Sprint 1 Planning Document
TEAM 16
WeatherPipe

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

1. Sprint Overview

This sprint will be focused on getting a bare-bones working prototype. Much of the time will be spent in research prototyping. By the end of the sprint we will have a command line tool that will submit an average analysis using a pre-coded hive script to EMR using NEXRAD data and return on average at a specific geospatial point.

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

- Submitting jobs to EMR
- Downloading data from S3
- Learning Hive Scripts
- Learning MapReduce

2. Current Sprint Details

1) **User Story:** As a user, I want to be able to choose specific time periods from the data such as a range of dates or as a specific scheme of dates so that I can limit and choose the data set to the sample for my calculations.

Task Description	Estimated Hours	Owner
Get the time periods from the input specified by the user and modify them into proper format.	4	Hanqi Du
List all the files in the bucket.	6	Hanqi Du
Search for the files we need according to the time periods defined in the input.	15	Hanqi Du

Acceptance Criteria:

1. Given the input from user in correct format when I need the time periods I expect no errors appear when I read the input and be able to modify it into proper format.
2. Given the bucket when the code executes I expect all the files in the bucket can be listed without errors.
3. Given the time periods in the proper format when the code executes I expect all the files we need can be found according to the time periods.

2) **User Story:** As a user, I want to be able to do my analysis from the command line with a command line tool.

Task Description	Estimated Hours	Owner
Argument Parsing	3	Stephen
Argument Type and Correctness Checking	4	Stephen
Interfacing with other Classes to Launch Analysis on EMR	6	Stephen

Acceptance Criteria:

1. Given a correct command line execution when the code executes I expect the command line interface will give no errors.
2. Given a command line execution with wrongly typed inputs I expect the command line interface will give the user a useful error message.
3. Given a correct command line I expect that the the analysis will be launched by the other components of this software.

3) **User Story:** As a user, I want to be able to do a simple analysis (mean) on the chosen data set so that I can analyze historical data.

Task Description	Estimated Hours	Owner
Write file downloading	5	Xiaoyang
Write results return	5	Xiaoyang
Write simple historical data analysis (mean) function	10	Xiaoyang

Acceptance Criteria:

1. Given the file from the hive script I expect to be able to download and use the data for an analysis
2. Given the raw data I expect to do a simple analysis(mean) on the the data.
3. Given the output of the analysis I expect to report the analysis output to the command line tool.

4) **User Story:** As a user, I want to be able to save the results in a file from a certain run so that I can use it later.

Task Description	Estimated Hours	Owner
Get the results.	5	Hanqi Du
Save the results to a file.	5	Hanqi Du
Store the file in the local machine.	10	Hanqi Du

Acceptance Criteria:

1. Given the results when code executes I expect the results can be got without errors.
2. Given the results above when code executes I expect the results can be saved in a file.
3. Given the file of results when code executes I expect the results can be saved in the local machine without errors.

5) **User Story:** As a developer, I want to be able to retrieve radar data from s3 so that I can do analysis.

Task Description	Estimated Hours	Owner
connect to s3 and download data file	3	Xiaoyang
uncompress the files to get data	3	Xiaoyang
load the files in NETCDF for use by analysis	10	Xiaoyang

Acceptance Criteria:

1. Given the correct file name from the hive script I expect to download the data file
2. Given the the compressed data file I expect to uncompress the file without writing it to disk.
3. Given the uncompressed data stream I expect to read it with the NETCDF reader classes.

6) **User Story:** As a developer, I want to be able to authenticate an AWS account and get access to compute resources.

Task Description	Estimated Hours	Owner
Use Amazon API (S3 or EMR) to authenticate AWS credentials	2	Lala
Write the AWSInterface class that will be responsible for running/launching jobs on EMR.	10	Lala
Add statistics feature to the AWSInterface such as cost of running the job.	3	Lala

Acceptance Criteria:

1. Given that the Amazon API authentication has been correctly implemented, when I try authenticating it using AWS credentials, I expect it to authenticate successfully.
2. Given that the AWSInterface class has been correctly implemented, when I try submitting a job and launching it, I expect AWS to successfully create a job and run it to produce some results.
3. Given that the jobs are running successfully using the AWSInterface, when the job ends or terminates unexpectedly, I would expect some statistics such a cost of run, time taken or cause of failure to be returned.

7) **User Story:** As a developer, I want to be able to use EMR so that I can do analysis.

Task Description	Estimated Hours	Owner
Generate Hive Script from the inputted parameters	12	Lala
Create Payload that contains the Hive script and the analysis and check if the payload has been formed correctly.	2	Lala
Upload payload to S3 and ensure it has been uploaded successfully.	1	Lala

Acceptance Criteria:

(1) Given the Hive Script generator works somewhat correctly, when I try creating a hive script, I expect the hive script to be created from the input parameters and for it to be syntactically correct.

(2) Given that creating payloads is successfully implemented, when I try creating the payload, I expect a payload to be created successfully.

(3) Given that the uploading payloads is successfully implemented, when I try upload the payload, I expect the chosen payload to be uploaded to S3.

8) **User Story:** As a developer, I want to be able to handle failures by reporting so that I can figure out the failure.

Task Description	Estimated Hours	Owner
Create general pattern for failure reporting	8	Stephen
Insert pattern into existing code and teach others about pattern	4	Stephen
Write reporting output interface	5	Stephen

Acceptance Criteria:

1. Given that EMR failures occur during runtime I expect the failures to get reported and sent back to the command line tool.
2. Given that the other team members listen I expect this pattern to be in many of the classes that we are writing.
3. Given that the EMR failures occur during the runtime I expect the failure data to be displayed in a useful way to the user.

3. Remaining Backlog

(a) Include all the other user stories from your Product Backlog document.

Functional Requirements:

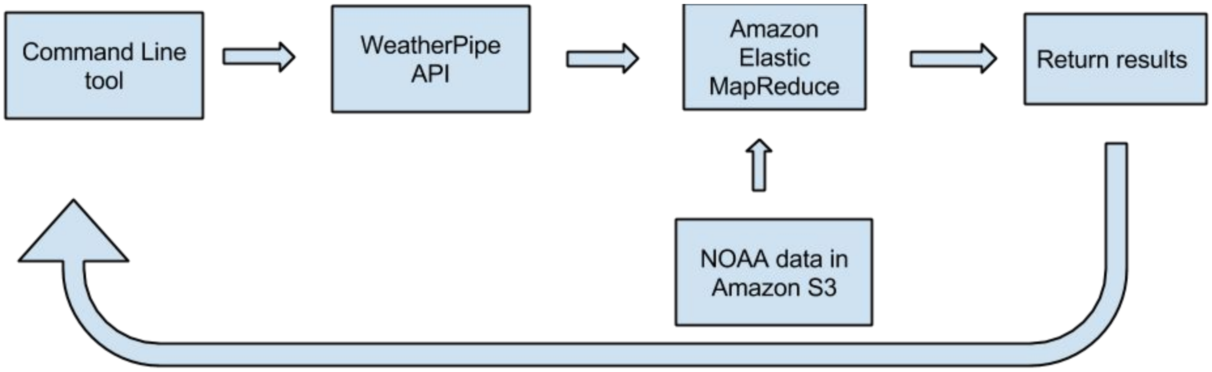
1. As a user, I want to have a simple config file to define inputs for each analysis, if time permits.
2. As a user, I want to be able to get results as a single value or in a table format so I can interpret the results, if time permits.
3. As a user, I want to be able to get results in the form of a histogram or a multivariate histogram or a 3D Space average diagram so that I can better understand the results of my analyses, if time permits.
4. 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.
5. As a user, I want to be able to be given the choice to restart, terminate the process at any time so that I can control the process, if time permits.
6. 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.
7. As a developer, I want to unit tests so as we make changes we can test them quickly, if time permits.
8. As a developer I want to be able to enumerate the different kinds of analysis that are possible so that I can generalize a solution.
9. As a developer I want to be able to enumerate the different kinds of data get output so that I can generalize a solution.

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.