Visualising and analysing climate change data via GIS, utilising a web-based microservice application

Updated problem description, solution approach and work plan

Aidan McKendry

Introduction

GIS (Geographic Information System) technology is a very powerful software tool and has a range of uses in both private and public sector organisations, such as development planning, resource management and is critical for a lot of geographic scientific research. "A geographic information system (GIS) is a computer system for capturing, storing, checking, and displaying data related to positions on Earth's surface." GIS systems also consist of additional functionality such as enabling users to manage, analyse and visualise data sets according to the users' needs. For example, managing and visualising the flow and location of all rivers and bodies of water in the UK.

These systems are commonly packaged as large monolithic style software products which require significant hardware, memory, and licensing requirements.

These GIS systems are becoming outdated from lack of accessibility and hardware requirements as historically it has required installation of a large software application and is very demanding on CPU & RAM. I will introduce microservice architecture concepts to mitigate these requirements by providing an available web-based application. This presents an opportunity to make GIS technology more accessible for entities that previously wouldn't have been able to access this technology due to cost constraints.

Problem description

As previously mentioned, one large drawback of desktop applications for GIS is the cost of hardware requirements to run. Any organisation that wants to use this technology need a significant investment in hardware and managing a large, centralised database. If insufficient investment is put into the hardware components, desktop GIS applications will have much longer response times and reduce its effectiveness in practice. If the centralised database is overloaded with data and not sufficiently maintained this will also result in a much slower application. Therefore, requiring training/ dedicated database maintenance teams in companies that are required to use these applications.

UDC Digital Utility Solutions which specialise in GIS-based software solutions have quoted start up fees for using GIS as: “The base install of GIS editing, and web browsing software, will be in the $500K to $800K range plus the GIS product vendor’s cost of their ELA being driven by number of users.”

By adopting a microservice application and utilising container technology, I aim to prove that GIS technology can be utilised without the need for this large capital expenditure.

Solution description

In this section I will illustrate what I aim to achieve in my solution, the architectural design of the system, the technology stack that will be utilised, the advantages that my solution will have over monolithic GIS solutions and what I will demonstrate in my initial proof of concept demo and in my final end-state final demo.

Goals and requirements

I have agreed with my supervisor to have an element of flexibility to change around the functional features, as of now the GIS system should be able to:

- Dynamically cluster points when a user zooms in/out.
- Implement a timescale/ play button which will progress the year.
- Have interactivity with databases, i.e., when a state is highlighted, the selected records can be displayed for that year.

I plan on using the popular MVC architecture for the application which separates key parts of the system into the distinct layers of: Model (database), Visualisation (front end display) and Controllers (handling the requested operations and interacting with the model). I plan on using MongoDB as the initial database for my project with the aim of adding at least one more for storing any secondary datasets that I will use during my demonstration. Below I have included a diagram showing at a high level how this architecture should look for this project. For the visualisation I plan on using Google maps APIs for the base map. I do not plan on deploying the application, using locally for the demonstration.

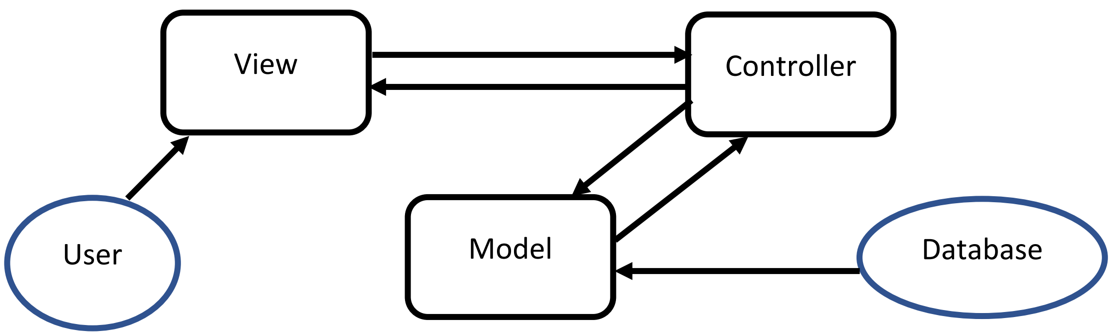

Figure 1 – MVC architecture

Figure 1 shows how the view is unaware of the model component, this is an example of the low degree of coupling but a high degree of cohesion which are cornerstone to a satisfactory microservice system [3].

Technology stack

I plan on using Oracle VirtualBox VM with a Linux environment for developing the application along with Docker for containerizing and orchestrating the services. Node JS will be used for the front-end map visualisation and the model connection to the database instance while Java Spring will be used for the building of the back-end microservices within the controller for performing the different features of the GIS system. I plan on working in an agile flow when developing my application and will plan the architecture around the 4+1 model which represents the architecture in multiple concurrent views giving a full idea of the workflow of the web application.

Advantages of solution

A microservice based web application eliminates the hardware requirements for users by hosting the database and application within a docker container which store the database and application remotely. Taking away the barrier of cost of hardware should make GIS systems a more efficient option to many organisations worldwide. The cost will drop significantly for end users with the benefit of providing this software, via any machine that has internet access, to anyone who wants to use it. The splitting up of one large, centralised database should also result in increased responsiveness. Increased responsiveness will also be achieved by containerising all functional requirements of the GIS system and having them accessible through endpoints. This will allow the ability to scale up areas of the system such as backend functions for visualising data during calls with large datasets.

Functional requirements
In this section I will list the functional requirements for the solution. I have used MoSCoW to prioritise these requirements [4].

Must have
- A base-map will be created using the Google Maps API.
- Data held will be used to populate to dynamically style individual states for the map
- Users will be able to interact with the map (zoom in, zoom out, drag cursor to move around the map).
- Front end should add selection on year of recorded data being displayed, either playing through a range of years or just setting the year they would like to visualise the data for.

Should have
- Multiple layers for viewing datasets using different GIS data visualisation techniques.
- Several GIS functions to aid with analysis, data management and distance.
- Users will be able to select which year of data they wish to have visualised on the map.

Could have
- Access for any device connected to internet, i.e., smartphone/ tablet access to application.
- Deployment via Kubernetes with auto scaling of services.

Won't have
- User can not add their own datasets from the app as this will need to be done during creating the database image.
- Database write access so datasets will not be editable by users.

Non-functional requirements
- Follow Microservice architecture concepts (single responsibility principle, high cohesion and low coupling).
- User will not have any hardware requirements.
- Fast response times for all GIS operations.

End-state Demo

I will finally demonstrate this application being used as a proof of concept against for visualising and analysing climate change data. My initial dataset will consist of yearly records of temperatures in the US for each state, in an aim to highlight how temperatures are increasing most rapidly in areas closer to the north and south poles of the planet. This has significant implications for the future of our planet since melting ice caps lead to rising sea levels worldwide, causing flooding in many low-lying areas as well as the production of methane gas which is roughly 4 times more harmful to the environment than carbon dioxide. I have chosen this topic because it is extremely relevant issue and arguably the biggest risk facing civilization today. I believe this should be discussed more and along with the accessibility benefits of a microservice based GIS system, hopefully this can help spread awareness of this major problem.

Success criteria

Should have a functioning microservice based system, satisfying the goals and requirements for the demo. This product should be efficient at reading data from the databases with quick response times. The demo should cover common features that users will want in a GIS system and keep costs to an absolute minimum.

Development plan

Start	End	Work
11/1/2022	12/1/2022	Design of workflows and architecture,
11/1/2022	11/20/2022	Requirements sign off
11/20/2022	3/1/2023	Development (with ongoing unit testing throughout)
1/1/2023	3/20/2023	Deployment
11/1/2022	1/13/2023	Initial demo & workplan
4/1/2022	4/17/2023	Demo recording
1/13/2023	4/17/2023	Final submission

Figure 2 – Gantt chart

Figure 2 outlines the planned start and end dates for key pieces of work during the development lifecycle of the application. I plan on using a GitLab repository for storing all work related to this project. For the testing of microservices I aim to use GitLab's CI pipelines to run unit tests that cover all areas of each microservices functionality. This will help ensure that regression testing is continually carried out through development.

References

[1] National Geographic, "Geographic Information System", n.d. [Online]
https://education.nationalgeographic.org/resource/geographic-information-system-gis

[2] UDC, "How much does a GIS system cost?", 30th April 2021, [Online]
https://www.udcus.com/blog/2021/04/30/how-much-does-gis-system-cost

[3] Fenton and Bieman, 2014, From Monolithic Systems to Microservices, Davide Taibi. [Online]
https://pdfs.semanticscholar.org/6a46/bb3280e6f4ad87c215a2d953d8c031f828a0.pdf

[4] Stanislava Simonova, "Requirements Gathering for Specialized Information Systems in Public Administration", 2021 International Conference on Information and Digital Technologies (IDT). [Online]
https://ieeexplore.ieee.org/abstract/document/9497618