# Optimizing climate data analysis workflows: Strategies and lessons learned from two case studies

**Alex Goodman** [1] , **Colin Raymond** [1,2] , **and Peter Kalmus** [1]

[1] *Jet Propulsion Laboratory, California Institute of Technology*

[2] *University of California Los Angeles*

## Why data analysis workflows need to be better optimized?

Workflows in Earth Science are increasingly relying on larger datasets, especially ones involving evaluations of multiple climate models. Often, the first resort to scale these up is to run the analyses in parallel which is leading to dramatic increases in overall usage of supercomputing resources over the past decade (Figure 1).

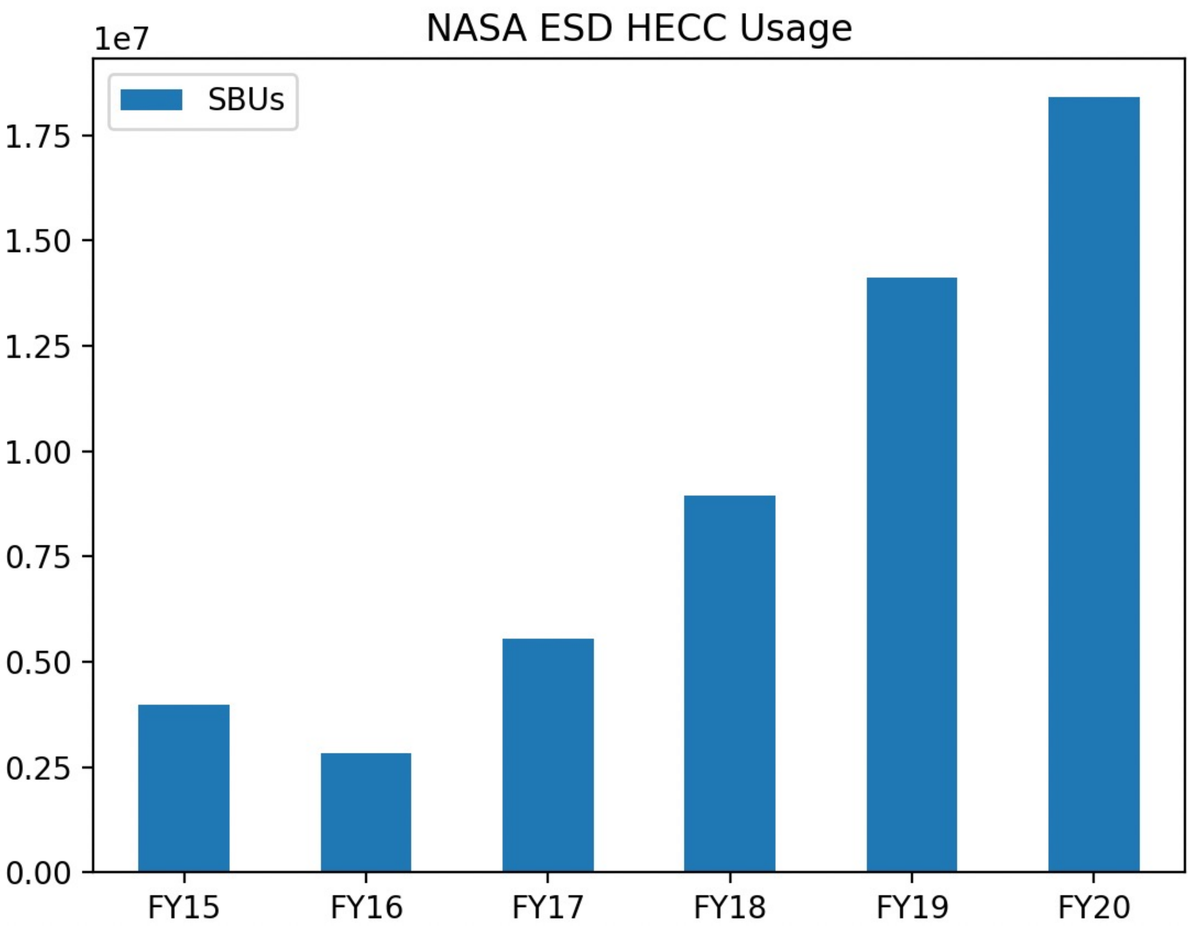


**Figure 1. Historical utilization from Earth Science Directorate (ESD) projects of the NASA HECC supercomputing resources[1]**

Consistent with this big data problem, NASA program managers are also expecting this demand to continue to grow over the next decade, with a possible concern that demand will increasingly outstrip available computational capacity as shown in Figure 2.

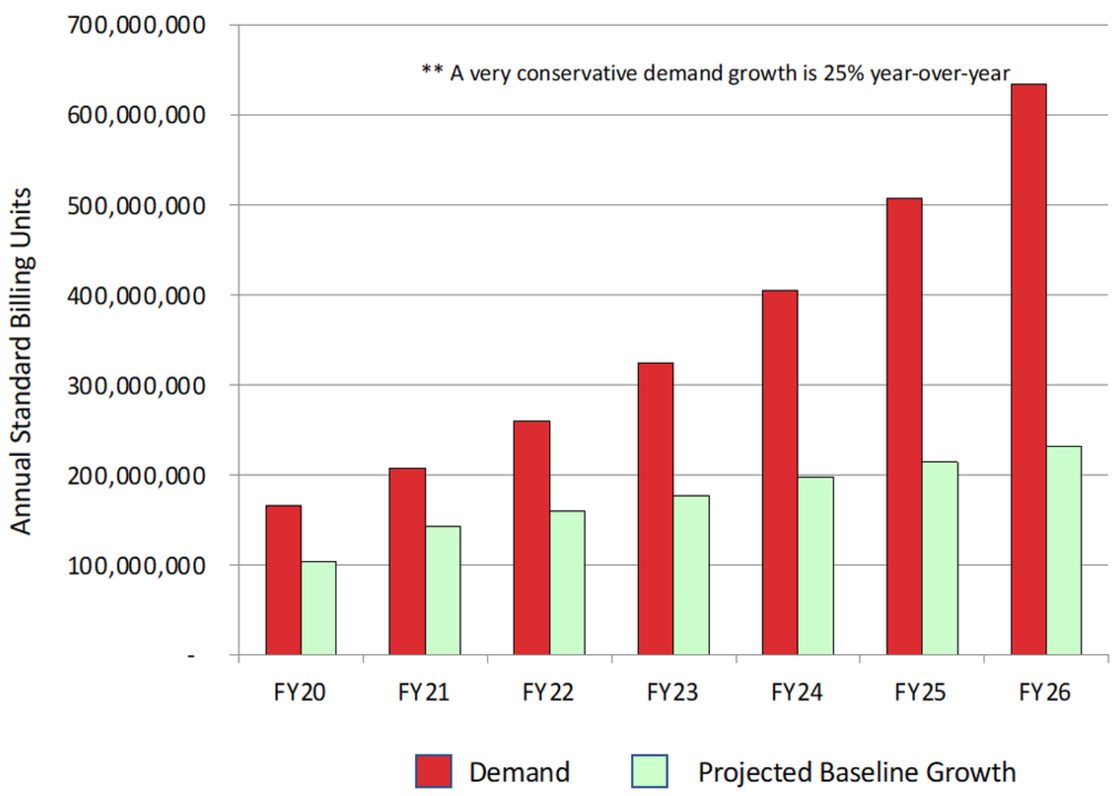


**Figure 2. Projections in NASA supercomputing user demand[1] (red) and actual available usage (green).**

**However, we contend that in some cases, the need to rely on high performance computing resources can be greatly reduced or even eliminated when the analysis code is better optimized.** We will demonstrate this by describing how we optimized widely distributed codes for calculating **heat index** and **wet-bulb temperature** leading to improvements in running time by over two orders of magnitude. Both algorithms rely on numerically intensive root finding methods (Bisection and Newton's method respectively) to derive their results.

## Why focus on these two use-cases?

Both of these quantities are highly relevant in analyzing projections of future climate extremes which are of great interest to decision makers. The current inefficiency of these codes has proven to be a great bottleneck to scientists, so code optimizations can greatly reduce project costs and reliance on supercomputing resources.

## Case Study 1: Extended Heat Index[2]

- Problem: Most data analysis codes are written in interpreted languages since more efficient compiled languages like C or Fortran are unsuitable for exploratory data analysis.
- Original code provided by authors is written in python and is not vectorized.
- **Instead, we can use the numba python library to accelerate and vectorize the code using Just in time compilation (JIT)**. This can result in workflows written in python that run nearly as fast as they would if they were written in C or Fortran.
- Using this along with some additional minor modifications to the algorithm, **we achieved a performance improvement of over 550x. (Figure 3)**

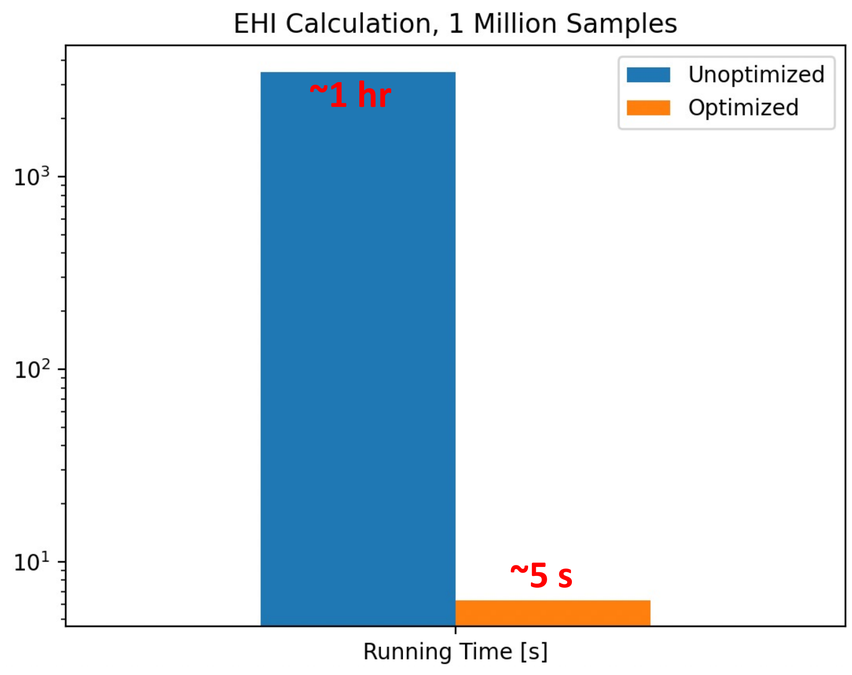


**Figure 3. Running time for EHI calculation using 1 million randomly generated samples.**

## Case Study 2: Wet Bulb Temperature[3]

- In this example (originally written in MATLAB), the code is already vectorized so using numba alone leads to more modest performance improvements than in case study 1.
- However, we determined through automatic differentiation techniques that derivatives were being calculated incorrectly.
- **Fixing this error allowed the algorithm to converge much more quickly, leading to a performance improvement of over 1000x. (Figure 4)**

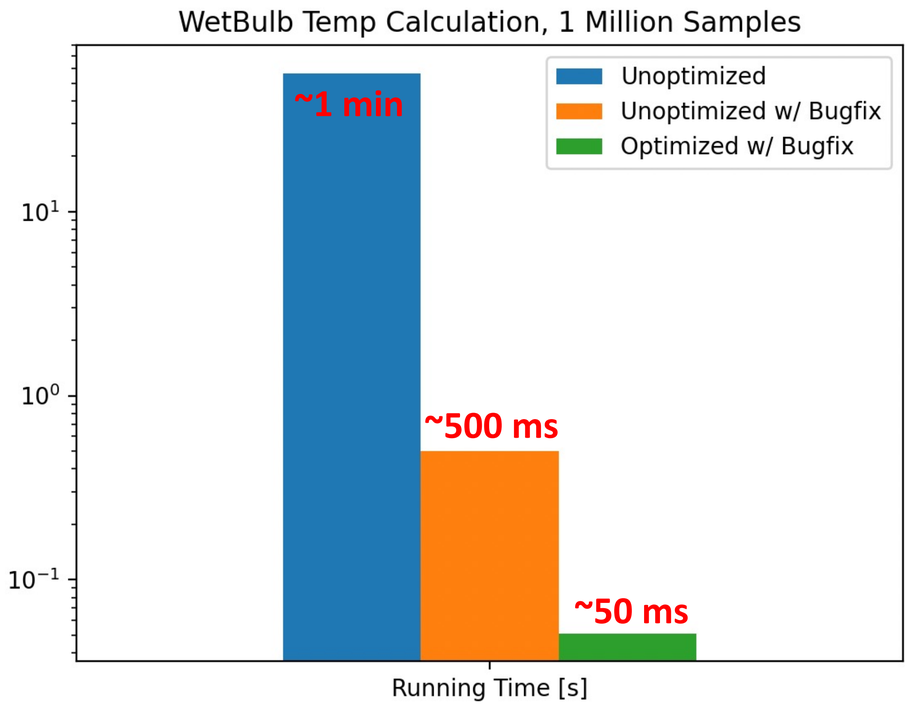


**Figure 4. Running time of WetBulb temperature calculation using 1 million generated samples**

**Acknowledgements**

We would like to thank David Romps and Yi-Chuan Lu for providing us their guidance on the EHI algorithm and code. Original versions of the code for calculating the wet bulb temperature and EHI respectively are available from:

1) https://github.com/bobkopp/WetBulb.m
2) https://romps.berkeley.edu/papers/pubdata/2020/heatindex/heatindex.py

**References**

[1] https://hec.nasa.gov/news/reports.html

[2] Lu, Y., and D. M. Romps, 2022: Extending the Heat Index. J. Appl. Meteor. Climatol., 61, 1367–1383, https://doi.org/10.1175/JAMC-D-22-0021.1.

[3] Davies-Jones, R., 2008: An Efficient and Accurate Method for Computing the Wet-Bulb Temperature along Pseudoadiabats. Mon. Wea. Rev., 136, 2764–2785, https://doi.org/10.1175/2007MWR2224.1.