2021 TECHNICAL PROJECTS

Undergraduate

Project 1. Expanding the GeoCAT-Examples Visualization Gallery (2 undergraduate positions)
Project 2. Machine Learning Data Commons Web Portal
Project 3. Machine Learning to Improve Weather Forecasts through Improved Data Assimilation
Project 4. Python integration of NCL Fortran Code for GeoCAT
Project 5. Understanding HPC Application Power Efficiency, System Power Controls, and Impacts

Undergraduate and Graduate

Project 6. GPU-Accelerated Insitu Analysis of Weather and Climate Model Data (1 undergraduate position and 1 graduate position)
Project 7. Software Engineering and Application Development for GDEX-Obs: Enabling Scientific Data Discovery and Use (1 undergraduate position and 1 graduate position)

Graduate

Project 8. Enabling Zero-friction Reproducible Analysis of Gridded Datasets with Xgcm
Project 9. Performance Portability of Weather and Climate Modeling Mini Apps (2 graduate positions)
Project 10. Pi-WRF Educational Modules and Classroom Activities Development
Project 11. Test Cloud Optimized Data Formats for Swath (L3) Satellite Data
Project 12. Using Machine Learning to Dynamically Allocate Resources for Data Delivery

NON-TECHNICAL PROJECT

- CISL Outreach, Diversity, and Education. Please see https://www2.cisl.ucar.edu/siparcs/cisl-outreach-diversity-and-education-code-intern

Updated 10/20/20 VD

Project 1. Expanding the GeoCAT-Examples Visualization Gallery

Areas of Interest in order of relevance: Visualization, 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, creating the GeoCAT-Examples Gallery. During the Summer of 2020, the GeoCAT-Examples Gallery was expanded significantly and has been continuously updated with new NCL to Python projection examples.

Over this summer internship, the student will explore and learn about data visualization in the atmospheric and oceanic sciences using matplotlib, cartopy, and holoviews. The student will generate plotting templates inspired by the NCL gallery including: box plots, contour plots, (r,theta) radar plots, vector plots, and 3-dimensional plots. The student will plot using different map projections and with different overlays of satellite, measured, and modeled data. Additionally, there will be opportunities to contribute to the GeoCAT-viz utility function library. This library contains utility functions created to reduce the amount of boilerplate code in the NCL to Python visualization scripts. Over the course of the internship, we will see the library of python plotting examples grow.

The student will summarize their accomplishments with: a publishable gallery of visualization examples and Python utility functions accessible on GitHub (if applicable).

Students: The project is open to undergraduate students. There are two (2) open positions for this project.

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. Ability and willingness to work with a team. Good communication and writing skills.
Optional Skills and Qualifications: Familiarity with NCL (NCAR Command Language). Experience with NumPy, Matplotlib, Cartopy, and Xarray.

Project 2. Machine Learning Data Commons Web Portal

Areas of Interest in order of relevance: Data Science, Visualization, Digital Asset Management

Description: The CISL Analytics and Integrative Machine Learning group is developing a set of machine learning challenge problems for the Earth System Science community to support both machine learning education and broader research on these problems. The challenge problems include predicting lightning from satellite, emulating microphysics and chemistry models, and identifying cloud and rain drops in holographic cloud particle imager data. To reach the community, we are developing a web portal that documents each challenge problem, provides access to the data, and connects to web-based notebook services for interactive experimentation with the different datasets. The intern will help develop and evaluate the web portal and Jupyter notebook tutorials for each challenge problem. They will help develop tutorial notebooks with the project team, export them to different notebook platforms including Binder, Google Colab, and Kaggle Notebooks, and test the performance of the challenge problems in each notebook environment. The intern will have the opportunity to work closely with the AIML group as well as our partner scientists in other NCAR laboratories.

Students: The project is open to undergraduate students.

Skills and Qualifications: The candidate needs to have prior experience with Python, Jupyter notebooks, and at least basic website development skills. Prior experience with machine learning, advanced website development skills, and Git/Github is desired.

NOTE Undergraduate students may only apply to one machine learning project (Projects 2 and 3). Students may apply to either Project 2 or Project 3, but not both, as part of their two (2) choices. If applying to a machine learning project, the other choice must be a non-machine learning project.

Project 3. Machine Learning to Improve Weather Forecasts through Improved Data Assimilation

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

Description: The Data Assimilation Research Testbed (DART) is a mature and widely used community software facility for data assimilation. One application of data assimilation is improving numerical weather prediction (NWP). An atmospheric model prediction is run on a supercomputer with hundreds or thousands of nodes, and the output from this model is then statistically combined with atmospheric measurements such as temperature or winds. The measurements may also be from much more sophisticated instruments like radars or satellite radiometers. The process of combining the model forecast and observations is known as data assimilation.

95% of the measurements of the atmosphere potentially available for data assimilation come from meteorological satellites orbiting the earth. However, these observations remain underutilized due to a variety of issues such as uncertainty regarding water and ice clouds. According to some estimates, more than 70% of these satellite observations are currently discarded by NWP forecasting centers around the world due to “cloud contamination.” Yet clouds are precisely where the severe weather is occurring. This project aims to help address these issues by combining machine learning techniques with data assimilation to enable the effective assimilation of more observations.

This project will use machine learning and satellite observations to find and exploit relationships between atmospheric elements like water vapor, temperature, and clouds. These extracted relationships will then be input into data assimilation systems in order to improve NWP, where improved forecasts could save life, limb, and property. These relationships also have the potential to benefit climate studies and improve the forecast models that are used for weather and climate prediction.

As there are billions of new satellite observations available every day and because the models involved work with billions of variables, this is a “big data” problem that is highly relevant to today’s world. This internship is a chance to apply machine learning packages and techniques to an important unsolved problem with substantial real-world consequences.

Students: The project is open to undergraduate students.

Skills and Qualifications: Programming (Python, Matlab, or similar), machine learning, data analysis

NOTE Undergraduate students may only apply to one machine learning project (Projects 2 and 3). Students may apply to either Project 2 or Project 3, but not both, as part of their two (2) choices. If applying to a machine learning project, the other choice must be a non-machine learning project.

Updated 10/20/20 VD

Project 4. Python Integration of NCL Fortran Code for GeoCAT

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

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 set a course to migrate many of NCL's highly specialized capabilities into the Python ecosystem. NCL's Fortran subroutines are of particular interest. The NCL GitHub (https://github.com/NCAR/ncl) contains a library of Fortran subroutines. The lack of an equivalent resource in Python is one of the most common concerns expressed by NCL users transitioning to Python. Currently, scientists have to alternate between NCL and Python to complete their work.

Over this summer internship, the student will explore and learn about how high performance Fortran code can be used within Python and Dask on the Cheyenne Supercomputer. The student will use Numpy's f2py utility to build compiled Fortran modules for import into Python. The student will test the performance of the compiled programs on Cheyenne using unit tests on large datasets. Over the course of the internship, we will see the library of Python's importable NCL subroutines grow.

Students: The project is open to undergraduate students.

Skills and Qualifications: Experience with Python programming. Familiarity with Numpy. User level familiarity with Linux and Unix-based tools for scripting and file manipulation. Ability and willingness to work with a team. Good communication and writing skills.
Optional Skills and Qualifications: Familiarity with NCL (NCAR Command Language). Experience with Dask, Xarray, and Conda.

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

Project 5. Understanding HPC Application Power Efficiency, System Power Controls, and Impacts

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

Description: Utility costs associated with NCAR's primary HPC systems approach $2 million per year, and reducing utility costs and the accompanying environmental impact are of significant interest to NCAR and other climate and weather centers. In this project, the student will first conduct an assessment of the options available for power consumption information and control within NCAR's Cheyenne environment, including the data available and data collection mechanisms. The student will then perform a series of experiments on applications in the typical NCAR workload, running a controlled set of HPC jobs to collect related power data. These experiments will determine the impact of various power controls on both the flops/watt for each application and the scientific throughput. Combined with data on the characteristics of the typical Cheyenne workload, the student will assess the possible impacts and trade-offs of using the various power controls, as well as provide input into a qualitative evaluation of the potential implementation of power controls in the production HPC environment.

Students: The project is open to undergraduate students.

Skills and Qualifications: A candidate for this position will benefit from a range of skills, but a candidate does not need to possess all the preferred skills, nor possess them at an expert level. It is expected that candidates will be able to learn some skills during the summer.

Experience using some flavor of Linux/Unix preferred. Experience using a shared, batch scheduled computer system preferred. Experience with a compiled programming language preferred. Experience using and querying databases preferred. Experience with a scripting language (Python, PHP, Perl, etc.) preferred.

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

Project 6. GPU-Accelerated In Situ Analysis of Weather and Climate Model Data

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

Description: This project will continue work that began as a 2020 SIParCS summer project. With recent software developments, it's possible to perform end-to-end data simulation and analysis entirely on GPUs. This technique reduces the need for data storage and unnecessary input/output (I/O), which remain challenging bottlenecks for high performance computing. Application Programming Interfaces (APIs) such as CuPy and Dask will be used, along with other in situ processing libraries.

The project will focus on in situ analysis of data on GPUs, including topics such validation and feature detection for extreme weather events. The starting point of this 2021 summer internship will be to take post-processing scripts used for the analysis of a certain weather and climate model's output, and offload the analysis calculations to GPUs. The goal is to perform the data analysis in situ on the model's gpu-resident datasets. The student's primary focus will be on using CuPy, (or a similar library), that supports classic analysis techniques as well as machine learning approaches to analyze large datasets on GPU memory.

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: Undergraduate: Good programming skills with at least one of the following languages - C, C++, and/or Fortran - is required. Familiarity with Python scripting. A basic understanding of parallel programming and knowledge of GPGPU. Familiarity with Numpy, CuPy, or Dask is considered a bonus.
Graduate: Strong programming skills with at least one of the following languages - C, C++, and/or Fortran - is required. Strong with Python scripting. Experience working with parallel programming and knowledge of GPGPU. Experience with Numpy, CuPy, or Dask is considered a bonus.

Project 7. Software Engineering and Application Development for GDEX-Obs: Enabling Scientific Data Discovery and Use

Areas of Interest in order of relevance: Software Engineering, Digital Asset Management, Data discovery, metadata and standards

Description: NCAR is establishing an archival repository for data that will be produced by NSF-funded Community Instruments and Facilities (CIF). These CIFs will make observations of the Earth's atmosphere. NCAR will expand and revise its existing Digital Asset Services Hub (DASH) Repository to serve as a "Geoscience Data Exchange for Observations" to house these data.

This software engineering project will explore extending the DASH Repository and DASH Search capabilities to support file and data object level discovery. We will explore search platforms such as Apache Solr and Elasticsearch to improve data file and data object search for our users. We may explore areas such as query suggestions, spelling, search term autocomplete, suggested hints, synonyms, and other semantic enhancements. We will explore deployment approaches using container technologies such as Docker and Docker Compose.

The project will be highly collaborative and team based, using the Agile Scrum methodology. SIParCS Interns will work together and with an existing software engineering team and stakeholders to organize and prioritize work. This work will be an opportunity to add new end user features to an existing production application and work with a software engineering team in an Agile model and learn software engineering practices.

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: Understanding of programming with experience in programming languages such as Java, Python or Javascript. Basic understanding of XML and HTML markup languages. Ability to interact with mentors and peers in a manner that supports collaboration and inquiry. Ability to work with diverse staff. Good problem solving skills. Good oral and written communication skills. Willingness to learn and use computing tools and programs. Curiosity to explore new things.

Graduate applicants additional required skills: Experience with the Java programming language. Understanding of XML and HTML markup languages. Understanding of controlled vocabularies and metadata schemas. Understanding of web services, dynamic and web UI. Understanding of query languages such as SQL or Solr query syntax.

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

Project 8. Enabling Zero-friction Reproducible Analysis of Gridded Datasets with Xgcm

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

Description: The age of big data in science comes with challenges. The sheer increase in available datasets demands that scientific analysis tools be general and frictionless. [Xgcm](https://github.com/xgcm/xgcm) is an open-source package specifically designed by scientists for convenient and fast grid-aware analysis of large gridded datasets, both observational and numerical. Xgcm unifies the analysis of gridded datasets in one general software package and builds on many tools in the scientific Python stack (xarray, numpy, dask, numba), making Earth Science more open, interactive, convenient, and fun.

Over the summer, the intern will: Learn about and contribute to multiple open source geoscientific Python projects (particularly xgcm and [cf_xarray](https://github.com/xarray-contrib/cf-xarray)). Enhance scientific productivity and the usability of xgcm by teaching it to understand standard metadata conventions that are in use. Increase the reproducibility of xgcm-based analysis code, by teaching xgcm to automatically record information about processing steps in dataset metadata. Engage with multiple open source projects on GitHub and become familiar with the GitHub Pull Request contribution workflow (if not familiar already). Create cloud-executable Jupyter notebooks demonstrating this newly-implemented functionality using public gridded datasets available in the cloud.

We will mentor you in general skills like open source development, git workflows, reproducible demonstration notebooks, and continuous integration testing, that will be of benefit in many modern career paths.

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. Experience working with gridded datasets.

Project 9. Performance Portability of Weather and Climate Modeling Mini Apps

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

Description: The computational capacity of high-performance computing platforms has increased rapidly during recent years, particularly due to the widespread adoption of GPUs. GPUs are naturally fitted for problems whose computations could be highly parallelized. As compared to CPUs, GPUs could also save energy to get the same amount of work done. More interests and collaborations from domain scientists and computational scientists have emerged to explore the portability of weather and climate models to GPU. However, multiple vendors have different GPU architectures and it is more desirable to execute the same codes on different GPU platforms with limited modifications. Hence this project will aim to assess the performance portability of some Weather and Climate mini applications.

The goal of this 2021 summer internship will focus on the programming model's performance portability for a certain weather or climate. The student's primary focus will be on evaluating the choice of programming models like Intel's OneAPI, Kokkos, HIP programming, OpenMP-GPU to achieve portability across multiple CPU-GPU platforms including but not limited to Intel-Xe GPUs, NVIDIA GPUs and AMD GPUs. Additionally, the student will also evaluate the performance on various hybrid platforms.

Students: The project is open to graduate students. There are two (2) open positions.

Skills and Qualifications: Strong programming skills in C, C++ is required. Strong motivation to learn new skills and resolve issues in a team is required. Experience with high-performance computing facilities is preferred. Experience with directive-based parallel programming models such as OpenMP, OpenACC, and knowledge of GPGPU is preferred. Familiarity with SYCL standard or OpenMP-GPU is considered a bonus.

Project 10. Pi-WRF Educational Modules and Classroom Activities Development

Areas of Interest in order of relevance: Science Education, Numerical Weather Prediction. Visualization

Description: The goal of this project is to develop educational modules and activities around running a real weather model on a Raspberry Pi to make a weather forecast. Users can run the simulation for specified days, aerial coverage, visualize the output, and make a forecast. During this project the intern will develop modules that will help users 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 some of the limits and shortcomings of some of our latest numerical weather prediction models.

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 forecasts 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.

This project will build a framework for introducing weather forecast models to K-12 by building on existing literature and approaches. Students will develop an evaluation mechanism and a set of performance evaluation criteria to empower teachers and administrators with metrics about how learning fulfills NGSS.

On the technical side, 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.

Students: The project is open to graduate students.

Skills and Qualifications: The intern will have experience with some of the following: science education, science writing, graphic design or educational illustration, user interface/user experience, Raspberry Pi, Adobe photoshop.

Project 11. Test Cloud Optimized Data Formats for Swath (L3) Satellite Data

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

Description: Orbiting satellite data are stored in swath (Level-3) data files that include the time, location, and data at each point (https://earthdata.nasa.gov/collaborate/open-data-services-and-software/data-information-policy/data-levels). NASA and NOAA are moving vast amounts of satellite data to the cloud. Initial efforts have focused on the gridded Level-3 data which already have some clear pathways to produce cloud optimized versions of the data. The capabilities of cloud infrastructure have the potential to improve access to and facilitate analysis of large Earth Observation datasets. For these capabilities to be fully met, data may need to be reorganized, restructured, or reformatted for easier manipulation in cloud-native analysis-access workflows.

Over this summer internship, the student will explore and learn about the best cloud-optimized format for this data by staging a sample of the MODIS L3P data on the Pangeo cloud, transforming that data into different formats (eg. Zarr, cloud-optimized HDF), and testing access and analysis times for typical workflows (e.g. time series analysis at a specific location or over a specific region, spatial analysis at a specific time).

The student will summarize their accomplishments with a notebook gallery demonstrating several workflows accessing different cloud-optimized data.

Students: The project is open to graduate students.

Skills and Qualifications: Intermediate Python programming experience (Xarray, matplotlib). Familiarity with Jupyter Notebooks. Familiarity with Linux. Ability and willingness to work with a team. Good communication and writing skills.
Optional Skills and Qualifications: Familiarity with Cloud resources and formats

Project 12. Using Machine Learning to Dynamically Allocate Resources for Data Delivery

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

Description: Determining an optimal allocation of computational resources using workload managers can often be a difficult task. This challenge is amplified when the resource requirements of a process change depending on the nature of its input, as is the case with the Research Data Archive's (RDA) data subset and conversion tools. These tools allow users from around the world to submit both large and small requests to manipulate data that then need to be executed within NCAR's high performance computing (HPC) environment. When using workload managers such as Slurm, it is critical to appropriately estimate memory, CPU, and time limit constraints for each job in order to quickly complete user requests and not disrupt workflows of other scientists.

Over the course of the summer, the student will be tasked with designing and developing a machine learning model which can dynamically estimate resource allocation for RDA requests. The ML framework, programming language, and overall approach to this problem will be informed by the interests and expertise of the student. The project will provide an opportunity to work in a production environment and gain experience in HPC, ML/DL, data management, object storage technologies, and software development workflows.

Students: The project is open to graduate students.

Skills and Qualifications: Applicants should be comfortable in a Unix-like environment, have some experience with shell scripting, and be proficient in at least one programming language--preferably Python. Successful interns will have knowledge and interest in machine learning and applied statistics