Location Aware Scientific Workflows — Nextflow Library Creation and Testing

Robin Jonker (1827572)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: The creation and testing of a location-aware scientific workflows library is presented. This library automatically sets the execution location of processes based on the locality of the principal data required. The library is created and tested on the Wits Core Research Cluster, an HPC environment. The library is successfully tested using existing workflows in Nextflow DSL 2 on the Wits Core Research Cluster. The library reduces execution times by more than 50% under all conditions. Repeating the same workflow version 15 times, it was found that the standard deviation of execution times is lowered by more than 85%. Testing also shows as the number of input files increases, the dynamic location-aware library's performance increases in comparison. The library's testing data is validated with a second workflow. The library along with various aspects such as development documentation and the different workflow versions can be found on the GitHub repository here.

Key words: HPC, Locality, Nextflow, Processes, Workflows

1. INTRODUCTION

Workflows are used to solve complex scientific programs [1]. Complicated interdependent programs feed into one another in order to achieve the desired outputs. Workflow languages such as Nextflow enables developers to coordinate the calling of the different procedures and their location of execution. This creates highly scalable, portable, and reproducible workflows that take full advantage of parallel computing [2]. The ability to assign processes to different distributed environments ensures the program can execute regardless of the location of the input data.

The central issue that arises is when the data that is used for the workflow is stored in a different location than that of the computation [3]. This causes the workflow execution to incur a vast increase in the load on the network and reduces the performance of the workflow significantly. Therefore, if the computation can occur in the same location as the data that is stored, the load on the network will reduce and the workflow performance will increase considerably in comparison.

In this report, a Nextflow library that automatically sets the execution location of the code based on the locality of the principal data required is created and its results are critically analysed. Section 2 gives background research on the different technologies and services used for this project. Section 3 refers to the research objectives and methods. Section 4 breaks down the different design solutions that are implemented whilst Section 5 showcases the testing and results of the library. Section 6 critically analyses many areas within the results, the overall project, and possible future improvements that can be considered for this library.

2. BACKGROUND RESEARCH

In order to understand the design and implementation of the library, there are technologies and services that need to be broken down further. It is assumed that the reader has general software development knowledge and understands the basic concept of high-performance computing (HPC); however, the concepts will be simplified below.

2.1 Slurm Workload Manager

Slurm is a job scheduler that allows multiple programs to be submitted and queued simultaneously until the required resources that are allocated to the job become available [4]. It also manages the resources and processors allocated to each job. Within a high-performance computing cluster, several nodes can store data and execute programs. With a simplified outlook, each node can be considered its own computing device. The Slurm scheduler is in charge of setting which device will perform the specific process within a workflow.

2.2 Gluster Software

The Gluster software is a scalable filing system for networks [5]. It provides a distributed storage solution for computing clusters. It is a file system that can scale up to petabytes of data whilst handling thousands of clients. A key attribute of the Gluster software is that it stores a duplicate of data to ensure reliable access to data in case a node goes down, in the Wits cluster's case the data is generally stored on 2 nodes. The Gluster software provides an abstraction layer of a unified file system to the entire cluster.

2.3 Nextflow

Nextflow is the workflow language of choice allowing interdependent processes to be executed using the Slurm scheduler [1]. Nextflow allows the integration of Groovy-coded functions along with allowing Bash commands within the script of each process. The key attribute of Nextflow that allows this library to be possible is their built-in directive called clusterOptions [6]. This allows specific cluster submission commands

to be given that set the settings within the execution. In our library, a list of nodes is passed as the parameter for this directive that sets which nodes should be excluded when considering nodes to process on. The exact method of deciding which nodes are not suitable for execution is broken down in Section 4.

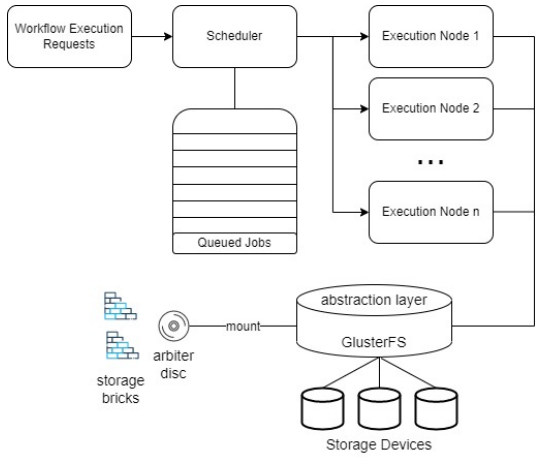


Figure 1 : Diagram showing how the Slurm scheduler sets inputs into queued jobs before allocating execution nodes which connects to the Gluster abstraction layer that unifies a set of nodes into a file system for the entire cluster

3. RESEARCH OBJECTIVES AND METHOD

The main focus of this project is creating a library that can be added to existing workflows that makes those scientific workflows location-aware. The success criteria of this project are creating and testing a library that works on the Wits Core Research Cluster with the following key targets:

1. Develop techniques to set code execution location to match the data storage location.
2. Write a Nextflow/Groovy library
3. Test the library by modifying an existing workflow in Nextflow DSL 2
4. Conduct experiments to test computational efficiency of new code

4. DESIGN SOLUTION AND IMPLEMENTATION

In summary, workflows are submitted using the Slurm scheduler with the specific list of nodes to be excluded passed into the clusterOptions directive that sets the execution location of the processes. This reduces the negative effect on performance that transferring the data has. The subsections to follow will dive deeper into the different versions of the library created along with how and when the execution locations get set. There will be 3 different versions of code designed to be tested. A standard non-library DSL 2 version of a workflow, a static location-aware scientific workflow, and a dynamic location-aware scientific workflow. Figure 2 shows a simplified activity diagram of the non-library workflow execution using a Slurm scheduler and the unified Gluster filing system. Figure 3 and 4 show the static version and the dynamic version's activity diagrams respectively.

4.1 DSL 2

Nextflow is a domain-specific language (DSL) designed especially for use within workflows. Among the success targets for this project is modifying an existing workflow in Nextflow DSL 2. DSL 2 refers to the specific syntax extension of Nextflow which allows the definition of module libraries and aids in complex pipeline creation [7]. The key alteration occurs with the distinct workflow module needing to be set along with setting specified inputs and outputs within it instead of it being directly assigned from different processes. This allows explicit process connection and module inclusion. A workflow is converted into DSL 2; however, the basic principles of its activity remain the same.

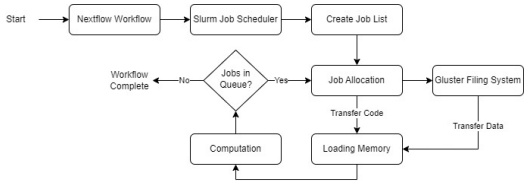


Figure 2 : Activity diagram of non-library version of workflow using Slurm scheduler and Gluster filing system.

4.2 Scientific Location Aware Library

We developed a Groovy library to set the location of computation to that of the data. There are two different versions of this library, one which sets the location statically at the start of the workflow execution and another that sets the location of computation dynamically for each input file as they get processed. The static and dynamic versions of the library both utilise similar techniques to extract the information needed to set the ideal execution location. There are 3 relatively common functions for both versions, with dynamic having another function that will also be explored.

1. Determine the node location where a file is stored. Using the following command, along with the file passed as a parameter, queries the Gluster software to determine where the file is stored by checking the arbiter discs. After extensive string manipulation, the function returns the nodes where the file is stored:

```
getfattr –n glusterfs.pathinfo X
```

where X = -e text $file_name

2. Determine which nodes are currently available for execution on the cluster. Using the following command, the state of all the nodes in the cluster is queried. With string manipulation and state separation, a map of all the nodes' state and a list of all possible nodes for execution is returned:

   `sinfo –p batch –O N,S`

   where N = NodeHost and S = StateCompact

3. Determine if storage nodes are also available for execution. This function calls the previous two functions and intersects the returned values to check if there are any matches. clusterOptions is set with all the nodes that need to be excluded during execution. Therefore if there are matches, a list of possible nodes excluding the matches is returned, and then it will process on the matches. If there are no matches, the Gluster software will determine which general node to execute on and incur the performance strain of transferring the data.

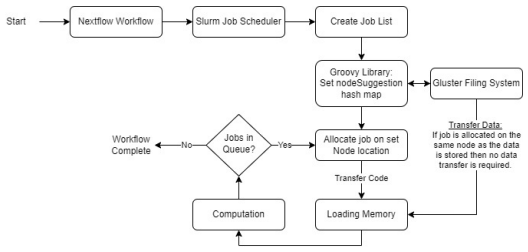


Figure 3 : Activity diagram of static version showing how the map of node locations is set before the process computation loop.

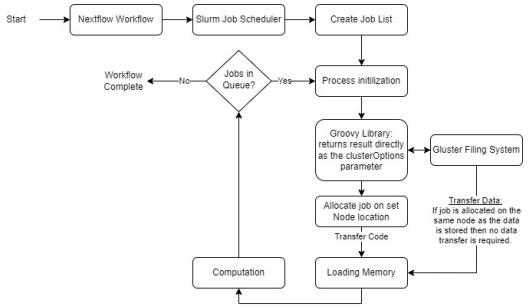


Figure 4 : Activity diagram of dynamic version showing how the Groovy library setting the node location is called for each process in the computation loop.

4.2.1 Static Location Aware Version: The static version executes all the functions in Section 4.2 at the start of the workflow execution. The results that indicate which node to execute on for all the different files are stored in a hash map. This map then gets queried within the process to set the clusterOptions directive for each file. This is shown in Figure 3.

4.2.2 Dynamic Location Aware Version: The dynamic version executes all the functions above, along with an additional function, throughout the processing of the workflow. The new function that is created is used to determine the ideal node to execute on. The static version returns all nodes that are available, regardless of how busy those nodes currently are, as nodes that are available to execute on. However, an additional function is created for the dynamic version which sets a priority scale based on the node's state and multiple factors to determine if any of the storage nodes are best for execution. Instead of a hash map storing the results at the start of the workflow execution, the dynamic version calls the functions and sets which node to execute on using clusterOptions for every different input file as each file gets processed. This is shown in Figure 4.

5. TESTING AND RESULTS

There are 3 different versions of the same workflow that is tested and their results are presented below. All 3 the versions are tested on the Wits Core Research Cluster, which will be explored below, under specified common conditions. The simple non-library version, the static, and the dynamic scientific location-aware versions are tested when the Wits cluster is largely idle and unloaded, and tested again when the cluster is under strain and loaded. The 3 versions of the workflow, with unloaded conditions, are repeated 10 times to determine an accurate average set of results. Under the loaded conditions, the workflow versions are repeated 5 times each. The impact that the number of input files has on the execution time of each version is also tested. Finally, the testing results are validated by testing all the versions of the library on a second workflow. The testing data can be found here.

5.1 Wits Core Research Cluster

The Wits cluster is a high-performing distributed environment to enable the processing of many different intensive tasks such as workflows whilst storing all the input data required for the workflows. The Wits cluster consists of 38 nodes, with 20 of them having the Gluster software installed on them. The data is stored in bricks of 3, of which each set consists of 2 storage nodes and an arbiter disc which stores information about the nodes and their contents without needing to access the large storage nodes for every query. The Wits cluster was the environment used throughout the testing of the library.

5.2 Comparison of execution times of different versions

The workflow is repeatedly tested under specific cluster conditions to determine which version executes the best depending on the state of the cluster. The unloaded cluster scenario, shown in Figure 5, has at least

15 idle nodes on workflow execution. The loaded cluster scenario, shown in Figure 6, has at least 13 allocated nodes on workflow execution.

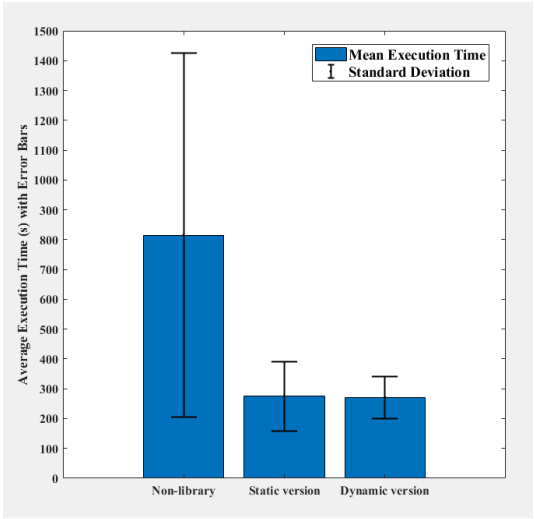


Figure 5 : Average execution time (s) with error bars for the simple, static, and dynamic versions while cluster is unloaded

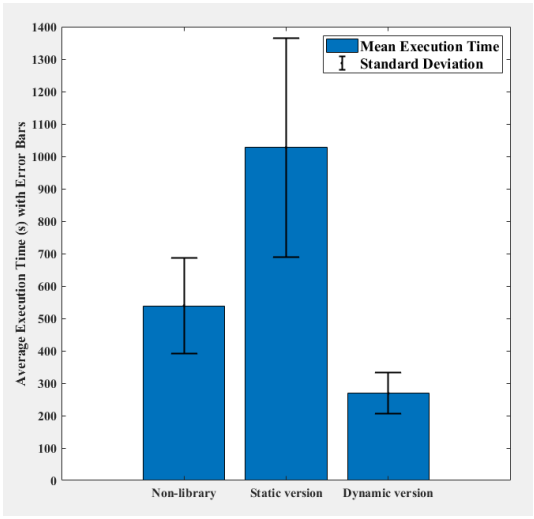


Figure 6 : Average execution time (s) with error bars for the simple, static, and dynamic versions while cluster is loaded

As shown in Figures 5, the use of the library for either version under no load reduces the execution time by more than 60%. Figure 6 shows how negatively a load of a cluster affects the execution time of the static version. Both the non-library and static versions are inconsistent with large error bars and perform considerably worse than the dynamic version.

5.3 Impact that the size of input files has on the execution times of different versions

In the workflow that is being tested, there are 168 different files that form the input channel. The workflow process is run for every file. In the comparison above, the workflows were executed to process all of those files. In the next test, we will measure the impact that the number of files has on the execution time of each version. Each workflow version will be executed using only the first 10 files, the first 90 files along with the execution time for all 168 files shown above. These tests were done while the cluster was under no load with at least 30 idle nodes.

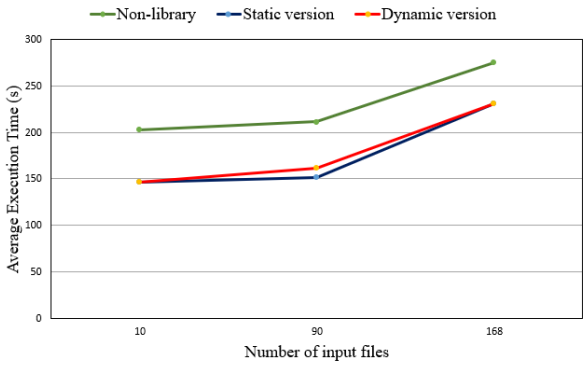


Figure 7 : Average execution time (s) of versions with different input file sizes

6. CRITICAL ANALYSIS

6.1 Results

Figure 5 and 6 show the average execution times of the different versions while the cluster is unloaded and while it was loaded. The general expectation for all the versions is that they will perform worse while the cluster is loaded as fewer nodes are available for execution compared to the unloaded cluster. The non-library version surprisingly performed better while the cluster was loaded. Based on the variance among the testing iterations, which will be explored further below, it can be considered that over a larger sample size, the results will perform as expected. The static location-aware version performed drastically worse while the cluster was under load. The reason this occurs is that at the start of the workflow execution, the location of computation is set, and by the time the workflow needs to process a file later on, that node that was set as the best node in the beginning, is no longer available causing the workflow to wait until it becomes available. This is the big flaw with the static version. While the cluster is unloaded, there are no competing processes and the best node at the start of the workflow will remain in the same state later on when it needs to be executed on. The dynamic location-aware version performed con-

sistently regardless of the load on the cluster. This is because the location of computation is determined as each process needs to be executed therefore the best node to execute on is always selected.

When comparing the impact that the number of input files being used for each workflow has on its execution time, shown in Figure 7 above, it was found that the static version performs slightly better with a lower number of input files compared to the dynamic version; however, as the number of files increases, the rate of increase in execution time is greater than that of the dynamic version. Therefore, as the number of input files increases, the most suitable version is the dynamic version. This is because as the number of files increases, the probability that the best node for computation, which is set at the beginning for the static version, remains the best node drastically lowers.

It is important that a library is stable and its results are repeatable. The consistency of results for each version is analysed further.

6.1.1 Variance: Variance refers to the change in execution times when the same workflow version is repeated. The standard deviation of the execution times of each version that is repeated 15 times is shown along with the peak variance. The peak variance refers to the difference between the execution iteration that performed the best and the iteration that performed the worst. This is combining both the unloaded and loaded cluster data, so it is the error regardless of cluster conditions. Table 1 shows the standard deviation and peak variance for each of the different versions. For the code to be reliable, the changes in results between testing iterations should not be as noticeable. This shows that the dynamic version is the most stable code as the results outputted are consistent whereas the simple non-library and the static versions both have very inconsistent results.

Table 1 : Variance (s) of different versions of workflow

Version	Standard Deviation	Peak Variance
Non-library	522	1766
Static	415	1421
Dynamic	68	280

The reason that there is such a large variance in both the static and the simple non-library version is due to the nature of its execution. For the static version, as mentioned above, its computation location is determined at the beginning. On some occasions, the best node will still be available when it needs to be executed on, and thus it performs well; however, on other occasions, the workflow is extremely slow due it waiting for the node to become available. Similarly, the non-library version relies on the Gluster software to determine the computation location. On some occasions the data storage node is available and by chance, the software sets this node as the location of computation. This is why the variance of the results is so large across the testing iterations.

6.1.2 Summary: The critical analysis of the testing results proved that the use of the dynamic version of the library not only greatly reduces the execution time but is also far more stable in use on the cluster.

6.2 Overall Project

The research objectives for the project were successfully met. The project plan was followed and the weekly Sprints were largely successful. A lot of time was spent learning and understanding a new field of software. If the learning period could have been shortened, the library could have been made to be more advanced and more testing could have been done. A complete agile development process was followed, consistent weekly meetings and proper teamwork allowed the library to be completed. The whole development process and iterations of versions along with all documentation can be found on the GitHub repository here.

6.3 Future Research and Improvements

The library was created and tested for use within a high-performance cluster environment. Another area that will be discussed in more detail below, is adapting the library to function for a cloud computing environment too, such as Amazon Web Services. In addition to AWS integration, the inclusion of a switching policy and search optimization will be explored.

6.3.1 Amazon Web Services: AWS is a major cloud computing solution for a very wide array of different tasks [8]. AWS allows the creation of an EC2 instance which is a virtual computing environment. This allows Nextflow workflows to be executed on AWS. In order to execute a workflow, the workflow is submitted to AWS Batch which requires the code to be containerised, using software such as Singularity which will be discussed below. Input data for workflows can be stored within S3 buckets. A computing environment along with a job scheduler is linked to allow the Slurm scheduler to process the workflow. The main difference between this AWS setup and using the Wits cluster is that the data stored on AWS is stored in general storage with S3 and not on specific nodes. This distinction causes the library to be largely different in functionality but with a similar goal. Both methods increase performance by reducing data transferring to the location where the code is executed. Wits cluster is based around executing on specific nodes where the data is stored whilst AWS is based around executing the code locally or with AWS

batch depending on where the input data is stored.

6.3.1.1 Singularity: Singularity is a containerisation software that enables the packaging of the source code along with all the required libraries and dependencies. Singularity is the alternative to the more popular Docker containerisation software, however, within high-performance computing, Docker has security flaws and does not allow integration. Singularity is more efficient for large-scale environments and was specially designed for HPC systems and Batch submissions. Therefore to execute with AWS Batch, the workflow code needs to be containerised using Singularity.

6.3.1.2 AWS Scientific Location Aware Workflow: The key difference between the AWS version of the library and a simple non-library version, is the process separation to handle different data. One process will perform scripts related to data stored locally and another process, performing the same actions, will perform scripts related to data stored in S3 buckets. Nextflow has a special feature called hybrid workloads which allows only processes with specific labels to be executed using a specific executor. The use of module inclusion with added parameters removes the replication of processes to comply with the DRY software principle. The Nextflow 'into' operator is used to split the input data channel into two datasets for their respective locations that can be used in their respective processes. The AWS version has not been tested successfully due to the inability to get AWS Batch to process correctly, the processes execute successfully when run locally on an EC2 instance, but not when the workflow is set to execute with the Batch scheduler. Therefore, enabling AWS integration for the library is left under future research and improvements.

6.3.2 Switching Policy: The addition of a switching policy to combine the static and dynamic versions of the workflows can be useful. Through more intensive testing it could be found that on some occasions the static location-aware version outperforms the dynamic, and if that occurs then you will want the library to consist of only one version that automatically switches between the static and dynamic under specified conditions.

6.3.3 Search Optimization: One of the main functions of the library is searching through all the nodes to determine their state and which files they contain. This library focused on node selection optimization, but in future, the searching through the nodes can be optimized. Methods such as graph-based search algorithms should be explored [9].

7. CONCLUSION

A Nextflow library that makes scientific workflows location aware was created and tested. The use of the library far outweighs using the simple built-in Slurm Gluster node allocation. The success criteria for the project were met. Using the dynamic version of the location-aware workflows reduces execution time under all conditions by more than 50%. The results were critically analysed to find that the standard deviation among repeated testing iterations drops from 521.5 to 68.4 seconds comparing the simple non-library and the dynamic versions. The impact that the number of files has on execution time shows that as the number of files increases, the performance of the static version deteriorates in comparison to the dynamic version. The library was also validated using a second workflow. Finally, the overall project and future improvements were analysed.

REFERENCES

[1] P. D. Tommaso. "Nextflow enables reproducible computational workflows." *Nature Biotechnology*, vol. 35, no. 4, pp. 316–319, Apr. 2017.

[2] E. Nogales. "Nextflow integration for the Research Object Specification." *ResearchGate, DOI:10.5281/zenodo.1472385*, Jul 2018.

[3] A. Szalay, A. Bunn, J. Gray, I. Foster, and I. Raicu. "The Importance of Data Locality in Distributed Computing Applications." *NSF Workflow Workshop*, Jan 2006.

[4] M. Jette, A. Yoo, and M. Grondona. "SLURM: Simple linux utility for resource management." Jul 2003.

[5] V. Kashansky and I. Kaftannikov. "Application of SLURM, BOINC, and GlusterFS as Software System for Sustainable Modeling and Data Analytics." vol. 173. Feb 2018.

[6] P. D. Tommaso. "Processes: clusterOptions." *Nextflow*, Feb 2014. Available at `https://www.nextflow.io/docs/latest/process.html?highlight=clusteroptions`.

[7] P. D. Tommaso. "DSL 2." *Nextflow*, Apr 2019. Available at `https://www.nextflow.io/docs/latest/dsl2.html`.

[8] P. D. Tommaso. "AWS Cloud." *Nextflow*, Sep 2016. Available at `https://www.nextflow.io/docs/latest/aws.html`.

[9] R. Patel and M. Pathak. "Comparative Analysis of Search Algorithms." *International Journal of Computer Applications*, vol. 179, no. 50, Jun. 2018.

APPENDIX A - GROUP REFLECTION

This appendix contains the group reflection for this project. My group partner for this project was Tristan Lilford (1843691). We were group 22G51 working on project 22P63.

Working as a group was successful. We worked well together achieving our goal for completing this project. Our approach for accomplishing the desired outcomes was having constant clear communication and proper planning. In combination of our weekly supervisor and cohort meetings, we split our work into weekly sprints which is where we planned out the work on scrum boards. It was within these scrum boards and weekly planning that we could constantly re-evaluate our progress. We split small sub-tasks amongst each other at the beginning of each week. A large proportion of the main tasks required to complete this project was completed together where we both had valuable inputs into completing it. Each main task had multiple smaller tasks that we could split between each other as we work towards our goals. We created a common project conventions file with coding guidelines and templates to ensure as we work on tasks separately, our style of work reflected that of the group. These conventions can be found on our GitHub here.

Within our GitHub we would often create new branches dedicated to a specific task so that we could both be working on areas in our code at the same time. This allowed us to progress and identify errors really quickly. We created pull requests which were largely thoroughly reviewed by the other group member. We believe our GitHub shows our progression of work and dedication to finding and fixing bugs while developing our desired features. In total we completed over 800 commits. The following table shows a basic separation of work for particular tasks, however, it should be noted that we largely worked together on all tasks in order to accomplish our goals. The field of work we were exposed to during this project was a field that neither of us had any experience in prior to the commencement of this project, therefore, we often focused on the same task at the same time, and often time this worked out perfectly as one group member gets stuck and another finds a breakthrough. The scope of the project was also relatively narrow, thus there was not that many main tasks to accomplish, so we tended to work together on most of them.

Table 2 : Separation of work amongst group members

Task	**Assigned To**
Nextflow Basics	Both
Cluster Basics	Both
Workflow DSL2 conversion	Both
Static location aware version	Both
Dynamic location aware version 1	Tristan
Dynamic location aware version 2	Robin
Dynamic location aware version 3	Tristan
Unloaded Testing Full Workflow	Tristan
Unloaded Testing Different Sizes Workflow	Robin
Loaded Testing Full Workflow	Both
AWS Virtual Environment Creation	Robin
Open Day Poster	Both

The reflection of working as a group for this project is a very positive one. We worked well together and successfully achieved our goals. We motivated and kept each other accountable. The process of working together gave me proper exposure to how it would be like working within a team in the workplace. I want to properly acknowledge our supervisor, Dr Hazelhurst, for his guidance and help throughout the project. I also want to acknowledge my group partner, Tristan Lilford, for his dedicated work and cohesive teamwork throughout the project.

APPENDIX B - TENDERS

This appendix contains the three unmarked original tender documents submitted during the 'Tender' process prior to the commencement of the investigation. These documents are attached below.

School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN4002/4012: EIE Investigation (Capstone Project) 2022

Tender Bid for Project Number: 22P21

Project Title: App to scan pool water test strips
Group Number: 22G51

Project Overview:

This project will focus on using the camera of a mobile phone to correctly scan a pool water test strip, isolate the various squares on the test strip, identify the colours for the ph balance and chlorine levels and approximate the chemical ranges. The app should recommend a course of action (for example "Add more chlorine"). The data should be logged, and the app should be able to perform reliably in different light conditions.

This will be done by designing and implementing an image classification tool on water test strip images. Data sets will be used to train a machine learning tool. Algorithms will be used on the training sets to process information which will then be presented in the app. The data that is to be used is images of water test strips and therefore, no human participants are required. The main project has four main steps. (1) Create machine learning models from image data sets. (2) Image and result matching for a variety of categories. (3) Outcomes based on results from images. (4) Integrate into a mobile app.

Preliminary Budget & Resources:

Budget:
This project has no need for a budget as there is nothing that needs to be purchased to perform the steps outlined in the project overview. The project also has the benefit of being able to be completed remotely.

Resources:

- Python and Matlab software for Machine Learning algorithms.
- React Native or Unity (Investigation needs to be done on deciding which cross-platform framework to use to develop the app).
- Teachable Machine with Google software (can be used to test our image classification model).
- Water test strip images or water test strips to take images to train the model.

Weekly Milestones:

Week 1: Planning and setup
The project specifications need to be clearly defined and understood. This step will also include choosing an effective image classification tool and outlining some algorithms to be used.

Week 2: Data collection and processing
Once the image classification tool is set up, the water test strip images need to be extracted and processed into a format/training set for the algorithms. The quality of the information needs to be evaluated to identify any shortcoming or bias which may be present.

Week 3: Algorithms and Frontend app setup
A script will be created to process the information. This will be a general model which will work in conjunction with the various algorithms. The algorithms to be used will also be chosen. The app UI will be designed.

Week 4: Algorithm refinement and back-end application
The final solutions should be developed, all bugs need to have been removed and testing will begin. Backend of app will be developed.

Week 5: Processing, testing and evaluation.
Frontend development of the application. Conclusions will be made which will include an error analysis to determine the accuracy of the developed systems.

Risks and Mitigation:

1. **Time management:**
   The project members currently have not completed a machine learning project before however to mitigate this completely the members chose the course Software Engineering which has a focus on machine learning and AI. This will allow smooth and timeous operations of the project.
2. **Project setbacks:**
   Unseen requirements may appear which may create setbacks. To mitigate this, there will be effective communication with the supervisor, Dr. Estelle Trengove, to ensure that these are identified as soon as possible so that they may be dealt with.
3. **Bias:**
   Bias is a serious problem which can cause extreme inaccuracies. To mitigate this all forms of bias, need to be understood in the early stages of the project so that it can be avoided when choosing algorithms and selecting applicable data.
4. **Data quantity and quality:**
   Low data quantity and quality will lead to inaccuracy. To mitigate this an understanding of how much data is needed for accurate results will be obtained. If this is not possible a full breakdown as to why the results were skewed, will be given.

- Once complete, save as pdf before submitting – no other format will be accepted.
*- Any submission longer than **1 page** will be recorded as a non-submission and an SP warning will be issued.*

Ver 1.0

School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN4002/4012: EIE Investigation (Capstone Project) 2022

Tender Bid for Project Number: 22p64

Project Title: Toys are a child's best friend
Group Number: 22G51

Project Overview: *(give a brief outline of how you will approach the investigation.)*

The aim of this project is to design as system which would in theory be applied to children's toys to help kids in their early development. The system would accomplish this by teaching the child about its surroundings and emotions. To do this the system would need to not only identify objects and facial expression, but also detect speech patterns in order to have a basic conversation. To make these image classifications of objects and facial expressions, image processing needs to be done. To do this the data set of images needs to be processed, annotated and generic for ML image processing. Computer vision (CV) will then be used with these images to create a training set of data for the machine learning algorithms. With the system able to identify objects, it now needs to be able to detect speech. To do this, sampled sound data needs to be preprocessed into an effect training set before being read into a neural network to develop an ASR system. Once the system can convert speech to text, the system must then interpret that text. By making use of an NLP tool text simplification and information retrieval can be performed. A set of basic classifications such as greetings, questions and scanning can be done to allow for basic conversation

Preliminary Budget & Resources:

Budget:
This project has no need for a budget as there is nothing that needs to be purchased to perform the steps outlined in the project overview. The project also has the benefit of being able to be completed remotely.

Resources:
Python and matlab
Sample sounds
Image data

Weekly Milestones: *(give specific deliverables for each week.)*

1. **Week 1: Planning and setup**
   The project specifications need to be clearly defined and understood. Basic setup for the three different machine learning systems needs to be done.
2. **Week 2: Data collection and processing**
   All three different systems namely image processing, speech to text recognition and text interpretation need training data. To do this preprocessing of sample sounds, imaging and text needs to be underwent so that a model to operate the systems together can be designed.
3. **Week 3: Algorithms and setup**
   Each system will have an algorithm chosen to best process the training sets. A model will then be created to bridge the different systems to one another.
4. **Week 4: Algorithm refinement and application**
   The final solutions should be developed, all bugs need to have been removed and testing will begin.
5. **Week 5: Processing and evaluation.**
   Evaluation of system and shortcoming will be noted.

Risks and Mitigation:

1. **Time management:**
   This project has numerous deliverables, the time constraint does put pressure on the whole task. To mitigate this the students are currently doing a Machine learning and AI course to prepare themselves.
2. **Bias:**
   Bias is a serious problem which can cause extreme inaccuracies. To mitigate this all forms of bias, need to be understood in the early stages of the project so that it can be avoided when choosing algorithms and selecting applicable data.
3. **Data quantity and quality:**
   There are three different systems to be trained specifically sample sound needs to be adequate as different pronunciations of words can affect the training and classification processes. To mitigate this extensive and large data sets will be used in the training process.

- Once complete, save as pdf before submitting – no other format will be accepted.
*- Any submission longer than **1 page** will be recorded as a non-submission and an SP warning will be issued.*

School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN4002/4012: EIE Investigation (Capstone Project) 2022

Tender Bid for Project Number: 22p65

Project Title: Personality prediction using online social platforms.
Group Number: 22G51

Project Overview: *(give a brief outline of how you will approach the investigation.)*

The aim of this project is to predict an individual's personality type from their social media account. The personality type will fall into 1 of 16 types outlined by the Myer Briggs personality test. This personality type is based off of the individual introversion, intuition, perceiving and thinking capabilities. To make this prediction a Natural Language Processing (NLP) tool and various sentiment analysis techniques will be implemented on the social media posts. This will create a data set which will be analysed by multiple machine learning algorithms, such as the Naïve Bayes Algorithm, to make a classification of the individual's personality type. This will then be compared to the individual's actual personality type to determine the accuracy of the algorithm to make a correct prediction. This process will be done for multiple algorithms to determine the most effective machine learning algorithm for this scenario by weighing performance vs complexity.

Preliminary Budget & Resources:

Budget:
This project has no need for a budget as there is nothing that needs to be purchased to perform the steps outlined in the project overview. The project also has the benefit of being able to be completed remotely.

Resources:
Python and matlab
Social media posts/data (already acquired)

Weekly Milestones: *(give specific deliverables for each week.)*

1. **Week 1: Planning and setup**
   The project specifications need to be clearly defined and understood. This step will also include choosing an effective NPL tool and outlining some algorithms to be used.
2. **Week 2: Data collection and processing**
   Once the NPL tool is set up, the social media posts need to be automatically extracted and processed into a format/training set for the algorithms. The quality of the information needs to be evaluated to identify any shortcoming or bias which may be present.
3. **Week 3: Algorithms and setup**
   A script will be created to process the information. This will be a general model which will work in conjunction with the various algorithms. The algorithms to be used will also be chosen.
4. **Week 4: Algorithm refinement and application**
   The final solutions should be developed, all bugs need to have been removed and testing will begin.
5. **Week 5: Processing and evaluation.**
   Comparisons of different algorithms need to be conducted so that conclusions can be drawn. This will include an error analysis to determine the accuracy of the developed systems.

Risks and Mitigation:

1. **Time management:**
   The project members currently have not completed a machine learning project before however to mitigate this completely the members chose the course Software Engineering which has a focus on machine learning and AI. This will allow smooth and timeous operations of the project.
2. **Project setbacks:**
   Unseen requirements may appear which may create setbacks. To mitigate this, there will be effective communication with the supervisor, Dr. Yuval Genga, to ensure that these are identified as soon as possible so that they may be dealt with.
3. **Bias:**
   Bias is a serious problem which can cause extreme inaccuracies. To mitigate this all forms of bias, need to be understood in the early stages of the project so that it can be avoided when choosing algorithms and selecting applicable data.
4. **Data quantity and quality:**
   Low data quantity and quality will lead to inaccuracy. To mitigate this an understanding of how much data is needed for accurate results will be obtained. If this is not possible a full breakdown as to why the results were skewed, will be given.

- Once complete, save as pdf before submitting – no other format will be accepted.
*- Any submission longer than **1 page** will be recorded as a non-submission and an SP warning will be issued.*

Ver 1.0

APPENDIX C - PROJECT PLAN

This appendix contains the complete project plan created before the commencement of the project. In order to ensure the project is successfully completed within the time constraints, proper planning throughout is required. The project is completed over a period of 8 weeks. The first 6 weeks was based around the creation and testing of the library with the final 2 weeks based around concluding final tests, the Open Day and completing the report. The 6 weeks of creation and testing was split into weekly Sprints which planned out the weeks tasks. There are 2 scrum boards for each Sprint to indicate the start and end of the Sprint. This weekly planning is shown on our GitHub here. The original unmarked project plan is also contained in this appendix. The meeting minutes taking during the project planning phase are also attached below.

PROJECT PLAN FOR LOCATION AWARE SCIENTIFIC WORKFLOWS - NEXTFLOW LIBRARY IMPLEMENTATION AND ANALYSIS

Tristan Lilford (1843691) and Robin Jonker (1827572)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: A project plan for the analysis and implementation of a Nextflow library that will allow for location aware scientific workflows is to be developed. This library can be added to existing workflows in order to improve pipeline performance by choosing the optimised methods for execution. The increased performance will be achieved through the automatic identification and setting of the best executor based on the data, network constraints and processor constraints. The executor will therefore be based on locality of the principal data. This report details the processes that will be taken in order to achieve this. The created library will be tested with actual workflows and the computational efficiency of the library will be evaluated and reported.

Key words: Nextflow, Executor, AWS, Docker, Workflow

1. INTRODUCTION

This report details the project plan for the creation and testing of a Nextflow library that enables scientific workflows to be aware of their location. In order to achieve this, appropriate techniques need to be developed for creating methods based on locality of the principal data required. Modern big data scientific problems are solved using workflows [1]. A Workflow is a structure within which dependant and independent complex programs are run both sequentially and in parallel to compute a specific output. A set of tasks and/or data are passed from one program to another for action, according to a set of procedural rules [2]. This allows for large complex data sets to be analysed using distributed environments and high degrees of parallelism. Data may need to be sent to the computers where the code is to be executed, this can be heavily dependant on network resources and can effectively create a bottle neck lowering overall performance of the workflow. An alternative method would be to the send the code in containers (Docker) to where the data is to be executed, however this to includes its own implications. There are also various methods which can be used in order to execute workflows such as using clusters. The choice of these execution configurations is dependant on what data is to be processed, the network constraints and the processing constraints available. The creation of a library that is able to identify the most effective way to execute the workflow will allow for a considerable increase in performance as the code can be executed using the most optimal conditions with minimal idle times.

Nextflow is unique in the fact that the pipelines that are created are very portable. Without changing the code that is executed, the pipeline can be run on different systems, be it in the cloud or locally. This is because the workflows have two separate files, one relating to the pipeline scripts and another relating to the configuration settings of it. This allows one to alter the configuration file of the pipeline and effectively set different locations to process the data. This ability to change the executor of the workflow makes way for the design on a location aware scientific workflow.

This report is split into 4 main sections, namely the scope, milestones, methodology, and the work breakdown. The scope contains aspects such as the project outline, specifications, budgets and notable risks. The milestones and methodology sections refers to specific tasks and their respective time allocations and how each will be accomplished. The work breakdown will be a summary of the tasks and how the work will be assigned. Lastly this report will be clarified with a conclusion in order to summarise the process to be taken in order to develop a location aware scientific workflow.

2. SCOPE

2.1 Project Outline

It is required that scientific workflows can be location aware which will improve performance as containerised code can be transmitted to the data sets of the geographically distributed processing points instead of transmitting the data sets to the location where the code is present.

A library will be written that can act as a black box for the user. This library will identify the location of the data set at run time, by inspecting the metadata of the input of the pipeline. Based on this information as well other variables such as network constraints and processing constraints, the executor within the configuration file will be set to the most optimal choice. This is expected to often result in sending the code in containers to where the data presides in order to achieve better performance results. This is to be evaluated through extensive testing.

This library will be tested by either developing a new workflow or rather by modifying an existing workflow. These workflows will utilise the designed library to expectantly improve pipeline performance. An array of different parameters will be used to test whether this was achieved, but ultimately the execution time of the two different methods is the main factor for this test.

2.2 Project Specifications

Depending on the data used for the workflow, the code that needs to be executed may be submitted

to different computers in a cluster or even sent to the cloud for computing. In order to accomplish this and the project's goals, the following will need to be achieved:

1. Develop appropriate techniques to determine the optimal executor based on locality of the principal data required.
2. Write a Nextflow/Groovy library to do so.
3. Test the library by developing a new workflow or better modifying an existing workflow in Nextflow DSL 2 and,
4. Conduct experiments to test the computational efficiency of the new code.

2.3 Budget and Resources

No physical components are needed to be able to develop the tools for this project. There will be no fees associated with travel as both students have the devices and internet required to accomplish this project remotely. Therefore, a budget is not required. It is assumed that free services and/or student credentials will allow for free access to the specific cloud services (WITS cluster and AWS).

Resources that will be used within the project relate to different software programs and services that will be used. As both students use Windows operating systems, both students will require Oracle VM Virtual Box Manager where a Linux based virtual machine can be operated on. Within that system, Nextflow and its required prerequisites will be installed as the software for the project. Code is to created on a text editor of choice. Additional software that will be used for different aspects include Docker for containerisation and the Wits Core Research Cluster and/or Amazon Web Services for cloud services. To allow for collaboration with ease, version control will be applied using GitHub. Required software is largely open-source. This allows for a large range of flexibility and adaptability if additional resources are required as they can be easily attainable.

2.4 Risks and Mitigation

Due to the large learning curve needed for a new field that both students are entering for this project, additional time is allocated to this project to occur before the commencement of the project. As per supervisor recommendations, an approximate 3 hours of learning every week will be allocated for the 8 weeks prior to the commencement of the project. In relation to this learning curve, the project plan could have slight alterations made to it as more information is gathered relating to the project. There is a large uncertainty in the time and complexity of each task required and therefore this additional time will be used to mitigate this risk. In relation to this, an agile coding approach will be taken where plans and ideas can change rapidly as more information is gathered and more feedback is received from the supervisor.

Running large workflows that analyse large data sets can require large execution times. Nextflow has a caching system integrated that saves and stores progress therefore if errors occur within the execution of the pipeline, previous progress is not lost [3]. This built in feature in the software of our choice mitigates the risks of execution times altering our time required to accomplish tasks.

3. MILESTONES

Project Duration: 7 weeks (15 weeks including the schedule prior learning time) with 5 days unassigned to mitigate risks.

A full week is left unassigned before the final report is due to account for any unexpected risks or challenges. The first task, 'Prior Knowledge Required', will commence 8 weeks before the official commencement of the project.

3.1 Prior Knowledge Required

This task will commence before the start of the project. This is needed in order to gain the appropriate skills needed to accomplish the tasks to come.

Sub-tasks:

1. Learn Nextflow
2. Learn Linux
3. Learn Wits Cluster
4. Learn AWS
5. Learn Docker
6. Learn Singularity

- Start Date: 25 July 2022
- Due Date: 16 September 2022
- Allocated Time: 8 weeks (approximately 3 hours per week)

3.2 Initialization and Setup

Sub-tasks:

1. Configure Linux Virtual Machine (VM)
2. Download Required Software
3. Setup Agile Version Control (Git)
4. Import Basic Nextflow Model

- Start Date: 19 September 2022
- Due Date: 19 September 2022
- Allocated Time: 1 Day

3.3 Techniques for Execution Configuration (a)

Sub-tasks:

1. Input Data Analysis
2. Identify Network Constraints
3. Identify Need for Multiprocessing/Clustering
4. Identify Optimal Execution Location
5. Identify Optimal Executor

- Start Date: 20 September 2022
- Due Date: 26 September 2022
- Allocated Time: 5 Days

3.4 Nextflow Library Implementation (b)

Sub-tasks:

1. Code Nextflow Processes
2. Set Pipeline Parameters

- Start Date: 27 September 2022
- Due Date: 7 October 2022
- Allocated Time: 9 Days

3.5 Library Testing (c)

Sub-tasks:

1. Test Designed Library (b) on Simple Datasets
2. Develop/Modify New Workflow using Library on Complex Datasets

- Start Date: 10 October 2022
- Due Date: 18 October 2022
- Allocated Time: 7 Days

3.6 Experimental Evaluation (d)

Sub-tasks:

1. Test Computational Efficiency
2. Create execution report

- Start Date: 19 October 2022
- Due Date: 26 October 2022
- Allocated Time: 6 Days

3.7 Conclusion

Sub-tasks:

1. Project report
2. Open Day poster
3. Submit/Deliver a final presentation

- Start Date: 27 October 2022
- Due Date: 4 November 2022
- Allocated Time: 7 Days

4. METHODOLOGY

The project is to be developed using GitHub for version control. The project is also to be tackled using agile coding methods and as such will include the use of user story mapping, scrum boards, sprint meetings and iterative coding methods. The GitHub repository can be found here. Documentation covers the architectural design records, sprint meeting minutes, scrum boards and project conventions. The only relevant functionality currently present is the project conventions, where templates for code reviews, pull requests and sprint checklists can be found. It also includes a coding guides which outline the styling to be used in order to create readable and meaningful code. The development team will utilise trunk based development to ensure continuous working releases. In order to begin tackling the problem at hand a large amount of prior knowledge is needed.

4.1 Prior Knowledge Required

The entirety of the workflow is to be coded using Nextflow DSL 2. Nextflow scripting language is an extension of the Groovy programming language [4]. In order to do so, an in depth knowledge of Nextflow scripting, channels and processes needs to be understood. An understanding of Nextflow configuration files also needs to be acquired so that executors and containers can be used to create a location aware workflow. Nextflow is only compatible with POSIX compatible system such as Linux. With this being the case an understanding of Bash and Linux is also needed in order to effectively design and run the created workflows.

Furthermore, an understanding of the WITS Cluster, AWS (Amazon Web Services), Docker and Singularity needs to be achieved in conjunction with Nextflow. These systems and services allow for tasks such as containers and cluster processing. This will be extremely relevant in order to compare various executors efficiencies. This knowledge is required so that the project can be started and is to be acquired from 25 July 2022 until 16 September 2022. The rest of the methodology can be summarised by Figure 1 below.

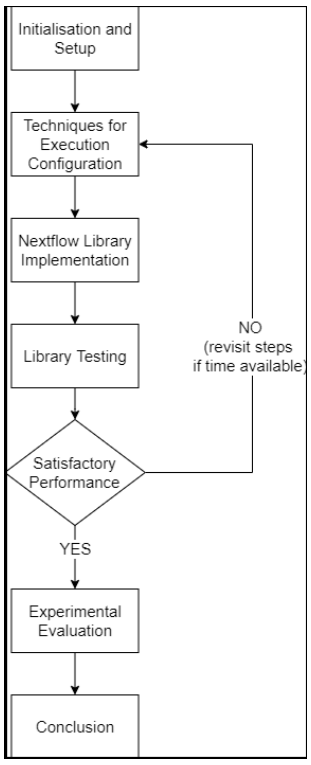


Figure 1: Flowchart of Project Methodology

4.2 Initialization and Setup

In order to effectively begin the project certain software needs to be installed and setup. Seeing as Nextflow needs to run on a POSIX compatible system it was chosen to be run on a Linux Machine. To do this a virtual machine is to be installed for a Ubuntu Operating System. This is to be done using Oracle Virtual Box. Nextflow requires Bash 3.2 or later and Java 11 or later. Therefore Bash, java and Nextflow must be installed, this installation process can be found here. The designed GitHub repository project backlog should also be populated to commence the start of sprint 1. Sprints are to run week to week with sprint meetings taking place at 1pm every Monday.

From this point basic Nextflow models should be imported in order to see different workflows and data inputs in order to better conceptualise how to determine techniques for execution configuration (a). An example would be H3Agwas which is a simple human GWAS analysis workflow [5] and can be found here. From this point models can be built using pseudo code to best determine where to execute workflow processes.

4.3 Techniques for Execution Configuration (a)

There are many aspects which need to be considered when deciding on where and how the data should be processed. There are 3 main areas which need to be studied in order to best determine this.

- The data and meta data
- Network constraints
- Processing constraints.

4.3.1 Data analysis The data that needs to be processed using the workflow needs to be analysed. If a large amount of data is is to be used it may place unneeded strain on the network to transfer it to a central processing machine. This would effectively bottle neck the workflow. If the data is small however this could be considered viable.

4.3.2 Network constraints If the network bandwidth is low, sending large amounts of data may not be viable and therefore the code should be sent to the data using containers. Also if there is a large demand for a required cloud service it may be more viable to send data to be processed on a more available machine. Also it may be recommended to process the data at the location it is stored.

4.3.3 Processing constraints The computer used to processed the data may need to have a high processing speed. If the data storage system has low processing speed it may take a large time to finish the workflow processes. Using clusters to take advantage parallelism may prove to be heavily advantageous.

All these aspects need to be taken into consideration when choosing an executor for the workflow.

4.4 Nextflow Library Implementation (b)

This step involves creating a Nextflow library based on the principles stated in Section 4.3. This would effectively be able to read input meta data, network details and processor values in order to decide the best executor for the workflow. These 3 aspects however are not to be weighted of the same importance. The best method in order to determine such weightings would be to use Machine Learning. However for the time constraints of this plan, values will be manually assumed and tuned to achieve the greatest successes.

4.5 Library Testing (c)

Once the library has been created it can be used within any workflow model. Testing will be to ensure the system is working as intended. This step will involve basic test runs with the created library and then more extensive tests imploring this library in conjunction with other workflows. The results are to determine if an acceptable outcome was achieved. This process would include the fine tuning of the weighting values described in Section 4.4.

4.6 Experimental Evaluation (d)

This process would include the utilisation of the in-built Tracing visualisation Nextflow ad hoc methods. These results can be used to compare time stamps in order to determine whether the added library to determine an optimal executor achieved its purpose. These comparisons are to be run over many instances in order to average performance differences. This step will effectively determine whether the library was successful in effectively creating a smart location aware scientific workflow.

Results should be acquired and outputted automatically to pdf. These are to added to the GitHub Repository in a summarised format.

4.7 Conclusion

This step includes the creation of the project report, the open day poster and the final presentation. This process is to cover all the steps which went well and those which fell short. Recommendations will be given and further detailed explanations will be given.

The entirety of the methodology is to be tackled using agile coding methods where scrum meetings will be held to determine what went right and what is needed to be improved. The task at hand is out of the scope of the developers and a more accurate understanding of the methodology and will be achieved after gaining an understanding of the prior knowledge required.

5. WORK BREAKDOWN

The tasks described in Section 4 are further broken down in the work breakdown structure (WBS) provided in Appendix A, Figure 2. Tasks are split into further sub tasks in a non-chronological order. A gantt chart is provided in Appendix A, Figure 3. This illustrates task time allocations and distinguishes which tasks should not be delayed. With agile coding, testing is a continuous part of development and therefore the point of the testing task is rather more for fine tuning models than bug identification and fixing. The task is seen to be completed by the 4th of November. Work designation is illustrated in Table 1 below.

Table 1: Work Designation

Task	Assignee
Initialization and Setup	Tristan & Robin
Input Data analysis	Robin
Network constraint identification	Tristan
Nextflow library implementation	Tristan & Robin
Configuration file parameters	Robin
Library testing and optimisation	Tristan & Robin
Modify an existing workflow	Tristan
Test computational efficiencies	Tristan & Robin
Create execution report	Tristan & Robin

6. CONCLUSION

This report effectively documented the plan and design process to be underwent in order to develop a location aware workflow. It provided the necessary steps, working practices, time management and work designations required to complete the task within the set time. Risks were noted and possible mitigation for them are put in place, notable the addition of an entire week to ensure the project is completed in time. The required prior learning task which is set to occur 8 weeks before the commencement of the project should prepare both students fully, therefore the tasks at hand could be achieved. If the milestones are followed in accordance, the plan will allow for the developers to create an effective location aware scientific workflow.

7. REFERENCES

[1] Khan, Samiya Shakil, Kashish Alam, Mansaf. (2017). Big Data Scientific Workflows in the Cloud: Challenges and Future Prospects.

[2] Owen-Hill, A. (2019). How to Set Up a Strong, Streamlined Software Workflow. [online] RoboDK blog. Available at: https://robodk.com/blog/streamlined-software-workfl
ow/: :text= [Accessed 24 Jul. 2022].

[3] www.nextflow.io. (n.d.). Demystifying Nextflow resume — Nextflow. [online] Available at: https://www.nextflow.io/blog/2019/demystifying-nextflow-resume.html [Accessed 25 Jul. 2022].

[4] www.nextflow.io. (n.d.). Get started — Nextflow 22.04.0 documentation. [online] Available at: https://www.nextflow.io/docs/latest/getstarted.html [Accessed 24 Jul. 2022].

[5] S. Baichoo et al., "Developing reproducible bioinformatics analysis workflows for heterogeneous computing environments to support African genomics," BMC Bioinformatics, vol. 19, no. 1, Nov. 2018, doi: 10.1186/s12859-018-2446-1.

[6] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," Nature Biotechnology, vol. 35, no. 4, pp. 316–319, Apr. 2017, doi: 10.1038/nbt.3820.

[7] T. Reiter et al., "Streamlining data-intensive biology with workflow systems," GigaScience, vol. 10, no. 1, Jan. 2021, doi: 10.1093/gigascience/giaa140.

[8] J.-T. Brandenburg et al., "H3AGWAS : A portable workflow for Genome Wide Association Studies," May 2022, doi: 10.1101/2022.05.02.490206.

Appendices

A Appendix

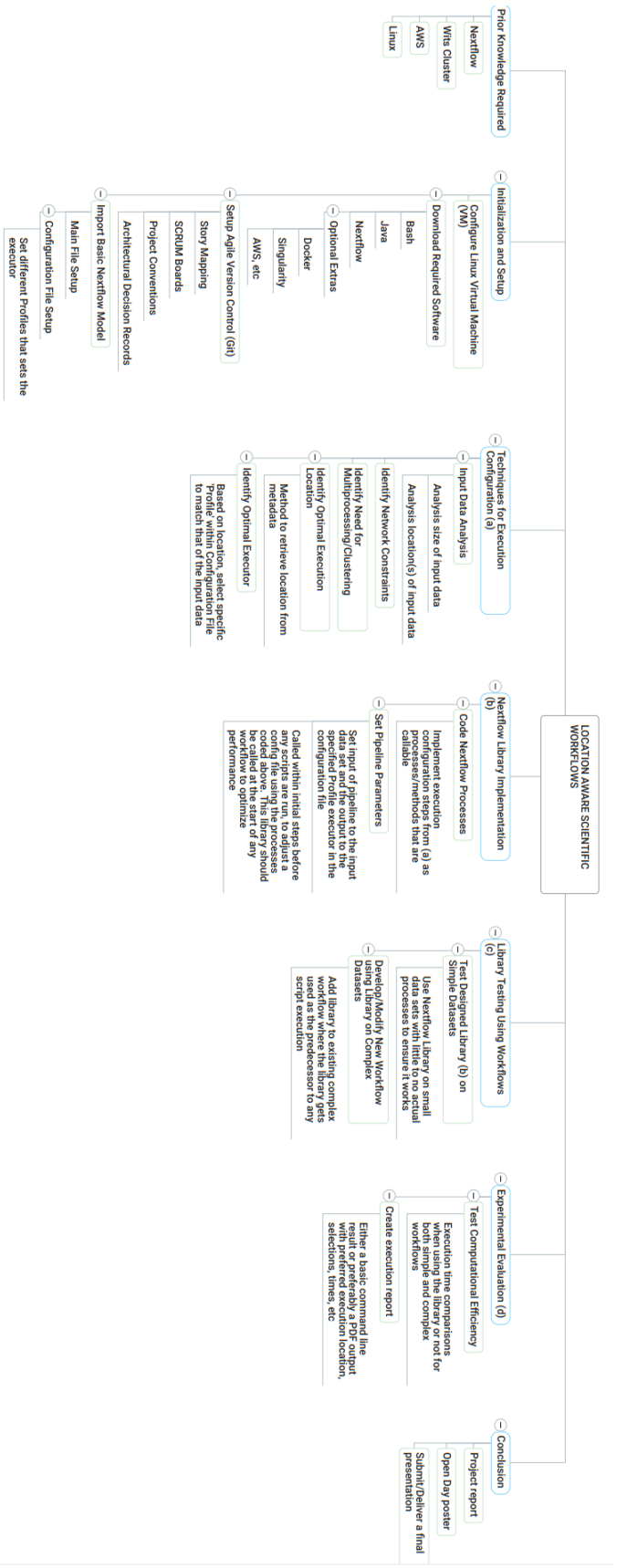
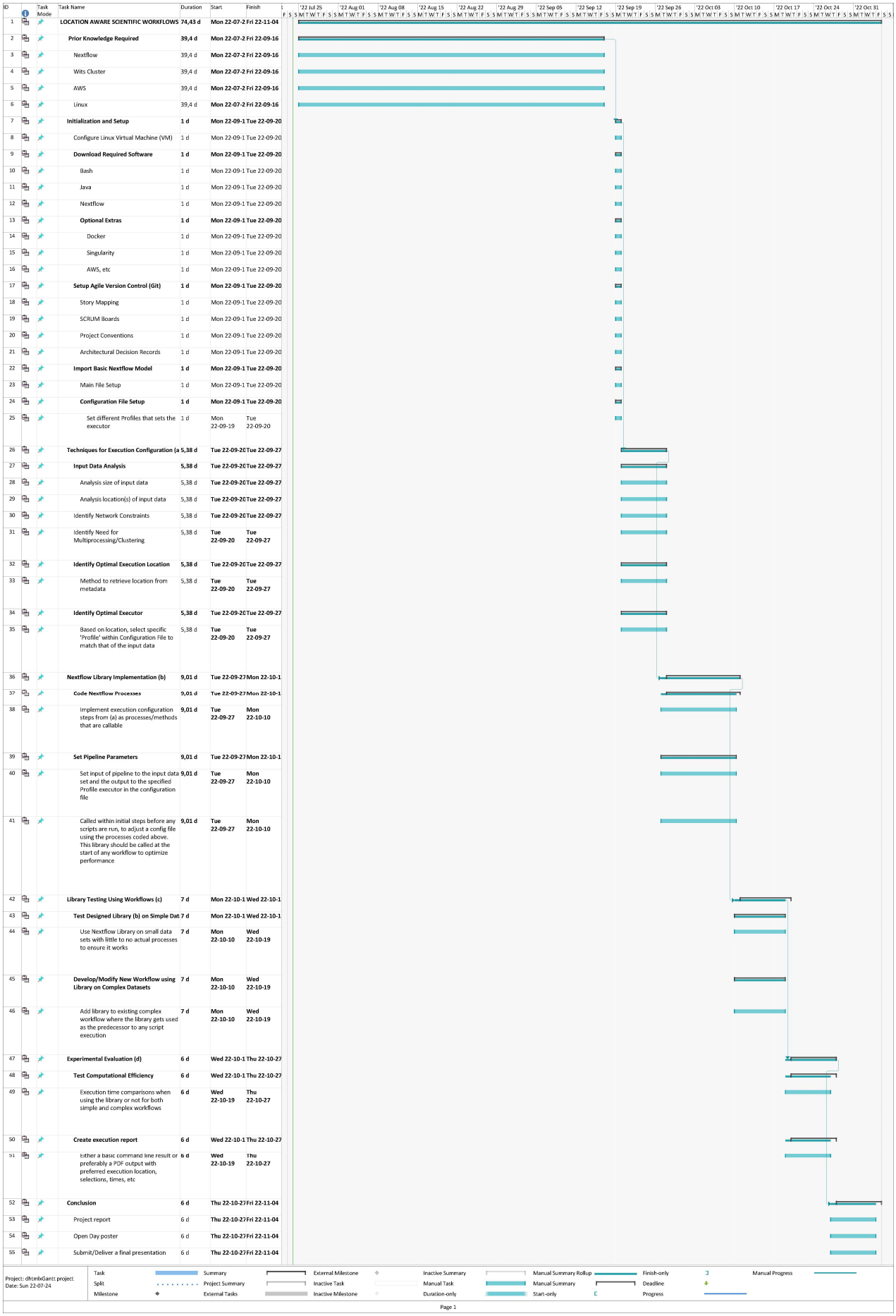


Figure 2: Detailed Work Breakdown Structure

B GANTT CHART



ID	Task Mode	Task Name	Duration	Start	Finish
1		**LOCATION AWARE SCIENTIFIC WORKFLOWS**	**74,43 d**	**Mon 22-07-2**	**Fri 22-11-04**
2		**Prior Knowledge Required**	**39,4 d**	**Mon 22-07-2**	**Fri 22-09-16**
3		Nextflow	39,4 d	**Mon 22-07-2**	**Fri 22-09-16**
4		Wits Cluster	39,4 d	**Mon 22-07-2**	**Fri 22-09-16**
5		AWS	39,4 d	**Mon 22-07-2**	**Fri 22-09-16**
6		Linux	39,4 d	**Mon 22-07-2**	**Fri 22-09-16**
7		**Initialization and Setup**	**1 d**	**Mon 22-09-1**	**Tue 22-09-20**
8		Configure Linux Virtual Machine (VM)	1 d	Mon 22-09-1	Tue 22-09-20
9		**Download Required Software**	**1 d**	**Mon 22-09-1**	**Tue 22-09-20**
10		Bash	1 d	Mon 22-09-1	Tue 22-09-20
11		Java	1 d	Mon 22-09-1	Tue 22-09-20
12		Nextflow	1 d	Mon 22-09-1	Tue 22-09-20
13		**Optional Extras**	**1 d**	**Mon 22-09-1**	**Tue 22-09-20**
14		Docker	1 d	Mon 22-09-1	Tue 22-09-20
15		Singularity	1 d	Mon 22-09-1	Tue 22-09-20
16		AWS, etc	1 d	Mon 22-09-1	Tue 22-09-20
17		**Setup Agile Version Control (Git)**	**1 d**	**Mon 22-09-1**	**Tue 22-09-20**
18		Story Mapping	1 d	Mon 22-09-1	Tue 22-09-20
19		SCRUM Boards	1 d	Mon 22-09-1	Tue 22-09-20
20		Project Conventions	1 d	Mon 22-09-1	Tue 22-09-20
21		Architectural Decision Records	1 d	Mon 22-09-1	Tue 22-09-20
22		**Import Basic Nextflow Model**	**1 d**	**Mon 22-09-1**	**Tue 22-09-20**
23		Main File Setup	1 d	Mon 22-09-1	Tue 22-09-20
24		**Configuration File Setup**	**1 d**	**Mon 22-09-1**	**Tue 22-09-20**
25		Set different Profiles that sets the executor	1 d	Mon 22-09-19	Tue 22-09-20
26		**Techniques for Execution Configuration (a**	**5,38 d**	**Tue 22-09-20**	**Tue 22-09-27**
27		**Input Data Analysis**	**5,38 d**	**Tue 22-09-20**	**Tue 22-09-27**
28		Analysis size of input data	5,38 d	**Tue 22-09-20**	**Tue 22-09-27**
29		Analysis location(s) of input data	5,38 d	**Tue 22-09-20**	**Tue 22-09-27**
30		Identify Network Constraints	5,38 d	**Tue 22-09-20**	**Tue 22-09-27**
31		Identify Need for Multiprocessing/Clustering	5,38 d	**Tue 22-09-20**	**Tue 22-09-27**
32		**Identify Optimal Execution Location**	**5,38 d**	**Tue 22-09-20**	**Tue 22-09-27**
33		Method to retrieve location from metadata	5,38 d	**Tue 22-09-20**	**Tue 22-09-27**
34		**Identify Optimal Executor**	**5,38 d**	**Tue 22-09-20**	**Tue 22-09-27**
35		Based on location, select specific 'Profile' within Configuration File to match that of the input data	5,38 d	**Tue 22-09-20**	**Tue 22-09-27**
36		**Nextflow Library Implementation (b)**	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-1**
37		**Code Nextflow Processes**	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-1**
38		Implement execution configuration steps from (a) as processes/methods that are callable	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-10**
39		**Set Pipeline Parameters**	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-1**
40		Set input of pipeline to the input data set and the output to the specified Profile executor in the configuration file	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-10**
41		Called within initial steps before any scripts are run, to adjust a config file using the processes coded above. This library should be called at the start of any workflow to optimize performance	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-10**
42		**Library Testing Using Workflows (c)**	**7 d**	**Mon 22-10-1**	**Wed 22-10-1**
43		**Test Designed Library (b) on Simple Dat**	**7 d**	**Mon 22-10-1**	**Wed 22-10-1**
44		Use Nextflow Library on small data sets with little to no actual processes to ensure it works	**7 d**	**Mon 22-10-10**	**Wed 22-10-19**
45		**Develop/Modify New Workflow using Library on Complex Datasets**	**7 d**	**Mon 22-10-10**	**Wed 22-10-19**
46		Add library to existing complex workflow where the library gets used as the predecessor to any script execution	**7 d**	**Mon 22-10-10**	**Wed 22-10-19**
47		**Experimental Evaluation (d)**	**6 d**	**Wed 22-10-1**	**Thu 22-10-27**
48		**Test Computational Efficiency**	**6 d**	**Wed 22-10-1**	**Thu 22-10-27**
49		Execution time comparisons when using the library or not for both simple and complex workflows	**6 d**	**Wed 22-10-19**	**Thu 22-10-27**
50		**Create execution report**	**6 d**	**Wed 22-10-1**	**Thu 22-10-27**
51		Either a basic command line result or preferably a PDF output with preferred execution location, selections, times, etc	**6 d**	**Wed 22-10-19**	**Thu 22-10-27**
52		**Conclusion**	**6 d**	**Thu 22-10-27**	**Fri 22-11-04**
53		Project report	6 d	**Thu 22-10-27**	**Fri 22-11-04**
54		Open Day poster	6 d	**Thu 22-10-27**	**Fri 22-11-04**
55		Submit/Deliver a final presentation	6 d	**Thu 22-10-27**	**Fri 22-11-04**

EIE LAB.

- Meeting 1: 19 July 2022 22GS1 LocAwareComp.

Topic: Project Plan Review.

Location: Zoom

Time: 8 am

Attendees: ~~Prof Scott H, T, R~~ Dr Hazelhurst, Tristan, Robin

Agenda items:

steps needed? Use thruster, instead of own comp.

Groovy → Java → Dev on it. no more 200 liner.

WC vs AWS →

↑ run here ; data location dependant

→ What do we do then.

↳ what we do before → conf nextflow, cluster, cloud

week level? Tasks

Video: 50 min vid

Nextflow

- set executors

Local → AWS → WCS → library that determines where.

- Summary → a guide on what to expect and basic instructions on the process of learning the needed language.

EIE LAB

Meeting 2 : 8 September 2022 22CS51 LocAwareComp

Topic : Feedback of plan

Location : zoom

Time : 1 pm

Attendees : ~~Prof Scott H, T, R~~ Dr Hazelhurst, Tristan, Robin.

Agenda : monday meeting.

→ Suggestions :

- Get on cluster, get comfortable
- Write basic nextflow scripts in
- Groovy
- Instruction cluster use.

- Summary : set weekly meetings, next steps to gain access to the wits cluster and its use. Send access public key to auth.

APPENDIX D - WEEKLY SUPERVISOR MEETINGS

This appendix contains the minutes taking during our weekly meetings with our supervisor, Dr Hazelhurst. All 6 meetings minutes are attached below.

EIE LAB

Weekly Supervisor Meeting 1 : 19 September 22451 LocAware Comp.

Topic : Meeting.

Location : Zoom

Time : 8:40 am

Attendees : Dr Hazelhurst, Tristan, Robin

Agenda : Focus for this week :

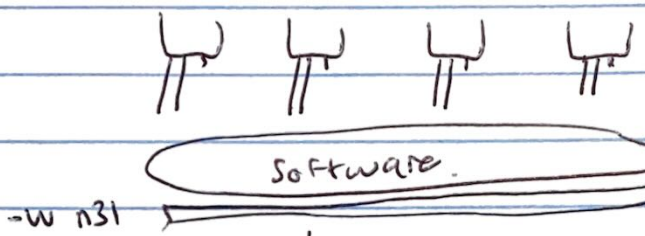
→ today : comfortable with cluster.

run basic model on it

- slurm
- simple nextflow script that use slurm
- look at gluster

↳ 20 machines.

- e.g.



- provide common interface

gluster external disc c ls, look one file system.

↳ slow cause 60 discs.

- gluster → every file replicated twice.

↳ disad. expensive, slow, space

↳ adv. tolerate Failure.

↳ read using gluster from multiple machine

↳ better performance.

→ when job submit? using slurm? work out where file stored? which machine physical stored.

- exeute job on that machine instead of general.
- hide high level abstraction cost to better performance
- still when job queue, has directive, tell it which node to run on. Add to job. Might use general instead of specific job if job too busy.

EIE LAB

Weekly Supervisor Meeting 2: 28 September 22GSI LocAwareComp

Topic: Meeting

Location: Zoom

Time: 8:30 am

Attendees: Dr Hazelhurst, Tristan. Robin

Agenda: • Dynamic directive
↳ clusterOptions.
• DiskC access
• Every process called?

DiskC → not mounted to head node.
↳ use slurm, log on to worker node
↳ access there.

Array → nodeSug → clusteroptions → use that

• call routine to check, to decide

Should add clusteroptions within each process?
↳ amend code? which file should
BAM 50 gb? BAI 50 kb? Different nodes.

Discuss in report possibly.

• Call to log into specific node. ssh into it.
↳ node name, srun to ask for node based on demand.
↳ instead of select "ssh n10" e.g.

Next meeting 8.30 monday.

EIE LAB

Weekly Supervisor Meeting 3: 3 October 22GS1 LocAwareComp.

Topic: meeting

Location: Zoom

Time: 8:30 am

Attendees: Dr Hazelhurst, Tristan, Robin

Agenda: Update of what we have done

↳ Hello world version, static version

↳ input multiple files

- Ask for workflow to integrate.
- Make slurm gluster dynamic
- Generate testing report of a job.

End of week goal: Dynamic done

next: testing done

then: Aws implementation.

, workflow will be given to integrate with

↳ cannot use real data due to ethics clearance, no time for that.

. will research to find something for us.

EIE LAB

Weekly Supervisor Meeting 4: 10 October 22GS1 LocAwareComp.

Topic : Meeting

Location : Zoom

Time : 8:30 am

Attendees : Dr Hazelhurst, Tristan, Robin.

Agenda : · Dynamic : use subscribe to trigger function?
convert to bash so can use function?

· weighing : Aggression set by file size?
→ what size? Testing? · Consider leaving out. or testing for ranger

· Find out about workflows we can use. → will find.

· Running on specific nodes, we set clusterOptions.
↳ always on worker nodes
↳ does commands only trigger head node? Use squeue.
↳ in script use (hostname) dot command out file to show which proces it was.

· Next meeting in person on wednesday.

→ clusterOptions uses name, not file object

clusterOptions > staging working
↓ ↓
File not visible File visible

pre-allocate clusterOptions

modify co. instead hash map look up
↳ more into name.

· 8am wednesday.

EIE LAB

Weekly Supervisor Meeting 5: 19 October 22G51 LocAwareComp.

Topic: Meeting

Location: zoom

Time: 8 am

Attendees: Dr Hazelhurst, Tristan, Robin

Agenda:
- Dynamic aggression. AWS sets network performance speed, how to find that out or solely set through testing.
- Dynamic is ready for testing.
- Research on AWS
- AWS CLI commands.
- Our workflow, on our gluster.
  1: shotgun. sets of sequences_compressed fastqueue files.
  2: scott-test: simple workflow.
  → simple workflow, can be made more complex.
  ↳ uses QC on files, relatively expensive.
  - more complex later on, max forks...
  → use as much of cluster of possible
  ↳ not node reserved. → run test over weekend.
  ↳ best effect with reserved nodes, but best performance on general.
  only → n-29 to n-45
  n 2 8 10
  ↳ directly on gluster DiscC, 2 nodes
  ↳ physical discs mounted on them.
  - idea, import process, twice, explore hybrid workloads
- AWS is high-risk so likely fail, worth while to try.
- Document what we doing to show

use these 2 for test.

EIE LAB

Weekly Supervisor Meeting 6 : 26 October 22G51 LocAwareComp.

Topic : Meeting

Location: Zoom

Time : 11:30 am

Attendees : Dr Hazelhurst, Tristan, Robin

Agenda :
- Docker on wits cluster → how to use it.
  - ↳ execute on Aws batch → no nodes? Only region not nodes
- General aws tips, how to? → data local Nxtflow → S3 data. ect
- what does he expect in the poster
  - ↳ code snippet
  - ↳ explain purpose
  - ↳ testing
  - ↳ nice graphic about gluster file system
  - ↳ using gluster vs not.

5pm
run workload for us to test
- Close to having final version dynamic
- complete testing over next two nights
- → submit on 7th 6 pages → telling a story no surprises all clear and logically. assumptions for external examiners her concerned.
- → use both workflows for testing
  - ↳ two sets of tests
  - run at different times day vs night.
    - ↳ busy vs quite
  - static: pay overhead of network network
- Dr will run extensive workflow and then ours.
  - ↳ will run for us, sensitive data
- → Docker not on cluster → security
  - ↳ singularity / Apptainer
  - his example → fastqc → containers, dockerHub. reg.
- local → upload data to instance
  - nextflow responsible, running locally.
- where data stored? Not likely.
- show experiments. other approaches for Aws.
  - using DSL 2
  - use labeling.
- what cloud? what message objective

② Features of DSL2 to allow more dynamic, import same process twice instead of text copy. → modules (use diff params)

local vs cloud vs mix (test)

executor slurm vs aws

① process, copy statically. one uses aws batch one uses locally.
- channel filter. split channel, local to one process S3 to other process.

test. if could dynamic process → see performance bonus

how to add dynamic scheduler.

use groovy

APPENDIX E - WEEKLY COHORT MEETINGS

This appendix contains the minutes taking during our weekly cohort meetings. We were apart of cohort 12 which contains groups 22G15, 22G18, 22G36, 22G48, 22G51, 22G67, 22G74, 22G75, and 22G90. The supervisors for our cohort meetings were Dr Hazelhurst, Dr Trengove, and Dr Bekker. All 6 meetings minutes are attached below.

Cohort 12 1st Meeting 23/09: Group 18

Chair: Raphi Druion , Minutes: Ben Palay

Minutes:

Prof Estelle: Introduction as to how these meetings will run. Their purpose is to provide aims for the next week, and say whether a group achieved their goals, what the obstacles were that week etc .

Devlan: Network firewall visualisation. Develop program in, assist visualisation for network admin to see changes being made. Done research into a few things, mostly optimising front end. looked into libraries for similar applications, also looked at wits cluster usage.

James : mapping rainbow nation, categorise twitter users into groups and map using sociogram, done by analysing various factors in twitter usage. So far quantified edges, running into performance issues. Done research into the twitter api area. How to get that data and filter it meaningfully.

Joe : track protests using twitter data and api, pull tweets based on hashtags, content etc. Do content and sentiment analysis, data process, plot on SA map. Check level of protests, violence etc. Looked into research at twitter api, data sources for where to find protests. Started research into sentiment analysis.

Kevin: ML and satellite imagery and pattern recognition to identify former lost cities by stones and archaeological remnants to track old settlements. So far just familiarising with tensorflow, research, trying to feed in circular images in order to train model. Built basic classification of images. Set up git etc

Ishmael and Nkosingiphile: develop genotype, cluster red and green files into groups. Aim is to classify genotype. DNA sequencing program. Started with investigation into different genotype models, working on cluster, getting used to it. Studying Rust ways to use read and write files.

Rael and Gia: Implementing low cost sensors in under developed areas, micro based device with air quality sensors to measure air quality. Started with basic air quality monitor. Stored info and time taken on SD card.

Mohammed: Water test strips: app to scan test strips, tell users what optimal chemical composition to add by using ML. Looked at object edge colour detection on the test strips, learning about app development.

Raphi and Ben: Matching a user's description of a face to a labelled image. We started with feature analysis, extracting various features such as eye colour and mouth size.

Tristan: location aware scientific workflows, so far familiarising with cluster, progress with Slurm and Groovy, trying to analyse data and best ways of queuing.

Prof Estelle: The report is what is marked, so even if it doesn't work the way you want, it's important to know why it didn't work, be critical, what could you have done differently etc. Show considerations and weighing of different solutions. Also you should locate project and its context in

current literature. Use wits database. You aren't expected to create new knowledge, but rather how can yours be different or how it works in a particular context.

Prof Scott: Document what you're doing and the thought process. Justify decisions that were made. Do it as you make the decision.

Prof Martin: Start strong-> finish strong. Also document failures.

Cohort 12 2nd Meeting 30/09: Group 17

Chair: Kevin Naidoo, Minutes: Johann

Apologies: Professor Scott Hazlehurst, Ahmed Ibrahim

Minutes:

Ishamel and Nkosingiphile: Managed to locate the given data on the cluster. The data is sorted into different intensities (red and green) across multiple folders. The main investigation has begun to convert data into vectors using a system that doesn't need to know the size of the data.

Raphi and Ben: Facial recognition – existing facial recognition models with will used and tweaked to fit the scope of the project. Creating a new model would require too many resources to be viable. The prebuilt model should be able to successfully identify gender, race, eye colour, and hair. To identify features on a face 68 points are used for detection, which unfortunately does not include hair. Further investigation is being done to improve racial and hairstyle recognition. The error rate is still relatively high and is caused by additional features on the photos that aren't specifically trained for, e.g. beards or very diverse skin tones. Investigating RGB – HEX conversion.

Prof. Bekker: Remember to document adversarial events that go wrong. Maybe consider focusing mainly of eyes, e.g., compensate for mask wearers. Consider Dall-E, Mid Journey and Stable Diffusion as interesting ML resources.

Rael and Gia: The air quality sensor device cannot be powered by the main power supply. Currently testing how long the device can run using powerbanks. Ideally the device should be powered for 2 weeks to a month on a single powerbank. Unfortunately, an Arduino cannot be used as power supply. IOT: A suggestion was made to use Adafruit and a webhook instead of polling. The data captured by the sensor is also stored on an SD card. The device will measure the humidity in an area. The humidity can influence the air quality readings.

Joe and Hraklas: The ML model to track protests is progressing. Currently the main sources of data are the ACBD(?) protest websites and twitter. Twitter is used to identify date and location of tweets about protests in SA. Managed to pinpoint IP addresses within a bounding box that could localize an area with a protest. IF the location is too accurate, the ethical consideration should be considered.

Tristan: Using some of Prof Hazlehurst's developed workflows with additional tweaking and testing identifying the allocated nodes on the cluster that receive jobs can be identified. For the moment, the process is static, and the data created for testing. The following week will be used to make the process dynamic.

Devlan and Chavi: The firewall visualization is currently explored using FireViz. Blocked requests are shown in red, passing requests are red. For now, it works on a single machine/IP and created a graph to visualize the firewall. The aim is to implement on a complete network. The ruleset for the cluster is needed, but Prof Hazlehurst is withholding the information for the time being.

James and Johann: Capturing the data from twitter is proving a lot more complex than initially anticipated. Only 1% of tweets contain the geographical data. However, the user's bio might contain a

location tag and will be explored. The visualization of the sociogram is complex, because titter users have large amount followers and an when going even as deep as 3 layers, the dataset becomes large and complex. A step-by-step approach is implemented.

Mohamed and Thabo: Finding datasets for pool water test strips is proving trickier than anticipated. For the MVP, we are building a test strip reader that identifies the colors on a plain background. Android Studio is used to create the basic application. Currently the linking of the app to an image processing technique is being explored. The position of the different color tests, e.g., chlorine level, for now will be static.

Kevin: Exploring pattern recognition using TensorFlow. Managed to create a CNN that can identify images of cats and dogs. Further patterns will be used for identification to steer the CNN in the direction of archeological pattern recognition.

General:

Reminder that the weekly cohort meetings are compulsory and that both members of each group must attend. Additionally, the meeting starts at 11:00 exactly and late-comers will be reprimanded.

Cohort 12 3rd Meeting Minutes

Group: 15, **Chair**: Rael Ware, **Minutes**: Gia Croock

Mohammed Haffejee and Thabo Tshabalala- App to scan pool water test strips: Developed an application (the camera part). Struggling with detecting the edges. The algorithm used was giving gaps. Managed to use a system called dilation which fills the gaps. Main aim to use unsupervised learning technique to classify colours. Trying to find other algorithms for colours. Using the K mean clustering. Need to consider colour spaces. Supervised requires a lot of time to create data set therefore decided to use unsupervised. Idea to consider from Dr Bekker is to make the white next to the test strip a feature so that the algorithm uses this as a feature. The supervised learning does actually do this, it looks at the strip as a whole (the block and the bit next to it). The algorithm extracts all the colours. Questions to consider: What do you use as your baseline? Are you weighting your white? Can you also use the white to measure the lighting conditions and if you know its not white then maybe you need to adjust all the colours.

Benjamin Palay and Raphi Druion- Facial recognition: Developing a static web app in angular to get a user interface. Managed to deploy. Managed to filter faces based on ethnicity. Carrying on with the facial recognition. Managed to get algorithm that reads in everything about colour. Grouped 800 colours into 5 different colour names. An issue they are facing is that they are getting eyes that are meant to be blue to be silver. Don't know if they going to use prebuild models. Next week they going to try train their own model for the limited scope they created. The problem is the pretrained models are too specific. They thought of a continuous scale, but the problem is they must classify that scale. Suggestion: just let the user choose a colour visually. Only want the most common eye colours so that all images can be filtered. Questions to consider: Can you use the reduction already made to group eye colour and then refine search. There are lots of manual classification. Potentially can run averages.

Ahmed Ibrahim and Kevin Naidoo - The lost cities of South Africa: Preparing the pipeline for the model. The satellite images for training and testing are missing. Goal for next week is to obtain these images and prepare it to see what kind of results they are getting. Contacted the computer science student to get data.

Johann Gouws and James Allsop - Mapping the Rainbow (twitter): Not a very productive week. Managed to generate data sets needed for the graphs. Managed to visualise the graphs on HTML. Next week they are moving on to cleaning up the data.

Joseph Baggott and Hraklis Papageorgiou- The world's protest capital: Worked on trying to link the process data. They have their two data sets and are on the verge of linking them. Once this is done, they can be automated. Had problems with the locations but they solved it. Twitter doesn't mention how they get the location. Content sentiment analyses as a general concept will be worked on next week. Mainly considering English and then may expand.

Devlan Mckenzie and Muchaveleli Manjat- Network firewall management and visualisation: They worked on visualising the firewalls. Saved the firewall rules into a text file. Trying to make sense of the different tags. Managed to quantify the firewall rules. Looking into how to run the app on the cluster. If they classify their app as containers, they can be classified. They are using C++ which is not compatible with Doca. Manually identifying the tags. Made a simpler set of rules that read from file and assign IP data. Goal for next week is to get the visualisation working with some basic data.

Nkosingiphile Ndabandaba and Ishmael Sithole- Fast genotype calling: Started by programming a code that will extract data. Instead of specifying the path that the data is extracted from the user should specify the path. Biggest challenge was getting logged on to the cluster, but they managed to get that working. Looking at how to copy the local work to the cluster. Next step is to compile the code. Look at using git to copy to the cluster. Use slurm to run your stuff you can see the progress. Goal is to get the cluster working so that they can test to see if what they have created is able to extract the correct stuff.

Robin Jonker and Tristan Lilford- Location aware scientific workflows: Last week the system was working statically. This week's challenge was to get it working dynamically. It is working but not as they want it to. Started getting the Execution reports done. Getting comparable testing environments is not ideal because the cluster is being used by many groups. Using slurm scheduler for running the jobs. Will look to use AWS if they have time.

Rael Ware and Gia Croock- Cheap Air Quality Reporting Station: The device needs to be fully completed by the end of this week as the device is going to be evaluated and tested using the air quality reference monitors at SBIMB next week. Still waiting on the components necessary to finish implementing the GSM module. Spent the week preparing the device for testing this included 3d printing the devices housing, soldering the components onto the Vero board, and adjusting the code to ensure the results obtained from the device are comparable to the results obtained from the reference monitor. Spent time refining the evaluation and testing procedure. Reducing the power consumption of the device is still providing a significant challenge. While the device is being tested next week, they will develop our own API at the moment they are using things speak.

General notes:

It is important to have a GitHub repo and make sure you are doing things in a proper engineering way and not just doing it at the last minute.

Next week Mohammed and Thabo are chairing the meeting and taking minutes.

Next week there will be a projector for demonstration.

Cohort 12: 4th Meeting Minutes

14th of October 2022

Group: 48, Chair: Nkosingiphile Ndabandaba, Minutes: Ishmael Sithole

Apologies: Joseph Baggott, Robin Jonker

Benjamin Palay and Raphi Druion (Facial recognition)

Checked the hair color and If someone is bald. Analyzed all the features and have integrated and it's all good. The object is to display a face and improve the image shown.

Haffejee Mohammed and Thabo Tshabalala (App to scan pool water test strips)

Low-brightness and contrast of image resultant from images taken in normal light conditions with a smartphone which led to false detection of edges for color patches and a lot of noise in the image. The solution is the use of image enhancement techniques to increase the contrast and brightness of the image. This enable for more accurate color detection of the patches and test strip detection.

Objective next week:

- Create an algorithm for comparing reference chart colours from the patches and provide a results for each component.
- Compare colour spaces (LAB, HSV and HLS) that are close to the human perception to improve accuracy of results.

Joseph Baggott and Hraklis Papageorgiou (The world's protest capital)

Had an issue with total API. It takes 3 hours to get 100 subscribers. Its working but we will have to make some changes. Working on a way to display the data.

Muchaveleli Manjate and Devlan Mckenzie (Network firewall management and visualization)

We swapped to python and made a data frame for the ruleset. We met with the prof and got advice on binary decision diagrams and looked at tulip and control dd packages to make the bdd and this week we want to try get the bdds up and running

Started with the expression. Boolean expression work fine, the problem is the IP address. The idea is to convert each rule into a Boolean expression

James Allsop and Johann Gouws (Mapping the Rainbow)

Locating the images, look for more data to test the model. Qgis which open source and stoichiometry. Tested the model with picture. Next week integrate.

Robin Jonker and Tristan Lilford (Location aware scientific workflows)

Got dynamic node selection working as we wanted. Can be integrated into any workflow that uses a slurm scheduler as its choice of execution. Did research into executing on AWS.

Rael Ware and Gia Croock (Cheap air Quality Reporting Station)

Finished the angular app for the user interface where a user can iterate through images and tweak the features. They can also submit a rating of success at the end which we can see in a real-time database.

The plan for next week is to build an ML model to try and predict hair length and color and integrate with the app.

Ahmed Ibrahim and Kevin Naidoo (The lost cities of South Africa)

Implemented an SSD network, right now they have some pictures for the dataset. The machine learning model is able to label circular stone images. Need more images in their dataset. They will apply their CNN for faster RCNN. Use map technology to find the images that the learning model can identify.

Nkosingiphile Ndabandaba and Ishmael Sithole (Fast genotype calling)

Managed to read the idat files and output raw bytes and use UTF8 reader to convert them into readable characters.

PROF Scott. write report and be busy with repo and maintain. Justify every solution

Meeting adjourned.

Cohort 12: 5th Meeting Minutes

21st of October 2022

Group: 90, Chair: Muchaveleli Manjate, Minutes: Devlan Mckenzie

P101 Benjamin Palay and Raphi Druion (Facial recognition)

Determined that manual machine learning wont work. Found a model that can be trained however the model struggles when zoomed in on features like the eye due to pixelation. To solve this problem, we took 9 points and averaged their values. Attempting to use the cluster to get 10 000 images for training. Proceeded to demonstrate a prototype which is trained using 700 images. A real time database has been implemented which tracks the number of iterations it took to get a similar face and creates a graph showing how closely related the features are. Trying to get a larger set to increase the accuracy of the program.

G74 Haffejee Mohammed and Thabo Tshabalala (App to scan pool water test strips)

Demonstrated their program which can scan test strips but is inaccurate. Over the next 2 weeks they plan to further increase the scan accuracy which is currently correct 1 out of 10 or maybe 20 times. Proceeds to further demonstrate the program showing features such as photo capture, new pool and auto processing.

Prof Estelle: Remember to document what you did and didn't manage to achieve for future work to flow smoothly. Additionally, the code must be accessible for the future work.

They found 2 methods to increase the accuracy of the program. The first method is to fix under and overexposed images. The next method is to generate colour charts and uses those to group accuracy. The plan for the coming week is to work on increasing the accuracy.

G75 Ahmed Ibrahim and Kevin Naidoo (The lost cities of South Africa)

Didn't bring a laptop and will demonstrate the program next week. They refined the SSD model which works acceptably now. Want to swap to an SNN model with increased accuracy and prof wants to get specific location images scanned. SNN will take longer to work with and analyse but will increase the overall accuracy of the program and they plan to work on it in the coming week.

G51 Robin Jonker and Tristan Lilford (Location aware scientific workflows)

Greetings, on Wednesday they were given workflows to work with and have been integrating the static and dynamic aspects of their project with the new workflows. Run time is 6m 30s approximately and 2m 18s for the optimal. The dynamic times are 4m 16s. Lately been looking into using AWS and spoke with the prof but its not looking like its going to happen. They have been doing lots of research into S3 backet. Currently they can pick a region and moved onto dummy tests and are in the process of moving into actual testing. In response to a question about reserving nodes to increase run time they said that they cannot reserve nodes as that would defeat the purpose of the project which is to dynamical assign nodes and determine which nodes would be best to use.

G48 Nkosingiphile Ndabandaba and Ishmael Sithole (Fast genotype calling)

Finalised the data which was generated from an array. Testing if the program can read the data and did 2 sets of testing which involved assuming different file sizes. They demonstrated the data set which the prof gave them. The first task was to determine the pathing of each file and to open each file. There are 4 genotypes using red and green colours. They spoke with prof and were

recommended to use integers to store the data and to place them in an csv file. Then use the csv file to plot the data. An issue they ran into is the lack of publicly available resources related to this project. So, they attempted to research the problem in greater depth to find resources. Looking into clustering the data as they require it. They have 2 aims in mind the first is to use parallelism to improve the code and the second is to use the prof's function to plot the red and green data as the 4 genotypes. An issue is that they are unable to cluster all the data together and are looking into a way to solve that.

G67 Johann Gouws and James Allsop - Mapping the Rainbow (twitter):

The main issue is getting followers as even a small set of data takes 3 hours to process. The prof suggested that they contact the API people to get their professional help but only managed to get a meeting for next Friday. They did however manage to improve the code so that it takes 1h 30 minutes but that is still too time consuming, and automation limits the control they have over the data. Looking into research material, which is centred around topics, search for topics and then search for tweets which is faster but gives a poor connection in terms of corelation. This change would alter the entire project scope, direction and dynamic. As the scope has basically changed to a topic basis, they plan to create a bounding box next week and pull topics to see sample size. If all else fails, they are just going to use topics to pull data because if they get sample data then they can apply a model to it and obtain a variety of information. They are aiming to get a MVP as there is simply too much information to filter through.

G36 Joseph Baggott and Hraklis Papageorgiou- The world's protest capital:

Last weeks issue with the API was resolved by pulling all the SA protests and creating their own database which is much faster to pull data from than before, however this process did take awhile due to the tweet analysis, spark, tactics, and location data collected in a broad sense. The program does sentimental analysis on a scale of 1 to 100.

Asked by prof Estelle to do sentimental analysis on the lightning storm which occurred during the week. And to dump the tweets for later use.

Look into vid analysis but determined that it's a bit beyond the scope of the current project.

G15 Rael Ware and Gia Croock- Cheap Air Quality Reporting Station:

Demonstrated the application which shows live data from the SDI location. The air is explained to be clean and for testing purposes they will go burn toast at the location after this meeting. The data is analysed in Matlab on an hourly and daily level. They proceed to discuss various air quality monitor costs and explain that their monitor can sense at a 2.5 level and a 1 level which is less accurate than the 2.5 level. The 1 level tends to overestimate and is thus less accurate, but the data is in the right shape and thus they plan to calibrate the measurement device to be more accurate. At the 2.5 level the device is very accurate and not much work is needed for this level. They explain that at 1 level it is 77% accurate and after processing they get 92% accuracy but discuss how external factors can affect this accuracy.

P61 Devlan Mckenzie and Muchaveleli Manjate- Network firewall management and visualisation:

They explained that they met with the prof and got further advice with regards to converting the rules into Boolean expression and creating the BDD. They have been working in tulip and control and have basic rule conversion working. There is still and issue with IP address conversion, but all other field can convert into a Boolean expression successfully. The group also started looking into using

Pyeda which is another python package which deals with Boolean expressions and BDD creation due to the technical difficulty of using the lower-level tulip and control packages. The basic level of BDD can be created and is seen to work but the group has not yet combined all the rule expression into a single BDD. The plan for the coming week is to further refine the logic around rule conversion and to store and create the entire ruleset as a BDD.

Meeting adjourned.

University of the Witwatersrand, School of Electrical and Information Engineering

Software and Machine Learning 4th Year Investigation Projects

Weekly Meeting Minutes

WEEK 6: 28 October 2022

The 6th meeting of the Software and Machine Learning 4th Year Investigation Projects was held in the 4th year Common room. The meeting commenced at 11:10 am and was facilitated by group 22G74, Chairperson Mohammed Haffejee(1435060) and Secretary Thabo Tshabalala(1826096), with assistance from Joseph Baggott(2169705).

Open Remarks: Mohammed Haffejee welcomed everyone to the meeting. The meeting began with an announcement from Prof. Estelle Trengove about the details on Open Day which takes place the following week.

Attendees:

Supervisors: Prof. Estelle Trengove, Dr. Martin Bekker.

Groups: 22G75, 22G18, 22G15 ,22G51, 22G74, 22G67,22G36,22G90 ,22G48

Apologies:

Prof. Scott Hazelhurst

General Meeting Agenda: Groups discuss the current week's project progress and next week preparations for next week's Open day.

Opening Issues and Announcements:

- **Students are reminded that** Wednesday is the dry run and Thursday is the presentation for Open Day. Students won't be able be physically demonstrating their project in person, but you will need to have good photos. Don't have too much text but have nice photos. It should be able to appeal to a wide range of people, students, etc.
- Students need to explain their projects in such a way that you could explain it to your genius granny. Think about how you would explain your project to the ordinary layman project in a simple and efficient way. The presentation won't count towards to the final mark of the project.
- During the interview you need to be clear about what the individual work of the project was. The marks are then decided there and then after you the examiners have gone through all the projects.
- For the posters look at something called better poster. Invariable a bad poster is too in-depth about how the actual algorithm works. Some bullet points and good photos are what you should look for.
- Focus on the report, leave yourself 7 or 8 days to work on the report. You then need to talk about why you couldn't finish.
- Students need to show that you followed an engineering process and how you could do things differently the route that you followed. You need to do a serious introspection of your project, including reflecting on dead ends – this can be used as how hard you worked. The solution didn't just fall on your lap.

Students Presentations:

Project 22P101

Project Title: Facial recognition

Project progress, plan and problems encountered: Yesterday the app was released to retrieve feedback from the cohort group. The aim is 70% accuracy, and this is something that they are achieving. (Accuracy refers to the image that the user returns to the original image). Most of the time was spent on detecting whether the face had glasses, however this was a failure. Finding beards was a success. It hasn't been tested on large data sets, but on smaller data sets have been successful. The beard detection algorithm uses a similar software to find whether users have a mask, and the group is hoping to use this once the last outcome has been achieved. The work has been split, such that half the group

Project 22P34, Group 22G67:

Project Title: Mapping the Rainbow

Project progress, plan and problems encountered:

Discussion on data:

This week the group decided to finally analyse tweets in Johannesburg, this includes likes, mentions ect. Furthermore, the group focused on the tweets within Johannesburg. The group can find all the usernames and they are able to find all the people that they have mentioned last month. However, they were getting users that weren't found in Johannesburg. They have managed to display some graphs for 500 graphs that show the relationships between the users within Johannesburg. The final graph won't be a circle as shown in the demonstration. The issue is that even 50000 tweets is creating a very large graph and so the final graph will be more in depth The next goal is to show the communities within Johannesburg better in a more visual way.

Final takes from Group:

Final scrape of all the tweets. Going to look at student led protests. Going to use content analysis to find the tweets that don't fall within the correct municipalities.

Note from bekker:

The main challenge of this is how you going to cluster the communities.

Project 22P33, Group 22G75:

Project Title: The lost cities of South Africa.

Project progress, plan and problems encountered:

Plan for the week was to get the ResNet model to work, there were problems with the previous performance of the older models. When it gets to 60% the older model starts to crash. The new model already works better than the previous model and operates at 60%.

Plans for next week:

Feed the model images and when an image achieves 60% confidence, they will move the model to a new folder to be used later. The model can detect kraals with good confidence, higher resolution has better confidence. The front-end part of the system allows the user to upload either a photo or a satellite image and the system will crop to the image of where the kraal is located. The system might be able to return the location if the location is contained within the data. A DATAs set from the South African satellite service could be used.

Project 22P62, Group 22G48:

Project Title: Fast Genotype calling

Project progress, plan and problems encountered:

The group explored the different clustering algorithms and have settled on the caymans clustering algorithm. However, this algorithm works well on only small data sets. The problem with the algorithm is outliners, furthermore the issue is that the data sets that are given is larger than what is required by the algorithm needed in the project. The group was given a document that they are using to explore possible future solutions as to how to run the code in the cluster. The aim is to remove the outliers and to be able to run the algorithm on the cluster.

Project 22P87, Group 22G15:

Project Title: Cheap Air Quality Reporting Station.

Project progress, plan and problems encountered:

Last week they were discussing tests and evaluations, furthermore machine learning was applied to the learning data. Humidity is causing problems as it is being read as pollution, machine learning is being used to counter act this. They are allowing the microcontroller release wifi which allows the user to access the settings. When high pollution is detected, the device will send a SMS to the operator of a warning.

Project 22P61, Group 22G90:

Project Title: Network firewall management and visualisation

Project progress, plan and problems encountered:

Last week struggled to create conversions, hoping to complete the project before the deadline. Furthermore, hoping to have visulisasions for open day. IP address is an issue as the model does not see numbers as variables even if you use it as a string variable or an integer.

Project 22G63, Group 22G51:

Project Title: Location aware scientific workflows.

Project progress, plan and problems encountered:

Everything is working with slerm, this week there is a focus on testing. The load testing is challenging as the cluster is relatively quiet, as a result the group is going to need to create a backlog. The static and the dynamic characteristics of the loas are very similar. Three tests are run per condition to achieve an average as well as two conditions per test. AWS has been delayed as focus was put on the graphics. Implementing the AWS will be difficult in terms of technicality as well as security. A new library is going to be used determine when load is has been placed on the system.

Project 22P21, Group G74:

Project Title**:** App to scan pool water test strips

Project progress, plan and problems encountered:

Adjournment: The meeting was concluded at 12:15 pm by Prof. Estelle Trengove and Dr. Martin Bekker and they gave their final regards.

Minutes Submitted by: Thabo Tshabalala and Mohammed Haffejee.