Internet of Things and Cloud Computing
2013; 1(1): 5-14
Published online August 10, 2013 (http://www.sciencepublishinggroup.com/j/iotcc)
doi: 10.11648/j.iotcc.20130101.12

Processing intensive full-waveform aerial laser scanning Matlab jobs through condor

Fanar Mansour Abed

Surveying Engineering Department, College of Engineering, University of Baghdad, Baghdad, Iraq

Email address:
fanar_mansour@yahoo.com

To cite this article:
Fanar Mansour Abed. Processing Intensive Full-Waveform Aerial Laser Scanning Matlab Jobs through Condor. *Internet of Things and Cloud Computing.* Vol. 1, No. 1, 2013, pp. 5-14. doi: 10.11648/j.iotcc.20130101.12

Abstract: Full-waveform aerial laser scanning is a laser system that records the entire backscattered signal of the laser pulse and stores it in the system recorder for post-processing. Capturing the complete waveform of the backscatter signal enables distinguishing between neighborhood echoes of a range smaller than the pulse length. Full-waveform has shown potential to better describe land cover features through the additional physical information it can provide alongside the standard geometric information. To fully utilize full-waveform for enhanced object recognition and feature extraction, it is essential to develop an automatic and effective routine to manage and process full-waveform datasets in a manner which requires less human effort and reduces time needed to process large laser datasets efficiently. This research tackled this problem through introducing a novel processing strategy for full-waveform data based on a developed pulse detection methodto run through Matlab environment. The solution adopted a grid computing Condor-based approach, which showed significant potential to reduce the time and effort needed to process large datasets such as full-waveform aerial laser scanning to more than 300% in specific conditions.

Keywords: Laser Scanning, Lidar, Full-Waveform, Signal Analysis, Grid Computing, Condor

1. Introduction

Topographic aerial laser scanning (ALS) is a laser scanning system mounted on an airborne platform and provides range measurements to the Earth's surface. It delivers a vertical accuracy of 5-10cm and decimeter-level plan metric accuracy, depending on the terrain type and flying height [1,2]. Land cover features are scanned either from a fixed wing or a helicopter platform in order to collect the necessary information to model the topographic surface [3]. A digital camera is usually flown alongside the ALS systems for orthophoto generation and data integration. This can be achieved by processing the photogrammetric information alongside the laser information to support laser range data for subsequent optimal interpretation.

This can play a significant role in identifying land cover features in cases where the objects are found to be difficult to interpret from the blind range data only. However, this requires some extra effort to register and georeference the sensors after data collection and through post-processing analysis [3]. Topographic ALS systems are now fully operational to meet the needs of specific applications such as surface modeling [2]. In contrast to photogrammetry, ALS is an active system, which can operate during day and night. Both ALS and photogrammetry are able to produce a digital terrain model (DTM).With regard to the technical physical principles, ALS can be grouped into two main systems, discrete-return and full-waveform (FWF) [4,2]. Both systems are designed to estimate range measurements using the physical concept of the pulsed laser mechanism. When the receiver only provides the start and the end of a signal at a certain rise time of the echo, then the system is called discrete-return. However, if the complete digital signal is digitized with extra information about the echo shape, then the system is called full-waveform.

FWF-ALS is unlike the discrete-return systems in that it records the entire backscattered signal of the laser pulse and stores it in the system recorder for post-processing [5,6]. Furthermore, complex and weak laser echoes can be detected towards improving modeling products such as DTMs [7]. In contrast with the discrete-return systems, which provide end users with a single range measure to the ground target, FWF stores the entire time history of the backscatter signal with a high sampling resolution as shown in Figure 1 [3]. This gives the user the opportunity to model the received signal and applying a function that better fits with the physical trend towards robust range estimations and accurate data modeling [8].

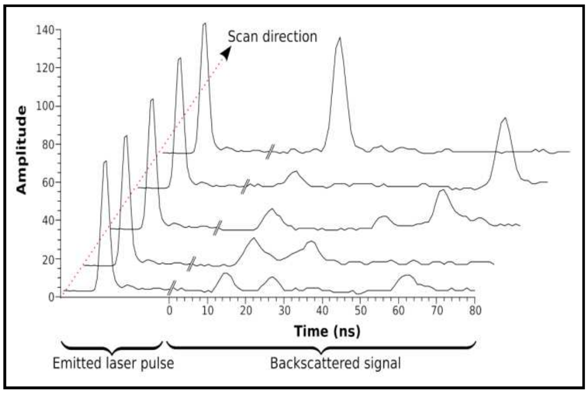

Figure 1.*Simulation of raw FWF data show five emitted signals alongside their received backscatter digitized signal [6].*

FWF analysis and post-processing may increase the accuracy and the resolution of the range measurements by providing end users with the chance to interpret the physical backscatter signals of the individual pulses [9]. This is achievable through pulse detection methods which give the user a significant opportunity to select the function which best fits the signal. In contrast with the discrete systems, FWF is applicable to determine the errors acquired from the limitations in the standard pulse detection methods that lead to inaccurate range measurements [7]. Following waveform post-processing, denser point cloud data are generated than those delivered from the discrete systems. This deliver a great potential for the most land cover applications towards optimal data modeling [6]. FWF analysis also provides additional information about the physical backscattering properties of the illuminated targets [10, 11]. The pulse width of the echo delivers information about surface roughness, slope, scan angle, or the depth of the volumetric object, while echo strength (amplitude) delivers information about target backscatter properties [12].

2. Literature Review

2.1. Post-Processing of Full-Waveform ALS Data

It is evident from the signal analysis literature that pulse detection is considered to be a challenging task with respect to retrieving information in a geometric form (e.g. 3D point clouds) [11,13]. Several methods have been developed to detect echoes from FWF signals. This includes threshold, constant fraction, peak, and center of gravity (COG) detection methods [13]. However, each method has its own weaknesses when applied to small-footprint FWF-ALS data, and may limit the final range accuracy [13]. Therefore, for high accuracy range resolution, it is necessary to adopt more sophisticated pulse detection methods.

Gaussian decomposition is a popular pulse detection technique to model laser waveforms of approximate Gaussian distribution [12,5]. It was found that the Gaussian function (1) can best describe, and therefore effectively model, the small-footprint FWF-ALS data from the Riegl LMS-Q560 system [12,5]. Therefore, the Gaussian decomposition technique can be used fit the Gaussian function to the Riegl LMS-Q560 signals in order to detect all possible echoes within individual waveforms.

$$y = N_{level} + \sum_{i=1}^{n} A_i \, exp\left[-\left(\frac{x-x_i}{width_i}\right)^2\right] \qquad 1$$

Where:
y is the quantized amplitude values
N_{level} is the noise level in the waveform signal
n is the number of Gaussians
A_i is the amplitude of the i^{th} Gaussian
x is the samples time values
x_i is the i^{th} Gaussian peak
$width_i$ is the pulse width of i^{th} Gaussian = $\sqrt{2} * \sigma_i$
σ_i is the standard deviation of the i^{th} Gaussian

The Gaussian pulse detection approach presumes that the laser pulses are transmitted with a Gaussian-like distribution and thus the received signal can be treated as a sum of multiple Gaussian pulses [12]. It aims to detect multiple echoes from individual waveforms as an amplitude-against-time measure. As a result of fitting the waveforms to the Gaussian function, multiple echoes can be detected and geometric and physical information can be extracted for individual echoes (Figure 2). This method can deliver accurate range resolution and provide a reliable solution as compared with other available methods. Consequently, accurate geo-referenced 3D point clouds alongside echo width and amplitude measurements can be provided to end users for individual echoes in addition to the total number of detected returns [14]. However, the Gaussian decomposition method is considered to be problematic in the case of complex and weak waveform signals [12,10].

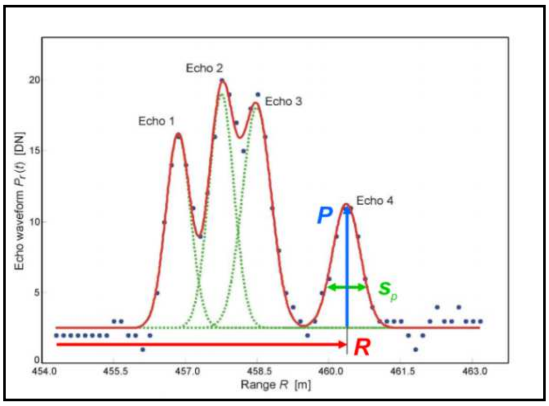

Figure 2. *Fitting Gaussian function to detected FWF signal from the Riegl LMS-Q560: (Blue dots) recorded waveform signal; (Green dotted lines) fitted Gaussians per echo; (Red lines) sum of all Gaussian functions; (R) is the range to the sensor; (P) is the echo amplitude; (S_p)is the echo width [15].*

Motivated by overcoming the weaknesses in the available pulse detection methods, a Rigorous Gaussian pulse Detection (RGD) method was developed by reference [7]. This was motivated by the need to develop a sophisticated and reliable method, which could decompose complicated waveform signals. RGD was originally designed to improve range resolution and accuracy and overcome information loss due to limitations in range estimations from standard approaches [7]. The RGD approach also provides a solution to tackle complex overlapping waveforms and difficulties in detecting weak signals [13]. These two limitations are considered to be the main challenges faced by the standard pulse detection methods. RGD is an iterative technique based on Gaussian decomposition definition. It is implemented with rigorous initial values and applies a sophisticated iteration procedure [13]. The method can detect overlapping signals in complex waveforms by analyzing the second derivative of the Gaussian function. Figure 3 demonstrate the detection of visible and overlapping peaks with RGD from complex waveforms. However weak pulses are detected based on analysis of residuals derived from the least squares fitting procedure and the pulse width value delivered from individual echoes [7].

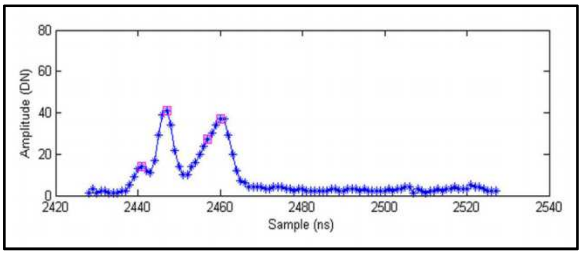

Figure 3. *RGD methoddetecting overlapping peaks in a complex waveform from the Riegl LMS-Q560 system [13].*

The method has proven to be capable of extracting a greater number of valid echoes from individual laser pulses than those extracted through standard available approaches [7]. Figure 4 demonstrates the performance of the RGD method against two popular algorithms available from leading commercial software. It shows that RGD outperforms the comparator algorithms by delivering extra valid echoes for DTM and canopy modeling applications as highlighted by red dots. Both complex and weak waveform pulses can be resolved with the RGD approach, which was validated through comparison to ground truth data. Refer to reference [7] for further details. Adding to this, the improved range resolution results from the RGD method shows particular potential when the vertical separation between targets is small (less than one pulse length). This can reduce the chance of overestimating and underestimating classification procedures and deliver more robust products such as DTM or canopy height models (CHM) [13].

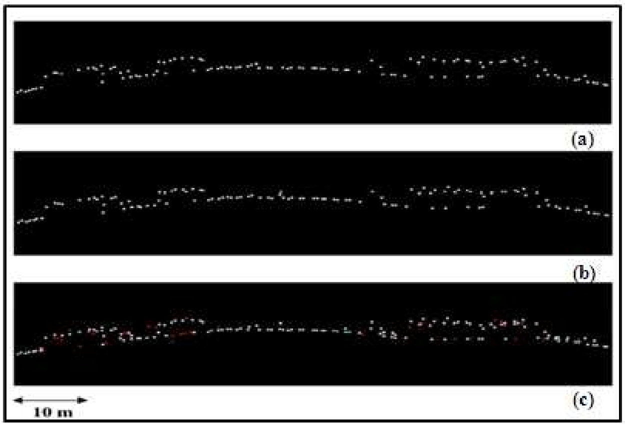

Figure 4. *RGD pulse detection method vs. algorithms from commercial software assessing an 80m profile section in a vegetated area: (a) center of gravity(COG) method (b) Gaussian pulse fitting (GPF) method (c) RGD method [13].*

In addition to the 3D object location of multiple laser echoes, output from the RGD software, which was developed in-house at Newcastle University/ UK, delivers backscatter properties, including echo amplitude and width parameters. These parameters have demonstrated a capability to enhance available classification/filtering algorithms [16, 17]. However, managing and processing dense FWF datasets is a challenging task [18,19]. This is because of the dramatic increase in the data volume due to unlimited return echoes and delivers a serious problem in computing processing and storage capacity. However, organized management of computing resources can effectively improve computing usage, but more human effort is required [20]. Therefore, there is a high demand for an effective processing tool or strategy to reduce processing time for laser scanning data.

2.2. Handling FWF-ALS Data

The efficient handling of laser scanning data has been an issue since FWF systems first emerged. The stored waveform profiles delivered from a FWF sensor after a flight campaign of 1.6 hours with a mean pulse repetition frequency (PRF) of 50 kHz can occupy about 140 GB [18]. Therefore, managing this huge amount of information for further processing is considered to be a challenging task in the context of large flight campaigns.

For efficient data management, effective resources should be utilized, where resources here refer to processor, memory, and disk space. There are three main types of resources available for processing approaches: single processor, symmetric multi-processor, and distributed processor. With the single processor, only one central processing unit (CPU) is used, with centralized memory and disk space. The symmetric processor (e.g. quad-core, dual-core) is efficient to use in some cases when the dataset is relatively large as the user can take advantage of the multi-core system design. The limitation of the symmetric processor is that all CPUs share the same memory, which can significantly slow the processing. However, the distributed processor system is considered to be the most efficient solution to run massive datasets, as all processors have their own dedicated memory but have to communicate with each other to access the centralized memory resource (e.g. network) [21]. Figure 5 shows the three mentioned resource types used in data managing and processing.

Reference [22] developed the software package, OPALS, as a complete set of processing tools for ALS datasets. This software included multiple modules to process and visualize ALS data, including FWF for various applications. It is provided in Shell script for Linux users, patch for Windows, and also in Python code [23]. As the software modules are originally designed to run under a Linux environment, no graphical user interface (GUI) is available. The software can be downloaded from [24]. However, the software is originally designed for use with both single and symmetric processor types.

Reference [18] introduced a toolkit called FullAnalyze to visualize and process laser scanning datasets, including FWF, as a 1D signal or in 3D point cloud format. This software is also applicable for either single or symmetric processor types and it was released in October 2009 as open source software [25].

The software runs under a Linux environment, and requires a virtual Linux environment to be installed on the machine if the user decides to operate under Windows [25]. Further, all available commercial software designed for laser scanning data management and processing, such as TerraScan from Terrasolid [26] or RiAnalyze from Riegl [27], are designed to run under high specification single or symmetric processors.

On the other hand, [19] demonstrate the potential of using a symmetrical (multi-core) processor to manage laser scanning datasets and improve performance against the single processor scenario. They processed raw laser scanning data by partitioning these data into multiple blocks, which have been interpolated individually and finally merged into one integrated Digital Elevation Model (DEM). Their approach achieved a powerful increase in speed of performance, and effective reduction in the overall processing time. However, the proposed technique is not the optimal solution for massive datasets.

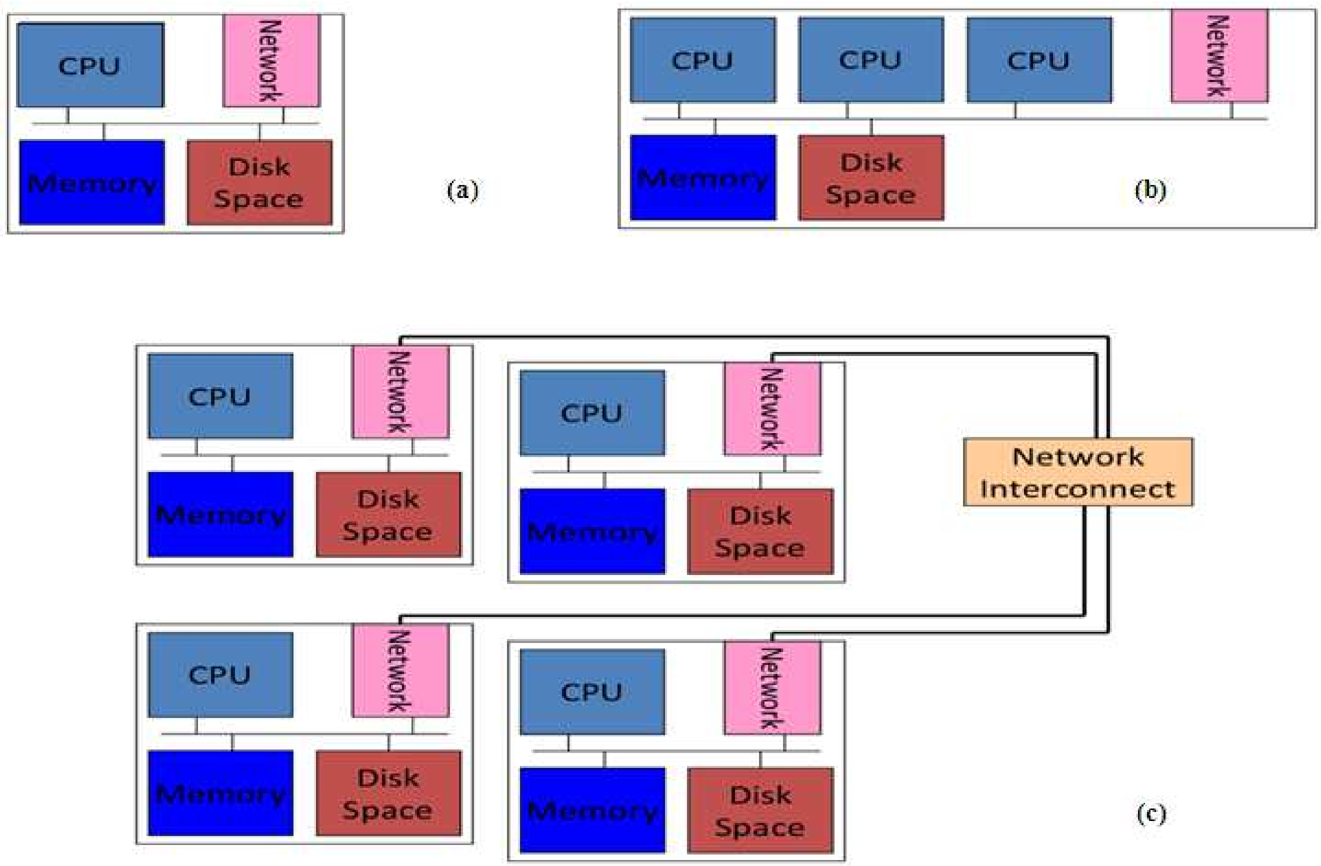

Figure 5. *Resource types used in data management and processing: (a) single processor (b) symmetric multi-processor(c)distributed processor [21].*

The most efficient processor system which is extremely powerful for handling massive datasets is the distributed system [28]. The key leveraging in the distributive resources is its compatibility to run independent parallel jobs on different physical computers and later integrate these jobs to deliver one combined solution. It is best suited to manage multiple independent processing operations where individual tasks are highly independent and do not require input from other operations. In this context, this mechanism can therefore efficiently meet the requirements for laser scanning data processing. The most powerful processing technique which can efficiently manage the distributed processor system of large-scale resource sharing, is grid computing mechanism [29,20]. Grid computing is a technique of combining multi-computer resources to achieve one single goal [28]. It provides consistent, independent and flexible access to intensive computational capabilities in the presence of network connection to maintain the necessary resources [30]. The ideal solution to manage, process, visualizes, and also monitor powerful distributed jobs of massive datasets is through Condor. Condor is a well-known computing project which provides resource optimization and support for high- throughput computing techniques including grid computing on large-scale distributed computing resources [31]. It can run a large number of jobs concurrently and provide high quality service with a high network usage [32]. Submitting jobs through Condor can be designed as illustrated in Figure 6. The process starts by designing a robust code that solves the current problem and ensures that the code is well designed to run in the Condor environment before submission. In certain cases, a compilation is needed against middleware on the Condor server, which can be managed through a third-party tool. Following a well-structured submission plan, Condor can deploy the independent jobs to the computing resources within the network. So far Condor has proven to be extremely effective in improving the productivity of massive datasets [33, 34] and significantly reducing the time needed to generate the results [35], which is an important factor in the case of laser scanning datasets.

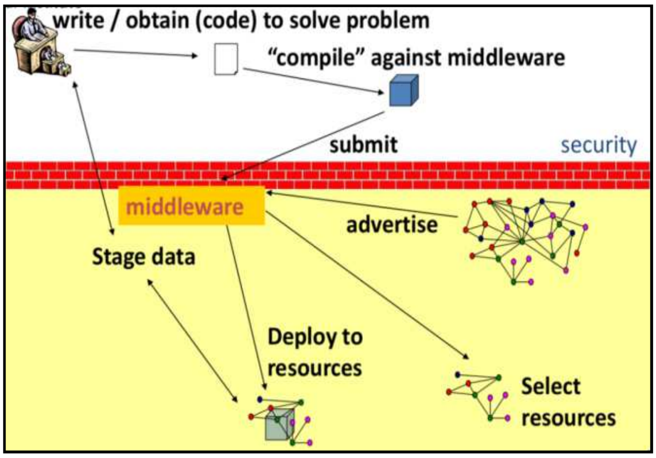

Figure 6. *Principles of the Condor project [21].*

3. Dataset and Study Site

Small-footprint FWF-ALS dataset was utilized to develop the methodology in this research. The data was captured with a 1550 nm wavelength Riegl LMS-Q560 scanner. The technical specifications of this system are described in the Table 1.

Table 1. *Technical specifications of Riegl LMS-Q560 FWF-ALS system (Riegl, 2009).*

Laser wavelength	1.5 µm
Laser beam divergence	≤ 0.5 mrad
Scanning mechanism	rotating polygon mirror
Scan pattern	parallel scan lines
Scan angle range	± 22.5° = 45° total (± 30° = 60° total)
Scan speed	10-160 lines/sec
Angle measurement resolution	0.001°
Laser pulse repetition rate	up to 120 kHz @ 45° scan angle
Footprint size	0.5 m @ 1 km
Pulse width at half maximum	4 ns
Minimum range	30 m
Intensity measurements	16 bit intensity information is provided for each echo signals

The study site investigated and covered by this dataset is located on the south coast of England. It includes an urban area over Bournemouth city Centre, with additional data acquired at a rural site (Hurn) located to the north-east of the city and composed of natural terrain cover with various landforms. Figures 7 show the extents (red lines) of the Bournemouth study site, together with their flight lines marked in grey lines.

The Bournemouth dataset is composed of nineteen flight lines with an average flying height of 350 m and was collected from a helicopter platform in May 2008. It offers a high point density with more than 15 points/m^2 and a 0.18 m footprint diameter size. The swath width and scan angle of the Bournemouth dataset is ~430 m and ±30° respectively. The dataset was directly geo-referenced through an on-board GNSS-IMU system. The Bournemouth dataset has been assessed as having an average RMS accuracy of 0.09 m in the urban area and 0.12 m in the rural area (refer to [13]). The dataset, together with the trajectory information and orthophoto coverage, were provided by Ordnance Survey, Great Britain's national mapping agency.

In Bournemouth, nine flight lines (1-9) were captured for the rural site at Hurn while ten flight lines (10-19) cover the urban site. These flight lines which occupy more than 50 GB storage and capsulated millions of points have been processed and analyzed to test the research methodology herein.

Figure 7. *Bournemouth study site, with red blocks representing the ground coverage. Urban area to the south-west and rural area to the north-east.*

4. Methodology

Motivated by overcoming limitations in available post processing techniques, RGD method was selected as the optimal pulse detection algorithm to post process FWF-ALS data in this research. For detailed information about the implemented routine in terms of assumptions, parameters, and thresholds adopted for the Bournemouth dataset, refer to [13].

Due to processing complexity, which stems from large datasets and the substantial number of echoes that the RGD method can detect [13], an effective processing strategy was developed using a grid computing technique. The new routine relies on high-throughput computing utilizing the Newcastle University PC network and taking advantage of Matlab functionality provided through the Matlab Distributed Computing Server [35]. Grid computing provides the opportunity to run large numbers of independent jobs concurrently [20,19]. This technique can be implemented through utilizing the intelligent processing Condor project [31,32].

Condor provides a powerful job invocation environment which is capable of successfully executing large sets of parameter sweep jobs. Parameter sweep operations relate to the execution of many similar jobs, which are run by changing only the input parameters. Therefore, adopting a Condor-based approach was essential in order to feasibly process the datasets in this research. Condor provides the ability to perform check pointing and migration of executions on remote computers where inputs and outputs from a user program are staged back to the submitting computer. Check pointing is an intelligent application which is often used in grid computing to save intermediate data on a reliable storage for a period of time during long term processing. This technique is basically used to recover the run in case of job failure rather than restart the application from the beginning [36]. This can effectively save computing time and provide more elastic processing workflow in cases of complicating computation and large datasets. However, this requires the user to compile his/her own code alongside the Condor libraries which run under a UNIX based operating system. This is something which is not always possible - such as when using a commercial package like Matlab. Therefore, it would be desirable to provide some equivalent functionality to check pointing, to help reduce failed execution time in Condor.

For effective run time reduction, the execution routine implemented within Condor in this research is based on two main aspects. Firstly, the data can be used many times as soon as it is staged to the Condor computer, where data staging is a managing process between the submitting and the remote computers for efficient grid deployment [37]. Secondly, data generated on the Condor computer can be staged back to the submitting computer as soon as possible. That means the routine has separated the data staging part of the Condor job submission from the job deployment phase and provides a mechanism for returning data to the user while the code is still running on the remote computer. The user is required to provide new logic in the form of how to process the returned data and how to deal with incomplete returns when the job is evicted before complete execution.

The application that is used within the Condor system to run the data is packaged into a compressed archive to facilitate transfer files and reduce time needed for submission. The Condor cluster at Newcastle is composed predominantly of ~1100 Windows computers (~3000 CPUs in total), thus the scripts are written as Windows Batch files. Therefore, these files are converted to UNIX shell scripts to be compatible with the Condor library's operating system. The 7zip archive format (see [31] for details) is used for data compression as this was already deployed across the Newcastle Windows clusters. As the clusters work under the Windows operating system, the server then starts to submit Condor jobs containing archive and Java client to the cluster, as Java Script code is written into an HTML page and can be read with any browser regardless the computer operating system [38]. To prevent excessive load on the server, the number of jobs that can be launched at any one time and the frequency at which these are launched is limited. As each job starts to request sub-jobs from the server, the server can deploy new jobs into Condor until the pre-defined limit is reached.

Following this, the Java client can request the next piece of work from the server side. As the link between data transfer and execution has now broken, and the data has already been staged to the server, requesting sub-jobs can be as small as possible with the client asking for further tasks without having to re-request data from the submitted computer or re-downloading the original dataset. The execution of the original application is invoked by Java which is able to send back the results of these sub-jobs to the server immediately on completion. The client is now able to contact the server for further sub-jobs and will terminate only when being instructed by the server or due to eviction of a Condor job from the host computer. Figure 8 illustrates the overall architecture of the developed routine.

The Matlab code has been compiled into a binary executable using the Matlab compiler. This code requires a number of configuration arguments. The data file, the index file which defines a unique index number for individual FWF echoes, the data point within this file to process, the path where to write the output and index data for this output file. All of these parameters will remain the same except for the data point to process. The shell script is passed through this index before invoking the Matlab executable. The code written for the client takes the output file and the index entry for this output file and returns them after successful execution.

The presented technique essentially reverses the normal Condor push job model (sending successive jobs without any interaction with the client after submission) into a client based pull model (an efficient interaction between the client and the server computers controlled by the client). This is particularly useful in situations where the user has large datasets which require significant time to distribute to worker nodes, allowing nodes which already have the data to keep on requesting sub-jobs until either evicted or all sub-jobs are completed. The developed approach was used to process FWF-ALS datasets (Bournemouth 2008). The introduced technique was capable of reducing run time by 100-300 % depending on dataset density and submission configurations, as it lends itself best to programs where a large data set is used repeatedly, which means that a large number of jobs can run concurrently on Condor.

5. Results and Discussion

The reason for using the RGD method to post-process FWF-ALS data in this research lies in its reliability as a pulse detection method to deliver higher range accuracy than other available routines. Furthermore, the RGD method was utilized because of its capability to resolve challenging signals and detect targets from complex overlapping waveforms. This leads to enhanced representation of land cover features by translating the complete received energy into rigorous spatial and physical information. This algorithm has already been demonstrated to deliver a greater number of valid echoes from individual waveforms than those delivered from the standard approaches, leading to better descriptions of surface features. Thus, more accurate 3D data for diverse applications can be delivered, including terrain generation, 3D modeling, forest mapping, etc.

As the proper processing strategy to deliver optimal outputs, high-throughput computing was adopted to run the RGD method. The potential of this type of processing can be identified as the capability to run independent parallel jobs on different physical computers and later merge their output to deliver one combined solution. Therefore, it saves time and cost needed to run successive jobs on single computer with limited memory size, which can slow processing down considerably. This is basically relying on the efficiency of accessing independent memories and using a network for communication. Whereas the processing time is an important factor here to deliver FWF point cloud, the Condor workflow was adopted.

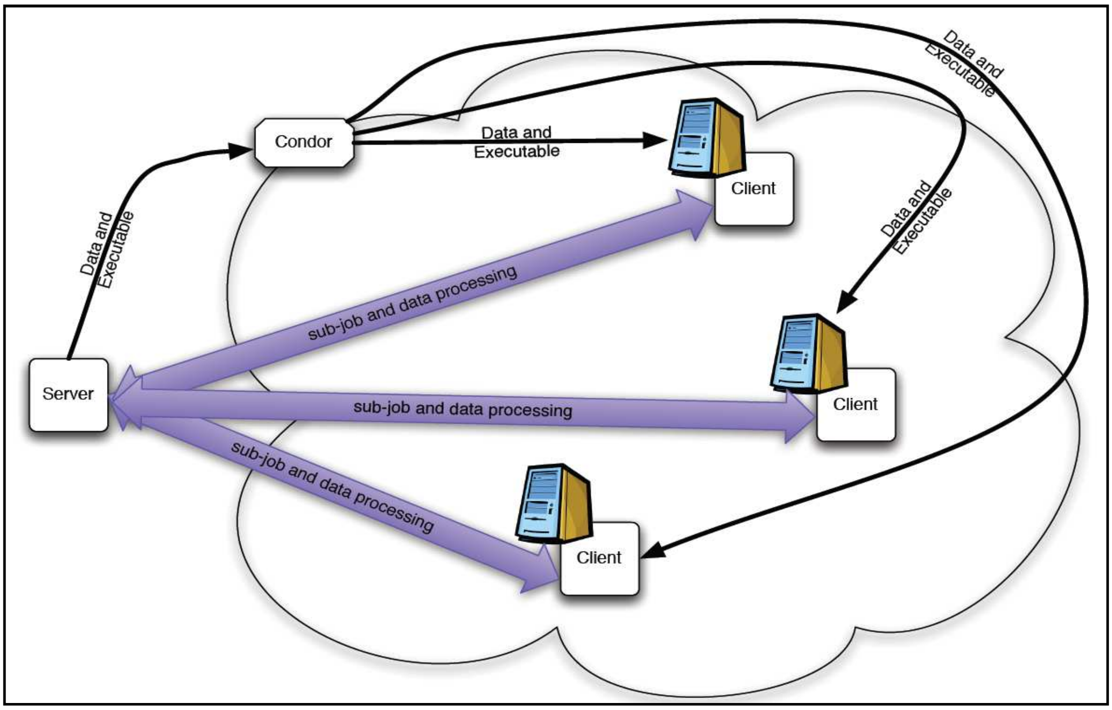

Figure 8. *The general architecture of the developed Condor-based FWF-ALS data post processing.*

However, with Condor based student clusters, such as the case in Newcastle University, large "cycle stealing" can occur, as the computers are not dedicated for high-throughput computing only. In order to reduce network usage and the wastage of computation time that occurs in cases of eviction by student login, the new technique of returning data to the user while the code is still running on the remote computer was developed, as explained in the methodology section. The routine was developed in cooperation with the Digital Institute in Newcastle University to tackle problems in managing large datasets through the Condor system, which had less than six years history in the University network.

This technique was modified to fulfill the needs for a model that keeps on processing until all jobs are completed, thus no data is lost due to computer eviction. As check pointing can provide reliable data storage for a period of time, the developed pull mode helped the user to re-request a new job upon eviction without wasting time to re-data transfer. As a result, this pull model allowed the running of a large number of jobs without the need for shared file space. This provides an advantage of using the free time on worker nodes more dynamically, such as in the case of non-dedicated Condor environments, which has the potential to run large datasets. Furthermore, the developed routine addresses and analyses any log run errors that might occur, which helps the user to diagnose the problem for a potential manipulation while the code is still running remotely. The project shows significant and powerful potential to speed up the processing records to more than 300%, which can potentially increase with customized settings and larger network usage.

6. Conclusions

A new effective processing strategy for FWF-ALS data has been developed using a grid computing Condor-based technique. The presented technique shows potential in situations where large datasets such as lidar data are utilized. This is achieved by reversing the normal Condor push job model into a client based pull model which helped to reduce processing time by 100-300 % in the case of the datasets investigated in this research.

The discussion of the FWF post processing and data management shows the potential of the RGD method to post-process FWF data to deliver more rigorous estimations from the waveform signal and provide the user with additional valid echoes. Further, the RGD adoption has shown reliability through the implementation of the Condor pull model, using high-throughput computing and producing savings in terms of time and cost exerted to process massive datasets. The routine contributes in reducing network usage and wasted computation time by providing a successful processing environment for non-detected Condor networks.

Acknowledgment

The author wish to acknowledge the assistance of Ordnance Survey (the greatest mapping institute in the UK) for providing dataset used in this research. The author would also like to acknowledge Newcastle University to provide materials needed to develop the processing technique and appreciate help and support from people in the Digital institute.

References

[1] F. Ackermann, (1999), 'Airborne laser scanning- present status and future expectations', *ISPRS Journal of Photogrammetry and Remote Sensing*, 54 (2-3), pp. 64-67.

[2] C. Mallet, and F. Bretar, (2009), 'Full-waveform topographic lidar: State-of-the-art', *ISPRS Journal of Photogrammetry and Remote Sensing*, 64 (1), 16 p.

[3] G. Vosselman, and H.-G. Maas, (2010), *Airborne and terrestrial laser scanning*. Scotland, UK: Whittles Publishing.

[4] J. Shan, and C. K. Toth, (2009), *Topographic laser ranging and scanning - principles and processing*. FL, USA: Taylor & Francis.

[5] B. Jutzi, and U. Stilla, (2006), 'Range determination with waveform recording laser systems using a Wiener Filter', *ISPRS Journal of Photogrammetry and Remote sensing*, 61 (2), pp. 95–107.

[6] F. Bretar, A. Chauve, J.-S Bailly, C. Mallet, and A. Jacome, (2009), 'Terrain surfaces and 3-D landcover classification from small footprint full-waveform lidar data: application to badlands', *Hydrology and Earth System Sciences*, 13 (8), pp. 1531-1545.

[7] Lin, Y.-C., Mills, J. P. and Smith-Voysey, S. (2010), 'Rigorous pulse detection from full-waveform airborne laser scanning data', *International Journal of Remote Sensing*, 31 (5), pp. 1303-1324.

[8] C. Mallet, F. Bretar, M. Roux, U. Soergel, and C. Heipke, (2011), 'Relevance assessment of full-waveform lidar data for urban area classification', *ISPRS Journal of Photogrammetry and Remote Sensing*, 66 (6), pp. S71-S84.

[9] M. Doneus, C. Briese, M. Fera, and M. Janner, (2008), 'Archaeological prospection of forested areas using full-waveform airborne laser scanning', *Journal of Archaeological Science*, 35 (4), pp. 882-893.

[10] Y.-C Lin, J. Mills, and S. Smith-Voysey, (2008), 'Detection of weak and overlapping pulses from waveform airborne laser scanning data', In: *Hill, R., Suárez, J. & Rosette, J. (Eds.), Proceedings of SilviLaser 2008*, Edinburgh, United Kingdom, 10 p.

[11] W. Mücke, (2008), *Analysis of full-waveform airborne laser scanning data for the improvement of DTM generation*. Master thesis.Vienna University of Technology, 67 p.

[12] W. Wagner, A. Ullrich, V. Ducic, T. Melzer, and N. Studnicka, (2006), 'Gaussian decomposition and calibration of a novel small-footprint full-waveform digitizing airborne laser scanner', *ISPRS Journal of Photogrammetry and Remote Sensing*, 60 (2), pp. 100–112.

[13] Y.-C Lin, (2009), *Digital terrain modelling from small-footprint, full-waveform airborne laser scanning data*. PhD thesis.Newcastle University, 205 p.

[14] W. Wagner, A. Roncat, T. Melzer, and A. Ullrich, (2007), 'Waveform analysis techniques in airborne laser scanning', *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 (3/W52), pp. 413-418.

[15] G. Mandlburger, C. Briese, and N. Pfeifer, (2007), 'Progress in lidar sensor technology-chance and challenge for DTM generation and data administration', In: *Proceedings of The 51st Photogrammetric Week*. Stuttgart, Germany, pp. 159-169.

[16] M. Doneus, and C. Briese, (2006), 'Digital terrain modelling for archaeological interpretation within forested areas using full-waveform lasers canning', In: *Proceedings of The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST2006)*. Nicosia, Cyprus, pp. 155-162.

[17] Y.-C. Lin, and J. P. Mills, (2009), 'Integration of full-waveform information into the airborne laser scanning data filtering process', *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (3/W8), pp. 224-229.

[18] A. Chauve, F. Bretar, S. Durrieu, M. Pierrot-Deseilligny, and W. Puech, (2009), 'FullAnalyze: A research tool for handling, processing and analysing Full-waveform Lidar data', In: *Proceedings of Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 200,,* Cape Town, South Africa, pp. 841-844.

[19] X. Guan, and H. Wu, (2010), 'Leveraging the power of multi-core platforms for large-scale geospatial data processing: Exemplified by generating DEM from massive lidar point clouds', *Computers and Geosciences*, 36 (10), pp. 1276-1282.

[20] S. McGough, P. Robinson, C. Gerrard, P. Haldane, S. Hamlander, D. Sharples, D. Swan, and S. Wheater, (2010), 'Intelligent power management over large clusters ', In: *Proceedings of Green Computing and Communications (Green Com 2010)*. Hangzhou, China, 8 p. :On CD-ROM.

[21] S. McGough, (2011), 'Parallel computing, myths and truths',

Available at: https://researchtools.ncl.ac.uk/portal/site/!gateway/page/!gateway-100 (Accessed: 24 Nov 2011).

[22] G. Mandlburger, J. Otepka, W. Karel, W. Wagner, and N. Pfeifer, (2009), 'Orientation and processing of airborne laser scanning data (OPALS) - concept and first results of a comprehensive ALS software', *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (3/W8), pp. 55-60.

[23] G. Mandlburger, N. Pfeifer, C. Ressl, C. Briese, A. Roncat, H. Lehner, and W. Mücke, (2010), 'Algorithms and tools for airborne lidar data processing from a scientific perspective', In: *Proceedings of The European Lidar Mapping Forum (ELMF) 2010*. Hague/ Netherlands: On CD-ROM.

[24] OPALS (2009), Orientation and processing of airborne laser scanning data. Available at: http://www.ipf.tuwien.ac.at/opals/opals_docu/index.html (Accessed: 18 Sep 2009).

[25] FullAnalyze (2009), Open source lidar software. Available at: http://code.google.com/p/fullanalyze/ (Accessed: 16 Dec 2009).

[26] Terrasolid. (2008), 'TerraScan User's Guid ', Available at: http://www.terrasolid.fi/en/users_guide/terrascan_users_guide (Accessed: 03 Feb 2009).

[27] Riegl. (2008), 'RiAnalyze 560 for Full Waveform Analysis user Manual', Available at: http://www.riegl.com/index.php?id=231 (Accessed: 4 March 2009).

[28] R. Raman, M. Livny, and M. Solomon, (1998), 'Matchmaking: Distributed resource management for high throughput computing', In: *Proceedings of The 7th IEEE International symposium on high performance distributed computing*. Chicago, USA, 7p.

[29] Y. S. Dai, M. Xie, and K. L. Poh, (2002), 'Reliability analysis of grid computing systems', In: *Proceedings of the 2002 Pacific Rim International Symposium on Dependable Computing*. Tsukuba-City, Ibarski, Japan, pp. 97-104.

[30] F. Dong, and S. G. Akl, (2006), 'Scheduling algorithms for grid computing: State of the art and open problems', School of computing, Queen's University, Ontario, Canada, 55p.

[31] Condor (1988), Condor Team. Available at: http://www.cs.wisc.edu/condor/ (Accessed: 15 Oct 2009).

[32] M. Litzkow, M. Livny, and M. Mutka, (1988), 'Condor - A hunter of idle workstations', In: *Proceedings of The 8th International Conference on Distributed Computing Systems*. San Jose, California, USA, IEEE-CS, pp. 104-111.

[33] J. Dean, and S. Ghemawat, (2008), 'Map Reducer: simplified data processing on large clusters', *Communications of the ACM*,51 (1), pp. 107-113.

[34] D. Thain, T. Tannenbaum, and M. Livny, (2005), 'Distributed computing in practice: The Condor experience', *Concurrency and computation: practice and experience*, 17 (2-4), pp. 323-356.

[35] F. M. Abed, and S. McGough, (2010), 'Processing data intensive Matlab jobs through Condor', In: *Proceedings of All Hands Meeting 2010*. Cardiff, Wales, UK:On CD-ROM.

[36] J. Blythe, E. Deelman, Y. Gil, and C. Kesselman, (2003), 'Transport grid computing: a knowledge-based approach', In: *Proceedings of The fifth Innovative Applications of Artificial Intelligence conference (IAAI-03)*. Acapulco, Mexico, 8 p.

[37] W. Elwasif, J. Plank, and R. Wolski, (2001), 'Data staging effects in wide area task framing applications', In: *Proceedings of IEEE ISCC and the Grid Brisbane*, Australia, 9 p.

[38] T. Lindholm, and F. Yellin, (1999), *Java virtual machine specification*.2nd Ed. Boston, MA, USA Addison-Wesley Longman.