CDP Community Diagnostics Package

Zeshawn Shaheen

ESGF F2F 2017

Thursday, December 7, 2017

ESGF Earth System Grid Federation

Introduction

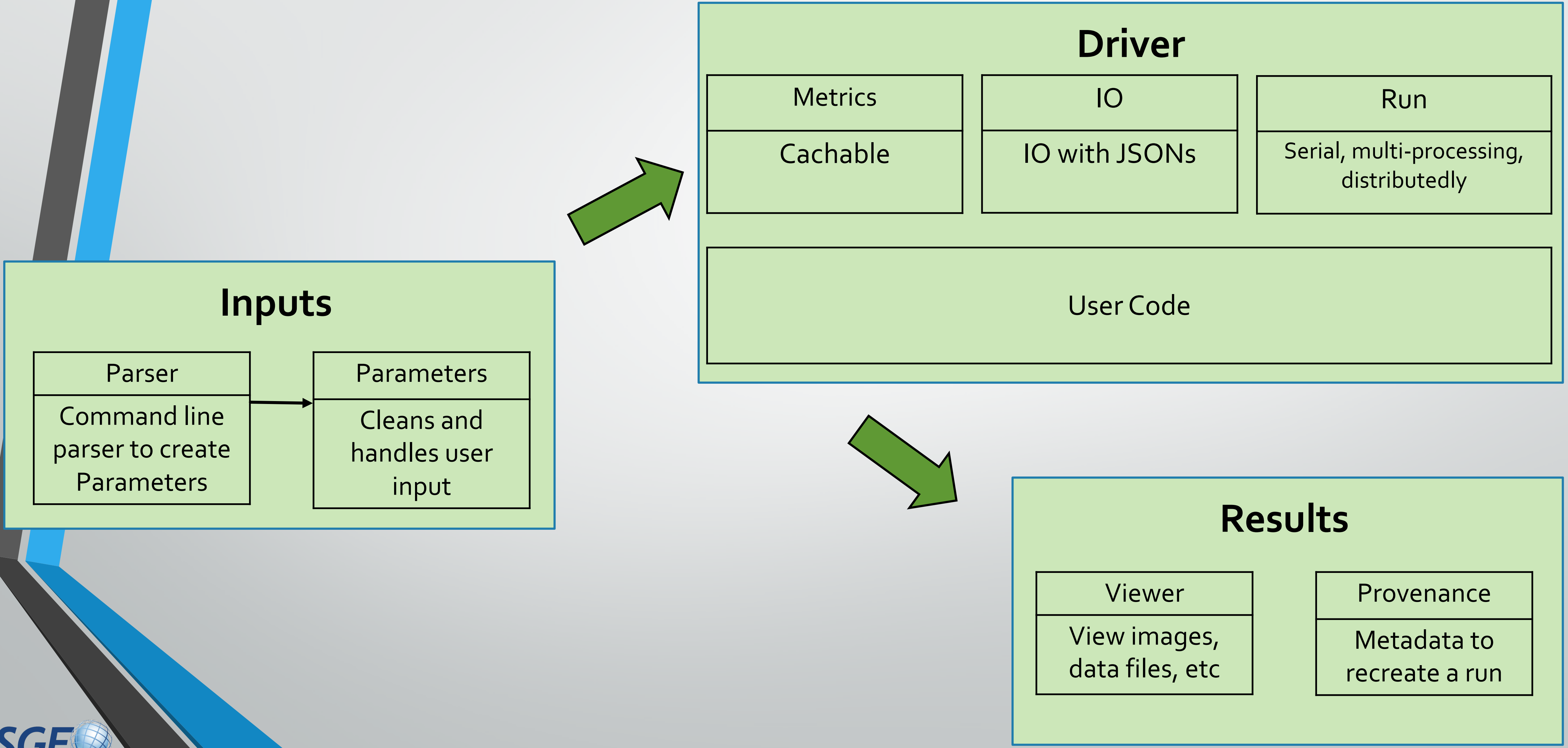
- Framework for managing/modularizing tasks related to diagnostics
  - Handle using input
  - Computing metrics
  - Provenance capture
  - Running diags in parallel (multi-processing, distributedly)
- Basic structure but allows for independence, no dependencies
- Optional support for commonly used tasks
  - Graphing with VCS
  - Viewing results on a webpage created with the CDP viewer API
  - Metric calculations with CDAT (GenUtil, CdUtil)

Problems Solved

- Scientific code is complex, so has a short life
- Scientists don’t have the urgency to implement good software engineering principles
  - Should focus domain-specific work, not viewing results, parallelism, etc.
- Provides a framework for diagnostics to be shared
- Diagnostic packages built with CDP have similar architecture, easy for developers to transition across projects

ESGF Earth System Grid Federation

Design and Architecture


Inputs
Parser
Command line parser to create Parameters
Parameters
Cleans and handles user input
Driver
Metrics
Cachable
IO
IO with JSONs
Run
Serial, multi-processing, distributedly
User Code
Results
Viewer
View images, data files, etc
Provenance
Metadata to recreate a run

Design and Architecture

- Parameters object:
  - Used as input, created from a Python script
  - Encapsulates sanitization of user input
- Parser object: creates Parameters object from the command line
  - Takes raw file: `diags_package.py` **`-p myparams.py`**
  - Individual parameters are command line arguments, ex: `diags_package.py -p myparams.py` **`--seasons ANN`**
- Example parameters script:

```
variables = [‘T’, ‘PRECT’]
regions = [‘global’]
seasons = [‘ANN’, ‘DJF’]
```

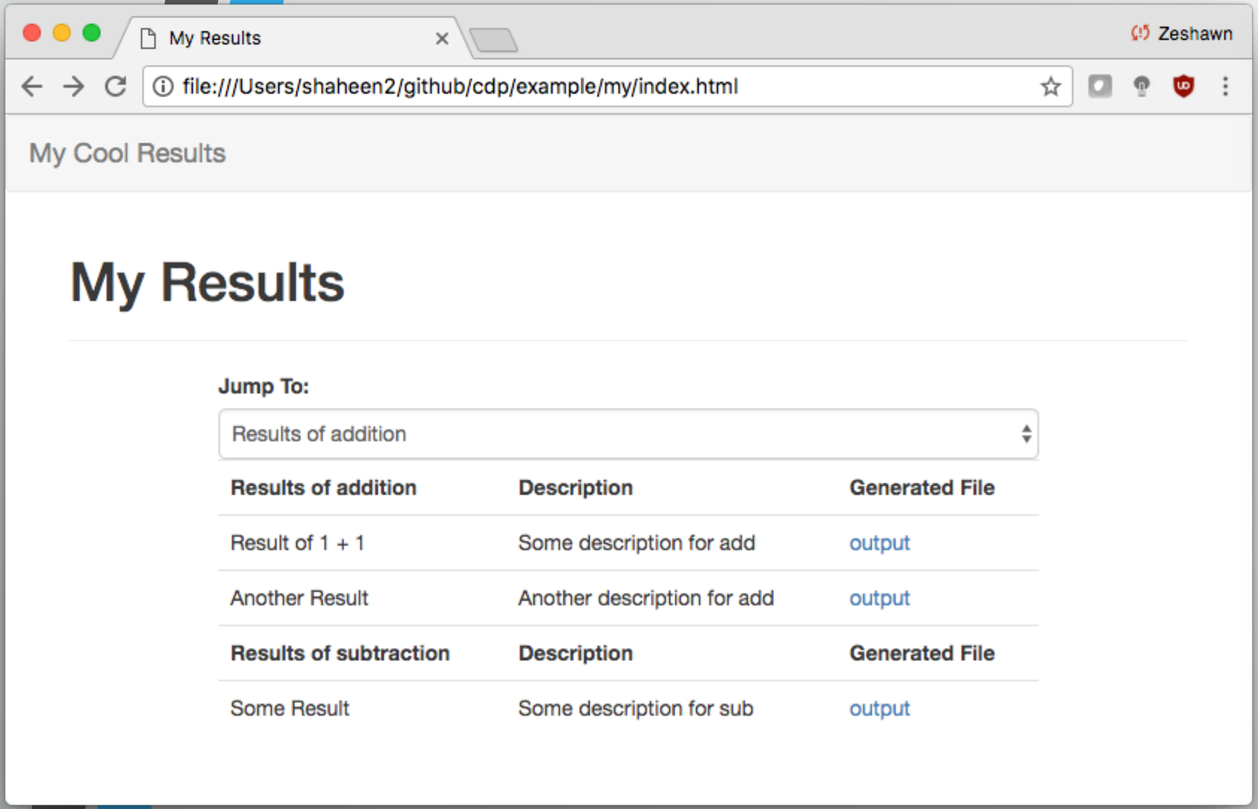
Design and Architecture

- Metrics:
  - Cachable, not in by default in Python 2
  - Single interface to work with Fortran, C, Python code
  - ESGF CWT-based metrics as well
- IO:
  - Handles input/output with JSONs
- Main script (driver), designed for a single run
  - Input: Parameters object
  - Do calculations using metrics, save info with IO, etc
  - Results: View results with CDP viewer, provenance capture through CDP provenance.

Design and Architecture

- Run:
  - Run a driver in serial, or parallel using multi-processing or with distributed computing
  - Data parallelism with different parameters
- Viewer:
  - Easily create an interactive, sharable HTML page
  - In Python, no HTML/Javascript
- Provenance:
  - Create a backup of the Parameters object
  - Log of output, utils for data validation (hashing, etc)

Design and Architecture



My Results — file:///Users/shaheen2/github/cdp/example/my/index.html

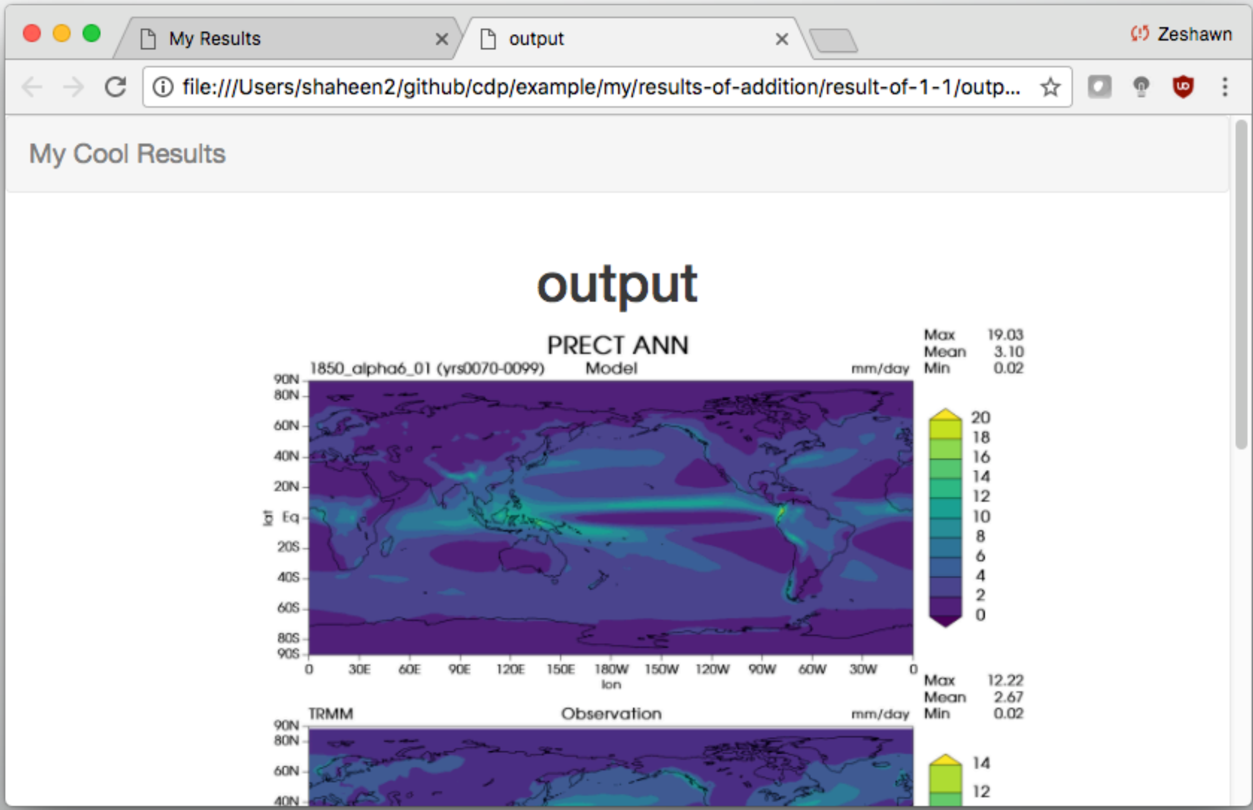
My Cool Results

My Results

Jump To:

Results of addition

Results of addition	**Description**	**Generated File**
Result of 1 + 1	Some description for add	output
Another Result	Another description for add	output
Results of subtraction	**Description**	**Generated File**
Some Result	Some description for sub	output



output — file:///Users/shaheen2/github/cdp/example/my/results-of-addition/result-of-1-1/outp...

My Cool Results

output

- Only 14 lines of code

Design and Architecture

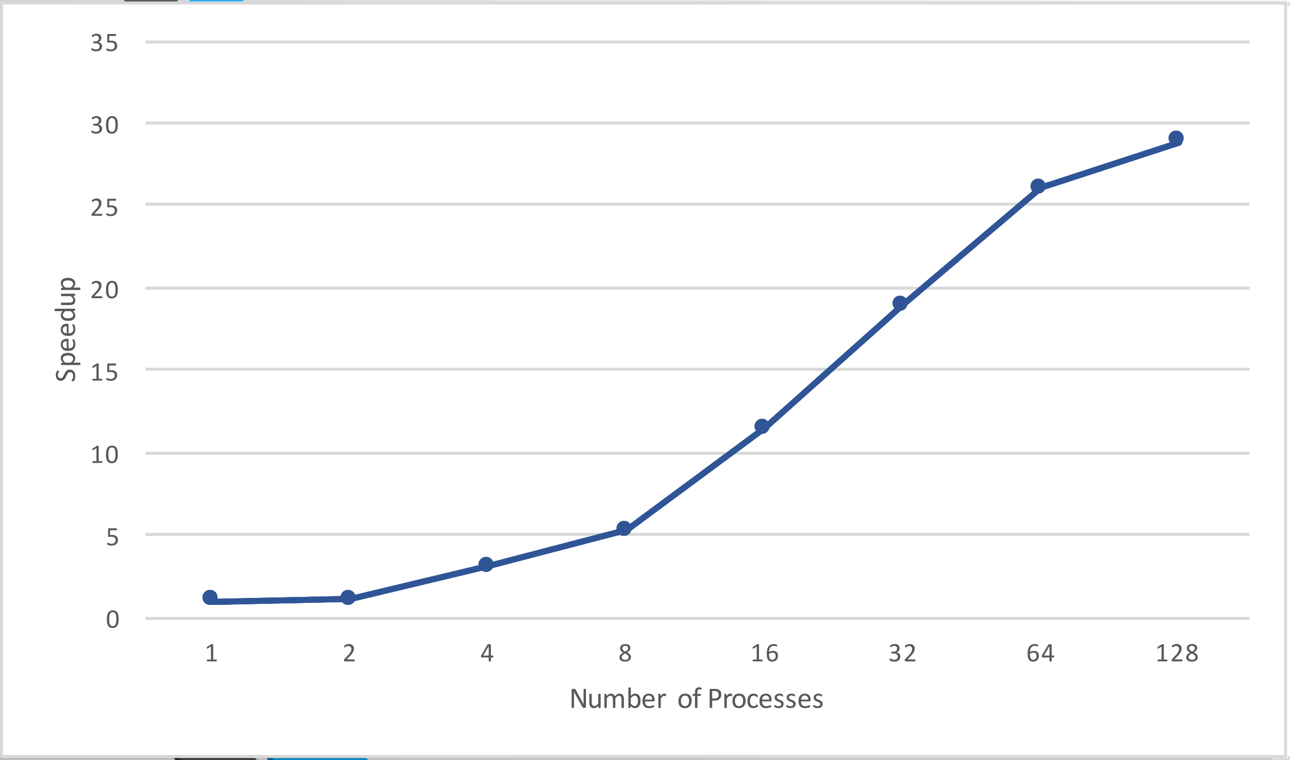
- Data parallelism
  - User can submit multiple runs:
    - `diags_package.py` **`-p myparams1.py myparams2.py`**
    - `diags_package.py` **`-p myparams.cfg`**
  - Compose parameters:
    - `diags_package.py` **`-p myparams.py –d myparams.cfg`**
  - **Transparency**:
    - Each run is a job
    - Same interface to run in serial, parallel, parallel w/ distributed
    - Users don’t need to tailor input to match the type of run
  - CDP CLI
    - Tool for viewing status of, restarting, killing distributed jobs

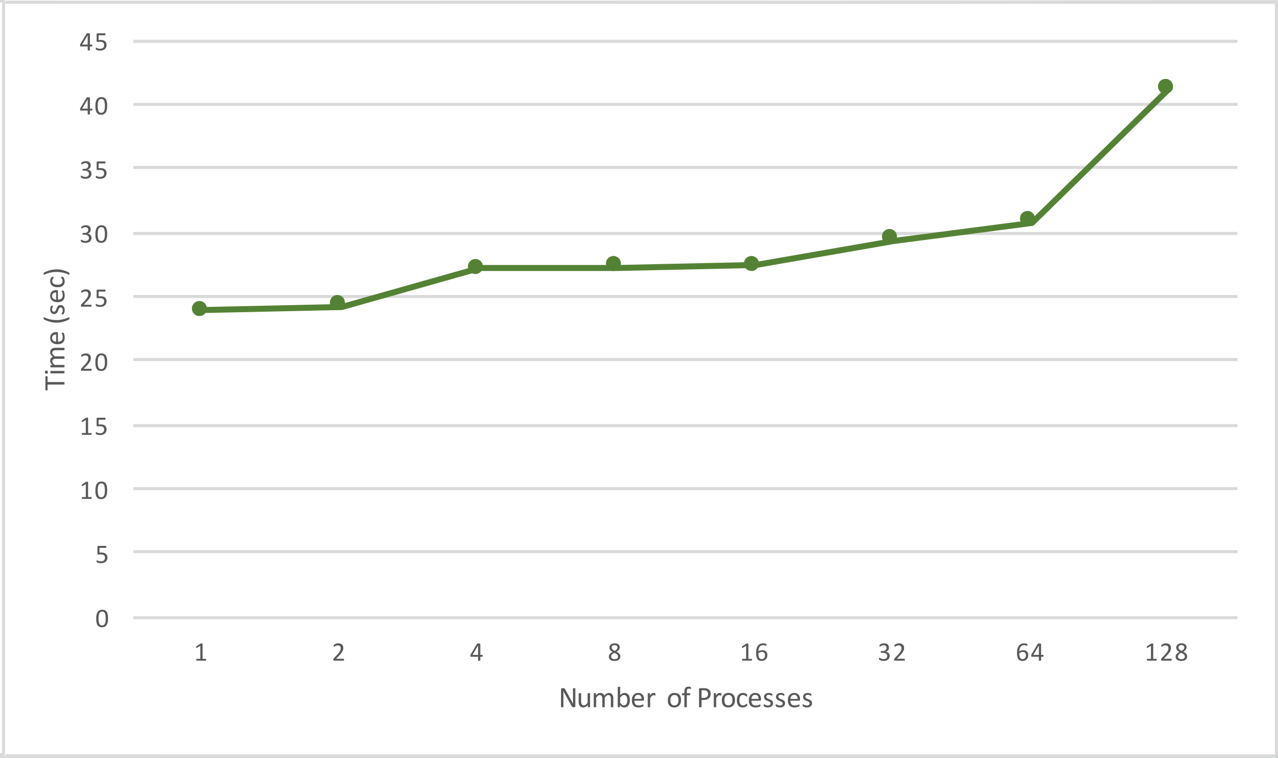
```
[diags1]
vars = ["T"]
seasons = ["ANN", "SON"]

[diags2]
vars = ["PRECT"]
seasons = ["JJA"]
```

ESGF Earth System Grid Federation

Design and Architecture

- E3SM Diagnostics Package performance, w/ 1330 individual diagnostics
- Good strong and weak scaling





ESGF Earth System Grid Federation

Uses

- PCMDI Metrics Package:
  - Completed January 2017
  - Need to add new features
- E3SM Diagnostics Package
  - In progress, 7 plot sets done
  - Replacement for AMWG Diagnostics
- ARM Diagnostics

Future Work

- Cloud computing
  - Containerize software, deploy on PaaS (AWS, Google Cloud Platform, etc)
- Library of metrics, but would introduce more dependencies
- Standardize more components
  - Ex: Interface for plotting reference, test, and diff data
    - `plot(reference, test, difference)`
- Finish implementing E3SM Diagnostics, ARM Diagnostics, expand on PCMDI Metrics Package