thesis_text.1

by Alexandar Mechev

General metrics

25,993	**3,917**	**210**	**15 min 40 sec**	**30 min 7 sec**
characters	words	sentences	reading time	speaking time

Score

99

This text scores better than 99% of all texts checked by Grammarly

Writing Issues

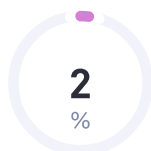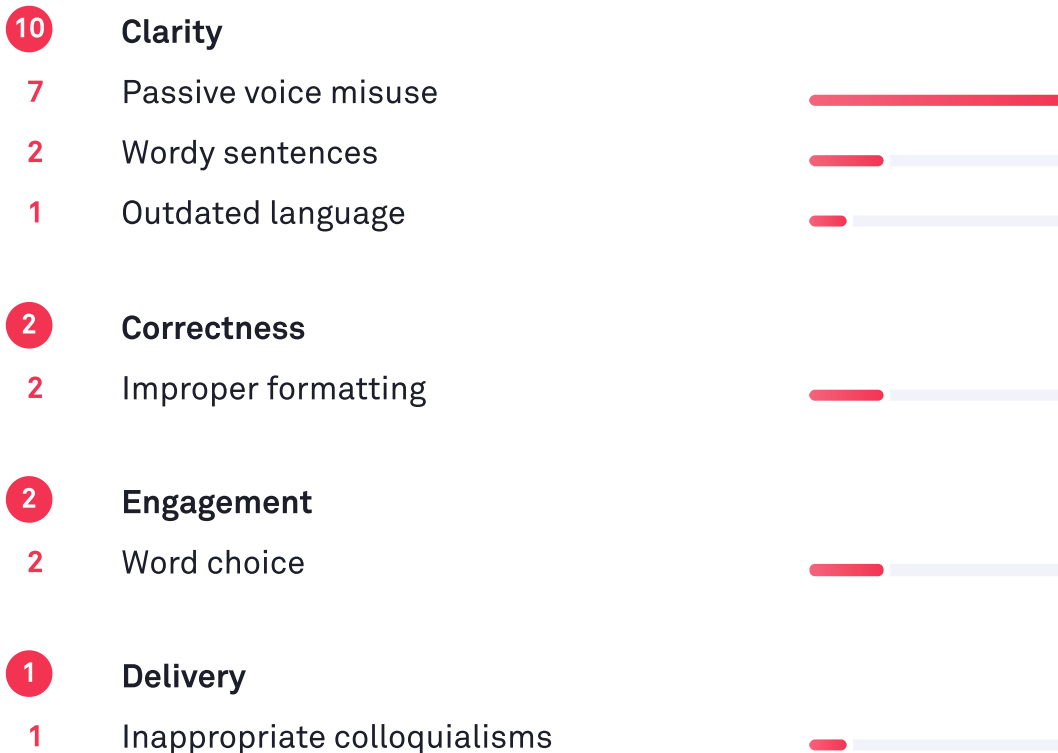| **15**	**1**	**14**
Issues left	Critical	Advanced

Plagiarism

2 %

2 sources

2% of your text matches 2 sources on the web or in archives of academic publications

Writing Issues

10   **Clarity**

7   Passive voice misuse

2   Wordy sentences

1   Outdated language

2   **Correctness**

2   Improper formatting

2   **Engagement**

2   Word choice

1   **Delivery**

1   Inappropriate colloquialisms

Unique Words

Measures vocabulary diversity by calculating the percentage of words used only once in your document

26%

unique words

Rare Words

Measures depth of vocabulary by identifying words that are not among the 5,000 most common English words.

41%

rare words

Word Length

Measures average word length

5.3

characters per word

Sentence Length

Measures average sentence length

18.7

words per sentence

thesis_text.1

Introduction

Frontmatter1whitegray

Introduction

For almost a century, scientists have used computational machines as a tool to improve scientific research. In part driven by national security concerns during the First and Second World Wars, as well as the Cold War, increasingly complex computers have been designed. Early computers were entirely designed for application-specific tasks, with a considerable drive behind them being hydrodynamics simulations for the first Hydrogen bomb. In 1945, the popular Von-NeumannNeumann:1945:FDR:1102046 architecture was developed to make Monte Carlo simulations easier to develop and to facilitate general-purpose computing. This architecture was a significant improvement over previous computers where changing the program required physically flipping switches and changing cables on the computer itself. One of the first computers built according to the von-Neumann architecture was the MANIACwood1985early computer commissioned by Los Alamos National Laboratory, seen on the left in Figure .

With the advancement of an architecture that treats code and data identically, it was possible to create more complex programs including compilers: programs that could create machine code from human-readable code. As the '50s and '60s passed, general-purpose computers were increasingly used in science. From weather dynamicslorenz1956empirical to fluid dynamicsharlow1965numerical, from chaos theory to game theory, these computers were being adopted by a wide range of scientific fields. Astronomy was, likewise, also a driving force for computational innovation. In 1953, for example, the first high-level programming language for IBM computers was developed by John Backus, a programmer frustrated with the difficulty of accurately calculating the moon's position using only machine code. John Backus' `Speedcode' was a direct predecessor of Fortranallen1981history, a language developed at IBM in the '50s and still used by the scientific community today. Another important discovery on our road was the Fast Fourier Transform (FFT), discovered by two researchers from Princeton and IBMbingham1967modern. The FFT has been described as `the most important numerical algorithm of our lifetime' and the author's personal favourite `an algorithm the whole family can use'top_10_algos. As we will soon see, Radio Astronomers quickly became part of this family.

As computers became more widely available, they became increasingly adopted by universities and research institutes. In the '70s, computers began to talk to each other over a network connection. This capability not only made scientific collaboration easier but made it possible to distribute computation across multiple sites. Moreover, the development of the integrated circuit and subsequent drop in price/performance of computers made it financially feasible for scientific institutes to purchase multiple computers dedicated to scientific research. As hardware, networks, and software matured, clusters of computers became more widely usedtel_2000. In part because of their cost-effectiveness, and potential for parallelization, computer clusters became more widely used as the '80s wound down. By then, general-purpose computing was widely adopted by the astronomical community. In the '80s several astronomical software suites have been developed, with software such as AIPSAIPS_NRAO and IRAFiraf and standards such as FITS1979fits used to this day.

The '90s continued the distributed computing trend with the appearance of commodity compute clusters, with virtual `supercomputers' being created from Commercial-Off-The-Shelf (COTS) hardware, and networking. These clusters became quickly adopted by scientists to perform simulations and data processing. Around the same time, the idea of `grids' was createdcheng1989software. The concept was that of a country, continent, or even worldwide network of hardware that can transparently handle distributed tasks, and provide researchers with a vast pool of resources. CERN pioneered grid processing to meet the computational and storage requirements of their High Energy Physics modelling and data reduction. While this infrastructure

was built initially for HEP experiments, it is also useful for other scientific projects, particularly low-frequency Radio Astronomy.

Astronomy and Computing

Since the early days of computing, the field of astronomy has embraced digitization of data acquisition and processing. Being able to store astronomical data digitally makes it possible to transfer, copy, backup, and process them efficiently. With the rapid development of CCDs, optical astronomy entered the digital age. However with the extensive availability of Analog-Digital Converters (ADCs), radio astronomy has been digital since the 1970s. By the end of the '70s, the Very Large Array (VLA) in New Mexico and the Westerbork Synthesis Radio Telescope (WSRT) had consistently been using processing pipelines, running on Digital Equipment Corporation's line of PDP, and later VAX, minicomputers. Notably, their imaging algorithms were taking advantage of the FFT developed a decade earlierclark1980_clean.

With the complete digitization of astronomical observations, over the past decade, all of astronomy has entered the big data regime. As of 2019, there are multiple planned and ongoing large-scale sky surveys across the electromagnetic spectrum, each expecting to produce multiple tens of petabytes. This breadth of data is poised to expand the frontiers of astronomy and astrophysics and allow us to study and understand various phenomena in more detail.

The longest wavelength of the spectrum accessible to Earth observatories lies in the Megahertz range, starting at 10 MHz, up to 300 MHz. This regime corresponds to wavelengths of 30 meters up to 1 meter. In astronomy, this range is termed the low-frequency or meter-wave regime. These wavelengths help uncover physical phenomena invisible to telescopes in the X-ray, Visible, Infrared, or Microwave. In particular, long-wavelengths can be used to study supermassive black holes, galaxy formation and evolution, magneto-hydrodynamics, solar physics, radio spectroscopy, and many more science cases. Additionally, data in this domain can complement other telescopes in multi-wavelength studies.

Photons provide us with rather few independent properties. We can use them to record the wavelength, the direction, the intensity, and the polarization of a distant source. We can, moreover, record the change of those properties with time. Astronomers need to measure properties accurately and use the data to model distant sources better and validate or reject astronomical theories. The accuracy of these models, or the rejection power of our observations, depends critically on how accurately we can measure the four properties listed above.

Astronomical observations in the long-wavelength regime have always been at the mercy of the diffraction limit, an effect that relates the wavelength of light, the diameter of the aperture, and angular resolution obtained with that aperture. The angular resolution of a telescope determines how accurately the direction of an incoming photon is determined. Unfortunately, the diffraction limit dictates that the angular resolution of a telescope with a fixed aperture decreases inversely proportional to the wavelength observed. For example, if one takes a telescope at 100MHz and one at 10GHz, the 100MHz telescope

would need to have 100 times the radius of its higher frequency counterpart in order to reach the same angular resolution. In other words, for a low-frequency telescope (at 100 MHz) to match the 100-m Effelsberg telescope (at 10GHz), it would need a dish with a diameter of 10 kilometers. Constructing, and operating a telescope of that size is currently outside our engineering capabilities, and thus low-frequency astronomers have developed a method to synthesize a telescope aperture of arbitrary size, termed `Aperture Synthesis.'

Aperture Synthesis

Aperture synthesis is the practice of combining the signal of multiple antennas to produce data with the angular resolution of a much larger antenna, as seen in Figure. More specifically, the maximum angular resolution achievable is related to the distance between the furthest two antennas. This technique is used in a wide wavelength range, from the near- and mid-infrared (VLTI), sub-millimeter (ALMA) and radio wavelengths (VLA, GMRT, LWA).

Aperture synthesis can be used to increase the angular resolution of an array of radio telescopes; however, the resulting data requires significant post-

processing. A single telescope operates in the `image' domain, meaning the data collected by it is directly related to an area in the sky. Conversely, an array of telescopes records waveforms between pairs of antennas. To obtain a map of the sources on the sky, this data needs to be transformed. The equation relating the data recorded by these arrays and the `true' sky distribution is the Radio Interferometry Measurement Equation, RIME. This equation describes propagation effects from the source to two antennas, and, with effects towards antenna and effects towards antenna . Each effect is described by a 2x2 `Jones' matrix describing the transformation of the original signal. This formulation is shown in Equation . When expressed in terms of the directions in the sky (,), and a continuous sky model, , the measurement equation becomes Equation .

Equation 1 neatly separates the direction independent terms (and) seen by antenna 1 and 1, and the direction-dependent effects that correspond to directions p and q inside the integral. This formalism neatly separates the direction-independent and dependent effects. A comparison between equations 1 and 2 shows the similarity between the RIME and the Fourier Transform. Specifically, l represents the sky brightness, B represents our directions p and q, and the transformed function f is the visibilities () measured by the telescope. This formalism separates the direction-dependent and independent effects, and further shows how we can use Fourier transforms to obtain a model of the radio sources. As Fourier Transforms are computationally expensive, an efficient solution is to use the Fast Fourier Transform algorithm mentioned above.

In this work, we will discuss the technical challenges of creating radio images of astronomical sources and our solutions to these challenges. In the following section, we aim to introduce LOFAR, the European Low-Frequency Array, the data sizes and processing challenges that come with LOFAR data as well as our solutions to these challenges. We will conclude with the scientific results this work has led to, as well as suggestions for future large-scale astronomical projects.

LOFAR

LOFAR is a large low-frequency radio telescope centered near Dwingeloo, Drenthe, in the Netherlands. LOFAR has thousands of Dutch antennas grouped

in Core (near Dwingeloo) and Remote stations. LOFAR also has International stations across Europe, spanning from Ireland to Estonia, Sweden to Italy. These international stations make it possible to create images of radio sources with a similar angular resolution to leading higher frequency telescopes. Much like these telescopes, LOFAR was also designed to support a variety of science cases, from large scale broadband studies to spectroscopy and transient detection.

LOFAR stores its broadband data at one of several Long-Term Archive locations. These locations store the data on tape, due to its large size and infrequent access. Typical broadband observations are up to 16TB in size, which can drop down to 10TB with compression. While individual researchers use this data to study their object of interest, the majority of the broadband data will be imaged to produce the LOFAR Two-Meter Sky Survey (LoTSS).

LoTSS

The LOFAR Two-Meter Sky Survey, LoTSS, is an ambitious project to map the Northern Radio sky at low frequencies, namely 120-168 MHz. Expected to produce more than 3000 8-hour observations, LoTSS will create radio maps with sensitivity below 100 Jy/beam. This survey will help study supermassive black holes and their impact on galaxy formation in the early Universe. Additionally, understanding the formation and evolution of galactic clusters and the interaction of galaxies within these clusters will be made possible with this low-frequency data. Furthermore, the survey will enable us to study star formation in nearby and distant galaxies and galactic sources such as

supernova remnants. Finally, LoTSS will help study and discover patterns in the large-scale structure of the Universe. The observations produced by LoTSS will total more than 30 petabytes and require extensive processing before completing the survey.

Processing Requirements

With its 3000+ observations, the LoTSS project requires a large amount of processing, bandwidth, and storage infrastructure in order to complete its scientific goals within the survey timespan. The total size of raw data is more than 30 petabytes, while the total size of the finished products will be on the order of 10s of terabytes. Furthermore, moving all the raw data to a processing facility is limited by the bandwidth of the connection between the archive site and the processing facility. Finally, each data set requires 500 core-hours for the DI pipeline and roughly 3000 core-hours for the DD pipeline. In total, this means that the LoTSS project will take more than 10 million core-hours to produce scientific results, assuming no re-processing of data.

In addition to the raw hardware requirements, an ambitious project as such needs to be able to track the status and location of data products, automate processing and make results readily available. As multiple locations store LOFAR data, it is also essential that the framework tasked with processing LoTSS data is portable and can run independently of the infrastructure details.

SURFsara

One of the archive locations storing LOFAR data is SURFsara at the Amsterdam Science Park. Aside from an extensive storage archive, SURFsara also supports several clusters, including the Gina cluster, part of the Dutch Grid infrastructure. Grid computing is a non-interactive application-oriented computational paradigm for distributed computing where a 'grid' consists of a large pool of nodes where users can submit batch jobs. A grid can consist of one cluster or groups of clusters at one or multiple geographical locations, connected with high-speed links and a standard job management interface. Using this interface, users can scale out their projects, given that their processing is massively parallel. Computational resources on such a platform are granted based on a scientific proposal and are used freely across the Grid, while jobs are scheduled based on the job requirements and the current resource availability of the grid nodes. This processing paradigm is perfect for extensive grid-search simulations, but also the first steps of LOFAR processing. Furthermore, the high-speed connection to the LOFAR archive and available storage makes SURFsara a logical location to orchestrate large scale LOFAR projects and distribute processed data.

LoTSS Processing

The observations produced by LoTSS will total more than 30 petabytes and require extensive processing before completing the survey. Each observation is stored as a set of 244 individual files spanning frequency space from 120MHz to 168MHz. Each of these files, named a Subband, is a CASA Measurement set and is identified by its three-digit Subband number, starting from 000. Each Subband, thus, contains a sub-sample of the data in frequency space, stored

at a resolution of 1 second and 12.2 kHz per sample. While this high-resolution data is useful for some science cases, our processing algorithms scale with data size, and thus it is necessary to average our data in order to complete the LoTSS processing within the project's time-frame.

In order to create an image from an archived data set, the data needs to be staged, retrieved, and processed. Staging the data refers to sending a request to the archive site to move the data from tape to disk. Once all the data is on disk ('staged'), it is ready to be transferred from the storage to the processing cluster. On this cluster, a science-ready image is produced by processing the raw data through two pipelines. The first pipeline, Direction Independent Calibration pipeline removes artifacts created by `direction independent' effects, i.e., effects that are constant across the field of interest. This pipeline is followed by the Direction Dependent Calibration pipeline, which removes effects that change within the field of view.

The Direction Independent Calibration pipeline (DI pipeline) consists of two main stages. The first stage is calibration on the calibrator, which uses a short observation of a bright calibration source to determine systematic effects that are independent of the direction of the pointing. The solutions obtained from this step can be applied to the scientific target, improving the data quality. The second step of the DI pipeline is the calibration of the target field against a sky model produced by a previous survey. This calibration determines the gain parameters of all antennas; however it does not correct for effects that vary across the field of view.

In order to create a high fidelity radio image, we need to correct for effects that not only change in time but also across the field of view. These effects, such as the ionosphere or the beam response, can be modelled and removed, and their

removal is the responsibility of the Direction Dependent pipeline (DD pipeline). Upon successful completion, the DD pipeline produces a radio image to be used for further scientific studies.

There are a few software packages used to process LOFAR data, typically used in a series of steps creating a processing pipeline. The LOFAR processing pipeline steps use the software, each step encoding the processing parameters in a parameter-set (parset). A pipeline is defined by the list of steps, and the parameters of each step concatenated together into a parset file. The LoTSS DI pipeline, prefactor contains a set of scripts used to remove direction independent effects from LOFAR data. Many of the prefactor steps can be executed on the data in parallel: each Subband can be processed independently. Because of the large amount of data, the best architecture for these steps is a cluster of isolated machines with dedicated disks and a high-speed connection to the data. Later we will show the benefit of automating these steps on the Dutch Grid infrastructure.

The life of a data set

A broadband LOFAR data set has to be processed by the Direction Independent and Direction Dependent pipelines. In this section, we will show the progress of one observation from raw data to a final scientific image. Because of the large size of the data involved, we only use half of the bandwidth, from 132.2MHz to 156.1MHz.

Figure 1 shows an image of the data downloaded from the LOFAR Long Term Archive. There have been no corrections done to this data, resulting in a large amplitude offset (see scale bar below figure) and distinct artifacts around bright[1] sources. In order to decrease these artifacts and calibrate the brightness of the sources, we use calibration data from a known bright radio source and apply these solutions to our data. The resulting data produces Figure 1. We remove Radio Frequency Interference (caused by manmade sources) from our data and correct for bright[2] off-axis radio sources. Finally, we use a model of the radio sky obtained by a previous survey to calibrate all our direction independent gain parameters. The resulting data produces Figure 1.

After the Direction-Independent calibration, we perform correction for Direction Dependent effects. This correction is done by the ddf-pipeline scripts using the DDFacet and killMS software packages. To produce the images, we use the `tier1-jul2018' parameters with bootstrapping turned off. This processing produces two images, a high resolution and a low-resolution image of our data. These are shown in Figures 5 and 7[3] respectively.

Problem Statement and Research Questions

Radio Astronomy data sets are too large to process in bulk on individual workstations and often strain the resources of small clusters at universities and other institutions. This limitation in resources requires high throughput processing capability, and automation in order to serve processed data in bulk to astronomers. The LOFAR radio telescope acquires data at a rate of roughly a terabyte per hour. This data is stored in a Long Term Archive as it can serve multiple science cases. Our goal is to create the tools for scientists to be able to process this data with their scripts and software efficiently. As such, these tools need to be fast, easy to use, general, and scalable.

[4em]8emProblem Statement: How can we efficiently process broadband LOFAR data in a generic way?

Research Question 1

To efficiently process LOFAR data, we need to take advantage of the data level parallelism of the early processing steps, each of which can be parallelized by a factor of 244. Because of the large sizes of LOFAR observations, transfer time can be comparable to the processing time. To minimize transfer time, we need to study how to deploy processing pipelines at the LTA storage sites. In this research question, we ask how to best build a framework for a massively

distributed shared platform for LOFAR, and how to deploy LOFAR processing in parallel when possible.

[4em]8emResearch Question 1: How can we use a distributed shared infrastructure for efficient LOFAR data processing?

Research Question 2

Once we have determined the utility of distributed processing for the LOFAR case, we ask how to automate complex LOFAR workflows. The LOFAR radio telescope serves multiple science cases, each of which is served by a multi-step pipeline with a broad set of parameters. Running an entire pipeline on a single computational node is inefficient; thus, a workflow orchestration software is needed to parallelize the appropriate steps. In this research question, we ask how to build software to efficiently integrate scientific pipelines with a massively parallel distributed processing platform.

[4em]8emResearch Question 2: How can we build software to effortlessly accelerate complex pipelines for Radio Astronomy?

Research Question 3

Once complex pipelines can be executed on a distributed environment, researchers may ask whether the software concurrently running on hundreds of

systems is running optimally. Manually monitoring automated runs is not possible, hence software is needed to collect this performance data per pipeline step. Furthermore, some of the processing parameters for LOFAR pipelines result in large data sets. In order to serve LOFAR processing to the scientific community, we need to understand how our resource usage scales with each of the processing parameters. We ask whether it is possible to integrate monitoring tools to our processing framework in a way that we can transparently collect performance data along with scientific processing.

[4em]8emResearch Question 3: Can we automatically collect performance information during massively distributed processing and predict run times for future data sets?

Contributions

Software Contributions

For this work, we built several software packages to define, launch, and orchestrate jobs on a high throughput cluster. The software packages built are GRID_LRT, GRID_PiCaS_Launcher, and AGLOW. They are available on GitHub, and their documentation is hosted on ReadTheDocs.[5]

Statement of Originality

I hereby certify that the content of this thesis is my own original work, consisting of six manuscripts submitted to peer-reviewed journals and conferences. This thesis and the works therein have not been submitted to any other degree program. Finally, I certify that the intellectual content in this work and all software referenced therein are of my own work unless explicitly cited otherwise and that all assistance in compiling this work has been adequately acknowledged.

Publications

Chapter 1 describes our first attempts to do large-scale distributed LOFAR processing on a shared infrastructure. We detail our successes with the LOFAR Radio Recombination Lines and Pre-Processing Pipelines, our software set-up as well as the limitations and future uses of this platform.

Chapter 2 is our implementation for portable LOFAR processing on a massive scale. We show early results encapsulating LOFAR processing pipelines, and discuss future uses on other clusters. It is based on mechev17.

Chapter 3 discusses collecting and studying detailed performance statistics from automated LOFAR processing. It is based on mechev_pipeline_collector.

Chapter 4 describes the capabilities of the initial workflow manager for automatic processing of LOFAR SKSP data. It shows several different scientific

workflows and is based [12] on ieee_AGLOW.

Chapter 5 describes a parametric model of resource usage for the LOFAR prefactor pipeline. We discuss difficulties that may arise from scaling LOFAR data, as well as the utility of our modelling method for SKA-size data. This chapter is based on [13,14] parameteric_model.

Finally, Chapter 6 demonstrates the flexibility of our AGLOW software by implementing a full Continuous Integration pipeline for LOFAR software containers and LOFAR scientific pipelines. This pipeline can verify software pipelines against a test data set and can be used to track the data quality of data products as the processing software evolves. This work is submitted [15] to the IEEE e-Science Conference 2019.

1.	~~bright~~ → light	Word Choice	Engagement
2.	~~bright~~ → light	Word Choice	Engagement
3.	~~7~~ → seven	Improper Formatting	Correctness
4.	generically process broadband LOFAR data	Wordy Sentences	Clarity
5.	*is hosted*	Passive Voice Misuse	Clarity
6.	~~hereby~~ → at this moment, now, as a result of this, with this	Outdated Language	Clarity
7.	*my*	Inappropriate Colloquialisms	Delivery
8.	~~own~~	Wordy Sentences	Clarity
9.	*been submitted*	Passive Voice Misuse	Clarity
10.	*is based*	Passive Voice Misuse	Clarity
11.	*is based*	Passive Voice Misuse	Clarity
12.	*is based*	Passive Voice Misuse	Clarity
13.	~~is based~~ → is based	Improper Formatting	Correctness
14.	*is based*	Passive Voice Misuse	Clarity
15.	*is submitted*	Passive Voice Misuse	Clarity
16.	*The angular resolution of a telescope determines how*	Optics and Telescopes - web.njit.edu https://web.njit.edu/~cao/Phys320_L6.pdf	Originality
17.	*Because of the large amount of data, the*	Innovative Design of Marine Resources Database System Based on Fuzzy Cluster Analysis	Originality