# Using HPC to Solve PDEs for Improved Weather Forecasting

Gurvir K. Bawa, Jaclyn Welch | Embry Riddle Aeronautical University

## Abstract


- This project aims to improve weather forecasting by decreasing the time it takes to calculate partial differential equations (PDEs) using high performance computing (HPC)
- Several PDEs are associated with weather forecasting, and these PDEs are incredibly complex – integration of multiple questions, unknown functions, and multiple independent variables
- Highly complex problems are incredibly taxing to compute, requiring HPC for efficiency and time
- Modern weather data collection is faster than the data can be processed, leading to the waste of copious amounts of data as well as a decreased accuracy of results due to discarded relevant data
- This research uses Vega, a supercomputer, to perform HPC on the chosen PDEs and observes the faster processing's effect on weather forecasting and the difference in time to predict (see Figure 1)


## Terms

- PDE – Partial Differential Equation
- HPC – High Performance Computing
- IEM – Iowa Environmental Mesonet (out of Iowa State University)
- ASOS – Automated Surface Observing Systems
- AWOS – Automated Weather Observing System
- NWS – National Weather Service (in the United States)
- MPAS – Model for Prediction Across Scales
- MPI – Message Passing Interface
- KISTI – Korean Institute of Science and Technology Information
- KNL – Knight Landing
- GRIB – Gridded Binary File
- CSV – Comma Separated Values
- AWC – Aviation Weather Center
- I/O – (File) Input/Output
- NOAA – National Oceanic and Atmospheric Administration
- NCEI – National Centers for Environmental Information
- NAM – North American Mesoscale (Forecast System)

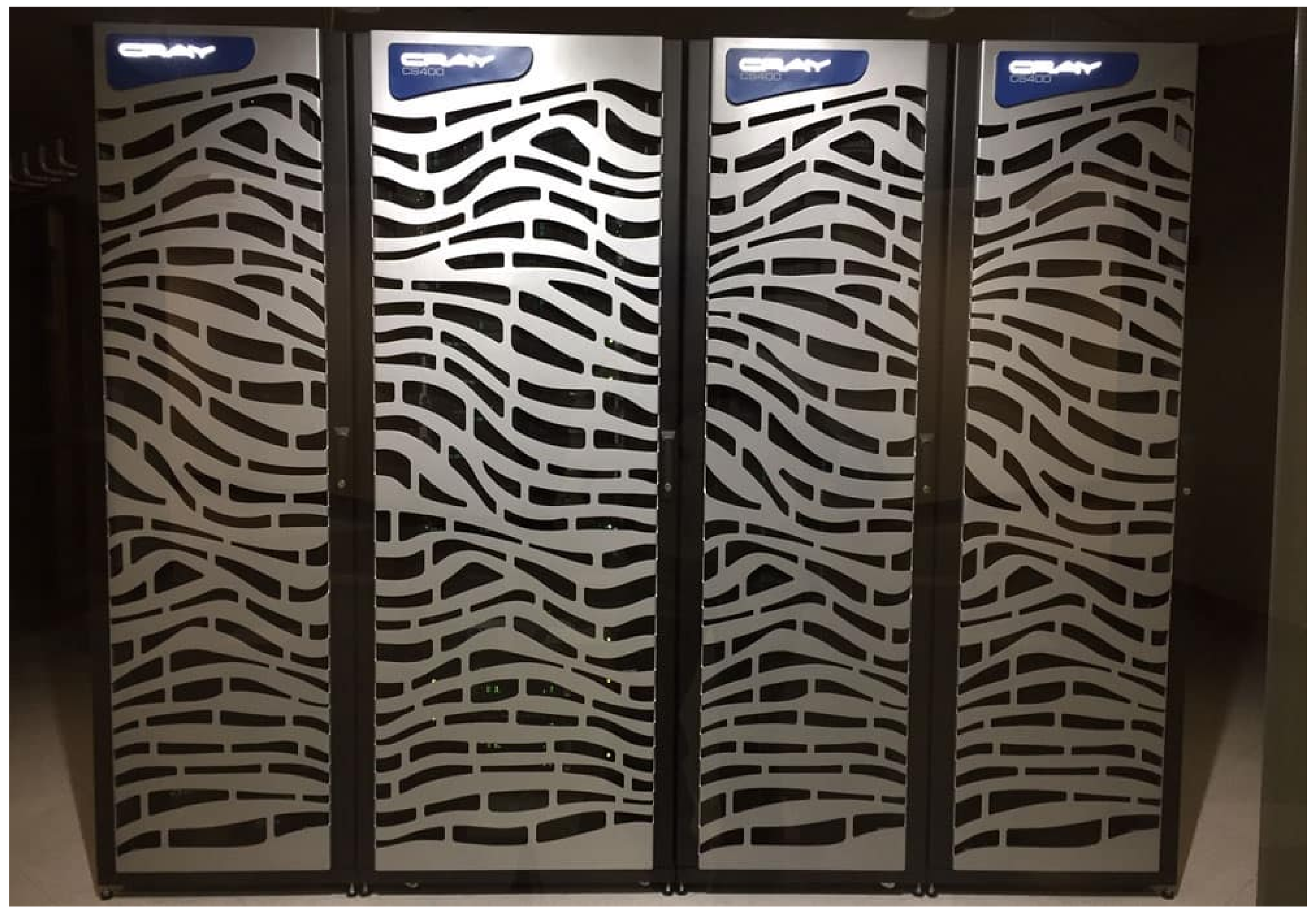

**Figure 1.** Pictured above is Vega, ERAU's four-cabinet Cray® CS400™ cluster supercomputer used for the HPC in our research. Vega has 84 compute nodes 2x E5-2697v4 18-core, 2.3GHz CPU's, 256GB RAM and EDR Infiniband (100Gb/s), 2 GPU nodes, each with 2 NVIDIA P100 GPU's 2x E5-2697v4 18-core, 2.3GHz CPU's, 256GB RAM and FDR Infiniband (56Gb/s), 2 login nodes each with 2x E5-2697v4 18-core, 2.3GHz CPU's, 384GB RAM, EDR Infiniband (100Gb/s) and 2x 10 GbE Ethernet uplinks and 240 TB of scratch storage in a high-performance parallel Lustre file system with a nominal bandwidth of 9 GB/s.

## Introduction

- To begin our research, we chose from a variety of publicly available weather datasets
- Weather data consists of many different variables, many of which are not relevant to our research
- The macroscopic state of the atmosphere is what is most relevant to the objectives of our research, can be described in seven variables[1]:
  - Three components of the wind velocity (u, v, and w)
  - Temperature (T)
  - Pressure (p)
  - Density (ρ)
  - Specific humidity (q)
- A closed set of seven *governing equations*, also known as *primitive equations* relate these variables and consist of[1]:
  - Three momentum equations (one for each coordinate direction)
  - A mass continuity equation
  - A water-mass continuity equation
  - A thermodynamic energy equation
  - An equation of state
- Through our review of the existing literature, we discovered several relevant studies to provide context and direction to our research
- The authors of [3] tested the scalability of a new global numerical prediction model MPAS on NURION, 5th supercomputer of KISTI and world's 11th fastest computer at the time it was launched in 2018
- In this study, MPI processes were used to integrate MPAS, potential I/O bottleneck issues were addressed by introducing a burst buffer and by adjusting stripe count for the Lustre file system
- Various experiments were performed throughout this study by changing the number of cores (512-4096) per node on KNL nodes
- This study highlights a need for technical improvement of I/O within massive computer clusters in addition to the development of better CPU cores
- Our study will mirror several aspects of this study using ERAU's supercomputer instead of the NURION supercomputer

## Data

- There are two main datasets we aimed to use due to their quick resolution time and their size:
- IEM ASOS/AWOS
  - Has minute and hourly data resolutions
  - Data is available in a csv format
- NOAA NCEI NAM Data
  - Has an output timestep of three hours, with +0 hour to +84 hours predictions
  - Data is available in grib2 format and would need pre-processing to convert it into an array
- Currently we are working with the NAM data because of its spatial properties, allowing us to isolate patterns in a specific area in the U.S.
  - We will be using C scripts run in parallel to drastically reduce the amount of data by removing unnecessary data and parameters
- Critical missing data is a challenge with any weather forecast data processing, and based on our literature study we will be either averaging the existing data to replace missing inputs, or applying spatial equations to predict the missing values within an acceptable margin of error

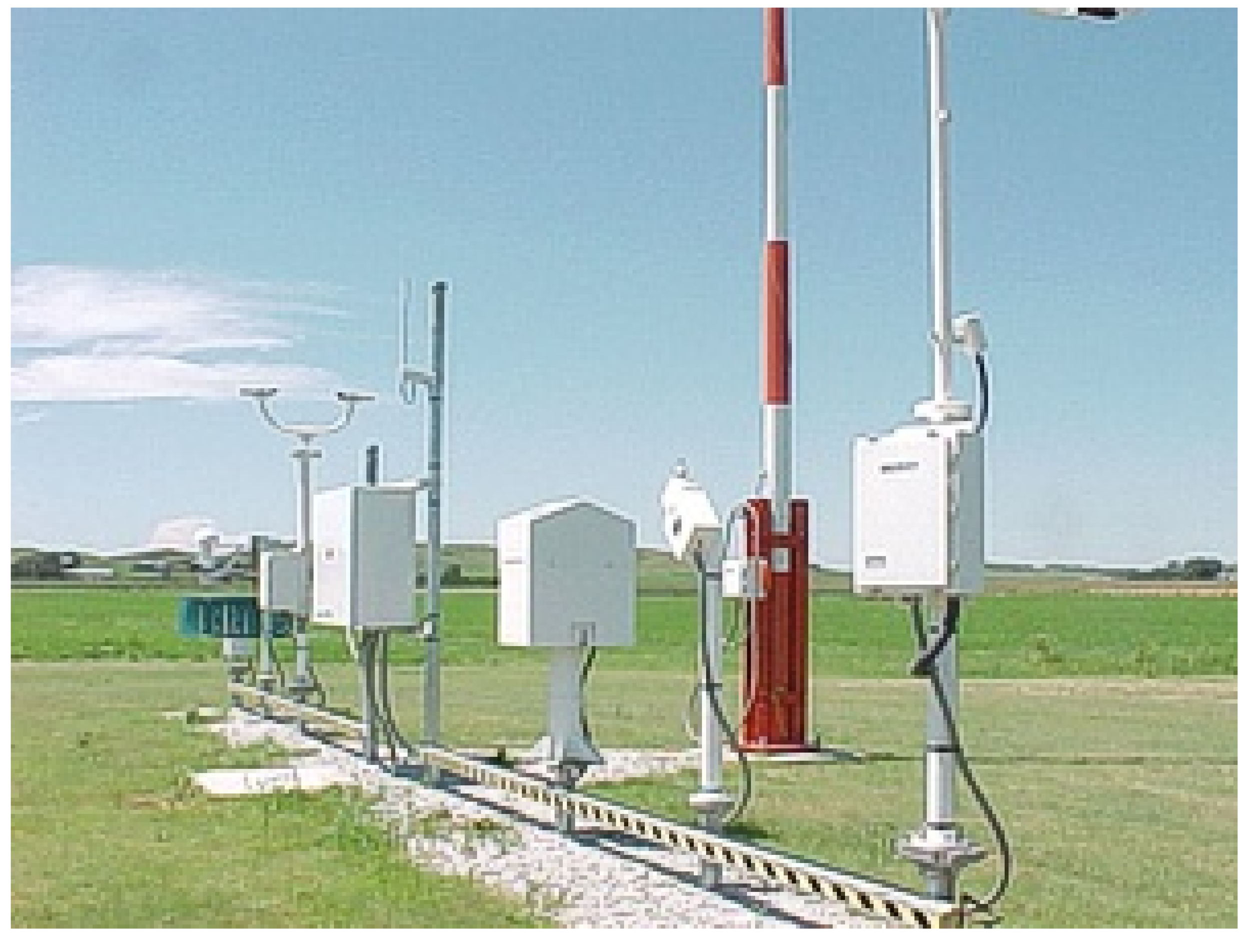

**Figure 2.** Above is a photograph of ASOS equipment recording weather data provided by NWS and NOAA. This equipment is located in the Raleigh area and runs non-stop, updating weather observations every minute 24 hours a day. The ASOS network has more than double the number of full-time service surface weather observing locations in the NWS Raleigh Country Warning Area.

## Research Questions

- How does the use of Vega improve our ability to make early and accurate weather predictions?
  - How significant is the improvement?
  - What improvements are required in the system to continue improving?
- How do the complex aspects of data affect the processing of PDEs with HPC?
- How does Vega perform where compared to other supercomputers such as NURION?
  - How does the architecture (number and type of nodes and cores) of a supercomputer affect its performance?
  - How do they handle the complications of larger datasets such as I/O bottleneck?
- How many days in advance can we make an accurate weather forecast prediction with our system?
  - How accurate are the predictions?
  - How does the accuracy relate to the time difference?

## Methodology

- We began with data selection, followed by the ingestion of that data into Vega's scratch directory, with C scripts used in parallel for data cleaning
- After data cleaning, we processed the data with the following steps [2]:
  - Step 1: Declaration and initialization of necessary variables/arrays
  - Step 2: Solution initialization to zero time step
  - Step 3: The main loop of the program runs the following substeps:
    - Substep 3.1: Solving the recursive equation for each grid point
    - Substep 3.2: Apply boundary conditions
    - Substep 3.3: Apply necessary filtering
    - Substep 3.4: Swap variables and/or arrays
    - Substep 3.5: Display or write output to a file
- This research is ongoing, and after we finish processing our data, we plan to visualize it and display the forecast time and quality difference

## Contact


Gurvir K. Bawa
PhD. Computer Engineering
College of Engineering
Email: bawag@my.erau.edu
Phone: (386) 290-5332

Jaclyn Welch
Senior, B.S. Computer Science
College of Engineering
Email: welchj12@my.erau.edu
Phone: (609) 686-3157

Embry Riddle Aeronautical University
1 Aerospace Boulevard
Daytona Beach, Florida, 32114-3900
Website: www.erau.edu
Phone: (800) 862-2416


## References


[1] Alex J. DeCaria, and Glenn E. Van Knowe, "Governing Equations and Assumptions," in A First Course in Atmospheric Numerical Modeling. Madison, Wisconsin: Sundog Publishing, LLC, 2014, p. 5-11.

[2] Alex J. DeCaria, and Glenn E. Van Knowe, "Programming Numerical Models," in A First Course in Atmospheric Numerical Modeling. Madison, Wisconsin: Sundog Publishing, LLC, 2014, p. 65-72.

[3] K. Ji-Sun, H. Myung and Y. Jin-Hee, "Examination of Computational Performance and Potential Applications of a Global Numerical Weather Prediction Model MPAS Using KISTI Supercomputer NURION," Journal of Marine Science and Engineering, vol. 9, (10), p. 1147, 2021.

[4] Cabral, F.L., Oliveira, S.L., Ostho, C., Costa, G.P., Brand~ao, D.N., & Kischinhevsky, M. (2020). An evaluation of MPI and OpenMP paradigms in finite-difference explicit methods for PDEs on shared-memory multi- and manycore systems. Concurrency and Computation: Practice and Experience, p. 32.