CS 4624 - Multimedia, Hypertext, and Information Access

Ocean DB

Team: Justin Nelson, Will Hillhouse, Matthew Eberhard, Derek Hackenbracht

Client: Luis E. Escobar

Instructor: Edward A. Fox

Date: 5/9/2023

Virginia Tech, Blacksburg, VA 24061
# Table of Contents

- Table of Figures ........................................................................................................... 2  
- Table of Tables .......................................................................................................... 3  
- 1.0 Executive Summary ............................................................................................ 4  
- 2.0 Introduction ......................................................................................................... 5  
  - 2.1 Problem ............................................................................................................... 5  
  - 2.2 Motivation ......................................................................................................... 5  
  - 2.3 General Approach ............................................................................................ 5  
- 3.0 Requirements ..................................................................................................... 7  
- 4.0 Design ................................................................................................................ 9  
- 5.0 Implementation ................................................................................................ 10  
- 6.0 Testing ................................................................................................................. 11  
- 7.0 User’s Manual ................................................................................................... 12  
- 8.0 Developer’s Manual .......................................................................................... 13  
- 9.0 Lessons Learned............................................................................................... 14  
  - 9.1 Timeline/Schedule .......................................................................................... 14  
  - 9.2 Problems ......................................................................................................... 14  
  - 9.3 Solutions .......................................................................................................... 15  
  - 9.4 Future Work ................................................................................................... 16  
- 10.0 Acknowledgements ......................................................................................... 17  
- 11.0 References ....................................................................................................... 18  
- 12.0 Appendix A - Methodology ......................................................................... 20  
  - Users ....................................................................................................................... 20  
  - Goals ....................................................................................................................... 20  
  - Implementation ..................................................................................................... 22  
  - Workflows ............................................................................................................ 23
Table of Figures

Figure 1: Original File Format ........................................................................................................ 7
Figure 2: Desired File Format .......................................................................................................... 8
Figure 3: Database Diagram ........................................................................................................... 21
Figure 4: Scripting Plan Diagram .................................................................................................. 22
Figure 5: Diagram of Workflows ................................................................................................... 23
Table of Tables

Table 1: Timeline and Milestones ........................................................................................................14
Table 2: List of Services ....................................................................................................................23
1.0 Executive Summary

For this project, our goal was to collect relevant ocean data as well as automate the collection of future ocean data in order to help our client with his research of studying the effects of climate change on the features of the ocean and the likely effects on human health and biodiversity. By looking at relevant ocean data such as surface temperature as well as surface salinity, our client can draw conclusions on the behavior of pathogens that could lead to disease. Our team collected the relevant data from the source that the client specified on sea surface temperature and salinity, and put it into a database for our client to access. We then created a script to automate the process of collecting and transforming data in order for our client to be able to easily collect any new data as it comes in.
2.0 Introduction

2.1 Problem

The problem this project wishes to address is the relationship between changes in ocean conditions and changes in the habits of certain pathogens, particularly in coastal areas. The client, Dr. Luis E. Escobar wants data on ocean conditions, involving variables such as salinity and temperature, in order to understand the relationship between climate change and the behavior of certain pathogens. This requires looking at both micro trends, changes in conditions in the recent past, as well as macro trends, those changes occurring over a longer period of time. Collection and analysis of these variables in ocean conditions over both the long term and the short term will lead to a better understanding of the effect climate change has on pathogen behavior.

2.2 Motivation

As the climate changes, certain pathogens are able to grow, produce, and infect much more effectively than before. One of the primary areas where the effects of climate change are observable is the ocean. The ocean is a vast, unknown, and unpredictable force, and one of the greatest sources of uncertainty in our world. As such, it should be a central focus of research, especially considering the prevalence of dangerous pathogens in it. Moreover, climate change in general has led to increases in the spread of diseases. “Milder winters, warmer summers, and fewer days of frost make it easier for these and other infectious diseases to expand into new geographic areas and infect more people” [1]. The warmer temperatures brought about by climate change have made it easier for pathogens to spread, which is true of ocean-based pathogens as well as land-based ones. “For example, we could see more frequent and more severe instances of harmful algal blooms, which are the rapid growth of algae or cyanobacteria in lakes, rivers, oceans, and bays” [1]. Outbreaks of dangerous diseases are becoming increasingly more dangerous, and it is becoming ever more crucial to be able to predict and prevent them.

2.3 General Approach

The general goal is to collect data on ocean temperatures and salinity from a few different sites that the client has specified. In order to solve this problem, there are 4 general tasks to complete. First, for each source specified, all available data must be downloaded, including from as far back as 1900, if possible. The second task requires transformation of the data from whatever format it comes in into a TIFF file format. For instance, given a netCDF file from a source, the specific parameter for analysis must be pulled out and then converted into a raster file format. A simple way to accomplish this example transformation is via a Python script. The third task is simply to upload the collected data into the client’s data repository and organize it by type. The final task, and the most involved, is to create Python scripts to pull data from each
API that we have collected from, in order to automate the process of data collection for the future.
3.0 Requirements

There are two key deliverables for this project. The first is ocean surface temperature and salinity data dating back to 1980. The client has provided our team with sources from which he would like us to collect this data. These sources return a .nc file format and contain multiple data points. Figure 1 shows an example of this format.

Our job will be to download these files and extract the specific data points required by the client (surface temp. and salinity). The final product will be in a TIFF file format, which the client has specified as the desired format for his research. Figure 2 shows an example of this desired format.
Our team will keep an organized file system sorted by month and year that is easily navigable, before we deliver it to the client. The second deliverable for this project will be a program that automates the collection of future surface temperature and salinity data. The sources that our client has provided us possess APIs for automation purposes. Our team will design and implement a script around these APIs that takes a range of dates as input, and returns the desired data available between the provided dates. The script will automatically format and organize the data, mimicking the final product from deliverable one.
4.0 Design

The design of our second deliverable will involve four key stages: the input, the API calls, the data extraction, and the output. Firstly, the input to our script will be in the form of a range of dates. The user of the script will provide two dates such that the first date comes before the second. These dates will go through a series of processing checks by the script to ensure that they are valid input. Some examples of processing checks include: checking that the input is a valid date, checking that the second date is greater than the first, checking that the input dates are before the present day, etc. After the input has been processed and validated, our script will move on to the next stage: the API calls. Our script will take the processed input from the user and build an API call to each of the databases for ocean surface temperature data and ocean salinity data between the provided dates. Our script will send the requests and save the data before moving forward to the next stage: data extraction. This stage is important because its goal is to extract the data from the API call into a TIFF file format, which is the client’s desired format for his research. For the sake of the continuity and the fourth stage, this stage will aim to closely mimic the historical data from the first deliverable of this project. Finally, stage four aims to organize the files from stage three into an easily navigable directory that can be added to the preexisting directory of historical data.
5.0 Implementation

The implementation of the system can be broken down into a few categories. The first was downloading the data. The implementation simply looks like following conventions to download data based on the source of choice. For instance, some of the sources allowed direct downloads, whereas another required an FTP client to download files. At first we were required to use several data sources to build a complete dataset. These sources were provided by the client and our team just had to extract the data from them. However, the data from these original data sources ended up not working for our client’s research, so a new data source was found later in our development process. The second category was transforming the data. The implementation of the data transforms comes in the form of a short Python script to convert a netCDF file to a raster file format. We ended up going through several implementation strategies for the conversion technique before we found one that worked. The third was storing the data. The implementation of this was simply a shared Google Drive with the client. The client only needed the data locally on their computer, and he asked that we transfer it to him through a designated folder on Google Drive. The fourth was automating the process. The automation of data collection involved modifying the existing code to take user input, and utilize the data source’s API to grab data automatically.
6.0 Testing

Alongside the development of the script, our team has also developed test cases for the various functions. This has allowed our team to track the progress of the development process and make sure that our final product meets the client’s expectations. Our team has developed test cases for each of the input processing checks to ensure that we are building a robust system, as well as test for edge cases including if the given date is included in our data as well as if we are given a valid date in the API Request. Additionally, our team has designed sample API calls as a test of what the expected return should be. Our team has also built test cases for extracting data from the API calls to ensure correctness of our implementation. Finally, we have observed the output of our script to check for organizational issues with the final product.
7.0 User’s Manual

As part of our second deliverable, we included a written documentation outlining how to use the script that our team developed. This document explains to the client how to run the script, what the input should look like, and what the output should look like. The goal of this document is for anyone who reads it to understand how to properly interact with our team’s product. If this goal is successful, then this document will act as an important resource to the client and any future users.
8.0 Developer’s Manual

Similarly to the User’s Manual, the Developer’s Manual is a written documentation for the team’s second deliverable. However, unlike the User’s Manual, this document outlines how the script works, rather than how to use it. This document explains to the reader our methodology behind specific design decisions, how the API works, how the new conversion method works, and other comments we feel are necessary for future developers to understand the code. The goal of this document is to act as a reference for future developers that wish to make changes to the script or utilize specific features from the code. Our script is broken down into functions so that specific features, such as API calls and conversion techniques, are separated and neatly organized.
9.0 Lessons Learned

9.1 Timeline/Schedule

Table 1 shows an outline of the project schedule, which our team was able to closely follow for the whole project.

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 14</td>
<td>Clarify project details with client</td>
</tr>
<tr>
<td>February 27</td>
<td>Local download of a portion of temperature and salinity data</td>
</tr>
<tr>
<td>March 1</td>
<td>Have temperature and salinity data downloaded locally</td>
</tr>
<tr>
<td>March 3</td>
<td>Combine, format, organize both datasets into one file system. File system broken down by year and month</td>
</tr>
<tr>
<td>March 4</td>
<td>Transfer locally downloaded data to client’s folder</td>
</tr>
<tr>
<td>March 17</td>
<td>Date input system implemented for script (The date input system tells the script a date range to collect data from with the API call, so the user can give a range of dates and receive the necessary files from that range of dates.)</td>
</tr>
<tr>
<td>March 24</td>
<td>Have formatting functionality implemented for data retrieved via API calls</td>
</tr>
<tr>
<td>March 31</td>
<td>Ease of use features and cleaning up source code</td>
</tr>
<tr>
<td>April 14</td>
<td>Testing the script and validating results</td>
</tr>
</tbody>
</table>

9.2 Problems

The first challenge presented by our OceanDB project is the initial data collection/downloading from the many available sources. Each source API possesses different interaction functionality which will be a challenge to automate in the future. Downloading the data is also a challenge
because of its size. Approximately 63 GB of raw data has been stored in our client’s repository. Data verification is also a challenge due to the data’s size. Before moving forward with parsing/sorting the data, we must first ensure that it is clean and available in a uniform format. The primary file type we are working with is GeoTIFF which presents itself as a grid of cells projected onto a global map. The problem with collecting GeoTIFF files from several different sources is consistency. The different sources are unlikely to have identical cells covering identical regions. Another problem our team is encountering involves the file transformation from .nc to .tif. A primary concern in this transformation is maintaining completeness of our data. It is possible that the data conversion is not performed robustly and will result in data loss.

Another issue we faced during the project was the incompleteness of the data from the source we were given. We were given several data sources to collect and convert data from. We wrote scripts to convert the files from those sources and turned the data over to our client. After a week of reviewing the data and testing the data for his research, our client returned to us and informed us that there were issues with the data from the original source that we were provided. Our client came to us with a new data source that was more complete and included all of the parameters he needed in one location. However, this source came with its own issues, and a new conversion method needed to be created.

9.3 Solutions

To overcome the challenge of collecting data from different sources with different APIs, we might have had to design an acquisition pipeline. This would include standardizing the data format, verifying the data quality, and storing the data in a uniform way. However, the client resolved this issue by clarifying that he only wanted the data from a single source. To ensure data consistency amongst files with different sources, we used 3rd party data cleaning tools to standardize the data. Alternatively, another group in the future could use data visualization methods to manually inspect the data and identify any quality issues or inconsistencies. In addressing the inconsistent cell/grid coverage from different GeoTIFF sources, they may be able to establish their own global grid system, match geospatial coordinates with their corresponding GeoTIFFs, and pair ocean data cells in this manner. Finally, we designed custom tests to verify file integrity before and after transformation from .nc to .tif.

To solve the issue with the new data source we had to use a roundabout method to get an accurate conversion. The .nc files from the new data source used a different coordinate system than the files from the original sources. When we used our original conversion method on the new data source, the code produced errors. Thus, we had to come up with something new. After a period of research and discussion with our client’s assistant, we came up with a new conversion method that involved converting the .nc files to CSVs, before converting to the final TIFF format. This method showed promise, and was ultimately the method we chose to use. However, the increased complexity of the conversion method required a lot of testing and debugging to produce the correct results.
9.4 Future Work

Future teams who work on this project can progress these efforts in several different ways. One direction would be to implement data visualization tools. This would make the data more accessible and useful to a wider audience. This could also include interactive, front-end development, allowing potential users to explore and interact with the ocean data. Another direction a future team may take this project is to focus on pattern identification and predictive models. This direction may involve a heavier focus on artificial intelligence and machine learning to draw conclusions about the future using data from the past and present.
10.0 Acknowledgements

Dr. Luis E. Escobar (escobar1@vt.edu) guided the data collection process and helped direct the group’s efforts.

Dr. Edward A. Fox (fox@vt.edu) has helped the group with project management, settings and upholding milestones, and working together as a team.

GTA Ryan Wood (ryanw23@vt.edu) has helped with answering any questions about the whole process and being a valuable resource in class.
11.0 References


12.0 Appendix A - Methodology

Users

**Data Analyst:** Have as much ocean salinity and temperature data as possible in a usable format. Preferable format is geoTiff, so that the user has a visual way to analyze the data. The data analyst wants data from as far back as possible in order to have the widest view possible of ocean trends.

**Data Collector:** Wants a script(s) to pull data as it becomes available from one of several sites. Whenever new data is collected or data is updated, the data collector wants an automated process to pull and format the data for use.

Goals

Figure 3 shows a diagram of the tasks necessary to complete our first goal, which is creating a database of ocean data. Figure 4 shows a diagram of the processes necessary to complete our second goal, which is automating the process of the data collection. Table 2 shows a list of the necessary services to complete these goals, and Figure 5 shows a diagram of how these goals work together.

**Goal 1: Database of ocean data**

**Tasks:**

1. Collect data  
   a. Find sites with ocean temperature and salinity data  
   b. Find out how far back data is available from each site  
   c. Organize sites and data by what years are available for download  
   d. Download corresponding dates from each site

2. Format data  
   a. Figure out what file formats came out of which sites  
   b. Organize files by formatting  
   c. Find a way to convert from each data source to geoTiff or some ASCII format  
   d. Perform conversion

3. Store data  
   a. Find place to store data (repository or client’s machine)  
   b. Organize all files by year so database is easy to navigate  
   c. Transfer formatted data
Goal 2: automate data collection

Task:
1. Create script to collect data
   a. Find a site with up-to-date data and an available API
   b. Learn how the API works by building small sample test program
   c. Scale up test program to full functionality
Goal 2: Automate data collection

Figure 4: Scripting Plan Diagram

Implementation

Table 2: List of Services

<table>
<thead>
<tr>
<th>Service ID</th>
<th>Service Name</th>
<th>Input file name(s)</th>
<th>Input file IDs (comma-sep)</th>
<th>Output file name</th>
<th>Output file ID</th>
<th>Libraries; Functions; Environments</th>
<th>API endpoint (if applicable)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Collect Data</td>
<td>All data given by Dr. Escobar</td>
<td>*.ncdat</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td>-------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-----</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1A</td>
<td>Format Data</td>
<td>Past data files (*.ncdat)</td>
<td>N/A</td>
<td>*.geoTif</td>
<td>N/A</td>
<td>Possibly ArcGIS</td>
<td></td>
</tr>
<tr>
<td>2A</td>
<td>Automated Process</td>
<td>New data files</td>
<td>N/A</td>
<td>N/A</td>
<td>Python Script</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Workflows**

Goal 1: Workflow 1 = Service1A + Service1B  
Goal 2: Workflow 2 = Service2A

*Figure 5: Diagram of Workflows*