NAU NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing, and Cyber Systems

Fossilized Controller

Connecting Paleoclimatologists with Docker Containers

Team: Emily Ramirez Serrano, Jeremy Klein, Jadon Fowler, Mumbi Macheho-Mbuthia

Client: Dr. Nicholas McKay, NAU, School of Earth and Sustainability **Team Mentor:** Melissa D. Rose

What is the Problem?

Paleoclimatologists, such as Dr. McKay, create code that show how climates changes over time and in different regions. This code is known as a climate reconstruction model.

This code is difficult to share due to dependencies that differ across operating systems. Dr. McKay aims to solve this problem with containers. The Fossilized Controller is a command-line interface tool that guides scientists through containerization of their climate models.

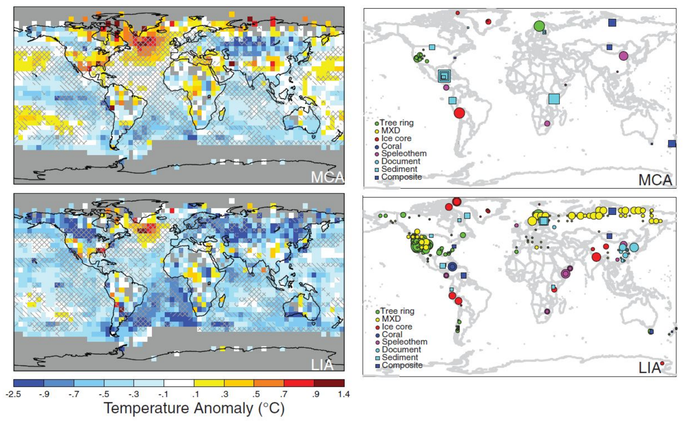

Figure: Output from a Climate Model

Solution Overview

Our solution is a command-line interface (CLI) that automates the containerization process.

The interface must:

- **Simplify the containerization process:** Users will be able to create a Docker Container with their models
- **Run models with new parameters:** Containers built by the Fossilized Controller can be sent new parameters and files via the CLI
- **Be language agnostic:** Models can be built in Python or R, the primary languages for climate models

How It Works

Below will showcase the steps to containerizing and running a climate model. The **First Step** is to create a Docker Image with the *presto create* command:

```
# navigate to the file where the reconstruction is located
home/user/example/projects $ cd Temp12k

# run presto create to generate a DockerFile
/.../Temp12k $ presto create
What is the command to run your main file?
Here are some examples:
- python3 main.py
- r main.R
- sh main.sh
> python3 Temp12kMain.py
```

The **Second Step** is to use the newly created Dockerfile to build a Docker Image named “lmrt” with the *presto build* command:

```
# run presto build with a name for the image
/../Temp12k $ presto build Temp12k
Building the image... This may take several minutes
```

The **Third Step** is to run a Docker Container using the “lmrt” Docker Image with the *presto run* command:

```
# run presto run to run the created docker container
/../Temp12k $ presto run Temp12k
Running the container...
Output files successfully saved at ./reponse_data.zip
```

Feature Highlights

```
user:~/../~$ docker logs --follow $(docker ps -q)

 * Serving Flask app 'adapter' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
```

Figure: Logs of the Processes Occurring Inside the Docker Container Created by the Fossilized Controller

Architecture and Technologies

- **Command-Line Interface:** Built using the Python *Click* package. Gives the user the ability to create, run, and destroy Docker containers.
- **Container Manager:** Written in Python and uses the *docker* package to access Docker.
- **Adapter Libraries:** Uses the Python *Flask* package and R’s *httpuv* library to create and run an HTTP server.

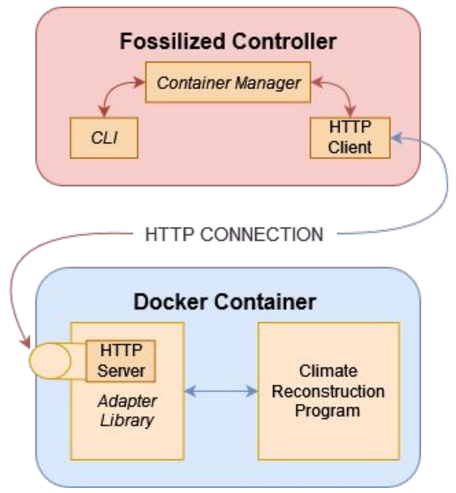

Figure: Module Interactions within the Controller

Future Work

- **Website that connects to Fossilized Controller:** Allows anyone to be able to run containers with new inputs
- **Write adapters in additional languages:** Allows for climate models not written in Python and R to be easily containerized