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.

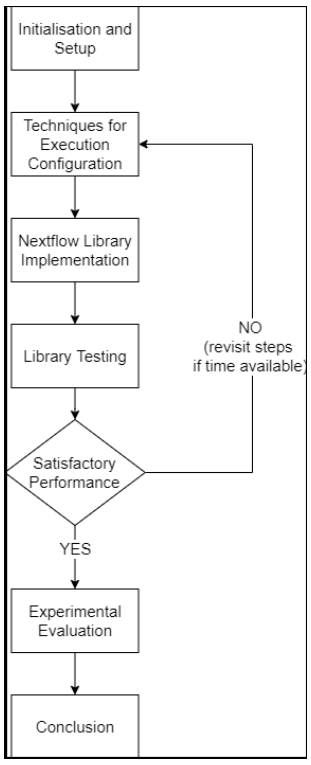


Figure 1: Flowchart of Project Methodology

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

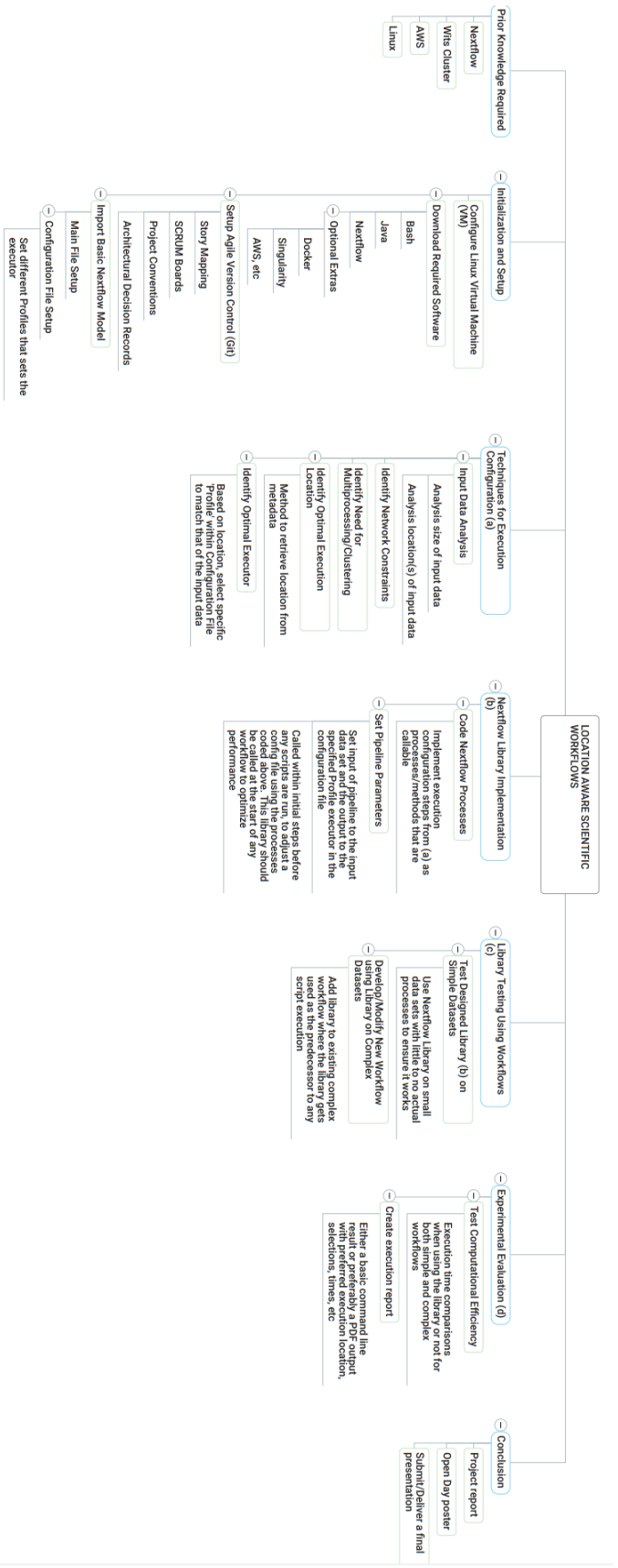
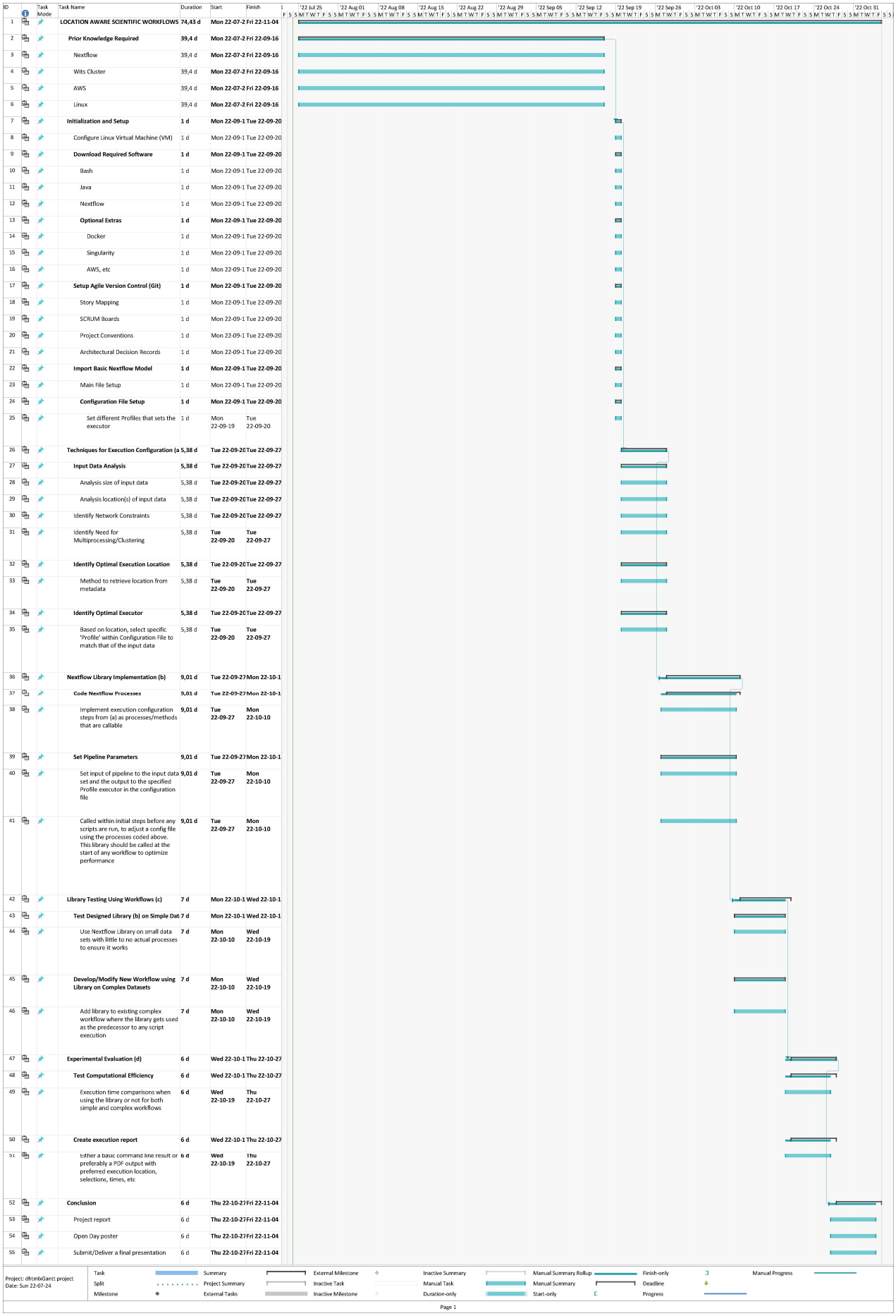


Figure 2: Detailed Work Breakdown Structure

B GANTT CHART



ID	Task Mode	Task Name	Duration	Start	Finish
1		**LOCATION AWARE SCIENTIFIC WORKFLOWS**	**74,43 d**	**Mon 22-07-25**	**Fri 22-11-04**
2		**Prior Knowledge Required**	**39,4 d**	**Mon 22-07-25**	**Fri 22-09-16**
3		Nextflow	39,4 d	**Mon 22-07-25**	**Fri 22-09-16**
4		Wits Cluster	39,4 d	**Mon 22-07-25**	**Fri 22-09-16**
5		AWS	39,4 d	**Mon 22-07-25**	**Fri 22-09-16**
6		Linux	39,4 d	**Mon 22-07-25**	**Fri 22-09-16**
7		**Initialization and Setup**	**1 d**	**Mon 22-09-19**	**Tue 22-09-20**
8		Configure Linux Virtual Machine (VM)	1 d	Mon 22-09-19	Tue 22-09-20
9		**Download Required Software**	**1 d**	**Mon 22-09-19**	**Tue 22-09-20**
10		Bash	1 d	Mon 22-09-19	Tue 22-09-20
11		Java	1 d	Mon 22-09-19	Tue 22-09-20
12		Nextflow	1 d	Mon 22-09-19	Tue 22-09-20
13		**Optional Extras**	**1 d**	**Mon 22-09-19**	**Tue 22-09-20**
14		Docker	1 d	Mon 22-09-19	Tue 22-09-20
15		Singularity	1 d	Mon 22-09-19	Tue 22-09-20
16		AWS, etc	1 d	Mon 22-09-19	Tue 22-09-20
17		**Setup Agile Version Control (Git)**	**1 d**	**Mon 22-09-19**	**Tue 22-09-20**
18		Story Mapping	1 d	Mon 22-09-19	Tue 22-09-20
19		SCRUM Boards	1 d	Mon 22-09-19	Tue 22-09-20
20		Project Conventions	1 d	Mon 22-09-19	Tue 22-09-20
21		Architectural Decision Records	1 d	Mon 22-09-19	Tue 22-09-20
22		**Import Basic Nextflow Model**	**1 d**	**Mon 22-09-19**	**Tue 22-09-20**
23		Main File Setup	1 d	Mon 22-09-19	Tue 22-09-20
24		**Configuration File Setup**	**1 d**	**Mon 22-09-19**	**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 22-10-10**
37		**Code Nextflow Processes**	**9,01 d**	**Tue 22-09-27**	**Mon 22-10-10**
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 22-10-10**
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 22-10-10**	**Wed 22-10-19**
43		**Test Designed Library (b) on Simple Dat**	**7 d**	**Mon 22-10-10**	**Wed 22-10-19**
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 22-10-19**	**Thu 22-10-27**
48		**Test Computational Efficiency**	**6 d**	**Wed 22-10-19**	**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 22-10-19**	**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**