# Sprint 2 Planning Document
# TEAM 16
# WeatherPipe

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

## 1. Sprint Overview

Now that our prototype is finished, this sprint will focus on making it more robust, adding important core features and finishing up what we didn't get to in the last sprint. The major points will be having everyone learn the build system, creating an auto build system for the MapReduce jar file, implementing a more complex analysis and creating a simple method for others to add code to the analysis without changing anything else. 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:**

- Making this easy for a user to use
- Learning the different components
- Modifying the build system to make the project simpler
- Communication

# 2. Current Sprint Details

1) **Clean up from last sprint:**
    a) **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 Type and Correctness Checking | 5 | Xiaoyang |

**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.

    b) **User Story:** As a user, I want to be able to get statistics and estimated cost from the tool.

| Task Description | Estimated Hours | Owner |
|---|---|---|
| Add statistics feature to the AWSInterface such as cost of running the job. | 3 | Lala |

**Acceptance Criteria:**

1. 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.

    c) **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 |
|---|---|---|
| Downloading log files and inputs for user | 5 | Stephen |

**Acceptance Criteria:**

1. Given that the EMR failures occur during the runtime I expect the failure data to be displayed in a useful way to the user.

2) **User Story:** As a user, I want to be able to have time periods larger than a day for my analysis.

| Task Description | Estimated Hours | Owner |
|---|---|---|
| Determine which days (inclusive) are within the time range | 9 | Hanqi |
| List every bucket (one bucket per day) that is needed for the search. | 6 | Hanqi |
| Search for the files we need according to the time periods defined in the input. | 5 | Hanqi |

**Acceptance Criteria:**

1. Able to do analysis on more than one days worth of data
2. Multiple bucket lists are searched

3) **User Story:** As a user, I want an integrated build system so that the Map Reduce jar is built with the command line tool

| Task Description | Estimated Hours | Owner |
|---|---|---|
| Design and implement a multi-target build system that auto-builds the map reduce jar file | 20 | Stephen |
| Set the code to automatically find the built Map Reduce jar file | 5 | Stephen |

**Acceptance Criteria:**

1. Able to build entire project with one gradle build command
2. Able to run the command line tool without moving or specifiying the jar file

4) **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 |
|---|---|---|
| Add Radar Station flag | 5 | Xiaoyang |
| Check to see that data is available for radar station specified | 5 | Xiaoyang |
| Add Job ID and Bucket name flag | 5 | Xiaoyang |

**Acceptance Criteria:**

1. I expect the radar station flag to be taken on the command line.
2. 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
3. I expect the JobID and bucket name to be taken on the command line and have acceptable defaults.

5) As a user I want to be able to do analysess that have arrays for outputs

| Task Description | Estimated Hours | Owner |
|---|---|---|
| Modify analysis to do an average over an array | 20 | Lala |
| Create output system that can proccess MapReduce output into arrays in json | 12 | Lala |

**Acceptance Criteria:**

1. Able to write an analysis that outputs an array
2. Able to get an json file as output from the job

6) 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. | 20 hours each | Hanqi/Xiao yang |
| Create regression test for entire command line tool | 5 | Stephen |

**Acceptance Criteria:**

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

7) As a user, I would like to be able to modify two simple classes to change my analysis without having to modify the bulk of the project.

| Task Description | Estimated Hours | Owner |
|---|---|---|
| Create super class that can be subclassed for analysis | 13 | Stephen |
| Give the user access to the netcdf file and outputting functions | 2 | Stephen |

| | | |
|---|---|---|
| Give the user an easy to use output function for different types of data (int, array) | 5 | Stephen |

8) As a user, I would like to be able to search a large range of dates quickly and run my analysis on larger number of files.

| Task Description | Estimated Hours | Owner |
|---|---|---|
| Create a thread pool when the number of days is greater than 0 | 3 | Lala |
| Synchronize necessary methods and termination of the thread pool. | 2 | Lala |

# 3. Remaining Backlog

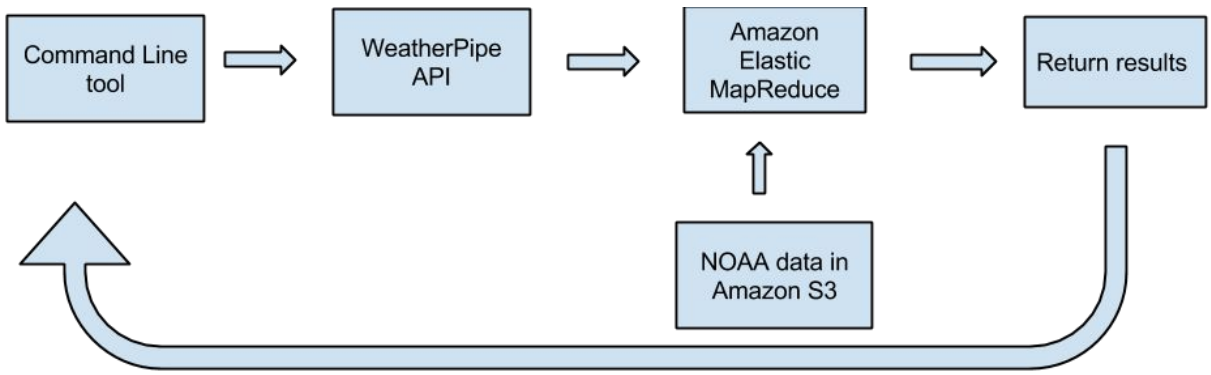
## Functional Requirements:

1. As a user, I want to have a simple config file to parameters for the analysis, if time permits.
2. As a user, I want to have a well written help menu to help me work with my jobs.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. As a user, I want to be able to use this analysis on my own map reduce cluster without using Amazon.

## 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.