GENG5551 Research Project

RESEARCH PROPOSAL

Thomas Hill Almeida (21963144)

Supervisor:
Professor Linqing Wen

2020-04-20

Contents

1 Introduction

1.1 Background

In 1687, Isaac Newton published his book, "*Philosophiæ Naturalis Principia Mathematica*", containing his law of universal gravitation:

$$F = G\frac{m_1 m_2}{r^2}, \tag{1}$$

where F is the resultant gravitational force, G is the universal gravitational constant, m is the mass of the objects and r is the distance between the objects.

It has since been taught in high-schools across the world in introductory physics classes, due in part to its simplicity and general accuracy for predicting movement. Despite its general applicability, one major issue of the theory is that it infers that gravitational is instantaneously applied, without any apparent method through which it could be transmitted.

Roughly two centuries later, in 1905, Albert Einstein presented the theory of special relativity in his paper "*Zur Elektrodynamik bewegter Körper*" (English: *"On the Electrodynamics of Moving Bodies"*). The theory introduced the concept of **spacetime** to describe inertial reference frames as a four-dimensional coordinate system, (t, x, y, z), where t is time and (x, y, z) are the three spatial dimensions. He further stated two important axioms; that the speed of light in a vacuum is the same for all observers regardless of motion and that the laws of physics are invariant in all inertial frames of reference. About ten years later, Albert Einstein incorporated the effect of gravity with special relativity, forming the general theory of relativity.

The general theory of relativity postulates that the effect of gravity can be characterised as each gravitational potential source changing the curvature of spacetime. The relationship of gravitational mass-energy and the shape of spacetime is given by Einstein's field equations:

$$G_{\mu v} + \Lambda g_{\mu v} = \frac{8\pi G}{c^4} T_{\mu v}, \tag{2}$$

where $T_{\mu v}$ is the stress-energy tensor [1], $G_{\mu v}$ is the Einstein tensor, $g_{\mu v}$ is the spacetime metric, Λ is the cosmological constant, G is the universal gravitational constant and c is the speed of light.

An implication of gravity curving spacetime is that massive accelerating objects would cause 'ripples' in fabric of spacetime called gravitational waves. The existence of gravitational waves remained a theory until 1974, when Russell Hulse and Joseph Taylor discovered a binary pair of neutron stars that were orbiting each other. After several years of measurement, they found that the speed at which the stars were orbiting each other was slowing in a manner consistent with the predictions of the general theory of relativity, showing that gravitational waves did indeed exist [1].

There were several experiments performed in the 1960s and 1970s to determine methods to detect gravitaional waves, resulting in several large laser interferometric detectors (that is, detectors that use interferometry – the phenomena by which waves superpose on each other to create a resultant wave – for detection) being built throughout the early 2000s, including the American Laser Interferometric

[1] Tensors are very similar to matrices of vectors and are typically used to describe mathematical geometric relationships

Gravitational-Wave Observatory (LIGO) [15], the Italian Virgo [18] detector, and the German GEO600 detector. The initial observation runs between 2002 and 2011 by the various detectors failed to directly detect any gravitational waves, and as such, the majority of the detectors began work to increase their sensitivity throughout the 2010s [11]. The increase in detector sensitivity has brought success in the search for gravitational waves, with the first direct detection occurring on the 14th of September, 2015 [6][14].

Due to their design, the detectors have a significant amount of noise from sources that are not gravitational waves, in addition to the gravitational waves themselves having very weak signals. As such, a large amount of data processing needs to be done to the outputs produced by the detectors in order to filter and extract any possible gravitational waves. These data processors are known as '*pipelines*', and are mostly created by research groups that are a part of the LIGO Scientific Collaboration [16]. These pipelines are used throughout observation runs for real-time data analysis.

One such pipeline is the Summed Parallel Infinite Impulse Response (SPIIR) pipeline, created by Shaun Hooper in 2012 [4]. The pipeline uses a number of IIR filters – which are commonly used in signal processing for bandpass filtering – to approximate possible gravitational wave signals for detection. The pipeline was further developed by Qi Chu in 2017, by using GPU acceleration techniques to increase the speed of analysis, as well implementing a method to use a frequentist coherent search [7]. The pipeline is currently the fastest of all existing pipelines, and has participated in every observation run since November 2015, successfully detecting all events that were seen in more than one detector.

The SPIIR pipeline uses GStreamer, a library for composing, filtering and moving around signals, in addition to the GStreamer LIGO Algorithm Library (`gstlal`). After receiving data from the detectors, the pipeline performs data conditioning and data whitening, followed by the usage of the IIR filters. The data is then combined for post-processing, where events are given sky localization and then inserted into the LIGO event database [9].

1.2 Problem Description & Goal

As of 2020-04-20, the SPIIR pipeline supports the use of two or three detectors for gravitational wave searching – the two American LIGO detectors and the Virgo detector. There are several issues with the current pipeline design that this research project aims to address.

Further detectors are likely to be coming online in the near future, with old detectors occasionally being removed from detection for maintainance. For example, the Japanese KAGRA detector is undergoing testing with the goal of being used in the next observation run, and LIGO India is currently being installed. With the current design of the pipeline, adding and removing detectors is a significant undertaking that takes a substantial amount of development time.

In addition, if a detector is indeed added to SPIIR, the detector ***must*** be used for the coherent post-processing, and can't be used just for synchronization purposes. This presents an issue as additional detectors are added, as each detector has its own sensitivity, reading variations and range of observable gravitational wave frequencies, resulting in some detectors being suitable for searching specific frequency ranges whilst showing no discernable change in output for other detectors, causing many false negatives.

An ideal architecture for the pipeline would be significantly more composable, able to add and remove the usage of different detectors for post-processing with minimal effort.

As such, this research project aims to complete a subsection of this idealised architecture. The project aims to remove the requirement for all detectors to be used for coherent post-processing with sky localization, and instead aims to provide a generic interface that would allow for any number of detectors to be used for coherent post-processing, whilst still allowing the unused detectors to undergo all other parts of the pipeline and remain synchronized with the used detectors. The project shall explore a number of different possible codebase refactorings as well as exploring new techniques for efficiently combining N data sources for coherent search, and shall measure the performance impact of the changes using a number of benchmarks.

2 Literature Review

This research project aims to refactor the SPIIR pipeline codebase and explore techniques for combining some unknown N number of data sources for coherent search. Whilst the refactoring part of the project is somewhat tangential to the aims of the project, looking at the literature for refactoring will help with the development of the research methodology. The literature for both refactoring and coherent search are developing and varied.

Refactoring is defined by [5] as the process of changing the structure of software without changing its behaviour. Of course, this isn't the only way to modify source code to address known issues. George Fairbanks offers the options of ignoring, refactoring or rewriting code as potential methods to deal with problems in [10], and makes several distinctions between the options. According to Fairbanks, the major difference between refactoring and rewriting is the process by which the code is modified. When refactoring, incremental changes are made to the odebase, with the major goal being to keep the newly written code integrated with the existing codebase and tests, as the outputs of a module given some inputs should still remain identical. In contrast, when rewriting, the new code is written using none of the existing codebase, possibly resulting in majorly different outputs and possibly even data flow architecture. Unlike with refactoring, existing tests might not be able to be leveraged, but it becomes much easier to make sweeping architectural changes to the codebase.

This still leaves the third option – ignoring the issues. Fairbanks points out that ignoring issues simply means that they will have to be dealt with at a later date, and contribute to '*techinical debt*' – a term coined by Ward Cunningham in [2] to help explain why otherwise working code may need to be refactored or rewritten, and has since turned into its own area of academic research, as well as a major focus of industry. Some examples of activities that "accrue" technical debt are; a lack of documentation, implementing sub-optimal algorithms and a lack of testing. [3] notes that technical debt has a number of similarities to financial debt, in that there can be advantages and disadvantages to accruing the debt. One such advantage, is that the codebase can be shipped without being entirely complete, and may indeed reach functional completeness in a shorter time than if the technical debt was not accrued. Some potential disadvantages, however, include faults in the system, increased maintainance and extensibility effort, as well as increased time onboarding new members of staff.

Technical debt, then, is clearly an area that needs to be managed over the course of a programming project, even when rewriting or refactoring. [8] notes that whilst rewriting or reworking an existing codebase can reduce or eliminate technical debt, the project also risks accumulating additional debt if not correctly managed. To help with management, [8] suggests adopting a development process that includes regularly reviewing expected and actual progress, whilst [3] encourages regular internal and external documentation, as well as maintaining an index of prioritised debts.

The SPIIR pipeline uses a group of IIR (infinite impulse response) filters with time delays to approximate a matched filter. [9] states that the output of the ith IIR filter can be expressed with the equation:

$$y_k^i = a_1^i y_{k-1}^i + b_0^i x_{k-d_i}, \tag{3}$$

where a_1^i and b_0^i are coefficients, k is time in a discrete form and x_{k-d_i} denotes input with some time delay d_i. After summing the output of the filters, the resulting signal undergoes coherent post-processing to determine the likelihood of an event having occurred.

Coherent post-processing was introducted in [7] as an alternative to coincidence post-processing. [7] states that the multi-detector maximum log likelihood ratio to be equal to the coherent signal to noise ratio ρ_c^2, which can be expressed as:

$$\rho_c^2 = \underset{max\{A_{jk},\theta,t_c,\alpha,\delta\}}{\ln \mathcal{L}_{NW}}, \tag{4}$$

where A_{jk} describes the relative inclination of the detector to the source, θ is the mass of the source, α and β are the sky directions of the source and $\mathcal{L}_{NW}$ is the network log likelihood network.

Whilst SPIIR's coherent search currently only supports the use of two or three detectors, [7] estimates the computational cost of the coherent search to be:

$$O(2N_d^3 N_m N_p), \tag{5}$$

where N_d is the number of detectors, N_m is the number of sets of IIR filters, and N_p is the number of sky locations. [9] discusses a number of optimizations made to the pipeline, including in sections of the coherent post-processing, but only reduced constant factors and not the computational cost of the overall process. As such, as more detectors are introduced, the computational time for coherent search increases cubically.

An interesting parallel can, however, be made to sorting, a very widely studied area of computer science. Comparison sorts are known to have a lower bound of $\Omega(n \log n)$ number of comparison operations [12] which can be increased depending on the algorithmic inputs, however when implemented on a distributed network, the number of comparators per thread can be reduced to $O(\log^2 n)$ using a sorting network, which uses a fixed set of comparisons on the order of $O(n \log^2 n)$ [13]. By measuring the total number of comparitors, it would appear that the distributed algorithm is less efficient, however [13] notes that the distributed algorithm can sort more keys per second than an optimal $O(n \log n)$ algorithm running on a single thread.

The SPIIR coherent search is distributed on a number of GPUs using CUDA [9], however the computational cost in [7] is computed as being on the order of the number of computations, not the number of computation per thread. It is therefore possible that the computational cost per thread is different to the overall computational cost, and thus as detectors are added using the existing algorithm, the growth of the computational runtime may not be cubic.

3 Methods

The process for development will follow the suggested method in [8] to minimize technical debt. The gravitational wave research group at the University of Western Australia meets once a week to present progress reports and submit the planned progress for the next week. By tracking the difference between planned progress and the actual progress each week, the level of technical debt that the project is incurring can be roughly determined. Another method by which technical debt can be minimized is through correct testing [3]. Unit tests will be created for the coherent search function to determine the correct outputs using the existing algorithm, and as components are refactored, the tests will be used to ensure that there are no regressions. Internal and external documentation shall also be created and maintained throughout the project as per [3], with internal documentation being comments on the function of code, and external documentation being the methods stated in the final report.

Development for the new pipeline will be performed on the OzStar cluster at Swinburne University of Technology [17]. The OzStar cluster already contains a set of sample data to run the pipeline on, as well as the tools for running the pipeline. As such, all testing will be performed on the cluster using the existing sample data.

Validation of the correctness of the eventual coherent search algorithm will be an important factor to show that there are no behavioural regressions in the course of the project. Validation will be performed using a copy of the existing codebase and comparing the outputs of the coherent search function for equivalent inputs. If the new algorithm raises any false positives or false negatives that were not also seen in the existing codebase, then it can be stated that the new algorithm is not correct.

This project will also measure the performance and computational cost difference between the 2-3 detector specific coherent search function and the generalised coherent search function. The difference in computational cost will be measured by an analysis of the average case of the resulting algorithm and comparing it to the average case of the original coherent search function. The performance cost will be measured by determining the runtime of coherent search and comparing it across equivalent inputs for different sized inputs.

4 Expected Findings

There are a number of expected findings from this research. First, it would be expected that the new coherent search algorithm would replicate the outputs of the existing search algorithm for the same inputs (i.e. it is valid as per section 3). Second, it is expected that no new coherent search algorithm is found, and that the generalisation of the existing algorithm results in a regression of less than 5% of the current coherent search runtime.

5 Proposed timeline

Task	Time	Description
Proposal Writing (Due 2020-04-20)	2020-03 to 2020-04	Write research proposal.
Literature Review	2020-04 to 2020-06	Investigate efficient coherent search methods
Analyse Existing Codebase	2020-04 to 2020-07	Determine expected inputs and outputs of coherent search methods and determine per-thread computational cost
Determine valid unit tests	2020-05	Determine tests that can be used to determine the correctness of coherent search algorithms
Oral Progress Report (Due 2020-05-22)	2020-05	Give report detailing progress on research
Refactor Existing Codebase	2020-06 to 2020-08	Perform changes on pipeline
Validate New Pipeline	2020-08	Compare outputs from changed pipeline to the previous version to ensure there is not a regression of results
Measure Performance Differences	2020-08	Determine performance cost of changes
Abstract	2020-08 to 2020-09	Prepare and submit research abstract
Seminar	2020-09 to 2020-10	Prepare and give seminar on research
Paper Writing	2020-09 to 2020-10	Prepare and submit paper on research

Bibliography

[1] Joel M. Weisberg, Joseph H. Taylor, and Lee A. Fowler. “Gravitational Waves from an Orbiting Pulsar”. eng. In: 245.4 (1981), pp. 74–83. ISSN: 0036-8733. URL: `https://www.jstor.org/stable/24964580`.

[2] Ward Cunningham. “The WyCash portfolio management system”. In: *ACM SIGPLAN OOPS Messenger* 4.2 (1992), pp. 29–30.

[3] Eric Allman. “Managing Technical Debt”. In: *Commun. ACM* 55.5 (May 2012), pp. 5055. ISSN: 0001-0782. DOI: `10.1145/2160718.2160733`. URL: `https://doi.org/10.1145/2160718.2160733`.

[4] Shaun Hooper et al. “Summed parallel infinite impulse response filters for low-latency detection of chirping gravitational waves”. eng. In: *Physical Review D - Particles, Fields, Gravitation and Cosmology* 86.2 (2012). ISSN: 1550-7998.

[5] E Murphy-Hill, C Parnin, and A. P Black. “How We Refactor, and How We Know It”. eng. In: *IEEE Transactions on Software Engineering* 38.1 (2012), pp. 5–18. ISSN: 0098-5589. URL: `https://ieeexplore.ieee.org/document/6112738`.

[6] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116 (6 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061102.

[7] Q. Chu. *Low-latency detection and localization of gravitational waves from compact binary coalescences.* eng. 2017.

[8] Richard E Fairley and Mary Jane Willshire. “Better Now Than Later: Managing Technical Debt in Systems Development”. eng. In: *Computer* 50.5 (2017), pp. 80–87. ISSN: 0018-9162. URL: https://ieeexplore.ieee.org/document/7924273.

[9] Xiaoyang Guo et al. “GPU-Optimised Low-Latency Online Search for Gravitational Waves from Binary Coalescences”. eng. In: vol. 2018-. EURASIP, 2018, pp. 2638–2642. ISBN: 9082797011. URL: https://ieeexplore.ieee.org/document/8553574.

[10] George Fairbanks. “Ignore, Refactor, or Rewrite”. eng. In: *IEEE Software* 36.2 (2019), pp. 133–136. ISSN: 0740-7459. URL: https://ieeexplore.ieee.org/document/8648269.

[11] *About “aLIGO”.* URL: https://www.ligo.caltech.edu/page/about-aligo.

[12] Thomas H. Cormen et al. *Introduction to algorithms.* eng. 3rd ed. MIT electrical engineering and computer science series. Cambridge, Mass: MIT Press, pp. 191–193. ISBN: 0070131430.

[13] *GPUGems Chapter 46. Improved GPU Sorting.* URL: https://developer.nvidia.com/gpugems/GPUGems2/gpugems2_chapter46.html.

[14] *LIGO Detected Gravitational Waves from Black Holes.* URL: https://www.ligo.caltech.edu/detection.

[15] *LIGO Lab: Caltech: MIT.* URL: https://www.ligo.caltech.edu/.

[16] *LIGO Scientific Collaboration.* URL: https://ligo.org/.

[17] *OzStar – Supercomputing at Swinburne University of Technology.* URL: https://supercomputing.swin.edu.au/.

[18] *Virgo Detector.* URL: http://www.virgo.infn.it/.