Interactive Visualization of Climate Dataset

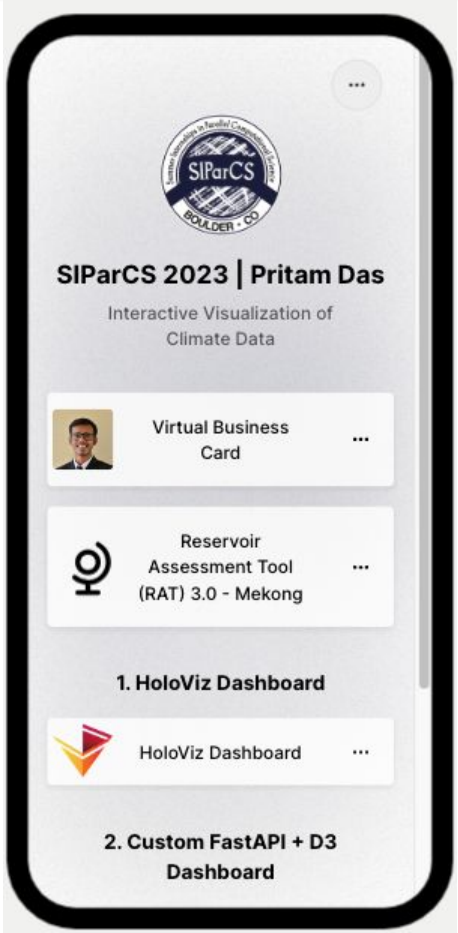
Pritam Das, Negin Sobhani,
Tammy Zhang and Nihanth Cherukuru

NCAR
UCAR

August, 2023

Before we begin…

linktr.ee/**pdas47**



I encourage you to **follow along**

or save it for later!

> whoami

Pritam Das

3rd year PhD student

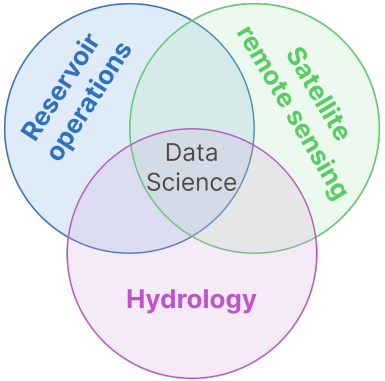
Civil & Environmental Engineering

University of Washington

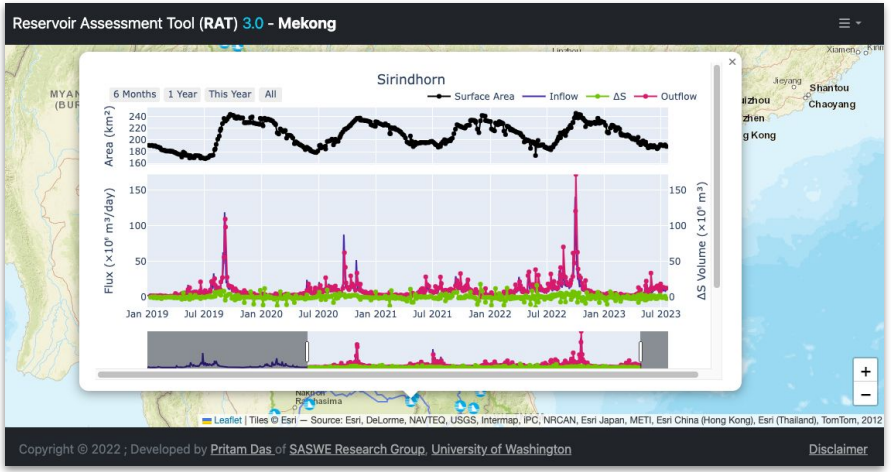
saswe research group

UNIVERSITY of WASHINGTON

linktr.ee/**pdas47**



- Reservoir Assessment Tool (RAT)



Objectives and Outline

Objectives

- Building a dashboard for the CESM LENS2[1] climate dataset.
- Explore different python based dashboarding solutions.

Outline

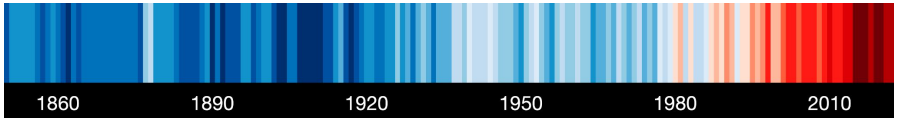
- Introduction
- Data
- Methods
- Dashboard demo and functionality
- Performance comparisons
- Conclusions

linktr.ee/**pdas47**

[1]Rogers et al., (2021) - full reference at the end

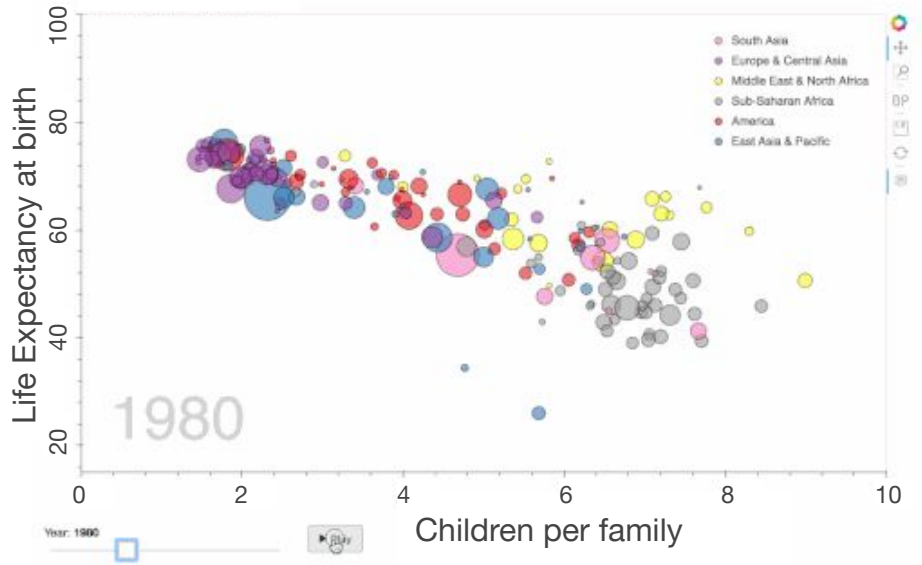
Interactive Visualization in Science

“A picture is worth a thousand words”



source: Modified (cropped) from #ShowYourStripes by Ed Hawkins, National Centre for Atmospheric Science, UoR. CC BY 4.0.

Interactive visualizations let you **read your data** like a **book**



source: https://holoviews.org/gallery/apps/bokeh/gapminder.html

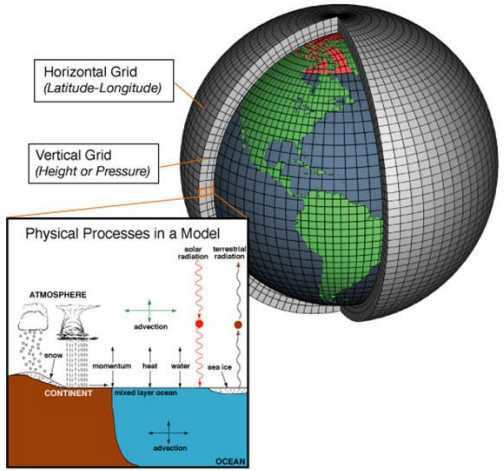
In *Science*

- **Exploration** - what to focus on
- **Analysis** - discover patterns
- **Results** & **Communication** - share with the community

Data

LENS2 - CESM2 Large Ensemble Community Project

- Output of **CESM** (Earth System Model) - simulates various Earth processes
- **Ensemble** Simulation - 100 different simulations with varying initial conditions
- **1°** spatial resolution, between **1850-2100** (daily/monthly)
- Forcing Types - CMIP6 & SMBB[1]



Source: NOAA, https://celebrating200years.noaa.gov/breakthroughs/climate_model/modeling_schematic.html

Pre-processing

- Originally **~70TB** - difficult to load it in memory
- **Annual Mean** and **Std. Deviation** across ensemble members
- Final Data Size: **~400MB** / variable * **8** Variables (subset)

xarray

dask on *casper*

Intake-ESM

[1]Rogers et al., (2021) - full reference at the end

Explored methods

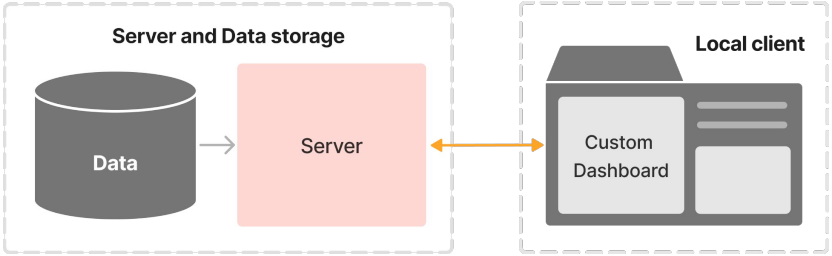
The HoloViz Ecosystem

High-level python plotting tools

- **Declarative plotting** - describe your data to make it interactively visualizable
- **Choice** between **plotting backends**
  bokeh: interactive visualization
  matplotlib: publishing quality static plots
- **Extends existing packages** in the scientific python stack - xarray, pandas, dask

HoloViews GeoViews Param Panel

Custom solution

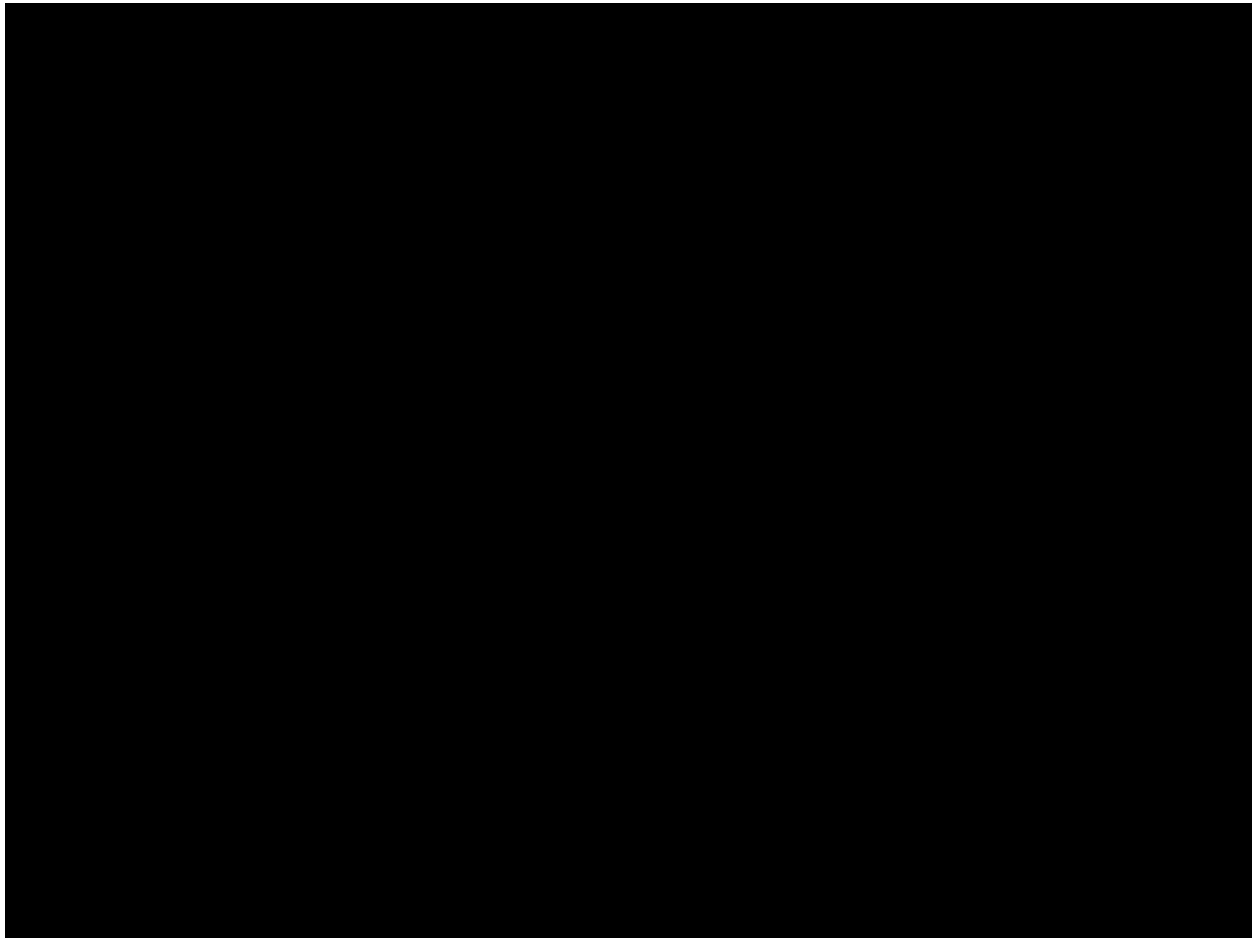


- Custom Client-Server model
- *Server*: performantly **querying** and **sending data**
- *Front-end/Client*: making appropriate **requests** and **visualization**

FastAPI Leaflet D3

xarray dask

HoloViz based dashboard demo



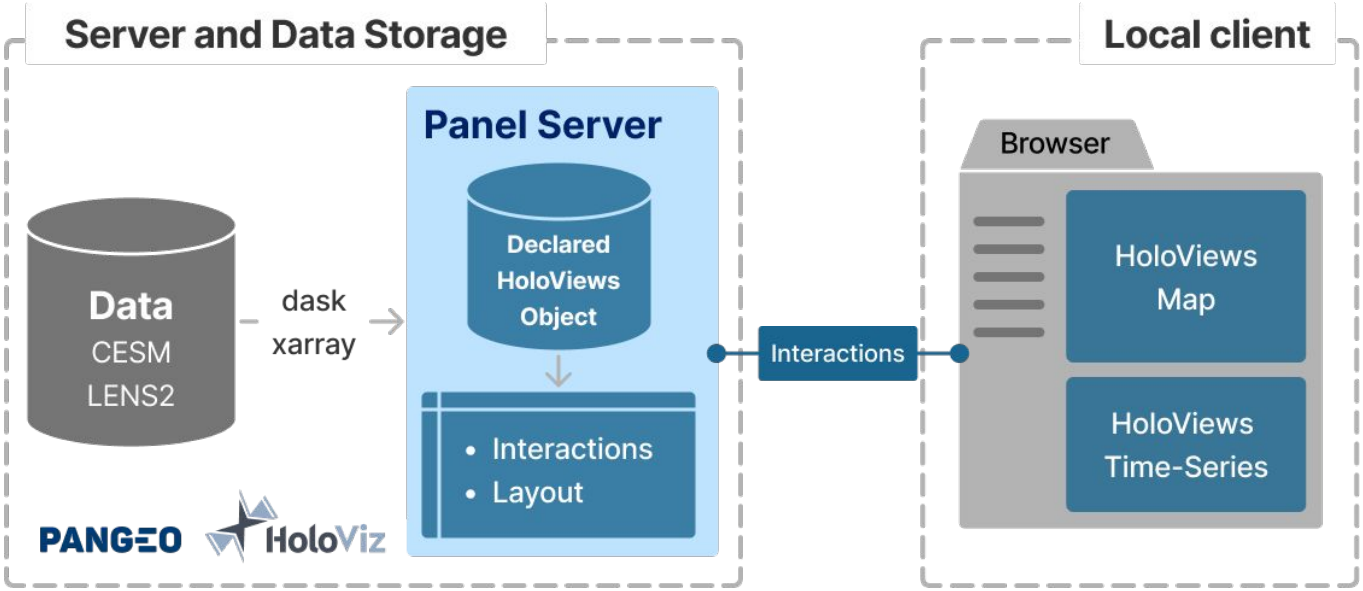
Visit dashboard at

http://146.190.13.102/

link also available in linktree

Big screen (laptop) preferred

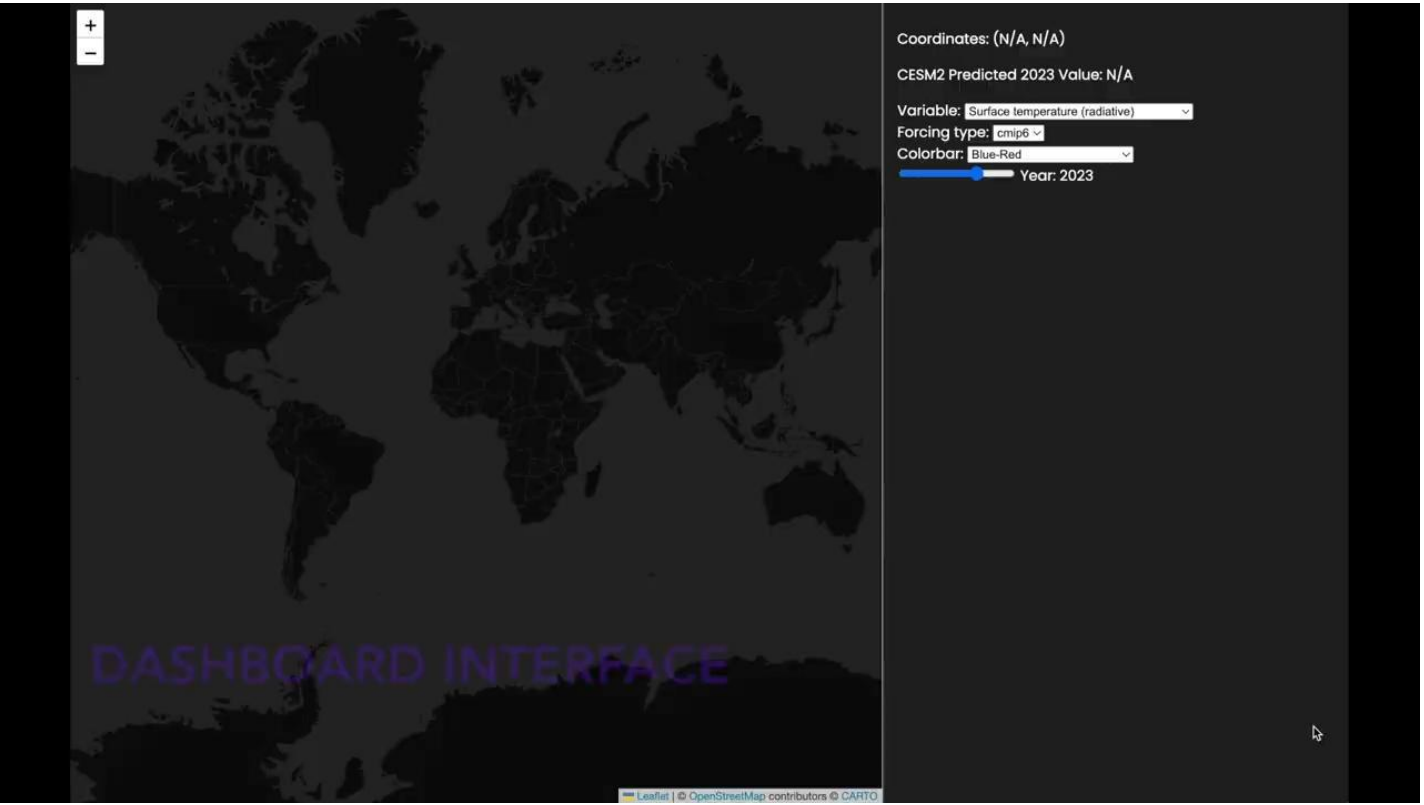
HoloViz based dashboard conceptual diagram



- Standing on the shoulders of Giants - **out-of-the box interactivity**
- Jupyter notebooks to servable dashboards

- Easy to begin, **difficult to customize**
- Performance comparatively lower especially for larger datasets

Custom FastAPI + D3 based dashboard demo



FastAPI

Visit dashboard at

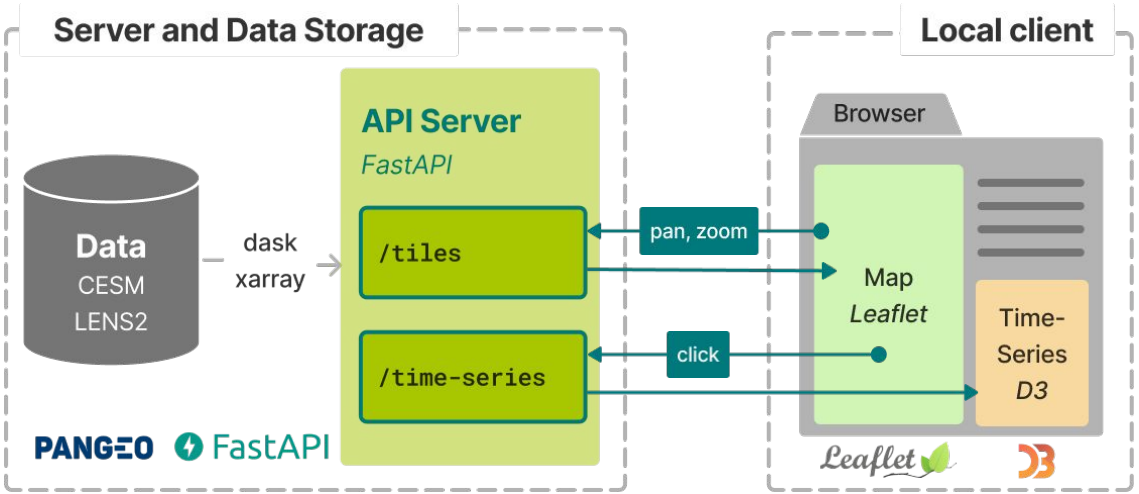
http://143.244.211.235/

link also available in linktree

Big screen (laptop) preferred

Custom FastAPI + D3 based dashboard conceptual diagram

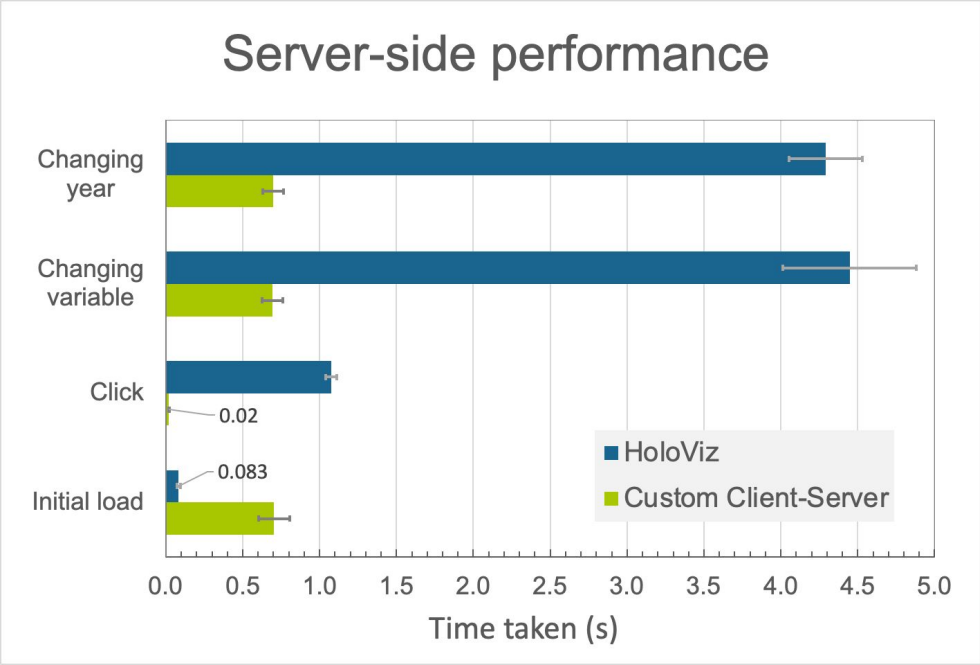
- API server built using FastAPI
- The website makes API requests for either tile or time-series data, and displays accordingly



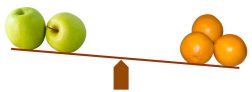
- **Minimal overhead** - higher performance is possible.
- More customizations with D3 is possible

- Higher human time for development
- Each interaction will have to be explicitly defined - **more design considerations**

What about performance?



Server configuration: 2 vCPUs, 16 GB Memory, 50GB SSD.



- Performance alone is not the only consideration
- Personnel time for development, learning curve of Javascript for scientists
- Differing features and design choices

👈 Time required to execute **all** the **functions** following an **interaction event**.

- Overall, custom solution is faster.
- Depends a lot on design choices.

Bottlenecks?

Depends on *data-size* and *interaction type*

- ***HoloViz***:
  - Memory - Low data size (in-core)
  - Disk IO - High data size (out-of-core)
  - CPU - Regional mean
- ***Custom***:
  - Memory/Disk - data size
  - Network - higher data-transfer

Conclusions

HoloViz

- Quicker to get started
- Interfaces well with existing tools - xarray, jupyter, matplotlib, …
- Suitable for smaller datasets
- **Exploration** and **Analysis**

Custom solution

FastAPI + D3

- More headroom for performance optimization
- Higher human (development) time
- **Results** and **Science Communication**

Recommendations and Future work

- Use HoloViz to *get started*, and use a custom solution for *final results*.
- *Caching* can help boost performance.
- Further *optimizations* to callback function calls and API calls.
- *Containerization* for easy deployment.

Thank You!

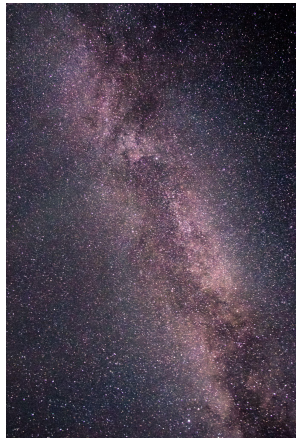
To NCAR, SIParCS and CISL for this opportunity,

To Negin, Nihanth and Deepak for being awesome mentors,

To Virginia, Jerry, Julius, and Ben for making the internship such a smooth and fun experience,

To all the staff, especially admins, who made it all possible,

and, to all the fellow interns and friends I made along the way!



References and Acknowledgements

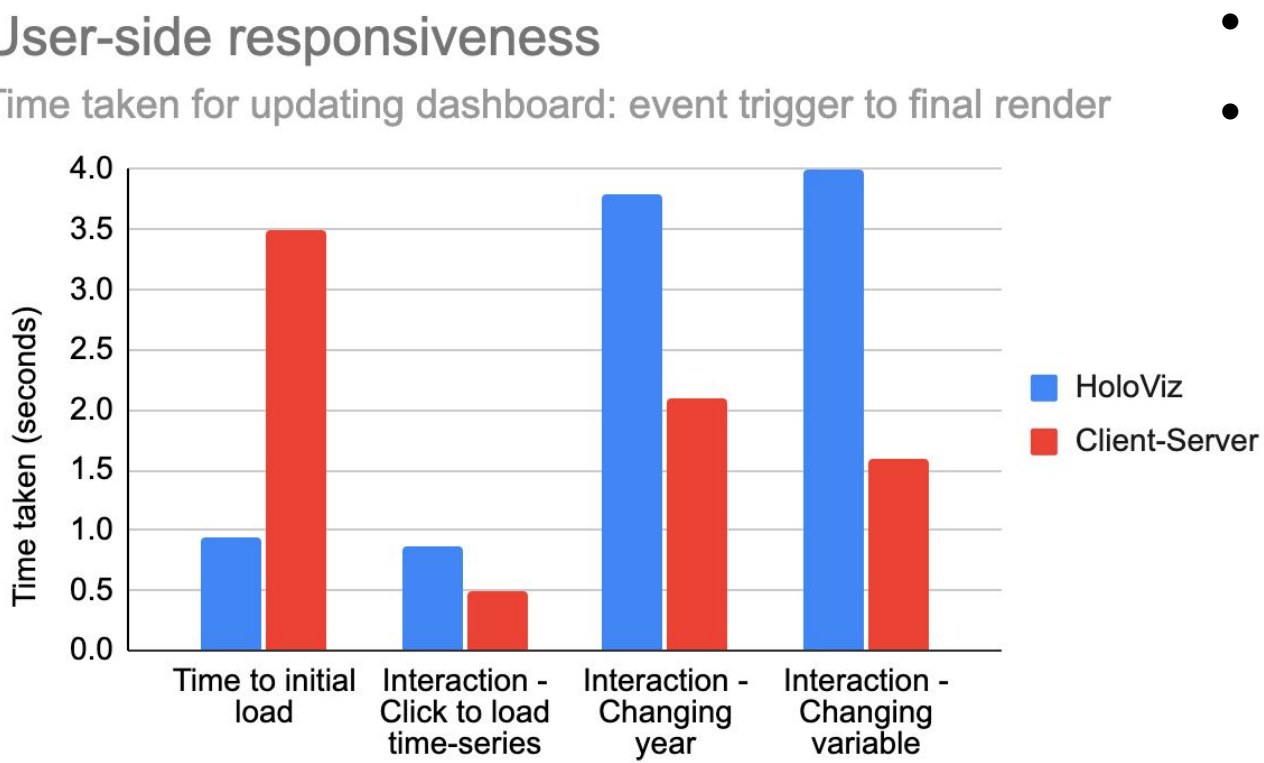
1. Rodgers, K. B., Lee, S.-S., Rosenbloom, N., Timmermann, A., Danabasoglu, G., Deser, C., Edwards, J., Kim, J.-E., Simpson, I. R., Stein, K., Stuecker, M. F., Yamaguchi, R., Bódai, T., Chung, E.-S., Huang, L., Kim, W. M., Lamarque, J.-F., Lombardozzi, D. L., Wieder, W. R., and Yeager, S. G.: **Ubiquity of human-induced changes in climate variability**, *Earth Syst. Dynam., 12, 1393–1411*, https://doi.org/10.5194/esd-12-1393-2021, **2021**.

Acknowledgements

NCAR, Boulder CO, USA and the IBS Center for Climate Physics, Busan, South Korea for jointly developing the CESM2 Large Ensemble Community Project (LENS2) climate dataset.

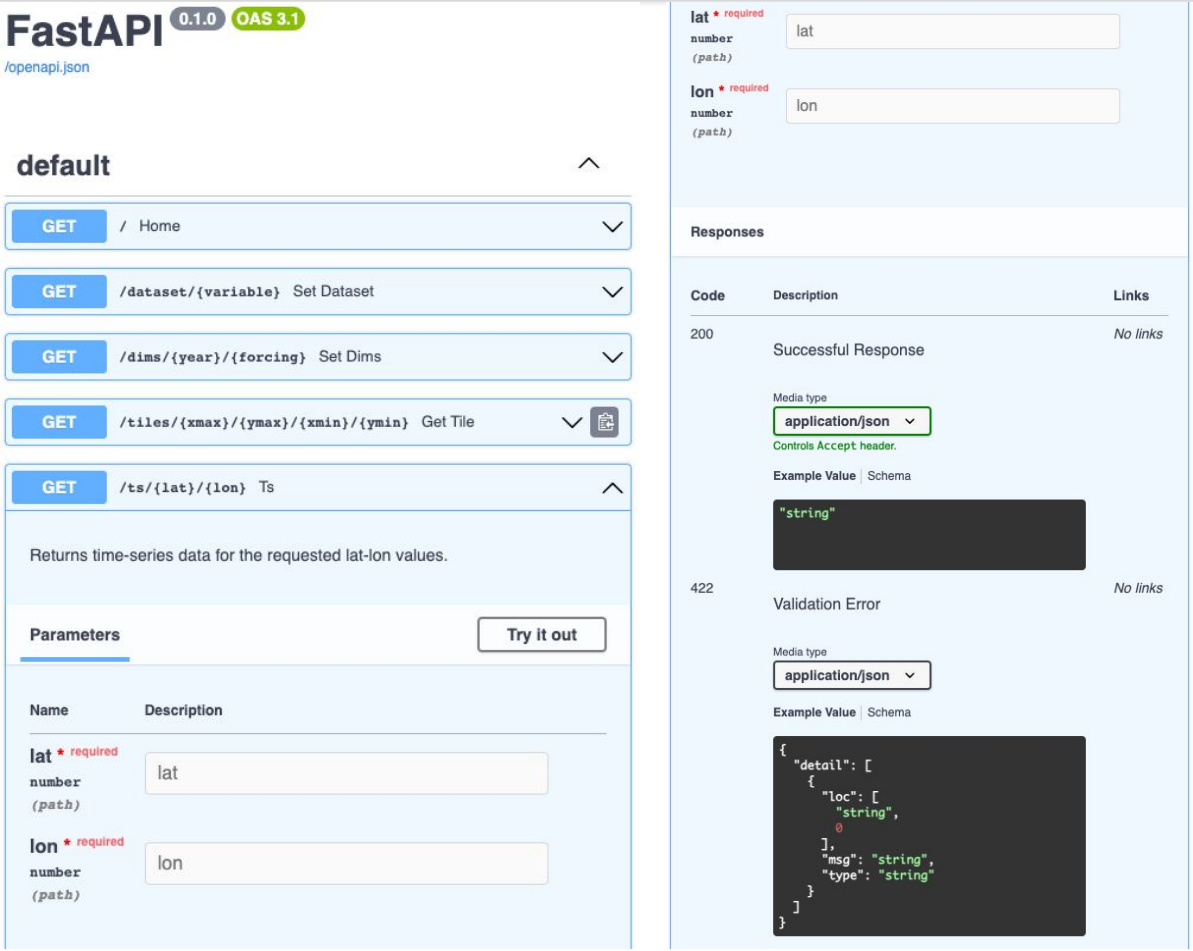
Supplemental materials follow

User side performance



- Measured using chrome DevTools, *manually*
- Includes all the time components
  - Server processing
  - Data transfer
  - Client side processing and rendering

FastAPI - docs



- server/docs

Provides useful info about the API, its endpoints, and you can try out these endpoints in this interface.

Carbonplan

- How does carbonplan methods work in comparison with the presented dashboards?