2022 SIParCS TECHNICAL PROJECTS

Undergraduate

- Project 1. Expanding GeoCAT's Visualization Capabilities
- Project 2. Forecasting the COVID-19 Pandemic and the Immune's Response against the Virus using Ensemble Data Assimilation
- Project 3. Measuring GPU power usage
- Project 4. Modernizing the CISL Visualization Gallery Website
- Project 5. Parallel algorithms to recognize spatial patterns for climate analysis
- Project 6. Physics-Guided Machine Learning Algorithm Evaluation
- Project 7. Project Pythia Content Development
- Project 8. Python data analysis & visualization and Jupyter notebook development for unstructured grids data
- Project 9. Python reimplementation of Fortran subroutines for use on supercomputers
- Project 10. Using Binder as a fast and scalable educational platform

Undergraduate and Graduate

- Project 11. Apply Search Techniques for Scientific Data Discovery and Exploration (2 positions)

Graduate

- Project 12. Development of a Python Knowledge-based weather station interpolation algorithm
- Project 13. Exploring performance of GeoCAT data analysis routines on GPUs
- Project 14. Future shock: Speed and Scalability in Data Assimilation
- Project 15. Integrating a Lidar Sensor into a Low-cost Weather Station
- Project 16. Leveraging Xarray for reproducible and scalable analysis of remote sensing data in the cloud
- Project 17. Pi-WRF Community Driven Educational Modules and Classroom Activities Development
- Project 18. Using containers and Spack to simplify the portability and reproducibility of scientific applications

Note: Please apply to no more than two (2) SIParCS projects.

Project 1. Expanding GeoCAT's Visualization Capabilities

Areas of Interest in order of relevance: Visualization, Data Science, Software Engineering

Description: For nearly two decades researchers in the atmospheric, oceanic, and related sciences have employed the NCAR Command Language (NCL) to analyze and plot their data. With the emergence of Python as the scripting language of choice for scientific workflows, NCAR has begun its course to migrate many of NCL's highly specialized capabilities into the Python ecosystem. During the past two summers, the GeoCAT-Examples Gallery and GeoCAT-viz were created and have been continuously updated with new NCL to Python projection examples and functionality.

Over this summer internship, the student will contribute to the GeoCAT-viz and GeoCAT-examples repositories and learn about data visualization in the atmospheric and oceanic sciences using matplotlib, cartopy, and xarray. In the GeoCAT-examples repository, the student will generate plotting examples inspired by the NCL gallery. The student will plot using different map projections and with different overlays of satellite, measured, and modeled data. In the GeoCAT-viz repository, the student will expand plotting functionality to reduce the amount of boilerplate code in the NCL to Python visualization scripts. Over the course of the internship, we will see GeoCAT's visualization capacity grow.

Students: The project is open to undergraduate students.

Skills and Qualifications:
-Experience with Python programming
-Familiarity with Jupyter Notebooks
-User level familiarity with Linux and Unix-based tools for scripting and file manipulation.
-Experience with NumPy, Matplotlib, Cartopy, or Xarray
-Experience with Git or GitHub
-Ability and willingness to work with a team
-Good communication and writing skills

Project 2. Forecasting the COVID-19 Pandemic and the Immune's Response against the Virus using Ensemble Data Assimilation

Areas of Interest in order of relevance: Software Engineering, Numerical Methods, Mathematical Modeling (biology)

Description: The outbreak of the coronavirus disease (SARS-CoV-2) has received increasing attention among scientists all over the world. Efforts have been dedicated to understand the epidemiology of the disease and curb the spread of the virus. This project considers two mathematical models that simulate the novel disease and study its dynamics in infected patients. The first model is an extended SEIR model with a vaccination compartment. The model considers various stages of infection, including susceptibility, exposure, infection, quarantine, vaccination, recovery and death. The second model considers both innate and adaptive immune responses to COVID-19 and consists of healthy cells, infected cells, viral load, cytokines, antibody cells, among many other biological cells. The two models have been studied and tuned using real data from different countries and patients around the world in order to improve their prediction skill. One efficient and widely-known technique that scientists often use to confront models with data is ensemble data assimilation (DA). The idea behind DA is to run several instances of the model simultaneously and use real-world data along the way to enhance the model's trajectory.

The main goal of the project is to migrate the code that runs these two models from MATLAB into the Data Assimilation Research Testbed (DART). DART [https://dart.ucar.edu] is an open-source community facility that supports ensemble DA activities in many earth-system models and across different computational platforms. DART is a Fortran-based software and thus code migration would require translating a few MATLAB functions and routines into Fortran. After migration, testing the predictability of the models and the effectiveness of the ensemble DA framework will be performed using available data. The final stage involves analyzing the results using already established DART diagnostic tools.

Overall, the project offers the opportunity for interested applicants to work on exciting real-world pandemic problems while collaborating with experts in science fields such as computer science, modeling and data assimilation. Working with these SEIR-based mathematical models in addition to DART will help applicants gain extensive experience in modeling and inverse problems. Furthermore, working on such a project will help open the door to understand the behavior and spread of any infectious-type disease, not particularly COVID-19.

Students: The project is open to undergraduate students.

Skills and Qualifications:
- Undergraduate program such as computer science, mathematics, biology or related field
- Good oral and written communication skills
- Ability to read, write and understand code
- Open to work with people from diverse backgrounds
- Ability to work in a team environment

Project 3. Measuring GPU power usage

Areas of Interest in order of relevance: Software Engineering, Supercomputer Systems Operations

Description: A large percentage of the cost to operate a modern data-center is due to the electricity required to run the computing platform. One technique to significantly reduce the electricity cost is through the use of Graphics Processing Unit (GPU) as they typically require less electricity to perform an equivalent calculation versus a contemporary CPU. Unfortunately measuring the exact energy usage is often non-trivial because it requires interfacing with specialized hardware and understanding the measurement limitations. Often estimates of energy usage use different assumptions and can be inconsistent even on similar hardware.

The goal of this 2022 summer internship is to simplify the measurement of energy consumption on both GPU and CPU based codes. We aim to create a software package that is as easy to use and understand as the wall clock timers. This project will develop an easy to use energy measuring methodology that can be shared and utilized by researchers in the Application Scalability and Performance (ASAP) group. This work will target energy measurement on the upcoming NCAR Linux cluster which will have significant numbers of both CPU and GPU nodes.

Students: The project is open to undergraduate students.

Skills and Qualifications: Familiarity with python, linux environment and compiled languages like C/C++. Strong motivation to learn new skills and resolve issues in a team is required. Experience with computer architecture and high performance computing is highly desirable.

Project 4. Modernizing the CISL Visualization Gallery Website

Areas of Interest in order of relevance: Web Design, Digital Asset Management, Software Engineering

Description: The CISL Visualization Gallery webpage (https://visgallery.ucar.edu/) serves as a platform to showcase NCAR's scientific data and to highlight visualization research done in the Visualization Services and Research Group. This content is used to communicate scientific findings to domain experts, policy makers, and the general public in order to promote the understanding of geosciences.
The current page was created on a WordPress/Pantheon platform. It is difficult to update, maintain, and add new content. The student in this position would primarily focus on migrating the page away from a WordPress platform and modernizing the page to make it responsive and easier for staff to update content. The student would also have an opportunity to update the design and layout and to add interactive content based on their level of web design expertise.

Students: The project is open to undergraduate students.

Skills and Qualifications: Experience with HTML/HTML5, Markdown, and CSS is a must. Bonus qualifications: familiarity with static site generators (e.g. Zola, Hugo, or Gatsby); advanced CSS skills such as CSS Grid and Flexbox; responsive web page design for different desktop and mobile platforms; ability to apply color, layout, and design to web pages; knowledge of best practices for web page accessibility

Project 5. Parallel algorithms to recognize spatial patterns for climate analysis

Areas of Interest in order of relevance: Data Science, Application Optimization/Parallelization, Software Engineering

Description: To study the Earth's current and future climate, scientists use physically based computational models to represent the earth system. One challenge scientists face is that all such models are imperfect. Often that imperfection results in coherent biases in space and time that a human can readily interpret, but humans have limited mental bandwidth. Computers can aid in this effort via a combination of brute-force search, supervised and unsupervised classification techniques, and machine learning algorithms. This provides an opportunity to make better use of existing climate models by retrieving more of the information available in these imperfect models in an automated fashion. This problem is particularly important for regional climate studies because the enormous computational cost of running higher resolution models means that regional climate models are either lower resolution than desired (with larger potential for errors) or have large simplifications in their physical representation (with larger potential for errors). However, any such technique must process many terabytes of data using costly search and sorting functions.

This project will have a student develop and apply efficient parallel algorithms to an archive of regional climate model simulations to improve our understanding of local scale changes in climate that are critical for end users. By leveraging existing simulations performed for historical time periods, a student will be able to explore the trade-offs associated with different algorithms and different regional climate modeling approaches in comparison to observations. This work will also provide an opportunity to use and learn modern data science parallel analysis platforms (e.g. xarray, dask, and tensorflow) on the NCAR supercomputer (Cheyenne) and associated GPU accelerated analysis platform (Casper). Languages that provide efficient parallelization structures will be explored (UPC++, coarray-fortran, python+dask, Julia, or cuda-C), with the specific implementation to be selected by the student and advisors.

Students: The project is open to undergraduate students.

Skills and Qualifications: Experience with parallel algorithms desired. Familiarity with one of the following languages (C, Python, Fortran).

Project 6. Physics-Guided Machine Learning Algorithm Evaluation

Areas of Interest in order of relevance: Data Science, Machine Learning, Visualization

Description: The goal of this project is to evaluate a suite of machine learning algorithms containing physical constraints that enhance the spatial and temporal consistency of their predictions. One of the ongoing challenges in using machine learning in Earth System Science problems is that the algorithms will sometimes produce sudden non-physical changes in their prediction over time, resulting in large prediction errors and confusion from downstream users. Multiple approaches to addressing these issues are being developed but need to be evaluated on a diverse set of Earth System Science use cases to determine how well the approaches generalize across domains. The intern will develop and run both parallel and interactive programs to evaluate each physics-guided machine learning approach on at least one use case, which may include severe storms, winter weather, and coastal weather. They will develop experience working with large environmental science datasets, custom machine learning architectures, NCAR HPC systems, GPUs, interactive data analysis, and statistical evaluations. The intern will collaborate with the NCAR Analytics and Integrative Machine Learning group and the NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography (AI2ES) as part of the project.

Students: The project is open to undergraduate students.

Skills and Qualifications: Must have: differential equations. Interest or experience in scientific Python, machine learning, statistics, physical science, good oral and written communication skills

NOTE This project has the possibility of an extension for a second summer.

Project 7. Project Pythia Content Development

Areas of Interest in order of relevance: Software Engineering, Education, Visualization

Description: Project Pythia (https://projectpythia.org/) provides a public, web-accessible training resource to help educate current and aspiring earth scientists to more effectively use both the Scientific Python Ecosystem and Cloud Computing to make sense of large volumes of numerical scientific data. We are looking for help developing, curating, and testing our content. This involves finding external learning resources to add to the Pythia Resource Gallery (https://projectpythia.org/gallery.html), working through our resources on the Pythia Foundations Book (https://foundations.projectpythia.org/) while documenting your learning experience, and writing more documentation on various Python tools, packages, and techniques to add to the Pythia Foundations Book. If the student is interested, work can be expanded to include outreach efforts to help grow our audience, help with tutorial workshops or other events, and potentially some web development.

Students: The project is open to undergraduate students.

Skills and Qualifications: Python, Atmospheric Science, Jupyter Notebooks (optional), Markdown (optional), Git (optional), GitHub (optional)

Project 8. Python data analysis & visualization and Jupyter notebook development for unstructured grids data

Areas of Interest in order of relevance:
Description: Project Raijin was recently awarded by NSF EarthCube in order to develop community-owned, sustainable, scalable tools for data analysis and visualization that can operate on unstructured (i.e. not regular lat/lon grids) climate and global weather data at global storm resolving resolutions. The development of Project Raijin leverages the Scientific Python Ecosystem (SPE), particularly the open development Xarray and Dask packages, and the Pangeo community. Our work is conducted under an open development model that encourages participation of the community in all aspects of the project's development.

During the SIParCS 2022 summer internship, the intern will have the opportunity of working in a novel research and development project in the SPE by helping implement example plotting scripts for various unstructured grid datasets and create Jupyter Notebook based training modules. Throughout these efforts, the student will explore and learn about data visualization in geosciences using commonly used Python tools such as Matplotlib, Cartopy, and Holoviews. The student will also learn high performance computing (HPC) principles through use of NCAR's HPC clusters as well as using parallelization and optimization packages such as Dask, Numba, Datashader, etc. Most or all of the student's work will be made publicly available through our open development model, which will in turn help the intern create a strong Python portfolio in data analysis and visualization. The Project Raijin team is excited to provide the student with an in-depth experience of working within a professional software engineering team; therefore, the student will participate in all project development activities such as regular team meetings, morning standups, hackathons, cross-team discussion/collaboration meetings, debugging and bug-fixing, pair programming, documentation, etc.

Students: The project is open to undergraduate students.

Skills and Qualifications:
- Experience with Python programming
- Familiarity with Jupyter Notebooks
- Ability and willingness to work with a team
- Good communication and writing skills

Optional Skills and Qualifications:
- User-level familiarity with Linux and Unix-based tools for scripting and file manipulation
- Experience with Xarray, NumPy, Matplotlib, Cartopy, Dask, Numba, Datashader, Holoviews
- Familiarity with unstructured grids/meshes

NOTE This project has the possibility of an extension for a second summer.

Project 9. Python reimplementation of Fortran subroutines for use on supercomputers

Areas of Interest in order of relevance: Software Engineering, Application Optimization/Parallelization, Supercomputer Systems Operations

Description: The project is to reimplement existing Fortran subroutines into Python for use with NCAR's supercomputing resources.

The GeoCAT project is an open development, community-owned effort, managed by the National Center for Atmospheric Research (NCAR). Its primary goal is to produce Python based tools that help make sense of geoscience data. GeoCAT-comp is the computational component of the GeoCAT project. Many of the computational functions in GeoCAT are implemented in a pure Python fashion. However, there are some others that were implemented in Fortran but are wrapped up in Python as part of GeoCAT-comp (namely, GeoCAT-f2py). The GeoCAT team considers replacing the Fortran-based computational routines with pure Python implementations to solely leverage the scientific Python ecosystem capabilities.

Over this summer internship, the student will explore and learn about how the existing Fortran-based functionalities can be re-implemented in pure Python with the help of Dask parallelization on high performance computing environments such as NCAR's Cheyenne Supercomputer. The student will have the opportunity to implement algorithms using commonly used packages in the scientific Python ecosystem such as Xarray, Numpy, Dask, etc. The student will test the scalability performance of the implemented functions on Cheyenne using unit tests on large datasets.

Students: The project is open to undergraduate students.

Skills and Qualifications: Skilled with writing in Python 3, willingness to learn to read Fortran.

NOTE This project has the possibility of an extension for a second summer.

Project 10. Using Binder as a fast and scalable educational platform

Areas of Interest in order of relevance: Cloud Computing, DevOps, Software Engineering

Description: Binder (mybinder.org) is a cloud-based technology for launching interactive notebooks in their own environment on cloud platforms. Binder makes sharing Jupyter Notebooks extremely easy: you create a GitHub repository containing your Jupyter Notebooks, a definition of the environment the notebooks need, and a simple Binder configuration file, and BinderHub (e.g., mybinder.org) does the rest! Following a properly-formed URL pointing to the BinderHub, the BinderHub will spin up cloud resources (e.g., EC2 instances), build Docker containers, and launch a JupyterLab instance with the ability to interactively run the Jupyter Notebooks in the "binderized" GitHub repository. The ease with which Binder makes sharing interactive notebooks is tremendous, but the speed with which the process works (spinning up cloud resources, building Docker images, etc.) can be quite slow, especially on freely-provided hardware. In this project, we would like to investigate how to optimize the Binder launch process, making the "access to the platform" as fast as possible. If time permits, this project will also develop additional features for our optimized Binder experience, such as how to allow free "public" usage as well as authenticated "privileged" access (possibly for the same notebooks, but with different user capabilities), Binder on high-performance computing systems like NCAR's NWSC supercomputer, and more.

Students: The project is open to undergraduate students.

Skills and Qualifications:
-Experience with Python
-Experience with running Jupyter Notebooks
-Experience with Linux operating systems
-Experience with git and GitHub
-Experience building Docker containers
-Some knowledge of cloud computing platforms like AWS, GCP or Azure
-Eagerness to learn new technologies

Optional Skills and Qualifications:
-Experience with cloud computing platforms like AWS, GCP or Azure
-Knowledge of web APIs
-Knowledge of HPC-friendly containers like Singularity

Project 11. Apply Search Techniques for Scientific Data Discovery and Exploration

Areas of Interest in order of relevance: Software Engineering

Description: NCAR's diverse scientific data holdings have historically been difficult for external scientists and users to search across and find the data they need to do their science. While we have a current search system that aggregates these data holdings, we are experimenting with a simpler approach.

This project is focused on moving a new scientific data search Java-based web application towards deployment. This summer we're enhancing our current software by including search facets, improving the usability of our web front-end, making our front end more responsive to both large and small screen internet enabled devices, and storing and reading more metadata from our Solr search platform. We also are focusing on validating incoming scientific metadata for completeness and possibly notifying data providers when metadata is incomplete.

This project does NOT involve artificial intelligence or machine learning.

Students: The project is open to undergraduate and graduate students. There is one (1) open undergraduate position and one (1) open graduate position.

Skills and Qualifications:
For both undergraduates and graduates: Basic understanding of software development programming. Basic experience in languages such as Java, Javascript, XML, HTML, CSS, and a query language like SQL. Basic understanding of web services and web based user interfaces. Basic understanding of controlled vocabularies and metadata schemas. Ability to interact with diverse mentors and peers in a friendly, professional manner that supports collaboration and inquiry. Good problem solving skills. Good oral and written communication skills. Willingness to learn and use software development tools and programs. Curiosity to explore new things.

Graduate applicants additional required skills: Please check back!

Project 12. Development of a Python Knowledge-based weather station interpolation algorithm

Areas of Interest in order of relevance: Geostatistics, Data Science, Software Engineering

Description: Scientists need gridded observational weather data for a wide variety of applications, including weather and climate model validation, hydrologic modeling, and climate model downscaling. However, converting point-based observations from irregularly-placed weather stations to values on a uniform high-resolution grid is challenging, especially in areas of complex terrain, and many different methods have been developed to create these data products. The Topograpically InformEd Regression (TIER) method implements a well-known, knowledge-based interpolation system in an open-source repository (https://github.com/NCAR/TIER). It makes a popular method more widely available, and allows users to easily experiment with aspects of the interpolation routine to better understand the impacts of different options such as algorithmic decisions on the final interpolation product and associated uncertainties. These decisions are typically hidden from users with little description of the sensitivity of these choices on the quality and character of a product.

We are looking for an interested student to accomplish several goals that are flexible depending on experience and progress throughout the summer:

1) Port the MATLAB code to Python
2) Add parallelization (e.g., Dask) for use on clusters and other high performance computing environments
3) Add additional interpolation methodologies in a user-friendly manner
4) Develop documentation, test cases and tutorial material

This project will introduce the student to geostatistical interpolation routines, development of gridded meteorological products, and contributing to a community resource. The updated TIER repository will enable users to experiment with new interpolation methods, better contribute to the TIER code-base by using a more popular and free programming language, and make TIER more efficient for large spatiotemporal domain processing.

Notes:
This project could also be tailored as a two year project. If we were able to complete task 1 or tasks 1 and 2 above in year 1, that would be great, and lay the ground work for adding parallelization and/or methods in year 2 depending on the year 1 progress.

Students: The project is open to graduate students.

Skills and Qualifications: Experience with Python and scientific computing with Python (e.g. Xarray, numpy, Pandas, etc), cluster computing and parallelization. Would be nice to have experience with Dask, netCDF, MATLAB, interpolation routines or geostatistics

Project 13. Exploring performance of GeoCAT data analysis routines on GPUs

Areas of Interest in order of relevance: Data analysis, Application Optimization/Parallelization, Software Engineering

Description: With recent supercomputers adapting to use GPU accelerators as the primary resource, the general purpose computing of many models are being optimized for both performance and power consumption. However, data analysis is yet to take full advantage of GPGPU computing. The goal of this SIParCS 2022 project is to take advantage of GPGPU computing capabilities for the GeoCAT data analysis library.

The GeoCAT library is written in python and packaged with conda. Most of the functions in GeoCAT are either serial or take advantage of Dask to parallelize the computation on CPUs. This project wants to take advantage of GPU-Python packages like CuPy, Rapids, Numba, and Dask-GPU to run the compute intensive parts of GeoCAT examples on GPU. The primary goal of the student is to adapt GPU-Python packages into GeoCAT computation routines and evaluate the performance. Additionally, the student will be working closely with the GeoCAT team to provide a path to port and optimize other GeoCAT computation routines.

Students: The project is open to graduate students.

Skills and Qualifications:
- Experience with GPU programming in any language
- Experience with Python programming
  - Xarray (or Pandas)
  - Numpy
  - CuPy
  - Numba
- Experience with Linux environment
- Github
- Ability and willingness to work within a multidisciplinary team
- Good communication and writing skills

Highly desirable skills:
- Software documentation
- Experience with Python packages like matplotlib or scipy
- Experience with Dask parallelization in Python

Project 14. Future shock: Speed and Scalability in Data Assimilation

Areas of Interest in order of relevance: Application Optimization/Parallelization, Software Engineering, Data Science

Description: The Data Assimilation Research Testbed (DART) is a widely used community software facility for data assimilation. One application of data assimilation is improving numerical weather prediction. To do this, DART ingests a group of model forecasts, say 80 predictions of weather in the United States, and uses statistics to combine these model forecasts with observations to produce a better estimate of the weather.

There are two opposing data decompositions within DART. For forward operators, i.e. what the model predicts an observation should be, it is better to have spatially local data on the same processor. For load balancing in the assimilation phase, you want to spread the spatially local data across as many processors as possible. Additionally, observations can be irregular in space and time. For example, satellite observations are very dense, but are only available when the satellite passes over an area. Processors have to be aware of which observations have already been assimilated, and which observations are next. Together, these make a challenging and interesting parallel computation problem.

The project will start with profiling of the DART code on NCAR's supercomputer. The student will use the profiling results to guide the direction of the project:

The student may choose to simulate the computational and communication load for a given set of observations, and use this predictive model to solve for the optimum observation layout on a given set of processors. This work would provide a valuable tool to evaluate various data decompositions for novel models and future high resolution observation networks.

The student may choose to explore ways to improve DART observation assimilation runtime and scalability on NCAR's supercomputers. This could include algorithmic changes, for example work stealing; or redesigning data structures for better parallelism; or offloading computation to GPUs. Work that improves the DART runtime would be impactful for science, opening up billions of satellite observations that are available daily, and provide the student with an opportunity to design code to run efficiently on hundreds of thousands of processors.

Students: The project is open to graduate students.

Skills and Qualifications: Experience with a compiled language such as Fortran, C, or C++, CUDA. It would be good to have experience in the following libraries: MPI, openMP, openACC

Project 15. Integrating a Lidar Sensor into a Low-cost Weather Station

Areas of Interest in order of relevance: Geostatistics, Data Science, Electrical and Mechanical Engineering

Description: Abundant, high-quality hydrometeorological measurements are critical to Earth System Science, including weather forecasting, climate measurement and modeling, and streamflow predictions. However, the high cost of such measurements has meant that only a limited number are available. Lowering the cost of measurements enable more people to participate in the collection of such data, and it enables scientists to install far greater numbers of instruments, thus increasing community engagement and better measuring the real world. Of all the observations needed, snow is a particularly important component of the earth system that is surprisingly challenging to measure. Snowfall is not well measured by traditional precipitation measurements, snow on the ground is highly variable in space, and both of these are due to the interaction between wind and snow (which is challenging to measure itself).

The use of low-cost consumer electronics (e.g. the raspberry pi) and the rise of lower cost lidar sensors designed for the self-driving car industry, present an opportunity to significantly improve snow measurements. In this project, a student will learn how to program a raspberry pi to control a scanning lidar sensor for optimal data collections and develop techniques to process the large datasets collected. The lidar sensor is able to map snow on the ground, as well as individual snow-flakes in the air, returning a point cloud with 100s of thousands of points measured per second. Processing this dataset is a computational challenge due to the large size, and due to the unstructured nature of the data.

Students: The project is open to graduate students.

Skills and Qualifications: The student will have the opportunity to work with NCAR scientists to learn new techniques for processing data on GPUs, or on traditional high performance computing platforms (NCAR's Cheyenne supercomputer.) Code will be written in some combination of C/C++, Fortran, or Python, and will make use of an existing lidar SDK, and optionally cuda, OpenCV, Tensorflow, Dask, or other big-data processing tools.

Project 16. Leveraging Xarray for reproducible and scalable analysis of remote sensing data in the cloud

Areas of Interest in order of relevance: Data Science, Reproducible Science, Software Engineering

Description: Science today requires software that enables expressive and easily-parallelized workflows on gigabyte to petabyte sized datasets. Xarray is an actively developed open source library that provides scientists with a powerful interface for parallelized computation with multi-dimensional raster datasets (e.g. image stacks), which are prevalent today across all scientific domains. While Xarray was initially designed to work with small data files on personal laptops or servers, a paradigm shift in scientific computing is underway and modern workflows can now leverage Xarray to analyze massive cloud-hosted archives such as NASA's Earth observation archive. Come work with us to transition your research workflows to the Cloud, learn new technical skills, and become an open source contributor!
Over the summer, you will have opportunities to: Collaborate with a team of research scientists, data scientists, and software developers to produce publicly-accessible tutorials that leverage Xarray for scientific analysis of Cloud-hosted remote sensing data. Learn about and contribute to multiple open source geoscientific Python projects (particularly Xarray and RioXarray) as well as general open source tools such as JupyterBook. Learn to access and effectively use cloud-based datasets and computational resources. Gain experience with collaborative software development workflows via GitHub. Understand the technical components of reproducible computational workflows including testing, continuous integration, and efficiently sharing datasets and analysis results. You'll learn skills that are key to practicing open science and that are transferable to both academic and non-academic career paths.

We welcome applications from any intern interested in cloud-hosted, geospatial data, and open science and will tailor the project and specific learning objectives to the intern's interests and experience.

Students: The project is open to graduate students.

Skills and Qualifications:
Experience with basic Python programming.
Familiarity with Jupyter Notebooks.
Ability and willingness to work with a team.

Optional Skills and Qualifications:
Exposure to the Python data-science stack (NumPy, Xarray, Pandas, etc...). Exposure to git, GitHub, GitHub Actions continuous integration. Experience working with geospatial data (coordinate reference systems, GIS)

Project 17. Pi-WRF Community Driven Educational Modules and Classroom Activities Development

Areas of Interest in order of relevance: Education, Software Engineering, Data Science

Description: The goal of this project is to develop educational modules and build a framework to facilitate further community contributed modules involving running a real weather model on a Raspberry Pi to make a weather forecast. Users will run the simulation for specified days, aerial coverage, visualize the output, and make a forecast. During this project the intern will develop instructional modules to encourage k12 teachers and university faculty to build modules and contribute them to a community collection of Pi-WRF teaching boxes. Our vision is that Pi-WRF educational module development becomes a community driven effort in which we provide the tools and example resources to drive community engagement.

During this project the intern will develop modules that will help users and community contributors gain an understanding of how a weather forecast is made, how a meteorologist adds “value” to the forecast, analyzes and reports forecast results, and communicate what some of the limits and shortcomings of some of our latest numerical weather prediction models. During this project the intern will help make it easier for community members to contribute content and become more engaged.

This project will require a multi-disciplinary approach and unconventional solutions to create educational modules and activities that address Next Generation Science Standard (NGSS) while engaging k-12 audiences and beyond. We are seeking candidates who are creative thinkers, artists, technicians, and communicators to create engaging educational modules and activities. Introductory modules will include topics such as “what is weather forecasting and how are they made?” and “what is numerical weather prediction?”. Activities and modules such as “Tools to Make Weather Forecasts” would bridge to mesoscale models running on Raspberry Pi with the WRF model.

Students: The project is open to graduate students.

Skills and Qualifications: This project builds on Jupyter notebook, JupyterBook, and a dockerized Pi-WRF package. We will be utilizing Raspberry Pi systems which will require experience or willingness to learn to setup, configure, docker, github, or similar computing environments. Graphic design, UI/UX, adobe photoshop, or expertise in other communication mediums is essential for creating the most effective, fun and creative Pi-WRF modules. Understanding this environment will help develop a technical bridge to PiWRF.

Project 18. Using containers and Spack to simplify the portability and reproducibility of scientific applications

Areas of Interest in order of relevance: Portability, Reproducibility, Containers, GPU applications, Software Engineering

Description: Modern scientific applications are usually complex regarding their functionalities and installation procedure. For example, building a scientific application may require multiple additional software packages that are dependent on other software packages, compilers and even architectures. Large software dependency trees make for difficult installations and also hamper the ability to reproduce results. How to handle module dependency and compatibility across multiple platforms is typically a big challenge for the scientific researchers and can even prevent the use of high-productivity language frameworks. A viable solution to address the portability and reproducibility of scientific applications is through the use of containers. Containers allow for the encapsulation of the necessary software packages into a customized environment. In this way, other researchers could easily and quickly use the same scientific application on a different platform with the same container image file and reproduce the results for verification.

The goal of this 2022 summer internship is to improve the portability of a scientific application important to the research of the Application Scalability and Performance (ASAP) group on different computing platforms. Initially, a C++ based application with GPU enablement through use of OpenACC directives will be utilized. This project will likely use Singularity containers augmented with the Spack package manager to simplify the execution on several different platforms like the Casper and Derecho Linux clusters at NCAR and Amazon Web Services. The student's primary focus will be building a container environment with the necessary compatible packages installed by Spack. The student will also run this container image on different machines to verify the portability and evaluate the performance of the scientific application across the platforms. Moreover, the student may also merge the existing containers from the ASAP Group into a single one so that it is easier to deploy important benchmark tests on the other computing platforms.

Students: The project is open to graduate students.

Skills and Qualifications: Familiarity with high performance computing (HPC) cluster, linux environment and make/cmake is required. Strong motivation to learn new skills and resolve issues in a team is required. Experience with container runtimes (e.g., Singularity or Docker) is preferred. Experience with Spack, GPU programming and cloud computing is desirable.