

# Map Visualization of Large Data

## CIS5930 - Python Programming Project

Adam Stallard  
Florida State University  
aps10d@my.fsu.edu

Eric Serbousek  
Florida State University  
els16@my.fsu.edu

### 1. INTRODUCTION

This project will take on the task of creating a high performance web application that provides the user agent with a map visualization of a large dataset. Map visualization is a common tool used in many web applications to provide geographical information that pertains to some dataset. When that dataset is large, performance tends to degrade and more advanced techniques are required to present the user with a response in a practical time frame.

### 2. DATASET

The major dataset which will be used by our application is the Shipboard Automated Meteorological and Oceanographic System (SAMOS). This dataset is produced by the Center for Oceanographic and Atmospheric Predictive Studies (COAPS). Essentially this data is collected from research vessels traveling across the globe, and is processed through a quality control system.

The native data resides on a Thematic Real-time Environmental Distributed Data Services (THREDDS) server. This service is provided by Unidata. They also provide relevant python package software to interact with this service at a high level.

Currently, this dataset has been fully indexed for several parameters using an Apache Solr document store engine. This engine provides an enterprise level search platform that can be used for high performance data search and retrieval. There are several python client software packages available for high level interaction with this platform.

This project will utilize the SAMOS dataset, and explore different techniques that facilitate visualizing large point sets on a web-accessible map application.

For this project, other test datasets may be used such as air traffic, satellite tracks, or other geolocated objects. Essentially all that is required for the data to integrate into

the platform is a latitude and longitude pair associated with each object. Since the SAMOS dataset comes with time, animation will be explored as an opportunity for rich graphic visualization. However, the primary goal will be a practical online static map display for large data.

### 3. RESOURCES

For this project, python version 3.5 will be used. To integrate with the Solr back end, the SolrClient library package will be used. This library is chosen for its cursor mark support, a Solr convention used to support deep paging. The web framework that will be used for the project has not been chosen, but will likely either be Tornado, Django, or Flask. Each of these frameworks has great documentation, and work well with various mapping software tool-kits. The mapping software will depend on which web framework is chosen. Depending on what tools are available for these packages, the server side code may require processing using the scipy ecosystem. Scipy contains many scientific libraries written with a C based backend that are capable of powerful rendering in real time.

In terms of hardware, this project may be hosted at COAPS, using the Distributed Oceanographic Matchup Service (DOMS) server. This server is currently being used to serve SAMOS data on a distributed data service application centralized at the Jet Propulsion Lab (JPL). Contention for resources will be closely monitored on the system to ensure that both applications have availability. If problems arise, more resources will be requested for this project. However for testing and early development, it is likely that the page will be hosted on a local machine with all the dependencies installed.

### 4. ALGORITHMS

As of now, there are no algorithms that have been chosen for this project. It is likely that a KD-tree or Quad-tree will be used to manipulate data returned from the Solr back end. These are typical data structures used in mapping applications for their high performance, especially with geo-spatial oriented data.

### 5. CONCLUSION

This project aims to achieve a real time visualization of geographical point data. This will require research on various data structures, web application frameworks, and advanced techniques for displaying aggregated data points on a map efficiently.