2019

High-Speed Distributed Data Process of Photometric Astronomical Data

Paul Doyle
Technological University Dublin, paul.doyle@tudublin.ie

High-Speed Distributed Data Process of Photometric Astronomical Data

Paul Doyle
Technological University Dublin
Paul.doyle@dit.ie

Abstract

Since the 1970s the CCD has been the principle method of measuring flux to calculate the apparent magnitude of celestial objects within astronomical photometry. Each CCD image must be digitally cleaned and calibrated prior to its use. As data archives increase in size to Petabytes, the data processing challenge requires image processing techniques to continue to exceed the rate of data capture.

This paper describes NIMBUS, a rapidly scalable, failure resilient distributed network architecture capable of processing CCD image data at a rate of hundreds of Terabytes per day. NIMBUS is implemented using a decentralized web queue to control the compression of data, the uploading of data to distributed web servers, and the creation of web messages to identify the location of the processed data. This paper demonstrates the horizontal scalability of NIMBUS which has demonstrated a processing rate of 192 Terabytes per day with clear indications that higher processing rates are possible.

1. Introduction

Photometry is defined as the branch of science that deals with measuring the intensity of the electromagnetic radiation or flux of a celestial object [1]. This science can be traced as far back as 130 BC to Hipparchus [2], who devised the first measurement system categorizing objects' apparent brightness from brightest to faintest. Since their first introduction to astronomy, CCDs (charge coupled devices) [3] have received considerable attention from the astronomical community [4] and revolutionized this field of science, providing levels of sensitivity beyond the capability of photographic plates, extending the detection range into the infrared spectrum, providing immediate results with a linear response, and allowing for software to compensate for CCD defects.

When a CCD digital image is recorded, it contains a digital count of the electrical charge of each of the pixels on the CCD array. The electrical charge per cell is converted to a digital pixel value by first transferring the charge to a corner of the array and then using an analogue-to-digital conversion to record its value. This digital image contains a number of different artefacts, introduced by the process of recording and reading, which must be removed. These and other sources of noise require a computation operation to be performed across the image pixels in order to quantify the signal-to-noise ratio. For each image taken, there is a computational overhead incurred before scientific analysis can be performed. As the number of images increases, so does this computation cost.

In order to address the issue of cleaning and preparing terabytes or even petabytes of CCD-based astronomical photometry images per day, a distributed elastic cloud based computing model, to perform standard image data processing, is required. A processing pipeline has been designed, which demonstrates a working CCD image reduction pipeline, and which incorporates an elastic data processing model. Resources can join or leave a swarm of distributed computing workers which communicate via a distributed web-based messaging queue. Furthermore, taking advantage of the fact that CCD images can be cleaned in isolation from each other, image data is distributed for parallel processing to eliminate sequential image processing bottlenecks.

2. Background

To ensure clarity of the terms used within the context of astronomical photometry, the following definitions are provided for reference.

2.1. Apparent Magnitude

The apparent magnitude of a source is based on its apparent brightness as seen on Earth, adjusted for the atmosphere. The brighter the source appears, the lower the apparent magnitude value.

.

2.2. Absolute Magnitude

The absolute magnitude is a measure of a star's brightness as seen from a distance of 10 parsecs (32.6 light years) from the observer. The absolute magnitude of an object can be calculated given the apparent magnitude and luminosity distance, which is measured in parsecs.

2.3. Instrumental Magnitude

The instrumental magnitude is an uncalibrated measure of the apparent magnitude of an object which is only used for comparison with other magnitude values on the same image.

2.4. Luminosity

The Luminosity of an object is a measure of the total energy emitted by a star or other celestial body per unit of time and is independent of distance and is measured in watts. The luminosity of a star is related to temperature and the radius of the star.

2.5. Flux

The flux is a measure of the apparent brightness of a source which is inversely proportional to the square of the distance and is measured in watts per square meter. How bright a source appears is based on the distance from the object and the luminosity of the object.

2.1. Photometry

What is being measured during the photometric process is the apparent brightness (or apparent magnitude) of an object and not its actual magnitude. To highlight the difference in actual versus apparent magnitude, consider the apparent brightness of a 40-watt bulb as seen from 10 meters versus 10 kilometers. In both cases the light bulb retains the same luminosity, but the apparent brightness is dramatically different due to the distance between the observer and the light bulb. Figure 1 provides examples of the apparent brightness of well-known objects for reference.

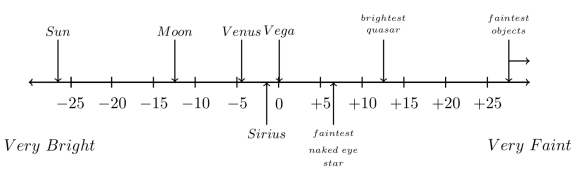


Figure 1. Apparent brightness using magnitude system

2.2. Data Processing Challenge

When a single CCD detector records an image, the size of the digital image is usually dependent on the number of pixels on the device and the number of bytes used to store the value for the pixel. The size of the dataset, generated by an array of CCDs, is dependent on the size of each digital image, the image capture rate (ranging from milliseconds to minutes), the time period over which images are taken, and the number of CCDs in the array. While a small telescope may use a single CCD, larger telescopes may employ an array of CCDs, and robotic telescope farms may use an array of telescopes, each with its own CCD array. With megapixel CCD arrays already in use and with frame rates per second increasing, the tsunami of data production is already beginning. Indeed, Graham [5] refers to the data avalanche, tsunami and explosion of data with telescopes generating petabytes of data on a nightly basis in the near future. Ferguson et al [6], looking to the next decade of data reduction and analysis, sees the three major challenges as follows:

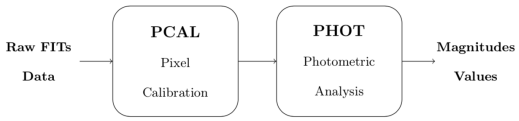
- Data rates growing rapidly as CPU processing rates level off.
- Industry trends in computing hardware leading to major changes in astronomical algorithms and software.
- Computationally demanding analysis techniques becoming more essential with increasing pressure on computing resource.

It is only when considering the combination of these challenges that the extent of the problem of large dataset production and processing can be fully appreciated. The factors which contribute to large dataset generation are summarized as follows.

- *Resolution*: Number of pixels captured per image.
- *Capture Rate*: Number of images taken per second.
- *Capture Period*: The length of time over which images can be taken.
- *Device Count*: The number of capture devices operating at one time.
- *Capacity*: Ability to read and store data generated.

Figure 2 provides a summary of the operations performed by the NIMBUS pipeline which stops short of performing any actual science on extracted magnitude values from images. To ensure that the ability to analyze magnitude values can be done in

real-time, PCAL (pixel calibration function) and PHOT (photometric analysis function) should process data at the same rate as data is being generated and supplied to the pipeline. Just-in-time processing must be completed within a twenty-four hour period which would mean data processing must be no less than three times slower than data acquisition before a bottleneck is created, assuming an eight hour image capture period per day.



Processing rate $\geq$ *Data Input rate* (Real-time Processing)

Processing rate $\geq$ *Data Input rate* /3 (Just-in-time Processing)

PCAL	Bias reduction, dark current removal, flat fielding for selected pixel regions around objects of interest and pixel calibration
PHOT	Sky background removal, centroid calculation, and aperture photometry for magnitude estimation

Figure 2 Overview of calibration and photometric analysis on RAW CCD images within NIMBUS

3. Astronomical Data Processing

CCD imaging systems have well-understood reduction processing steps designed to calibrate a raw image. The accuracy of photometric measurement is based on these well-defined cleaning techniques which are discussed in more detail in this section.

Aperture-photometry techniques provide a clear process for the estimation of apparent magnitude of objects, using a standard reference scale. Finding the centre of objects, estimating the sky background and calculating the flux of an object for a range of aperture sizes are all well-defined procedures.

3.1. Standard Reduction Techniques

When a CCD instrument is used, the recorded file output stored on the computer contains a measure of the source signal in addition to unwanted random noise for various sources. The noise introduces an error into the measurement. In this section the sources of noise in CCD image reading are described along with the techniques used to deal with them. These techniques are incorporated into the NIMBUS system.

3.2. Noise Sources

Noise is the introduction of unwanted variations to the image, distorting the readings in some way. If a CCD pixel has a well depth of 100,000 electrons (the total amount of charge that can be stored in a pixel) and the average noise can be determined to be approximately 40 electrons per pixel then the SNR (Signal to Noise Ratio) is 100,000/40 or 2,500. If the amount of noise can be reduced, then the SNR is increased. The process of reducing the level of noise in an image is critical to performing high precision photometry. The standard equation for SNR is often unofficially referred to as the CCD Equation [7].

The main contributions to noise within a CCD are dark current, pixel non-uniformity, read noise, charge transfer efficiency and cosmic rays [8].

3.3. Bias and Dark Frames

A bias frame has a dark frame with an exposure time of zero and is a measure a pixel's read-noise. This value is usually caused by a low-level spatial variation caused by the on-chip CCD amplifiers. Read-noise from a CCD is an additive noise source that is introduced during the pixel read process which does not vary with exposure time. This is a systematic noise source which must be removed.

A master bias frame is created through the combination of multiple bias frames using the average pixel values seen across each frames as shown in Figure 3. An average value is considered acceptable given that the CCD should not be exposed to cosmic rays since there was no exposure of the CCD sensors. The master bias frame can then be used in cleaning data images by subtracting the master bias value for each pixel.

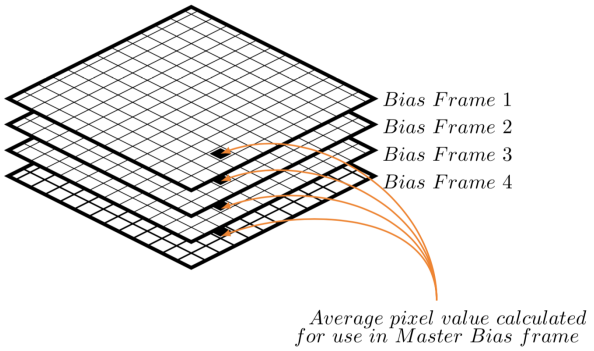


Figure 3. Master bias frame created using multiple bias frames

3.4. Flat Fielding

A flat field image is taken when the CCD has been evenly illuminated by a light source. Flat fielding is used to compensate for differences in pixel-to-pixel variations of the CCD response to illumination when the same amount and spectrum of light is illuminated across each pixel on the CCD. This technique also helps remove the effects of dust which can cause dark spots on an image and uneven illumination caused by vignetting in the optical system.

The flat field value is used to modify image pixel values to account for these variations. There are varying opinions on the best method to create a good flat field image, such as the use of an illuminated painted screen inside the telescope dome [9]. Howell provides an excellent overview of many of these approaches [3].

3.5. Image reduction

The process of characterizing the level of noise within a CCD pixel is well documented [8]. Using the estimation techniques identified, a basic image calibration process designed to reduce noise from the CCD raw images, a necessary process in preparing the CCD images for analysis, can be summarized as follows.

A pixel value on a CCD frame has the bias and dark current removed and is then adjusted for the calculated responsiveness of the pixel relative to all other pixels. This calculation must be performed on all pixels which are ultimately used in the calculation of magnitude values. A new version of the image can then be created containing the calibrated pixel values. The creation of the master bias, flat field or dark frames is often done once for each night of observation and are then used in the calibration of pixels for that night.

3.6. Photometry using CCD images

The general steps in classical photometry using a cleaned digital image created using the image reduction techniques described are usually identified as follows [10] [11].

- Image centering, the process of finding the center of an object.
- Estimation of the sky background for the purpose of removing it from the flux intensity value.
- Flux value intensity calculation for an object for a specific aperture size.
- Magnitude calculation for an object for a specific aperture size taking into account the sky background.

Multiple magnitudes can be generated based on variations in the software aperture size used in the calculation of the flux intensity.

3.7. Data Sources

With an understanding of CCD calibration and magnitude calculations, it is important to consider the context within which these operate. For any world-class scale project (space or ground based), significant investment is required in information technology (IT). Data products are produced, preprocessed to a predefined level, and made available to a Principal Investigator, supporting institutes or potentially to the public, either directly via download servers or via the VO [12]. For large projects, data capture, transfer, calibration and reduction, basic processing, archiving and access are considered as part of the observatory capabilities for which bespoke solutions are often implemented. Smaller institutes often capture less data due to the their relatively less capable instruments. However, whole investment in IT is still required, more modest computing resources may be sufficient.

As smaller research groups have the capacity to generate ever larger volumes of data, a gap in processing capabilities emerges. Figure 4 shows how quickly terabytes of data can be generated by high framerate and high-resolution cameras. As the pressure for data generation rates increases, there should be commensurate pressure to keep the associated IT costs in line so that smaller institutes can continue to take advantage of instrument improvements.

This research seeks to address the key question of whether a distributed model can be created when the computation to data ratio is low while allowing for tens of terabytes of data to be processed. The distributed model used in NIMBUS potentially offers a cost advantage to the smaller institute or facility, while providing a powerful processing network.

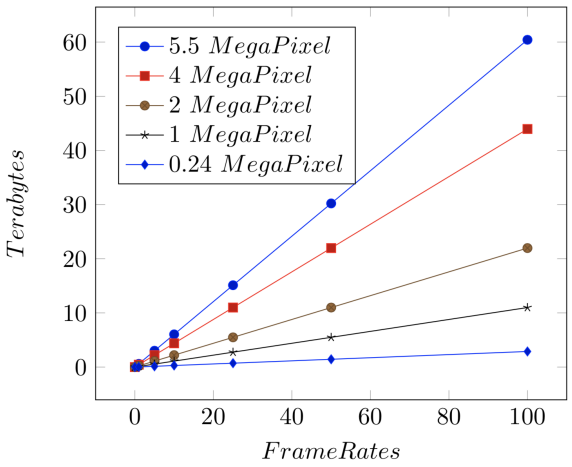


Figure 4. Data generation rates per 8hrs for varying camera resolutions

3.7. Dataset

For the purpose of testing the NIMBUS system, a dataset was provided by the Blackrock Castle Observatory (BCO), a research facility engaged in high-speed photometry research. The reference dataset contained 3262 cubed FITS files [13], each containing 10 images with each being approximately 512x512 pixels in resolution (0.7MB per image) and with the total size of the dataset being 26GB. This data was replicated to simulate a multi-terabyte data. The dataset was generated on September 22nd 2003 at Calar Alto, targeting S5 0716+71 as part of an engineering equipment test of a new hardware/software stack using an Andor CCD device.

4. Our Approach

The approaches to processing large datasets are largely dependent on the performance requirement of the task and the volume of data. It is perfectly reasonable to use a brute force approach to solving a problem when the problem is sufficiently small, or computing resources are sufficiently powerful. In these cases, results can be produced within a reasonable amount of time so there is no need to process data using any specific method other than sequential processing. As the volume of data increases to terabytes, then the traditional approaches start to incur unreasonable delays in processing time, and further thought is required to address the problem of performance and processing efficiency.

A distributed processing approach has the advantage of potentially employing large numbers of resources concurrently. To distribute the processing of data in a meaningful way, the data must be parallelized to some extent. If the data must be processed in a sequence then distributed computing may not be very relevant in that there are fewer opportunities for parallel processing. Fortunately, astronomical CCD data can be reduced in parallel once the calibration frames are provided with each image.

NIMBUS, a globally distributed pipeline, is described in this paper as an alternative approach to the data processing techniques reviewed. This requires that the images be processed in parallel with only the necessary work needing to be performed without compromising the quality of the data. Using the analysis of magnitude calculations, it can be shown that data can be safely processed in parallel with the same outcome as an equivalent sequential pipeline as is seen in some existing pipelines. The methods used to allow the NIMBUS pipeline to scale should ensure that the distribution of computing nodes can truly reach global levels and not be restricted to local network domains.

4.1. NIMBUS Architecture

The basic workflow is for a controller to instruct a data capture node to publish the address of all files in its data store and to then activate AWS EC2 nodes, which make up the global processing cloud. Each EC2 node upgrades its software when activated, by downloading the latest version of the package software with instructions on how it should operate. The node then proceeds to take messages off the SQS system, download the file named within the message and processes the file. Once results are obtained, they are written to an AWS S3 facility. Nodes can be added or removed at any time. Any work not completed is automatically reinserted into the queue for another node to take. A node can run multiple threads, the number of files downloaded can be configured, the queue which is used can be updated and the software used for processing can be updated centrally. Multiple web servers containing data can all contribute to the worker-queue, the instances can be of any size or configuration once they can run the software stack downloaded from the software distribution web server. The NIMBUS architecture is shown in Figure 5.

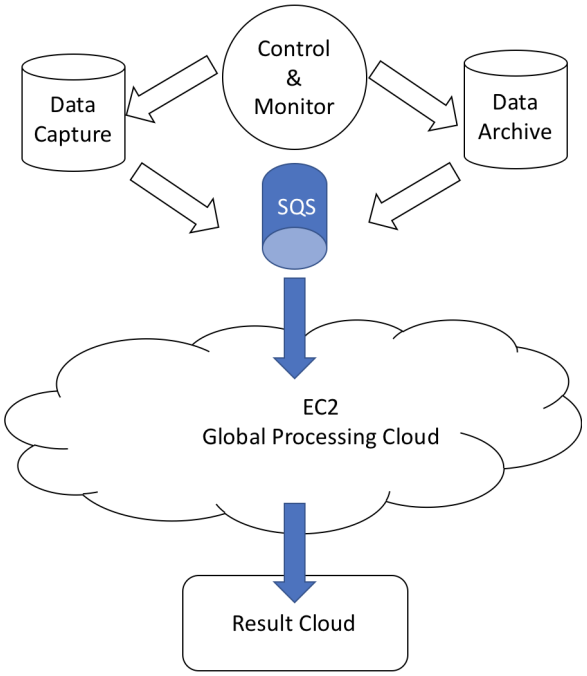


Figure 5. NIMBUS Architecture

4.2. Data Capture Cloud

The data capture cloud consists of multiple distributed telescope sites containing CCD devices which record image data to a local storage device. Lossless data compression on images is performed to reduce the bandwidth required for data transfer.

4.3 Data Archive Cloud

The data archive cloud consists of multiple distributed websites containing image datasets. Images will already be compressed and possibly reduced in size. The images are stored on fast storage disks attached to static web servers which serve http requests from the global data processing cloud. The web servers advertise files to be processed via the distributed worker queue.

4.4 Distributed Worker SQS Queues

When the worker web queue is informed of a file available for processing it stores the url of the file in a simple message which is available for worker nodes to read. The web queue ensures that only one copy of a message can be read from the queue at a time. When a worker completes its processing it permanently deletes the message. If a worker node fails to complete the processing of the image, the message will eventually reappear on the queue as per the SQS protocol. This ensures that the overall system is resilient against compute node failures

4.5 Global Processing Cloud

Worker nodes contain an initialisation boot script which installs worker sandboxes using tools downloaded from a predefined URL. These tools ensure that the work performed is configurable, both in terms of the work to be performed and web queues to listen or write to. Worker nodes within the processing cloud can be located anywhere in the world. Workers can join or leave the processing cloud at any point without impacting the overall processing pipeline.

4.6 Results Cloud

When a worker has completed its work, the resulting data file is uploaded to a distributed storage facility and a message is then written to the result queue that contains the URL for the location of the upload file. Using this queue, a processing cloud can be reconfigured to read the message queue to identify the URL of the result and to download results to a central location if required.

5. NIMBUS Pipeline

The NIMBUS pipeline, uses a public web queue to publish work to distributed computing nodes built explicitly for this pipeline which are referred to as workers. Workers are computing instances that can reside anywhere on the internet but are required to have internet access using port 80, as all services accessed are HTTP based. Each worker uses, at its core, the `acn-aphot.c` program used in the ACN Pipeline [14] which runs in single step mode. For this pipeline, the BCO dataset is also used and replicated so that there are multiple terabytes of data available for processing.

5.1. Experimental Methodology

There are six components central to this pipeline; data capture and staging, serving archive data, distributed worker queues, distributed data processing, and results storage and monitoring. Each component is required to operate continuously and asynchronously, allowing for resource utilization to be varied without interrupting the overall pipeline. While tested to a processing rate of 200 terabytes per day, the

experiments were not at the limit of possible processing rates, with the primary restriction being a lack of additional resources available. Some of the larger experiments utilized over 10,000 processing worker threads across 100 distributed servers. Table 1 summarizes the high-level experiments run on the NIMBUS pipeline.

Table 1. Experimental Objectives

Reference	Measure	Objectives
Exp-NIM1	SQS performance	Testing the read and writing times of the web message queues
Exp-NIM2	Single Instance	Determine the variables which affect the performance of the overall processing power of a single instance
Exp-NIM3	Multi-Instance	Focus on scaling the number of instances up to 100 looking for factors which could affect the scalability of the system.

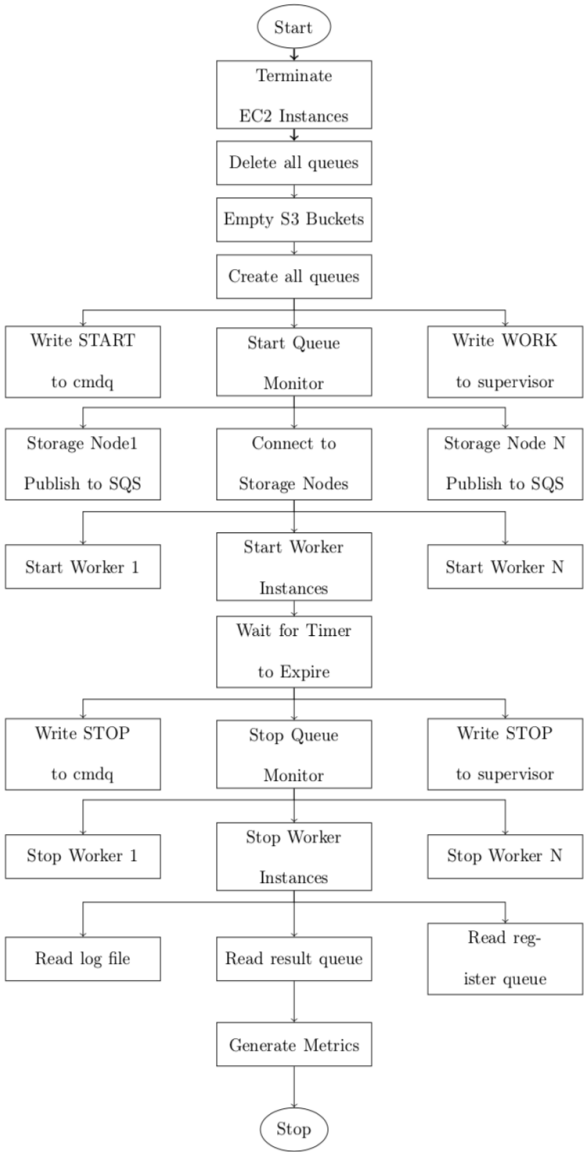


Figure 6. Experimental Control Flowchart

5.2. Experimental Control

The function of the control system is to initiate all experiments and ensure that all systems are available and functioning correctly. It is important that experiments can be compared, and to do this, the starting state must be consistent in all cases. The control system runs a Python script which tears down the experimental infrastructure and then rebuilds it before the start of the experiment. All systems must be accessible from the control system which resides on a virtual machine within the AWS cloud, running an Ubuntu instance on the EC2 service. A batch script contains the series of experiments to run, which in turn

calls a script to start and experiment. The control flowchart for the experimental protocol is shown in Figure 6.

5.3. Results and Discussion

Limits imposed on the experiments were based on limits of available resources although, where possible, indications of scaling opportunities were identified. For the pipeline to be active, a minimum of one worker is required to perform image cleaning and reduction. Multiple worker processes can run on a worker node (compute instance) which is typically a virtual AWS instance. The number of instances activated within the final experiments was 100, but the number of workers was 10,000. In some cases, multiple runs of the same experiment were performed to ensure results were repeatable. Given additional funding, additional resources could be activated.

5.4. SQS Performance

To achieve a data cleaning rate of terabytes per hour, it is essential that the queuing mechanism is able to advertise data sufficiently quickly to present work at a rate higher that the expected cleaning rate, and to ensure that work creation rates are expandable as the number of files to be cleaned increases. This requires that the storage nodes within the NIMBUS architecture can collectively create messages on the SQS worker queue at a rate of over 100 messages per second. In addition to writing messages to the queue to generate work, the architecture of the system requires that queues are also used for monitoring and obtaining the results of an experiment. Experiments were devised to determine the SQS queue read performance.

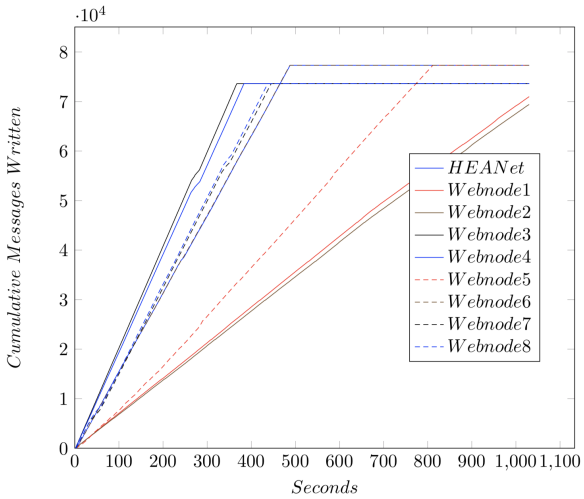


Figure 7. NIM1-1 Message write performance

To determine the performance of the SQS distributed queue two experiments were run to determine the capability of message reading and writing.

Messages written to the queue over time from each storage node are shown in Figure 7 and indicate the write rate is linear, although there are differences intrinsic to the storage itself. This is likely to do with network and processing power on the individual storage nodes.

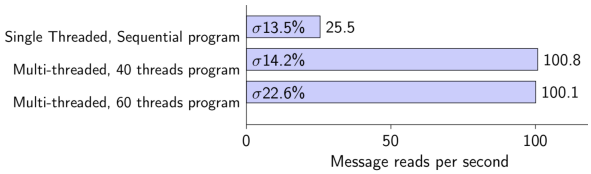


Figure 8. Exp:NIM1-2 Message read performance

Messages read rates from a single monitor server node, using varying levels of threads running with the standard deviation, are shown in Figure 8.

The message queuing system provides a number of advantages to the pipeline as summarized below.

- Exp:NIM1-1. Using multiple web nodes writing at the same time, the advertised rate for the pipeline is over 26TB per hour, although this is unlikely the limit as write rates were linear with the number of web nodes included.
- Exp:NIM1-2. A single node read performance for messages is similar to the single node write performance. Downloading of messages is naturally distributed for the pipeline. A limit per queue existing which is equivalent to a processing rate of 2.8 PB per hour. All that is required to overcome this is to increase the number of queues being used for reading.

5.5. Single Instance Node Performance

For this group of experiments, a variety of physical and virtual machine instances are used to look at the impact of running multiple workers on the same instance. The assumption is that if an instance is busy downloading an image, then the CPU resource is not being used. To fully utilize the CPU, additional workers can run to balance the load of the CPU over time. Workers are designed to cycle through downloading batches of files, processing them, and then uploading them. A single worker will is very unlikely to fully utilise all of the instance resources at

the same time. By increasing the number of workers, it would be reasonable to conjecture that the overall resources are being more fully used, but that there is a point beyond which the number of workers being added does not increase the performance of the instance.

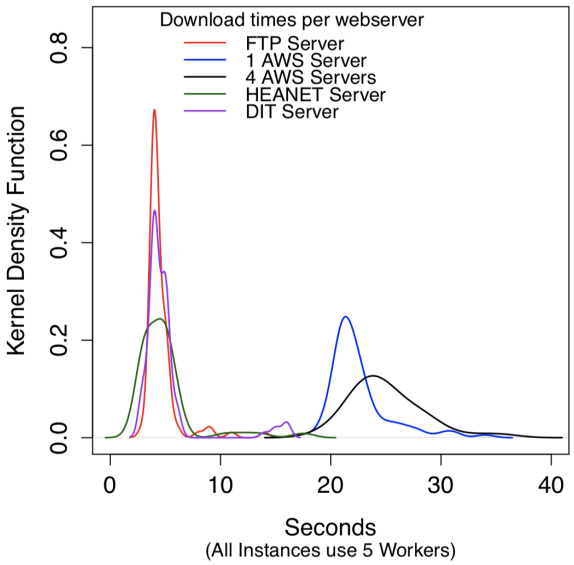


Figure 9. Exp:NIM2-1 Single Server Download Performance

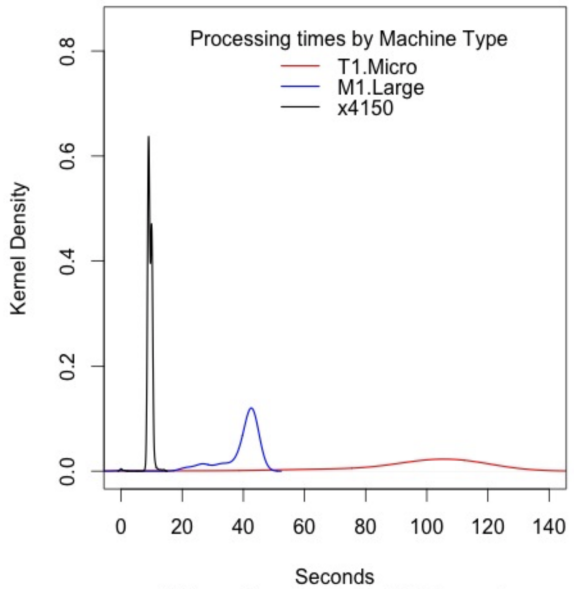


Figure 10. Exp:NIM2-2 Server Performance with 10 Workers

Figure 9 looks at the download performance of multiple webservers showing that server type can impact the download speed. By first identifying the fastest web server, a second set of experiments were conducted. Figure 10 shows the performance of different processing servers using multiple worker threads within the server configuration. By using the fastest FTP server the issue of downloads was eliminated from this processing experiment.

These experiments provide basic information regarding the performance of a single instance within the pipeline architecture.

- Exp:NIM2-1 For a single worker instance, running a single worker there is clearly a difference observable in the download times from the AWS based web servers used.
- Exp:NIM2-2. Each worker type will contain different characteristics such as CPU performance and memory size. If the number of workers is increased then, providing there are sufficient CPU resources, the processing rate will increase. The overall pipeline will therefore run faster as more powerful servers are utilized until the capacity available to download data becomes a bottleneck.

5.6. Multiple Instance Node Performance

In the final set of experiments, a large EC2 instance (C1.XLarge) was deployed running a total of 100 worker threads per server-instance, along with other smaller server instances running different worker loads as shown in Figure 11. This experiment was then run for a period of 300 seconds for the fastest server type and the processing rate was sustained for that period of time.

Using the fastest AWS server available, an experiment was run to show that the performance of a single server and 100 servers is linearly correlated. This final experiment was designed to show whether the addition of more servers could sustain an increase in the file processing rate.

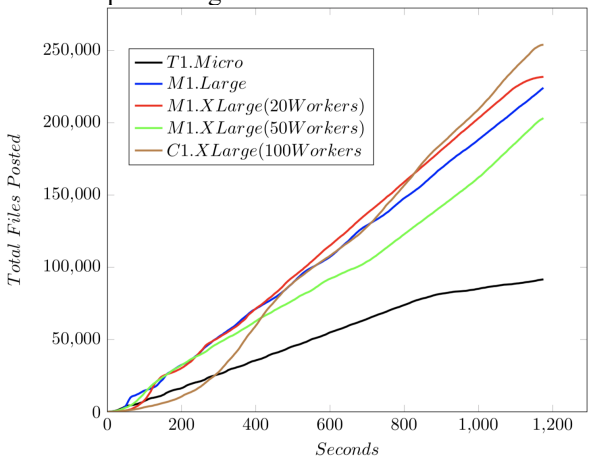


Figure 11. Multiple Server File Processing Rates over Time

To test the statistical significance of the increase in overall system performance, a set of statistical tests have been run on the AWS instance. Before running a correlation or a T-test, a test for normality of the data must firstly be performed. Taking two experiments, both using the FTP server and 10 workers per instance, in which the first has a single instance running and the second has 100 instances running, a density plot and the corresponding Normal Q-Q plot was performed showing that the data is normally distributed and that it is appropriate for running a correlation test and T-test.

The Pearson product-moment correlation coefficient is used to measure the dependence between instance numbers and files processed and the scatter plot along with the Pearson Coefficient is given in Figure 12 showing a strong and positive correlation between the number of instances and the number of files process.

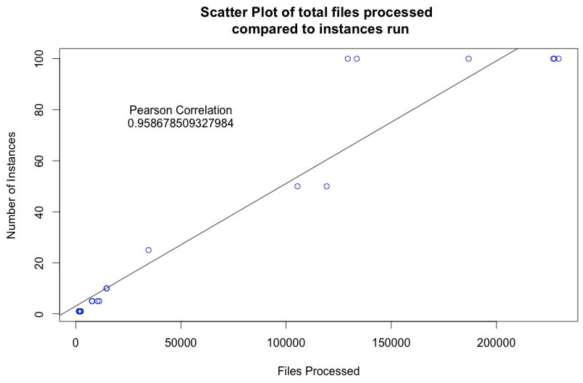


Figure 12. Scatter Plot of Files Processed vs Instances Run

The null hypothesis was then tested. The P-value is calculated to help determine if the null hypothesis should be rejected. The result of the one-way ANOVA is considered significant with a P value < 0.001, so a pairwise comparison was performed to test if the differences are statistically significant, while adjusting for Type 1 errors. The results of the pairwise test give a p-values < 0.001 in most cases, it can be concluded that there is a statistically significant difference comparing instance numbers to files processed.

6. Conclusion and Future Work

The final experiment demonstrated that horizontal scaling of all primary components is possible, and that this approach will ensure system bottlenecks are overcome once the data and servers are distributed. Statistical significance was also demonstrated between the number of instances and the files processed.

While funding limited the ability to run additional experiments, the result of this final experiment was such that 192TB of data per twenty-four hours processing could be achieved, with evidence that further improvement would be possible through the use of more instance types being fed by more web servers data sources.

Further optimizations of the pipeline are considered as possible future areas of research. Much of the experimentation performed in this paper demonstrated the extensive capability of a distributed system and identified the key factors within the system. It is possible to take these factors and monitor them such that a machine learning approach could be used to optimize a running system by monitoring the overall efficiency of the server data processing, taking into account the web server capabilities, the networking performance the capacity of the CPU.

To significantly reduce the overall data movement where live telescopes are being used, data processing at the telescope site, using a GPU system, could result in the transmission of processed data, instead of the raw image. Work on light curve generation within the pipeline could also be incorporated into the worker nodes. Further research would be required into data reduction at the source of data production which would ensure that the NIMBUS pipeline could increase the overall processing rates by changing the ratio between data movement and data processing.

10. References

[1] Merriam-Webster. The Merriam-Webster dictionary. Merriam-Webster, Springfield, Mass, 2014.

[2] Robert R Newton. The crime of Claudius Ptolemy. Baltimore : Johns Hopkins University Press, 1977.

[3] Steve B Howell. Handbook of CCD astronomy, volume 5. Cambridge University Press, 2006.

[4] T. B. McCord and J. P. Bosel. Potential usefulness of CCD imagers in astronomy. In Charge-Coupled Device Technology for Scientific Imaging Applications, pages 65–69, June 1975.

[5] M J Graham. Astronomy 2020: A Pragmatic Approach. Astronomical Data Analysis Soft- ware and Systems XVIII, 2009.

[6] H Ferguson, Perry Greenfield, Tim Axelrod, Stefi Baum RIT, Alberto Conti, Dennis Crab- tree, Eric Feigelson, Mike Fitzpatrick, Wendy Freedman, Kim Gillies, et al.

Astronomical data reduction and analysis for the next decade. The Astronomy and Astrophysics Decadal Survey, 2009.

[7] A Fowler, P Waddell, and L Mortara. Evaluation of the rca 512x320 charge-coupled device (ccd) imagers for astronomical use. In Solid State Imagers for Astronomy, pages 34–44. International Society for Optics and Photonics, 1981.

[8] Radu Corlab. Gcx user's manual @ONLINE. http://astro.corlan.net/gcx/html/node7. html, December 2004.

[9] P Massey and G H Jacoby. CCD data: The good, the bad, and the ugly. Astronomical CCD observing and . . . , 1992.

[10] Mark Adams. Stellar magnitudes from digital pictures. AURA, Association of Universities for Research in Astronomy, 1980.

[11] Otto Klotz. Magnitude of Stars. Journal of the Royal Astronomical Society of Canada, 15:289, October 1921.

[12] RJ Hanisch and Peter Quinn. The international virtual observatory. Space Telescope Science Institute, Baltimore, MD, 21218, 2003.

[13] D C Wells, E W Greisen, and R H Harten. FITS - a Flexible Image Transport System. Astronomy and Astrophysics Supplement Series, 44:363, June 1981.

[14] Paul Doyle, Fred Mtenzi, Niall Smith, Adrian Collins, and Brendan O'Shea. Significantly reducing the processing times of high-speed photometry data sets using a distributed computing model. In SPIE Astronomical Telescopes+ Instrumentation, pages 84510C–84510C. International Society for Optics and Photonics, 2012.