University of Washington Climate Analysis: Utilizing computing clusters to analyze large climate data sets.

Jason Woodring
STEM
University of Washington Bothell
Bothell Washington
jman5000@uw.edu

***Abstract*—this document contains information on a web application which utilizes a parallelization library to analyze scientific climate data across a computing cluster in an effort to improve performance and provide more accessible Time of Emergence calculations for climate research stakeholders.**

Introduction

The earth has been undergoing rapid climate change in modern times. The most direct reason for this is how much Co2 is getting trapped in the atmosphere. When sunlight travels through the atmosphere towards earth, the amount of Co2 in the atmosphere determines how much heat gets reflected back. If there were no Co2 in the atmosphere, then the earth would be an ice planet. Too much Co2 would cause the earth to become too hot. The planet Venus is an example of a runaway greenhouse effect; although not much heat penetrates the atmosphere, even less escapes it. Temperatures soar on Venus preventing any sort of life from developing.

Climate scientists believe that there is a >95% chance that human activities, specifically greenhouse gas emissions, aerosols, and land surface changes have exerted a substantial net warming influence on climate since 1750. They also believe that that there is a >90% chance that human greenhouse gas increases caused most of the observed increase in global temperature since the mid-20th century [9].

Skeptics claim that there has been severe climate change in the earth's history so what is happening now is natural. However what is truly alarming is the rapid rate of change, and the correlation to human industrialization / greenhouse emissions.

The climate change that the earth is experiencing could have drastic impacts. As temperature rises, many different phenomenon's may be experienced. Precipitation in the mountains in the wet season which would normally freeze will not, and that will result in several problems. There will be increased flooding because the moisture is not being caught in the mountains [7]. This will also generate extra power in the winter because of increased stream flows, but droughts and less power in the hot season due to less snowpack melt. Agriculture will be affected also; the Columbia River basin relies on the snowpack melt in the warmer months to irrigate crops [8].

In an effort to understand the coming climate changes and warn humanity, climate scientists have taken measures to predict the future state of the earth's climate. Several different climate models are produced by climate science facilities in the form of common data sets such as NetCDF files which contain grid like data suitable for analysis. NetCDF has several different software packages available for working with NetCDF data from different software environments such as C++, Java, etc [10].

A calculation of interest for climate scientists is Time of Emergence (ToE) . Time of Emergence is the time at which certain climate properties become apparent. In other words, when a certain climate attribute such as temperature starts consistently rising above a certain threshold, then that could be considered the Time of Emergence for that property [12]. ToE is important to predict, because that is the perceived indicator that could be a warning that extreme weather events could be increasing in frequency, such as storms or floods.

There are challenges with analyzing this climate model data however. The amount of data is large and special computing hardware and advanced computer science concepts may be necessary to process and

analyze the data. While large climate science facilities may have special computing setups to perform these analyses, many climate researchers are without these resources [4]. For the climate scientists without the computing resources, they tend to rely on custom software to do their analysis. This presents a problem for climate scientists, they are usually not versed in advanced computer science concepts, so they may create scripts / programs which are unmaintainable, have poor performance, not distributable, or simply not working.

Time of Emergence Calculation (ToE)

Input Climate model sets

Several different climate model sets can be used for different types of ToE calculations. The climate models typically come in the form of 3 dimensional arrays stored in NetCDF files. The x, y, and z dimensions may represent the longitude, latitude, and time for a given geographical region. Each element of the data grid holds a value, for example, a temperature based climate model would have temperature values for each element. An example of the structure of the data can be viewed below. This particular data set covers the latitude and longitude coordinates for the United States and areas of British Columbia.

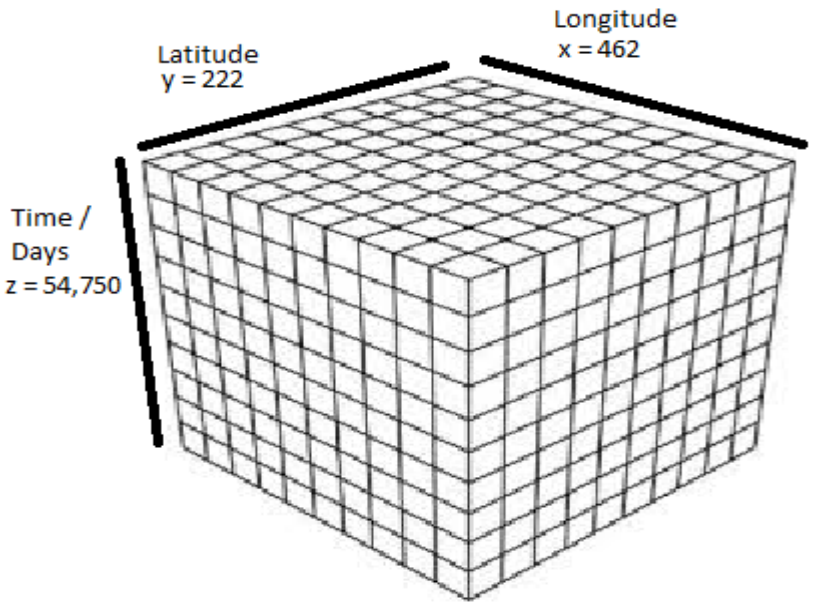

Figure x: Input Climate Model data format.

What the ToE calculation represents

The ToE calculation represents the year in which the average variable such as temperature exceeds the historical values to such an extent that it cannot be attributed to noise. This is done by analyzing historical climate data, finding averages and trends based on that and projecting that into the future.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

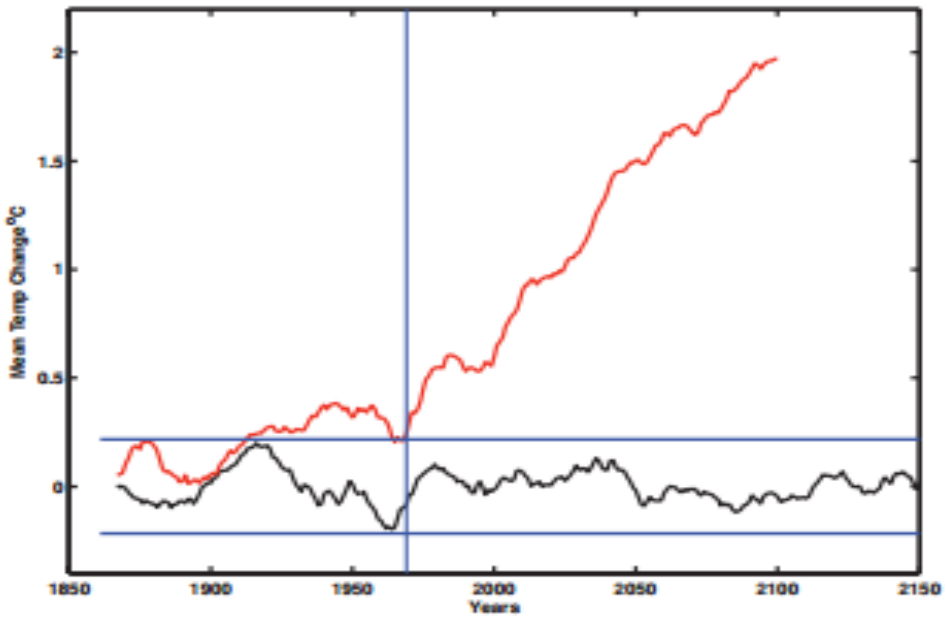

Figure 1: ToE trend representation

University of Washington Climate Analysis (UWCA) Software

UWCA was created specifically to perform ToE calculations for climate scientists and other interested parties such as funding agencies. It provides a simple and high performance way to perform these calculations so that people who would not normally be able to install and write the climate data specific languages would be able to still execute the ToE calculations and see results in a quick easy manner.

MASS: Parallel-Computing Library for Multi-Agent Spatial Simulation

The MASS library developed by Professor Munehiro Fukuda was used for this project because it was specifically designed for these types of simulations.

MASS Programmability

MASS abstracts out many more difficult computer science concepts such as parallelization and distributed computing. With MASS you can easily declare large size grid data structure which will be distributed out among many computing nodes with little effort by the programmer. It then only takes one line of code to specify how many threads should operate on the data on each computing node. Normally this kind of functionality would only be achievable with great programming effort, and would require many lines of hard to maintain code.

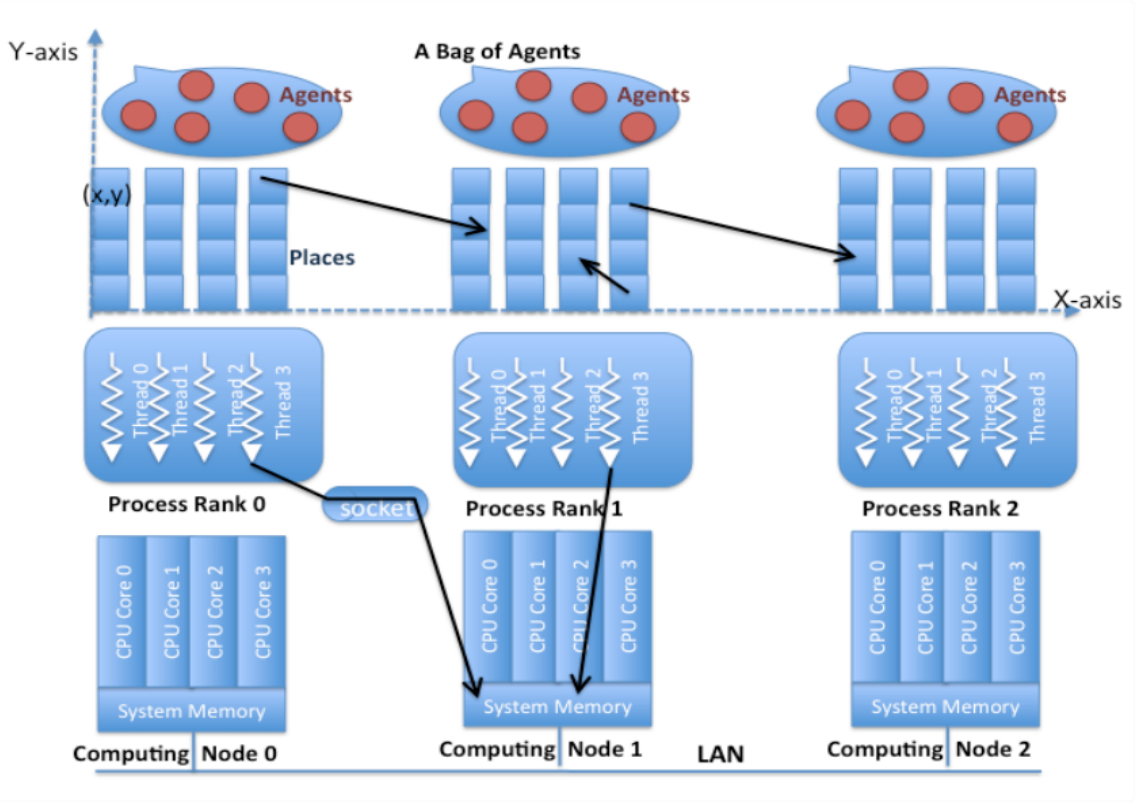

Figure x: MASS Architecture

Software Libraries and Tools used

- Java: The core language used for the server side application code. It is quite useful for the purpose of deploying the application to heterogeneous platforms.
- MASS (Multi-Agent Spatial Simulation) Library: A Java based library for handling distributed and parallelized processing across computing clusters.
- Glassfish: An open source Java Application server which hosts the web application.
- NetBeans: The primary IDE used for development which is well suited for Java development and has excellent integration with Glassfish server.
- Maven: A project management build tool which has integration with NetBeans.
- HTML: Used for the components in the web browser client.
- JavaScript: Used for the functionality in the web browser client.
- CSS: Used for the styling in the web browser client.
- GIT: Used for the source control of the project.

Graphical User Interface

The GUI for UWCA was designed to be simple, while still providing all the features for submitting ToE jobs, viewing the status of those jobs, and being able to download the files produced at any time.

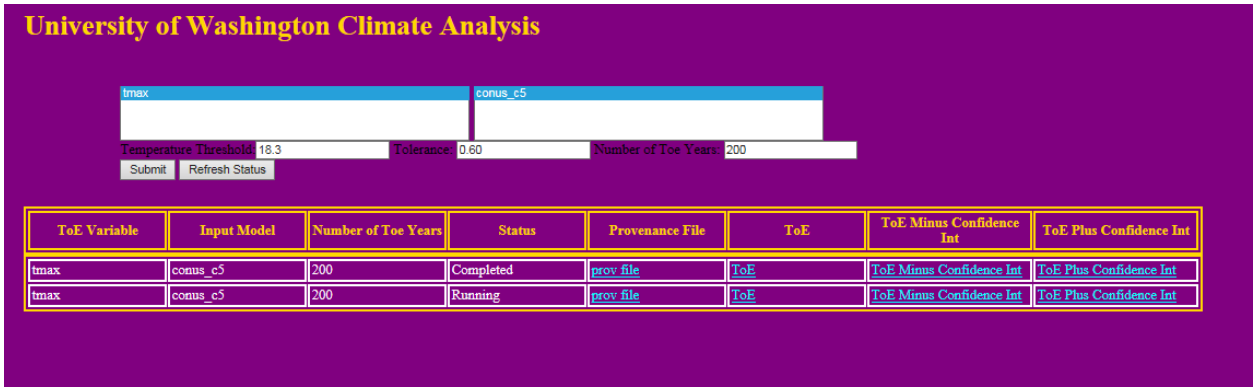

Figure x: UWCA GUI

The top left section of the GUI is for selecting the ToE variable to be calculated, and the top right section is for selecting the input climate model to be used. Directly below this area is where the user parameters are selected. The 3 parameters available for the tasmax (temperature based) calculation are:

- **Temperature Threshold**: the value to compare the day temperature value to, in order to discover if that day was over the threshold.
- **Tolerance**: The range used to select minimum and maximum days over temperature values based off historical periods.
- **Number of ToE years**: The amount of years to project into the future when performing the final ToE calculation.

All values will be defaulted to acceptable ranges and recorded in the provenance log if the user fails to enter valid values.

The bottom section is the Job Status Viewer. It allows a user to view which jobs have been submitted, and what the statuses of them are. The three possible values for status are Queued, Running, and Completed. The Job Status Viewer also allows for downloading of the provenance and output files for viewing.

ToE Algorithms Steps

Although many different climate properties can be analyzed using the ToE calculation such as precipitation and humidity, for this example we will analyze the ToE calculation to analyze future temperatures.

Step 1: Find Days over Threshold

In this step the input climate model data is transformed from the day based temperature values, into z dimension grid cells which represent a particular year (from 1950 – 2099) and the values in those cells amount of days over a specified temperature threshold (user defined). Approximately 365-6 time dimension elements get transformed into one element representing the amount of days over threshold.

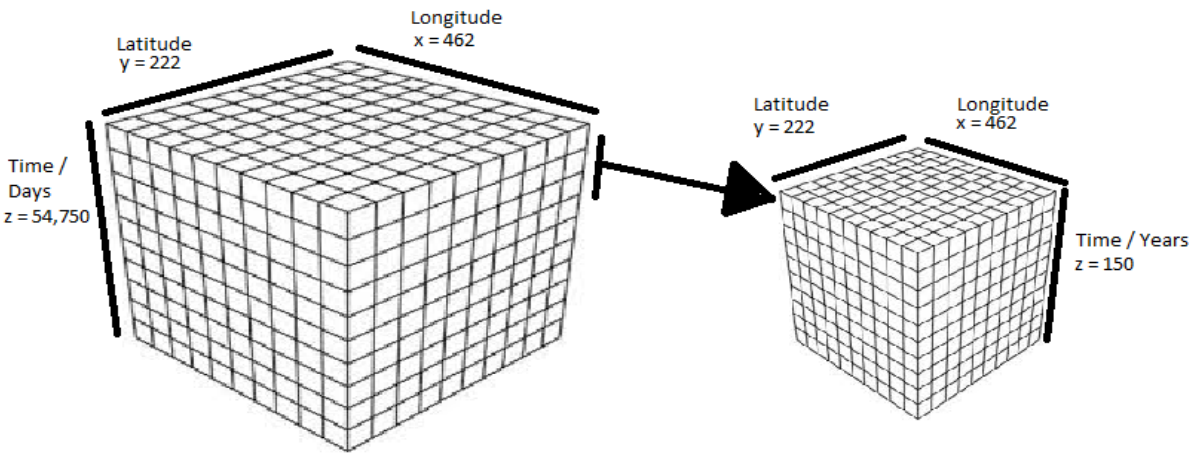

Figure x: Find Days over Threshold

Step 2: Find Historical Tolerances

Step 2 analyzes a 50 year historical period to analyze minimum and maximum values. For this calculation, x * y MASS Agents are spawned at the z[0] dimension which represents the year 1950, and travel down to the z[49] dimension representing the year 1999 collecting days over threshold values. The collected values are analyzed and the maximum value is multiplied by a user defined percentage such as 90% to find the maximum value. The minimum value is found in a similar manner. For example, if the calculation was decided to be done with an 80% min / max range, the minimum value was 0, and the maximum was 100, then the calculated minimum value would be 10%, and the maximum value would be 90%.

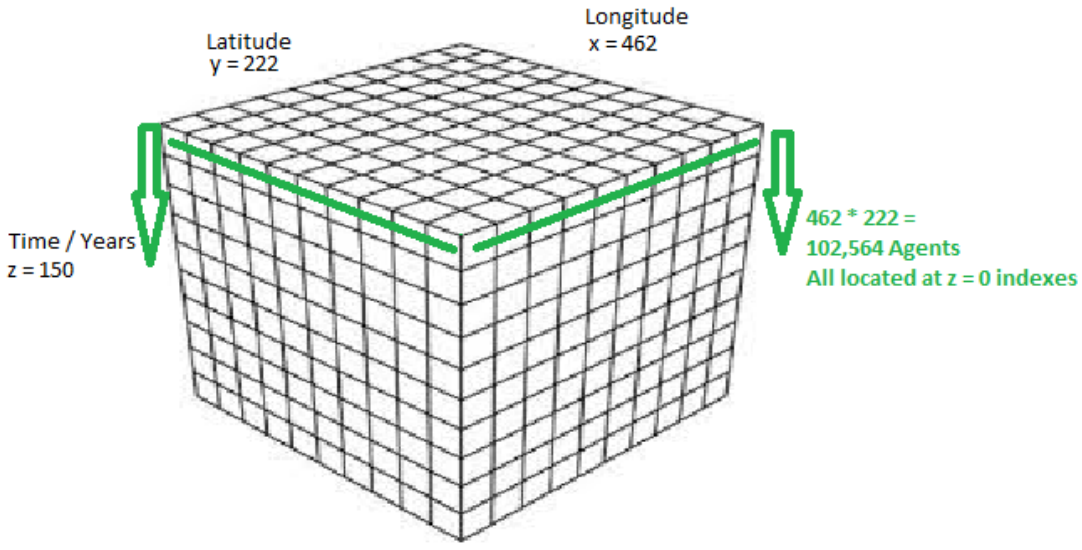

Figure x: Find Historical Tolerances

Step 3: Find Climatology

Step 3 moves the same agents from the old position, to a new z-index position representing year 1980. The agents then travel along the z dimension, collecting 30 years of values, adding them together. The total is then divided by 30 to get the average. This average represents the climatology.

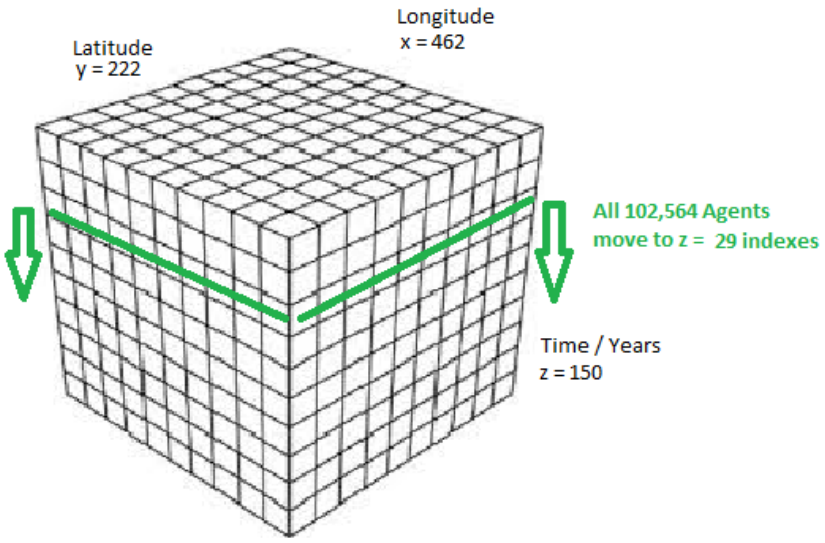

Figure x: Find Climatology

Step 4: Least Squared Regression

Step 4 moves the same agents from the ending step 3 positions to new z-index positions representing year 2006. The agents travel all the way down to 2099 gathering days over threshold values. For each latitude and longitude coordinate, the slope and confidence intervals are calculated for the set of values collected by the agents.

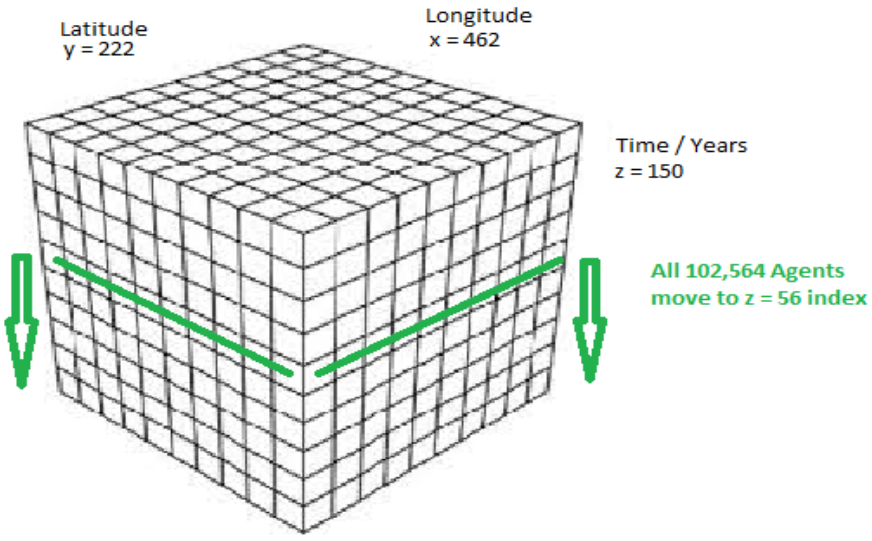

Figure x: Least Squared Regression

Step 4 results in 3 2 dimensional arrays

- Slopes for each latitude and longitude x and y coordinates
- Slopes + Confidence Interval for each latitude and longitude x and y coordinates
- Slopes – Confidence Interval for each latitude and longitude x and y coordinates

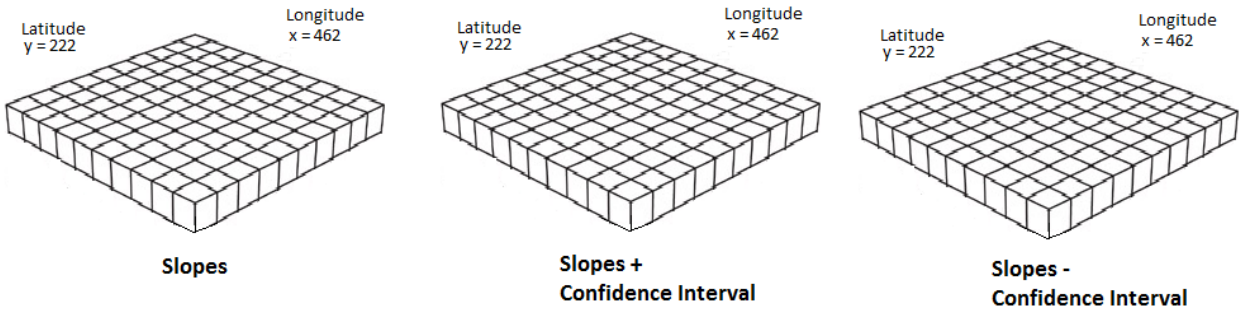

Figure x: Step 4 output artifacts

Step 5: Find ToE

For step 5, 3 3-dimensional arrays are created using the values derived from previous steps. For each of the arrays, the z[0] index becomes the climatology value derived for that latitude and longitude coordinate. Beyond the z[0] element the following pattern is used:

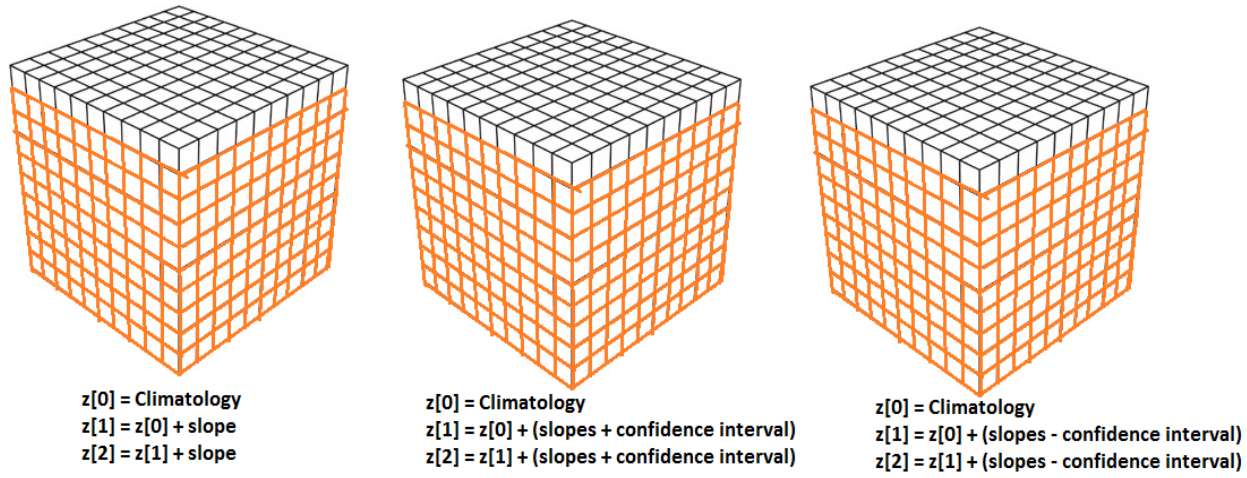

Figure x: Creation of 3d ToE arrays

The amount of z-dimension elements created is determined by user parameter, but is usually 200. The continual adding of the slopes and confidence intervals results in a positive or negative trend which when projected out far enough into the future, will cross the minimum or maximum values determined in step 2. The element (which represents a year in the future) at which the value exceeds the minimum or maximum values represent the ToE year. The final output of the ToE calculation will be 3 2-dimensional arrays which will be the same x * y dimensions as above, but will contain the year at which that grid cell exceeded the minimum or maximum values.

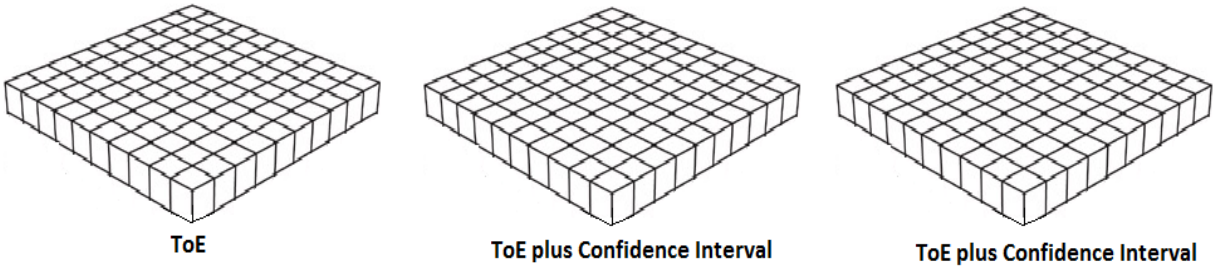

Figure x: Final ToE output

Software Architecture

The project consists of two main modules, the Servlet and EJB module which get wrapped up into an EAR which gets deployed to a Java application server such as Glassfish. The main application will initiate connections with the other machines in the computing cluster which should be able to all access the network file system to access climate model data sets.

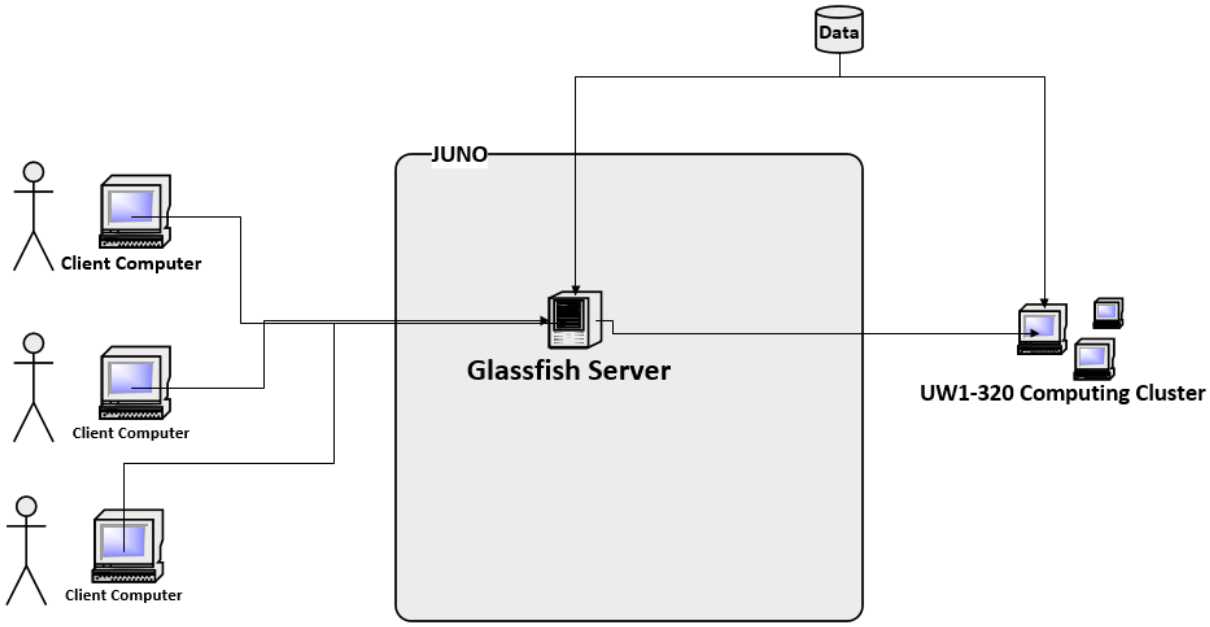

Figure 1: Hardware Architecture View

The servlets simply handle GUI operations and pass requests off the EJB module to be processed by the Job Runner class. The Job Runner class handles the dispatching of the user requested job to the MASS library and waits until the MASS library has completed the requested operations, then updates the Job Manager with the output of the calculation. The Job Runner then requests a new job to be processed by the MASS library from the Job Manager, if one is available.

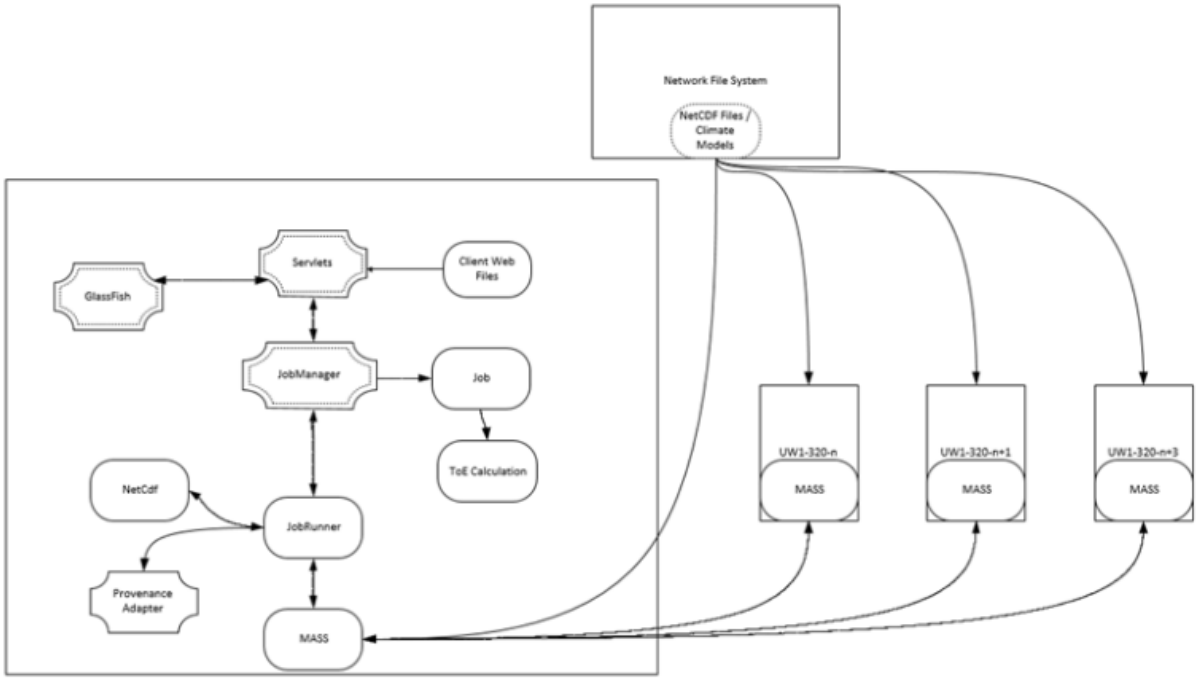

Figure 2: Process Architecture View

Provenance Features

UWCA features a simple provenance collection mechanism which logs many of the events which happen within the application. The following information is collected from the application and logged to a special provenance file:

- All times for events
- Which climate model input files were used for the ToE calculation
- The parameters for the calculation
- What steps were executed for the job
- What output files were produced

The file serves the purpose of knowing what happened when. As long as a user is familiar with the ToE calculation steps the provenance log file is understandable.

Data Visualization

The downloaded ToE NetCDF files produced can be viewed using several different free pieces of software such as Panopoly Viewer or ncBrowse. They each have their own way of visualizing the data which is meaningful.

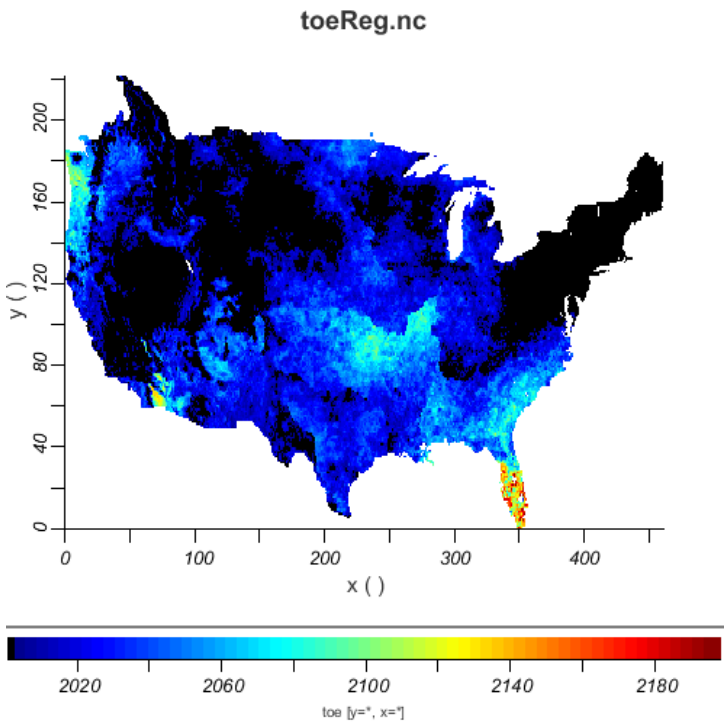

Figure x: Visual view of ToE output file using ncBrowse software

Analysis

Usability of UWCA

In an effort to determine the usability of the application, a senior climate science researcher was chosen to use the application and fill out a survey form which gathered information on the different areas of the application she found useful. The researcher claimed to have 10+ years in the role, and has used many different languages and tools for performing climate analysis. The following information was gathered:

	1 - Most	2	3	4	5 - Least
Usefulness of captured Provenance		x			
Adaptability of the tool with respect to your analysis task	x				
Acceptability of time spent on recording / capturing provenance	x				
Acceptability of time spent on analyzing recorded / captured	x				
Time spent Learning the software compared to other tools	x				
Ease of Using the Software	x				

The researcher went on to explain that the tool in its current form is also useful to climate science stakeholders who may not know how to create the CDO / NCL scripts which climate researchers traditionally use. This is of significance because funding agencies which are interested in executing climate science calculations will be able to with ease.

The researcher did mention one drawback of the application; it requires a software developer skillset to make modifications to it.

UWCA Performance

Overall the performance of UWCA was far better than the comparison CDO/NCL scripts which produce the same calculations.

	1 node	8 node	17 node
1 Thread	6:01	18:26	10:47
4 Thread	5:22	11:33	11:04
8 Thread	5:00	12:14	11:39
16 Threa	5:04	11:27	11:45
Script:	21:00		

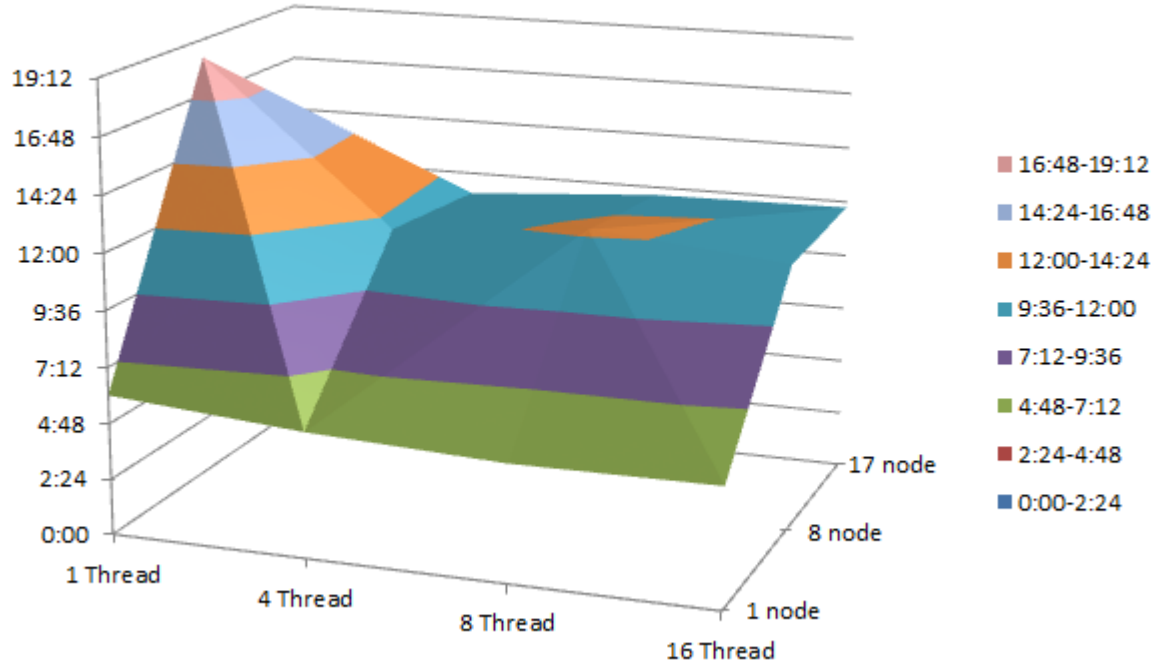

Figure x: Performance Measurements

The performance for the script execution was 21 minutes, whereas the fastest timing for all the thread and cluster configurations was 5 minutes for UWCA. Overall single node execution was much faster than all clustered configurations. This is because the master node is responsible for reading all the data and it sends it out to all slave nodes over a network connection. However, even our worst clustering timing still performs better than the script timing.

Conclusion

UWCA has many strong points such as being structured well architecturally, performing well, and being highly useable. The MASS library has proven to be perfect for analyzing the grid-like climate model data used for the ToE calculations with much less effort than without the library. However there is much ongoing work that can be done on the application to enhance different parts of it. The input model data reading algorithm which is Master node based currently can be modified to do a distributed node reading algorithm to further improve performance. Also, the ToE algorithm can be adapted to do various types of ToE algorithms rather than just the one implemented currently. Because the architecture allows for extendibility, the application can also be extended to include calculations which are not ToE based at all, however then greater UI changes will be needed.

Acknowledgments

University of Washington Climate Science Department

References

[1] [1] Fukuda, M., Stiber, M., Salathe, E., & Kim, W. (2013). CDS&E: Small: Multi-Agent-Based Parallelization of Scientific Data Analysis and Simulation.

[2] [2] Michalakes, J., Dudhia, J., Henderson, T., Klemp, J., Skamarock, W., & Wang, W. (n.d.). The Weather Research and Forecast Model: Software Architecture and Performance.

[3] [3] Fukuda, M. (2010). MASS: Parallel-Computing Library for Multi-Agent Spatial Simulation.

[4] [4] Yasutake, B., Simonson, N., Asuncion, H., Fukuda, M., & Salathe, E. (2014). Supporting Provenance in Climate Research.

[5] [5] Zender, C., & Mangalam, H. (2007). Scaling Properties of Common Statistical Operators for Gridded Datasets. International Journal of High Performance Computing Applications,21.

[6] [6] Dalton, M., Mote, P., & Snover, A. (2013). Climate Change In The Northwest Implications for our Landscapes, Waters, and Communities. Island Press.

[7] [7] Salathe, E., Hamlet, A., & Mass, C. (2014). Estimates of 21st Century Flood Risks in the Pacific Northwest.

[8] [8] Climate Change Impacts and Adaptation in Washington State. (2013). Climate Impacts Group University of Washington.

[9] [9] Climate Change. (2014, January 1). Retrieved July 11, 2014, from http://cses.washington.edu/cig/pnwc/cc.shtml

[10] [10] NetCDF FAQ. (n.d.). Retrieved July 12, 2014, from www.unidata.ucar.edu/software/netcdf/docs/faq.html

[11] [11] Software for Manipulating or Displaying NetCDF Data. (n.d.). Retrieved July 13, 2014, from www.unidata.ucar.edu/software/netcdf/software.html

[12] [12] Muir, L., Brown, J., Risbey, J., & Wijffels, S. (n.d.). Determining the time of emergence of the climate change signal at regional scales. The Center for Australian Weather and Climate Research, Hobart, Australia.

[13] [13] Hawkins, E., & Sutton, R. (2012). Time of emergence of climate signals. American Geophysical Union.

[14] [14] Keller, K., Joos, F., & Raible, C. (2014). Time of emergence of trends in the ocean biogeochemistry.

[15] [15] Time of Emergence of Climate Change Signals in the Puget Sound Basin Quality Assurance Project Plan. (2013). Climate Impacts Group University of Washington.

[16] [16] Mahlstein, I., Knutti, R., Solomon, S., & Portmann, R. (2011). Early onset of significant local warming in low latitude countries.

[17] [17] Maraun, D. (2013). When will trends in European mean and heavy daily precipitation emerge?

[18] [18] Ho, C., Hawkins, E., & Shaffrey, L. (2012). Statistical decadal predictions for sea surface temperatures: A benchmark for dynamical GCM predictions.

[19] [19] Capalbo, S., Eigenbrode, S., Glick, P., Littell, J., Raymondi, R., & Reeder, S. (2014). NORTHWEST. In Climate Change Impacts in the United States.

[20] Science, 1989.