Google Summer of Code 2018: Real time monitoring of space weather

author: Jitao Zhang

organization: Space@VT

mentor: keith kotyk

email: zhangjitao0405@gmail.com

1. Summary

Develop a web tool with some data visualization libraries to monitor weather data from superDARN radars in real time for further research.

2. Project description

Space weather directly influences several technologies we are dependent on, such as GPS, Satellite communications and aviation. It is therefore necessary to monitor and asses space weather in real time. Currently, there are several SuperDARN radars making continuous measurements of space weather. However, it is not straightforward to visualize the collective data from all these radars in real time. The goal of this project will be to develop a tool to collect, process and visualize data from several SuperDARN radars in real time. Such a tool would be immensely useful not only to the space science community but also to a wider group of researchers who are impacted by space weather.

3. Deliverables with a timeline

3.1 Beautify and customize the globe using echarts and its plugins

The current globe in the website[1] is too abstract and not vivid. One of the main functions for the globe is to show basic geo information easily for the users and the researchers to analyze and also helping additional data visualization in the project. So changing the globe appearance and show the basic terrain and landform are the first priority. There is a powerful, interactive and popular data visualization open source project -- echarts[2] in China. I found two globe demos in their official site [3] and [4]. I think the globe should like that which will better support the data visualization part. I will also add some click, hover or drag events on the globe for showing more detailed information for a certain radar and the geo position and finally add the data visualization as a layer on that.

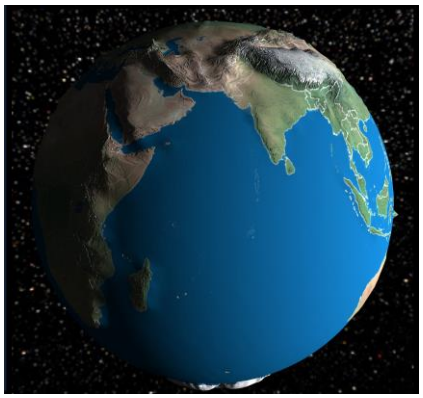


Figure1: Demo Globe 1

Figure2: Demo globe 2

3.2 Show weather data in better visualization as a layer on the new globe

Some data types such as plasma velocity, elevation angle, SNR, and spectral width should show the conditions in a certain area not as a point but in a more fancy way. Using color and movement both will show the data information more vivid and straightforward. I will use different animation for different types in weather data requested from backend. And different colors show the degree of a certain type data in its whole range. All the colors will still use the current colors in the project.

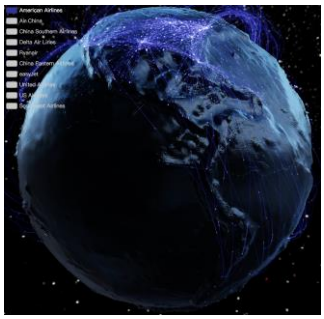


Figure3:
Flight movements[5]

Figure4:
Animating Contour[6]

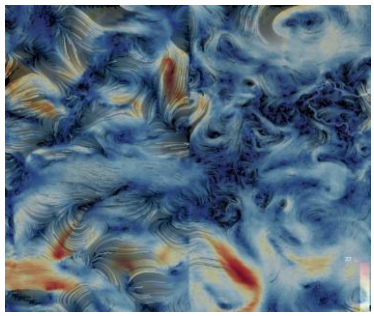


Figure5:
Wind Visualization[7]

With the combination of such animations and colors, it is definitely much cooler and straightforward than before.

3.3 Add a real time updated basic chart to show the trend in relative weather elements

Even though the globe can show the real condition in some certain areas, it is so difficult for the researchers to make analyze and conclusions based on the globe and the layer on that especially for the frequency, noise and average data. So a real time updated line or other basic charts are so important because it can help the researchers see the trend easily. The frequency, noise and averages data are requested from the backend in every 3 seconds. So the frontend should update the data in current chart in every 3 second or after an certain interval which is n * 3 seconds. Chart.js[8] is a simple yet flexible JavaScript chart library. I used it to finish my demo[9] which used the line chart to show some trends for IMF data in different time span such as a day, a month and even a year.

3.4 Compare the real time data with the data in a certain past interval

Researchers need to use the past data for a lot of comparison experiments. So getting the data in the past is also important. At first, my idea is that, in the backend, after the server received some data in an certain interval which might be 3 -5 mins (need to further consider) , the server can save the data to a cloud database asynchronous. In this way, the research in frontend can easily using ajax to get the past data to do some comparison. In the email with Keith, I know that the project had added this feature to the current project and the interval is much longer than I expected. So the compare time span is mostly dependent on the past data format in the database and the delay resulted from query in database.

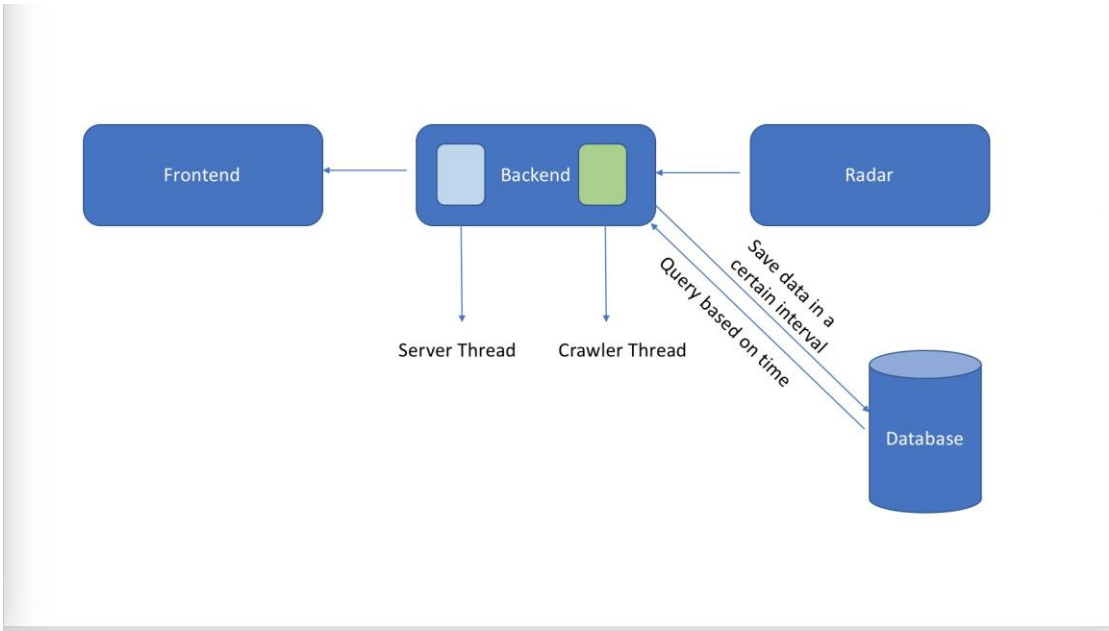


Figure6: Architecture

3.5 Maybe need some methods to control error both in the frontend and backend

Sometimes we have bad internet connections. In this project, we definitely need some error correction methods to control the server and prevent the error in data visualization especially in the updated chart. This part is not so clearly now. But I guarantee it will be a tough problem and need to have a deep understanding of the websocket. The methods both need to be added in frontend and backend. In the backend, the server needs to recognize the error or delay of the data and send the information to frontend. And the frontend needs to stop the chart and make correction in the next connection.

(I won't include this part in the timeline directly because it will be added during I develop other features)

According to the timeline of GSoC 2018 there are 12 weeks including 3 evaluations. I plan to work at least 40 hours per week from Monday to Friday and using the weekends as buffer time(in most time I will still work). Even though I have my university capstone evaluation in June, I guarantee that it will only spend no more than 2 – 3 days which will not delay the GSoC. And after that, I will spend all my time in GSoC and finish all the features I proposed. Here is the timeline.

Timeline Table

April 24 - May 14 ·····	Community Bonding – coordinate the project with mentor. Get familiar with the current version of realtimedisplay and echarts, and discuss with the mentor about possible problems in the following 12 weeks.
May 15 - May 25 ·····	1.5 weeks coding - beautify and customize the globe using echarts and its plugins
May 26 - June 11 ·····	2.5 weeks coding - show weather data in better visualization as a layer on the new globe (2 / 4)
June 12 – June 15 ·····	Phase 1 evaluation, 4 days – prepare results for evaluation
June 16 – June 26 ·····	1.5 weeks coding - show weather data in better visualization as a layer on the new globe (4 / 4)
June 27 – July 9 ·····	2 weeks coding – add a real time updated basic chart to show the trend in relative weather elements (3 / 3)
July 10 – July 13 ·····	Phase 2 evaluation, 4 days – prepare results for evaluation
July 14 – July 24 ·····	1.5 weeks coding - compare the real time data with the data in a certain past time or interval
July 25 – August 6 ·····	2 weeks coding – final fix bugs and optimization
August 7 – August 21 ···	Final evaluation

4. Benefits to the community

If given the opportunity to the project, I will not only work hard and try to finish all the features I proposed in this proposal for GSoC 2018, but also continue to follow the project and improve it in my free time. Or maybe continue the project in the next year 2019! I always believe using the code can change the world. So making something meaningful in another area is a cool thing for me.

5. Report on the task

(As a result of the requirements of the report is 1 – 2 page, full report is available here[10])

5.1 Demo Title

Analyzing Interplanetary Magnetic Field (IMF) data from OMNI. from 2011 to 2017

5.2 Description

IMF is the Sun's magnetic field carried by the solarwind into the interplanetary space. IMF plays an important role in transferring energy into the Earth's atmosphere and generating the northern lights (aurora).
The features which have been finished

1. Show a certain element in the canvas in some days based on hour
2. Show a certain element in the canvas in some months based on day (the number is avg, max or min of the day)
3. Show a certain element in the canvas in some years based on day or month (the number is avg, max or min of the day or month)

Live Demo is here[9]

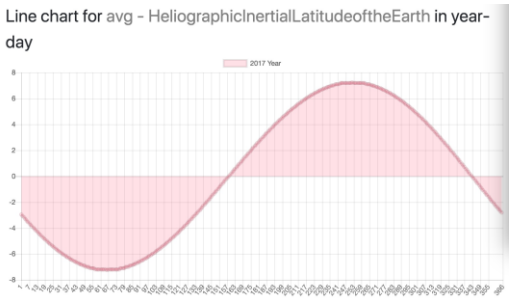
5.3 Teck Stacks

Figure7: Tech Stacks in demo

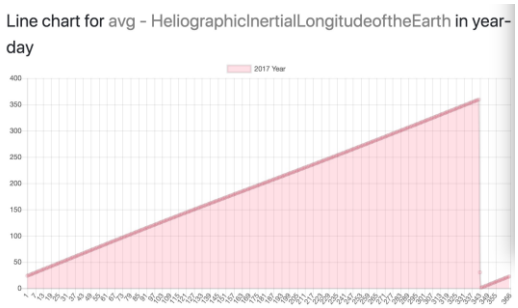
1. Vue (The Progressive JavaScript Framework)
2. Express (Fast, unopinionated, minimalist web framework for Node.js)
3. ES6 (Modern JavaScript)
4. MongoDB (The leading modern, general purpose database platform)

5.4 Some simple conclusions and verification based on the demo

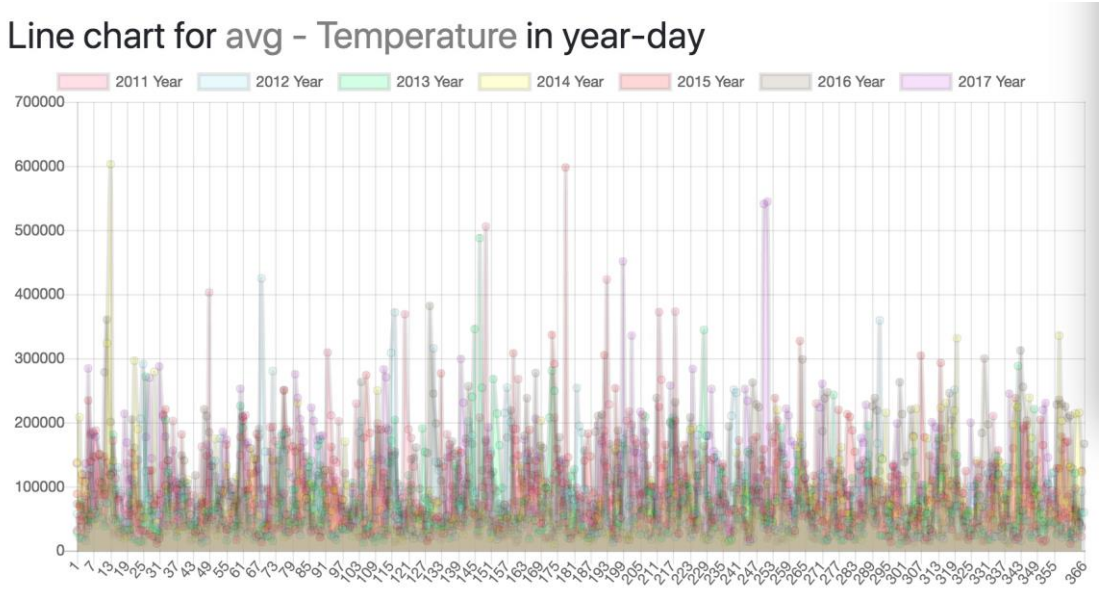
5.4.1 Heliographic Inertial Latitude of the Earth - time should be a sin or cos



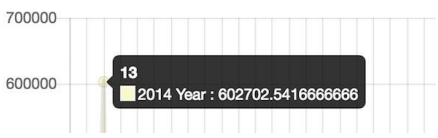
5.4.2 Heliographic Inertial Longitude of the Earth - time should be linear function



5.4.3 The Top 5 hottest day from 2011 to 2017



1. 2014-13 avg: 602702.54K

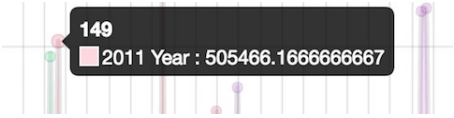


2. 2015-178 avg: 597657.125K

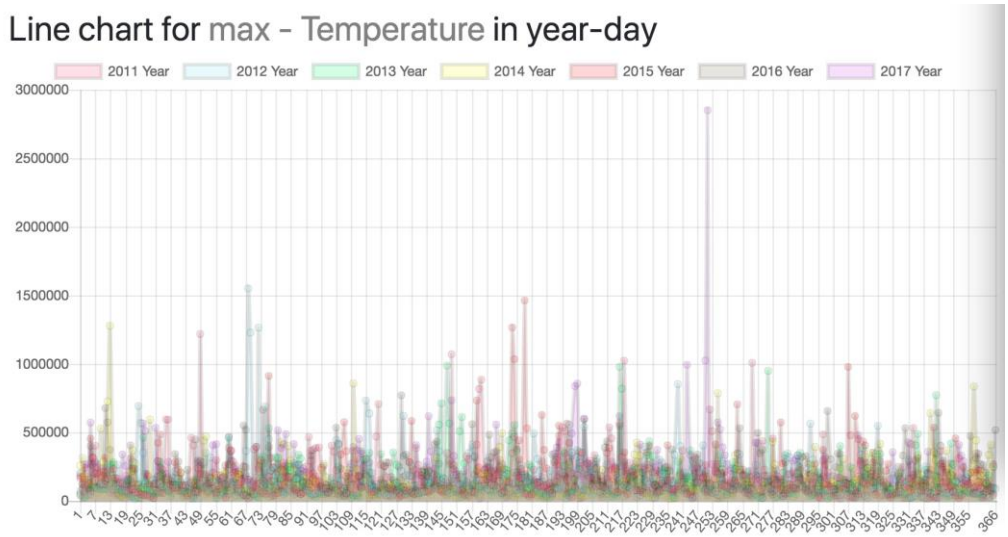
3. 2017-251 avg: 544412.75K

4. 2017-250 avg: 540802K

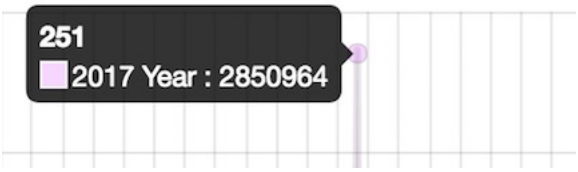
5. 2011-149 avg: 505466K



5.4.4 The hottest moment from 2011 to 2017



2017-251 2am 2850964K



6. Bio-sketch

I am a senior student in BUPT from China and also I am a frontend developer. I have done a lot of full-stack projects using Vue, Express and MongoDB. And also I have some experience with some data visualization tools such as chart.js and g2 in my internship in China. During this period, I used line, bar, bubble and pie charts to show financial data and help the financial researchers to analyze the trend and make further conclusion. I am so interested in the data visualization part because it is totally a different area. In fact, the data visualization tools are becoming more and more powerful than before and I am so excited that it will be used in the research project because it will help the researchers know the data better and easily to analyze. Using the tools I learned in my major to solve the problems in another area is a cool thing for me.

Here is my Github: https://github.com/Tivcrmn

Hope that the project in Space@VT will help me open the door to open source!!!

References:

[1]: Current project website http://superdarn.ca/real-time-display#polar-cap
[2]: echarts.js https://github.com/ecomfe/echarts
[3]: echarts globe demo http://gallery.echartsjs.com/editor.html?c=xrJWQzgc6e.
[4]: http://gallery.echartsjs.com/editor.html?c=xS1MbGycpg
[5]:Flight movements : http://gallery.echartsjs.com/editor.html?c=xHJN4TV5pg
[6]:Animating Contour: http://gallery.echartsjs.com/editor.html?c=xSkHJIBc6l
[7]:Wind Visualization http://gallery.echartsjs.com/editor.html?c=xHJD3BZY5-
[8]:Chart.js http://www.chartjs.org/
[9]: My demo http://tivarea.top/demo/client/index.html
[10]: Report on task https://github.com/Tivcrmn/Space-VT-task2-demo/blob/master/Report.md