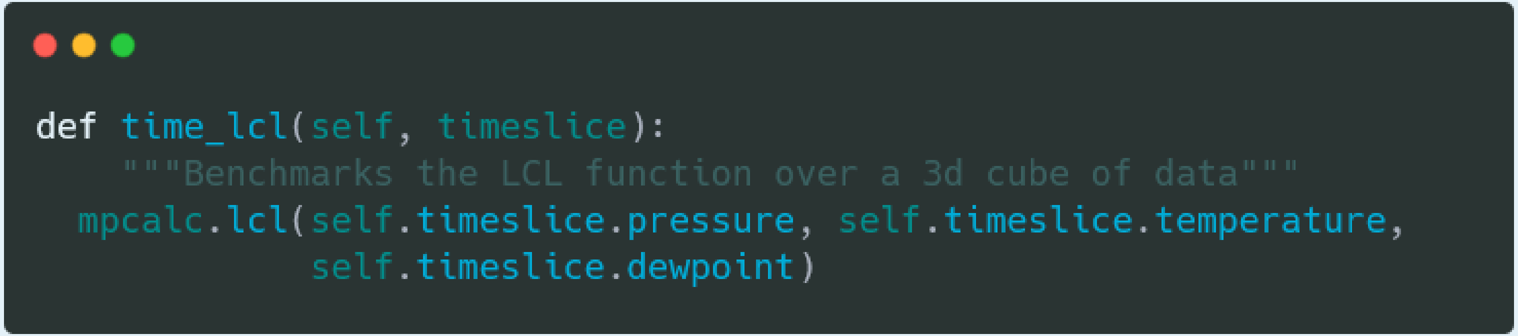


```python
def setup(self, ds):
    """Set up the appropriate slices from the sample dataset for testing.

    Parameters
    ----------
    ds : dataset
        The dataset made in setup_cache which contains the testing data
    """
    self.timeslice = ds.isel(time=0)
    self.pressureslice = ds.isel(time=0, pressure=0)
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Example Benchmark



```
def time_lcl(self, timeslice):
    """Benchmarks the LCL function over a 3d cube of data"""
  mpcalc.lcl(self.timeslice.pressure, self.timeslice.temperature,
             self.timeslice.dewpoint)
```

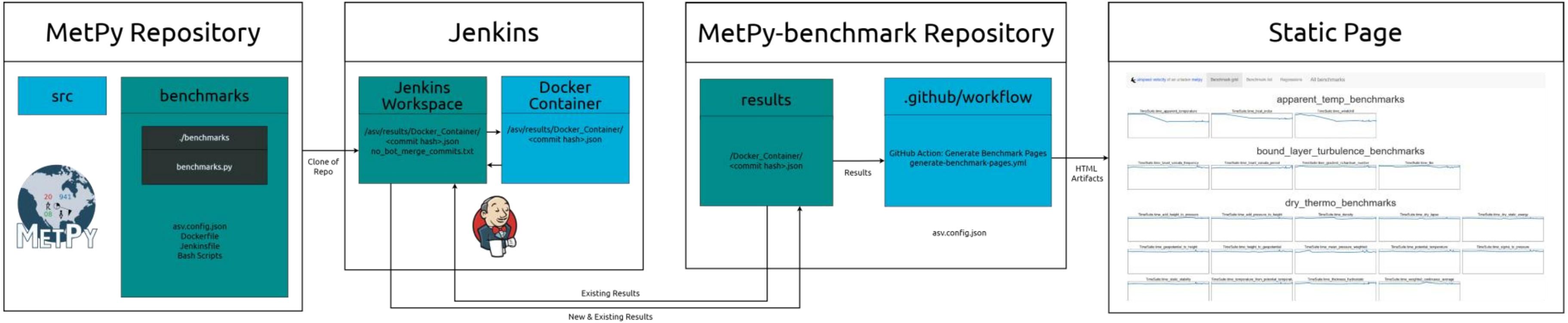
This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

How does the workflow run?

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins Trigger

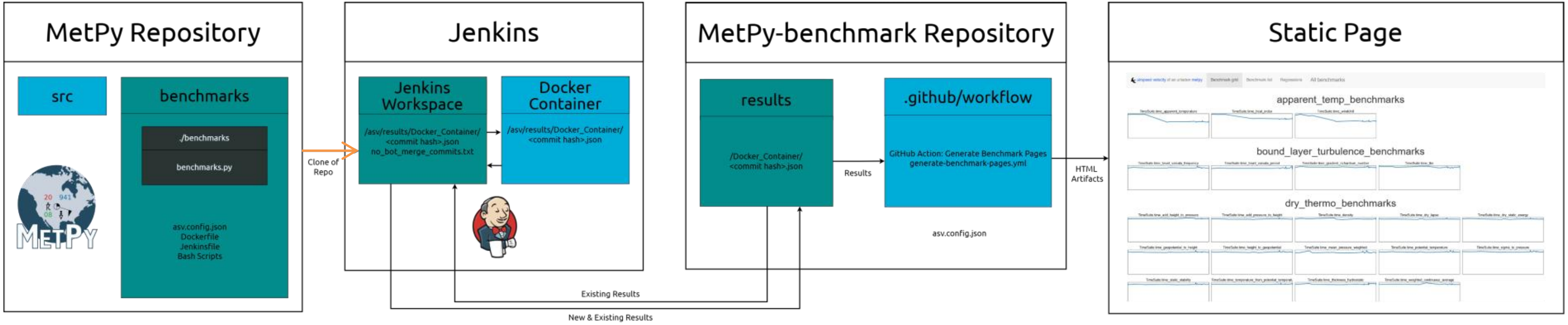
1) Jenkins is triggered Saturday morning



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins Setup

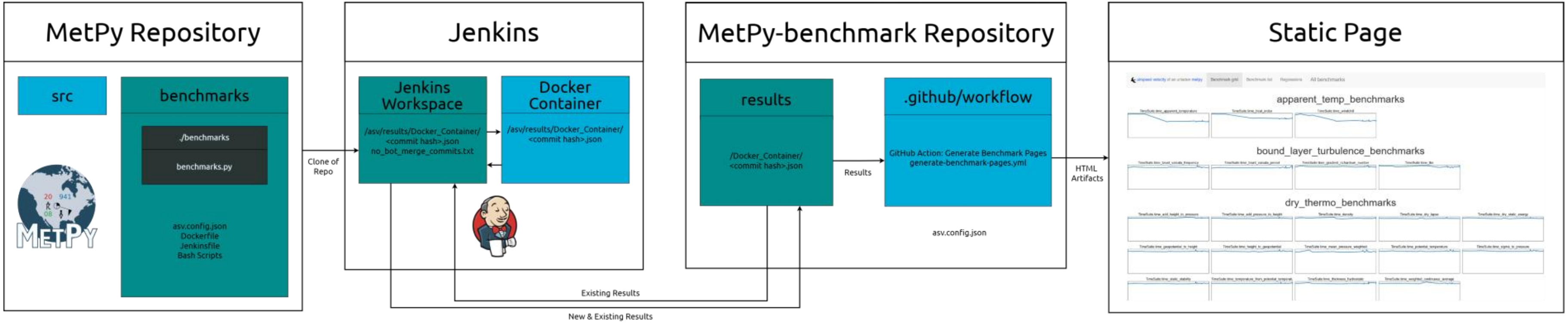
2) Jenkins clones MetPy



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins Setup

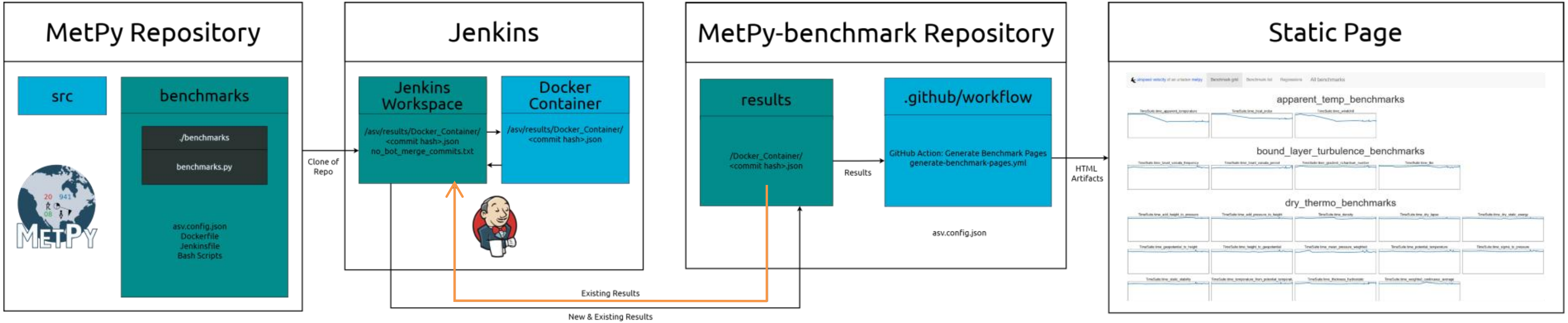
3) Jenkins searches MetPy's history for minor version commits and recent merges



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins Setup

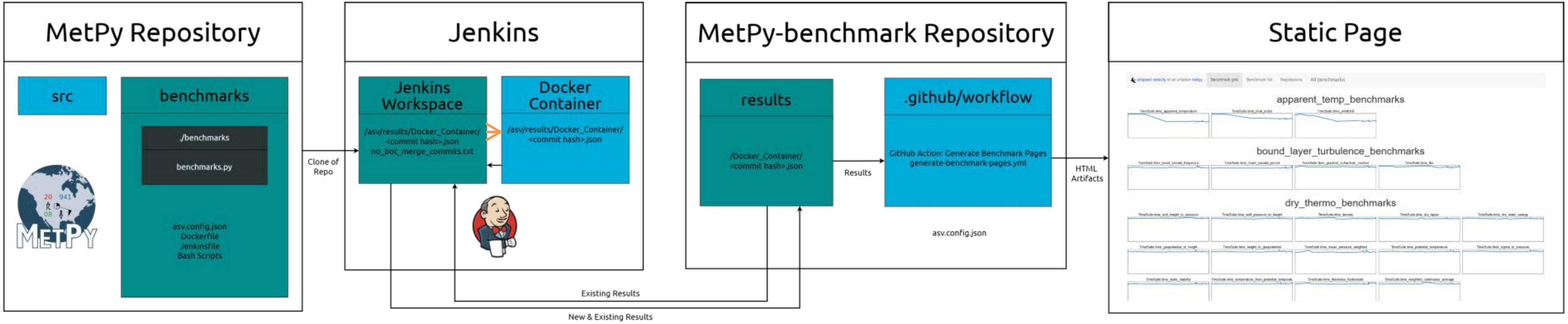
4) From MetPy-benchmark, Jenkins copies the existing results



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins Setup

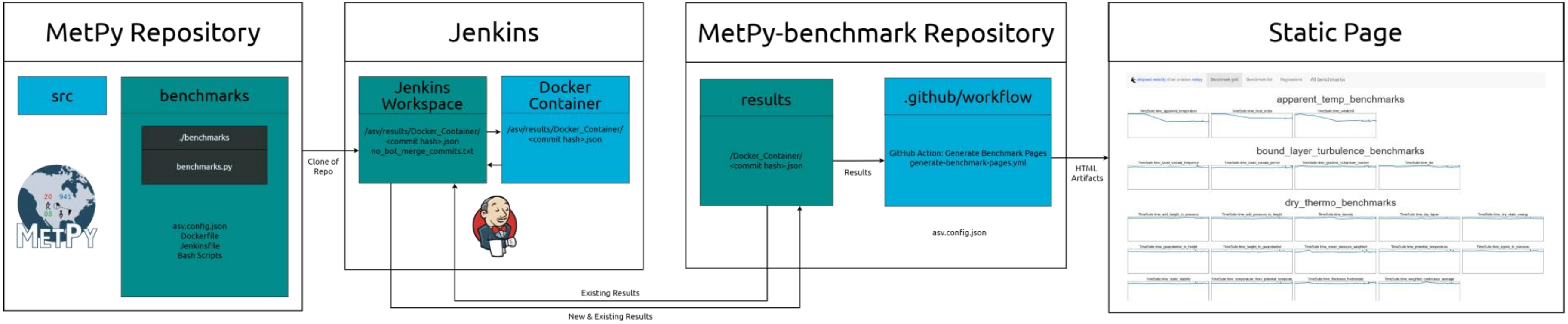
5) Jenkins builds a docker container from the Dockerfile



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Docker runs Benchmarks

1) With a docker run command, the docker file runs benchmarks on the commit file



MetPy Repository

src

benchmarks

./benchmarks

benchmarks.py

asv.config.json
Dockerfile
Jenkinsfile
Bash Scripts

Clone of Repo

Jenkins

Jenkins Workspace

/asv/results/Docker_Container/
<commit hash>.json
no_bot_merge_commits.txt

Docker Container

/asv/results/Docker_Container/
<commit hash>.json

MetPy-benchmark Repository

results

/Docker_Container/
<commit hash>.json

Results

.github/workflow

GitHub Action: Generate Benchmark Pages
generate-benchmark-pages.yml

asv.config.json

HTML Artifacts

Existing Results

New & Existing Results

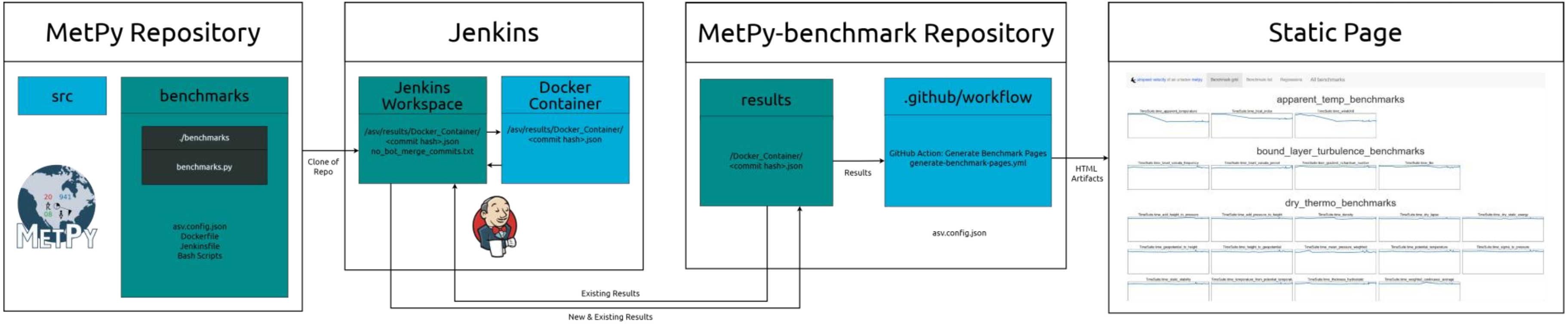
Static Page

airspeed velocity of an unladen metpy
Benchmark grid
Benchmark list
Regressions
All benchmarks
apparent_temp_benchmarks
TimeSuite.time_apparent_temperature
TimeSuite.time_heat_index
TimeSuite.time_windchill
bound_layer_turbulence_benchmarks
TimeSuite.time_brunt_vaisala_frequency
TimeSuite.time_brunt_vaisala_period
TimeSuite.time_gradient_richardson_number
TimeSuite.time_tke
dry_thermo_benchmarks
TimeSuite.time_add_height_to_pressure
TimeSuite.time_add_pressure_to_height
TimeSuite.time_density
TimeSuite.time_dry_lapse
TimeSuite.time_dry_static_energy
TimeSuite.time_geopotential_to_height
TimeSuite.time_height_to_geopotential
TimeSuite.time_mean_pressure_weighted
TimeSuite.time_potential_temperature
TimeSuite.time_sigma_to_pressure
TimeSuite.time_static_stability
TimeSuite.time_temperature_from_potential_temperat
TimeSuite.time_thickness_hydrostatic
TimeSuite.time_weighted_continuous_average

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Docker runs Benchmarks

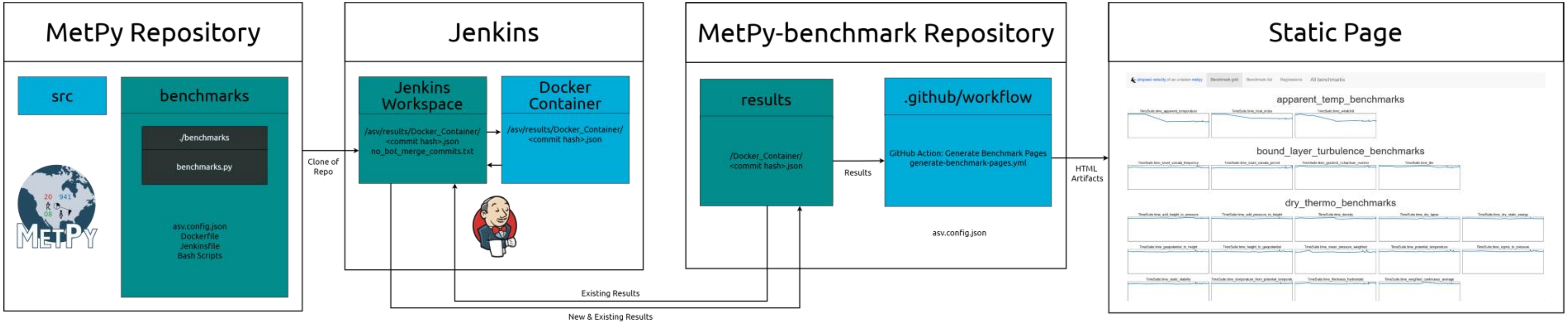
2) If there are already successful results for a certain commit, the benchmarks are skipped



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Docker runs Benchmarks

3) If there are not, like for a new commit, the benchmarks are run for this commit



MetPy Repository

src

benchmarks

./benchmarks

benchmarks.py

asv.config.json
Dockerfile
Jenkinsfile
Bash Scripts

Clone of Repo

Jenkins

Jenkins Workspace

/asv/results/Docker_Container/<commit hash>.json
no_bot_merge_commits.txt

Docker Container

/asv/results/Docker_Container/<commit hash>.json

MetPy-benchmark Repository

results

/Docker_Container/<commit hash>.json

Results

.github/workflow

GitHub Action: Generate Benchmark Pages
generate-benchmark-pages.yml

asv.config.json

HTML Artifacts

Existing Results

New & Existing Results

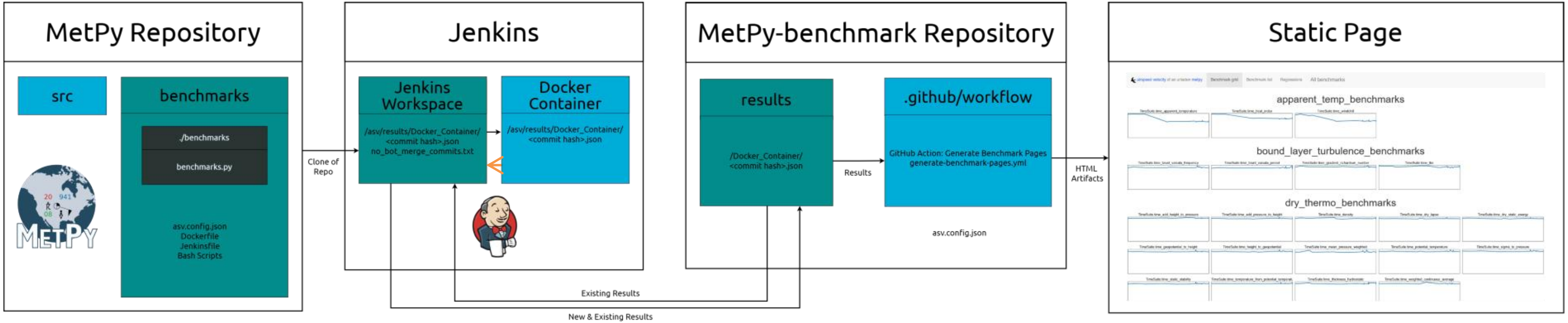
Static Page

airspeed velocity of an unladen metpy
Benchmark grid
Benchmark list
Regressions
All benchmarks
apparent_temp_benchmarks
TimeSuite.time_apparent_temperature
TimeSuite.time_heat_index
TimeSuite.time_windchill
bound_layer_turbulence_benchmarks
TimeSuite.time_brunt_vaisala_frequency
TimeSuite.time_brunt_vaisala_period
TimeSuite.time_gradient_richardson_number
TimeSuite.time_tke
dry_thermo_benchmarks
TimeSuite.time_add_height_to_pressure
TimeSuite.time_add_pressure_to_height
TimeSuite.time_density
TimeSuite.time_dry_lapse
TimeSuite.time_dry_static_energy
TimeSuite.time_geopotential_to_height
TimeSuite.time_height_to_geopotential
TimeSuite.time_mean_pressure_weighted
TimeSuite.time_potential_temperature
TimeSuite.time_sigma_to_pressure
TimeSuite.time_static_stability
TimeSuite.time_temperature_from_potential_temperat
TimeSuite.time_thickness_hydrostatic
TimeSuite.time_weighted_continuous_average

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins saves results

1) The docker container terminates when it’s finished



MetPy Repository

src

benchmarks

./benchmarks

benchmarks.py

asv.config.json
Dockerfile
Jenkinsfile
Bash Scripts

Clone of Repo

Jenkins

Jenkins Workspace

/asv/results/Docker_Container/
<commit hash>.json
no_bot_merge_commits.txt

Docker Container

/asv/results/Docker_Container/
<commit hash>.json

MetPy-benchmark Repository

results

/Docker_Container/
<commit hash>.json

Results

.github/workflow

GitHub Action: Generate Benchmark Pages
generate-benchmark-pages.yml

asv.config.json

HTML Artifacts

Static Page

apparent_temp_benchmarks

bound_layer_turbulence_benchmarks

dry_thermo_benchmarks

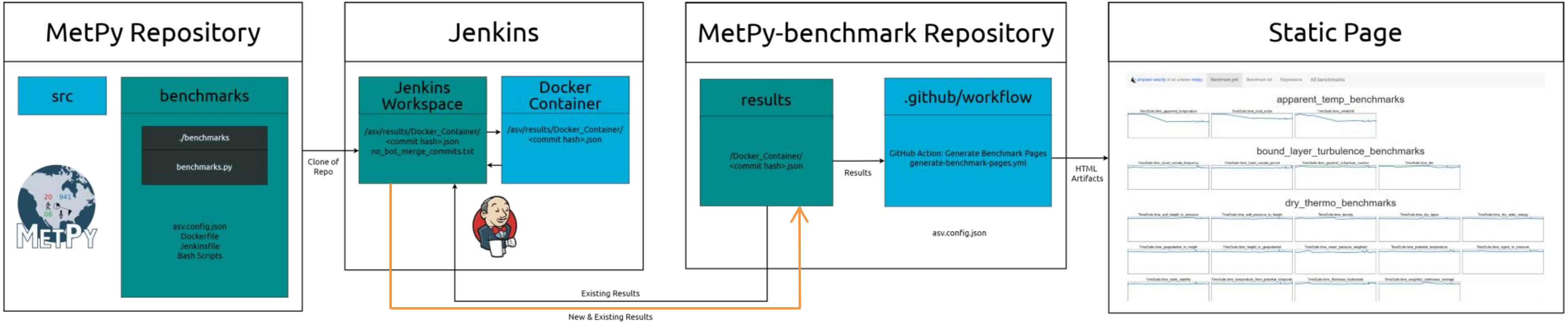
Existing Results

New & Existing Results

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Jenkins saves results

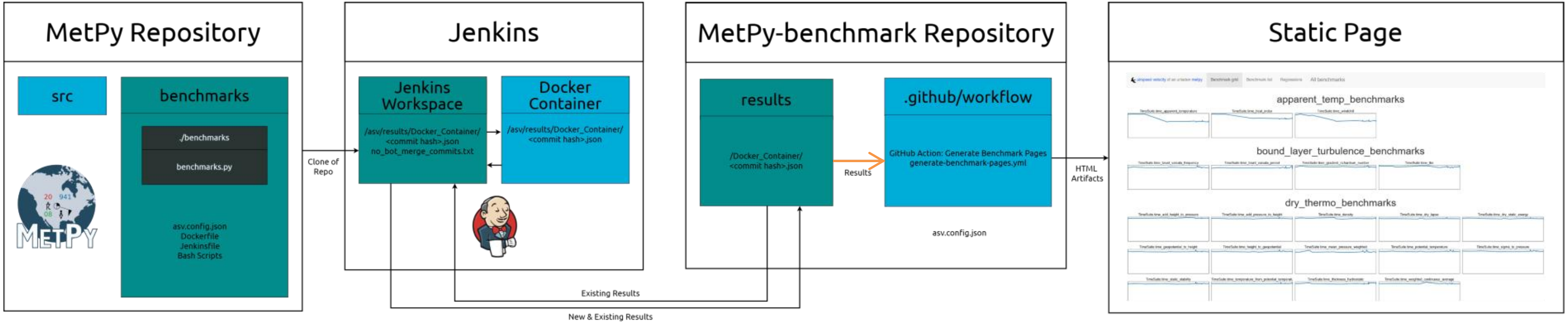
2) Jenkins pushes the results, old and new, to the Metpy-benchmark repository



This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Metpy-benchmark Deployment

1) Upon push to Metpy-benchmark, a GitHub Action creates the html for the static page



MetPy Repository

src

benchmarks

./benchmarks

benchmarks.py

asv.config.json
Dockerfile
Jenkinsfile
Bash Scripts

Clone of Repo

Jenkins

Jenkins Workspace

/asv/results/Docker_Container/<commit hash>.json
no_bot_merge_commits.txt

Docker Container

/asv/results/Docker_Container/<commit hash>.json

Existing Results

New & Existing Results

MetPy-benchmark Repository

results

/Docker_Container/<commit hash>.json

Results

.github/workflow

GitHub Action: Generate Benchmark Pages
generate-benchmark-pages.yml

asv.config.json

HTML Artifacts

Static Page

apparent_temp_benchmarks

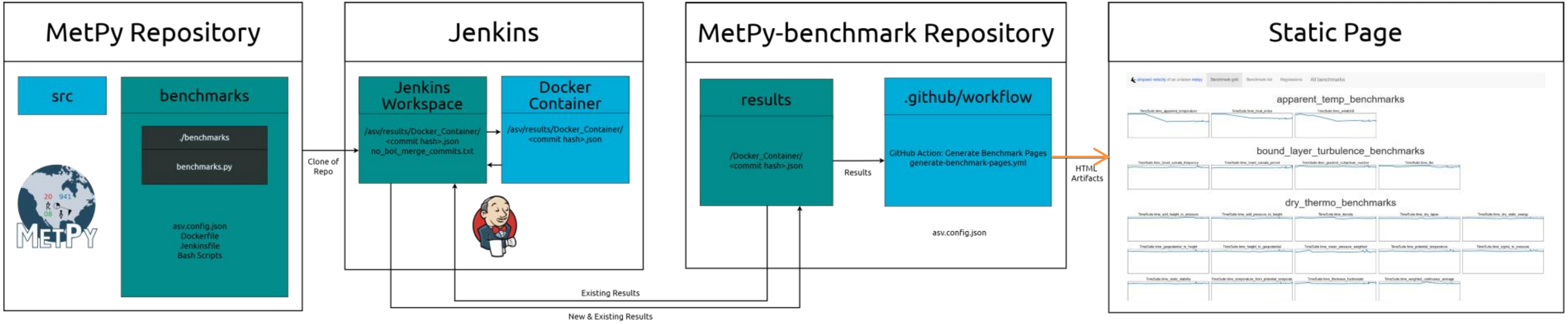
bound_layer_turbulence_benchmarks

dry_thermo_benchmarks

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Metpy-benchmark Deployment

2) The GitHub action deploys the html to the static page


MetPy Repository
src
benchmarks
./benchmarks
benchmarks.py
asv.config.json
Dockerfile
Jenkinsfile
Bash Scripts
20 941
08
METPY
Clone of Repo
Jenkins
Jenkins Workspace
/asv/results/Docker_Container/<commit hash>.json
no_bot_merge_commits.txt
Docker Container
/asv/results/Docker_Container/<commit hash>.json
MetPy-benchmark Repository
results
/Docker_Container/<commit hash>.json
Results
.github/workflow
GitHub Action: Generate Benchmark Pages
generate-benchmark-pages.yml
asv.config.json
HTML Artifacts
Existing Results
New & Existing Results
Static Page
airspeed velocity of an unladen metpy
Benchmark grid
Benchmark list
Regressions
All benchmarks
apparent_temp_benchmarks
TimeSuite.time_apparent_temperature
TimeSuite.time_heat_index
TimeSuite.time_windchill
bound_layer_turbulence_benchmarks
TimeSuite.time_brunt_vaisala_frequency
TimeSuite.time_brunt_vaisala_period
TimeSuite.time_gradient_richardson_number
TimeSuite.time_tke
dry_thermo_benchmarks
TimeSuite.time_add_height_to_pressure
TimeSuite.time_add_pressure_to_height
TimeSuite.time_density
TimeSuite.time_dry_lapse
TimeSuite.time_dry_static_energy
TimeSuite.time_geopotential_to_height
TimeSuite.time_height_to_geopotential
TimeSuite.time_mean_pressure_weighted
TimeSuite.time_potential_temperature
TimeSuite.time_sigma_to_pressure
TimeSuite.time_static_stability
TimeSuite.time_temperature_from_potential_temperat
TimeSuite.time_thickness_hydrostatic
TimeSuite.time_weighted_continuous_average

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Can we see performance changes before we merge commits?

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Comparative Benchmarking

- Comparative benchmarking is when you compare the performance of two branches
- ASV has a built-in function for this, and when combined with GHA, can do it automatically on pull requests

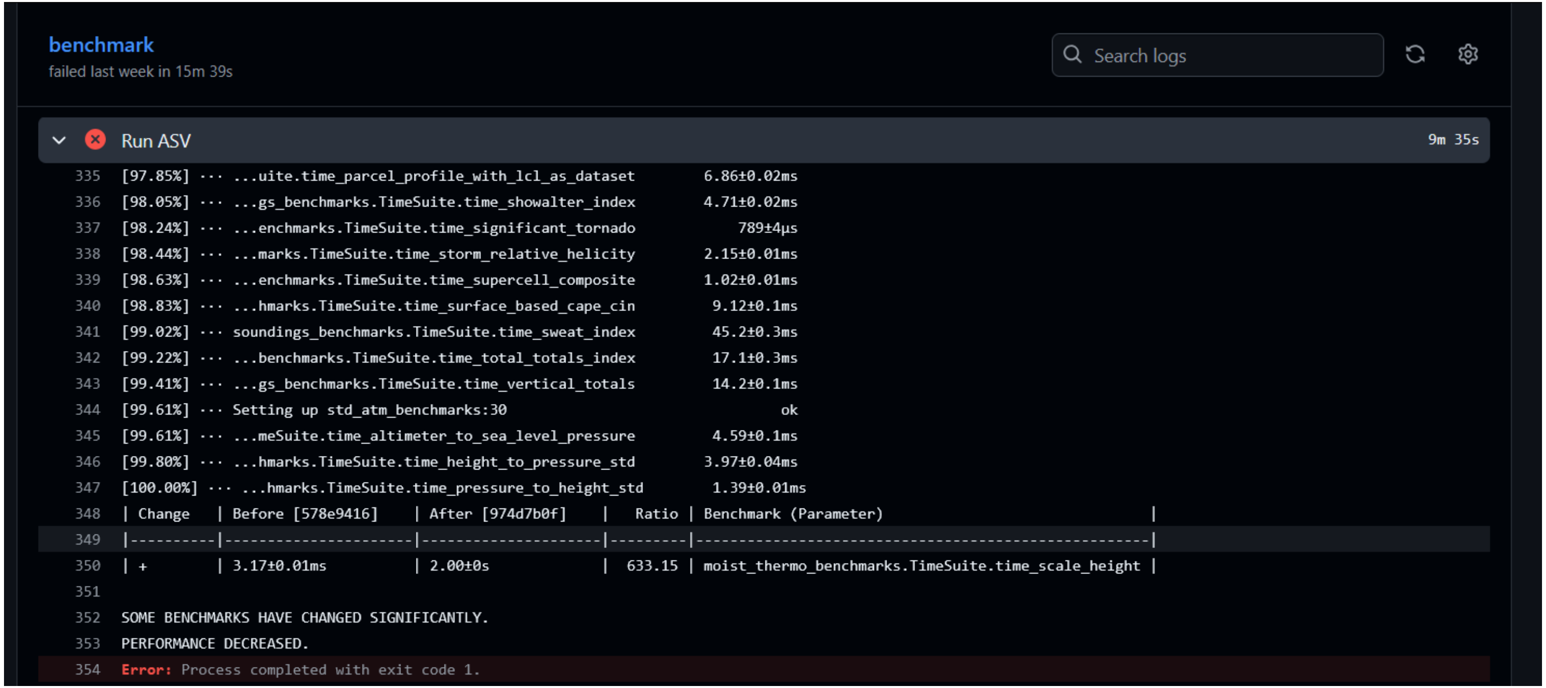
This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Comparative GitHub Action

- We can compare the current main branch to the pull request branch using their commit hashes if the PR is labelled.
- Currently a failure occurs when a benchmark is 10% slower, but this is customizable

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Example GitHub Action



```
[97.85%] ··· ...uite.time_parcel_profile_with_lcl_as_dataset        6.86±0.02ms
[98.05%] ··· ...gs_benchmarks.TimeSuite.time_showalter_index        4.71±0.02ms
[98.24%] ··· ...enchmarks.TimeSuite.time_significant_tornado            789±4μs
[98.44%] ··· ...marks.TimeSuite.time_storm_relative_helicity        2.15±0.01ms
[98.63%] ··· ...enchmarks.TimeSuite.time_supercell_composite        1.02±0.01ms
[98.83%] ··· ...hmarks.TimeSuite.time_surface_based_cape_cin         9.12±0.1ms
[99.02%] ··· soundings_benchmarks.TimeSuite.time_sweat_index         45.2±0.3ms
[99.22%] ··· ...benchmarks.TimeSuite.time_total_totals_index         17.1±0.3ms
[99.41%] ··· ...gs_benchmarks.TimeSuite.time_vertical_totals         14.2±0.1ms
[99.61%] ··· Setting up std_atm_benchmarks:30                                ok
[99.61%] ··· ...meSuite.time_altimeter_to_sea_level_pressure         4.59±0.1ms
[99.80%] ··· ...hmarks.TimeSuite.time_height_to_pressure_std        3.97±0.04ms
[100.00%] ··· ...hmarks.TimeSuite.time_pressure_to_height_std        1.39±0.01ms
| Change   | Before [578e9416]    | After [974d7b0f]    |   Ratio | Benchmark (Parameter)                                 |
|----------|----------------------|---------------------|---------|-------------------------------------------------------|
| +        | 3.17±0.01ms          | 2.00±0s             |  633.15 | moist_thermo_benchmarks.TimeSuite.time_scale_height |

SOME BENCHMARKS HAVE CHANGED SIGNIFICANTLY.
PERFORMANCE DECREASED.
Error: Process completed with exit code 1.
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

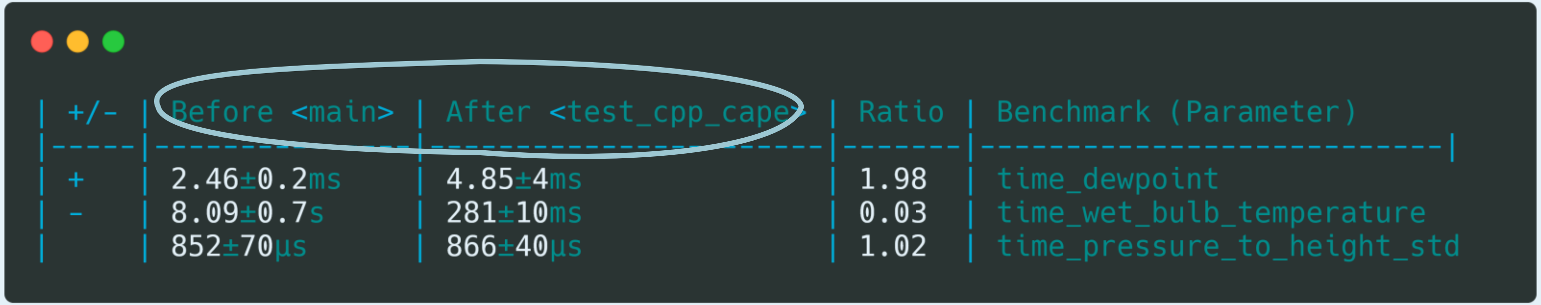
Local Comparisons

- You can also locally run a comparison assuming you have an untouched local main branch and have ASV installed
- This allows you to see if your changes are working as you anticipate

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Example Local Comparison

- This compares main and test_cpp_cape

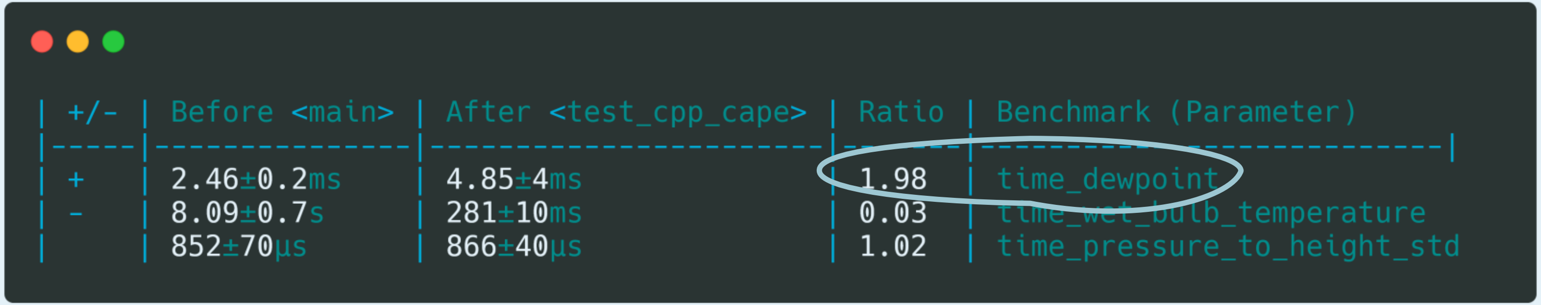


```
| +/- | Before <main> | After <test_cpp_cape> | Ratio | Benchmark (Parameter)
|-----|---------------|-----------------------|-------|---------------------------|
| +   | 2.46±0.2ms    | 4.85±4ms              | 1.98  | time_dewpoint
| -   | 8.09±0.7s     | 281±10ms              | 0.03  | time_wet_bulb_temperature
|     | 852±70µs      | 866±40µs              | 1.02  | time_pressure_to_height_std
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Example Local Comparison

- The dewpoint function is 98% slower

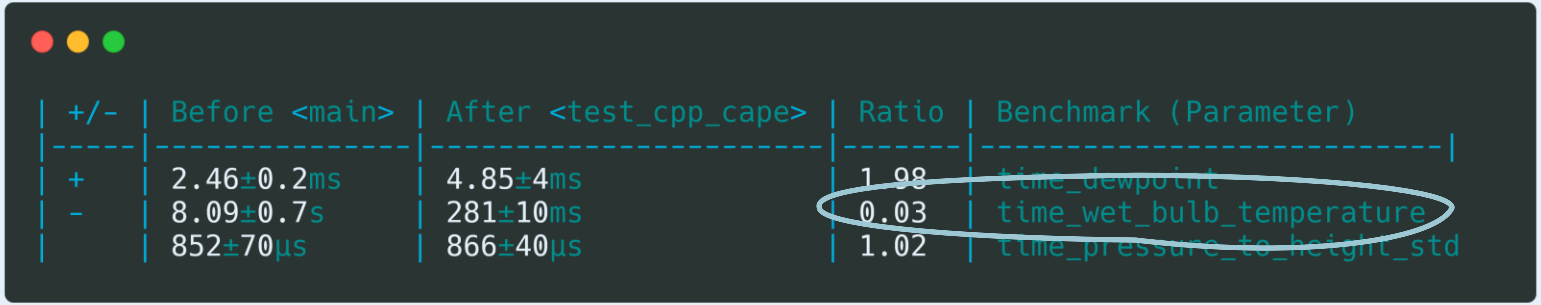


```
| +/- | Before <main> | After <test_cpp_cape> | Ratio | Benchmark (Parameter)
|-----|---------------|-----------------------|-------|-----------------------------|
|  +  | 2.46±0.2ms    | 4.85±4ms              | 1.98  | time_dewpoint
|  -  | 8.09±0.7s     | 281±10ms              | 0.03  | time_wet_bulb_temperature
|     | 852±70µs      | 866±40µs              | 1.02  | time_pressure_to_height_std
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Example Local Comparison

- The wet_bulb function is 97% faster

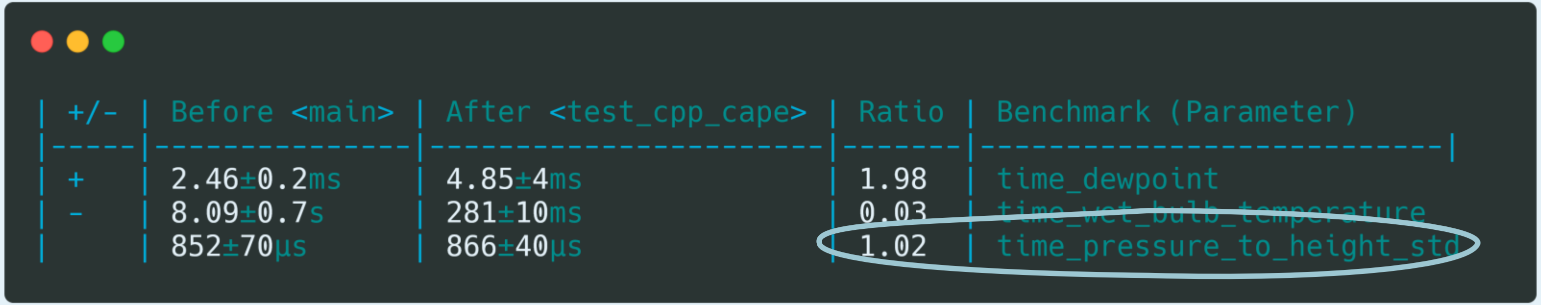


```
| +/- | Before <main> | After <test_cpp_cape> | Ratio | Benchmark (Parameter)
|-----|---------------|-----------------------|-------|---------------------------|
| +   | 2.46±0.2ms    | 4.85±4ms              | 1.98  | time_dewpoint
| -   | 8.09±0.7s     | 281±10ms              | 0.03  | time_wet_bulb_temperature
|     | 852±70µs      | 866±40µs              | 1.02  | time_pressure_to_height_std
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

Example Local Comparison

- The p_to_h function hasn't changed much



```
| +/- | Before <main> | After <test_cpp_cape> | Ratio | Benchmark (Parameter)
|-----|---------------|-----------------------|-------|---------------------------|
|  +  | 2.46±0.2ms    | 4.85±4ms              | 1.98  | time_dewpoint
|  -  | 8.09±0.7s     | 281±10ms              | 0.03  | time_wet_bulb_temperature
|     | 852±70μs      | 866±40μs              | 1.02  | time_pressure_to_height_std
```

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.

NSF Unidata is one of the University Corporation for Atmospheric Research (UCAR)'s Community Programs (UCP), and is funded primarily by the U. S. National Science Foundation (Grant AGS-1901712).

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.