

Research Proposal

by William Baker

FILE	3611_WILLIAM_BAKER_RESEARCH_PROPOSAL_50131_1600093589.DOC X (460.85K)		
TIME SUBMITTED	03-MAY-2019 11:30PM (UTC+0100)	WORD COUNT	4 108
SUBMISSION ID	106296754	CHARACTER COUNT	267 19



Research Proposal

Mixed Mode Parallelism - Programmatic Elicitation of Optimal Combinations of Parallelism Method(s)

William Baker – 1703131

Software Engineering

Dr Tim Bashford



Table of Contents

Table of Figures	2
1. Introduction / Background to the Proposed Research	3
1.1. Introduction	3
1.2. Title.....	3
1.3. Background	3
1.4. Field of Study	5
1.5. Aims and Objectives.....	6
1.6. Outcomes and Deliverables.....	6
1.7. Conclusion	7
2. Research and Development Methods.....	7
2.1. Introduction	7
2.2. Research Methods.....	7
2.2.1. Research Philosophy	7
2.2.2. Research Strategy.....	8
2.3. Generalisability, Reliability, and Validity	8
2.3.1. Reliability	Error! Bookmark not defined.
2.3.2. Validity	Error! Bookmark not defined.
2.3.3. Generalisability	9
2.4. Data Collection.....	9
2.5. Development Methods.....	10
2.6. Risks, Limitations, and Legal Issues	11
2.7. Data Analysis	11
2.8. Conclusion	11

3. Resources and Planning.....	12
3.1. Introduction	12
3.2. Required Resources.....	12
3.3. Project Planning	12
3.4. Conclusion	15
References.....	16

Table of Figures

Figure 1 - Waterfall vs Agile Development Methods [13]	10
Figure 2 - Gantt chart for the research project.....	13
Figure 3 - A section of the Gantt chart	13
Figure 4 - More of the Gantt chart	14
Figure 5 - The last piece of the Gantt chart, showing the total development time of the project	14

1. Introduction / Background to the Proposed Research

1.1. Introduction

This section of the research proposal will cover the background information of the project. It will cover some basic information about the research, the background area, and some aims and objectives of the research. It will also cover the expected deliverables and outcomes of the project.

1.2. Title

- Give your research a working title.

The title for the research is Mixed Mode Parallelism – Programmatic Elicitation of Optimal Combinations of Parallel Method(s).

1.3. Background

- Describe clearly the background in the subject area of study.

The subject area of study is parallel computing. Traditional software is broken into a series of instructions which are computed sequentially by a single processor, one after another. Since early computers only had one processing core, this was the fastest way of executing the instructions. Modern computers have more than a single core in each CPU (Central Processing Unit), and some computers even have multiple CPUs. However, most modern software is still computed on a single CPU core, meaning most of a computer's cores are left unused. Parallel computing is the idea of using the multiple cores or compute resources to compute the instructions in a parallel way, making the most of the CPU's power. [1]

Parallelism goes beyond utilizing the extra cores in a CPU. There are three common ways of parallel computing. The first is simply multi-threading software, as well as some different ideas such as GPGPU (General Purpose Graphics Processing Unit) and cluster processing.

Cluster computing is the idea of using multiple computers (a cluster) to run a single program; this is similar to multi-threading in the sense you are effectively connecting multiple processing cores, except this time they are from an entirely different CPU in an entirely different computer. Cluster computing uses the combined resources of a computer network to run a single

program or task. This differs from cloud computing which is generally used to run many different programs or services, for many different people. Cluster computing is often used in modern supercomputers, which are generally used for complex scientific calculations and research. One of the benefits of cluster computing over traditional supercomputing using shared memory is the fact that one computer can be removed for maintenance and the cluster will still be usable. [2]

GPUGPU is pretty much as the name states, using the GPU (Graphics Processing Unit) for general purpose algorithms. The GPU is usually reserved for rendering 3D graphics, but it can be used to run many generic algorithms. Hackers often use the GPU for cracking passwords; it can also be used for mining cryptocurrency or other mathematically intense operations. A GPU has a limited instruction set compared to a CPU, and therefore not all programs can be run on it. Because of this, it is not common for a programmer to utilise it in their software, and many may not know how or what it can be used for. Kyrnin states "Thanks to the speed of processors, most people do not notice any real slowdowns. There are a variety of tasks though that can bog down a computer's central processor." [3]. Using the GPU for specific tasks could significantly increase the performance of many applications. [3]

Modern software is severely lacking parallel computing methods; this is generally because parallel computing methods are not easy to implement meaning many programmers either do not understand or cannot be bothered to implement any of these methods. A lot of software could benefit from the use of multi-threading, but it is not a straightforward process and can create many issues for the programmer. One type of issue that is commonly caused with multi-threaded code is sharing a variable; this can cause concurrency errors in the code since multiple threads are trying to access or change a variable at the same time. [4]

There are five common problems with multi-threaded software [5]:

- Deadlock – Where two processes are waiting for each other to finish, meaning neither can finish.
- Starvation – This is where a process is unable to gain access to a resource, meaning it is unable to finish executing.
- Race Conditions – This is when two processes that are supposed to occur in a set order end up out of their order due to the threads running concurrently. This can cause all sorts of weird outputs depending on the program.
- Livelock - When two threads are depended on data or signals from each other and respond to each other, it can cause a loop which can be described as something between deadlock and starvation.

For many smaller pieces of software, multi-threading does not make sense. It is tricky for the programmer to do and when the program is basic, will only benefit the performance a tiny amount, if at all. In some cases, multi-threading could slow down the execution time depending on what is bottlenecking the program.

Like with multi-threading, GPGPU or running code on the GPU is also not a simple task. Although there are many platforms out there to make this easier such as OpenCL or CUDA, many programmers are not taught or do not take the time to learn these, or they do not realise the benefit of running code on the GPU.

The main reason that parallelism is increasing in popularity and is required more often in modern software is that Moore's law is effectively coming to an end. Moore's law was an observation made by Gordon Moore in 1965; it stated that every 18 to 24 months the number of transistors in a processor would double. However, in the last 15 years or so, the number of transistors has mainly been increasing through core count and not transistor size. We have almost reached a point that silicon transistors physically cannot get smaller. The smallest transistors in a consumer CPU today are 7 nanometres, and although it is possible to go smaller, it is not currently commercially viable due to an effect called quantum tunnelling which is effectively a current leakage in the transistor causing it not to function correctly. [6]

Although clock speed does not entirely tell us about the speed of a processor, it is one of the significant factors. In the last ten years clock speed has not increased a significant amount, therefore utilising the extra cores of the processor is the only real way for computers to get faster in the future. [7]

1.4. Field of Study

- Provide the reader with an understanding of the field in which your study lies and an idea of where and how your study fits into this field.

The project that will be developed will be an automatic code analyser that will be able to discover which sections of a specified code sample could benefit from parallel methods. This software should be usable by any level of programmer; it would show the programmer exactly how and where they could use parallel methods to increase the efficiency of their application. This would allow even novice programmer to make better use of the increasing core counts and available GPU within a computer.

Intel created a piece of software called 'Intel Advisor XE' to help programmers optimise code for modern computer architectures. However, this software is focused on solely the use of multi-threading and vectorization. Intel created this software because they know too well of

the lack of parallelism in modern software and the lack of use that the extra cores in the CPUs get. [8]. The proposed research project aims to extend further than Intel has done with their advisor software and to make use of a combination of parallel methods.

1.5. Aims and Objectives

- Define one overarching aim of the study. Construct a series of objectives that specify what you will do to achieve the aim.

The main aim of the study is to understand how code can be analysed in order to give suggestions on the implementation of parallel methods. Some objectives must be set to achieve this:

- Investigate the best method of code analysis, pre-compile or compile.
- Find the best method of calculating/choosing the combination of parallel methods that are best suited to the input software.
- Investigate and implement a way of automatically applying parallel methods to a piece of code, rather than merely suggesting them to the user.
- Evaluate the new program, benchmarking and collecting data on the performance of the software before and after their suggested optimisations.

1.6. Outcomes and Deliverables

- Expected outcomes and deliverables.

The expected outcome of the research is a program that will be able to automatically parallelise a given piece of software using a combination of parallel methods, or at the very least, suggest some places where parallel methods could be implemented for better performance. For example, it would be able to automatically detect places in the code that multi-threading could have been used or code could have been run on the GPU in order to optimise it

This piece of software could be split into two sections, one that will just analyse the code and suggest places that could benefit from parallel methods and one that automatically implements them. This way, the user has more of a choice of the outcome of the software. The part that suggests optimisations will be very similar to Intel's Advisor, except it will include suggestions for multiple parallel methods. The ideal project could also automatically implement these methods, making it even more powerful than Intel's Advisor.

The project will most likely be implemented using C/C++; however, it is possible that Java or other languages could be used.

1.7. Conclusion

This section has outlined the different types of parallel methods, why they are useful, and why they are needed now more than ever. It has also shown any issues with implementing these methods and why the proposed project could make a programmer's life much more comfortable. It has also discussed the expected outcomes of the project, and what the ideal deliverables will be.

The next section will cover the research and development methods that will be used for the project.

2. Research and Development Methods

2.1. Introduction

The previous section covered the background and aims of the research project. This section will cover the research and development methods of the proposed project. This includes philosophies behind the research, the strategies, and the ethical and legal issues.

2.2. Research Methods

- Identify and justify the research methods that you will use to gather your data and evaluate your results.

2.2.1. Research Philosophy

The research project will have a positivist paradigm. Positivism is the idea that only factual knowledge gained through observation and measurement is trustworthy. Since experiments will be carried out to find the best method of implementation, a positivist approach suits the project perfectly. In the positivist paradigm, the researcher is totally independent from the study and focuses on the facts rather than their own interest. [9]

The alternative paradigm would be phenomenology or a phenomenological approach. The phenomenologist paradigm is mainly focused on events and how humans experience the phenomenon. This would not suit the project in the slightest, as the project requires is purely based on the measurable performance and optimisations of software, not any person's experience of the software. [10]

There are two main approaches to research, deductive and inductive. The deductive approach involves having a hypothesis which is then tested throughout the research. In the case of this research project, the hypothesis is that software performance can be effectively improved through the automated application of a combination of parallel methods. This hypothesis will then be tested through the attempted development of said software, such as the structured code analysis that will choose the optimal combination of methods.

The inductive approach usually involves collecting data and analysing it to develop a theory. This approach is partially suitable for the project because different parallel methods will be tested on specific algorithms to see which performs best, this will aid the development of the final program as the data recorded should show when and where a parallel method is suitable. It could be said that this project is simply extending what we already know, that parallel method(s) CAN improve software performance, and is just seeking to understand the best choices for specific software and how a programmer can be aided in the implementation of those methods. However, there is still a hypothesis for the project, and therefore the deductive approach is the correct one. [11]

2.2.2. *Research Strategy*

The research strategy for this project would be experimental research the whole idea behind the research is to create a program to find and apply an optimal combination of parallel methods automatically. Experimental research is as the name suggests – carrying out experiments as research. In the case of this research project, most of the research will have to be experimental as the data required does not exist, and the idea is to produce a completely new result. The idea of combining multiple methods of parallelism is relatively new. [12]

Another research strategy for this research could be 'exploratory' since it is exploring parallel methods which are already known. The research will be carried out in how these parallel methods can be implemented. Exploratory research can be split into three necessary steps: [13]

- Identifying the problem – A researcher must decide about a subject area of research
- Create a hypothesis – The researcher must then create a hypothesis within this area.
- Researching the data – The researcher must then actually study this hypothesis.

This project will use a combination of both experimental and explorative research. Since parts of the implementation of this project are already known and will be researched, that counts as exploratory research, but most of the project will be done via experimentation. The primary experimentation of the project will be the decision of which parallel methods suit an algorithm, then developing software that can automatically detect said algorithm and suggest or apply the parallel method(s).

2.3. **Generalisability, Reliability, and Validity**

- Generalisability, reliability and validity

2.3.1. Reliability

Because of the research that will be carried out is a piece of software, the result could differ to some degree based on the implementation by the researcher. Although in general, the result should be similar. The quality and performance of the implementation could differ somewhat. However, the research should give a good overview of what can be expected from the implementation of parallel methods and show how they can be automatically suggested and implemented.

2.3.2. Validity

Developing this software is an incredibly appropriate way of researching the topic because it will show the real-world results of implementing a combination of parallel computing methods. The right data will be collected as it will use conventional programming techniques such as for-loops, mathematical operations, and standard algorithms when testing the different combinations of parallel methods.

2.3.3. Generalisability

The results of the project should apply to most software, depending on how it is implemented. The choice of programming language and techniques could change the usability somewhat as it may only be compatible with certain languages depending on the implementation.

2.4. Data Collection

- Identify your data collection methods for each objective and explain what data you intend to gather.

In the first task of the project, researching the optimal parallel methods for a given algorithm, an observational data collection method will be used. Several algorithms will be created, each having one or more versions, the first will be without any parallel methods, and the rest will have a specific combination (or singular) of parallel methods implemented. The performance data and run times of these algorithms will be recorded and observed to see which parallel methods perform best with a given piece of code.

Observational data collection will also be used for the implementation of the main application since data will have to be recorded and observed, the data being the run time and efficiency of the optimised applications before and after their optimisations. [14]

2.5. Development Methods

- Identify and justify any development methods and techniques that you will use as part of systems development.

There are many different development methods for software development, there are a number that could suit the research project incredibly well. However, there is one that sticks out as being the most suitable, agile development. Agile development differs greatly from the traditional waterfall methodology which works by splitting the software into different sections for the requirements, design, development and testing. Instead, with agile development, there is limited upfront planning and instead, multiple iterations of the software are created, each different in some way. Figure 1 shows the difference in the two methods and how they are split up.

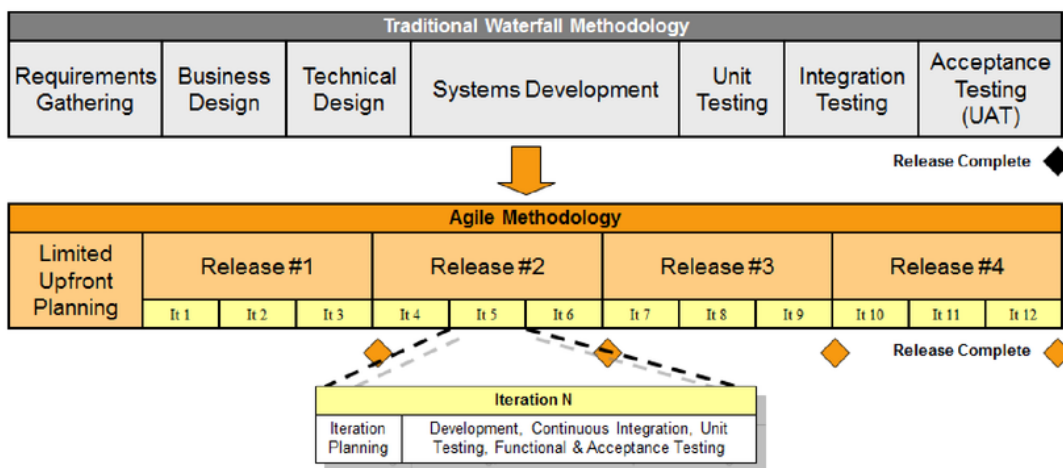


Figure 1 - Waterfall vs Agile Development Methods [15]

Agile development suits the project well because there are many parts of the project that need to be worked out. At the beginning of the project, structured code analysis will have to be carried out, and there are many ways of doing this. Applying the optimisations are also very tricky and may require a few attempts to work out. The problem with this project is that there are many options for development, with very little documentation for each. Therefore, the program may be changed many times before the final version meaning a feature-driven development method is most suitable. [15]

One of the problems with agile software development is the uncertainty that it brings since very little planning is done, it is harder to predict when deadlines will be met, and the project will be finished. Between each iteration of development, there can be significant delays as

some planning has to be done to decide what the next step is, versus having it planned from the beginning and development can continue all the way through to the end. [16]

2.6. Risks, Limitations, and Legal Issues

- Identify any risks/limitations/ethical/legal issues and ways that you might deal with those risks.

The only risks identified with the project would be due to an error on behalf of the researcher as opposed to any external issues. These include:

- Running out of development time.
- Choosing unsuitable libraries or methods for doing the structured code analysis.
- Being unable to reach the requirements and goals of the implementation - this would be due to the complexity of the system.

There are no legal issues that could arise from this project; any applications required for testing the project will be created by the researcher and made for testing the performance of specific algorithms. There is no personal data or sensitive information required to complete the research, and all finds will be able to be published freely.

2.7. Data Analysis

- How will you analyse the data/evaluate your results?

A multitude of methods are available to be used for analysing the data recorded. Software tests will be used for the development of the prototype software used for testing the parallel methods, as well as the final project which will automatically suggest the optimal combinations. Benchmarking will be used to create a model of the correct parallel methods to be used with specific algorithms, as well as being used to record the before and after the suggested/automated optimisations by the system to see if there has been a significant improvement in performance.

2.8. Conclusion

In this section, several research and development methods were discussed, explaining the reliability, validity, and generalisability of the research. It covered some necessary details on how data will be collected for the project and how that data can be used to create the project most effectively. It has also covered any issues that may arise during the development – be it ethical or legal issues.

The next section will cover all of the resources that will be required to complete the project, and also any project planning,

3. Resources and Planning

3.1. Introduction

The previous section covered the research and development methods to be used in the research project. This section will cover any resources that will be required to complete the project, as well as the planning and project management.

3.2. Required Resources

- Identify any resources required.

The following resources will be required to complete the project successfully:

- PC - To be used for programming the system, recording data, and researching.
- Microsoft Visual Studio - For programming the application.
- Microsoft Excel - For recording data/performance values when testing different parallel methods on algorithms.
- Internet access - For researching some of the core concepts of the project, such as structured code analysis.

3.3. Project Planning

- Construct a Gantt chart that demonstrates evidence of project planning and management. Identify the timescales for each component of the study.

A Gantt chart has been created to help plan the project. Each task that is required of the project has been put into the chart, with a set a date for its completion. The tasks were put in the required, logical order of completion, for example, developing software for testing the combinations of parallel methods logically comes before these tests are carried out. The total hours to be spent on the project are shown on the Gantt chart, although these will likely be updated at a later date.

Mixed Mode Parallelism							
Task name	Assigned	Start date	Duration (hour)	Progress	Estimation (hours)	Cost	+
		06/09/2019 09:00	1359		992	0	
☐ Initiation/Planning		06/09/2019 09:00	40	0%	40	0	⋮
Project Planning	W William Baker	06/09/2019 09:00	40	0%	40	0	⋮
Initiation/Planning Activities Completed	W William Baker	13/09/2019 09:00				0	⋮
Add a task Add a milestone							
☐ Development		01/11/2019 09:00	1039	0%	952	0	⋮
Develop Test Algorithms	W William Baker	01/11/2019 09:00	112	0%	112	0	⋮
Test Algorithms and Record Results	W William Baker	21/11/2019 09:00	48	0%	48	0	⋮
Analyse and Choose Optimal Parallel Methods for Algo...	W William Baker	29/11/2019 09:00	40	0%	40	0	⋮
Develop Structured Code Analysis	W William Baker	06/12/2019 09:00	608	0%	608	0	⋮
Test Software	W William Baker	23/03/2020 09:00	56	0%	56	0	⋮
Analyse Software Optimisations	W William Baker	01/04/2020 09:00	88	0%	88	0	⋮
Development Completed	W William Baker	30/04/2020 17:00				0	⋮
Add a task Add a milestone							
ⓘ Add a new subproject							

Figure 2 - Gantt chart for the research project

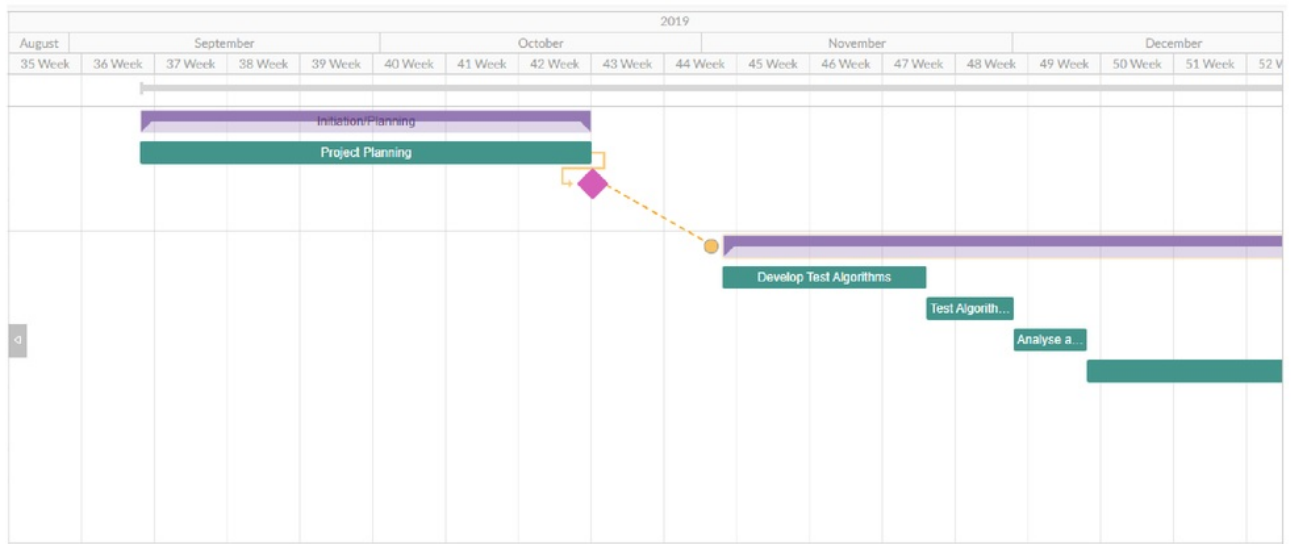


Figure 3 - A section of the Gantt chart

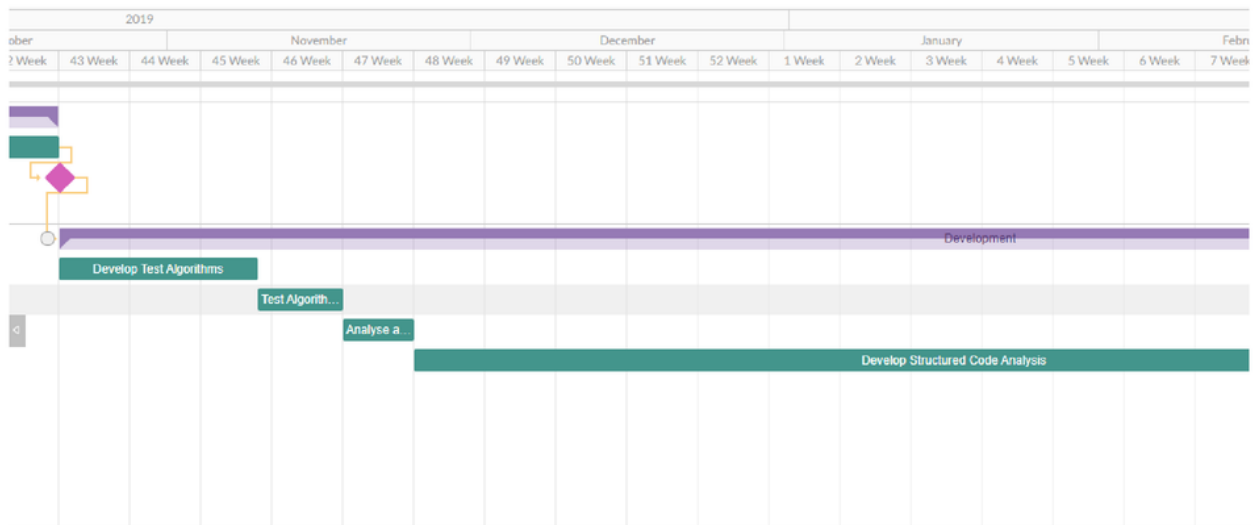


Figure 4 - More of the Gantt chart

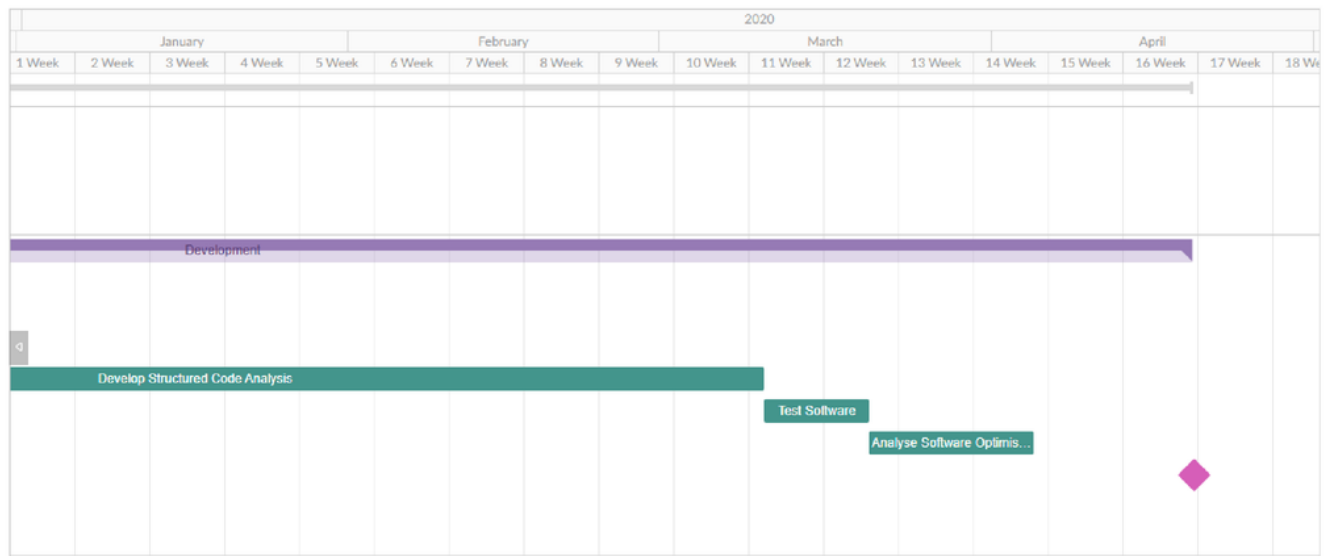


Figure 5 - The last piece of the Gantt chart, showing the total development time of the project

3.4. Conclusion

This section has covered the resources that will be required to complete the project, and the planning behind the project. This will greatly aid the development as the requirements and planning have been (mostly) carried out already. This research proposal has covered all of the information pertaining to the project, the background information, and the idea behind the implementation.

References

- [1] B. Barney, "Introduction to Parallel Computing," Lawrence Livermore National Laboratory, 25 June 2018. [Online]. Available: https://computing.llnl.gov/tutorials/parallel_comp/#WhatIs. [Accessed 1 May 2019].
- [2] F. Pierson, "Four things you need to know about Cluster Programming," Big Data Made Simple, 23 June 2017. [Online]. Available: <https://bigdata-madesimple.com/4-things-need-know-cluster-programming/>. [Accessed 2019 May 01].
- [3] M. Kyrnin, "Using Graphics Cards for More Than Just 3D Graphics," Lifewire, 23 April 2019. [Online]. Available: <https://www.lifewire.com/graphics-cards-3d-graphics-834089>. [Accessed 2019 May 01].
- [4] D. Bolton, "4 Problems with Processor Thread Programming," Dice, 7 February 2018. [Online]. Available: <https://insights.dice.com/2018/02/07/4-problems-processor-thread-programming/>. [Accessed 2 May 2019].
- [5] A. G. Walters, "Multithreading: Common Pitfalls," Austim G. Walters, 5 May 2014. [Online]. Available: <https://austingwalters.com/multithreading-common-pitfalls/>. [Accessed 2 May 2019].
- [6] Kurzweil, "Overcoming transistor miniaturization limits due to 'quantum tunneling'," Kurzweil, 7 June 2018. [Online]. Available: <https://www.kurzweilai.net/overcoming-transistor-miniaturization-limits-due-to-quantum-tunneling>. [Accessed 2 May 2019].
- [7] P. P. Mattsson, "Why Haven't CPU Clock Speeds Increased in the Last Few Years?," Comsol, 13 November 2014. [Online]. Available: <https://uk.comsol.com/blogs/havent-cpu-clock-speeds-increased-last-years/>. [Accessed 2 May 2019].
- [8] Intel, "Intel® Advisor User Guide," Intel, 6 February 2019. [Online]. Available: <https://software.intel.com/en-us/advisor-user-guide>. [Accessed 2 May 2019].
- [9] Research-Methodology, "Positivism Research Philosophy," Research-Methodology, [Online]. Available: <https://research-methodology.net/research-philosophy/positivism/>. [Accessed 3 May 2019].
- [10] Research-Methodology, "Phenomenology," Research-Methodology, [Online]. Available: <https://research-methodology.net/research-philosophy/phenomenology/>. [Accessed 3 May 2019].

- [11 D. D. Gabriel, "Inductive and deductive approaches to research," 17 March 2013.
] [Online]. Available: <https://deborahgabriel.com/2013/03/17/inductive-and-deductive-approaches-to-research/>. [Accessed 3 May 2019].
- [12 A. Bhat, "EXPERIMENTAL RESEARCH- DEFINITION, TYPES OF DESIGNS AND ADVANTAGES,"
] Question Pro, [Online]. Available: <https://www.questionpro.com/blog/experimental-research/>. [Accessed 3 May 2019].
- [13 A. Bhat, "EXPLORATORY RESEARCH: DEFINITION, METHODS, TYPES AND EXAMPLES,"
] QuestionPro, [Online]. Available: <https://www.questionpro.com/blog/exploratory-research/>. [Accessed 3 May 2019].
- [14 Cyfar, "Data Collections Techniques," [Online]. Available: <https://cyfar.org/data-collection-techniques#Observations>. [Accessed 3 May 2019].
- [15 Inflectra, "Introduction to Agile Software Development Methods," Inflectra, 1 March 2018.
] [Online]. Available:
<https://www.inflectra.com/ideas/whitepaper/introduction%20to%20agile%20development%20methods.aspx>. [Accessed 3 May 2019].
- [16 M. Bridges, "The problems with agile development," App Developer Magazine, 16
] February 2017. [Online]. Available: <https://appdeveloper magazine.com/the-problems-with-agile-development/>. [Accessed 3 May 2019].

Research Proposal

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

80/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

WILLIAM BAKER

80

ENGAGEMENT WITH LITERATURE



Some very nice work here. The sources selected need some work, however – ideally, as you move into level 6, you want to move away from web-based sources. We'll discuss this in upcoming supervision meetings.

20

KNOWLEDGE AND UNDERSTANDING



A nice introduction with good aims and objectives. Take a look at SMART as a way of structuring your aims – again, we'll have a chat about this during our supervision meetings. A good overall conclusion – and I especially like the use of sub-conclusions.

13

COGNITIVE, INTELLECTUAL SKILLS



The proposed methodology is clear and sensible, though will inevitably evolve with the project.

16

PRACTICAL SKILLS



Good research and development methodologies proposed. We'll discuss and develop these through the year. Good planning in terms of resources needed etc. The Gantt charts are possibly a little optimistic, but again, we'll develop a robust plan through supervision meetings.

20

TRANSFERABLE SKILLS



A very nicely presented, well written document. As a very minor comment, you could turn some of the pages containing images landscape and potentially reduce the number of times you need to, for example, split the Gantt chart.

11