Sprint 3 Planning Document
TEAM 16
WeatherPipe

Stephen Harrell, Lala Vaishno De, Hanqi Du, Xiaoyang Lin

1. Sprint Overview

Now that we have implemented several of the core features, this sprint will focus on adding useful functionality to WeatherPipe that would turn it into a more useful and complete research tool. The major points in this sprint would be extending analysis to be carried out over multidimensional arrays, presenting output in the form of writing NetCDF files, running the mapreduce jobs on a local hadoop cluster, building an independent test suite as well as providing a config file to set variables and parameters for the analysis. In addition, there a few user stories we missed in the last sprint that we will be including in this one.

Scrum Master: Stephen Harrell
Scrum Meeting Time: Tuesday and Thursday at 5:00pm

Risks/Challenges:

- Challenges for this scrum will be communication and time. It is the end of the semester and we all have many commitments.
- Learning how to write NetCDF files
- Learning to use a local hadoop cluster
- Setting a config file to take researcher inputs

2. Current Sprint Details

1) Clean up from last sprint:

a. **User Story:** As a user I want to be able to control many parameters of the running of the analysis

Task Description	Estimated Hours	Owner
Check to see that data is available for radar station specified	5	Xiaoyang

Acceptance Criteria:

1. Given that the checking of availability of the radar station flag, when user gives non-existent radar station, I expect that the user will know it is invalid

b. **User Story**: As a user I want to be able to run test cases externally.

Task Description	Estimated Hours	Owner
Delete buckets after running the test cases so that all the users can run the test cases under their account	4	Hanqi Du
To make all the Junit test running externally	10 hours each	Hanqi Du, Xiaoyang

Acceptance Criteria:

1. Able to run the test cases externally.
2. Given a command line, I expect that the user will know whether the functions work or not.

2) **User Story:** As a user I would like to use this tool on a standard hadoop cluster in addition to EMR.

Task Description	Estimated Hours	Owner
Replicate the interface of AwsInterface Class	5	Lala

Learn to use a local hadoop cluster	5	Lala
Draw required data from S3 for each analysis in the hadoop cluster	7	Lala
Run map reduce jobs on a local hadoop installation	15	Lala

Acceptance Criteria:

1. Be able to run and get the right outputs from running the mapreduce on a local hadoop cluster

3) **User Story:** As a user, I would to use this tool to have a config file that allows me to set variables related to the running of the program and of the MapReduce job

Task Description	Estimated Hours	Owner
create general config file	5	Xiaoyang
create separate sections general, emr and hadoop for different types of variables	5	Xiaoyang

Acceptance Criteria:

1. Given the correct general config file, I would expect that it is allowed for users to change variables easily
2. Given the separate sections of config file, including general, emr and hadoop, I expect that users can make change for different types of variables

4) **User Story:** As a user, I would like an easy to understand help menu that describes all flags as well as their config file equivalents.

Task Description	Estimated Hours	Owner
Write help menu about all flags	5	Xiaoyang
include information about config file and equivalent flags	5	Xiaoyang

Acceptance Criteria:

1. Given the help menu about all flags, when users run the program, I expect that the users know what flags they need.
2. Given the information about config file and equivalent flags, I expect that the users know how to use config file.

6) **User Story:** As a user, I want to be able to get results in the form of a histogram or a NetCDF file so that I can better understand the results of my analyses.

Task Description	Estimated Hours	Owner
Write job to hold netcdf files in the classes.	5	Stephen
Write output file writer to write out netcdf file.	10	Stephen
Convert netcdf file to histogram.	5	Stephen

Acceptance Criteria:

1. Be able to write out NetCDF files as output in the correct format.
2. Be able to produce histograms from the outputted NetCDF files, if time permits

7) **User Story:** As a user, I want to be able to perform an complex analyses involving several radar stations and 3 dimensional objects.

Task Description	Estimated Hours	Owner
Create average analysis over area.	10	Stephen
Create job to transfer 3D information.	5	Stephen

Acceptance Criteria:

1. Be able to transfer 3D data between the mappers and producers.
2. Be able to perform an average analysis over a geographical region and produce the required output.

8) **User Story:** As a user, I want to be able to be given the choice to terminate the process at any time so that I can control the process.

Task Description	Estimated Hours	Owner
Accept the cntrl-c from users.	3	Hanqi Du
Send the termination signal to Amazon to terminate the current process	3	Hanqi Du

Acceptance Criteria:

1. Given cntrl-c, I expect that a signal can be sent to Amazon to terminate the process.

9) **User Story:** As a developer I would like tests to make sure my changes work correctly.

Task Description	Estimated Hours	Owner
Create unit tests for all functions that it is possible for.	12	Hanqi Du

Acceptance Criteria:

1. Test suite can run and pass
2. Test every testable function

3. Remaining Backlog

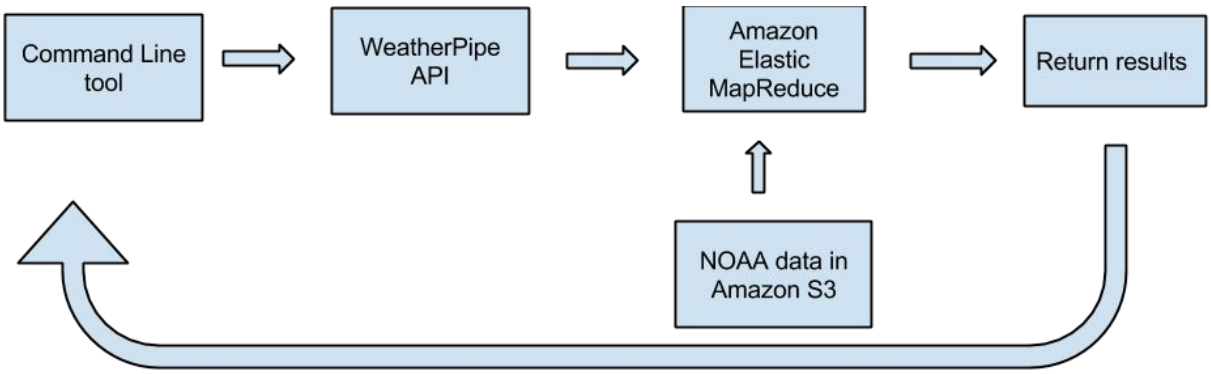
Functional Requirements:

1. As a user, I want to be able to get results as a single value or in a table format so I can interpret the results, if time permits.
2. As a user, I want to see a time remaining indicator so that I know how long it would take so I will know when it is done, if time permits.
3. As a developer, I want an index of s3 datasets to be able to retrieve specific data related to dates and geospatial shapes so I can find the data that is relevant to my research, if time permits.

Non-Functional Requirements:

1. **Optimize by cost :** Optimize the cost of using amazon servers by finding the the most economical server-time combination that would minimize the cost of running the analysis in a suitable time frame. There are two types of research that dictate time frames. The first is operational research which is data that goes into a current forecast. The second is research that may go into a paper or journal article. The time frame for the first type of research is typically 4 hours and two weeks for the second one.

2. **Design:** By using a modular design such that radar data can be swapped with other data, this system can be used as a dock for weather processing using different sets of data from different sources. The command line tool will use the WeatherPipe API. This will allow us to develop other interfaces if we need to. The WeatherPipe API will take an Amazon security token, type of analysis, specific dates and location then it will submit the job to MapReduce. The analysis will pull the correct data from the NOAA repository of radar data in S3 and run the MapReduce. The results will be returned to the command line tool.



3. **Output Format:** The output of our initial analyses will be integers or arrays of integers. We will be enumerating different kinds of analysis beyond this with Dr Baldwin next week.

.

4. **Security:** We will be using Amazon AWS Security Tokens so we need to use them and delete them out of memory quickly.

5. **Demos/Tutorials:** We will write documentation to explain how to submit analyses as well as get results.

6. **Failure Modes:** For individual failures within a piece of the analysis we will retry at least 5 times before giving up. On the whole for the entire analysis we will give the researcher control over the failure rate and at which point the job should be abandoned wholy.

7. **Better analysis tools:** We will make the analyses pluggable so weather researchers can write their own analyses. Each Analysis requires it's own inputs and outputs. Although many of them overlap each analysis will require it's own config file, analysis code and output code.