

The Utilization of Polar Grid Resources to Analyze, Process, and Display Data from the National Data Buoy Center

Bryce Carmichael, Lee Godley III, Diaminatou Goudiaby, Unquiea Wade
Mentor: Dr. Eric Akers
1704 Weeksville Road, Box 672
Elizabeth City, North Carolina 27909

Abstract— This project presents an example of a real world problem with data collection and processing. In order to solve this dilemma, cluster research was conducted to construct a model to serve as a base for a super computing pool, allocating data to a central location, and examining the data from the National Data Buoy Center. The model serves as a standard for solving real world problems using resources such as Condor, which utilizes two computer laboratories on Elizabeth City State University's campus. In order to standardize the model, major components must be accessed using Condor. These components include the allocation of data to a central location, the access of data from that location, the distribution of various jobs to both the laboratory and cluster computers, and the automatic installation of GNUPlot onto a computer.

Condor was developed by the Condor Research Project at the University of Wisconsin-Madison. Through collaborative efforts of incorporating the Condor Cluster Systems with the NBDC meteorological/ environmental observations, we were able to perform data collection and processing .

GNUPlot is the most widely used program to plot and visualize data for UNIX systems [4]. It is an open source, maintained, stable, and mature program that runs through the command-line used to generate two and three-dimensional plots of data. Executing GNUPlot is required to show the statistical data of buoys from the NBDC. The program is capable of producing high-quality, publication graphics and deals with large data sets.

I. INTRODUCTION

The National Data Buoy Center (NBDC) is an agency within the National Weather Service (NWS) of the National Oceanic and Atmospheric Administration (NOAA). On the coast of North Carolina, NBDC continues to deploy buoys in strategic locations for meteorological and environmental observations. These observations then provide such data in real time from automated observing systems. The meteorological data categories are wind speed, wind direction, wind gust, atmospheric pressure, and air temperature. All buoy stations measure sea surface temperature, average wave period, dominant wave period and wave height; however, certain stations also measure the conductivity of water. These measurements are then submitted through Condor for processing and graphically visualized with GNUPlot for analyzing purposes [2].

Condor is a free-ware, software system that creates a High-Throughput Computing environment by implementing the power of a cluster on a network [1].

II. METHODOLOGY

The research conducted by the 2008-2009 Polar Grid Condor Team focused on the procedural endeavors used to process the data received from the National Data Buoy Center. To accomplish this operation, specific tasks were assigned to the members: a script file consisting of the required variables used for graphing, a submission file to render the data needed to transfer files and fulfill the job, and the creation of a program used to submit the jobs to Condor for processing.

The data received from the National Data Buoy Center was a mass spreadsheet with information recorded every half of an hour. When plotted, data compiled of this magnitude over the years is very difficult to differentiate. Through the combination of the tasks previously stated and GNUPlot, various sectors were able to be broken down and plotted of time and type based upon the center's request. To initiate this process, a script file was written containing three sections [3].

The first section held variables that were initialized according to the region of information requested to be plotted. In addition to making the script more robust, it

increased its overall versatility. Any desired deviation from the previous request could be granted with the re-initialization of associated variable instead of the creation of a new script.

Next, a command was used to create a “png file” (a file containing the actual graph of the data). Finally, a series of commands set the correct formats for the plot of each category of data.

```
set terminal png
# GNUPlot recommends setting
terminal before output
set output "output.png"
# The output filename; to be
set after setting terminal
```

Fig. 1. Script file consisting of a series of commands and formats used to formulate the justifications of the designed graph.

Once the script file was completed, the submission file was created. This portion of the project acted as the “vehicle” used to transport the information into the Condor universe.

The submission file consisted of the commands that would enable the transfer of data files for processing, execution, and output [5]. It acknowledged the universe for which the program is operating upon, which portion of the program should be used to process the job, and where the results should be stored.

```
should_transfer_files = YES
when_to_transfer_output = ON_EXIT

transfer_input_files = plot.plot, Station_41035

Transfer-Output-Files = WindSpeed41035.png
universe = vanilla
executable = plot_results.exe
arguments = "WindSpeed41035.png 'Wind
Speed - Station 41035' Station_41035 'Wind
Speed' 6"
log = 41035.log
output = 41035.output
error = 41035.error

queue
```

Fig. 2. Submission file granting the request of critical files to be transferred for data processing and execution.

To complete the procedure, a program was created using the java programming language containing the lines of code for which the script file accessed and the submit file passed [7]. Within the program, the first line of action was outputting the file which contained the

data that was received for processing. Next, a condition was set to restrict the file from submitting less than three arguments, or sections of information. This was critical in the submission process because the three arguments formulated the structure of the rendered visual. Finally, the building of this program was completed by a command that provided access to the data, script, and submit files.

```
FileOutputStream fout =
new FileOutputStream
( "init.plot" );
PrintWriter pout =
new PrintWriter( fout );

ProcessBuilder proc =
new ProcessBuilder(
"wGNUPlot.exe", "init.plot",
"gnuscript.txt" );
proc.start();
```

Fig. 3. Screen shot of compiled program used to transfer data and submit jobs to Condor

III. RESULTS

These results were obtained by graphing the evolution of wind speed from the Southeast USA Historical Marine Data observatory. The station 41037 is exactly located 27 miles SE of Wrightsville Beach, NC. Fig.4 shows a very unstable evolution of wind speed in this area. Wind speed goes from 350 to 0 km/h, and back to 350 from January 3rd of 2008 to January 5th. That change indicates very unstable meteorological trends in the area: the wind speed values, which are higher average, indicate that the area is prone to tornadoes.

Fig. 5 is a graph of data collected from a different station in the East Coast: Station 41035 located in Onslow Bay, NC. The result is similar to the ones found in station 41037. Wind speed is unsteady and tends to reach high maximums of 350 km/h, indicating an area of high disturbance.

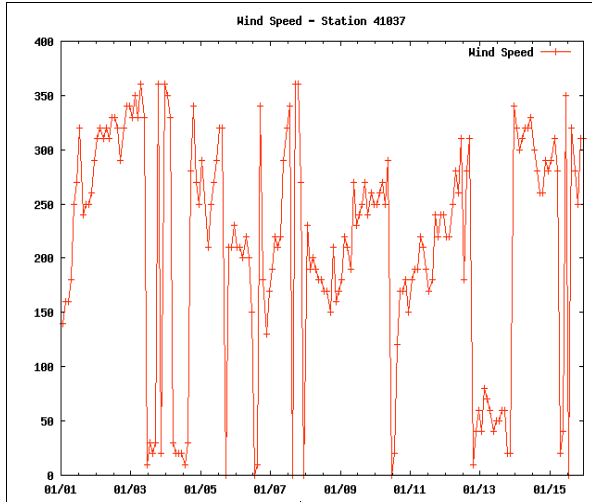


Fig. 4. Results of primary data submission to Condor cluster. Plotted wind speed of station 41037.

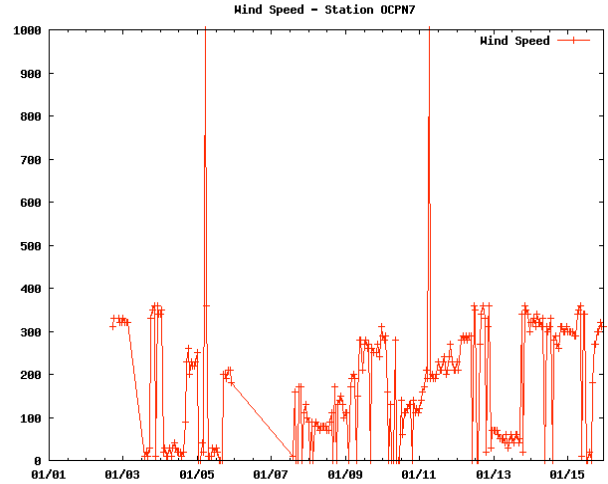


Fig. 6. Results of primary data submission to Condor cluster. Plotted wind speed of station OCPN7.

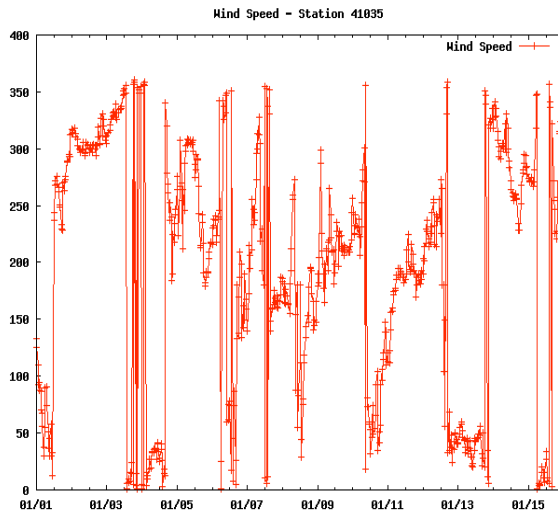


Fig. 5. Results of primary data submission to Condor cluster. Plotted wind speed of station 41035.

Figure 6 shows high overall trends in wind speed from January 1st to January 15th and maximal peaks of 1000 km/h on January 5th and January 11th. Compared to Figure 4, the trends of wind speed show a drastic decrease during the same time frame. This could have occurred as a result of the station's location in respect to satellite observation. Also, the tide and current had a significant impact of the buoy's readings.

IV. CONCLUSION

The results from the research project yield evidence of successful completion of solving our real world problem. Once the program was written efficiently, the jobs were rightfully submitted to Condor, processed, and plotted through GNUPlot. This endeavor opens the door to greater data submission and more efficient plotting.

V. FUTURE WORKS

For future goals, plans have been recognized to find and solve the incompatibility issues between the java universe and Condor on the Mac OSX computers. No perfectly written java program would ever yield successful completion; however, every C++ program submitted would run successfully within seconds. Also, actions are being taken to discover innovative ways to make Condor more efficient within our workstations. Intention is to find a method or an accompanying application will cause Condor to choose machines of which to submit jobs that meet the minimum criteria to complete a job instead of its idle status. This will increase the time frame needed to complete jobs on the network, thus enabling more jobs to be submitted simultaneously. In addition, other software packages will be researched to see if there are more applicable packages other than GNUPlot that can be used to graph data [6]. GNUPlot will suffice when plotting data of acute time ranges, but it lacks proficiency in clearly displaying plotted data over extended periods.

VI. REFERENCES

- [1] Z. Juhasz, *Distributed and Parallel Systems: Cluster and Grid Computing* (The Springer International Series in Engineering and Computer Science). New York, NY: Springer, Inc., 2005, pp. 34–57. W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
- [2] “Condor Project Homepage”, Retrieved from the World Wide Web <http://www.cs.wisc.edu/condor/>
- [3] H. Bauke, *High Performance Cluster Computing: Architectures and Systems*, Vol. 1. Upper Saddle River, NJ: Prentice Hall PTR, 1999, pp. 320–339. E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.
- [4] “Building a Linux Cluster on a Budget”, Retrieved from the World Wide Web <http://www.linux.com/articles/49654/>
- [5] R. Buyya, *Cluster Computing*. Hauppauge, NY: Nova Science Publishers, 2001, pp. 23–45.
- [6] “GNU Plot Homepage”, Retrieved from the World Wide Web <http://www.GNUPlot.info/>
- [7] G. Pfister, *In Search of Clusters* (2nd Edition). Upper Saddle River, NJ: Prentice Hall PTR, 1997, pp. 67–89, 234–277.
[1]. Z. Juhasz, *Distributed and Parallel Systems: Cluster and Grid* G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.